

Introducción

Raro es el día que no escuchamos de uno u otro alguno de estos términos (y muchos otros). Con los años me he dado cuenta que la gran mayoría no sabe diferenciar entre unos y otros, de lo que realmente es un problema o de lo que simplemente es una mala práctica por parte del usuario. Así que lo primero que debería de quedar claro son el tipo de amenazas que existen a día de hoy (en su gran mayoría claro) y las defensas de las que contamos, si el problema es de nuestro sistema, si el problema es de la tecnología o si el problema es nuestro en último término.

Por otro lado leemos estadísticas que dicen que el 80% de los ordenadores están infectados con algún tipo de virus, de la necesidad o no de los antivirus, de lo seguro o inseguro que es un Sistema Operativo (OS) concreto... lo que sucede es que como digo la mayoría de las personas desconocen a qué tipo de amenazas estamos expuestos, cuales podemos evitar y cuáles no. Visto esto voy a intentar explicar con palabras simples los diferentes problemas de seguridad de hoy en día y de cómo evitarlos o solucionarlos. Tener en cuenta que es tan solo un "prefacio" a cuanto existe hoy en día a nuestro alrededor, pero al menos que sirva a modo de introducción.

Antes de entrar en materia, hay q tener en cuenta q los sistemas informáticos son por naturaleza inseguros. Esto es simple de comprenderlo. Ya sea un sistema operativo como Windows o Linux, una suite de ofimática como Microsoft office u Open Office, navegadores web... todo ellos no son más que programas creados por personas, y como tales, son imperfectos. A veces es el programador quien no tiene los conocimientos suficientes o simplemente no se da cuenta de un potencial agujero de seguridad. Muchas otras veces es una inseguridad innata de la misma tecnología (ya veremos ejemplos de esto). Y por supuesto, hay que tener en cuenta las dependencias de un software respecto a otro. Es decir, casi cualquier software hoy en día funciona bajo un sistema operativo concreto, luego en última instancia depende de este. Todo esto es crucial tenerlo en cuenta, sobre todo cuando veamos los exploits o los sistemas de protección de un OS frente a diferentes ataques.

La evolución histórica ha modificado algunos de estos conceptos con los años y se han acuñado nuevos términos. Es decir, esto puede ser válido para hoy, pero desde luego no para mañana.

Primero vamos a ver los diferentes "malware". Es decir, por malware nos referimos a un programa informático con fines maliciosos.

En segundo lugar dejaremos las amenazas más comunes y de las que somos más culpables y pasaremos a los verdaderos problemas de la seguridad.

Malware

Virus, Gusanos y Troyanos

Aunque a veces es complicado diferenciar entre diferentes conceptos, no todo es un virus. Quizás es cierto que los virus fueron el nacimiento de toda la inseguridad informática, pero en cambio, hoy en día sería el último de nuestros problemas. Erróneamente muchas veces usamos (yo incluido) el término "Virus" para referirnos a algún malware. Para muchos por desconocimiento, para otros para evitar explicar por ejemplo a un cliente que no le interesan detalles la diferencia entre un virus o un troyano.

Un **virus** sería el caso más simple de todos, es un programa malicioso el cual su única opción de infección a otros dispositivos es por medio de una interacción directa, es decir, el virus (el programa en sí) se copia por ejemplo a un disco, a una carpeta de red... y una vez en el medio que sea, necesita de nuevo la ejecución directa por parte del usuario descuidado (al menos la primera vez).

Su expansión depende de la inteligencia de los programadores para inventar métodos cada vez más sofisticados de replicación del virus. Por ejemplo una opción sería que el virus se copiase a sí mismo con un icono de documento de Word a todas las carpetas compartidas e incluso a todos los Pendrive insertados, creando en estos un autoarranque para que el virus sea ejecutado de forma automatizada al ser este insertado en otro PC, pero esto sería ya más próximo a lo que comprendemos como un Gusano.

La diferencia de un **Gusano** respecto a un Virus no es más que las capacidades de contagio que tenga cada uno. A diferencia de un Virus, un Gusano intenta replicarse constantemente por todas las formas posibles. Podemos decir que es un virus más activo, más contagioso. Por ejemplo como hemos dicho antes, buscando constantemente unidades de Pendrive insertadas, reenviándose a sí mismo a todas las direcciones de correo que sea capaz de encontrar en el equipo, reenviándose por programas de mensajería instantánea... cualquier sistema es válido.

Un **Troyano** hace referencia directa al famoso Caballo de Troya, del cual la mayoría conocerá la historia. El objetivo de un Troyano no suele ser tanto la infección, sino la creación de una backdoor (puerta trasera), es decir, crear en el PC víctima un acceso remoto. Realiza un proceso muy similar a lo que podríamos lograr con muchos programas de administración remota que existen (llamados RAT). A diferencia de estos, el troyano tiene una visión filosófica diferente: Engañar. Para mí un RAT se convertirá en troyano cuando este se camufla pasándose por ser un programa genuino o tiene capacidades de infección y replicación.

Pero por lo mismo que hemos comentado, un Virus/Gusano que disponga de capacidades de crear y manejar un acceso remoto, podríamos llamarlo también un troyano, dejando tan solo las definiciones de Virus/Gusano cuando el malware no tiene este tipo de capacidades. Es

decir, un troyano es en sí mismo un virus o gusano por naturaleza. En cambio, para otras personas un troyano por definición nunca tendrá capacidades de contagio, pero si existirán para estas personas virus/gusanos con capacidades de creación de puertas traseras.

Sin ser nombrados en el título y siendo algo de más actualidad, podríamos nombrar en este grupo también a los **RootKit**. Los Rootkit son un invento relativamente moderno. De un modo simple la idea del un RootKit es instalar un virus en el sistema a muy bajo nivel, de modo que sea capaz de tener control sobre todo el sistema. Por poner un ejemplo, pensar en un driver de sistema que sea malicioso. Hoy en día un PC con Windows tiene dos espacios diferentes, uno llamado modo usuario y otro llamado modo kernel. Por decirlo de un modo simple, en modo usuario trabajaría el usuario y en modo kernel el sistema. El código de los drivers por ejemplo se ejecuta en modo kernel, mientras que cualquier aplicación se ejecutaría en modo usuario. Esto es muy importante, dado que en modo kernel, se puede tener un control total del sistema. ¿Pero para que es útil esto? Los primeros Rootkit como virus lo tenían claro... para el sigilo. Se configuraba el sistema para ocultar de forma completa el virus, de modo que fuese imposible su detección. Imaginar un virus que de cara al usuario no está instalado, no se muestra en el registro, ni en el explorador de archivos, ni en ningún lado. ¿Cómo es esto posible? La única forma es pudiendo ejecutarse en modo kernel. Otras aplicaciones prácticas se centran en la intercepción de las rutinas de llamadas del sistema. Esto es fácil de comprenderlo, imaginar que cada vez que el OS tenga una llamada a ejecutar una subrutina cual sea o una interrupción, antes de ser llamada el rootkit la intercepta y hace que se ejecute su código malicioso. Es decir, el virus se podría ejecutar cada vez que simplemente se produce una interrupción en el sistema o cualquier tipo de rutina. La imaginación es el límite.

Por ello los rootkit entrañan un peligro enorme, y ello sin hablar de que sucedería si se instalase un rootkit en una firmware, como por ejemplo una BIOS (ya sea la BIOS del sistema, la BIOS de un lector de DVD, de la tarjeta de video...). Por suerte, es complicada la creación de un rootkit, y más aun su infección. Aun cuando obedece al grupo de virus, por así decirlo, y aun que el contagio se realiza por interacción directa del usuario, los OS de hoy en día están constituidos de tal modo que impiden el contagio de estos rootkit.

Ojo!! Tanto unos como otros requieren ¡¡SIEMPRE!! La interacción directa del usuario, al menos la primera vez. Existen técnicas que permiten que esta interacción sea transparente, como es el caso de la auto-ejecución de los Pendrive o DVDs/CDs, pero a fin de cuenta depende también de la interacción del usuario hoy en día, y como tal puede ser evitada dicha infección. Esto quiere decir que todo lo que podamos meter en este saco, el responsable último es el propio usuario descuidado que ejecuta (ejecutar = interacción directa del usuario) el malware. A veces pueden ser simples engaños por correo, otras veces un software descargado de algún lugar no legítimo...

Pese lo que piensa la mayoría, un malware no se suele crear con la intención de dañar un PC. Hoy por hoy de forma mayoritaria, lo que se desea de un buen virus/gusano es una alta capacidad de infección, una alta capacidad de sigilo y la creación de Backdoors (puertas traseras) en el equipo infectado. Es decir... un malware programado correctamente debería de pasar inadvertido para el usuario, cuanto menos sepa el usuario de su existencia, menos

posibilidad habrá de que sea eliminado. En cambio si apreciamos un comportamiento anómalo en nuestro equipo, lo primero que haremos será una busca y captura. Esto ha cambiado con los años. Antiguamente los virus se creaban principalmente para molestar, algunos de ellos hay que decir que eran muy ingeniosos. Desde virus que eliminaban nuestros archivos, virus que hacían que las letras de nuestra pantalla se corriesen... a día de hoy es difícil encontrar un comportamiento de este tipo, a no ser que se realice por una cuestión de querer dar publicidad a algo (por ejemplo un virus que por las mañanas al encender el PC muestre una foto de Ramoncín diciendo: "La piratería es positiva").

El objetivo último de este tipo de malware es diverso. Antiguamente, como hemos dicho, era una cuestión más de fama o de reivindicación que cualquier otra cosa. Es decir, existía como una competición por ver quién era capaz de crear el virus más ingenioso, más gracioso. Otros en cambio aseguran que el primer virus que existió fue realmente algo casual, un programa que no tenía como fin producir un comportamiento anómalo-dañino, pero que simplemente un error de programación causó dicho comportamiento. Con el tiempo los virus se convirtieron en un auténtica mina de oro para las compañías de software de seguridad, tal es así que siempre se les ha atribuido la creación de muchos virus a estas mismas compañías: Suelto el virus y vendo la vacuna. Los más reivindicadores encontraron en los virus una forma de queja a la sociedad, por ejemplo infectando miles de ordenadores para mostrar un mensaje político, religioso o de cualquier otra índole.

Actualmente no obstante, aunque todo lo comentado aun existe, el objetivo suele ser mucho más dañino desde mi punto de vista. Nos encontramos en una sociedad en la que lo más importante es la información. Quien tiene la información tiene el poder. Así, la mayoría de todos los virus y gusanos que podemos encontrar hoy en día, tienen como objetivo último uno de estos dos: El primero el robo de información, ya sea con la creación de backdoor u otras técnicas que comentaremos más adelante. El segundo, la creación de redes de ordenadores Zombi para el envío de Spam o ataques DDoS, de lo cual también hablaremos más adelante.

Adware y Spyware

Por **adware** entendemos un software normalmente legítimo (es decir, no ha sido modificado) sobre cualquier cosa, sin ningún tipo de funciones de replicación/infección que muestra de forma indiscriminada publicidad no solicitada sobre algo. Dependiendo de la implementación de este tipo de técnicas, pueden ser simples zonas del programa en las que se inserta un anuncio o técnicas mucho más agresivas, como por ejemplo la apertura de ventanas emergentes redirigiendo a una web concreta, aunque de nuevo son solo ejemplos. Ejemplos de adware tenemos muchos, por ejemplo el viejo Kazaa o incluso el famoso Daemon Tools, aunque este último te da la opción de no instalar el adware.

Como todos los términos que vamos viendo, el problema es que todos se pueden cruzan con todos en algún momento y no podemos hacer una definición exacta, dado que muchas veces

es más la interpretación del experto de seguridad que sea el que para él es un malware de una clase u de otra. Con los adware, por ejemplo, ¿Donde estaría el umbral entre un programa shareware y un programa Adware? Recordemos que un programa shareware es un programa normalmente de pago pero que se puede usar de forma gratuita con ciertas limitaciones. Es más... personalmente pienso que el software Adware fue la evolución de los programas Shareware en los que esas limitaciones se sustituían por publicidad de cualquier tipo. Pero es solo opinión personal.

Por otro lado, está claro que si le preguntamos al creador del software nos responderá que esa publicidad no es más que un aporte económico para sufragar X gastos, y que a fin de cuenta el software es gratuito. Yo ante esto replicaría que sí, es cierto, que lo puedo comprender o no comprenderlo, pero que su motivo (el que sea) no deja fuera el hecho de que su software sea un Adware.

Por **Spyware** entendemos cualquier software que recopila información de nuestro PC sin nuestro consentimiento. Este tipo de software tiene una función muy diferente al adware, aunque muchas veces camina de la mano. No se debe de confundir un software Spyware con un Keylogger (ya se hablará de esto más adelante). Con el nacimiento del Adware, en algún momento a alguien se le ocurrió hacer una publicidad dirigida... ¿pero cómo puede hacerse esto? -> Recopilando datos.

No todos los Adware son Spyware ni mucho menos, aunque la mayoría de los Spyware si suelen ser también Adware. Esta información podría ser desde las web visitadas, el uso del PC, estadísticas... lo normal es que dicha información sea enviada a posteriori a algún servidor con fines normalmente no muy agradables para nosotros. Esta información se puede usar para saber que publicidad mostrarnos por ejemplo en cada momento, o para hacer un estudio de mercado antes de lanzar un producto...

OJO!! El peligro es evidente, todo esto se hace sin nuestra autorización expresa!! No podemos hablar de un patrón común, cada software Spyware es diferente, pero creo que queda claro de lo que estamos hablando.

Antes de acabar, decir que muchos de estos Adware tienen capacidades propias para descargar un Spyware desde la web sin que nos sea comunicado para poder llevar a cabo su labor, o incluso un Spyware/Adware podría descargar e instalar un virus/gusano/troyano... Como vemos, todos los malware están íntimamente relacionados.

Seguridad Informática

Anteriormente hemos visto más que nada una introducción para poder explicar los verdaderos problemas de seguridad. No se trata ya de que puedan robarnos información (al menos no demasiado importante para un usuario doméstico) sino lo que entrañan los problemas de seguridad en un sistema informático.

Hay una diferencia fundamental entre los problemas comentados con anterioridad y lo que vamos a tratar ahora. El culpable último de todo el malware explicado con anterioridad es el usuario mismo. Creo firmemente que cualquier persona que navegue por internet, use correo electrónico, redes sociales... tendría que tener una noción mínima de lo que está haciendo, del uso o mal uso que está haciendo. ¿Es algo paradójico verdad? Un niño de 16 años sabe perfectamente manejar Tuenti, enviar correos, fotos... y en cambio, ¿no se preocupa de conocer unas pautas mínimas para evitar tener el PC lleno de malware? No es una cuestión de torpeza, sino de educación.

