

## Test de Intrusion

Estos documentos han sido escritos y publicados por:

**Chema Alonso**, MVP de Windows Security y escribe diariamente en su blog de ["Un Informático en el lado del mal"](http://UnInformaticoenelLadoDelMal.com).

Chema trabaja en [Informática 64](http://Informatica64.com) y escribe en los blogs [Un Informático en el lado del mal](http://UnInformaticoenelLadoDelMal.com) y [vista-tecnica](http://vista-tecnica.com)

**Recopilación:** Cristian Borghello, Director de [www.segu-info.com.ar](http://www.segu-info.com.ar)

## Test de Intrusion (I de VI)

<http://elladodelmal.blogspot.com/2007/02/test-de-intrusion-i-de-vi.html>

\*\*\*\*\*

Artículo publicado en PCWorld (Enero 2007)

\*\*\*\*\*

Es guay, que te paguen por romper cristales es mejor a que te paguen por arreglarlos. Este es un chiste que he hecho mil veces porque realmente lo siento así. Los tests de Intrusión o test de penetración (pentests) son divertidos, muy divertidos diría yo. En este artículo quería hablaros de los pasos para llevar a cabo un test de penetración en una empresa. Aunque a grandes rasgos los tests de penetración son todos similares es cierto que cada uno tiene su arte y su punto fuerte, y eso es lo que otorga el valor añadido a las distintas empresas que nos dedicamos a esto.

### Auditoría de caja Negra

Los tests de penetración no llegan a ser una auditoría ya que estas deben ser exhaustivas en extensión y profundidad, es decir, deben evaluar todos los riesgos, todos los "caminos" y evaluar el nivel de criticidad de cada uno de ellos. Un pentest es encontrar un camino para resolver el sudoku. Lo cierto es que cuando se realiza un test de intrusión al final sale casi todo, incluido, lo que es más importante, las malas prácticas de configuración, implantación o desarrollo, pero hay que dejar clara la diferencia entre una auditoría y un test de intrusión.

Cuando se realiza una auditoría de seguridad estas serán distintas si se realizan desde dentro de la red con una cuenta poco privilegiada, desde Internet sin ninguna credencial o desde dentro con los privilegios del propio administrador. Se deben realizar todas, no es que una sea mejor que otra, todas se complementan y dan distinta información. Las que se realizan desde fuera y sin ninguna credencial se llaman auditorías de caja negra, mientras que las que se realizan desde dentro se llaman auditorías de caja blanca.

Podríamos decir que un test de intrusión es una auditoría de caja negra y la diferencia será si se buscan todos los caminos o solo se busca justificar la necesidad de una auditoría de seguridad en profundidad. Sí, aunque parezca extraño en muchas compañías a día de hoy aun hay que justificar la necesidad de una auditoría de seguridad. Porque... ¿a quien le importa si mañana en la Web ponen un graffiti o si los datos de la base de datos han sido cambiados parcialmente durante los últimos tres meses y no podemos recuperar ninguna copia de bases de datos porque no tenemos garantía de que ninguna sea fiable?

**¡Agazápate! Comienza la fiesta**

En primer lugar debes elegir el punto de ejecución del test, como ya hemos visto antes, a lo mejor queremos realizar una auditoría simulando que somos un usuario externo o tal vez, pensemos que el enemigo es un cliente remoto o pueda ser un trabajador interno. ¿Quién sabe? ¿O no? Elegimos el punto de ejecución y empieza la campaña. Para ello empezamos por la fases de "combate".

### **Elección de objetivos**

Hay que buscar donde puede haber una puerta para entrar dentro y hacer...iya se verá! Aun es pronto, no nos ponemos objetivos, se va conquistando poco a poco el enemigo. Para ello vamos a analizar los activos de la empresa expuestos a nuestro punto de ejecución. Es decir, desde donde estamos que servicios y/o servidores están en nuestra linea de tiro. Servidores web, servidores de conexiones VPN, el servidor de correo, de ficheros, el dns, etc.... Inventariamos los activos a testear y empezamos la segunda fase.

### **Recogida de Información**

¿Qué nos interesa? TODO. Toda información que se pueda conseguir es útil, siento decir esto pero hasta la información sobre las personas que operan y o trabajan en la empresa directa o indirectamente con los sistemas es útil. Para ello realizamos dos batidas de recogida de información distintas utilizando dos filosofías diferentes. En primer lugar vamos a recoger toda la información que sea pública de la empresa y de los objetivos. Es pública, así que recojámosla y aprendamos todo lo que podamos de ellos. Las herramientas que utilizamos son herramientas de Footprinting o de seguimiento de rastros de huellas.

### **Footprinting**

Las herramientas que vamos a utilizar son sencillas públicas y "legales" es decir, aún no necesitaríamos un documento de Exoneración de Responsabilidades que nos autorice a realizar el test, con lo que si quieres probar puedes hacerlo con quien desees solo por practicar.

Para conocer la información de la empresa empezamos por consultar al servidor DNS para ver que servidores tiene registrados con que servicios. Normalmente vamos a sacar las Direcciones IP de los servidores DNS, de los servidores Web, de los servidores de correo y ... todo lo que se pueda. Para hacer esta parte yo uso el propio Nslookup que viene en el sistema y con sencillos comandos podemos sacar toda esta información.

### **Nslookup**

Cuando entramos en el interfaz de comandos de nslookup estamos realizando consultas directamente contra el servidor de DNS que tenemos configurado en nuestra máquina, así que lo primero es averiguar cual es el servidor de dns de nuestro objetivo y preguntarle a él. Para ello elegimos el tipo de registro que queremos consultar con el comando set type. Para sacar los servidores dns: set type=ns; Para los hosts: set type=a; para los intercambiadores de correo: set type=mx, etc....

Una vez elegido el tipo se realiza la petición de resolución con el dominio que se quiere consultar y nos devuelve las Direcciones IP de los servidores DNS primario y secundario. Así que cuando sepamos cuales son, configuramos a esos servidores como los receptores de nuestras consultas con el comando: Server IP y una vez que estemos allí sacamos toda la información pública que tengamos. Si el DNS está mal configurado nos permitirá dos cosas que nunca se deben permitir, la

transferencia de zonas y el listado de todos los registros. Para ello basta conectarse al servidor de dns y desde nslookup realizar una sencilla prueba con el comando `ls nombre_de_dominio`. Si cuela, nos volcará toda la información de la zona por pantalla. Y preguntando al registro SOA de la Zona DNS podremos saber cual es el correo del responsable del dns y lo más probable webmaster.

