EJEMPLO DE LA UTILIDAD DE INTERNET PARA LA ENSEÑANZA ESPECIALIZADA UNIVERSITARIA

Rafael Albo Sánchez, Emilio Escobar Reyero, David Kau, Oscar Lozano Gutiérrez y Rafael del Pozo Barajas

SUMARIO:

Se analizan en este artículo distintas posibilidades de la red Internet para completar la formación de los estudiantes universitarios en nuestros centros. En concreto se muestran una serie de informaciones relativas a la Administración de Sistemas Informáticos que habrían sido imposible recoger de otro modo, ante la escasez de bibliografía sobre el tema. Si a esto añadimos que dicha información afecta a la seguridad de los sistemas informáticos, que es justo la tarea prioritaria de los diplomados en este módulo, se comprende la importancia transcendental de la incorporación de estas nueva tecnología de comunicación a nuestra formación.

SUMMARY:

This item analyses the different ways in which the Internet network can be used to complete the education of University students at our centres. It specifically divulges information related to Computer Systems Management that would not have been possible to collect through any other means due to the lack of bibliography concerning this issue. If we also add that this information affects computer system security which is exactly the main task of the graduate students included in this module , one understands the transcendental importance of incorporating this new media technology to our training.

Introducción

De unos años a esta parte la red Internet está alcanzando una gran expansión mundial, multiplicándose cada año el número de usuarios. Este incremento de difusión se ha visto favorecido principalmente por el uso de tecnologías hipermedia, que permiten presentar en la pantalla de nuestro ordenador texto, imágenes, sonido e incluso vídeos. Si a esto le unimos la posibilidad de interacción con el usuario nos encontramos con un nuevo medio de comunicación muy atractivo, tanto para los usuarios, que pueden obtener casi cualquier información (o así lo creen), como para las empresas, que pueden ofrecer publicidad individualizada a cientos de miles de personas interesadas en sus productos, a precios muy bajos.

Sin embargo, la gran difusión alcanzada por la Red no parece que haya aumentado su utilidad respecto a la antigua Internet, que sólo hacía uso de textos sin más, al menos en lo que respecta a la faceta universitaria. Esto se debe, a nuestro juicio, a dos problemas principales que hemos detectado: En primer lugar, en la mayor parte de los casos, los nuevos usuarios tienen más interés en ver la composición de las páginas que en su contenido informativo. En segundo lugar, la gran mayoría de transferencias de información a través de Internet tiene carácter comercial, y no científico, al contrario de lo que ocurría hasta hace pocos años. Como resultado tenemos una red que, aunque contiene gran cantidad de información, ésta es difícil de obtener debido a que se confunde entre un gran volumen de datos irrelevantes.

Este problema puede resolverse de dos formas: utilizando los servicios de Internet todavía no invadidos por la publicidad, o acudir a personas que ya hayan seleccionado previamente la información interesante en cada área. En este artículo hemos tratado las dos posibilidades, utilizando el servicio de News, y seleccionando determinadas páginas WWW que aportan información muy interesante para los futuros diplomados en Administración de Sistemas Informáticos. Hay que señalar que toda esta búsqueda, y la mayor parte del artículo, ha sido realizada por los alumnos de ASI que lo firman. Aunque ésta última parte tiene utilidad casi exclusiva para la rama de ASI, la primera, la del servicio News, puede ser interesante para cualquier otra carrera.

El servicio News

Este servicio consiste en una imitación de un tablón de anuncios, en el que cualquier persona que lo desee puede poner el suyo, siendo leído por un gran número de interesados que pueden dejar otro anuncio (o mensaje) como respuesta. La utilidad aumenta al haber varios tablones, cada uno con un tema específico, lo que permite que los internautas sólo lean aquellos artículos que les interesen. Hay gran variedad de tablones, llamados en realidad *grupos de noticias*, apareciendo en el servidor de news de la Red Informática Científica de Andalucía cerca de diez mil.

Para utilizar este servicio es necesario estar dado de alta en un proveedor de servicios de Internet; en el caso de los miembros docentes de la comunidad universitaria andaluza, éste puede ser el Centro Informático Científico de Andalucía, que presta sus servicios gratuitamente. Si ya se cumple este requisito, es necesario disponer de un cliente de news (el programa que nos permitirá leer y enviar mensajes al servidor), y acceso a un servidor de news (el programa residente en el ordenador del proveedor que se encarga de recibir nuestros mensajes, almacenarlos, y enviarlos a todos aquellos que lo soliciten). Normalmente el acceso al servidor de news se le da al usuario cuando se suscribe al proveedor; el programa cliente puede cogerse gratuitamente realizando un FTP anónimo a cualquier servidor FTP público. La explicación de estos términos y procesos quedan fuera del objetivo de este artículo, ya que serían necesarios varios artículos de gran extensión para explicar cada uno de ellos con detenimiento. Sin embargo, normalmente el proveedor de Internet se encargará de instalar todo lo necesario en el ordenador del usuario para que éste pueda utilizar dichos servicios sin dificultad.

Una vez conocido el servicio, queda por ver su utilidad para la investigación y aprendizaje. La principal ventaja de este servicio es que nos permite relacionarnos con un gran número de personas que están interesadas en el mismo campo, sin necesidad de conocerlas previamente. Obviamente la utilidad dependerá de las personas que se conecten al servicio, principiantes o expertos, y de su disposición a enseñar o difundir sus conocimientos. Normalmente en un grupo hay de los dos tipos de personas, tanto principiantes como expertos, y unos requerirán ayuda y otros la prestarán; lógicamente nosotros

deberemos prestar ayuda cuando podamos hacerlo si queremos que otros nos ayuden cuando lo necesitemos.

Quizá al comienzo sólo podamos pedir ayuda, si nuestro conocimiento es escaso, pero entonces quedaremos en deuda con la comunidad Internet y se supone que tendremos que saldarla en un futuro difundiendo nuestro conocimiento, bien sea a través de la Red, bien mediante artículos, libros o ponencias. Estas normas de comportamiento no escritas son las normales en cualquier sociedad educada y su seguimiento queda al libre albedrío del usuario, ya que nadie nos exigirá su cumplimiento. En realidad las normas de comportamiento educado en la red (la etiqueta de la red) sí que están escritas, aunque con la aparición de la WWW, en la que sólo se recibe información y no se aporta nada, están cayendo en desuso. Los interesados en estas normas puede buscar información en la Red sobre **netiquette** (por ejemplo en http://www.ezine.com /eznetiquette.html), incluso hay un grupo de noticias dedicado a la misma.

Dependiendo del grupo de noticias los usuarios predominantes serán de un tipo u otro, aunque lo normal es que siempre haya algunos que pueden y quieren aportar soluciones. En los casos que más hemos utilizado, las áreas de mensajes relacionadas con el Intercambio Electrónico de Documentos, el nivel de los interlocutores era muy alto. Como anécdota podemos señalar un agrio debate sobre lo que querían expresar algunas normas de las Naciones Unidas sobre EDIFACT, en el que varios interlocutores exponía sus ideas de forma más o menos educada; la discusión se saldó cuando uno de los participantes hizo saber que él pertenecía al grupo de Naciones Unidas que había redactado la norma, y dio la interpretación correcta. Esta ejemplo sirve para mostrar hasta qué punto es posible obtener información de gran utilidad mediante las News.

Centrándonos en nuestra diplomatura de Administración de Sistemas Informáticos y, en concreto, en nuestra asignatura de Fundamentos de Programación, hemos seleccionado algunos grupos de noticias que pueden ser de interés. Estos son: *alt.comp.lang.learn.c-c++, comp.lang.c.moderated, comp.lang.c, comp.lang.c++, comp.lang.c++.leda, comp.lang.c++.moderated, es.comp.lenguajes.c,* y *es.comp.lenguajes.c++*. Los grupos que empiezan por es. están en

español, y el resto en inglés; también hay grupos en otros idiomas, pero éstos no han sido aquí recogidos. Como muestra de lo que podemos encontrar en estos grupos transcribimos aquí una pequeña serie de mensajes de la última lista, todos respondiendo a la petición de ayuda del primero. Cada mensajes se ha reducido a un solo párrafo, y se le ha quitado toda la información propia del mensaje, como la fecha y el emisor.

P: Hola, Tengo una duda que no se si es especifico de programación o de C++. Alguien sabe que se entiende por "Callbacks". Gracias.
R1: Callbacks son funciones que utiliza Windows para responder a los eventos. Por ejemplo, cuando tu lanzas un timer, tu puedes no especificarle una función callback con lo que el timer enviara un mensaje WM_TIMER cuando expire el tiempo o, si le pones un función callback, la llamara. De todas formas esto ultimo es un ejemplo, porque creo que la posibilidad de poner callback a un timer no es posible en Win32 aunque si en Win16. Pero de todas formas quédate con el ejemplo, que creo que ilustra bastante bien lo que es una función callback en Windows. Un saludo.

*R2: El concepto de callback no pertenece a ningún lenguaje de programación. Es una técnica de programación que se usa en diversas situaciones. Básicamente consiste en la posibilidad de especificar un procedimiento que debe ejecutarse cada vez que se produce un cierto evento en el sistema. Este mecanismo tiene dos pasos. 1.- Registro del callback. Mediante una función de registro se especifica que función debe llamarse para responder a un cierto evento. 2.- Ejecución del callback. El sistema ejecuta esta función cada vez que dicho evento se produce. Aunque el uso de callbacks está muy generalizado en el desarrollo de interfaces gráficas de usuario (MS-Windows, X-Windows, OSF/Motif, ...) no es esta su única aplicación. Otros eventos para los que se suelen usar callbacks son: *Temporizadores, para ejecutar una tarea periódicamente. *Mensajes entre procesos. *Modificaciones en algún dato de la aplicación. Aunque este mecanismo se ha usado durante mucho tiempo con éxito, dentro de los entornos orientados a objetos se*

pueden utilizar mecanismos mejorados. Estas mejoras suelen basarse normalmente, en el patrón de diseño COMMAND (Gamma et al.). Para el caso concreto de las comunicaciones entre procesos, es interesante ver los mecanismos usados en el entorno ACE (Doug Schmitt).

R3: CallBack, significa "marcha atrás", y por convenio, se utiliza para las funciones que van a ser llamadas por Windows 95. Por ejemplo, las funciones que gestionan las ventanas, son CALLBCAK, ya que es Windows el que las llama para pasarles los mensajes que necesite.

La serie de mensajes aquí presentada es una muestra de la información que puede obtenerse sin demasiado esfuerzo mediante este servicio, aunque queremos señalar que no siempre la respuesta es tan útil, incluso hay veces en las que nadie responde. La utilidad real de cada grupo de noticias sólo la conoceremos cuando estemos suscritos a él durante un tiempo.

Páginas de World Wide Web

Como ya comentamos anteriormente, el problema que tiene el uso del servicio WWW es la excesiva abundancia de información que contiene, que hace difícil encontrar un tema específico. Aunque hay servidores especializados en facilitar estas búsquedas, no es raro que la selección que nos presente sobre el tema que le hemos pedido sea también desbordante, recogiendo cientos de artículos, de los cuales sólo unos pocos tendrán cierto interés. Es por ello por lo que últimamente la mejor forma de obtener información sea acudiendo a otras personas que ya han realizado la búsqueda sobre el tema anteriormente, y han seleccionado la mejor. Nosotros hemos seleccionado una serie de páginas de gran interés para los Administradores de Sistemas, que recogen un tipo de información imposible de encontrar de otro modo.

Queremos señalar que una parte de información recogida es para usuarios muy avanzados, casi expertos en informática, y para este tipo

de personas uno de sus mayores retos (por lo menos, el más emocionante), es romper la seguridad de los sistemas informáticos de empresas y administraciones. Por eso algunas de las páginas seleccionadas tratan sobre cómo acceder a sitios no autorizados, entorpecer el trabajo de otras personas, o incluso destruir o cambiar ficheros. Obviamente, este tipo de información no aparece en libros ni artículos, y tampoco es nuestra intención explicar aquí cómo (ni dónde) conseguir estos resultados tan poco lícitos. Sin embargo, creemos que para los futuros administradores de sistemas es muy importante saber que este tipo de información está accesible para todo aquel que sepa buscarla, y conocer cuáles son los efectos, a veces demoledores, que pueden sufrir en sus equipos para, si es posible, remediarlos.

Expondremos ahora la información recopilada en las distintas páginas, que hemos dividido en cuatro grandes grupos: privacidad, protocolo TCP/IP, bugs de sistemas y virus. Al final de cada grupo se presenta la lista de páginas WWW consultadas para la realización del mismo.

Privacidad

La "privacidad" como tal no existe en Internet a priori. Al enviar correo, al postear en las News, al realizar una búsqueda y simplemente navegando vamos dejando un claro rastro de datos sobre nosotros: nuestros gustos, nuestros horarios de conexión, nuestras tendencias ideológicas, políticas, sexuales, religiosas, navegador que empleamos, dirección IP, correo y un largo etcétera que son acumulados por quienes estimen oportuno. ¿Ha pensado alguna vez en la cantidad de información que pueden disponer sobre usted?

Hay muchas empresas que pagarían mucho por disponer de este tipo de información. Hoy día no está de moda traficar con armas ni drogas sino con información, por algo vivimos en la llamada Era de la Información. Quizás usted no esté dispuesto a ser parte de esas bases de datos, o simplemente pretenda pasar desapercibido. Pues a continuación vamos a introducirle en el interesante tema de la privacidad o "anonimato" en La Red. Es solamente una introducción en dicho tema,

no pretendemos entrar con demasiada profundidad ya que el tema es extenso.

Navegación Anónima

¿Cuántas veces "navegando" por Internet ha visto esos inocentes contadores de acceso en una página Web? Pues no sólo pueden saber que es la visita nº xxxx, sino que además pueden saber de dónde viene, de qué página web accede,... ¿Cómo cree sino que controlan la rentabilidad de los banners publicitarios que encuentra en la mayoría de las paginas?