El usuario como primera causa del problema

Antes de que muchos puedan pensar: "No todos nacemos sabiendo", yo les pregunto: "¿Alguna vez te has preocupado de conocer unas pautas mínimas para evitar los problemas?". Yo no he nacido sabiendo, ni mis hermanos, familia, amigos... y en algún momento es posible que alguno haya metido la pata y haber tenido un problema con algún malware. No sucede nada, sucede cuando el episodio se repite por no poner los medios necesarios, y con medios no hablo de un Antivirus (AV), hay técnicas mucho más eficientes, fáciles y casi casi imposible de fallar.

Un virus, gusano... o cualquier variante de malware es un problema de seguridad. Si es un virus molesto no dejará de ser algo.... molesto, pero si es un troyano sí que vemos de forma más clara el problema de la seguridad, y si pensamos en un gusano que se dedica a enviar Spam... Y es el usuario quien mete la pata en su 90%. Vamos a aprovechar y explicar unas pautas básicas, que seguro que todo el mundo es capaz de seguir, realizar... y que no cuestan NINGUN TRABAJO!! Y pasados unos días, se convierte simplemente en algo mecánico.

Para explicar estas pautas, primero vamos a pensar en el ciclo de vida de un virus cualquiera, llamémosle Virus Perico. Hemos visto a groso modo que pasará cuando el Virus Pepito esté en el sistema, y sabemos que nos borrará toda nuestra música cada 10 días (por ejemplo). Eso es lo que sabemos de "Perico", pues vamos a preguntarnos algunas cosas más, que son triviales!!, pero es cierto que normalmente uno no se las pregunta:

1º. Si Perico está en nuestro sistema... ¿Cómo ha llegado ahí?

2º. Si Perico hace lo que hace, es porque Perico (como cualquier programa) se está ejecutando en nuestro sistema, no deja de ser un programa!! (No aplicable a los Rootkit).

3º. Si Perico no es más que un programa, se tiene que ejecutar siempre, porque si no se ejecutase siempre los efectos serían momentáneos y ya está. Ojo!! Que Perico no elimine la música todos los días no implica que Perico no esté ejecutándose siempre, solo que está programado para eliminar la música solo cada X días.

4º. Comprendiendo el punto 2 y el punto 3, podemos llegar a la conclusión de que si Perico no ha modificado archivos de sistema o los ha dañado de forma irreversible, igual que vino, puede irse, como programa que es podríamos finalizarlo y eliminarlo del sistema de forma completa.

Simplemente preguntándose estas 4 cosas, el resto puede desglosarse de ello:

Prevención: ¿Como ha llegado ahí? -Respuesta: La hemos cagado.

Un malware es un programa, ese programa, al menos la primera vez, debe de ser ejecutado por una mano humana. Quizás fuese un documento infectado, quizás un archivo que no era lo que parecía... pero el usuario fue quien apretó el botón. ¿He dicho documento/foto/imagen... infectado? Incluso esa opción sería muy rara y entraría para mí en exploits (se tratará después).

El problema es que el 80% de las personas son inconscientes y no se preocupan por saber que están ejecutando. Para aquellos que tengan un nivel más bajo sobre lo que estamos hablando voy a explicar rápidamente que es ejecutar algo. Podríamos diferenciar dos tipos de información que podemos encontrar en un PC, por un lado datos y por otro lo que sería el código. Así por ejemplo una foto o un documento de texto podríamos clasificarlos como datos y un archivo DLL o un archivo EXE serían los programas en sí, es decir, el código. Pues bien, el sistema conoce bien esta diferencia, de tal forma que el sistema reconoce por un lado los archivos que podríamos llamar "Ejecutables" y los archivos que podríamos llamar como "Asociados". Un archivo ejecutable tiene significado propio, es un programa en sí mismo, mientras que un archivo de tipo asociado tiene una función asociada a un ejecutable dado. Esto es a grosso modo.

Bueno, continuando, cuando ejecutamos una foto, lo que hace nuestro sistema es ejecutar el programa asociado a dicho tipo de archivo, en este caso una foto por defecto Windows abrirá el visor de fotografías. El programa sería el ejecutable, el visor. El archivo asociado sería la foto. La foto llama al visor. (Nota: Ojo!! esto es cierto siempre y cuando nuestro sistema no haya sido manipulado para realizar una asociación maligna, por ejemplo que al ejecutar una foto en vez del visor se ejecute el virus Perico, pero para ello hace falta la intervención de otro virus (o el mismo Perico) para que haya modificado previamente el sistema).

Con lo que acabo de decir ahora y con lo anterior, podemos decir que en un sistema limpio, la ejecución de un archivo de los que hemos llamado "asociados" no puede acarrear ningún problema para el usuario, ningún peligro (exploits a parte, luego veremos esto).

El peligro por lo tanto corresponde a los archivos que son programas, los que hemos categorizado como "Ejecutables". Nos hemos preguntado cómo llega Perico a nuestro sistema... pues bien, en algún momento el archivo estaba delante de nosotros y lo ejecutamos por algún motivo. ¿Pero realmente somos tan descuidados? No, no siempre, pero nos intentan engañar para creer que lo que tenemos delante a lo mejor no es un "ejecutable" sino a lo mejor una foto, o es un amigo quien nos dice que le demos, o creemos que es un programa que estamos buscando... las causas más comunes son:

-Correo Electrónico

Un Adjunto llega al correo, el correo reza que lo abramos por cualquier motivo. Aquí empieza el engaño, como hacer para que el usuario pique y descargue el ejecutable en su equipo y lo ejecute. ¿Un ejemplo? Yo, Theliel, envío un correo a toda mi familia en el que aparenta que adjunto unas fotos. Al virus le pongo nombre de foto de cámara, algo como "DSC02457" y lo cuelgo en mi servidor: "<http://theliel.es/DSC02457.exe>" y el icono de una foto. Para engañar a mi familia les digo que vean las fotos de mi último viaje, y le pongo una sucesión de supuestos enlaces a las imágenes que ellos ven como: DSC02457001.jpg, DSC02457002.jpg... Pero que en realidad son tan solo vínculos a mi virus. Mi hermano, que al ser un correo mío y dice ser unas fotos, al ser descuidado le da, descargar mi virus y lo pone en el escritorio. Una vez en el escritorio lo que ve es tan solo un archivo llamado "DSC02457", dado que las extensiones de archivos por defecto están ocultas, y ve que tiene el icono de foto. Lo ejecuta... y Perico acaba de entrar en sus sistemas. No se dio cuenta de que el nombre más extensión era "DSC02457.exe", el icono es indiferente. Tampoco se dio cuenta de que en realidad no eran fotos adjuntas en el correo, sino que en el correo lo único q había eran enlaces a mi virus, yo al vinculo le puedo poner el nombre que quiera!! A él le sonó la foto y metió la pata.

-Programas de Mensajería

Igual a lo explicado con anterioridad, pero los gusanos se valen del Messenger para enviar a todos los contactos de este un mensaje que puede decir por ejemplo algo como: "Te dejo las fotos del otro día -> <http://theliel.es/DSC02457.exe>". Esto se perfecciona por ejemplo escribiendo una ruta muy grande de modo que el .exe no aparezca en el mismo mensaje, pero si el enlace. El usuario descuidado le da, lo ejecuta y Perico en su sistema.

-Descargas de programas

Más engaños. Imaginar que estamos buscando por la red el compresor 7-zip. ¿Cuál sería el comportamiento de un usuario descuidado? Buscar en Google, ir a la primera página que vea y lo descargue de allí. Lo que no sabe es que la página que ha llegado es maligna, y en realidad no se está descargando 7-zip, sino Perico. Después cuando lo va a instalar... Perico para dentro. Esto es muy común en los supuestos Keygen que existen en la red y otros programas más... underground. Entramos en una web que dicen que tienen el Keygen del último Nero que tenemos instalado. Lo descargamos, lo instalamos y... vaya... no era un Keygen... ¡¡Era Perico!! O incluso a lo mejor era un Keygen, pero un Keygen con Perico incluido!!

-Redes P2P

Existe la creencia de que las redes P2P son un criadero de virus. Como hemos dicho la inconsciencia es del usuario. Amigo, si te estás descargando una película, una película no es un archivo Zip, ni un archivo EXE, ni... la picaresca está en todos lados, y todo vale para engañar al usuario poco precavido.

-Pendrive

Más recientemente la lacra de los Pendrive... Conectamos el pen a nuestro PC, abrimos el Pen y Perico para dentro. Sí, tan solo como abrirlo o incluso insertarlo (dependiendo del OS). ¿Quién es el culpable? Nosotros, por partida doble. Primero por fiarnos de un Pendrive que no es nuestro, y segundo por no tener la precaución al abrirlo por si estaba infectado.

Vale, ya tenemos las vías de infección... como evitarlas? Ya lo he dicho, unas pautas simples, mínimas, y fuera el 99% de todo el malware que existe:

a) Conocer las extensiones potencialmente peligrosas, un posit en el monitor durante unos días es suficiente, no son muchas, las hemos visto miles de veces. Si nos llega por correo, Messenger, Pendrive, por red interna, redes sociales... aquí está el peligro:

.exe, .bat, .com, .pif, .cmd, .scr, .vbs -> Son potencialmente peligrosas.

A menos que estemos seguros al 100% de que dicho archivo es legítimo de nuestro contacto y que sabemos que es, no lo abrimos, es así de simple. Si mi hermano me manda un archivo .exe, o sé lo que es perfectamente o no lo abro. Simple. Con esto nos quitamos de un plumazo el 50% del malware.

b) Con a) nos quitamos el problema del correo, mensajería instantánea... pero ¿y si es un programa que nos hemos descargado? SIEMPRE!! Acudir a las web oficiales. Si queremos descargar "7-zip" no vayamos a Softonic, vamos a la web oficial!! www.7-zip.org.

c) Si hablamos de un programa más underground, un Keygen... usar un AV. Ojo!! No, no soy partidario de usar AV en los equipos, y no estoy hablando de usar el AV de nuestro equipo. Si es un programa underground lo más normal es que sea tan solo un archivo .exe y listo. Lo más efectivo es acudir a la web de [virustotal](http://www.virustotal.com), y subir el archivo, en unos segundos será escaneado no por un AV, ni por dos... sino por la mayoría de los AV del mercado, y rápidamente nos mostrará un reporte sobre el análisis. Está claro que si da 30 de 50 como virus, es un virus. Si da 1 de 40, lo normal es que no sea un virus. Simple.

d) Muchos archivos están en .exe para descomprimirse ellos mismos, pero claro... como saber si es real o no real el paquete? Fácil, si es un programa genuino lo hemos obtenido de la web oficial, no hay nada que verificar. Si es un programa más raro, lo comprobamos con 7-zip si es real un paquete o si es un exe con icono de paquete.

e) Para los Pendrive, jamás ir a mi PC (o Mi equipo) y darle dos veces a la unidad en un Pendrive que desconfiemos. Siempre, siempre, botón derecho en la unidad, explorar. De este modo, sea como sea la configuración de nuestro sistema, no se auto ejecutará nada. Que hemos llevado nuestro Pen a otro PC y lo volvemos a conectar a nuestro PC? Botón derecho, explorar. Una vez la unidad abierta, verificaremos que no exista el archivo "autorun.inf", ni visible ni oculto. Si el explorador de archivos nos dice que hay archivos ocultos, más razón para darle a visualizar archivos ocultos. Si no hay "autorun.inf" aun cuando podría tener un virus, este sería de algún tipo comentado en a-d, es decir, no ejecutar un ejecutable que no conocemos, o usar la web de virustotal... etc. etc., pero si no hay "autorun.inf" (ni oculto/de sistema) el pen no auto ejecuta nada. No implica que dicho archivo sea malo, y puede existir sin necesidad de ser un virus, pero ante la duda...

No hay más. No ejecutar lo que no sepamos, cuidado con los ejecutables, y ante la duda virustotal. Con esto el 99% del malware desaparece y sin la necesidad de tener en nuestro equipo un AV, que tan solo sirven para restarnos rendimiento.

La hemos cagado. ¿Ahora qué?

Vale, la hemos liado, no hemos tenido cuidado y estamos seguros o creemos que Perico está danzando por sus anchas en nuestro equipo. ¿Ahora qué podemos hacer?

Ante esto no hay un procedimiento concreto, dado que cada virus es diferente y las medidas que tiene este de "protección" son diferentes. Pero si recordamos los puntos que comentamos, nos podemos hacer una idea clara:

-Finalizar Perico

Hemos dicho que no era más que un programa, así que antes de nada lo que hay que hacer es cerrarlo, finalizarlo, sea como sea. Esto puede ser desde muy fácil a muy jodido, porque si es un virus inteligente, tendrá medidas para protegerse de que lo finalicen. ¿Cómo es esto posible? Normalmente se hace con dos copias del mismo virus en ejecución constante, en la que una monitoriza constantemente la ejecución de la otra. Si en cualquier momento una es detenida, la otra ejecuta una nueva copia del virus.

El caso más simple es un simple proceso en ejecución, llamémosle Perico.exe. Tan solo tendríamos que acudir al Administrador de Tareas (Ctrl+Shift+Esc), buscar Perico.exe en la lista de procesos y finalizarlo.

El caso algo más complicado sería igual que el anterior, pero cuando el proceso no es tan claro, no tiene un nombre tan significativo. En este caso el proceso sería el mismo, pero con experiencia necesaria, un poco de Google y un poco de sentido común, para saber qué proceso es el dañino. Buscar en Google el nombre de un proceso es algo simple, y con el tiempo sabemos diferenciar claramente lo que es un proceso legítimo de uno que no lo es. Otra

suposición sería pensar que el proceso en cuestión está siendo ejecutado por el usuario, luego en "nombre de usuario" lo normal es que esté el nombre de sesión del usuario, lo que limita mucho la lista de procesos.

El caso medianamente complicado sería cuando el proceso está duplicado, ya sea con dos claras instancias en el administrador de procesos o ya sea duplicado pero oculto para este. En este caso no nos vale finalizar el proceso dado que el otro arrancará en su lugar. Lo más cómodo para esto es arrancar en modo de prueba de fallos. En este modo el sistema no carga nada que no sea imprescindible. En dicho modo directamente el virus Perico no se ejecutaría, con lo que problema resuelto. Esto también se podría usar para los casos anteriores.

El caso más complicado es cuando el proceso está anclado a algún proceso legítimo. La opción más sencilla es por ejemplo hacer que se ejecute junto con "explorer". Explorer es el escritorio, por así decirlo. De modo que cada vez que se arranca el equipo, este se ejecuta igualmente. Pero no solo esto, sino que el virus reside en el mismo proceso "Explorer", luego para finalizarlo deberíamos de finalizar este proceso. El problema que tiene esto, es que al finalizarlo deberemos de ejecutar lo que deseemos a mano, mientras que eliminamos la infección.