De la información que extraigamos de aquí podremos saber cosas como si los servidores están en hosting, housing o los tiene la compañía, si tienen servicios de respaldo externos, si el administrador es cuidadoso y detallista o no. Pensad que el proceso de ataque puede depender de estos pequeños detalles. Imaginemos un servicio web de una empresa que está en hosting, si compramos un dominio en el mismo proveedor tendremos acceso al mismo servidor de nuestro objetivo por poco más de 100 pavos al año y atacar a un compañero de hosting ofrece la posibilidad de un nuevo camino digno de explorar.

En este ejemplo con la empresa, elegida al azar, t2v.com nos encontramos con que los servidores los tienen ellos en propiedad, se puede hacer "ls" del dominio y permite ver la lista de todos los servidores que tiene y la IP interna de un servidor muy, muy significativo, que se llama bd (Base de Datos?) con un direccionamiento 192.168.1.1. Curioso.



```

C:\WINDOWS\system32\cmd.exe - nslookup
> nslookup
Server: predetermined: artenis.ttd.net
Address: 194.179.1.101

> set type=ns
Server: artenis.ttd.net
Address: 194.179.1.101

Respuesta no autoritativa:
t2v.com nameserver = pikasso.t2v.com
t2v.com nameserver = casiopea.t2v.com

pikasso.t2v.com internet address = 82.144.10.66
casiopea.t2v.com internet address = 217.11.114.130
> server 82.144.10.66
Server: predetermined: pikasso.t2v.com
Address: 82.144.10.66
Aliases: 66.10.144.82.in-addr.arpa

> set type=all
Server: pikasso.t2v.com
Address: 82.144.10.66
Aliases: 66.10.144.82.in-addr.arpa

t2v.com internet address = 82.144.10.66
t2v.com
primary name server = casiopea.t2v.com
responsible mail addr = jian.t2v.com
serial = 2086122901
refresh = 432000 (5 days)
retry = 7200 (2 hours)
expire = 2592000 (30 days)
default TTL = 172800 (2 days)
t2v.com nameserver = casiopea.t2v.com
t2v.com nameserver = pikasso.t2v.com
t2v.com MX preference = 5, mail exchanger = pikasso.t2v.com
t2v.com MX preference = 10, mail exchanger = casiopea.t2v.com
pikasso.t2v.com internet address = 82.144.10.66
casiopea.t2v.com internet address = 217.11.114.130
> ls t2v.com
[pikasso.t2v.com]
t2v.com. A 82.144.10.66
t2v.com. NS server = pikasso.t2v.com
t2v.com. NS server = casiopea.t2v.com
bd A 192.168.1.1
beni A 82.144.10.92
casiopea A 217.11.114.130
correo A 82.144.10.66
correo1 A 217.11.114.130
correo2 A 82.144.10.66
curiosodeverano A 82.144.10.86
desastres A 82.144.10.84
desastres A 217.11.114.140
euroleague A 82.144.10.93
focus A 217.11.114.153
ftpclientes A 82.144.10.68

```

Imagen 1: Nslookup

## Tracea y Posiciona

Una vez que se tienen los objetivos iniciales marcados con direcciones IPs lo siguiente es situarlos en la red y geográficamente, puede darnos alguna información curiosa. Herramientas como tracert o Visualroute nos va a permitir averiguar cual es la ubicación física y quienes son sus proveedores de acceso a Internet.

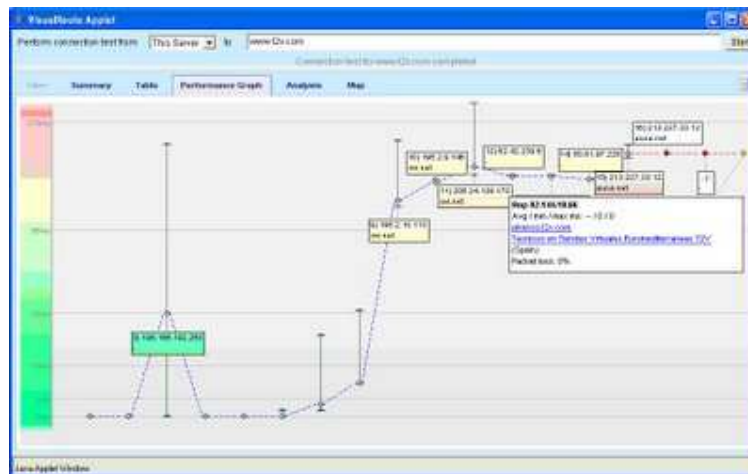


Imagen 2: Tracert visual de <http://visualroute.visualware.com>

En el apreciamos las redes que le conectan, en este caso con usa, pero podemos ver quien es su proveedor de servicios de Internet. Supercable (que daba servicio a Andalucía), luego AUNA, y ahora ONO. Telcos, je.

## Whois

Cuando una empresa registra un dominio en Internet debe rellenar una serie de datos en el registrador que deben estar en una base de datos de información que se llama Whois. La información que se registra en esta base de datos puede protegerse parcialmente hablando con el registrador, pero por defecto toda ella es pública. Cada registrador ofrece unas herramientas de acceso a la base de datos whois para que cualquiera pueda consultarla.



Imagen 3: Información que nos ofrece sobre el dominio [www.t2v.com](http://www.t2v.com) a través del acceso a la base de datos whois de <http://www.chatox.com>

## Arañas de Internet

Usa la información que ya han recogido las arañas y aprende a sacarle partido a las bases de datos sobre las sitios que tienen nuestros amigos los buscadores. Existe una base de datos que se llama Google Hacking Database (GHD) que tiene catalogadas en distintas categorías cadenas de búsqueda para usar en google para sacar información para hacking de empresas a través de las bases de datos del buscador google. Hay una sección que me encanta se llama "Passwords" y la explicación que dan en la propia base de datos sobre esa categoría es: "FOR THE LOVE OF GOD, GOOGLE FINDS PASSWORDS". Aprende a rebuscar en google/msn search para seguir los pasos de la empresa y de los administradores, te sorprendería lo que puede hacer un administrador de una empresa con sus correos corporativos.