¿Cuántas veces al apuntarse a un foro de debate, solicitar actualizaciones de algún software, drivers, etc. ha rellenado un formulario previo con una serie de datos como: edad, nombre, apellido, dirección, población, teléfono, e-mail, titulación, ocupación, equipo informático, sistemas operativos y un largo etcétera? Ellos siempre te garantizan "privacidad" pero no dicen nada sobre la venta de esos datos que tienen sobre ti a otras empresas o entidades...

¿Cómo puede evitar esto? Muy sencillo. Navegue de forma anónima. Para ello puede acceder a servidores a través de su puerto 80 y navegar "a través" de ellos. Hay algunos servidores que lo permiten, como el caso de Anonymizer (www.anonymizer.com), C2 (www.c2.org) y otros que lo hacen también, pero sin saberlo, debido a descuidos de los administradores. Para ello no hay mas que indicar la dirección del servidor a través del cual vamos a acceder a las páginas que nos interesan, seguido del puerto 8080 (la de World Wide Web TCP/ HyperText Transfer Protocol/UDP) y, a continuación, la dirección que nos interesa.

Por ejemplo:

Queremos acceder a la página web de Microsoft de forma anónima y lo vamos a hacer a través del servidor de Anonymizer, para ello, pondremos en nuestro navegador:

http://www.anonymizer.com:8080/http://www.microsoft.com

Evidentemente este sistema es mas lento que el normal ya que la página solicitada es enviada a anonymizer y posteriormente éste nos la

envía a nosotros, pero hemos conseguido lo que queríamos: ¡el anonimato!

Bueno, casi, queda otro detalle, que es el de los cookies. Los cookies son una serie de datos solicitados por el web que visita sobre... ¿adivina quién?, sí, sobre usted. Cuando recibe un cookie, almacena nuestros datos como dirección IP, fecha, hora, número identificación, sistema operativo, navegador y un sin fin de datos más. Cuando solicite una página, el navegador enviará esta información de vuelta al servidor que lo originó y sólo a éste (menos mal). Cada vez que nos conectemos a ese servidor que generó el cookie, éste le enviará de nuevo la nueva fecha, hora, IP, etc.

Por ejemplo:

Supongamos que realiza una búsqueda en un Buscador de Internet y éste buscador le envía un cookie con un número identificación. El servidor del buscador guardará automáticamente su número y el tema sobre la búsqueda que ha realizado, y usted guardará el cookie, con lo cual otro día cuando vuelva a conectarse a su buscador en busca de información sobre otro tema, su navegador volverá a enviar el cookie con su número de identificación y de nuevo almacenarán el tema sobre el que ha realizado la búsqueda. Al cabo del tiempo tendrán una base de datos estupenda sobre los temas que le interesan, a las horas que ha realizado las búsquedas, fechas...

Todos los navegadores actualmente tienen la opción de pedir confirmación por parte del usuario para aceptar un cookie o simplemente que no los acepte, o sea, que en sus manos está el enviar esta información o no.

Correo Anónimo

Enviar correo anónimo, como su propio nombre indica, consiste en enviar un e-mail de forma que se desconoce el remitente. Hay varias formas de enviar un correo de forma anónima: con algunos programas que afirman enviar correo anónimo sin uso de remailers (de los que no vamos a hablar debido a su ineficacia), y a través de remailers, que es la forma más eficaz.

¿Qué es un remailer? Un remailer es un servicio informatizado que privatiza su correo electrónico. Más adelante explicaremos su funcionamiento. Un remailer le permite enviar correo a un grupo Usenet o a cualquier usuario de forma que no aparezca ni su dirección de correo ni su nombre. La mayoría de remailers ofrecen sus servicios de forma gratuita. Hay varios tipos:

- Los basados en páginas web como el remailer de Replay (www.replay.com/remailer/anon_no_ssl.htm).
- Los remailers "simples" (envía un correo y ellos lo reenvían una vez) (www.anonymizer.com).
- Los remailers Cypherpunks (emplean una cadena de remailers, se reenvían a través de varios remailers; son los más eficaces, junto con el MixMaster).
- Los pseudo-anónimos. (en realidad no son remailers, son servidores nym; le asignan un Id de usuario;)(Lutz Donnerhacke's PSEUDO Anonymous Remailer, www.iks-jena.de/mitarb/lutz/anon)
- Los que requieren de un software específico para su uso, como MixMaster.

Hay dos tipos fundamentales de remailers, los anónimos y los pseudo-anónimos o servidores nym, los primeros son los del tipo MixMaster y Replay mencionado anteriormente y de los que hablaremos más adelante. Los segundos funcionan básicamente de la siguiente manera: usted como usuario ha de contratar una cuenta con un servidor nym. Esto quiere decir que yo, como operador conozco su verdadera dirección de correo y nombre. Su privacidad será tan segura como lo son mis métodos para proteger mis datos sobre mis usuarios. En principio no debe de haber ningún problema salvo si mi sistema es hackeado o bien recibo una orden judicial que me obligue a revelar los datos verdaderos de mi cliente.

Yo, como operador, le asigno una identificación de cliente, con lo cual, cuando recibo un correo suyo cambio las cabeceras (donde viene su dirección de correo) y lo cambio por un número de identificación y posteriormente lo envío(hay algunos remailers que emplean un método de encriptación como el PGP del que hablaremos más adelante). Si

su mensaje es contestado, se enviará al servidor nym, que se encargará de eliminar las cabeceras de nuevo del remitente y cambiarlo por su dirección de correo. Con lo cual respeto su privacidad y la del remitente.

Los remailers del tipo Cypherpunk y MixMaster de Lance Cottrelly son los verdaderamente anónimos. Emplean más de un remailer con lo que solamente el primer remailer conoce su dirección, pero no el destino final. Si se emplean correctamente, puede estar seguro de que NADIE sabrá su dirección ni nombre (ni el operador ni ningún snoop). Éste es el verdadero sentido del correo anónimo. En la práctica nadie puede conseguir sus verdaderos datos haciendo que el operador del remailer anónimo los revele, simplemente porque el propio operador los desconoce.

a) ¿Porqué querría usar un remailer?

Quizás sea un empleado de una conocida marca de componentes informáticos y quiera dar su opinión sobre ellos, una opinión no muy favorable, que tal vez su jefe no tome a bien. Quizás viva en una comunidad que no es muy tolerante con ciertas posturas sociales, políticas o religiosas. Quizás busque un empleo y no quiera poner en peligro el actual. Quizás quiera poner anuncios personales. Tal vez piense que si critica al gobierno ellos tomen nota. Quizás no quiera que bombardeen su e-mail por haber posteado algo. En resumen, hay muchas razones legítimas y legales por la que considerar hacer uso de un remailer.

b) ¿Porqué son gratis los remailers?

Por la sencilla razón de si piden un número de cuenta o de tarjeta de crédito, deja de ser anónimo.

c) ¿Entonces, porqué existe este servicio?

Hay gente que deciden montar un remailer para su propio uso y lo ponen al servicio de los demás. Lógicamente si es muy visitado seguro que hay empresas dispuestas a colocar publicidad en sus Webs con lo que consiguen algunos ingresos.

Joshua Quittner, co-autor del thriller Mother´s Day, entrevistó al Sr. Julf Helsingius para la revista Wired. Helsingius, quien tuvo funcio-

nando el remailer más popular del mundo durante tres años hasta que se retiró el 30 de Agosto de 1996, declaró:

"Es importante poder expresar ciertos puntos de vista sin que todo el mundo sepa quien eres. Uno de los mejores ejemplos es el del identificador de llamadas de los teléfonos(al hacer una llamada a un móvil o RDSI aparece el numero de la llamada entrante). A las personas que realizaban la llamada les molestaba que a quien llamaran supieran de antemano quienes eran. Con aparatos como los teléfonos la gente daba por hecho que si querían ser anónimos podían serlo y les molesta el hecho de que se les prive de esta posibilidad. Pienso que lo mismo ocurre con el correo electrónico."

> "*Viviendo en Noruega, pude ver de cerca como eran las cosas en la antigua Unión Soviética. Si poseías una fotocopiadora o una máquina de escribir tenías que tenerla registrada y para ello tomaban muestras de impresión para posteriormente poder identificarla si fuese necesario. El hecho de que tuvieras que registrar cualquier forma de enviar información, de una forma paralela, es como decir que tienes que firmar cualquier cosa que envíes por La Red para que podamos seguirte la pista.*"

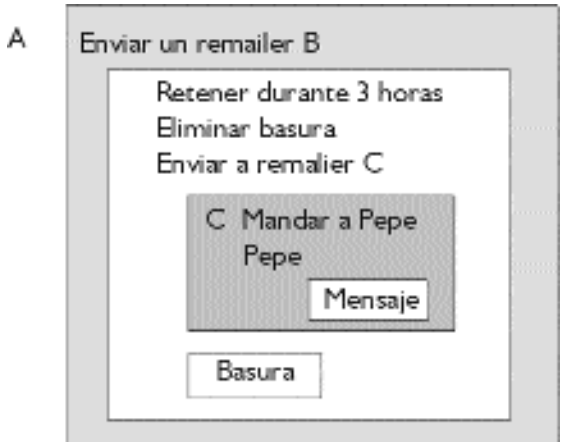
d) ¿Cuál es el remailer ideal?

Es aquél que cumple los siguientes requisitos:

1) Es fácil de utilizar
2) Está administrado por una persona lo suficientemente competente como para cumplir lo que promete, y además sea capaz de mantener la integridad de sus datos impidiendo el acceso a ellos por parte de hackers o entidades gubernamentales.
3) Que sea capaz de mantener su mensaje un tiempo aleatorio en el servidor antes de su envío, para así hacer más difícil por parte de snoopers linkar un mensaje que llega, digamos, a las 5:00 PM y que parte de tu máquina a las 4:59 PM.
4) Permite métodos de encriptación fuertes como el PGP (Pretty Good Privcacy).

Cypherpunk: Son remailers de primera generación. La estructura actual de los mensajes emitidos por remailers es un conjunto anidado de mensajes encriptados. Cada mensaje encriptado tiene como destino un remailer distinto. El mensaje contiene las instrucciones para cada remailer, como donde enviar el mensaje (próximo remailer) y el contenido del mensaje. Cada remailer elimina una capa de la encriptación y de las instrucciones que la acompañan a ese nivel, realiza las instrucciones dadas y envía el mensaje al próximo destino.

Funcionamiento de los Remailers



La figura anterior representa un mensaje enviado a través de tres remailers (A, B y C) y finalmente a Pepe. Las cajas representan la encriptación con el nombre de la persona a la que va dirigida en la esquina superior izquierda. Un factor importante y que salta a la vista en el diagrama es que tras cada salto nuestro mensaje inicial mengua en tamaño.

El mensaje inicial es enviado a Pepe a través del remailer A, el cual añade al mensaje que sea enviado al remailer B, éste a su vez lo retiene durante tres horas, elimina la basura (cabeceras dejadas por remailer A) y lo envía al remailer C que de nuevo elimina la basura y lo envía de forma correcta (anónima) al destinatario Pepe.

Los remailers Cypherpunk hacen diversas cosas:

- Envían un mensaje a otra dirección de correo o lo postean a un grupo de noticias.
- Aceptan mensajes encriptados con instrucciones para procesar posibles mensajes incluidos dentro de la encriptación.
- Eliminan de los cabeceros todas, o al menos parte, de las direcciones por donde ha pasado el mensaje.
- Añaden nuevos cabeceros.
- Eliminan parte de información del final del mensaje
- Encriptan parte del mensaje utilizando una llave especificada dentro del mensaje.
- Retienen durante un tiempo prefijado o aleatorio dicho mensaje.
- Reordenan mensajes de manera que siempre se mantenga un numero determinado en el remailer, para así evitar que los snoops puedan rastrearlos.

Ataques a remailers Cypherpunk

Ya que los remailers anónimos están diseñados para prevenir el análisis de tráfico, la mejor forma de entender sus debilidades es estudiar diversos ataques que un adversario puede emplear contra ellos. Supongamos un adversario muy potente, para así ponernos en una situación extrema y para demostrar que es posible resistir estos ataques con remailers de segunda generación.

Supongamos que el atacante es capaz de registrar los contenidos de los mensajes que entran y salen del remailer junto con hora de llegada y salida. Todos los mensajes son monitorizados tanto cuando salen de la máquina emisora tanto como cuando llegan a su destino. El atacante es capaz de enviar un número indeterminado de mensajes a través del remailer tanto si son nuevos como si son interceptados previamente. Puede impedir que esos mensajes lleguen a su destino. Conoce algunos de los remailers, digamos secundarios, origen, destino y contenidos de todos los mensajes que pasen por dichos remailers. Supongamos que lo anteriormente expuesto es nuestra situación inicial.

De lo anteriormente expuesto es evidente que los mensajes no encriptados pueden ser traceados, así que consideraremos solamente los que están encriptados. El uso de un solo remailer es también inseguro, si ese remailer es conocido por el atacante también conoce la dirección origen y destino.

Cadenas de remailers con encriptación son muchos mejores, pero aún así vulnerables, ya que los mensajes entrantes son directamente enviados tras ser procesados; cuando llega un mensaje, sale otro, con lo que el atacante sabe que es el mismo mensaje por muchas precauciones que se hayan tomado.

Este es el mayor problema que tienen los remailers Cypherpunk. El primer fix que se propuso fue retener durante un tiempo aleatorio el mensaje. Si este tiempo es mayor que el de respuesta de un mensaje, es imposible saber con certeza qué respuesta le corresponde a cada mensaje. Esto, de todas maneras, es un tanto débil, ya que mientras haya mucho tráfico, estupendo, pero en el momento en que este tráfico desciende resulta relativamente sencillo distinguirlos. Para evitar esto el tiempo de retardo ha de ser mucho mayor que el necesario para momentos de máximo tráfico.