-Impedir que Perico vuelva a ejecutarse

Suponemos por tanto que Perico ya no se está ejecutando, ya sea por un arranque en modo de prueba de fallos o ya sea por algún proceso anteriormente comentado. Ahora queda eliminar los sistemas de auto inicio que puede tener Perico.

Para ello tan solo debemos de conocer los sistemas que cuenta nuestro OS para arrancar un programa al inicio. Esto hay dos formas de hacerlo, confiando en algún programa o confiando en nosotros mismos. Para el usuario más experimentado lo mejor es siempre hacer lo a la vieja, a mano. Para el usuario más de casa le recomiendo usar mejor el programa Autoruns. Este viejo programa pertenecía a una empresa que fue absorbida por MS y a día de hoy todas las herramientas de esta empresa se encuentran en continuo desarrollo. Esta herramienta, fundamental, la podéis obtener de aquí:

[Autoruns](#)

La idea es clara, eliminar de los sitios de inicio cualquier referencia a Perico. Si se usa Autoruns, las pestañas a buscar serían preferentemente Logon y Winlogon. Autoruns permite comprobar el archivo en internet o mirar la firma del proceso o si la firma es válida. Si la firma de un proceso es de Microsoft y esta es validada, está claro que el proceso será genuino. Si la firma es de alguna empresa rara o ni siquiera está firmado... dará más que pensar. Para quienes prefieran el método manual, los lugares a mirar son fundamentalmente:

Menú Inicio -> Inicio Todo lo que exista en dicha carpeta se ejecuta al arrancar el sistema
Registro de Windows ->

Clave HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

Clave HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Winlogon

Clave HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Clave HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
Clave HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run
(Para procesos de 32 bits en OS de 64 bits)
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon
(Para procesos de 32 bits en OS de 64 bits)

Aunque pueden existir otros lugares, como las asociaciones de archivos, servicios, tareas programadas... estamos hablando de forma general. Esto no significa que no pueda esconderse el virus en otro lugar, tan solo que estos son los lugares más claros y primeros a buscar.

-Eliminar Perico

Vale, no se está ejecutando, lo hemos suprimido de todos los lugares de inicio... pero Perico como tal aun existe, aunque ahora mismo en este momento no debería de hacernos ningún daño, pero no quita que esté en algún lado. Simplemente, si hemos tenido la precaución a la hora de buscar las ubicaciones de inicio de mirar la ubicación real del archivo, tan solo debemos acudir a dicho lugar y eliminar del todo el malware.

Para acabar con el papel importantísimo del usuario, recordar que cualquier tipo de malware puede ser una puerta adicional para comprometer todo nuestro sistema. Un malware puede ser desde un virus meramente molesto a ser un gusano que está replicándose a todas las unidades de red a ser un malware con funciones de backdoor que están creando una red de PCs Zombis para algún propósito maligno. Sin añadir claro está el robo de información, el mal funcionamiento del equipo...

Ingeniería Social

El usuario está expuesto siempre a todo tipo de amenazas, sean responsabilidad de él o no lo sean. Internet es una Selva, algunos sabemos cómo vadearla con machetes y rifles, y otros están artos de que les piquen los mosquitos y tengan que subirse a los árboles para sentirse seguros.

El termino Ingeniería Social no sé sinceramente si nació a raíz del problema que vamos a tratar o ya existía anteriormente. Por Ingeniería Social entendemos como cualquier técnica que pueda emplearse para obtener el máximo de información sobre alguien de una forma directa. Esto es posible a día de hoy de muchas formas diferentes:

- Investigar los perfiles de las redes sociales de dicha persona.
- Hacer preguntas por programas de mensajería instantánea-
- Enviar correos "trampa" para provocar la contestación de estas personas.
- Hacerse amigo de un amigo o de la misma persona con el fin de obtener datos...
- Creación de perfiles y otros en redes sociales haciéndose pasar por la víctima

¿Para qué sirve todo esto? Cuanta más información se tenga de un usuario más sencillo resultará poder violar la seguridad de su sistema, es así de simple. Tan solo hay que ver las contraseñas de acceso o los nombres de usuarios. ¿Sabéis que más de un 50% de contraseñas pueden ser fácilmente encontradas por simple adivinación, conociendo un poco a dicha persona? Pero no solo hablo de contraseñas, sino de hábitos, costumbres... hoy en día controlar información equivale a tener poder. Con información sobre alguien puede desde no servirte para nada como tenerla toda su vida bajo tus pies

El problema nace de lo que yo denomino un movimiento social extremo que estamos viviendo estos años. Con Internet en una mano, la sociedad cada vez más abierta a todo... el paradigma a seguir es el todo vale. Ahora parece que todo el mundo tiene que ser socialmente abierto, socialmente respetable (al menos en apariencia), socialmente social. Si él hace esto yo no puedo ser menos... Toda esta corriente produce evidentemente se quiera o no a priori una especie de... "confianza" por así decirlo, a fin de cuentas si todos están en el mismo caso, ¿Qué importa? Un ejemplo tonto a esto, si dos amigos se van de putas, entre ellos dos no tienen problema alguno nunca!! Los dos lo hacen, ninguno de los dos lo reconocerá con otra persona pero entre ellos hasta frivolarán.

¿Que tiene que ver todo esto? Todo. Internet es ahora mismo el nexo de unión, el punto de encuentro o como queráis llamarlo de este mismo movimiento social. La tecnología está de moda, o tienes lo último o estás a la última o te miran mal. Ahora no puedes dejar de tener Messenger, Tuenti, Facebook, Twitter... porque si no tienes eres un bicho raro que no tiene amigos. La sociedad, sobre todo la juventud actual, ha cambiado radicalmente sus hábitos, y como último "responsable" para bien y para mal, Internet. Todo se maneja y se cuece a través de él. El conocer a personas en un lugar público pasa a ser el conocer a personas a través de las redes sociales. El irse a la biblioteca a buscar información se convierte en buscar en Google o en la Wikipedia...

No estoy diciendo que esto sea malo, como en todo hay cosas positivas y negativas. Hablo del peligro que encarna esto. Una chica de 14 años tiene Tuenti. Le da igual que la agreguen 10 que 20 que 100 personas, las acepta a todas. En su perfil tiene puesto desde día de nacimiento, lugar de residencia, hábitos, fotos, amigos... ¿estamos locos? Si hay buena voluntad perfecto, pero si esa persona tan solo quiere llevarla a la cama? Pornografía? Pederastia? Abusos? Claro, que soy exagerado... puede... pero creo que la mayoría conocemos a alguien que ha tenido problema por culpa de las redes sociales, personalmente conozco unos cuantos.

El problema no es la red social, el problema es el uso que se le da, el uso de la Ingeniería Social. Se ha llegado a un punto en que es mucho más fácil lograr que la persona te de la contraseña de su cuenta a molestarte siquiera a lograrla por otras técnicas más "bonitas". Si ir más lejos, hace unos 3 días, un usuario de cierto lugar, después de hablar con él unas palabras en plan agradable y tal, me estaba preguntando que si me importaba darle mi nombres de usuario. Señores... un poquito de vista.

No escribo esto para decir que hay q ser la persona más desconfiada del mundo, ni tampoco soy el padre de cada uno de mis lectores y tengo que decirle lo que debe de hacer, lo que no o que información publicar en sus perfiles o a quien añadir o a quién no. Aquellos en cambio que

hayan padecido lo que estoy comentando me comprenderán perfectamente, los que creen que esto son exageraciones... espero simplemente que jamás tengan problema alguno... pero ojo!! El problema simplemente aparecerá cuando cualquier persona quiera meter las narices. Una vez que las quiera meter... habrá problemas.

Una frase decía que "A quien revelas tus secretos, confías tu voluntad".

Hay que tener mucho cuidado a quien das acceso a tus datos. ¿Realmente un perfil de Tuenti o Facebook es normal que tenga 100-1000 amigos? No hacerme reír.

Spam, Pishing, Scam y Correos Fwd

Por supuesto tenía que dedicar unas palabras al buen amigo del Spam, al siempre peligroso Pishing, el increíble Scam y al peor dolor de muelas que ha conocido el hombre: los correos Fwd. Con esto cerraremos el tema.

-Spam

Spam (en el correo electrónico) no es más que un nombre genérico para designar cualquier correo electrónico no deseado que nos llega, tan simple como eso. Es una pregunta muy común que me hacen: ¿Qué finalidad tiene el Spam? ¿De dónde procede?

La finalidad del Spam es la publicidad o reivindicación de algo generalmente. Imaginar que todos los días al despertarnos, nuestro buzón estuviese lleno de cuartillas de miles de restaurantes, tiendas, fármacos... solo que no se tiene nadie que molestar al buzoneo, solo es cuestión de tener un servidor de correos y nada más. Como dato curioso... el 80% de todos los correos son Spam.

¿Quién los envía? Bueno, existen muchos medios. De una forma activa y directa, podríamos imaginar a un trabajador que se pasa el día buscando por la red cualquier dirección de correo electrónico que añadir al sistema para que este reciba el Spam. Pero es evidente que esto se logra de una forma mucho más eficiente y barata de forma automatizada. Vamos a poner algunas formas:

-> Crawlers: Sistemas automatizados que escanean Internet para obtener algo en concreto, en este caso direcciones email.

-> Suscripciones: Solicitar de forma directa el correo para algo, y estas empresas venden las bases de datos a terceros.

-> A través de los Correos Fwd, los cuales los veremos más adelante.

-> Mediante gusanos, una vez infectado el equipo, este utilizará al equipo para buscar direcciones email y usar el mismo equipo como emisor de Spam.

Una vez que se tienen las direcciones el envío se puede llevar a cabo desde servidores de correo electrónico particulares o mejor aun... infectando con algún tipo de malware un usuario, y que sea este quien envíe el Spam desde su PC. Otra forma común es la intervención de cuentas de correo electrónico, y usarlas masivamente para el envío de Spam, como resultado final del cese de dicha cuenta cuando nuestro proveedor es notificado de que la cuenta se está usando para dicho cometido.

El Spam es odioso, molesto... Combatir el Spam pasa por algunas pautas:

-> No exponer nuestro correo jamás en ningún sitio, esto es además una medida de seguridad. He visto perfiles de Tuenti y Facebook en los que tienen puesto sin pudor alguno su correo electrónico.

-> Evitar todo tipo de suscripciones de las cuales no estemos seguros.

-> Existen ciertos lugares que obligan incluso la suscripción a ciertos lugares para poder realizar el registro... evitar estos lugares. El ejemplo más común es Canalmail.com (creo que se llamaba)

-> Cuando se envía un correo electrónico, y esto es muy importante, JAMAS hacer que se convierta en un Fwd

-Phishing y Scam

El Phishing es verdaderamente algo muy peligroso. El Phishing es la esencia más pura de la suplantación, normalmente de una entidad, es una práctica completamente ilegal y con la que hay que tener mucho mucho cuidado. Scam es un término que suele ir cogido de su mano, entiendo como Scam digamos la técnica para poner en práctica un Phishing.

La idea del Phishing es normalmente valerse del envío de correos en los cuales las direcciones han sufrido Spoofing (han sido manipuladas para que aparenten ser correos genuinos de entidades genuinas). Esto es posible dada la naturaleza en sí misma neutral de Internet, se explicará después. La idea como digo es falsificar un correo electrónico. Imaginar que cualquier persona que tenga una cuenta de PayPal recibe un correo electrónico que procede de "Support@PayPal.com" con el asunto de: Cambio de políticas de privacidad. El usuario interesado claramente por su cuenta lo lee, a fin de cuentas no es la primera vez que le llega un correo de PayPal informándole de algo. El usuario lo lee y el correo le solicita que dado que se han realizado cambios en las condiciones de licencia, debe de acceder a PayPal a través del enlace de abajo para actualizar su perfil. Al finalizar aparece efectivamente el enlace a PayPal, el usuario ve algo así:

<https://www.paypal.com/es/>

o por ejemplo esta otra variante más sigilosa:

<https://www.paypal.com/es/>

Podríamos hacer una lista de posibilidades. Lo que está claro es que el usuario descuidado comprobaría a simple vista que es la página de PayPal, con lo que hasta ahí no vería nada de peculiar.

Al acceder a dicho enlace (el mío es tan solo un ejemplo que lleva a mi propio blog), se encontraría con la página oficial de PayPal... aunque solo en apariencia. Clonar un sitio web es "sencillo". Dicha Página web sería el Scam. La idea es que el usuario sin problema alguno registraría sus datos e iniciaría sesión en PayPal para realizar lo solicitado. En este momento el creador del Scam se haría con los datos del usuario, capturando su nombre y contraseña. Por último para que el usuario no notase ningún tipo de anomalía se le podría redirigir ahora sí a la web legítima de PayPal.

Esto que aquí explico es algo por desgracia demasiado común de encontrar y de hacer, y es solo la punta del iceberg, existen multitud de técnicas para que pase todo mucho más desapercibido.

-Correos Fwd

Este es uno de mis temas favoritos. Antes de hablar sobre ello, una pregunta:

¿Alguien me puede decir qué sentido tiene reenviar a todos tus contactos un correo que dice que una conjunción planetaria va a acabar con el mundo conocido, y que es importante que todos tus conocidos y amigos lo sepan?

Señores, el 99% de todos los correos que dicen "Reenvíame" son Hoax, bulos, leyendas urbanas, chorradas, estupideces... el único fin que tienen es recolectar direcciones de correo electrónico, y es el método más efectivo que tienen los Spammers para cazar direcciones. A esto es a lo que llamo yo los Correos Fwd (Fwd, Forward, Reenviar), y son un peligro sin límites. En mi época oscura, os diré que era una mina de oro.

Vamos a ver, no quiero parecer insensible. Repito, el 99% de ese tipo de correos son chorradas, y el problema es que no se sabe por qué, el 80% de las personas se los creen!! Cuando no es un correo de que Hotmail va a cerrar, cuando es otro de que Marte va a estar tan grande como la Luna, cuando no es un niño que se está muriendo y hace falta reenviar el correo porque tienen el mismo valor que las firmas... cualquier tema que pueda suscitar el interés del lector es bueno. Por ni hablar ya de los correos en cadena tipo: Reenvíalo o te sucederá algo y en fin... la lista es interminable.

Existe de entre todos los Fwd uno que es mi favorito, es la navaja suiza de todo usuario malintencionado: Los correos de 200 preguntas sobre ti para que tus amigos te conozcan, aunque este tipo de correos suele ir dirigido tan solo a un público más joven. Vamos a ver... si son tus amigos, no necesitan conocerte, ya saben quién eres y lo que significas par ellos. Segundo, os podéis hacer una idea de la cantidad de información que se está dando sin control alguno? Me parece increíble.