Imagen 4: Google Hacking Database (GHDB) <http://johnny.ihackstuff.com>

## Test de Intrusion (II de VI)

<http://elladodelmal.blogspot.com/2007/03/test-de-intrusion-ii-de-vi.html>

\*\*\*\*\*

Artículo publicado en PCWorld (Febrero 2007)

\*\*\*\*\*

### Spidering

Las técnicas de Spidering se utilizan para poder encontrar toda la información que se nos ofrece gratuitamente a través de los sitios web de la compañía. Nos ofrece páginas html, aplicativos, ficheros de imágenes, documentos, javascripts, applets, etc... Cuando realizamos un pentest hay que sacar toda la información que se pueda extraer de comentarios, meta información de archivos gráficos o documentos, etc... Por ejemplo los archivos doc de Microsoft Office no suelen ser limpiados de la información de sus creadores antes de ser publicados y generalmente nos van a dar rutas de archivos en máquinas locales y nombres de cuentas de los usuarios en los dominios. Sólo por poner un ejemplo. Además, cuando se está realizando una descarga completa de un sitio se busca también aquello que, aparentemente no está publico, pero que es predecible, como los directorios típicos /cgi-bin, /bin, /images, /admin, /privado, etc... y los archivos predecibles, como archivos de ejemplo, configuración etc... Para ello se utilizan herramientas que se llaman Scanners de Cgi y que, aunque en origen rastreaban programas cgi vulnerables, hoy en día se usan como rastreadores de ficheros en técnicas de spidering.

Ejemplo 1: Documento office descargado de la Universidad de las Illes Balears con información del usuario y la ruta del sistema de ficheros de su creador.



Imagen 5: Usuario tsiralb. <http://www.uib.es/recerca/osr/AIsolicitud.doc>

¿Os suena cierto escándalo en el gobierno británico con cierto documento sobre las armas de destrucción masiva en Irak que había sido modificado?



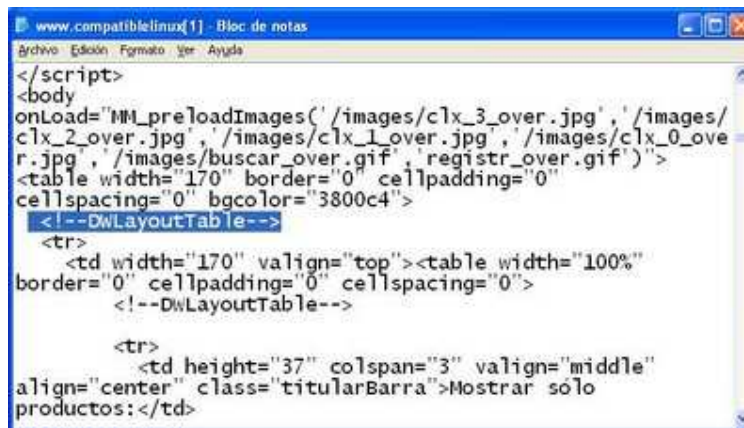


Imagen 6: Comentario en una página web que delata que ha sido desarrollada con Dreamweaver. ¿Hay Dreamweaver para Linux?

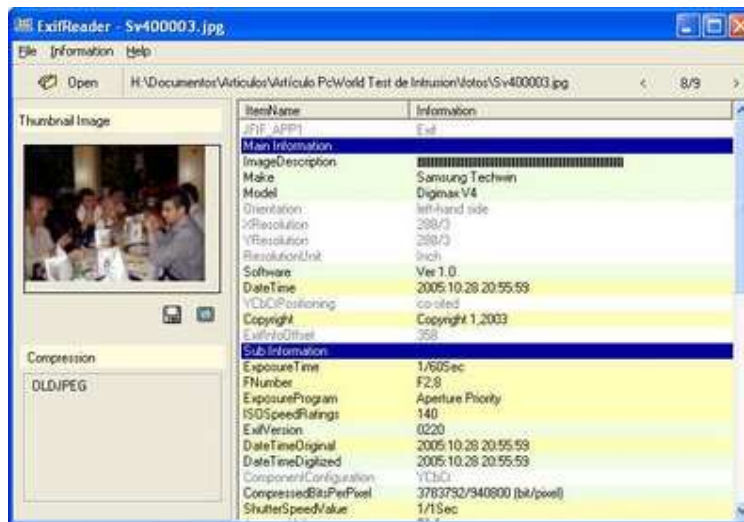


Imagen 7: Foto descargada desde Internet con meta información sobre el creador.

Es una foto de Arturo descargada desde un directorio que hay por aquí en la misma universidad de antes. [[http://www.uib.es/depart/dqu/fotos\\_arturo/](http://www.uib.es/depart/dqu/fotos_arturo/)]. Ya sabemos que día fue la fiesta, la camarita e incluso se puede averiguar la hora del día a la que se tiró, que como se aprecia era hora de cena.

Herramientas como Teleport Pro te permiten descargarte todos los ficheros que se ofrecen públicamente en un sitio Web, scanners de cgi como [voideye](#), whisker, cgi-scan o cualquier otro permiten cargar una lista de ficheros y directorios a buscar. Como se puede ver en la imagen, a voideye se le puede configurar el uso de servidores Proxy anónimos para evitar la detección de quien está realizando el escaneo.