Aunque el hecho de reordenar los mensajes ayuda a solventar este problema, también abre otra posibilidad al atacante, ya que implica siempre dejar un número de mensajes en el remailer. Esos mensajes reciben el nombre de "pool". La forma más eficiente de reordenarlos consiste en mantener en el pool N mensajes y enviar uno de los (N+1) mensajes del pool (incluyendo el que acaba de llegar) elegido al azar. Desafortunadamente, esto es susceptible a un spam. Si envía mucho mas que N mensajes, el pool solamente contendrá mensajes del atacante. Estos mensajes reemplazan los existentes en el pool dejando solamente los mensajes que puede identificar el atacante. Si el atacante envía otro conjunto de mensajes tras recibir un mensaje de un usuario, dicho mensaje será eliminado del pool. Como el atacante puede identificar sus mensajes, conocer el del usuario resulta obvio.

Combinando el retardo aleatorio y la reordenación se opone algo de resistencia al atacante. Si en vez de enviar un mensaje del pool cada vez que llega uno nuevo, enviamos todos los mensajes menos N, mucho mejor. Si durante un tiempo determinado varios mensajes ver-

daderos han llegado, entonces, incluso aunque se haya vaciado el pool, habrá más mensajes mezclados con los del atacante. Si el atacante emplea un spam junto con la negación de ofrecer servicio, entonces de nuevo su mensaje será el único que no es del atacante. No hay nada que hacer si el atacante puede asegurarse de que su mensaje es el único que pasa por la lista de remailers. Con remailers ideales, su mensaje podría ser cualquiera de los circulan por ellos en ese momento, pero como no está pasando ninguno salvo el suyo, se sabe cuál es.

Ahora suponga que su mensaje pasa por remailers que están retardando y reordenando en cada salto de remailer. Todavía se puede rastrear su mensaje por el tamaño del mismo. Por defecto en cada salto el mensaje mengua en un tamaño pequeño y más o menos conocido, por lo que aunque esté junto a otros muchos mensajes y de distinto tamaño, se puede rastrear. Como máximo puede eliminar hasta el tamaño mínimo del mensaje. También está limitado porque mensajes grandes llamarán excesivamente la atención al eliminar una fracción de ellos mismos en cada salto. Llegamos a la siguiente conclusión, todos los mensajes han de tener el mismo tamaño para así evitar distinguir unos de otros.

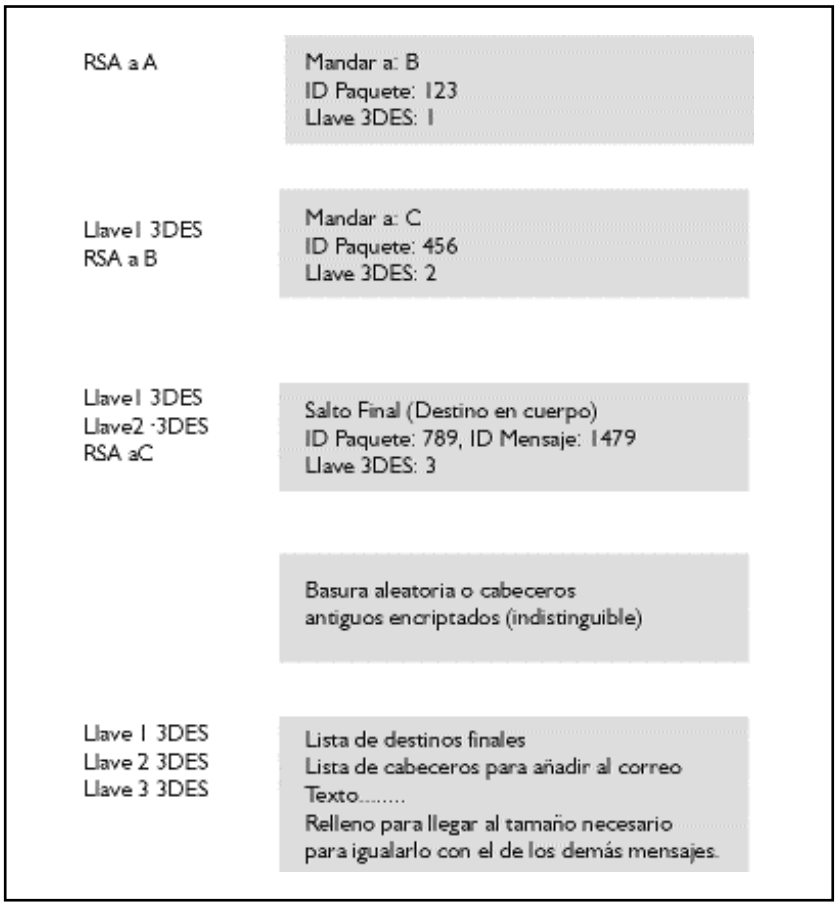
Aún así, los mensajes todavía pueden ser rastreados con un ataque Replay. Este método se emplea para seguir un mensaje a su destino y recorrer el camino a la inversa. Este tipo de ataque es un derivado del ataque spam. Para seguir la pista de un mensaje a través de una serie de remailers, el atacante captura el mensaje y lo copia varias veces enviándolas al primer remailer. Entonces aparecerán muchos mensajes que indicarán la ruta seguida por el mensaje, si resulta demasiado disperso debido el reordenamiento se puede hacer a través de varios remailers introduciendo varias copias del mensaje en los mismos.

Para prevenir este tipo de ataques, los remailers no deben de permitir enviar ningún mensaje más de una vez. Esto se puede hacer introduciendo un número aleatorio en cada mensaje a cada salto que es grabado por el remailer. Como inconveniente requiere de cierto espacio para guardar dicho log.

Para utilizar estos tipos de remailers directamente es necesario un software. El más completo que hay ahora mismo en el mercado es el Private Idaho, permite PGP, correo, News, etc. y está disponible con código fuente en http://www.eskimo.com/~joelm/pi.html .

MixMaster

El único remailer de segunda generación es MixMaster. La filosofía de su diseño está fuertemente influenciado por "los papeles de Chaum en mezclas digitales". A continuación mostramos la estructura de un paquete MixMaster. En lugar de cuatro cabeceros, hay veinte. Los mensajes se envían como uno o más paquetes (mensajes compuestos por múltiples paquetes se denominan mensajes multi-parte). En el siguiente diagrama las capas de encriptación están indicados por una anotación a la izquierda del objeto. La última llave (key) en la lista será la primera empleada. Están en el orden necesario para desencriptar la información.



Cuando el remailer A recibe este mensaje, desencriptará el paquete RSA de arriba para comprobar que el ID del mensaje no ha estado ahí anteriormente. Si ha pasado anteriormente será eliminado de inmediato. El primer cabecero (anterior a la desencriptación RSA) será añadido al final de la lista de cabeceros y todos los demás se despalazarán hacia arriba. Todos los cabeceros y el cuerpo (la parte que contiene el texto) se desencriptarán con la llaves 3DES del cabecero. Esto muestra arriba otro cabecero, encriptado con el método RSA, para el próximo remailer y oculta el antiguo de nuevo al final de la lista. MixMaster emplea triple llave DES para toda la encriptación y 1024 bit RSA para encriptación de la llave pública de las llaves 3DES.

El cabecero para el último remailer de la cadena contiene un flag que indica que es el último salto y que es parte de un mensaje multiparte. Si no es un mensaje multiparte el cuerpo se desencripta y el texto se coloca en el pool para ser reordenado y enviado de nuevo. Si es parte de un mensaje entonces el ID Mensaje es usado para identificar las otras partes tal como van llegando. Cuando todas las partes han llegado se une el mensaje y se coloca en el pool. Si no se reciben todas las partes en un tiempo determinado se descarta el mensaje. Solamente el último remailer puede distinguir si los paquetes recibidos son parte de un mensaje o no. Para los demás remailers son paquetes independientes.

Todos los paquetes tienen exactamente la misma longitud y todos los bits están encriptados con una llave 3DES en cada salto por lo que ninguna información del mensaje resulta visible al observador. Incluso un remailer "hackeado" solamente puede saber el remailer siguiente y anterior de la cadena. No puede saber cuántos saltos ha hecho ni cuántos le queda, salvo que sea el último.

Estos mensaje resultan un poco grandes, cada uno de los cabeceros ocupan 512 bytes y el cuerpo 10k. Pero a cambio ofrecen SEGURIDAD.

Para obtener una lista actualizada de remailers diponibles consulta la página web http://www.cs.berkeley.edu/~raph/remailer-list.html .

Encriptación de correo: PGP (Pretty Good Privacy o "intimidad bastante buena")

Pretty Good Privacy (PGP), de Phil's Pretty Good Software, es una aplicación informática de criptografía de alta seguridad para MSDOS, Unix, VAX/VMS y otros ordenadores. PGP permite intercambiar ficheros y mensajes con intimidad, autentificación y comodidad. 'Intimidad' quiere decir que sólo podrán leer el mensaje aquellos a quienes va dirigido. 'Autentificación' quiere decir que los mensajes que parecen ser de alguien sólo pueden venir de esa persona en particular. 'Comodidad' quiere decir que la intimidad y la autentificación se consiguen sin los problemas de gestión de claves asociados a otros programas de criptografía convencional. No se necesitan canales seguros para intercambiar claves entre usuarios, por lo que PGP resulta mucho más fácil de utilizar. Esto se debe a que PGP está basado en una potente nueva tecnología llamada criptografía de "clave pública". PGP combina la comodidad del criptosistema de clave pública de Rivest-Shamir-Adleman (RSA) con la velocidad de la criptografía convencional, resúmenes de mensajes para firmas digitales, compresión de datos antes de encriptar, un buen diseño ergonómico y una completa gestión de claves. Por otra parte, PGP realiza las funciones de clave pública con más rápidez que la mayoría de las demás implementaciones informáticas. PGP es criptografía de clave pública para todos.

¿Por qué se necesita PGP?

Es personal. Es privado. Y no le concierne a nadie más que a usted. Puede estar planificando una campaña política, tratando sobre sus impuestos o teniendo una aventura. O puede estar haciendo algo que en su opinión no debería ser ilegal, pero que lo es. Sea lo que fuere, no quiere que nadie más lea su correo electrónico (correo-E) privado ni sus documentos confidenciales. No hay nada malo en afirmar su derecho a la intimidad. El derecho a la intimidad es tan básico como la Constitución.

¿Qué pasaría si todo el mundo creyese que los ciudadanos respetuosos de la ley deberían utilizar postales para enviar el correo? Si algún espíritu valiente intentase afirmar su intimidad utilizando un sobre, levantaría sospechas. Las autoridades quizá abrirían su correo

para ver qué está ocultando. Afortunadamente no vivimos en un mundo así y la mayor parte del correo se protege con sobres. Nadie levanta sospechas por afirmar su intimidad con un sobre. Hay seguridad en los grandes números. De forma análoga, sería interesante que todo el mundo utilizase habitualmente el cifrado en el correo-E, fuese inocente o no, para que nadie levantase sospechas por afirmar de esa manera su derecho a la intimidad. Piense en ello como una forma de solidaridad.

Hoy en día, si el gobierno quiere invadir la intimidad de los ciudadanos corrientes tiene que emplear una cierta cantidad de esfuerzo y dinero en interceptar y abrir al vapor el correo normal, o en escuchar, y quizá transcribir, las conversaciones telefónicas. Este tipo de control, muy laborioso, no resulta práctico a gran escala. Sólo se realiza en casos importantes, donde parece que va a merecer la pena.

Cada vez una mayor parte de nuestra comunicación privada se dirige por canales electrónicos. El correo electrónico reemplaza gradualmente al correo convencional. Los mensajes por correo-E son demasiado fáciles de interceptar y de explorar para buscar palabras interesantes. Es algo que puede hacerse a gran escala, fácilmente, habitualmente, automáticamente y de una forma imposible de detectar.

Nos dirigimos hacia un futuro en el que los países estarán cruzados de lado a lado por redes de datos basadas en fibra óptica de alta capacidad, que conectarán todos nuestros ordenadores personales cada vez más ubicuos. El correo-E será la norma para todos, no la novedad que resulta hoy en día. Puede que el gobierno proteja nuestro correo-E con algoritmos de cifrado diseñados por ellos mismos. Y puede que la mayoría esté de acuerdo. Pero algunos preferirán tomar sus propias medidas de protección.

La propuesta de ley 266 del Senado de los Estados Unidos, una propuesta conjunta contra el delito, tenía oculta una medida inquietante. Si esta resolución no vinculante hubiese llegado a ley, habría obligado a los fabricantes de equipos de comunicaciones seguras a incluir "puertas traseras" en sus productos para que el Gobierno pudiese leer cualquier mensaje cifrado. Su traducción al castellano es la siguiente: "Es la opinión del Congreso que los proveedores de servicios de comunicación electrónica y los fabricantes de equipos para servicios

de comunicación electrónica deben garantizar que los sistemas de comunicación permitan al Gobierno obtener el contenido en texto normal de las comunicaciones de voz, datos y otras comunicaciones, cuando esté adecuadamente autorizado por la ley". Esta medida fue desestimada tras una rigurosa protesta por parte de defensores de la libertad civil y de grupos empresariales.

En 1992, la propuesta del FBI sobre intervención de telefonía digital se presentó en el Congreso norteamericano. Obligaría a todos los fabricantes de equipos de comunicaciones a integrar unos puertos especiales para la intervención a distancia, que permitiría al FBI intervenir todo tipo de comunicación electrónica desde sus oficinas. Aunque no consiguió ningún apoyo en el Congreso gracias a la oposición ciudadana, volvió a presentarse en 1994.