Y lo peor de todo, lo peor que se puede hacer con estos correos fwd es lograr que los Spammers logren sus resultados. Señores, si no tenéis remedio de reenviar un correo, que me parece

genial que se haga!! Por dios... eliminar las cabeceras de los correos, no dejéis 5 páginas de nombres y apellidos con sus direcciones de correo. Cada vez que se reenvía un correo, se suele incorporar la cabecera de este. Si este fue enviado a 100 personas, la cabecera incluirá estas 100 personas. Si quien lo reenvía lo reenvía a otros 100 contactos, ya son 200 contactos visibles!! En serio, no cuesta nada editar el correo, eliminar las 20 páginas de cabeceras y direcciones y dejar tan solo el cuerpo del mensaje limpio. Y para terminar de hacer las cosas en condiciones, envía el correo a 100, a 200 contactos... los que sean!! Pero añádelos como copias ocultas, que nadie de quien recibe el correo tiene por qué saber las direcciones de correos a quienes les has enviado el mismo mail.

Si hubiese recolectado de forma maléfica todas las direcciones de correos de este tipo de correos... creerme que serían cientos de miles. Esto para un Spammers es algo así como el Santo Grial.

Exploits, el verdadero problema

Antes de comenzar, hay que saber que es un Exploit.

¿Qué es un exploit? Un exploit no es más que un trozo de código (normalmente) que se aprovecha de un bug o fallo de otro programa, Sistema operativo... con el fin de causar en este un comportamiento anómalo, a veces controlado y a veces no controlado. Pero cuanto más avancemos en este importantísimo capítulo, comprenderemos el verdadero peligro que estos traen consigo.

Un exploit, al contrario que un malware, no requiere la interacción por parte de un usuario normalmente. Los exploits tampoco infectan, ni siquiera se les puede considerar normalmente programas en sí mismos, aunque a fin de cuentas todo código que se ejecuta se puede considerar un programa. Normalmente los Exploits o son minis programas o más comúnmente scripts creados en Perl, Bash, Ruby... por simplicidad.

Como comencé este artículo, recordad la premisa de que ningún software está libre de errores, a veces errores de la mano de un programador, a veces el error es del compilador, a veces simplemente existe una forma de programar lo mismo de una forma mucho más segura. Con esta premisa en mano, podemos por tanto asegurar que cualquier software, desde la calculadora de Windows hasta Internet Explorer, Firefox, Safari, Windows 7, Snow Leopard... todos posiblemente tengan algún bug que potencialmente podría usarse para crear un exploit y crear un problema en dicha aplicación.

La utilidad de los Exploits es muy diversa, y es la primera arma de asalto para cualquier hacker. Un verdadero hacker descubre los propios fallos de seguridad de un sistema objetivo, crea los exploits necesarios y asalta el sistema con ellos. Después por medio de malware diverso, normalmente se mantiene una backdoor en el sistema remoto para volver a él más adelante.

Esto que describo es el pan nuestro de cada día para un verdadero hacker, sería el patrón a seguir más usual.

El mayor peligro que entrañan los exploits es sin duda alguna la vulnerabilidad del usuario a ellos y la naturaleza en sí misma del exploit. Ante un exploit no hay Antivirus que valga, tan solo asegurarnos que nuestro sistema está siempre completamente actualizado, no solo el OS, sino cualquier programa que usemos, ya que muchas veces el peligro no solo viene de dentro. Así, podríamos esclarecer la primera clasificación de Exploits. Os recuerdo que esto es tan solo la forma en la que veo yo todo esto, es decir, si se acude a cualquier fuente de información, esto puede ser completamente diferente, yo uso clasificaciones para que sea más simple comprender las cosas.

-Exploits Locales

Los exploits locales son los que entrañan un menor riesgo para el usuario doméstico, aunque no lo es tanto para una empresa. Un exploit local tiene como objetivo atacar un programa en ejecución de dicho sistema para producir en este un error de funcionamiento. Pero... ¿para qué puede ser esto útil? Tiene muchos fines.

Un exploit local si suele ser lanzado por un usuario o quizás a lo mejor por algún malware que lleva consigo un exploit.

El ejemplo más sencillo para ilustrar este tipo de exploits, es imaginar que a nuestro Word se le encuentra un fallo, por el cual es posible crear un documento Word maligno que en sí mismo lleva un malware. Un hacker podría enviar dicho documento de Word a cualquier persona. El usuario verifica que efectivamente es un documento de Word, sin trampa ni cartón. Lo ejecuta y con él se abre Word como siempre. A partir de ahí el exploit entraría en acción y por medio del supuesto fallo de seguridad se ejecutaría el código maligno insertado en dicho documento Word, y el malware se instalaría en nuestro sistema.

Otro ejemplo, un exploit local hacia un AV podría producir que este dejase de funcionar, e impedir la detección de malware. Claro, mucho pensará que eso es una tontería, pero pensar en entornos donde las cuentas de seguridad tienen un papel muy importante. Si eres un usuario no administrador quizás no puedas finalizar el AV, y aunque el exploit se ejecutaría como usuario limitado, el exploit afectaría de igual modo, y el AV se cerraría. Hablamos claro de un supuesto ejemplo en el que el exploit permitiese lo que estoy hablando.

Otro ejemplo, imaginar que al sistema de cifrado de Winzip se le descubre un fallo de seguridad. Imaginar un exploit que se aprovecha de dicho bug para ser capaz de descifrar cualquier zip creado con dicha versión de Winzip.

Ventaja: Normalmente el ataque viene de dentro, con lo que suele ser más fácil detener este tipo de exploits

Desventaja: Al ejecutarse directamente sobre el sistema, las aplicaciones/servicios propensos a ser atacados es mucho mayor, dado que se tiene acceso a una infinidad mayor de recursos.

-Exploits Internos

Con exploit interno no me refiero a un sistema propio, sino más bien a un sistema dentro de nuestra propia red local LAN. En este caso se trataría de lanzar un exploit sobre la máquina objetivo de nuestra red para producir en ella (o en alguno de sus componentes) el error deseado.

Aquí la peligrosidad se aumenta considerablemente, ya que ahora no hablamos de interacción del usuario objetivo, simplemente de encontrar un fallo de seguridad en los recursos expuestos por el sistema de la víctima, y evidentemente un exploit para lograr el fin deseado.

Imaginar la base de datos de clientes de nuestra empresa. Lo normal es que dicha base de datos sea accesible desde la red local. Sin conocer nada del servidor donde está alojada, es evidente que hay un claro recurso expuesto a la red: La base de datos. El servidor estará ejecutando por tanto algún servidor de base de datos y configurado correctamente para que los PC que tengan acceso a dicha base de datos puedan acceder a ella. Con un escáner se podría conocer datos como el tipo de servidor de base de datos, incluso la versión de este. Con estos datos un hacker es capaz a lo mejor de conocer o lograr sacar un error grave de seguridad que permitiese un acceso no autorizado a la base de datos de clientes. Al día siguiente el Hacker accede a la red local de la empresa, a lo mejor de forma directa o incluso por VPN. Una vez en red local lanzaría su exploit hacia el servidor de base de datos. El código malicioso llegaría al servidor y por ejemplo, daría acceso de administrador a la base de datos. El hacker una vez dentro, podría volcar la base de datos completa en su equipo, robando toda la lista de clientes de la empresa.

Ventajas: El radio de acción es mucho menor, dado que tan solo es posible buscar exploits/fallos de seguridad de los servicios que estén expuestos hacia la red local, por ejemplo servidores de base de datos, de nombres de dominio, servicios de impresión.... Es evidente que el rango de acción es infinitamente inferior al de los exploits locales, pero tener en cuenta que cualquier equipo en red local, suele tener expuestos varios servicios.

Desventaja: La peligrosidad es inversamente proporcional a la disminución del radio de acción. Es decir, se dispara. Estamos hablando del primer ataque citado en estas letras que no depende en modo alguno del usuario!! Nuestro sistema es comprometido sin tener nosotros culpa (suponiendo que el equipo está completamente actualizado).

-Exploits Externos

Este tipo de exploits son capaces ya no solo de afectar a una red interna, sino cruzar todo internet, de punta a punta para llegar al equipo objetivo destino. No tiene misterio y el principio es exactamente el mismo que los otros.

Un ejemplo de este tipo de exploits muchos lo conocerán, el famoso malware Blaster de hace unos años. ¿Pero era un malware o un exploits? En realidad Blaster era un malware, un gusano con funciones de backdoor. Lo que permitía la infección remota de dicho gusano era un exploit que producía la peor consecuencia de todas: Ejecución de código. El cómo trabaja un exploit lo veremos ahora.

Un ejemplo más común de este tipo de exploits es la creación de páginas web maliciosas que hacen uso en su código de algún exploit que produce el fallo en un navegador determinado. Esto lo ha vivido la mayoría de los usuarios en los tiempos de Internet Explorer, aunque a día de hoy por supuesto existen para todos los navegadores, es una de las principales puertas de entrada de los exploits, dado que afectan directamente al navegador del cliente. No son lanzados propiamente dicho, pero si afectan a un grupo mucho mayor de sistemas.

Ventajas: Tienen un rango de acción a un menor, y con la proliferación de los routers con cortafuegos es muy difícil poder atacar un sistema de un particular cualquiera, a menos que este tenga expuesto al exterior algún servicio que necesite, como una base de datos, un servidor web, un acceso de escritorio remoto... pero para un usuario no doméstico si son un problema, dado que cualquier servidor en la web tienen expuestos muchos servicios diferentes hacia Internet, aunque evidentemente mucho menores que los que están en red local.

Desventajas: Encontrar un exploit de ejecución de código o de acceso no autorizado a un sistema es lo peor que le puede pasar a un administrador de sistema. Imaginar que se localiza un exploit que afecta a la base de datos de clientes de un banco. Imaginar que un atacante por medio de un exploit de esta índole es capaz de tener acceso de administrador no autorizado a dicha base de datos, volcarla en su equipo y listo... acaba de robar una información que no tiene precio a día de hoy. Luego creo que hablamos de cosas más que importantes.

¿Cual es la razón de ser de un exploit? Hemos hablado que un exploit a fin de cuentas no es más que un código o trozo de este que produce un comportamiento erróneo en el programa objetivo. ¿Qué finalidad tiene esto? Hay diferentes funciones que puede desempeñar, y evidentemente no todos los fallos de seguridad pueden usarse para todos los objetivos posibles de un exploit!! Esto es muy importante, dado la misma seguridad del sistema. Desde mi punto de vista, de más importancia a menos, las razones de ser de un exploit serían:

-Ejecución arbitraria de código

Este sería el peor escenario posible. El término ya ha aparecido en alguna ocasión... ¿Qué significa? Que el exploit se vale de un fallo de seguridad del objetivo para producir que de algún modo dicho error acabe con la ejecución del código que incluye el exploit (este código del exploit se llama payload). Esto se hace posible principalmente gracias a las famosas desbordaciones de Buffer, o métodos similares, los cuales hicimos una breve explicación en el artículo sobre Windows 7.

En este tipo de exploits, lo normal es que el exploit de algún modo sea capaz de desbordar algún buffer del programa objetivo, de modo que una vez desbordado pueda colocar un payload. Este payload, en el mejor de los casos será un shellcode, es decir, un acceso remoto de consola a dicho objetivo. Está claro, una vez se obtiene esa Shell, el resto es crear un backdoor permanente en el sistema, esto se hace usando los propios comandos del sistema. Lo normal sería por ejemplo aprovechar que se está dentro para descargar en el equipo alguna

aplicación simple como ncat a priori, y una vez realizado esto pues se puede instalar un troyano o cualquier otro sistema de administración remota.

Otra opción es incluir de payload un virus por ejemplo. Pero hay que tener en cuenta que los payload suelen ser trozos pequeños de código, lo principal suele ser un shellcode, o al menos lo más deseado.

-Acceso no autorizado

En segundo lugar de importancia, al menos para mí, tendríamos exploits que tienen como objetivo lograr un acceso no autorizado a un sistema. Esto no tiene por qué significa que sea posible la ejecución de código malicioso, simplemente que se ha encontrado un fallo de seguridad de tal modo que podemos acceder al servicio sin que pudiésemos de otro modo.

Pensar por ejemplo en un acceso por SSH, que la mayoría de lectores de este blog sabrá que es. Lo normal es que un acceso por SSH esté protegido por un usuario y una contraseña, como es natural. Pues bien, un exploit de este tipo hacia SSH quizás permitiría realizar el acceso sin necesidad de autenticarnos, es decir... sin necesidad de conocer a lo mejor la contraseña.

Existen multitud de tipos de exploit que podríamos considerar dentro de este grupo, como por ejemplo Directory Traversal, Cross-Site Scripting, Spoofing... muchas de ellas sería necesario todo un artículo para ellas solas.

-Inyección de código

Este tipo de exploits es igualmente peligroso. Consiste en mal formar normalmente una cadena de entrada (como un nombre de usuario, una contraseña...) para que estas contengan más información de la que deberían de llevar, y normalmente información beneficiosa para un atacante. Esto es algo muy común cuando se quiere atacar una base de datos SQL, los denominados ataques de inyección SQL (muy comunes).

Son muy importantes estos tipos de ataques porque cada día más, nuestra sociedad hace un uso más exhaustivo de internet para todo, los servidores de bases de datos están por todos lados, así como foros, blogs... y cualquiera de ellos puede ser una potencial víctima.

Este tipo de exploits lo que logran suele ser a fin de cuenta un acceso no autorizado de algún tipo, pero otras veces suplantar/sustituir información o lo que sea necesario.

-DoS y DDoS

Se denominan así por sus siglas, (Distributed) Denial of Service, o ataques de denegación de servicio. El objetivo es simple, producir una saturación de cualquier modo en el servicio objetivo, haciendo que este se bloquee por un tiempo dado o de forma permanente, y sea necesario reiniciar el servicio en el mejor de los casos o todo el sistema en el peor de los casos. La denegación de servicio puede expandirse al ataque DDoS, en el que el concepto es el

mismo, pero se suele usar no solo un atacante, sino un conjunto de ellos, lo que hace que el ataque DoS sea más efectivo.

Así por ejemplo imaginar un exploit que lanzado sobre un servidor de base de datos hace que este retrase todas sus transacciones durante un segundo. Si esto lo realizan 100 personas tendríamos bloqueada la base de datos durante 100 segundos!! Hablamos de Exploits, no de otras técnicas convencionales para la generación de ataques DoS, las cuales hablaremos más adelante también. Es decir, el objetivo de este tipo de exploits es bloquear un servicio concreto o todo el sistema.

¿Qué utilidad tiene esto? A lo mejor el sistema objetivo al saturarse el servicio X entra en juego el servicio Y que si es vulnerable por un exploit de ejecución de código arbitrario. O el servicio X comienza a tener un comportamiento anómalo e inesperado, dando accesos no autorizados por ejemplo, o simplemente caer el sistema completamente por mera diversión.