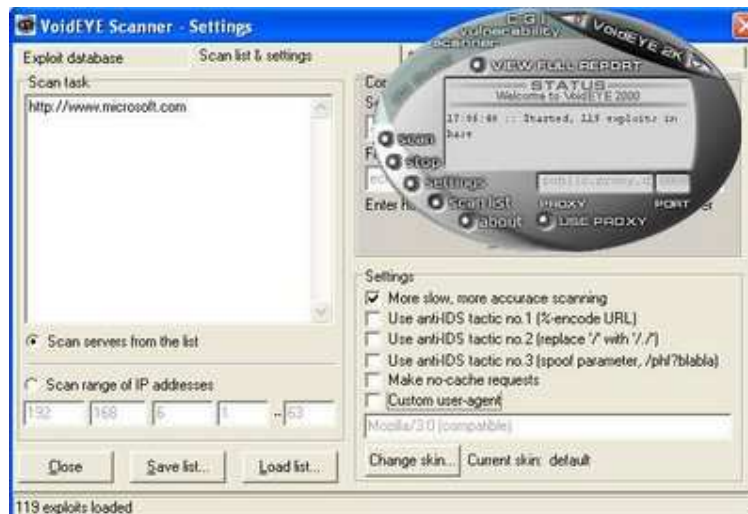


Imagen 8: Voideye

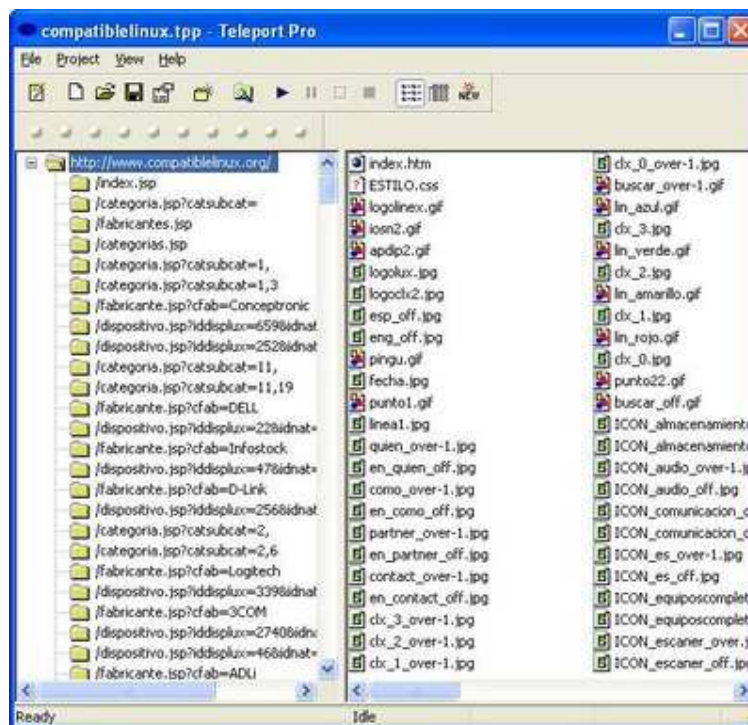


Imagen 9: Teleport Pro

## Fingerprinting

Ya hemos recogido toda la información que era pública, ahora vamos a recoger aquella que también está accesible públicamente pero que a priori no se puede ver. Es decir, vamos a inferir información a partir de pruebas que vamos a ir realizando a cada uno de los servicios y o servidores. Fácil, ¿no?

## Identificación de Sistemas Operativos y/o Firewalls

Para identificar los sistemas operativos que corren sobre los servidores o debajo de los servicios que queremos auditar existen diferentes herramientas. Todas estas herramientas se basan en jugar al ¿Quién es Quién? ¿Recordáis aquel juego en el



que se iba preguntando al contrario si tenía bigote o sombrero? Pues es la misma idea. Averiguar el sistema operativo consiste en enviarle distintas petición e ir analizando las respuestas. Por ejemplo, pensemos en un simple ping. Si enviamos en las tramas TPC/IP valores 1 en los bits de urgencia, que no van a ser utilizados por el sistema operativo, y observamos los resultados obtenidos se puede inferir que sistema es. Un sistema Microsoft devuelve esos valores siempre a 0, eso es porque cuando construye una trama TCP/IP de respuesta lo hace desde una nueva trama; sin embargo, un sistema Linux construye la respuesta a partir de la trama de petición. Esto hace que si enviamos 1 y recibimos 0 podamos inferir, con un porcentaje alto de éxito, que el sistema operativo es un Microsoft Windows. Lógicamente el número de pruebas que se pueden hacer son muchas y las diferencias pueden existir entre un Microsoft Windows NT 4 Server y un Microsoft Windows NT 4 Server Service Pack 6. A veces se puede afinar y saber exactamente la versión del kernel de un sistema Linux y otras no. Para ello vamos a utilizar distintas herramientas que ya nos ayudan con un amplio conjunto de diferenciación de sistemas operativos.

### **Nmap**

Es un scanner de información sobre servidores, y, aunque su principal utilidad es la de un scanner de puertos tiene opciones de reconocedor de sistemas operativos. La opción -O (jé,je) También puedes hacer simplemente clic en el interfaz gráfico, si usas un frontend.(;). El escaneo de sistemas operativos que realiza nmap no solo se queda en Windows, Linux, etc.. sino es capaz de detectar impresoras de red, centralitas PBX, o teléfonos IP. Para realizar esta detección cuenta con una enorme base de datos para hacer una detección entre miles, repito miles, de sistemas operativos (Windows Server NT, Windows NT Workstation SP4, Windows NT SP6, todos los sabores de Linux, etc...) Además, en las nuevas versiones realiza pruebas sobre los puertos TCP y UDP descubiertos y es capaz de sacar conclusiones del tipo Nokia Checkpoint redirigiendo a IIS 6 sobre un Windows Server 2003.

### **Hping2 (o hping3)**

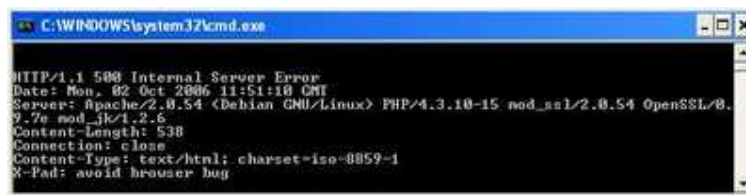
Esta utilidad [[hping](#)] se puede comparar con un ping pero a nivel TCP/IP, es decir, podemos, mediante comandos definir la conversación completa con un servidor con un tipo de paquetes enviados. La utilidad nos permite enviar paquetes con los flags que queramos activados, modificando todos y cada uno de los campos de una mensaje TCP/IP: MTU, TOS, Bit de Urgencia, flags de SYN, etc... Esta herramienta va a permitir averiguar el sistema operativo a base de realizar pruebas. Por suerte nmap implementa casi todas las técnicas conocidas para el reconocimiento de sistemas operativos, pero si alguna no estuviera implementada se puede utilizar hping2 para implementarla. Además de para reconocer el sistema operativo también vamos a poder utilizarla para averiguar los algoritmos de generación de números de secuencia (útiles en escaneos de puertos y técnicas de IP-Spoofing), la configuración de un firewall, e incluso la transmisión de ficheros encapsulándolos sobre los paquetes que configuremos.

### **Los puertos del sistema**

El siguiente paso, que en un escaneo rápido se realiza al mismo tiempo, es la detección de puertos abiertos en un sistema. A priori que un puerto esté abierto no es ni malo ni bueno. Un puerto se abre para dar servicio. El problema es cuando se abre un puerto que conecta con una aplicación o servicio que no está securizado o que realmente el administrador ignora que lo tiene abierto en una máquina. Desde el punto de vista de un test de intrusión hay que ver a que servicios se puede llegar.

Detectar un puerto TCP abierto debería ser tan fácil como establecer una conexión

con dicho servicio; para ello bastaría con establecer una conexión con un simple telnet, que, además, nos mostrará la información por pantalla que el servicio o aplicación al otro lado del puerto está enviando.



*Imagen 10: Telnet al puerto 80*

El problema radica en las alarmas que se activen o la información que se deje en los logs cuando se produce el scanear, hay que tener en cuenta que el número de puertos posibles es de 65536 puertos que pueden ser TCP o UDP.

Lo más probable es que un firewall deje de responder a una determinada IP cuando se haya producido más de un número de peticiones de conexión desde una determinada ip, o cuando se han pedido más de un número por unidad de tiempo, o cuando hay una violación de protocolo (comando no valido que suele denotar que un servicio está hablando con un ser humano detrás de un telnet y que ha cometido un error al teclear o en el comando), etc....

El objetivo de los métodos de escaneo es averiguar todos los puertos que se encuentran ofreciendo servicio por TCP/UDP sin levantar alarmas y dejar el menor rastro posible.