Lo más alarmante es la nueva y enérgica iniciativa de la Casa Blanca sobre política criptográfica, desarrollada en la NSA desde el inicio de la administración Bush y presentada el 16 de Abril de 1993. La parte central de esta iniciativa es un dispositivo criptográfico construido por el Gobierno, llamado el "chip Clipper", que contiene un nuevo algoritmo criptográfico secreto de la NSA. El Gobierno anima a las empresas privadas a que lo incluyan en todos sus productos de comunicaciones seguras, como teléfonos, fax, etc. AT&T está instalando Clipper en todos sus productos seguros para voz. La trampa: en la fábrica, cada chip Clipper se carga con su propia clave única y el Gobierno mantiene una copia en depósito. Pero no hay que preocuparse, el Gobierno promete que sólo utilizará esas claves para leer las comunicaciones cuando esté autorizado por la ley. Naturalmente, para que Clipper sea efectivo, el siguiente paso lógico sería proscribir otras formas de criptografía.

Si la intimidad se proscribe, sólo los proscritos tendrán intimidad. Los servicios de inteligencia tienen acceso a tecnología criptográfica de calidad. Lo mismo ocurre con los grandes traficantes de armas y los narcotraficantes. También disponen de ellos los contratistas del ejército, las compañías de petróleo y otros gigantes empresariales. Pero la mayoría de la gente normal y de las organizaciones políticas de base no han tenido nunca a su alcance una tecnología asequible para utilizar criptología de clave pública de "grado militar". Hasta ahora.

¿Cómo funciona?

Supongamos que quiero enviarle un mensaje que nadie excepto usted pueda leer. Podría "encriptar" o "cifrar" el mensaje, lo que significa revolverlo de una forma tremendamente complicada, con el fin de que resulte ilegible para cualquiera que no sea usted, el destinatario original del mensaje. Yo elijo una "clave" criptográfica para encriptar el mensaje y usted tiene que utilizar la misma clave para descifrarlo o "desencriptarlo". Por lo menos así funciona en los criptosistemas convencionales de "clave única".

En los criptosistemas convencionales, como el US Federal Data Encription Standard (DES) [Norma federal para cifrado de datos en EE.UU.], se utiliza una sola clave para encriptar y desencriptar. Por lo tanto, hay que transmitir primero la clave por medio de un canal seguro para que ambas partes la conozcan antes de enviar mensajes cifrados por canales inseguros. Este proceso puede resultar incómodo. Si se tiene un canal seguro para intercambiar claves, ¿para qué se necesita entonces criptografía?

En criptosistemas de clave pública, todo el mundo tiene dos claves complementarias, una revelada públicamente y otra secreta (llamada también clave privada). Cada clave abre el código que produce la otra. Saber la clave pública no sirve para deducir la clave secreta correspondiente. La clave pública puede publicarse y distribuirse ampliamente por una red de comunicaciones. Este protocolo proporciona intimidad sin necesidad de ese canal seguro que requieren los criptosistemas convencionales.

Cualquiera puede utilizar la clave pública de un destinatario para encriptar un mensaje y éste empleará su clave secreta correspondiente para desencriptarlo. Sólo él podrá hacerlo, porque nadie más tiene acceso a esa clave secreta. Ni siquiera la persona que lo encriptó podría descifrarlo.

También proporciona autentificación para mensajes. La clave secreta del remitente puede emplearse para encriptar un mensaje, "firmándolo". Se genera una firma digital, que el destinatario (o cualquier otra persona) puede comprobar al descifrarla con la clave pública del remitente. De esta forma se prueba el verdadero origen del mensaje y que no ha sido alterado por nadie, ya que sólo el remitente posee la clave secreta que ha producido esa firma. No es posible falsificar un mensaje firmado y el remitente no podrá desautorizar su firma más adelante.

Estos dos procesos pueden combinarse para obtener intimidad y autentificación al mismo tiempo, si se firma primero el mensaje con la clave secreta y se encripta después el mensaje firmado con la clave pública del destinatario. El destinatario sigue estos pasos en sentido contrario al desencriptar primero el mensaje con su propia clave secreta y comprobar después la firma con la clave pública del remitente. El programa lo hace automáticamente.

Como el algoritmo de cifrado en clave pública es mucho más lento que el cifrado convencional de clave única, el proceso resulta más eficaz con un algoritmo convencional rápido de alta calidad, de clave única, para encriptar el mensaje. El mensaje original sin encriptar se denomina "texto en claro". Sin intervención del usuario se utiliza una clave aleatoria temporal, generada sólo para esa "sesión", para encriptar convencionalmente el fichero normal. Después se encripta esa clave aleatoria convencional con la clave pública del destinatario. La clave de la "sesión" convencional, encriptada con esa clave pública, se envía al destinatario junto al texto cifrado. El destinatario recupera esa clave temporal con su propia clave secreta y ejecuta con ella el algoritmo convencional de clave única, más rápido, para desencriptar el mensaje cifrado.

Las claves públicas se guardan en "certificados de clave" individuales que incluyen el identificador de usuario del propietario (el nombre de esa persona [y algún dato único, como la dirección de correo-E]), un sello de hora del momento en el que se generó el par y el material propio de la clave. Cada clave secreta está encriptada con su propia contraseña, por si alguien roba la clave. Cada fichero ("anillo") de claves contiene uno o más de esos certificados.

Las claves se identifican internamente mediante un "identificador de clave", que es una "abreviatura" de la clave pública (sus 64 bits menos significativos). Cuando se muestra este identificador, sólo aparecen los 32 bits inferiores para mayor brevedad. Aunque muchas claves pueden compartir el mismo identificador de usuario, a efectos prácticos no hay dos claves que compartan el mismo identificador de clave.

PGP utiliza "resúmenes de mensaje" para elaborar las firmas. Un resumen de mensaje es una función "distribución" ("hash") unidireccional de 128 bits, criptográficamente resistente, de ese mensaje. Es aná-

logo a una "suma de verificación" o código CRC de comprobación de errores: "representa" el mensaje de forma compacta y se utiliza para detectar cambios en él. A diferencia de un CRC, sin embargo, resulta computacionalmente impracticable para un atacante idear un mensaje sustitutivo que produzca un resumen idéntico. El resumen del mensaje se encripta con la clave secreta para elaborar la firma. Los documentos se firman añadiéndoles como prefijo un certificado de firma, junto con el identificador de la clave que se utilizó para realizarla, un resumen de mensaje del documento (firmado con la clave secreta) y un sello de hora del momento de la firma. El destinatario utiliza el identificador de la clave para buscar la clave pública del remitente y comprobar la firma. El programa busca automáticamente la clave pública y el identificador de usuario en el fichero de claves correspondiente.

Los ficheros cifrados llevan como prefijo el identificador de la clave pública con la que se han encriptado. El destinatario utiliza este prefijo de identificación para encontrar la clave secreta y poder desencriptar el mensaje. Su programa busca automáticamente la clave secreta en el fichero de claves correspondiente. Estos dos tipos de fichero constituyen el método principal para almacenar y gestionar las claves públicas y secretas. En lugar de mantener las claves individuales en ficheros separados, se reúnen en anillos para facilitar la búsqueda automática, ya sea por identificador de clave o por identificador de usuario. Cada usuario mantiene su propio par de anillos. Las claves públicas individuales se guardan en ficheros aparte durante el tiempo necesario para enviarlas a algún amigo, que las añadirá entonces a su propio anillo de claves.

¿Cómo proteger las claves públicas contra manipulación?

En un sistema de clave pública no hay que proteger las claves públicas contra exposición. De hecho, es mejor que estén ampliamente difundidas. Sin embargo, es importante protegerlas contra manipulación para asegurar que una clave pertenece realmente a quien parece pertenecer. Este quizá sea el punto más vulnerable de un criptosiste-

ma de clave pública. Veamos primero un posible desastre y a continuación la manera de evitarlo con PGP.

Supongamos que quiere enviar a Alicia un mensaje privado. Recibe la clave pública de Alicia desde una BBS (Bulletin Board System: tablón electrónico de anuncios). Encripta la carta para Alicia con esa clave y la envía por medio del correo-E de la BBS.

Desafortunadamente, sin saberlo Alicia ni usted, otro usuario llamado Carlos se ha infiltrado en la BBS y ha generado una clave pública propia que lleva el identificador de usuario de Alicia. Pone secretamente esa clave falsa en lugar de la verdadera. Usted, sin saberlo, utiliza esa clave en lugar de la auténtica. Todo parece normal porque la clave falsa tiene el identificador de usuario de Alicia. Ahora Carlos puede descifrar el mensaje dirigido a Alicia, ya que tiene la clave secreta correspondiente. Puede incluso volver a encriptar el mensaje con la verdadera clave pública de Alicia y enviárselo a ella para que nadie sospeche nada. Aún peor, puede incluso hacer firmas en nombre de Alicia con esa clave secreta, porque todo el mundo utiliza la clave pública falsa para comprobar las firmas de Alicia.

La única forma de evitar este desastre es impedir que alguien pueda manipular las claves públicas. Si ha obtenido la clave pública directamente de Alicia, no hay problema. Sin embargo, esto puede resultar difícil si la persona se encuentra a mil kilómetros, o no es localizable en ese momento.

Podría conseguir la clave pública de Alicia de un amigo en el que confien los dos, David, que sabe que su copia de la clave pública de Alicia es buena. David podría firmar la clave pública de Alicia, respondiendo de la integridad de la clave. David realizaría esta firma con su propia clave secreta.

Así se crearía un certificado firmado de clave pública que demostraría que la clave de Alicia no ha sido manipulada. Este mecanismo requiere que su copia de la clave pública de David sea buena, para poder comprobar la firma. David podría también proporcionar a Alicia una copia firmada de su clave pública. Por tanto, David hace de referencia entre Alicia y usted.

David o Alicia podrían enviar a la BBS ese certificado firmado de clave pública de parte de Alicia y usted podría recibirlo más adelante.

Entonces podría comprobar la firma con la clave pública de David y asegurarse de que es la verdadera clave de Alicia. Ningún impostor podría hacer que aceptase una clave falsa como si fuera de Alicia, porque nadie puede falsificar la firma de David.

Una persona de amplia confianza podría incluso especializarse en ofrecer este servicio de "referencia" entre usuarios, proporcionando firmas para esos certificados de clave pública. Esta persona de confianza podría considerarse un "servidor de claves" o "autoridad de certificación". Podría confiarse en que cualquier certificado de clave pública con la firma del servidor pertenecería verdaderamente a quien parecía pertenecer. Los usuarios que quisieran participar sólo necesitarían una copia buena de la clave pública del organizador para poder verificar sus firmas.

Un servidor centralizado de claves o autoridad de certificación está especialmente indicado en grandes instituciones gubernamentales o empresariales con control centralizado. Algunos entornos institucionales ya utilizan jerarquías de autoridades de certificación.

Para entornos de base descentralizados, estilo "guerrilla", permitir a cualquier usuario actuar como referencia de confianza de sus amigos probablemente funcionará mejor que un servidor centralizado. PGP tiende a enfatizar este enfoque orgánico descentralizado no institucional. Refleja mejor la forma natural que tienen los humanos de interactuar personalmente a nivel social y permite elegir mejor en quién confiar para la gestión de claves.

Este tema de proteger las claves públicas contra manipulación es el problema individual más difícil con que se encuentra la aplicación práctica de la clave pública. Es el "talón de Aquiles" de la criptografía de clave pública; solamente en resolver este problema hay invertida una gran complejidad de programación.

Sólo debería utilizar una clave pública después de comprobar que es una clave auténtica no manipulada y que pertenece a la persona a la que dice pertenecer. Puede estar seguro de ello si obtiene el certificado de clave pública directamente de su propietario, o si lleva la firma de alguien en quien confía y del que ya tiene una clave pública auténtica. Por otra parte, el identificador de usuario debería llevar el nombre completo del propietario, no sólo su nombre de pila.

Por mucho que tenga la tentación, y la tendrá, nunca, NUNCA ceda a la comodidad y se fíe de una clave pública que haya recibido de una BBS, a menos que vaya firmada por alguien en quien confie. Esa clave pública sin certificar puede haber sido manipulada por cualquiera, quizá incluso el mismo administrador de la BBS.

Si le piden que firme el certificado de la clave pública de alguien, compruebe que realmente pertenece a la persona indicada en el identificador de usuario. Su firma en ese certificado de clave pública es su promesa de que la clave pertenece realmente a esa persona. La gente que confía en usted aceptará esa clave pública porque lleva su firma. No es recomendable hacerlo de oídas, no firme la clave a menos que tenga conocimiento independiente y de primera mano de que realmente pertenece a esa persona. Preferiblemente, debería firmarla sólo si la ha recibido directamente de ella.

Debe estar mucho más seguro sobre quién es el propietario de una clave pública para firmarla que para encriptar un mensaje. Para estar suficientemente convencido de la validez de una firma como para utilizarla, deberían bastar las firmas de certificación de las referencias de confianza. En cambio, para firmar una clave usted mismo debe tener conocimiento independiente y de primera mano de quién es el propietario de esa clave. Podría llamarle por teléfono y leerle el fichero de claves, para que confirme que es verdaderamente la suya; compruebe que está hablando con la persona indicada.

Tenga en cuenta que la firma en un certificado de clave pública no responde de la integridad de esa persona, solamente de la integridad (la pertenencia) de la clave pública de esa persona. No arriesga su credibilidad al firmar la clave pública de un sociópata, siempre que esté completamente seguro de que la clave le pertenece. Otras personas aceptarán que le pertenece porque usted la ha firmado (asumiendo que confíen en usted), pero no se fiarán del propietario de esa clave. Confiar en una clave no es lo mismo que confiar en su propietario.

La confianza no es necesariamente transferible. Si me fío de la firma de Alicia en una clave, y Alicia se fía de la firma de Carlos, eso no implica que yo me tenga que fiar de la firma de Carlos.