Saturar un servidor de un "particular" quizás no es "divertido", ¿pero qué pasaría si fuésemos capaces de bloquear un servidor raíz de DNS? Ahora mismo internet, toda ella, se asienta principalmente en 13 servidores de DNS principales. La mayoría de ellos tienen réplicas por todo el mundo por si este se cae entrar en funcionamiento el otro. Los servidores raíces de DNS son punto de mira para cualquier hacker con delirios de grandeza... digamos que sería algo así como un sueño. Bloquear un servidor o dos no sería problema, la misma infraestructura actual sería capaz de redirigir el tráfico de los servidores caídos al resto, e incluso preparar de mientras los servidores raíces réplicas. Lo máximo que se ha alcanzado hasta la fecha (que se sepa) ha sido bloquear tan solo alguno de ellos, sin causar demasiados problemas. Aun así la teoría dice que es posible, un ataque DDoS a gran escala podría ocasionar la caída de ellos, y si esto ocurriese, las pérdidas a nivel mundial serían multimillonarias, amén del daño causado a toda la sociedad. Esto es para que se sepa la importancia de lo que es un ataque DoS.

Ni que decir tiene, que para poder efectuar ataques DDoS, hemos dicho que intervienen muchos puntos diferentes. La mejor forma por lo tanto de lograr esta "colaboración" es con ordenadores Zombi, es decir, ordenadores de particulares, empresas, universidades... que han sido infectado previamente con algún tipo de malware que está programado para efectuar el ataque tal día a tal hora o que están simplemente a la espera de órdenes de quien los manipula. De esta forma un solo hacker puede disponer miles de sistemas a su disposición, tan solo al alcance de un clic de ratón.

-Escalada de privilegios

Quizás en último lugar y con relativamente menos importancia, encontraríamos los exploits que producen una escala de privilegios. Estos ataques suelen ser más comunes en exploits locales, que pretenden poder ejecutar un malware cual sea con permisos de administrador por ejemplo. Es decir... de algún modo tener acceso a un recurso del sistema que tan solo sería accesible en primera instancia por el administrador del sistema.

La relevancia de esto es como cualquier otra cosa... depende del fin. Muchas veces un exploit de ejecución de código requerirá anteriormente uno de escala de privilegios. O por otro lado muchas veces un exploit DoS lo único que hace es finalizar el proceso sin que tenga mayor importancia, y en ese caso tendría una repercusión aun menor.... todo depende del caso concreto.

Creo que queda claro cuál es el primer grado de peligrosidad de un exploits: Su dominio. Un exploit remoto siempre será potencialmente más peligroso. Pero al margen del dominio, no podemos presuponer que cualquier exploits sea una seria amenaza de seguridad para nuestro sistema, depende del exploit. Un exploit se basa a fin de cuenta en un fallo de seguridad, y no todos los fallos de seguridad tienen la misma relevancia. Si el software X se le descubre un fallo de seguridad remoto, se podría inducir de ello que sería posible (siempre de forma teórica) crear un exploit remoto para dicho fallo de seguridad. Pero igualmente no todos los fallos de seguridad son explotables (o sería ¿Explotables?).

Por eso cuando se habla de exploits se habla directamente de fallos de seguridad. El fallo de seguridad es el que permite la creación del exploit específico. Pero antes de ver otra clasificación, creo que es importante conocer el funcionamiento habitual de un exploit, como funcionan:

1º. Todo comienza analizando el sistema objetivo, para ello lo normal es usar herramientas de escaneo de puertos o escaneo de vulnerabilidades, como puedan serlo Nmap o Nexus respectivamente. Por supuesto, un buen hacker será capaz de analizar el sistema objetivo, ver los servicios expuestos de este y crear un exploit para dicho sistema. Esta vulnerabilidad podrá ser remota o local y podrá ser usada o no para uno u otro fin.

2º. Dependiendo del fallo encontrado, podremos crear un exploit más dañino o menos dañino... con un objetivo u otro.

3º. Dependiendo del objetivo del exploit, será necesario incluir un payload (por ejemplo en el caso de un exploit de ejecución de código arbitrario) . Un Payload es digamos un trozo de código, normalmente escrito en código máquina. Este Payload puede ser un malware, puede ser un shellcode (es decir, un código que nos devuelve una Shell remota o local del sistema objetivo)... o simplemente el exploit puede carecer de payload.

4º. Una vez terminado todo, se lanza el exploit contra el sistema objetivo.

Hemos hablado del peligro real de un exploit. Sí, cualquier sistema está expuesto a ellos y un hacker habilidoso es capaz de poner en jaque nuestro sistema sin necesidad de una intervención de este. Este es el motivo que hace que realmente los exploits sean peligrosos, frente al malware, que no deja de ser molesto.

La solución para el Malware la conocemos, la hemos dado: Tener dos dedos de cabeza. Hemos explicado incluso como eliminar el 90% de todo el malware en caso de infección e incluso pautas para no necesitar siquiera un AV. ¿Pero qué podemos hacer contra los exploits?

Defensas contra exploits

Para los usuarios domésticos, creerán que los exploits son armas tan solo a mano de algunos, que no existen, que no importan... lo que sucede con los exploits es que normalmente son completamente silenciosos, no quita que no hayamos sido víctimas de alguno de ellos. Pero por suerte para nosotros, es cierto igualmente que ser víctimas de un exploit tipo externo no es tan común para el usuario doméstico, aunque mucho más expuestos ante un exploit interno (dentro de la misma red)

Por lo que he explicado, tendríamos que pensar en un PC de un usuario medio conectado permanentemente a internet. Hoy en día lo normal es que esté conectado a través de un router que actúa también de Firewall y un dispositivo NAT, lo que hace mucho más complicado que un atacante alcance nuestro sistema, dado que con lo que se encontrará en primer lugar será el mismo router. Para que un ataque fuese directo a nuestro sistema, el router debería de tener abierta alguna conexión a nuestro sistema.

Si pensamos en un PC estándar, podemos imaginar que los servicios expuestos (potencialmente objetivos para los exploits) serían tales como el navegador web (Firefox, Internet Explorer, Chrome...), programas de mensajería instantánea (Como AIM, Skype, Messenger...), Gestores de correos (Como Thunderbird, Eudora, Outlook)... para que estos programas funcionen correctamente, necesitan establecer conexiones más allá del router. Qué pasaría si un hacker conoce un exploit remoto de ejecución de código de la última versión de Messenger? Con un escáner de puertos podría escanear el router, buscar alguna conexión activa que le llegue a pensar que está usando Messenger y lanzar el ataque ha dicho puerto concreto. Pues que si el sistema no tiene las medidas debidas de protección, un usuario felizmente usando su Messenger acaba de introducirse en su sistema a nuestro viejo amigo Perico, y no se ha dado ni cuenta. Esto mismo se extrapola a los navegadores o cualquier aplicación que interaccione con internet.

Si hablamos en un PC conectado a una red más grande como la LAN de una universidad, de una empresa... el problema es mucho mayor, dado que los servicios expuestos a la red interna son mucho más numerosos, con el consiguiente problema. Por ejemplo, en Windows no es raro encontrar en ejecución NetBIOS, SMB u otras tecnologías de Microsoft. En Linux a lo mejor encontramos que se tiene un servidor web Apache en funcionamiento (aunque solo en red local, sin acceso a internet). Es decir... el problema se multiplica por mil en una red local extensa. A fin de cuenta en la red de tu casa el único que puede molestarte es tu hermano, tu hijo... o, y esto es IMPORTANTE!! Alguien que pirateó tu WIFI por una falta de seguridad por tu parte y tiene acceso a tu red, y una vez en tu red puede hacer verdaderos destrozos.

¿Entonces estamos completamente indefensos? No.

La primera defensa efectiva es la **actualización** de todo el software. Asegurarse de que nuestro software está siempre al día es la mejor arma. Es cierto que el bien hacker será capaz de encontrar un nuevo agujero por el cual colarse, pero al menos las posibilidades disminuyen considerablemente. Tener en cuenta que Microsoft por ejemplo emite semanalmente o bisemanalmente un boletín de seguridad con X actualizaciones. ¿Para que pensáis que son estas actualizaciones de seguridad? Principalmente para quitar fallos de seguridad que podrían suponer la creación de un exploit. Pero está claro... los boletines de seguridad dan información a veces escueta a veces más extensa... pero información que pueden a fin de cuenta usar los hacker para crear exploits para máquinas que no han sido aun actualizadas. Estoy seguro que aun a día de hoy puedo encontrar máquinas que son receptibles al famoso Blaster, por no estar actualizadas.

Creerme, actualizar es imprescindible. Pero no solo el OS, que por supuesto es lo más importante!! Sino cualquier software que pueda ser un objetivo, principalmente navegadores y otros. Esto no es exclusivo de Windows, esto es aplicable a todos los sistemas. Es más, como expliqué en el artículo de Windows 7, Windows 7 es junto con Algunas distros de Linux el OS más seguro en la actualidad con mucha diferencia (sin incluir evidentemente OS propietarios), pasando y por mucho a sistemas como Snow Leopard, entre otros.

En segundo lugar, sobre todo para aquellos que necesitan servicios extras como programas P2P, servidores web, bases de datos... ya sean particulares o empresas, tener unas buenas políticas de seguridad en los routers/firewall, no abrir nunca más puertos de los necesarios, usar tan solo lo necesario. Un puerto abierto es un agujero de seguridad.

Por otro lado no hay que olvidar el uso de encriptación y autenticación en procesos sensibles. Por ejemplo cuando deseamos accesos a bases de datos o a servidores Active Directory, accesos VPN... usar siempre los sistemas de autenticación más fiables que existan, usar cuando se puedan certificados digitales. Aunque este bloque será más de interés para empresas que para particulares.

Y en último término, y no menos importante, disponer de un hardware y un OS que en el peor de los casos sea capaz de luchar contra estos exploits. Por ejemplo, el peor caso es evidentemente la ejecución de código arbitrario. Gracias a tecnologías hardware como XN (DEP llamada en Windows, bit de deshabilitación) que ya comentamos en su día o tecnologías como ASLR, prácticamente inexistentes en Snow Leopard. Estas tecnologías hacen que realizar un exploit que sea capaz de ejecutar código arbitrario sea casi imposible. En cambio estas tecnologías no impiden otro tipo de exploits, como escaladas de privilegios, inyección de código o ataques DoS.

Hemos hablado tan solo de algunas soluciones... pero es que tampoco existen muchas más. Para las inyecciones de código tan solo programar decentemente las aplicaciones que se enfrentan a ellos, con la creación de algoritmos que chequean de forma óptima todos los posibles ataques, pero aun así nadie es perfecto y siempre algo se olvida. Frente a la escalada de privilegios más de lo mismo, por bueno que sea el programador siempre se le escapará una posible vía de acceso... y mejor ni hablar de los ataques DoS, dado que las defensas frente a este tipo de exploits son prácticamente nulas, simplemente el servicio objetivo tiene un

malfuncionamiento y puede colgarse él o todo el sistema. Si es un buen sistema, lo normal es que tan solo sea anulado dicho servicio.

Por poner un ejemplo de Exploit real, hace unas semanas apareció un gracioso exploit que afectaba a todas las versiones de Windows Vista. El exploit era efectivo tan solo en redes locales y producía un ataque de denegación de servicio que producía el reinicio de todo el sistema. ¿Vale, esto sirve para algo? Bueno, imaginar lo gracioso que es este exploit en una red grande, como en la universidad o la empresa. El equipo remoto que quiera de la red que esté ejecutando Windows Vista se puede reiniciar a mi voluntad. ¿Gracioso verdad? Es evidente que para tal fallo de seguridad hace ya tiempo que tiene su actualización pertinente, pero no deja de ser útil, seguro, para al menos ¿un 40% de usuarios? Vale... no es un exploit "peligroso", pero cuanto menos interesante.

Esto es diario. Algunos exploits se hacen famosos, otros permanecen celosamente guardados con oro en paño. No es nada raro venderlos incluso. Imaginar un hacker que logra un exploit de acceso no autorizado a la versión más actual de SQL server. Estoy seguro que muchos pagarían fortunas por ellos.

Recordar que el Jailbreak de los iPhone e iPod Touch es posible simplemente por exploits :), en nuestro caso un exploit o dos de ejecución arbitraria de código. No hay ningún sistema seguro frente a esta gran amenaza.

Para acabar con los exploits, y lanzar un guiño hacia la comunidad Apple, os diré que si bien es cierto que por una cuestión de estadística existe más malware para Windows que para MAC OS, encontrar un fallo de seguridad explotable en MAC OS por un exploit es muchísimo más fácil que encontrarlo en y explotarlo en Windows. Prueba de ello es el propio iPhone/iPod, o las carentes medidas de seguridad mínimas de Snow Leopard frente a ellos.

La tecnología, insegura en sí misma: Redes

Hemos hablado de los dos problemas fundamentales, el usuario y los exploits. Pero existe un tercer enemigo, más silencioso aun que los exploits, aunque estos lo conocen bien como un gran aliado: La tecnología en sí misma.

Internet es algo relativamente nuevo, así como las redes LAN en general, ya sean redes Ethernet, Token Ring, Wireless... La mayoría de nosotros hemos visto nacer estas tecnologías y de cómo han evolucionado con los años. Pero la mayoría de los grandes inventos no nacen pensando en hacer la trampa, sino de un servicio que se pone para nuestra comodidad. Famosa frase de Einstein cuando le preguntaron sobre la bomba atómica: "Si lo llego a saber me hago relojero".

Internet se basa en un conjunto de protocolos que son de naturaleza inseguros y antiguos la mayoría de ellos. En su simplicidad está su gran beneficio para la sociedad, su fácil adaptación

su... pero igualmente su falta de seguridad. Con los años las tecnologías han evolucionado a un ritmo vertiginoso principalmente frente a la seguridad. Si pensamos en la web como tal, en Internet... en realidad funciona exactamente igual que en su nacimiento!! El esquema de Internet es fácil de comprender, es sencillo, es eficiente, es una máquina bien engrasada!! Pero es necesario aplicar capas de seguridad, cuando estas no se aplican, aparecen los problemas.

Esto mismo es aplicable a las redes locales... multiplicado por mucho.

El ejemplo más claro es la poca o ninguna seguridad que tiene una red WIFI usando WEP, o la falta de efectividad de las técnicas de filtrado MAC u ocultación del SSID de un AP (punto de acceso wifi). Es más, recientemente WPA-PSK ha sido también prácticamente roto. Es evidente que cuando nació WIFI nadie iba a pensar que una encriptación WEP sería posible romperla en 5 minutos (el record del mundo creo que está en 10 segundos). Pero gracias a esto emergieron nuevos sistemas como WPA/WPA2 (PSK y no PSK).

Pero sobre las redes Ethernet tampoco podemos decir que sean nada seguras. Una red Ethernet se forma normalmente por un router y un Hub (antiguamente) o un Switch (que es lo usado actualmente). En el caso de los Hub es más sangrante, dado que en los Hub, todos los clientes están conectados al mismo bus, es decir, todo lo que envía o recibe cualquier cliente puede ser recibido por cualquier otro cliente. Y en el caso de los Switch tampoco es que se mejore demasiado...