Hay muchos tipos de scanear, y me encantaría explicarlos todos aquí, pero no tengo espacio, pero si os los voy a enumerar para que podáis buscar más información sobre ellos. Al final la idea consiste en enviar paquetes con una determinada configuración y escuchar como reacciona el equipo escaneado. Un escaneo no tiene porque ser perfecto y a veces no salen todos los puertos si está detrás de un firewall en un único tipo de escaneo. Los más usados son el escaneo de [SYN](#), de [FIN](#), de [SYN+ACK](#), de [ACK](#), el escaneo de [NULL](#), que se realiza con todos los flags apagados y el de [XMAS Tree](#), que recibe ese nombre porque activa todas las banderas como si fueran luces en un arbolito de navidad [Y para [UDP](#) i [Idle Scanning](#)].

### Scaneo con Nmap

[Nmap](#), realiza el escaneo de establecimiento de conexiones (como si hiciéramos un telnet) que se llama TCP connect(), de SYN (o de conexiones medio abiertas, ya que se envía un paquete SYN a un puerto y se espera la confirmación por parte del cliente sin nunca cerrar la conexión), de FIN, de NULL y de XMAS Tree.

Para los puertos UDP implementa un sistema basado en el envío de mensajes UDP a un puerto de 0 bytes de tamaño. Si se recibe un mensaje ICMP que dice que es inalcanzable entonces es que el puerto está cerrado, por el contrario se asume que está abierto. Este tipo de escaneo, como muchos de los anteriores puede tener un alto índice de falsos positivos debido a que puede que el firewall prohíba todo tráfico ICMP de salida o que se haya perdido la respuesta.

## Test de Intrusion (III de VI)

<http://elladodelmal.blogspot.com/2007/03/test-de-intrusin-iii-de-vi.html>

\*\*\*\*\*

Artículo publicado en PCWorld (Marzo 2007)

\*\*\*\*\*

El mes pasado vimos como funcionaban las técnicas de footprinting y fingerprinting y vimos algunas herramientas como la google hacking database, los traceares, nmap o hping2 (ya está disponible hping3). Ahora vamos a centrarnos en la búsqueda de vulnerabilidades.

### Identificación de Servicios y Software

Una vez que sabemos cuales son los sistemas operativos, firewalls y/o puertos abiertos es necesario descubrir las versiones de software que corren por esos servicios. Lo primero es intentar ver la información que se nos ofrece abiertamente. Para ello, suele bastar con realizar una conexión y recoger el banner del mismo servicio. El mes pasado vimos como contestaba un servidor web, pero esto se puede realizar con servicios FTP, SMTP o Listeners de Bases de Datos.

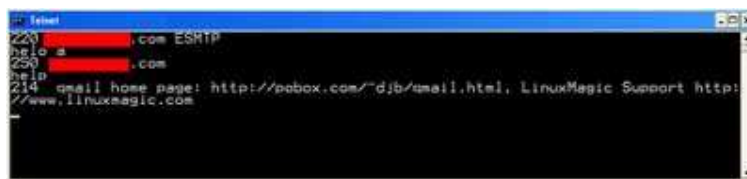


Imagen: Telnet a SMTP

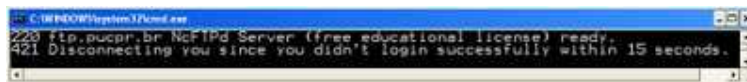


Imagen: Telnet a FTP

En el caso de no poder afinar con el banner, con la inferencia de cruzar el sistema operativo, el puerto utilizado, etc..., para identificar la versión, tendremos que utilizar herramientas de fingerprinting de servicio. La mayoría de los escanners de vulnerabilidades (que vamos a ver en el tercer artículo) implementan estas técnicas para la mayoría de servicios. Una vez acotado el software, hay que buscarle los problemas.

### Bug

El bug o fallo de seguridad es una característica de un software que puede hacer que el programa funcione incorrectamente o de manera no pensada. Muchos hackers definen bug como "funcionalidad no definida de un programa". Entendemos como Software Fiable aquel que hace lo que tiene que hacer y como Software Seguro aquel que hace lo que tiene que hacer y nada más. Ese algo más entre el software fiable y el software seguro son los bugs.

El que un software no tenga bugs es muy complicado de conseguir, hay que tener en cuenta que un programa escrito en lenguaje C, compilado con dos compiladores distintos no genera el mismo código binario, luego un mismo código puede ser o no vulnerable dependiendo de compiladores, arquitecturas donde se ejecute, linkadores, etc..

En la búsqueda de bugs se utilizan dos aproximaciones distintas, que pueden realizarse manualmente y con ojos expertos, pero que se automatizan y que son las herramientas de análisis estático de código y análisis dinámico.

### Herramientas de Análisis Estático de Código

Este tipo de herramientas mantienen una base de datos de patrones reconocidos como fallos de seguridad, como copia de parámetros con strcpy sin comprobar tamaños, etc... y lo que realizan es una búsquedas de esos parámetros en el código de un programa sin que este se esté ejecutando. Esta forma de escanear el código es lo que recibe el nombre de análisis estático. Estas herramientas pueden analizar códigos en ensamblador, desensamblados directamente del binario, o códigos fuente en lenguajes madre con .NET, C, php, C++, etc... o en códigos intermedios como Bytecodes o IL. La diferencia cualitativa entre estas herramientas es la base de datos de patrones que utilizan y la mayoría de las grandes casas de desarrollo de software consideran esto como un conocimiento competitivo y de valor de la compañía. Microsoft, desde el año 2002, utiliza herramientas propias de análisis de código estático en todos sus productos. En el mundo del software libre, a raíz del proyecto "bug hunting" financiado por el gobierno americano, utiliza a la compañía Coverity para realizar análisis estático de bugs en los principales proyectos de software libre desde Enero de 2006. Los informes con los primeros resultados de los análisis de Coverity están disponibles en <http://scan.coverity.com>

Existen múltiples herramientas de Análisis Estático de Código disponibles en la web, e incluso, muchos compiladores acompañan su entorno de desarrollo con herramientas de este tipo, como por ejemplo FxCop en Visual Studio que comprueba desbordamientos de buffer, condiciones de carrera, etc...