Resulta conveniente mantener su propia clave pública a mano con una colección de firmas de certificación de diversas "referencias", para

que la mayoría de la gente confíe al menos en una de las que responden de la validez de su clave. Puede enviar la clave con su colección de firmas de certificación a varias BBSs. Si firma la clave pública de alguien, devuélvasela con la firma para que pueda añadirla a su colección de credenciales.

PGP controla qué claves del anillo de claves públicas han sido certificadas adecuadamente con firmas de referencias en las que confía. Todo lo que tiene que hacer es decir a PGP en qué personas confía como referencia y certificar esas claves con la suya propia, que es fundamentalmente fiable. PGP puede continuar desde ahí, validando cualquier clave firmada por esas referencias designadas. Aparte, por supuesto, puede firmar más claves usted mismo. Seguiremos con esto más adelante.

Asegúrese de que nadie pueda manipular su anillo de claves públicas. La comprobación de cualquier nueva firma de clave pública depende en última instancia de la integridad de las claves de confianza que ya se encuentran en el anillo de claves. Mantega control físico sobre el anillo de claves públicas, preferiblemente en su propio ordenador personal en lugar de un sistema remoto multiusuario, tal como lo haría con su clave secreta. El objetivo es protegerlo contra manipulación, no contra exposición. Conserve una copia de seguridad fiable de los anillos de claves públicas y secretas en un medio protegido contra escritura.

Como su propia clave es la máxima autoridad para certificar directa o indirectamente las claves de su anillo, es la que más tiene que proteger contra manipulación. Para detectar cualquier manipulación de su propia clave pública, PGP puede configurarse para que la compare automáticamente con una copia de seguridad en un medio protegido contra escritura. PGP generalmente asume que va a mantener seguridad física sobre el sistema, los anillos de claves y la copia misma de PGP. Si un intruso pudiese manipular su disco, podría en teoría manipular el mismo PGP, dejando en entredicho cualquier sistema de seguridad que pueda tener para detectar la manipulación de claves.

Una forma algo complicada de proteger el anillo completo de claves públicas contra manipulación es firmarlo con su propia clave secreta. Desafortunadamente, sigue siendo necesario mantener una copia

aparte de su propia clave pública, para comprobar la firma que ha realizado. No puede fiarse de la clave almacenada en el anillo de claves públicas, ya que es precisamente parte de lo que intenta comprobar.

¿Cómo controla PGP la validez de las claves?

PGP lleva el control de las claves del anillo de claves públicas que han sido certificadas adecuadamente con firmas de referencias de confianza. Todo lo que tiene que hacer usted es decir a PGP en qué personas confía como referencia, y certificar esas claves con la suya propia. PGP puede continuar desde ahí, validando cualquier otra clave firmada por esas referencias elegidas. Por supuesto, usted mismo puede firmar más claves.

Hay dos criterios completamente distintos por los que PGP juzga la utilidad de una clave pública:

1) ¿Pertenece la clave realmente a quien parece pertenecer? En otras palabras, ¿ha sido certificada con una firma de confianza?

2) ¿Pertenece a alguien en quien podemos confiar para certificar otras claves?

PGP puede calcular la respuesta a la primera pregunta. Para responder a la segunda, usted, el usuario, debe informar a PGP explícitamente. Cuando se da la respuesta a la pregunta 2, PGP puede calcular la respuesta a la pregunta 1 para otras claves que hayan sido firmadas por esa referencia designada como fiable.

Las claves que han sido certificadas por una referencia de confianza ya se consideran válidas en PGP. Las claves de esas referencias deben estar certificadas por usted u otra referencia de confianza.

PGP también permite tener distintos márgenes de confianza para las personas que van a actuar como referencia. La confianza en el propietario de una clave para servir de referencia no refleja simplemente la estimación de su integridad personal, también debería reflejar cuál cree usted que es su nivel de conocimiento respecto a la gestión de claves, y de su buen juicio en la firma de estas. Puede designar una persona en PGP como desconocida, no fiable, de relativa confianza, o de completa confianza para certificar otras claves públicas. Esta informa-

ción se almacena en el anillo junto con la clave de esa persona, pero no se incluye con ella, ya que esas opiniones privadas sobre confianza se consideran confidenciales.

Cuando PGP está calculando la validez de una clave pública, examina el nivel de confianza de todas las firmas incluidas. Elabora una puntuación proporcional de validez, dos firmas relativamente fiables se consideran tan creíbles como una completamente fiable. El escepticismo de PGP es ajustable, por ejemplo, puede establecerse que hagan falta dos firmas completamente fiables, o tres relativamente fiables, para dar una clave por válida.

Su propia clave es "axiomáticamente" válida para PGP y no necesita ninguna firma de referencia para probar su validez. PGP sabe qué claves públicas son suyas buscando las claves secretas correspondientes en el otro anillo. PGP también asume que confía completamente en usted mismo para certificar otras claves.

Según pase el tiempo, irá acumulando claves de otras personas, a las que podrá designar como referencias de confianza. Cada uno irá eligiendo sus propias referencias. Y cada uno irá gradualmente acumulando y distribuyendo con su clave una colección de firmas de certificación, con la esperanza de que cualquiera que la reciba confíe al menos en una o dos de ellas. Se producirá de esa forma la aparición de una red descentralizada de confianza para las claves públicas, resistente a fallos.

Este enfoque de base, único, contrasta claramente con los esquemas habituales del Gobierno para gestionar claves públicas, como el Internet Privacy Enhanced Mail (PEM) {Correo mejorado en intimidad para Internet}, que se fundamentan en un control centralizado y una confianza centralizada y obligatoria. Los esquemas habituales confían en una jerarquía de Autoridades de certificación que dictan en quién debe usted confiar. El método probabilístico y descentralizado de PGP para determinar la legitimidad de las claves públicas es la piedra angular de su arquitectura de gestión de claves. PGP le permite que elija usted mismo en quién confiar, y le pone en el vértice de su propia pirámide personal de certificación. PGP es para personas que prefieren preparar sus propios paracaídas.

¿Cómo proteger las claves secretas contra revelación?

Proteja con cuidado su propia clave y su contraseña. Con mucho, mucho cuidado. Si su clave secreta se ve alguna vez comprometida, es mejor que corra la voz rápidamente y se lo diga a todas las partes interesadas (buena suerte...) antes de que alguien la utilice para hacer firmas en su nombre. Por ejemplo, podría firmar certificados falsos de clave pública, lo que podría causar problemas a muchas personas, especialmente si su firma tiene amplio reconocimiento. Por supuesto, el compromiso de su clave secreta podría poner al descubierto todos los mensajes dirigidos a usted.

Para proteger su clave secreta, puede empezar por mantener siempre control físico sobre ella. Es suficiente con tenerla en el ordenador personal en casa, o en un portátil que pueda llevar consigo. Si tiene que utilizar un ordenador de la oficina, que no siempre controla físicamente, lleve sus anillos de claves públicas y secretas en un disco extraíble, y nunca se lo olvide. No es conveniente permitir que la clave secreta se encuentre en un ordenador remoto multiusuario, como por ejemplo un sistema Unix con acceso telefónico. Alguien podría fisgonear en la línea del módem y conseguir la contraseña, y más adelante conseguir la clave secreta del sistema. Sólo debería utilizar la clave secreta en una máquina sobre la que tenga control físico.

No guarde su contraseña en el mismo ordenador que tiene el anillo de claves secretas. Guardar la clave secreta y la contraseña en el mismo ordenador es tan peligroso como guardar su número secreto en la misma cartera que la tarjeta del cajero automático. Sería más seguro que memorizase la contraseña y que no la guardase en ningún sitio más que en su cerebro. Si cree que debe escribirla, protéjala, quizá incluso mejor que el anillo de claves secretas.

Si ocurre lo peor, tanto su clave secreta como la contraseña se ven comprometidas, tendrá que emitir un certificado de "compromiso de clave". Este tipo de certificado se utiliza para advertir a los demás de que dejen de utilizar su clave pública. Después tiene que enviarlo a sus amigos; sus propios programas PGP instalarán ese certificado de compromiso en sus anillos de claves públicas y evitará que utilicen la clave por error. Puede entonces generar un nuevo par de claves

secreta/pública y distribuir la nueva clave pública. Puede enviar en un solo lote la nueva clave con el certificado de compromiso de la antigua.

Puntos vulnerables

Ningún sistema de seguridad de datos es impenetrable. PGP puede burlarse de diversas maneras. Los posibles puntos vulnerables que hay que tener en cuenta son, entre otros, el compromiso de la contraseña o de la clave secreta, la manipulación de las claves públicas, los ficheros que se han borrado pero que siguen todavía en el disco, los virus y caballos de Troya, los fallos en la seguridad física, las emisiones electromagnéticas, la exposición en sistemas multiusuario, el análisis de tráfico, y quizá incluso el criptoanálisis directo.

Mail - Bombing

El mail - bombing es un tipo de ataque que consiste en, como su propio nombre indica, bombardear una cuenta de correo. Hay muchos programas que realizan este tipo de ataque. Su funcionamiento es realmente sencillo: se escribe el mensaje que se desea mandar, y el programa se encarga de enviarlo de forma automática la cantidad de veces deseada. Este tipo de ataque no provoca otra cosa que la molesta de tarea de tener que llamar a tu ISP (Internet Service Provider) para pedirle que vacíen tu espacio dedicado a correo-E, con la posible pérdida de otros correos importantes, o bien vaciarlo tu mismo con alguna aplicación diseñada para ello.

El problema del mail-bombing es que no se puede realizar de forma anónima a través de remailers (ya que los remailers no permiten ni mail-bombing ni mail-spamming) con lo cual se puede rastrear el origen de dicho ataque, aunque se puede realizar a través de algún servidor de correo de alguna universidad o similar con lo que dificultaría bastante encontrar el autor del ataque.

Bibliografia sobre privacidad

www.anonymizer.com
www.salteador.com
www.pgp.com
www.uk.pgp.net
www.c2.org

Protocolo TCP/IP

Un protocolo no es más que un conjunto de normas o reglas a las que nos ceñimos para que la comunicación sea posible en cualquier campo de la vida. Los protocolos que se utilizan en la red no son sino eso, unas reglas definidas como estándar.

El protocolo más utilizado en Internet es el TCP/IP. Realmente, no es un único protocolo, es la unión de varios lo que forman los llamados protocolos de Internet, siendo los mas importantes el IP y el TCP. Para entender sus defectos y vulnerabilidades (ya que es eso lo que aquí se trata) hay que hacer una pequeña introducción.

Fundamentalmente hay dos tipos de protocolos: los orientados a la conexión y los que no lo son. La diferencia fundamental entre ellos es sencilla: los orientados establecen una comunicación entre dos maquinas y tienen control de errores. Los no orientados mandan su mensaje y se despreocupan, no les importa si llega a su destino o no; es una actitud francamente optimista. Pues bien, de los dos protocolos anteriormente citados, el TCP es orientado a la conexión y el IP es no orientado.

La falta de verificaciones hace de este protocolo IP un objetivo perfecto para el ataque. Lo único que identifica a los que intervienen en la conexión es su dirección IP y ésta es fácilmente cambiable. Con unos conocimientos básicos de programación se puede conseguir que el destinatario de un paquete piense que es él mismo el que manda y recibe un mensaje, con los peligros que ello conlleva.

Como ya se ha comentado, el IP es un servicio no orientado a la conexión, es decir, permite el intercambio de datos entre ordenadores sin establecimiento previo de llamada. Para tareas de fiabilidad se encarga de tareas de control, secuenciamiento, aperturas, cierres, etc.

Uno de los problemas más importantes que tiene el protocolo es que es muy conocido y es relativamente sencillo. Además, cada sistema operativo (Windows 3.11, Windows 95, Windows NT, UNIX, macOs, OS/2, etc) tiene diferentes implementaciones de los mismos protocolos, es decir, hacen lo mismo y tienen diferentes fallos.

El ataque básico que afecta al protocolo lo que busca es la desconexión de la red de la víctima, ni más ni menos, forzando a que reiniciar su maquina y volver a llamar por teléfono.

De este ataque hay muchas variaciones que explotan diferentes fallos en distintos sistemas. El primero que apareció afectaba a los sistemas operativos de Microsoft: sólo tenía que decirle a la víctima que el mensaje era de un tipo especifico.

Otros como el jolt mandan paquetes falsos a la víctima hasta provocar la perdida de conexión. El jolt además falsea la dirección del asaltante para dificultar su localización, esto es ya una cosa común a todos estos tipos de ataques. El land y sus variaciones básicamente mandan un paquete con la dirección de destino igual que la de origen para que la víctima entre en un círculo hasta que se cuelgue. Los más actuales como el teardrop explotan errores en la recepción de mensajes fragmentados: le dice a la víctima que le manda más o menos paquetes de los que llegan en realidad.

Para solucionar el problema que esto supone los distintos fabricantes de sistemas operativos sacan series de parches. En Windows NT la mayoría de los fallos se solucionan instalando el SP3, y en los sistemas UNIX actualizando la versión del núcleo.

Otra forma de solucionarlo es utilizando un cortafuegos o firewall, con el que se pueden bloquear los paquetes sospechosos de constituir un ataque.

De todas formas ninguna de estas dos soluciones es completamente segura, pero es mejor que no tener nada.

IP spoofing

En cuanto a la seguridad del protocolo, el mayor peligro que tiene es la posibilidad de variación de los contenidos de los mensajes IP. Esta

técnica se denomina spoofing. Actualmente, el spoofing no es el ataque en sí, sino un paso del mismo.

El IP es fácilmente atacable, pero el TCP, al ser orientado a la conexión como explicamos antes, es más complicado de atacar. El TCP es fiable, es capaz de recuperar datos perdidos, duplicados, cambiados de orden. Esto lo hace asignando números a cada mensaje enviado, utilizados por el receptor para ordenarlos, eliminar los duplicados, pedir los que le falten en el caso de que ocurra, etc.