De todo esto es de lo que vamos a hablar, de los peligros que entrañan la red, simplemente por el mero hecho de existir.

Internet

Efectiva, rápida, versátil... pero insegura. A día de hoy aun se usa el protocolo IPv4 lo que tiene el problema añadido de que ya nos estamos quedando sin direcciones IP. El protocolo IPv6 (su sucesor) aun se resiste a imponerse.

El principal debate es... ¿debe de ser internet completamente anónima? Todo PC conectado a internet tiene asignada una IP, ya sea de forma directa a la red o de forma indirecta a través de un dispositivo NAT (como u router) que le asigna una IP privada, pero de cara al exterior usa una IP asignada por su proveedor de servicios (ISP). Esta IP identifica directamente al cliente del ISP, ya sea el cliente un cibercafé, ya sea una empresa, ya sea un particular. Que la IP sea dinámica (cambia con el tiempo) o estática (No cambia) no implica que en todo momento cualquier persona puede ser identificada con una orden judicial por su IP. Es decir, Internet no es para nada anónimo.

Claro que esto no es tan así. En una empresa pueden existir cientos o miles de computadores en la misma red, y a lo mejor todos comparten la misma IP pública (la expuesta a internet). ¿Cómo saber quién es quién?

Todas estas preguntas introductorias son importantes, a la hora de comprender la seguridad intrínseca de internet. Al igual que los programas informáticos son inseguros por naturaleza, internet lo es del mismo modo en un escenario estándar (sin incluir protocolos de seguridad que se usan para ello).

Los problemas reales de seguridad de Internet, no obstante, no dependen en lo más mínimo de nosotros. Antes de explicar las vulnerabilidades de Internet, vamos a ver a muy groso modo como funciona internet en este ejemplo, que podría ser un ejemplo real perfectamente. En este supuesto, nuestro amigo, el virus Perico se quiere conectar al servidor de Google Earth para comprobar cierta información. Esto es algo trivial que hacemos nosotros cientos o miles de veces al día, cada vez que deseamos acceder a alguna web o enviar/recibir cualquier dato de la red:

Como dispositivos conectados a internet, cada nodo tiene que disponer de una IP que lo identifica dentro de la red de redes. Por un lado tenemos el sistema infectado por Perico y por otro lado el servidor de Google Earth. En cambio, Perico está en un sistema que a su vez está conectada a una Red local, es decir, pasa a través de un router. El servidor de Google en cambio está conectado directamente a Internet, luego la dirección IP que tiene en principio es siempre la misma.

Perico lo único que conoce es la web a la que desea conectarse, que es "earth.google.com". Luego este es el mapa que se nos plantea actualmente, y que es lo único que conoce Perico:

Perico:

Nombre de Host: Perico

Dirección IP: 192.168.0.2

Puerta de Enlace: 192.168.0.1

Servidor DNS: 192.168.0.1

Nombre de Host destino: earth.google.com

Router:

Nombre de Host: Router del usuario

Dirección IP: 192.168.0.1

Dirección IP del ISP: 80.85.25.100 (Una IP de Telefónica)

Servidor DNS del ISP: 80.58.61.250, 80.58.61.254

Perico (192.168.0.2) -> Router del Usuario (192.168.0.1 - 80.58.25.100) -> Earth.google.com

Este es el esquema genérico. Vamos a ver punto a punto como esto se lleva a cabo:

1º. Perico quiere establecer una conexión a earth.google.com

2º. Perico no conoce la dirección IP de earth.google.com, dado que no la tiene en su caché de DNS, necesita por tanto realizar una petición DNS a sus servidores de DNS antes de poder conectarse a earth.google.com.

3º. Perico tiene asignado unos servidores de DNS, lo que realiza por tanto es una petición DNS a dichos servidores. Esta petición no es más que un mensaje hacia dichos servidores de DNS que diría algo así:

¿Que IP tiene earth.google.com? lo pregunta 192.168.0.2 (Perico) -> Enviado al Router (192.168.0.1)

Los servidores de DNS de Perico apuntan al router, con lo que este mensaje se envía al router. El router a su vez recibe la petición, y como él tampoco tiene en su caché de DNS dicha entrada, reenvía la petición a los servidores de DNS establecidos en el router:

¿Que IP tiene earth.google.com? lo pregunta 80.58.25.100 (El router) -> Enviado a 80.58.61.250 (Servidor primario DNS de telefónica).

Y así mismo, el router recuerda que la petición vino por parte de Perico.

Como estamos imaginando el peor de los casos, en el que ningún servidor de DNS tiene en su caché la respuesta, la petición llega al servidor de DNS de telefónica, y como este no la conoce, tiene que realizar la petición a un servidor DNS de mayor jerarquía. El servidor de DNS de telefónica posiblemente haría lo siguiente:

Dirección buscada: earth.google.com

Se disecciona de derecha a izquierda, enviando la petición al servidor correcto. En este caso, imaginemos que telefónica tan solo conoce la IP del servidor de DNS que gestiona los ".com", un servidor raíz. Es decir, si le pregunto al servidor raíz sobre ".com" que es un dominio de primer nivel, seguro que puede resolverme la IP de google.com, dado que él es la principal entidad que administra esto.

El servidor raíz recibe la petición de resolución de DNS earth.google.com, pero él tan solo conoce la IP de google.com. No importa, se trabaja de derecha a izquierda. El servidor raíz reenviará su petición a google.com, con el mensaje: Necesito conocer la IP de Earth.google.com. La petición por última instancia llegará a los servidores DNS de google.com. En estos, lo único que queda verificar en la lista es "Earth", y como dicho servidor es precisamente quien lo gestiona, tendrá la IP dada. Como la tiene, envía la petición de vuelta con la IP de earth.google.com hacia el servidor raíz, este al servidor DNS de telefónica, este al router, y el router a Perico.

El proceso es simple: Cada servidor DNS solo conoce lo suyo (en realidad no es así, pero en el peor de los casos sería algo así). Así el servidor de telefónica no tiene por qué conocer a priori la IP de earth.google.com, ni siquiera de google.com, pero si conoce la dirección de los servidores ".com". Estos a su vez tampoco conocen la dirección de "earth.google", pero si conocen la dirección de "Google", enviando la petición a estos. Y estos últimos sí que conocen la dirección de "Earth". Como digo esto no es así dado que los servidores de DNS tienen bases de datos ingentes a modo de caché, para que no sea necesario todo el viaje, pero en el peor de los casos sería como digo así.

4º. Una vez que Perico conoce la IP de erath.google.com, puede comunicarse ahora con él directamente. Directamente entre comillas claro, dado que todo pasa por el router de Perico, por los routers de telefónica en primera instancia... y nuestra petición irá saltando de router en router hasta llegar al final al servidor de Google concreto. Este servidor cuando recibe la petición, le da curso y responde a dicha petición, produciendo el retorno de esta (no tiene porqué siquiera usarse el mismo camino)

Si se ha comprendido esto, podemos encontrar grandes problemas de seguridad en todo ello, y de como la intervención de un "ente" maligno puede tirar abajo todo el sistema, o mejor... apoderarse de ello. Vamos a lanzar preguntas al aire que iremos contestando:

-Si toda la información de Perico va al router ¿Cualquiera que está en la misma red local podría tener acceso a dicha información?

Efectivamente, esto es posible. Dependiendo si nuestra red sea administrada con un Hub o un Switch será necesaria una técnica u otra. Lo extenderemos más ampliamente en "Redes LAN".

-Si toda la información de Perico va al router y de este por los servidores de nuestro ISP ¿Nuestro ISP puede conocer todo lo que hacemos?

En teoría pueden, en la práctica lo hacen. Hay leyes que obligan a los ISP guardar creo que son durante X meses información importante, accesible en teoría tan solo por orden judicial en caso de terrorismo u otro tipo de delitos.

-Si toda la información de Perico a Internet va circulando por diferentes Routers... ¿Podría alguno de ellos monitorizar nuestras actividades?

En teoría se puede y se hace. El uso de esta información normalmente está restringida completamente a altos organismos, de nuevo hablamos de terrorismo o grandes redes de delincuencia. De esto hablaremos brevemente.

-He dicho que los servidores de DNS son los que nos dicen la IP del sitio al que deseamos conectarnos... ¿Nos tenemos que fiar que el servidor de DNS nos envíe seguro la IP correcta?

En teoría todo nace por una buena voluntad, pero quien hace la ley hace la trampa. Si se interviene un servidor DNS y se cambian las entradas, el resultado creo que es más que comprensible. Si hacemos una petición a nuestros servidores DNS de telefónica y estos han sido maliciosamente modificados, a lo mejor cuando solicitamos acceso a "cajasol.es" el servidor infectado de telefónica no nos devuelve la IP real, sino una IP que nos dirige por ejemplo a una web maliciosa de suplantación para capturar datos de clientes (usuarios y contraseñas), sin que el usuario se dé cuenta absolutamente de nada, dado que para él, estará en cajasol.es, sin saber que realmente no está conectado a dicho servidor.

Por un lado podemos estar tranquilos, sabiendo que hoy por hoy cualquier hacker lo tendría más que complicado la manipulación de las tablas DNS de los servidores o modificar las tablas de rutado de los routers. Aunque todo esto es más que posible, ojo!! Y si no puede hacerlo una persona corriente es simplemente porque por defecto nuestro ISP filtra determinada información. Pero qué pasaría si pudiésemos acceder a la red con un enlace más directo, sin pasar por un ISP? La red es universal, cualquiera puede conectar un servidor y listo, cualquiera podría modificar entradas de DNS o tablas de ruta... esto ha sido siempre un peligro!! Pero por otro lado ha sido lo que ha permitido la gran expansión de internet.

Pero por otro lado tenemos que fiarnos de los gobiernos y de los ISP, y de lo que en teoría deberían de hacer y lo que en la práctica pueden hacer.

-Eavesdropping

Esta palabreja es el término genérico para designar una escucha (normalmente completamente fuera de la ley) entre una "conversación" privada entre dos sistemas informáticos, ya sea sobre internet o sobre una red local. Aunque suene a ciencia ficción como he dicho, existen algunas tecnologías implementadas y funcionando a la perfección para la interceptación de cualquier comunicación entre computadores.

Así por ejemplo existió (ha salido en muchas películas) el proyecto "Carnivore" del FBI, el cual estaba designado a interceptar como digo cualquier comunicación por Internet de cualquier usuario americano. Este no obstante fue sustituido (esto no lo sabía hasta hoy) por otro proyecto, actualmente es usado "Narus", que no deja de ser un sistema similar. Básicamente supercomputadores que registran cualquier contenido que pueda ser peligroso para inteligencia que circule por internet. Es como un Sniffer a lo bestia.

Otro interesante proyecto es TEMPEST. La idea (que no es idea sino una tecnología real) es ser capaz de monitorear sin necesidad de estar conectado a la red, simplemente escaneando las señales eléctricas desprendidas de los mismos dispositivos electrónicos. Por medio de las radiaciones, espectros electromagnéticos... de estos dispositivos es posible por lo visto inducir la información que pasa por ellos.

Y por supuesto, posiblemente el más famoso de todos los proyectos fue y es Echelon. Hay pruebas reales de que existe y creo que hasta fue reconocido hace poco por EEUU. Básicamente, sería un sistema de Eavesdropping a nivel mundial, no solo de Internet, sino de todas las comunicaciones, ya sean telefónicas, vía satélite, Internet, de voz, de datos... todo.

Si, estáis escuchando bien, posiblemente con toda seguridad, todos nuestros datos son recogidos por sistemas informatizados que procesan dicha información y la almacenan o no en función de complejos algoritmos informáticos que evalúan la peligrosidad de la información. Es decir, posiblemente si enviamos correos electrónicos con palabras como atentado, bombas, nuclear... posiblemente sean identificados como información de riesgo y esta sea procesada o almacenada de algún modo. Y esto no es una broma, es completamente algo real que existe.

-DDoS

Ya hemos hablado de los ataques DDoS en los exploits... pero un ataque DDoS no solo puede llevarse a cabo por medio de exploits. Su uso más común es aprovecharse de Internet para perpetrar ataques de este tipo. El ejemplo más simple? Pensar en 1 millón de ordenadores, sincronizados, solicitando la misma información al mismo servidor una y otra vez durante una hora. Posiblemente esto produciría una degradación o un bloqueo total del servidor en cuestión. A esto se denomina **Flood** (o Inundación), es decir, una cascada de información de golpe que puede sobrepasar el límite de un sistema.

Existen multitud de tipos de Flood, aunque todos ellos normalmente lo que persiguen es lograr una denegación de servicio en el objetivo, es decir, un DoS. Por ejemplo, el más típico pueda ser el Flood por ping o el Flood por paquetes ICMP.

Por suerte esto es algo que puede ser relativamente subsanado con unas buenas políticas en los firewall, de forma que sean capaces de filtrar o bloquear el acceso o... aun así, los ataques DoS son diarios en todas las redes, algunas veces siquiera tienen importancia y a veces son capaces de bloquear el acceso al servicio durante segundos, minutos, horas...

-Escáner

Dado que cualquier sistema en internet está localizado por una IP y dado que ni siquiera necesitamos la IP, sino que nos vale su nombre de dominio, cualquier sistema queda expuesto permanentemente a internet. Para los servidores o recursos permanentes de internet esto es algo que no tiene más remedio, es necesaria la IP para poder acceder a ellos. Para el usuario doméstico en cambio, rara vez necesitaría que otro supiese su IP, y de hecho casi nadie tiene asociada la IP a un nombre de dominio, aunque esto último cada día se realiza más a menudo para poder acceder a sus sistemas desde cualquier parte del mundo sin necesidad de preocuparse de la IP dinámica/estática de su ISP.

Aun así, cualquier conexión entre dos sistemas, se realiza normalmente IP-IP, con lo que existen un millón de formas de poder conocer la IP de una víctima. ¿Pero por qué es tan importante esta IP?

Una vez que conocemos la IP de la víctima (o el nombre de dominio asociado a ella) podemos de forma simple realizar scanner en dicha máquina. Estos escáneres nos darán una cantidad de información increíble sobre nuestro objetivo!! Con los routers domésticos, es cierto que nuestro escáner tan solo se realizaría sobre el router, y no sobre el PC que estaría detrás de él, pero sí que tendríamos acceso a los puertos en ese momento usados por el PC detrás del router. Pensar en un router como una puerta. Si el PC que está detrás del router está usando Messenger, el router seguro que tendrá abierta alguna conexión a dicho PC, y por medio de un escáner podemos alcanzar al PC.