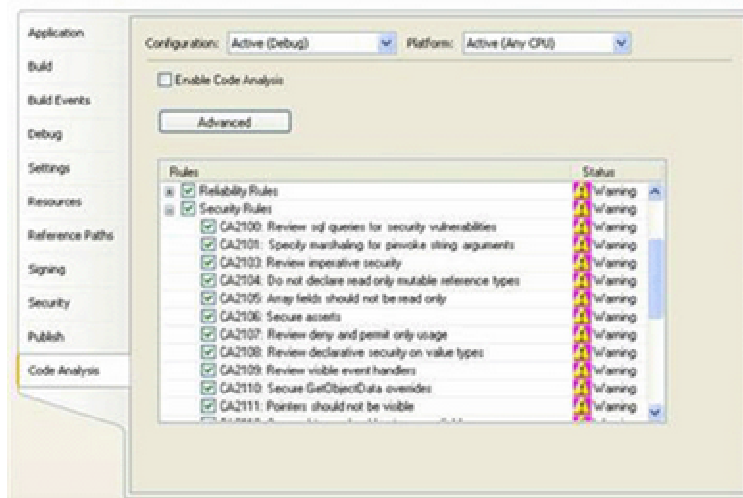


Imagen: FxCop en Visual Studio

### Proyecto Bug Hunting

En Enero del año 2006 comenzó, a instancias de una universidad y el gobierno Americano el proyecto Bug Hunting, cuyo objetivo era analizar los 50 proyectos Open Source más populares con las herramientas de Análisis Estático de Código de la empresa Coverity. Los resultados que aparecieron en Marzo de 2006 (un año para cuando salga publicado el artículo) están disponibles en la siguiente URL: <http://scan.coverity.com>. Se puede ver en la tabla, algunos de los resultados que se obtuvieron después de utilizar las herramientas de análisis estático.

Project Name	Fixed Defects*	Defect Report Summary**		Lines of Code	Defect Reports / KLOC	View Results	Please Register to View Results
		Outstanding Verified	Uninspected and Pending				
apache-httpd	2	4	25	133,919	0.217	<a href="#">Sign in</a>	<a href="#">Register</a>
CUWIR	9	0	158	224,614	0.703	<a href="#">Sign in</a>	<a href="#">Register</a>
Firebird	0	0	197	270,917	0.727	<a href="#">Sign in</a>	<a href="#">Register</a>
Firefox	352	65	168	1,855,717	0.126	<a href="#">Sign in</a>	<a href="#">Register</a>
FreeBSD	0	6	605	1,582,166	0.386	<a href="#">Sign in</a>	<a href="#">Register</a>
Gaim	197	1	19	217,906	0.092	<a href="#">Sign in</a>	<a href="#">Register</a>
GDB	0	0	267	436,375	0.612	<a href="#">Sign in</a>	<a href="#">Register</a>
glibc	83	0	1	582,329	0.002	<a href="#">Sign in</a>	<a href="#">Register</a>
Gnome	349	12	48	706,511	0.085	<a href="#">Sign in</a>	<a href="#">Register</a>
Icecast	11	0	8	37,640	0.213	<a href="#">Sign in</a>	<a href="#">Register</a>
Inatutils	5	3	21	73,836	0.325	<a href="#">Sign in</a>	<a href="#">Register</a>
KDE	1253	10	40	4,619,029	0.011	<a href="#">Sign in</a>	<a href="#">Register</a>
Linux-HA	39	1	0	211,209	0.005	<a href="#">Sign in</a>	<a href="#">Register</a>
LVM2	26	3	5	57,688	0.139	<a href="#">Sign in</a>	<a href="#">Register</a>
Mono	69	1	70	334,195	0.212	<a href="#">Sign in</a>	<a href="#">Register</a>
monotone	2	0	26	299,482	0.087	<a href="#">Sign in</a>	<a href="#">Register</a>
MPlayer	58	16	142	498,455	0.317	<a href="#">Sign in</a>	<a href="#">Register</a>
Net-SNMP	85	0	84	263,227	0.319	<a href="#">Sign in</a>	<a href="#">Register</a>
NetBSD	1267	196	1350	4,717,818	0.328	<a href="#">Sign in</a>	<a href="#">Register</a>
OpenMotif	16	0	273	509,952	0.535	<a href="#">Sign in</a>	<a href="#">Register</a>
OpenSSL	5	0	40	218,244	0.183	<a href="#">Sign in</a>	<a href="#">Register</a>
OpenVPN	0	1	0	69,971	0.014	<a href="#">Sign in</a>	<a href="#">Register</a>
Perl	47	1	7	511,792	0.016	<a href="#">Sign in</a>	<a href="#">Register</a>
PHP	75	0	1	467,757	0.002	<a href="#">Sign in</a>	<a href="#">Register</a>
PostgreSQL	53	0	23	846,884	0.027	<a href="#">Sign in</a>	<a href="#">Register</a>

Imagen: Resultados Coverity

## Herramientas de Análisis dinámico

La búsqueda de bugs con herramientas de análisis dinámico tiene otra aproximación distinta. En este caso el programa a evaluar se ejecuta y se le pasan pruebas y verificaciones automáticas que van a evaluar las respuestas ante todo tipo de situaciones distintas. Una herramienta de este tipo debe ser capaz de evaluar todas las posibilidades de todos los puntos de entrada de información desde cualquier punto externo hacia la aplicación y detectar los casos anómalos.

Algunos ejemplos de herramientas de este tipo son Valgrind, pensada para detectar condiciones de carrera en entornos multihilo y errores de memoria, Dmalloc.h, que es una librería que se usa para comprobar las reservas de memoria o VB Watch que inyecta códigos de análisis dinámico dentro de los programas para monitorizar su funcionamiento.

## Exploits

Una vez que se han encontrado los bugs, el objetivo es crear una herramienta que saque partido de ellos, es decir, que sea capaz de hacer uso de esa "funcionalidad no definida del programa". Para ello, se analizan las posibilidades y alcance de ese bug y se opta por un determinado exploit. Todos los exploits tienen dos partes diferenciadas, la cabecera, que es lo que se denomina exploit puramente, y el cuerpo, denominado payload. La cabecera es la parte artesana que depende de cada bug en concreto, mientras que el payload son acciones ya programadas que se reutilizan. Por ejemplo, una acción típica de un exploit sería devolver una shell de comandos de la máquina explotada a un determinado puerto. Este payload se va a poder reutilizar para múltiples cabeceras distintas.