Es decir, manda un mensaje con un numero, y el receptor le manda otro certificando que ha llegado correctamente.

Para el intercambio, los ordenadores tienen que establecer la conexión. El TCP la establece en tres pasos.

1 A ---SYN---> B
2 A <---SYN/ACK--- B
3 A ---ACK---> B

En (1) el cliente le dice al servidor que quiere conectarse. Este es el propósito del flag SYN. El cliente le dice al servidor que el campo del número de secuencia es válido y que debe ser comprobado. El cliente debe indicar el número de secuencia en el campo de la cabecera del TCP como ISN (número de secuencia inicial). El servidor después de recibir este segmento (2) debe mandarle su propia ISN y un ACKnowledgement del primer segmento del cliente (que será ISN+1). El cliente le manda el ACK del ISN del servidor y la transferencia puede tener lugar.

Es importante comprender cómo los números de secuencia son seleccionados inicialmente, y cómo cambian con el tiempo.

El IP-Spoofing consta de muchos pasos. Primero hay que elegir a la maquina objetivo, después tenemos que encontrar las relaciones de confianza entre las máquinas. La relación de confianza es desactivada y los números de secuencia TCP son analizados. Se suplanta a la máquina, los números de secuencia se copian y se intenta establecer una comunicación que sólo requiere autentificación basada en direcciones. Si tiene éxito, el atacante ejecutará un simple comando para dejar un backdoor.

Lo necesario para hacer este ataque es un ordenador objetivo, relaciones de confianza, una máquina atacante (con perfil de root), y

software de IP-spoofing. Generalmente este ataque se hace desde una cuenta root contra la cuenta root de la víctima.

Un factor muy importante en IP-spoofing y muy a tener en cuenta es que es un ataque ciego. El atacante va a tomar la identidad de un puesto con relaciones de confianza para deshabilitar la seguridad de la víctima. Después de que el objetivo se ha seleccionado, el atacante debe determinar las relaciones de confianza (suponemos que existen, de otra forma, el ataque termina aquí), si no se tiene información de ese tipo, el último recurso está en probar las direcciones del mismo tipo que la víctima.

Una vez que hemos encontrado al trusted host, debemos deshabilitarlo. Desde el momento que se va a suplantar debemos asegurarnos que no recibe ningún tráfico de la red.

Prevención del ataque

Una manera fácil de solucionar el problema es no utilizar autentificaciones basadas en las direcciones. Deshabilite todos lo comandos r*, quite todos los ficheros .rhosts y borre el contenido del /etc/hosts.equiv. Esto forzará a todos los usuarios a utilizar otros medios de acceso remoto (telnet, ssh, skey, etc).

Si su site tiene conexión directa a Internet, puede usar su router para ayudarle. Primero asegúrese que sólo los host de su red interna pueden participar en las relaciones de confianza. Después simplemente filtre todo el tráfico de fuera (Internet) que intente entrar.

Un método obvio para detectar IP-spoofing es requerir que todo el tráfico de la red este encriptado y/o autentificado.

Cortar una conexión ya establecida

Para cortar una comunicación tenemos que tener en cuenta que:

```
host A <------X------------------------>host B
          |       A,B tienen una conexión TCP en marcha
 host S <------/       S está en la misma subnet
```

Cortar la conexión con RST

Es un flag que se usa para cortar la conexión. Para ser aceptado, el número de secuencia tiene que ser el correcto (no hay ACK en un paquete RST). Calculamos el número de secuencia y ya está.

Cortar una conexión con FIN es similar a lo hecho hasta ahora. Un flag FIN quiere decir que no hay más información para ser enviada. La diferencia es que el host objetivo le manda un paquete de confirmación de corte.

Con el hijacking tenemos un método por el cual podemos "robar" una conexión generada por una aplicación de red iniciada por un cliente. Generalmente, la aplicación de red que más nos interesa para estos fines es el TELNET., y lo que vamos a hacer es hacernos pasar por el cliente que ha iniciado esa aplicación, sustituyéndolo y tomando los mandos de la aplicación.

Escaneo de puertos

El escaneo es un método para descubrir puertos de comunicación explotables. La idea es probar cuantos más puertos se pueda mejor, para encontrar cuales están preparados para recibir información.

Tipos

Desde hace un tiempo se han desarrollado muchas técnicas para verificar que puertos y protocolos estaban escuchando en una máquina. Algunos de los más comunes son:

- TCP connect() scanning: Esta es la más básica forma de escaneo de TCP. Con connect() su sistema inicia una conexión con los puertos más interesantes de la máquina. Si el puerto está escuchando, connect() tendrá éxito y no lo tendrá si está cerrado. Una gran ventaja de este método es que no necesita privilegios especiales, y es de los escaneos más rápidos. Su mayor desventaja reside en que es fácilmente filtrable y detectable.

- TCP SYN scanning: A esta técnica se le suele llamar "half-open" scanning porque no hace una conexión completa. Usted manda un paquete SYN como si fuera a establecer una conexión y espera que le responda, un SYN/ACK indica que el puerto está escuchando. Si recibe un SYN/ACK manda inmediatamente un RST para cortar la comunicación. La principal ventaja es que pocos sitios tienen log de este tipo, desafortunadamente, necesita ser root para utilizarlo.

- TCP FIN scanning: Algunos firewalls y programas como el synlogger son capaces de detectar ese tipo de escaneo. Con los paquetes FIN podemos pasar entre esas cosas molestas, la idea es que los puertos cerrados responden a un FIN con un RST, y los que está escuchando ignoran ese paquete en cuestión. Esto es un fallo del TCP así que no es 100% seguro.

- Fragmentation scanning: Este no es un método nuevo de escaneo en sí mismo, sino una modificación de otras técnicas. En lugar de mandar en paquete de prueba, lo que hace es mandarlo en pequeños fragmentos para hacer más difícil su detección.

- FTP bounce attack: Uno de los aspectos más interesantes del protocolo FTP es su soporte para conexiones proxy FTP. En otras palabras, puedo conectar desde evil.com al servidor FTP de target.com para establecer una conexión de control de comunicaciones, después puedo pedirle al servidor que inicie un DTP (data transfer process para enviar un fichero a cualquier sitio de internet). Utilizando esa base se puede usar para hacer escaneo de puertos.

- UDP ICMP port unreachable scannig: Este método varía de los anteriores en que utiliza el protocolo UDP en lugar del TCP. Escanear a este protocolo es significativamente más difícil porque los puertos abiertos no tienen que mandar un ACK de respuesta a nuestra prueba y los cerrados no tienen que mandar un mensaje de error. Afortunadamente muchos host te mandan un ICMP_PORT_UNREACH error cuando mandas un paquete a un puerto UDP cerrado. Ni los paquetes UDP ni los ICMP tienen seguridad de que lleguen. Es muy lento este tipo de escaneo.

- UDP recvfrom() and write() scanning: Cuando hace un recvform() contra un puerto UDP abierto te devuelve un EAGAIN si el error ICMP no ha sido recibido, y un ECONNREFUSED si sí lo ha sido.

- ICMP echo scanning: Esto no es en realidad un escaneo de puertos pero algunas veces puede llegar a ser útil para determinar que host en una red están funcionando haciéndoles pings.

Bibliografía sobre TCP/IP

- Libros: TCP/IP Illustrated; CIFS: Common Insecurities Fail Scrutiny; Redes de Ordenadores.
- RFCs: 768; 791; 792; 793; 1180; 1752; 1825; 1948
- Revistas en papel: PC Actual nº 93
- Textos: Windows Sockets (An Open Interface for Network Programming under Microsoft Windows v1.1); BSD Sockets (A Quick And Dirty Primer); Windows Sockets (A Quick And Dirty Primer); Programming UNIX Sockets in C - Frequently Asked Questions
- Revistas: Confidence Remains High 001, 003, 666; Phrack P48-14, P49-06, P49-07, P49-15, P51-06, P51-11; SET 09, 12, 13;
- Direcciones web: www.rootshell.com; www.islatortuga.com; www.phrack.com; www.codez.com; www.l0pht.com; www.underground.org; www.2600.com; www.geek-girl.com/bugtraq; www.angelfire.com/mi/JJFHackers;
- Listas de correo: bugtraq@netspace.org
- Otros: R34.linux, R34.internet, 425.linux (áreas de fidonet)

Bugs de Sistemas

Podemos definir un bug, de forma simple, como un error producido por un determinado programa. Estos errores, cuando esos programas resultan ser el propio sistema operativo, dan lugar a fallos de diversa índole en la seguridad de los sistemas informáticos, desde la pérdida de estabilidad del sistema operativo hasta la incorrecta asignación de permisos de los usuarios, pasando por todo tipo de acciones peligrosas. Aunque debemos señalar que no todos los bugs son peligrosos para la seguridad; por ejemplo, un fallo en las librerías gráficas no provocará fallos en la seguridad del sistema, aunque cause moles-

tias visuales al usuario.

Estos errores son descubiertos de muy diversas maneras: por la misma compañía desarrolladora del S.O., o bien por algún usuario, fruto de la casualidad, o más comúnmente, por las investigaciones que los hackers de alto nivel realizan con el objetivo de lograr el control de las máquinas en las que ese S.O. está instalado.

En Internet podemos encontrar varias bugtraps (listas de bugs) en las que, clasificados por sistemas operativos, podemos ver cuales son los bugs más recientes. Algunas de estas listas son de actualización casi diaria, e incluso incluyen los exploits (programas ejecutables o instrucciones paso por paso) necesarios para lograr producir ese error. Una tarea básica para el administrador de sistemas será la lectura de estas listas, con el objetivo de subsanar los errores presentes en todos los SS.OO. Naturalmente, los hackers sí las leen.

A continuación observaremos algunos bugs de los SS.OO. más comunes en el mercado.

Windows

Fallos generales de la red Windows

Estos fallos son aplicables a cualquier red Windows, en la que nos encontraremos con puestos Windows 3.x, 95 y servidores NT. Son independientes de las versiones de estos, y solo hemos comentado aquí dos de los más representativos. Casi cualquier programa de tecnología Microsoft tiene una lista de bugs considerable, y no se libran de ellos ni siquiera aquellos de los que más se pregona su seguridad. Sirva como ejemplo Windows NT Server 4.0, del cual Microsoft asegura su estabilidad y su cumplimiento como sistema con seguridad de clase C2, mientras en cualquier bugtrap (sitios en Internet que contienen una lista de bugs) podemos obtener como mínimo 25 fallos de seguridad de este servidor.

ActiveX

Un control ActiveX no es más que un componente de un programa que puede ser descargado de una máquina servidor y ejecutado en una máquina cliente. Los controles ActiveX pueden resultar tan útiles como letales. Cualquier control ActiveX puede contener código que dañe nuestro sistema, dado que puede contener cualquier tipo de código, incluidos, por supuesto, los víricos. Se debe de tener cuidado al descargar cualquier control de Internet, dado que a pesar de estar autentificado nada garantiza la identidad del fabricante del control. Esto es así por la tecnología empleada por Microsoft en su denominada "Microsoft's AuthentiCode technology", que permite la distribución de cualquier código, sin realmente demostrar la identidad del creador. Basta conectarse a un WEB, dar un nombre, dirección, número de tarjeta de crédito y alguna otra información (naturalmente, toda puede ser falsa, incluido el número de VISA, Internet está llena de número de VISA válidos), hacer clik en el botón "I Agree" tras leer un montón de inútil jerga legal, y esperar a que nos llegue un email con la información que debemos añadir a nuestro control para que quede registrado. Naturalmente, este email también puede ser recogido de cualquier sitio que garantice el anonimato. Nadie se preocupará de revisar nuestro control ActiveX. Un ejemplo no dañino de control ActiveX registrado de esta manera esta disponible en http://www.halcyon.com/mclain/ActiveX. Se llama "Explorer" y pretende ser una demostración de lo peligrosa de esta forma de registro automático.

Seguridad en la compartición de recursos

La seguridad de la compartición de recursos en redes Windows se basa en la protección mediante contraseñas para acceder al recurso, de tal manera que si no se conoce la contraseña el acceso al recurso compartido será denegado. Ahora bien, la seguridad en la encriptación de estas contraseñas deja mucho que desear por su debilidad. Hoy estos algoritmos de "encriptación" son conocidos, y cualquier navegante de Internet puede dar con ellos. Por lo tanto, estas passwords pueden ser averiguadas por cualquiera. Esta debilidad de las contra-

señas se debe a los siguientes puntos:

- Las contraseñas pueden tener una longitud variable de uno a ocho caracteres, no obligándose a la utilización de un mínimo de ellos.
- Las contraseñas de "Solo lectura" y "Acceso total" usan el mismo esquema de codificación, y por tanto, las dos pueden ser desencriptadas utilizando el mismo método.
- Hay una relación de "uno a uno" entre el número de los caracteres en la contraseña y el número de bytes de la contraseña encriptada guardada en el registro. De hecho, cada byte encriptado corresponde directamente al carácter de la misma posición en la contraseña. Es decir, si yo cambio la contraseña "cazo" a "pazo", sólo se cambiará el primer byte de la contraseña encriptada.
- El algoritmo de encriptación para cada carácter varia ligeramente con respecto al del anterior carácter, pero es conocido para todos.
- Todos los caracteres utilizados en una password se encontrarán, o tendrán su homólogo en códigos ASCII del 32 a 159. Si se utilizan otros superiores, podrán ser sustituidos por caracteres pertenecientes a ese rango y la password seguirá siendo válida.