Un escáner nos puede decir directamente los servicios que tiene expuestos la víctima, a veces su PC a veces su router. Pero no solo que servicios, sino que versiones, que software... nos puede decir el Sistema Operativo que están usando e incluso si el sistema objetivo es

vulnerable a ciertos exploits. Conociendo por ejemplo la versión de un servicio, puedo buscarle un fallo de seguridad que me permita crear un exploit que me permita el acceso a su sistema. Todo esto tan solo con saber la IP/Dominio de un objetivo.

Ejemplo de escáner sencillo con Nmap al servidor "elbruto.es":

```
22/tcp open ssh OpenSSH 4.3p2 Debian 9etch3 (protocol 2.0)
80/tcp open http Apache httpd 1.3.39 ((Unix) PHP/5.0.4)
222/tcp filtered rsh-spx
443/tcp open ssl/http Apache httpd
722/tcp filtered unknown
1022/tcp open ssh OpenSSH 4.7p1 Debian 8 (protocol 2.0)
1122/tcp open ssh OpenSSH 4.7p1 Debian 8 (protocol 2.0)
1720/tcp filtered H.323/Q.931
```

Simplemente con esto, y tan solo es un escáner simple y rápido realizado en unos segundos, podemos obtener toda esta información:

Casi con toda seguridad el servidor está corriendo Debian Etch, esto puede ser usado para buscar exploits para dicho OS.

Vemos diferentes puertos para SSH, concretamente OpenSSH 4.7p1, es decir, acceso remoto. ES decir, si encontramos un exploit para acceso no autorizado para OpenSSH 4.7p1, el servidor estaría completamente indefenso a nosotros.

Vemos también el servidor web que está usando y su versión, lo que sería igualmente útil para buscar un exploit contra él, quizás para un DoS o quizás algo más importante.

Y esto tan solo un simple escáner que puede ser mucho más riguroso y nos daría mucha más información del sistema objetivo. Con todo esto quiero decir que la amenaza está ahí fuera siempre, que es muy simple para alguien que tenga el conocimiento poner en jaque un sistema que no está debidamente protegido.

-eMail

Este punto bien lo podríamos a ver incluido en el anterior, pero así le doy más importancia. El correo electrónico es posible simplemente gracias a unos protocolos que existen en internet. Pero al igual que este, es un servicio universal, y con universal me refiero a que cualquier persona puede crear el correo que quiera.

Es evidente que no es práctico que cada persona tenga su propio servidor de correo, pero ello no quita que en teoría sea posible. Por ejemplo, puedo instalar un servidor de correo en mi equipo, con el realm que quiera: @gmail.com, @hotmail.com... da igual para dotar a la red interna de mi casa de un servicio de mensajería. Incluso si quiero podría enviar dichos correos al exterior!! Cuando se está usando el realm que pertenece a otra entidad, se habla de Spoofing de email, es decir, suplantación de identidad de un correo electrónico. Esto es diariamente usado para intentar engañar con el Pishing y Scam .

Es igualmente cierto que aunque está permitido este tipo de prácticas, los principales servidores de correos electrónicos filtran dichos correos si provienen de una IP de la cual no coinciden sus realms. Es decir, Google tiene en sus bases de datos que las IP de los correos de hotmail.com están a lo mejor en el rango 60.0.0.1 - 60.1.0.0.1, Si está en dicho intervalo permitirá a sus servidores recibir el correo, si no está en dicho intervalo, el servidor no permitirá el paso del correo por sus servidores, al categorizarlo como Spam. Otros servidores por ejemplo impiden el envío desde equipos con IPs dinámicas. Existen técnicas para evitar esto, no obstante no son ni mucho menos perfectos. Por ejemplo imaginar que tengo una empresa adscrita a listas que me dan acceso a estos servidores, y uso de fraudulenta mi servidor de correos para inundar todas las cuentas de correo q quiera con emails con las direcciones suplantadas. O pensar en un gusano, que usa el equipo objetivo para realizar el envío, valiéndose de la IP de la red del objetivo.

Esta inseguridad es el precio que hay que pagar por mantener una red neutral. Yo no tengo que pagar nada a nadie por montar un servidor web en mi casa, es completamente legal y si tengo además un dominio propio tener los correos que quiera tipo @theliel.es. No tengo que pedir permisos, no tengo que pagar tasas o royalties. Internet es libre, la crea cada persona. El único organismo real que existe en internet es la ICAN, y simplemente se encarga de gestionar los dominios, y es una organización sin ánimo de lucro. Otras organizaciones de estándares trabajan para internet claro está, pero en pos de mejorarla con mejores tecnologías. Algunos suelen ser hasta foros abiertos.

LAN

El Eavesdropping es algo real, pero las técnicas más "agresivas" de él tan solo es ficción para nosotros, es decir... no nos preocupamos de ello. Esto es simple, nos guste o no, existen gobiernos, leyes, criminales, mafias... no soy partidario de escuchas, de manipulación de datos... pero está claro que hay que asentar un punto medio entre lo necesario y la libertad. Honestamente no creo que hoy por hoy estos sistemas se usen de forma indiscriminada, aunque no soy nadie para decir o suponer algo así.

En cambio en un entorno LAN nuestra seguridad se puede ver seriamente comprometida de formas mucho más simples, sin gobiernos, sin conspiraciones... tan solo un usuario malicioso en dicha red es suficiente. Muchos pueden pensar que por esta razón su red LAN de casa es completamente segura... pero con la expansión exponencial de los puntos de acceso wifi, esto es un auténtico problema, dado que el 80% de los usuarios son completamente vulnerables por WIFI por una mala configuración.

Hay que tener en cuenta que tener configurado WIFI con una encriptación tipo WEP, no sirve para nada absolutamente. EL peligro no es que un usuario nos robe algo de conexión (que tampoco es de agrado de nadie), sino que al estar conectados a nuestra red puede tener acceso a todo nuestro tráfico de datos!! Y esto sí que es un auténtico problema. Esto se

incrementa exponencialmente de nuevo si nos encontramos en una LAN de gran tamaño, como la universidad (ya sea siempre por WIFI o cable), la empresa... puesto que nunca sabremos quien está escuchando al otro lado.

Es cierto que en una LAN no suelen existir problemas de DNS, y que el porcentaje de personas que puedan atacar por LAN son mucho menores que las personas que hay en Internet. No obstante las vulnerabilidades de una LAN son también mayores. Ya explicamos cómo funcionaban las cosas por Internet, pero dentro de una LAN esto suele ser bastante diferente, dependiendo en gran medida del dispositivo de red que tengamos. Así por ejemplo la primera distinción importante que tendríamos que hacer es entre Hub y Switch, aunque prácticamente en la totalidad de casos hoy en día, se usan Switch.

-Esquema básico de funcionamiento de una LAN

Supongamos dos equipos dentro de una misma red que quieren comunicarse uno con el otro. Pongamos que nuestro virus Perico quiere acceder a las carpetas compartidas de otro equipo, llamémosle "ObjetivoZero". Supongamos de nuevo el peor de los casos, en el que nadie conoce nada, tan solo que ambos equipos han sido conectados a un dispositivo de red Hub o Switch y que cada uno está apropiadamente configurados su IP, DNS, Puerta de Enlace:

Perico (192.168.0.2) -> Hub/Switch -> ObjetivoZero (192.168.0.5)

Perico se conecta al Hub/Switch y sabe que la IP de las carpetas compartidas a las que desea tener acceso es 192.168.0.5. Pero si los routers trabajan con IPs para enviar un dato a cualquier punto de la red, un Hub/Switch no suelen trabajar con IPs (de hecho para un Hub es imposible). Estos dispositivos entienden tan solo de direcciones físicas.

¿Pero entonces que es la dirección física? Cada adaptador de red (es decir, los adaptadores Ethernet, wifi...) tienen cada uno de ellos grabado en fábrica un ID único, algo así como un número de serie. Este ID se conoce como MAC, Media Access Control (Control de Acceso al Medio). Este ID tiene el formato de 6 octetos, representado comúnmente como una agrupación de 6 valores dobles hexadecimales: FF:AA:BB:12:34:00. Es interesante saber que simplemente conociendo la MAC de un adaptador, se puede conocer marca y modelo de nuestro adaptador, ya que si no me falla la memoria, el primer octeto siempre se debe de establecer a 00, el segundo y creo que el tercero representa el ID del fabricante, y el resto para el modelo y lote. No me hagáis mucho caso porque sinceramente no me acuerdo que octetos eran para cada cosa.

Pues bien, cuando estamos detrás de un Hub o un Switch, para poder enviar algo a alguien necesitamos conocer no solo su IP, sino también su dirección MAC. Un Switch podría incluso manejar un paquete IP, pero un Hub no entiende siquiera de esto. Dependiendo de si nos encontramos un Hub o un Switch, el esquema cambia ligeramente, el funcionamiento difiere.

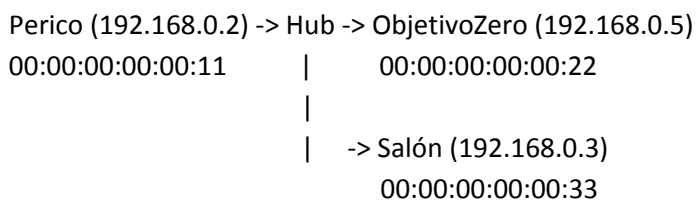
-Hub

Baratos, de muchas entradas, lentos, muy inseguros... Antes de la explosión de los routers caseros y las redes domésticas, los Hubs dominaban el mercado. Era simple... los Switch eran muy caros.

Un Hub tan solo opera en el nivel Físico del modelo OSI, esto es algo importantísimo!! Es decir, un Hub lo único que sabe hacer es reenviar la información recibida por una de las "bocas" a todas las demás. Nada más.

Esto acarrea muchos problemas de seguridad, rendimiento... Para empezar, cada dato que se quiera enviar de un equipo a otro, consume un ancho de banda igual al frame completo enviado por el número de equipos conectados en dicho momento. Un Hub comparte el medio, imaginar algo así como un cable interno que conecta todos los Pcs entre sí. Esto produce que la velocidad máxima de los Hub sea de tan solo 10Mb/s, y que ni siquiera sea Full-Duplex, es decir, que no es capaz de enviar y recibir datos al mismo tiempo:

El esquema sería algo así:



Perico antes de empezar la comunicación necesita conocer la MAC del destino. Para ello antes de comenzar cualquier tipo de comunicación verificará si esa MAC la conoce o no. Como no la conoce, tiene que enviar previamente un paquete por el protocolo ARP a toda la red. Recordar que hay direcciones IP y direcciones MAC denominadas de Broadcast (Multidifusión), que quieren decir que es un "mensaje" que se envía a todos los que se encuentran en la misma red. Dado que Perico no conoce la dirección MAC de ObjetivoZero, tampoco podría enviar el paquete ARP a él. Perico por tanto debe de hacer una petición ARP en Broadcast, sería algo así como:

¿Que MAC tiene el host 192.168.0.5? Responder a 192.168.0.2 (00:00:00:00:00:11) -> Enviado a 192.168.0.5 pero MAC FF:FF:FF:FF:FF:FF (Dirección Broadcast MAC)

Dado que estamos en un Hub, el mensaje será enviado como hemos dicho a TODAS las bocas de este. Todos los equipos recibirán el paquete. Los equipos comprobarán la MAC a la cual está dirigido, y como es una MAC de tipo Broadcast, será analizado por todos. Tanto "ObjetivoZero" como "Salón" atenderán el mensaje. "Salon" leerá el mensaje, y dado que él no es 192.168.0.5, no contestará nada. En cambio "ObjetivoZero" que si es 192.168.0.5 enviará una solicitud ARP de vuelta con el siguiente mensaje:

192.168.0.5 tiene la MAC 00:00:00:00:00:22 -> Enviado a 192.168.0.2 (00:00:00:00:00:11)

De nuevo este mensaje sería colocado en la red, el Hub lo reenviaría a todas las bocas, y tan solo 192.168.0.2 (00:00:00:00:00:11) lo atendería. Perico, al recibir la contestación por parte de "ObjetivoZero" ya estaría en condiciones para comunicarse con él, y ahora todos los consiguientes mensajes tendría como destinatario 192.168.0.5 (00:00:00:00:00:22). El mensaje sería recibido por todos los equipos, pero tan solo "ObjetivoZero" lo escucharía, dado que la MAC destino es la suya, el se hace cargo del mensaje.

Para quien sea capaz de ir siguiéndolo todo, comprobará que esto es una mina de inseguridad y bajo rendimiento.

-Switch

Los Switch funcionan de un modo muy diferente. Los Switch actuarían como si fueran nodos de red compuestos entre un Puente de red con funciones de redirecciones de paquetes más o menos, operando en los niveles OSI 1, 2 y 3 normalmente, aunque existen Switch que pueden operar en niveles más altos.

A diferencia de los Hub, que podríamos verlos como un mismo cable que une todos los equipos, un Switch, haciendo honor a su nombre, Interruptor, funcionaría de un modo similar. El Switch conectaría en cada momento los equipos que se están comunicando, el Switch no reenvía el mismo paquete a todos sus nodos, sino exclusivamente a aquel al que va destinado. Esto evidentemente tiene un ahorro de ancho de banda increíble!! Y además la velocidad de la red no depende en absoluto del número de nodos conectados. Las comunicaciones son Full-Duplex incluso entre diferentes equipos.

Con la expansión de las redes domésticas y los routers, a día de hoy prácticamente todo son Switchs. Ahora mismo son bastante baratos, aunque normalmente los usuarios no son conscientes siquiera de que poseen uno. Todos los routers domésticos que tenéis, que poseen normalmente 4 salidas de red... eso es un Switch. Los routers domésticos suelen incluir un Switch (integrado en el mismo router claro está) para que podamos de forma simple conectar hasta 4 equipos. Un router WIFI de ADSL de telefónica multipuerto, no es más que un conjunto de 4 dispositivos en 1: El router, El punto de Acceso que otorga la función WIFI, El Switch que permite la conexión de múltiples equipos (incluido el punto de acceso) y el Modem que recibe la entrada de nuestra línea y se encarga de la modulación de la señal. Como comprenderéis, es más cómodo tener un solo dispositivo que 4. Para un usuario doméstico esto suele ser más que suficiente.

El esquema es igual al que hemos explicado en el Hub, con algunos matices. Cuando un paquete ARP se envía a una dirección Broadcast este se continuará enviando a todas las "bocas" del Switch, pero en cambio cuando la dirección MAC no es de Broadcast, este mensaje tan solo será enviado por el Switch a su concreto destinatario. ¿Cómo es esto posible? Los Switch tienen y mantienen una caché, una lista de las direcciones MAC que están conectadas a él mismo. Cuando un mensaje les llega, este mira en su caché:

a) No se encuentra destinatario. Guardo/actualizo en mi caché la MAC e IP del emisor y rechazo el mensaje.

b) Encuentro destinatario. Guardo/actualizo en mi caché la MAC e IP del emisor y envío el mensaje al destinatario.