## Metasploit Framework



Esta herramienta es una herramienta básica en la tarea de realizar un test de intrusión en una compañía. Es un entorno que aúna una base de datos de cabeceras de exploits y de payloads para poder realizar el test en unos determinados servidores. Actualmente en la versión 3.0 en beta 3 tiene un actualizador automático de las bases de datos y herramienta de administración web.

```

[*] Checking metasploit.com for updates...
+ -- --[ nsfupdate v2.5 [revision 1.43]
[*] Calculating local file checksums, please wait...
[*] Version 2.7 of the Metasploit Framework is now available.
    - http://metasploit.com/projects/Framework/downloads.html
[*] Online Update Task Summary
    Update: ./exploits/Credits.pm
    Update: ./exploits/edirectory_inonor2.pm
    Update: ./exploits/freeftpd_key_exchange.pm
    Update: ./exploits/freesshd_key_exchange.pm
    Update: ./exploits/ie_iscomponentinstalled.pm
    Update: ./exploits/realnc_41_bypass.pm
    Update: ./exploits/tftpd32_long_filename.pm
    Update: ./nsfupdate
Continue? (yes or no) > yes
[*] Starting online update of 8 file(s)...
[0001/0008 - 0x00295e bytes] ./exploits/Credits.pm
[0002/0008 - 0x000cf4 bytes] ./exploits/edirectory_inonor2.pm
[0003/0008 - 0x000c0b bytes] ./exploits/freeftpd_key_exchange.pm
[0004/0008 - 0x000bcd bytes] ./exploits/freesshd_key_exchange.pm
[0005/0008 - 0x0014a3 bytes] ./exploits/ie_iscomponentinstalled.pm
[0006/0008 - 0x001334 bytes] ./exploits/realnc_41_bypass.pm
[0007/0008 - 0x000a69 bytes] ./exploits/tftpd32_long_filename.pm
[0008/0008 - 0x004ee5 bytes] ./nsfupdate
[*] Regenerating local file database
  
```

Imagen: Metasploit Update

En la siguiente imagen se puede ver como, mediante el interfaz web, podemos configurar los diferentes payloads para un exploit, en este caso, un exploit de Samba para Linux.

EXPLOITS	PAYLOADS	SESSIONS
Samba trans2open Overflow		
Select Payload:		
Payload	Description	
bsd_ia32_bind	BSD IA32 Bind Shell	
bsd_ia32_bind_stg	BSD IA32 Staged Bind Shell	
bsd_ia32_exec	BSD IA32 Execute Command	
bsd_ia32_findrecv	BSD IA32 Recv Tag Findsock Shell	
bsd_ia32_findrecv_stg	BSD IA32 Staged Findsock Shell	
bsd_ia32_findsock	BSD IA32 SrcPort Findsock Shell	
bsd_ia32_reverse	BSD IA32 Reverse Shell	
bsd_ia32_reverse_stg	BSD IA32 Staged Reverse Shell	
linux_ia32_adduser	Linux IA32 Add User	
linux_ia32_bind	Linux IA32 Bind Shell	
linux_ia32_bind_stg	Linux IA32 Staged Bind Shell	
linux_ia32_exec	Linux IA32 Execute Command	
linux_ia32_findrecv	Linux IA32 Recv Tag Findsock Shell	
linux_ia32_findrecv_stg	Linux IA32 Staged Findsock Shell	
linux_ia32_findsock	Linux IA32 SrcPort Findsock Shell	
linux_ia32_reverse	Linux IA32 Reverse Shell	
linux_ia32_reverse_impurity	Linux IA32 Reverse Impurity Upload/Execute	
linux_ia32_reverse_stg	Linux IA32 Staged Reverse Shell	
linux_ia32_reverse_udp	Linux IA32 Reverse UDP Shell	

Imagen: Configuración de Payload para un Exploit de SAMBA en Linux

## 0 days

El término "0-days" se va a repetir mucho en las auditorías de seguridad:

- Un servidor en 0-days es aquel que tiene actualizado todo su software a las



últimas versiones, es decir, que el software no tiene ningún bug conocido. Ese es el punto más seguro en que se puede tener el software de un servidor.

- Un exploit de 0-days es aquel que funciona en servidores 0-day, es decir, que actualmente no existe una versión de software más moderna para solucionar ese problema.

El objetivo de una auditoría de seguridad será obtener un servidor de 0-days y, si existiesen exploits de 0-days, tomar las medidas para que el software vulnerable no se vea expuesto. Es decir, bloqueando peticiones en firewalls, fortificando las restricciones en las máquinas o con políticas de seguridad en las directivas de seguridad de la red.

## POC y Los Meses temáticos

Cuando se quiere realizar un completo test de intrusión en un servidor no valen solo las herramientas comerciales y las que nos ofrecen los fabricantes ya que hay mucha información que no sale por los canales comunes. Las llamadas Pruebas de Concepto (POC) actualmente son muchas de pago y es normal que se sepa que existe un exploit pero que no sea de uso público. Durante el año pasado H.D. Moore, uno de los principales contribuyentes al proyecto Metasploit, decidió realizar meses temáticos junto a otros investigadores de seguridad, orientados a buscar exploits 0-days en diferentes áreas. Así, durante el mes de Julio se centró en los navegadores de Internet, durante el mes de Noviembre en los kernels de los sistemas operativos, durante el mes de Enero en Apple y ya, Stefan Esser ha anunciado para el mes de marzo, el mes de los fallos en PHP. Habrá que estar atentos. En estos proyectos "temáticos" el objetivo es buscar un bug al día y hacerlo público. Esto, lógicamente, creo mucha controversia ya que los canales de información tradicionales son los de avisar al fabricante para que lo corrija sin que los clientes queden expuestos. Esta controversia llevó, a que se anunciara para la última semana del año 2006 la Semana de los fallos de Oracle Database, por parte de Cesar Cerrudo, un investigador de seguridad centrado en Bases de datos, pero, debido a la presión pública la cancelara.

Las POC pueden encontrarse en muchos sitios Web dedicados a la seguridad informática: FRSirt (<http://www.frsirt.com>), Milworm (<http://www.milworm.com>) o Packet Storm (<http://packetstormsecurity.org/>) son algunos de los más utilizados.

[\[ home \]](#)
[\[ contents \]](#)
[\[ platforms \]](#)
[\[ shellcode \]](#)
[\[ search \]](#)
[\[ cracker \]](#)
[\[ index \]](#)
[\[ res \]](#)
[\[ forum \]](#)
[\[ archive \]](#)

MILWORM

[\[ remote \]](#)