Podemos considerar los ataques con el objetivo de averiguar estas passwords desde 2 puntos de vista, suponiendo que el intruso ya está como usuario en la red:

- A distancia: Utilizando el programa "l0pthcrack", de "L0pthCrak Heavy Industries" (http://www.l0pthcrack.com, aunque suele cambiar su dirección como norma habitual), podemos escanear los paquetes de la red en busca de aquellos que contienen una contraseña.
- Con acceso local a la máquina: Las passwords "encriptadas" de Windows 95 se guardan en la rama del registro: "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Network\LanMan", debajo de la cual se guardan 2 entradas, Parm1enc y Parm2enc, la primera corresponde a la clave para "Acceso total", la segunda es la de "Solo lectura". Si conocemos el algoritmo, y sabemos

donde encontrar esas passwords, solo queda decodificarlas. Es un juego de niños.

No obstante, ni siquiera necesitamos escribir por nosotros mismos el programa que las decodifique. En Internet damos fácilmente con algunos ya escritos, como "Revelation 1.1", creado por Snadboy Software. Este programa descubre las passwords que se ocultan tras las líneas de asteriscos y las presenta en texto plano. La potencia de este programa es elevada, ya que permite averiguar passwords de recursos compartidos, salvapantallas, conexiones telefónicas a redes, etc. El tamaño del programa es de 15 Kb., lo que permite que sea guardado en un disquete. Puede ser obtenido en más de 20 FTP-sites que se indican en la página WEB de Snadboy: http://www.snadboy.com. Como única debilidad del programa, destacar que solo funciona en modo local, es decir, no es capaz de averiguar passwords de máquinas remotas.

Consejos comunes a toda red Windows

- Se debe de seguir una política de claves estricta, para evitar que estas sean averiguadas con diccionarios.
- Se deben de evitar en redes Windows que se compartan recursos "comprometidos", ya que las passwords que se guardan en ficheros PWL son fácilmente desencriptables.

Windows NT

En este punto hablaremos sobre la seguridad de NT. Ya señalamos anteriormente que es fundamentalmente inseguro en su estado "por defecto", es decir, tal y como queda cuando se instala. Con los parches aplicados su seguridad aumenta bastantes enteros, pero aún así, como cualquier S.O. es imperfecto y puede ser atacado. Hemos recogido a continuación algunos de los bugs más significativos y más peligrosos con los que nos podemos encontrar, y hemos propuesto soluciones cuando ha sido posible para remediar estos problemas:

Bug del inetinfo.exe

Versiones afectadas: Win NT 4.0

Problema: NT es vulnerable a un ataque que utiliza el puerto 1031 (inetinfo). Este bug se puede explotar solo de forma local (es decir, cuando estamos conectados a esa LAN), basta con hacer telnet al puerto 1031 de la máquina NT, escribir "garbage" y desconectar, aunque parece ser que también funciona simplemente conectando y desconectando sin necesidad de escribir nada. El fallo viene del software IIS, y dado que este escoge los puertos de forma aleatoria, puede ser que el servicio inetinfo se encuentre en otro distinto del 1031.

- Solución: La única solución disponible hoy en día es cerrar todos aquellos puertos que no se utilicen.

Bug de algunos servidores SMTP

- Versiones afectadas: Cualquier Windows con los servidores de correo Microsoft Exchange o MDaemon 2.71
- Problema: Algunos de estos fallos son comunes a los dos, pero el más afectado es el Mdaemon. Existen varios bug detectados, el más peligroso de ellos es la posibilidad de parar los servicios de correo SMTP/POP. Esto se logra accediendo al puerto de configuración, que es el siguiente después del que utiliza el WebPop (WebPop suele funcionar en el puerto 8080, por lo que el de configuración será el 8081). Para bloquear el servidor después de conectar al puerto de configuración basta poner el comando USER seguido de unos 2000 caracteres. Se producirá un error de "buffer overflow", y saldrá el mensaje "Connection closed by foreign host", tras lo cual se dejarán de prestar servicios de SMTP y POP. El mayor peligro de este tipo de bug es que permite ejecutar cualquier comando en el servidor en que resida.
- Solución: Instalar el Service Pack 3 de Microsoft para solucionar los problemas de MS Exchange, o los parches del Mdaemon. En cualquiera de los dos casos, si no existiesen parches, se puede

utilizar un Firewall que bloquee los puertos "defectuosos", si es que es posible prescindir de los servicios que se presten a través de ellos.

Bug del cd..

- Versiones afectadas: Windows NT 3.5 y 3.51, Windows 95 también.
- Problema: Si un cliente Samba se conecta a un recurso compartido en una máquina NT y ejecuta el comando "cd .." en el directorio raíz del recurso compartido, causa una excepción en el kernel (STOP 0x0000001E). Según como esté configurada la máquina, puede reiniciarse automáticamente o bien requerir intervención manual del administrador. Si la máquina accedida funciona bajo Windows 95, ¡se obtendrá acceso en todo el disco duro!.
- Solución: En Windows NT 3.5 no existe. En Windows 3.51, este problema se soluciona instalando el Service Pack de Microsoft 4 ó 5.

IIS

- Versiones afectadas: Todos aquellos sistemas que instalen Microsoft Internet Information Server.
- Problema: Un intruso cualquiera es capaz de ejecutar comandos en el servidor NT que tenga servicios de Servidor WWW, ya que si se tolera la ejecución de scripts CGI en los directorios de los usuarios (que por defecto está activa), estos pueden poner scripts .BAT o .CMD en lugar de CGIs, que serán ejecutados en la máquina y no en el servidor WWW.
- Solución: Desactivar la ejecución de BAT y CMD en el IIS.

Otros fallos de NT

Todas las distribuciones de Windows NT (en cualquiera de sus versiones) presentan errores comunes de concepción en su instalación por defecto. Nunca debemos dejar una instalación de un servidor Windows NT en el estado en que esta se queda tras instalarse, ya que presenta altos riesgos para la seguridad del sistema. Además todos los "Service Pack" deben de ser instalados según vayan apareciendo, ya que realmente constituyen prácticamente por si mismos un nuevo sistema operativo. Los errores a corregir más comunes son los siguientes:

- Es conveniente cambiar el superusuario del sistema a otro que no sea "Administrador" (por todos es conocido que el superusuario es ese), con el objetivo de evitar ataques por el método de fuerza bruta.
- Se deben de revisar los permisos de recursos que se comparten por defecto. Especialmente, durante su instalación, Windows NT4 tiene la irritante manía de colocar en los recursos compartidos al grupo "Todos" con permisos de "Acceso Total".
- El usuario "Guest" o "Invitado" debe de ser eliminado del sistema, a no ser que se necesite específicamente. Este usuario es instalado por defecto en todas las versiones de Windows NT, con lo cual tiene acceso a los recursos que anteriormente señalamos se compartían al grupo "Todos".
- Deben de cerrarse todos aquellos puertos del protocolo TCP-IP que no utilicemos, por defecto quedan abiertos todos.

Sistemas Unix

Entendemos por sistemas Unix cualquiera de los existentes en el mercado. Los más comunes hoy en día son: AIX, BSD, HPUX, IRIX, Linux, SCO, Solaris y SunOS. En cualquiera de ellos lo que perseguiremos al explotar un bug es llegar a tener el nivel de superusuario o root, para ello deberemos seguir 2 pasos fundamentales:

1. Bajarnos o copiar el fichero que contenga las passwords. Suele ser el "passwd", y su localización varía según el S.O. con el que estemos tratando.

2. Tratar el fichero con un buen diccionario y un descifrador de passwords.

Nuestros esfuerzos irán dirigidos a obtener el fichero "passwd" o colarnos en el sistema. Veamos un ejemplo de como podríamos hacerlo:

Ejemplo: La técnica PHF

A pesar de ser un agujero muy antiguo, resuelto prácticamente por todas las nuevas versiones de servidores WWW, aún podemos encontrarnos con algunos sitios en los que funciona. Es realmente simple, se basa en que se permite ejecutar el comando "/bin/cat /etc/passwd", lo que nos devuelve el fichero en el que residen los usuarios y sus claves (en caso de no existir un sistema de "shadow password"). Todo lo que debemos de hacer es ejecutar un navegador y poner la siguiente dirección:

http://web/cgi-bin/phf-Qalias=x%0a/bin/cat%20/etc/passwd

donde:

- web, es la máquina de la que queremos obtener el fichero y
- /etc/passwd, es la ruta completa hasta el fichero que contiene las passwords en ese S.O.

Por ejemplo, si quisiéramos obtener el fichero con las passwords de la máquina www.microsoft.com, y funcionara bajo Unix SCO, bastaría con poner:

http://www.microsoft.com/cgi-bin/phf-Qalias=x%0a/bin/cat%20/etc/passwd

y el fichero passwd aparecería en nuestro navegador. Ya solo quedar copiar y pegar.

Programas que dan problemas

Sendmail , Pine, los demonios inetd y ftpd, sperl, librerías SVGA, servidores WEB (Apache, NCSA, entre otros), mount, term, ping, telnet, y hasta variables de entorno como LD_PRELOAD, los servidores de X, etc... la lista es interminable, y aumenta cada día. TODOS estos programas presentan al menos un bug que otorgará a aquella persona que lo conozca nivel de root en esa máquina.

Las técnicas para colarse en sistemas UNIX son múltiples, variadas, y generalmente, complejas. La más general, y en la que la mayoría de estos programas que otorgan el nivel root, es intentar sobrepasar la capacidad del buffer que maneje el programa. Resulta imposible escribir aquí donde falla cada uno y como solucionarlo; una lista de bugs puede obtenerse en http://www.geek-girl.com/bugtraq. Aquí podemos obtener listas para varios sistemas operativos.

Novell

La seguridad en Novell, en cualquiera de sus versiones, deja de existir dado que pasa por ser de las más amenazadas en Internet. Es posible conseguir programas que den a cualquier usuario el poder de Administrador con solo ejecutar un fichero: "hack.exe" o similar. En la bibliografía pueden observarse direcciones en las que obtener algunos de estos ficheros.

Esta es una lista de programas obtenida de la dirección Internet http://www.angelfire.com/mi/JJFHackersTeam/, muchos de ellos son ejecutables, con los que tomar el control de un sistema que no este protegido es tan fácil como teclear alguno de estos comandos...

Burglar.zip: Usado para que una persona tenga acceso de SYSOP desde el prompt del servidor.

- Control.zip: Da control remoto desde cualquier nodo.
- FAQ.zip - Ultima y no oficial Novell Netware Hack FAQ. (July 22)
- Hack.zip: 2 Exe's que te dan información vital. Por ejemplo, si el SYSOP esta conectado.
- Idleboot.zip: TSR que resetea ordenadores en networks (configurable).
- Madcht.zip: Programa de LAN Chat.
- Novellbfh.zip: Novell Brute Force Hacker.
- Novellffs.zip: Novell Fake File Server.
- Novelhak.txt: Novell Hacking Txt, escrito por los mismos que el FAQ.
- Nwpcrack.zip: Brute Force Hacker for ver. 3.11 and 3.12.

- Pzapdemo.zip: Le da al No-SYSOP la habilidad para resetear ID's y cambiar passwords (Req. Win 3.1 or higher).
- Setpass.zip: Cambia el password del SYSOP.
- Shutdown.zip: Apaga el servidor a cualquier hora, tu decides ;-).
- Shwusr10.zip: Te proporciona informacion vital.
- Novell.zip: Un txt que te enseña algun que otro agujero.
- Novell.zip: Unos cuantos programas, uno de ellos un sniffer del teclado (necesitas acceso al servidor).
- Novhack.zip: Programa que te da acceso de SYSOP (necesitas acceso el servidor).

Bibliografía sobre bugs de sistemas.

www.halcyon.com/mclain/ActiveX
www.l0pthcrack.com
www.snadboy.com
www.rootshell.com
www.geek-girl.com/bugtraq
www.angelfire.com/mi/JJFHackersTeam/

Virus

Definido como una intromisión, ilegítima, en un derecho básico del titular, los conocidos "virus informáticos" están consiguiendo marcar un campo en la informática. Su numero aumenta descomunalmente día a día y se prevé que en el año 2001 podrían existir diez millones de virus en circulación, y hasta hoy se conocen como mínimo mas de 5000 programas en el mundo con estas características.

Podríamos remontarnos al año 1949, para encontrarnos el primer indicio de definición de virus. John Von Neumann, experto en teoría de ordenadores, expone su teoría de programas con capacidad de multiplicarse en el articulo: "Teoría y organización de un autómata complicado". Diez años después, en los laboratorios AT&T Bell, inventan el juego Guerra Nuclear (Core Wars). Consistía en una batalla

entre los códigos de dos programadores, en la que cada jugador desarrollaba un programa cuya misión era la de acaparar la máxima memoria posible mediante la reproducción de sí mismo. En esta lucha, cada uno de los programas intentaba destruir al del oponente y tras un tiempo predeterminado ganaba quien tuviera la mayor cantidad de memoria ocupada con su programa. En 1983 Core Wars, que ya contaba con adeptos en el Instituto de Tecnología de Massachusetts (MIT) y en el Centro de Investigación de Xerox en Palo Alto, salió a la luz pública en un discurso de Ken Thompson en la entrega del premio Turing. Ese mismo año aparece el termino "virus" tal como lo entendemos hoy día, Fred Cohen en su tesis doctoral lo definió como "un programa que puede infectar otros programas modificándolos para incluir una versión de sí mismo". Presentó diversos experimentos donde demostraba la factibilidad de estos engendros y probó las limitaciones para defendernos de ellos y la imposibilidad de diseñar un sistema de detección universal. En los años 1986-87 se produce la explosión del fenómeno virus en PCs. Fue en el entorno universitario donde se detectaron los primeros casos de infección masiva.

Propiedades de un virus informático

Es capaz de generar copias de si mismo en ficheros o lugares distintos de donde se encuentra. Se introduce dentro del código de otro programa, pudiendo ejecutarse de este modo de forma parásita, siendo activado inconscientemente por el usuario en sus tareas diarias mediante programas del propio usuario o del sistema operativo, siendo este último su verdadero objetivo, ya que teniendo el control sobre el sistema operativo, se tiene en consecuencia el control sobre todo lo que se encuentra en el sistema informático infectado.