Esto podría parecer que produce un bloqueo, dado que si se parte de cero, sería necesario que antes de poderse establecer una comunicación, el equipo destino tendría q haber iniciado una comunicación inicial, porque si no, el Switch no tendría almacenada su MAC e IP. Esto no es así, por el mismo protocolo ARP, que es necesario cuando el equipo origen siquiera conoce el destino. El equipo origen enviaría el mensaje a Broadcast, el Switch añade a su cache en este momento la MAC e IP del origen. Como es un mensaje Broadcast lo emite a todas sus "bocas". Cuando el destino contestase la petición ARP, envía la respuesta al Switch. Este añade la MAC e IP del destino en su caché (tabla interna), mira el destinatario del paquete, que curiosamente es el que comenzó la comunicación, mira en su lista y dado que encuentra la MAC e IP en su lista, sabe donde enviar el paquete de vuelta, aun cuando este ya no es de Broadcast. Una vez el origen conoce la MAC del destino, puede comunicarse directamente con él, y dado que el Switch ya tiene la MAC e IP del destino almacenadas, la comunicación se realizará perfectamente.

Yo diría que esto es un sistema eficiente.

-Puntos de Acceso WIFI

Su funcionamiento es similar al de los Switch en tanto y cuando un mensaje se envía y recibe exclusivamente a un equipo, atendiendo su MAC. La principal diferencia es que un punto de acceso tan solo puede realizar una comunicación, no permite comunicación simultánea con diferentes nodos. Estos usan un sistema llamado CSMA/CA, usado para evitar el bloqueo de una transmisión en curso cuando más de un sistema quiere acceder al medio.

Imaginar un portátil que quiere enviar un dato a internet. Por otro lado encendemos el iPod Touch y nos ponemos a navegar. Aparentemente ambos dispositivos funcionan a la vez, pero esto no es así. En un bajo nivel, lo que veríamos sería lo siguiente:

a) El portátil desea enviar un dato al AP (punto de acceso). Primero tiene que comprobar el medio para verificar que está libre. Como está libre, envía una serie de señales al AP para establecer la conexión y decirle que va a transmitir y comienza la transmisión.

b) Pongamos que en el momento de la transmisión, el iPod Touch quiere navegar. Comprueba el medio y ve que este está siendo utilizado!! En ese momento, toma un tiempo aleatorio X dependiendo de lo que se crea que pueda durar la comunicación. Pasado ese tiempo X verificará de nuevo la disponibilidad del medio. Si ya está libre, se colocará él, si no lo está, a volver a esperar. Y esto es así ya sean 2 dispositivos los que quieren acceder ya sean 100.

Esto claramente tiene un inconveniente más que claro... el ancho de banda de un AP es proporcional al número de dispositivos wifi conectados. Es decir, si tenemos 10 estaciones

WIFI usando WIFI, la velocidad será dividida entre 10, sin contar que la comunicación no es Full-Duplex.

En cuanto a seguridad, es afectada por los mismos principios que los Switch, con el añadido de que al ser un medio al que todos tienen accesos, es imprescindible el empleo de encriptaciones, para evitar que cualquiera pueda ver nuestros datos.

-Sniffer

La traducción literal sería algo así como "husmeador", pero sinceramente dudo que alguien llame esto así. Un Sniffer no es más que un programa informático que es capaz de analizar, mostrar, recibir... todo el tráfico de red que es recibido por nuestro equipo. Dependiendo de si nuestra red está gobernada por un Hub o un Switch, obtendremos resultados muy dispares.

Un sniffer recibe todos los paquetes que llegan y salen de nuestro equipo. ¿Qué pasa si estamos en una red gobernada por un Hub?

Hemos dicho que un Hub reenvía la información siempre a todas sus bocas, luego en realidad nuestro equipo recibiría siempre todos los datos que envían todos los demás equipos!! En realidad un Sniffer así como así no sería capaz de verlos, dado que normalmente estos paquetes se descartan por no coincidir con nuestra MAC. Por ello existe el llamado modo Promiscuo. El modo promiscuo no es más que decirle a nuestro adaptador de red o a nuestro Sniffer que no se haga el sordo de los paquetes que no van destinados a él, que los queremos todos.

Luego el primer problema de seguridad está claro en un Hub... nuestro equipo es capaz de "husmear" TODO el tráfico de la red.

¿Entonces somos inmunes con un Switch? Tampoco, pero estamos más protegidos. Un Switch efectivamente tan solo envía el mensaje a su correcto destinatario. ¿Pero como sabe el Switch quien es el destinatario? Ya lo hemos dicho, porque tiene unas tablas que se van actualizando, eliminando, añadiendo... gracias al protocolo ARP, y de este modo el Switch puede conocer la IP y MAC de cada una de sus "bocas". Pero esa actualización/eliminación se hace por el protocolo ARP... ¿Qué pasaría si envío un paquete ARP fraudulento al Switch haciéndome pasar por otro usuario?

Recordar el ejemplo anterior:

¿Que MAC tiene el host 192.168.0.5? Responder a 192.168.0.2 (00:00:00:00:00:11) -> Enviado a 192.168.0.5 pero MAC FF:FF:FF:FF:FF:FF (Dirección Broadcast MAC)

Ese sería el envío de un mensaje ARP para conocer la dirección MAC de 192.168.0.5. Pero imaginar que en vez de hacer dicha petición, hago esta otra:

¿Que MAC tiene el host 192.168.0.5? Responder a 192.168.0.1 (00:00:00:00:00:11) -> Enviado a 192.168.0.5 pero MAC FF:FF:FF:FF:FF:FF (Dirección Broadcast MAC)

Fijaos que estoy enviando un paquete ARP a 192.168.0.5 pero haciéndome pasar por la IP de la puerta de enlace, conservando mi MAC por supuesto. Cuando a "ObjetivoZero" (192.168.0.5) llega nuestra petición, este contesta a 00:00:00:00:00:11, pero a partir de ahora para "ObjetivoZero" nosotros seremos la IP 192.168.0.1, con lo que cuando quiera enviar un paquete a Internet, lo enviará a la MAC 00:00:00:00:00:11, es decir, a nosotros.

De este modo hemos engañado al objetivo, hemos "Suplantado" la identidad del router. Las suplantaciones se denominan como técnica de **Spoofing** o **Decoy**. Si suplantamos a otro equipo normalmente se denomina Decoy, si estamos modificando la información que nos identifica, sin necesidad de hacernos pasar por otro equipo (dado que puede ser un equipo que no existe) se habla de Spoofing, aunque normalmente tan solo se habla de Spoofing, y se deja el término de Decoy para suplantaciones IP a niveles de Internet.

Para el que sea rápido, verá que esto tiene un problema... podemos capturar entonces con este ejemplo el tráfico del objetivo a Internet, pero no es útil. Primero porque no recibimos el tráfico del router al objetivo, y segundo porque si lo dejamos todo tal cual, ninguna petición del objetivo será atendida por el router.

Lo primero se soluciona con otro "**Envenenamiento ARP**" (es el nombre de esta técnica), en este caso no hacia el "ObjetivoZero", sino contra el router, modificando su caché ARP para hacernos pasar a nosotros por "ObjetivoZero". Es decir, para el Router seremos ObjetivoZero y Perico, y para el ObjetivoZero seremos el router.

El segundo problema se soluciona haciendo que nuestro equipo pueda reenrutar los paquetes que recibe del router hacia nuestro objetivo y de nuestro objetivo al router. Y con esto se concluye de forma relativamente simple un ataque de envenenamiento de ARP, haciendo de este modo posible Sniffar una red gobernada con un Switch.

Es evidente que es más complejo, que se requiere el empleo de técnicas que pueden ser evitadas con firewalls u otros monitores ARP, o con el uso de tablas ARP estáticas... pero lo cierto es que prácticamente cualquier LAN doméstica es vulnerable al 100% por esto.

¿Qué sentido tiene esto? Pues bien fácil... podemos conocer en todo momento todo lo que está haciendo cualquier usuario en su equipo por internet: Conversaciones Messenger, Correos electrónicos, páginas web visitadas, búsquedas... todo lo que ese usuario envíe o reciba de la red lo recibiremos nosotros. Esto es peligrosísimo!! Cuentas de usuario, contraseñas, números de teléfono, cuentas bancarias... Y es tan solo el ejemplo de una técnica de muchas que existen, aunque es cierto que muchas de ellas se basan en los mismos principios.

-Contramedidas

Cuando nos encontramos en una LAN, lo primero a tener en cuenta es siempre si tenemos acceso WIFI o no. Si no tenemos dispositivos WIFI, aunque nuestro router permita WIFI, deshabilitarlo. Si disponemos de enlaces WIFI usar siempre sistema de autenticación WPA2-PSK (en caso de particulares) y WPA2 en caso de empresas. WEP no es una alternativa en modo alguno, y WPA no es recomendable.

En segundo lugar y al margen de WIFI, tener siempre presente en que red nos encontramos. Es la red de casa? es la red del trabajo? es la red de la universidad? Si no tenemos WIFI activado, es la red de casa y tenemos plena confianza con nuestros hermanos/familiares, esto no debería ser un problema. Si tenemos WIFI sí es un problema enorme dado que podrían acceder si este no está correctamente configurado, y un cualquiera podría automáticamente controlar todo el tráfico de nuestra red. Si estamos en una red más amplia el problema se dispara... en tu casa puedes medio controlar quien tiene acceso a la red, en una red como la LAN de la universidad o en la del trabajo o en cualquier lugar público, no sabes jamás quien va a estar al otro lado. Es posible que no seas consciente de nada y todo lo que estás haciendo por Internet se está registrando en el PC de otro: Conversaciones, correos, documentos enviados, contraseñas, nombres de usuario... todo.

Al igual que la norma para el malware era simple: "No ejecutar lo que desconoces", la regla en las redes, sobre todo públicas, es igual de simple: Todo lo que sea información sensible, es OBLIGADO que viaje encriptado. ¿Qué podemos identificar como información sensible? pues esto ya es diferente para cada cual. Voy a poner algunos ejemplos de buenas prácticas:

-> Páginas Web donde es necesaria una autenticación: Acceso a Web Mail, Acceso a foros, chats, Facebook, Tuenti, compras...

Es muy simple. Si la web no usa certificados para proteger estos datos, no acceder desde dichas redes. Como sabemos esto? Muy fácil, las web no empiezan por http:// sino por https:// amén de aparecer un candado, el cual si pulsamos nos dirá la información del certificado.

Muchos sitios web tan solo encriptan o protegen la autenticación, dejando sin encriptar la web una vez se ha accedido, esto es el caso por ejemplo de Tuenti, Hotmail... lo cual es un problema de seguridad muy muy grande!! A lo mejor no pueden ver tu contraseña, pero si tus correos!! ¿Que para que sirve esto? Pues se me están ocurriendo al menos 4 ideas para poder aprovecharme de ello para obtener la contraseña de una cuenta de Hotmail. Imaginar que envío a MS un mensaje de restauración de clave por correo. Estos me envían el correo a la bandeja de entrada de mi objetivo. Como sé cómo funciona Hotmail, sé q cuando el usuario inicie sesión, su contraseña no la verá, pero sí sus correos, es más, estos se descargan aun sin abrirse!!. Luego tan solo con que acceda a su correo puedo tener acceso al correo enviado por MS de restauración de contraseña. Tan solo tengo que capturar el enlace que me servirá para restaurar la contraseña y listo. Esto es tan solo un ejemplo, evidentemente, lo que podríamos llamar un "proof of concept". La situación ideal por ejemplo es Gmail, desde que estamos en la pantalla de autenticación hasta que salimos de Gmail, TODO, está encriptado y protegido.

Todo lo que sea navegar que pueda incluir datos sensibles, SIEMPRE!!! Encriptación, si no hay https nada. OJO!! Es posible falsear un certificado digital, haciéndose pasar por otra entidad. Es decir, si accedemos a Gmail y nos aparece un cartel diciendo que el certificado no ha sido verificado... Cuidado!!! Puede ser realmente que el certificado de Google haya caducado y no pasa nada, o puede ser que alguien esté intentando inyectar un certificado maligno. Para los usuarios de Firefox esto es casi imposible de suceder, dado que Firefox te lanza una pantalla

bastante intimidatoria cuando el certificado no es válido, y ante este caso lo más probable es que aunque el certificado sea genuino, el usuario no lo acepte... ni siquiera sepa aceptarlo. Un ejemplo de este tipo de Webs:

<https://www4.sears.com/>

El mensaje es claro.

-> Correos electrónicos

Podemos acceder normalmente por Web (cosa discutida anteriormente) o por protocolos como POP/IMAP y SMTP. Tanto POP como IMAP y SMTP, y otros muchos soportan encriptación TLS/SSL (la misma que la que usan las web). Si enviamos un correo desde una red potencialmente incorrecta desde un gestor de correo la norma es la misma, asegurarnos de que nuestro servidor POP/IMAP/SMTP usa autenticación/cifrado SSL/TLS, de lo contrario nuestros datos estarían viajando sin protección alguna.

-> Mensajería instantánea

Esto sí es un problema, dado que la mayoría de clientes no soportan ningún tipo de encriptación, con lo que todas las conversaciones son enviadas sin protección alguna. Ante esto, es simple... si tenemos que mantener cualquier tipo de conversación por mensajería instantánea (esto incluye Messenger, Mensajes por Tuenti, Chat de Tuenti...) no hablar nada que pueda ser información sensible, puesto que puede ser interceptada.

Para el caso de administradores de sistemas, las protecciones son mucho más numerosas. Una solución integral sería por ejemplo aplicar IPsec a toda la red, de modo que absolutamente todo el tráfico que viaja por ella va encriptado. Otras soluciones pueden ser monitores ARP, la creación de tablas ARP estáticas, Firewall para filtrar posibles ataques...

Todo esto puede parecer a veces como conocimientos tan solo que pueden aprovecharlos unos pocos... pero esto que explico es tremendamente fácil de ponerlo en práctica, por ello no es una cuestión de tomarse a broma cuando se navega usando redes que no conocemos o las cuales no controlamos.

Conclusión

Después de todo esto, me conformaría con que cualquiera que haya sido capaz de llegar al final comprenda que la 1ª amenaza de sus problemas es él mismo y sus malas prácticas. Por culpa de esas malas prácticas existe la cantidad de malware que existe, existen tantas cuentas de correos hackeadas, tantos problemas. A fin de cuentas es más fácil echar la culpa al sistema operativo que es el que coge el malware que no a uno mismo por ser un inconsciente. Que estas tecnologías que nos hacen la vida más amena estén al alcance de todos hoy por hoy, no implica que no se deba de saber cómo utilizar. Si das un mal uso a tu equipo, este te lo devolverá con problemas.

¿De qué sirve el mayor sistema de seguridad jamás inventado para una caja fuerte combinacional de 1000 cifras si el código para abrirla está pegado en un posít junto a ella?