Las acciones para las que han sido programados, van desde un simple mensaje, hasta la destrucción total de los datos del soporte físico de estos, pasando por toda clase de variedades con mayor o menor grado de ingenio por parte del programador, pero no debemos olvidar que su funcionamiento es básicamente el mismo que el de cualquier otro programa que pueda existir en el mercado, pero con objetivos

claramente distintos, con esto queremos decir que un virus informático no es peligroso por el hecho de estar grabado en un disco duro, sino que necesita de una fuente externa que lo active para comenzar a ejecutarse, y poder así realizar su cometido, y de ahí que busque otro programa en el que incluirse de forma parásita, ya que mientras no se ejecute y se cargue en memoria, todo su código es totalmente inofensivo

Clases de virus

Según los criterios de los informáticos, hay quienes distinguen entre virus y programas de características similares (gusanos, caballos de Troya o bombas lógicas) pero que no tienen tal denominación, nosotros no vamos a entrar en el análisis de cuales son o no virus informáticos, tratando a todos como tal ya que su actividad es perniciosa en cualquiera de los casos. Gusano, Caballo de Troya, bomba Lógica, polimórficos, parásitos, virus de disquete, virus de sector de arranque, virus de macro, virus de multipartición, virus de encriptación, en fin, toda una gama para todos los gustos, aunque estos se asocian en pocos grupos.

Los virus son, sin duda alguna, los reyes de los programas dañinos. Sin embargo, no debemos olvidar que existen otras rutinas que pueden igualmente causarnos destrozos en nuestros sistemas. Los gusanos y conejos son programas que comparten con los virus su capacidad de reproducción. Tienen como objetivo realizar múltiples copias de sí mismo, lo que suele terminar por desbordar y colapsar al sistema. El gusano más famoso fue el de Robert Morris, que consiguió bloquear la red ARPAnet. No es casualidad que su padre fuera uno de los implicados en el desarrollo del Unix y pionero en las Core Wars.

Los virus gusano

Se propagan rápidamente a través de las redes. Gusano es un programa que se desplaza por la memoria interna del ordenador con

identidad propia, a diferencia del virus, que generalmente se adhiere a otros programas, está diseñado específicamente para que busque zonas de memoria desocupadas, donde se autoduplica sucesivamente produciendo un desbordamiento físico de la memoria y además, a diferencia de los virus, mantienen comunicación con el programa por el que han sido creados, su ámbito de actuación son redes publicas, y redes de área local, utilizando el correo electrónico como medio de propagación.

Los Caballo de Troya o simplemente troyanos

Son programas que se presentan en forma de aplicación "normal", pero que en su interior poseen código destructivo. Hay que destacar que los Caballos de Troya, a diferencia de los virus, no tienen capacidad de replicación. Se suelen presentar como utilidades comunes, por lo que podemos encontrarnos con Caballos de Troya que simulan el compresor ARJ o el antivirus McAfee. Uno de los troyanos que más impacto causó, por la repercusión en los medios de comunicación, fue el conocido AIDS.

Las bombas lógicas

Son programas que se ejecutan al producirse un hecho determinado; la condición puede ser una fecha, combinación de teclas, etc., y si no se produce la condición el programa permanece oculto sin ejercer ninguna acción. Esta técnica es, en ocasiones, utilizada fraudulentamente por programadores a medida en sus aplicaciones. Consiguen que en determinadas fechas el programa genere un error, de manera que el cliente esté forzado a recurrir al programador para subsanarlo, asegurándose de esta forma el mantenimiento.

No podemos olvidarnos de los virus, programas que utilizan características de los gusanos, Caballos de Troya y bombas lógicas. Así pues, se reproducen, se introducen en aplicaciones originales y pueden causar diferentes efectos cuando se cumple alguna determinada condi-

ción.

Por último, los applets Java y ActiveX, de la mano de los lenguajes orientados a Internet, han permitido la potenciación y flexibilidad de los desarrollos en la Red. Sin embargo, estas nuevas tecnologías abren también un nuevo mundo a explotar por los creadores de virus. Si bien aún no se ha producido un uso masivo de estas técnicas, pruebas realizadas de forma aislada vienen a demostrar la factibilidad del uso de estos lenguajes para realizar las funciones de los programas anteriormente comentados. La mayoría de las casas antivirus hace tiempo que vienen dando a sus productos un enfoque orientado a la Red. En estos momentos, y ante la amenaza que se prevé, son muchas las que han comenzado a implementar técnicas de detección de applets Java y controles ActiveX maliciosos.

Funcionamiento

Hay que tener siempre en cuenta que un virus es simplemente un programa. Por lo tanto, debemos dejar a un lado las histerias y los miedos infundados y al mismo tiempo ser conscientes del daño real que pueden causarnos. Para ello, lo mejor es tener un buen conocimiento de cómo funcionan y las medidas que debemos de tomar para prevenirlos y hacerles frente. El nacimiento de un virus parte de personas con un alto grado de conocimientos de programación. Las motivaciones que pueden llevar a crearlos son de lo mas variadas y quedan fuera del propósito de este artículo. Estos programas pueden desarrollarse en distintos lenguajes de programación, siendo el ensamblador el más utilizado por su potencia. El objetivo del virus consiste en replicarse a sí mismo de forma transparente al usuario y dificultando al máximo su detección. Para poder replicarse necesita ser ejecutado en el ordenador, por lo que recurre, de manera habitual, a unirse a ficheros ejecutables modificándolos o situándose en los sectores de arranques y tabla de particiones de los discos. Una vez que se ejecutan, ya sea por abrir un fichero infectado o por hacer una operación de un disco con el boot infectado, suelen quedar residentes en memoria a la espera de infectar otros ficheros y discos. Los virus residentes interceptan los

vectores de interrupción, modificando la tabla que los contienen, para que apunten a su código. Los vectores son los encargados de prestar los servicios del sistema; de esta manera, cuando una aplicación llame a uno de esos servicios el control es cedido al virus. Con el control del sistema, el virus se dispone a la replicación, ya que una llamada al servicio de ejecución o copia de un fichero puede ser interceptada gracias a las modificaciones de los vectores de interrupción y proceder a su infección.

Lo más usual para ello consiste en añadir el código vírico al final del fichero y modificar la cabecera de éste para que apunte al virus. Al final del código del virus habrá un nuevo salto al comienzo del programa original para que se ejecute con normalidad y el usuario no sospeche. Por último, el virus suele contener un efecto que se hará visible en determinadas circunstancias. Una fecha o un numero concreto de infecciones son comúnmente utilizados para hacer despertar el efecto, que puede variar desde inocentes mensajes en pantalla hasta la perdida total de la información de nuestro disco duro.

Los virus más avanzados utilizan técnicas para hacer más efectivo su trabajo. Así, mediante la técnica de Stealth el virus esconde los signos visibles de la infección que podrían delatar su presencia. Con el Tunneling intentan burlar los módulos residentes de los antivirus mediante punteros directos a los vectores de interrupción. Los módulos residentes de los antivirus funcionan de forma parecida a los virus, interceptando los servicios del sistema, pero lógicamente con un propósito totalmente diferente. Otra técnica muy utilizada es la autoencriptación, que permite que el virus se encripte de manera diferente cada vez que infecta un fichero. De esta forma dificulta la labor de detección de los antivirus. Normalmente son detectados por la presencia de la rutina de desencriptación, ya que ésta no varía. La contramedida de los virus para impedir ser detectados de esta forma es variar el método de encriptación de generación en generación. Es decir, que entre distintos ejemplares del mismo virus no existen coincidencias ni siquiera en la parte del virus que se encarga de la desencriptación; son los llamados virus polimórficos. En esta guerra abierta, los creadores de virus en su afán de no ser detectados por los antivirus llegan a implementar técnicas específicas para burlarlos.

Tipos de virus

Dependiendo del lugar donde se alojan, la técnica de replicación o la plataforma en la cual trabajan, podemos diferenciar en distintos tipos de virus. Los virus de Boot utilizan el sector de arranque, el cual contiene la información sobre el tipo de disco, es decir, número de pistas, sectores, caras, tamaño de la FAT, sector de comienzo, etc. A todo esto hay que sumarle un pequeño programa de arranque que verifica si el disco puede cargar el sistema operativo. Los virus de Boot utilizan este sector de arranque para ubicarse, guardando el sector original en otra parte del disco. En muchas ocasiones el virus marca los sectores donde guarda el boot original como defectuosos; de esta forma impiden que sean borrados. En el caso de los discos duros pueden utilizar también la tabla de particiones como ubicación. Suelen quedar residentes en memoria al hacer cualquier operación en un disco infectado, a la espera de replicarse en otros. Como ejemplos representativos tenemos al ya mencionado Brain. Los virus de fichero, como su propio nombre indica, infectan archivos y tradicionalmente los tipos ejecutables COM y EXE han sido los más afectados.

Dentro de la categoría de virus de ficheros podemos encontrar más subdivisiones. Así, los virus de acción directa son aquellos que no quedan residentes en memoria y que se replican en el momento de ejecutarse un fichero infectado. Los virus de sobreescritura corrompen el fichero donde se ubican al sobreescribirlo.

Los virus de compañía aprovechan una característica del DOS, gracias a la cual si llamamos a un archivo para ejecutarlo sin indicar la extensión el sistema operativo buscará en primer lugar el tipo COM. Este tipo de virus no modifica el programa original, sino que cuando encuentra un archivo tipo EXE crea otro de igual nombre conteniendo el virus con extensión COM. De manera que cuando tecleemos el nombre ejecutaremos en primer lugar el virus, y posteriormente éste pasará el control a la aplicación original. Una familia de virus de reciente aparición y gran expansión son los virus de macro. Normalmente insertan el código del virus al principio o al final del archivo. Cuando se

ejecuta, el virus puede hacerse residente en memoria y luego devuelve el control al programa original para que continúe de modo normal. En estos momentos son los ficheros de documentos (DOC, XLS, SAM...) los que están en boga gracias a los virus de macro. Se ha de destacar, de este tipo de virus, que son multiplataformas en cuanto a sistemas operativos, ya que dependen únicamente de la aplicación. Hoy día son el tipo de virus que están teniendo un mayor auge debido a que son fáciles de programar y de distribuir a través de Internet, y aún no existe una concienciación del peligro que puede representar un simple documento de texto. Sin duda, el más extendido de este tipo de virus fue el Concept, gracias al descuido de Microsoft que lo incorporó accidentalmente en un CD, el cual se distribuyó por millares a mediados del año 1995.

Sin duda alguna, el ensamblador es el lenguaje por excelencia en la creación de virus. Ningún otro lenguaje puede ejercer tal control sobre el sistema, ni permite tan alto grado de optimización. En realidad, es prácticamente factible programarlos en cualquier lenguaje, desde C hasta Basic. Los virus BAT vienen a reafirmar este hecho, ya que empleando ordenes DOS en archivos de proceso por lotes consiguen replicarse y provocar efectos dañinos como cualquier otro tipo virus.

En ocasiones, los ficheros de proceso por lotes son utilizados como lanzaderas para colocar en memoria virus comunes. Para ello se copian a sí mismo como ficheros COM y se ejecutan. Aprovechan que las ordenes @ECHO OFF y REM traducidas a código máquina son "comodines" y no producen ningún efecto que altere el funcionamiento del virus.

Los virus del Mirc vienen a formar parte de la nueva generación Internet y demuestra que la Red abre nuevas formas de infección. Consiste en un script para el cliente de IRC Mirc (programa de charlas a traves de internet). Cuando alguien accede a un canal de IRC, donde se encuentre alguna persona infectada, recibe por DCC un archivo llamado "script.ini". Por defecto, el subdirectorio donde se descargan los ficheros es el mismo donde está instalado el programa, C:\MIRC. Esto causa que el "script.ini" original sea sobrescrito por el nuevo fichero maligno. El nuevo script permite a los autores, y a cual-

quier persona que conozca su funcionamiento, desde desconectar el usuario infectado del IRC hasta acceder a información sensible de su ordenador. Así, por ejemplo, pueden abrir un FTP en la máquina de la víctima, acceder al archivo de claves de Windows 95 o bajarse el "etc/passwd" en caso de que sea Linux.

El usuario de IRC tiene varias formas de protegerse. Como primera medida debe desactivar la opción AUTO GET que recoge los ficheros DCC de forma automática. De esta forma cada vez que intenten un envío por DCC el cliente le informará del fichero en cuestión, del nick que nos lo envía y, lo más importante, nos dará la opción a rechazarlo. Otra medida de protección consiste en cambiar el subdirectorio por defecto del DCC para evitar que sobreescriba el "script.ini".

Por último, no todos los virus son malignos aunque siempre se asocie el termino con destrucción. Podemos señalar que hay virus benignos y que, como ocurre con otras disciplinas del undergroung, todo depende de la utilidad que se le de a esta técnica. Con el termino "virus benigno" no nos referimos a aquellos que tienen como payload (efecto del virus cuando se activa) alguna pantalla graciosa y no destruyen datos. Nos referimos a que una buena utilización de las técnicas que emplean los virus pueden, aunque desgraciadamente no es lo habitual, reportarnos beneficios y ser sumamente útiles. Además también se pueden utilizar dichas técnicas para parchear sistemas a través de extensas redes LAN. El programa se infecta de ordenador a ordenador modificando parte de un programa que causa fallos en el sistema, y una vez solucionado el error se autodestruye.

Debido a la naturaleza de las páginas www de donde se encontró la información utilizada en este apartado de virus, éstas están poco tiempo disponibles en la Red. A la hora de revisar este artículo para su publicación definitiva ya no existía ninguna de las originales. Hemos encontrado nuevas páginas relacionadas con el tema, como cualquier lector con un poco de conocimiento de Internet podría encontrar, pero la naturaleza mucho más agresiva de éstas nos ha hecho pensar que sería mejor que no aparecieran en un artículo de este tipo.