DATE	DESCRIPTION	CVSS	TYPE	AUTHOR
2007-02-08	SAP Web Application Server 6.40 Arbitrary File Disclosure Exploit	5.718	R	Stefan
2007-02-08	MySQL 4.1.12-4.1.14-4.1.15-4.1.16-4.1.17-4.1.18-4.1.19-4.1.20-4.1.21-4.1.22-4.1.23-4.1.24-4.1.25-4.1.26-4.1.27-4.1.28-4.1.29-4.1.30-4.1.31-4.1.32-4.1.33-4.1.34-4.1.35-4.1.36-4.1.37-4.1.38-4.1.39-4.1.40-4.1.41-4.1.42-4.1.43-4.1.44-4.1.45-4.1.46-4.1.47-4.1.48-4.1.49-4.1.50-4.1.51-4.1.52-4.1.53-4.1.54-4.1.55-4.1.56-4.1.57-4.1.58-4.1.59-4.1.60-4.1.61-4.1.62-4.1.63-4.1.64-4.1.65-4.1.66-4.1.67-4.1.68-4.1.69-4.1.70-4.1.71-4.1.72-4.1.73-4.1.74-4.1.75-4.1.76-4.1.77-4.1.78-4.1.79-4.1.80-4.1.81-4.1.82-4.1.83-4.1.84-4.1.85-4.1.86-4.1.87-4.1.88-4.1.89-4.1.90-4.1.91-4.1.92-4.1.93-4.1.94-4.1.95-4.1.96-4.1.97-4.1.98-4.1.99-4.1.100-4.1.101-4.1.102-4.1.103-4.1.104-4.1.105-4.1.106-4.1.107-4.1.108-4.1.109-4.1.110-4.1.111-4.1.112-4.1.113-4.1.114-4.1.115-4.1.116-4.1.117-4.1.118-4.1.119-4.1.120-4.1.121-4.1.122-4.1.123-4.1.124-4.1.125-4.1.126-4.1.127-4.1.128-4.1.129-4.1.130-4.1.131-4.1.132-4.1.133-4.1.134-4.1.135-4.1.136-4.1.137-4.1.138-4.1.139-4.1.140-4.1.141-4.1.142-4.1.143-4.1.144-4.1.145-4.1.146-4.1.147-4.1.148-4.1.149-4.1.150-4.1.151-4.1.152-4.1.153-4.1.154-4.1.155-4.1.156-4.1.157-4.1.158-4.1.159-4.1.160-4.1.161-4.1.162-4.1.163-4.1.164-4.1.165-4.1.166-4.1.167-4.1.168-4.1.169-4.1.170-4.1.171-4.1.172-4.1.173-4.1.174-4.1.175-4.1.176-4.1.177-4.1.178-4.1.179-4.1.180-4.1.181-4.1.182-4.1.183-4.1.184-4.1.185-4.1.186-4.1.187-4.1.188-4.1.189-4.1.190-4.1.191-4.1.192-4.1.193-4.1.194-4.1.195-4.1.196-4.1.197-4.1.198-4.1.199-4.1.200-4.1.201-4.1.202-4.1.203-4.1.204-4.1.205-4.1.206-4.1.207-4.1.208-4.1.209-4.1.210-4.1.211-4.1.212-4.1.213-4.1.214-4.1.215-4.1.216-4.1.217-4.1.218-4.1.219-4.1.220-4.1.221-4.1.222-4.1.223-4.1.224-4.1.225-4.1.226-4.1.227-4.1.228-4.1.229-4.1.230-4.1.231-4.1.232-4.1.233-4.1.234-4.1.235-4.1.236-4.1.237-4.1.238-4.1.239-4.1.240-4.1.241-4.1.242-4.1.243-4.1.244-4.1.245-4.1.246-4.1.247-4.1.248-4.1.249-4.1.250-4.1.251-4.1.252-4.1.253-4.1.254-4.1.255-4.1.256-4.1.257-4.1.258-4.1.259-4.1.260-4.1.261-4.1.262-4.1.263-4.1.264-4.1.265-4.1.266-4.1.267-4.1.268-4.1.269-4.1.270-4.1.271-4.1.272-4.1.273-4.1.274-4.1.275-4.1.276-4.1.277-4.1.278-4.1.279-4.1.280-4.1.281-4.1.282-4.1.283-4.1.284-4.1.285-4.1.286-4.1.287-4.1.288-4.1.289-4.1.290-4.1.291-4.1.292-4.1.293-4.1.294-4.1.295-4.1.296-4.1.297-4.1.298-4.1.299-4.1.300-4.1.301-4.1.302-4.1.303-4.1.304-4.1.305-4.1.306-4.1.307-4.1.308-4.1.309-4.1.310-4.1.311-4.1.312-4.1.313-4.1.314-4.1.315-4.1.316-4.1.317-4.1.318-4.1.319-4.1.320-4.1.321-4.1.322-4.1.323-4.1.324-4.1.325-4.1.326-4.1.327-4.1.328-4.1.329-4.1.330-4.1.331-4.1.332-4.1.333-4.1.334-4.1.335-4.1.336-4.1.337-4.1.338-4.1.339-4.1.340-4.1.341-4.1.342-4.1.343-4.1.344-4.1.345-4.1.346-4.1.347-4.1.348-4.1.349-4.1.350-4.1.351-4.1.352-4.1.353-4.1.354-4.1.355-4.1.356-4.1.357-4.1.358-4.1.359-4.1.360-4.1.361-4.1.362-4.1.363-4.1.364-4.1.365-4.1.366-4.1.367-4.1.368-4.1.369-4.1.370-4.1.371-4.1.372-4.1.373-4.1.374-4.1.375-4.1.376-4.1.377-4.1.378-4.1.379-4.1.380-4.1.381-4.1.382-4.1.383-4.1.384-4.1.385-4.1.386-4.1.387-4.1.388-4.1.389-4.1.390-4.1.391-4.1.392-4.1.393-4.1.394-4.1.395-4.1.396-4.1.397-4.1.398-4.1.399-4.1.400-4.1.401-4.1.402-4.1.403-4.1.404-4.1.405-4.1.406-4.1.407-4.1.408-4.1.409-4.1.410-4.1.411-4.1.412-4.1.413-4.1.414-4.1.415-4.1.416-4.1.417-4.1.418-4.1.419-4.1.420-4.1.421-4.1.422-4.1.423-4.1.424-4.1.425-4.1.426-4.1.427-4.1.428-4.1.429-4.1.430-4.1.431-4.1.432-4.1.433-4.1.434-4.1.435-4.1.436-4.1.437-4.1.438-4.1.439-4.1.440-4.1.441-4.1.442-4.1.443-4.1.444-4.1.445-4.1.446-4.1.447-4.1.448-4.1.449-4.1.450-4.1.451-4.1.452-4.1.453-4.1.454-4.1.455-4.1.456-4.1.457-4.1.458-4.1.459-4.1.460-4.1.461-4.1.462-4.1.463-4.1.464-4.1.465-4.1.466-4.1.467-4.1.468-4.1.469-4.1.470-4.1.471-4.1.472-4.1.473-4.1.474-4.1.475-4.1.476-4.1.477-4.1.478-4.1.479-4.1.480-4.1.481-4.1.482-4.1.483-4.1.484-4.1.485-4.1.486-4.1.487-4.1.488-4.1.489-4.1.490-4.1.491-4.1.492-4.1.493-4.1.494-4.1.495-4.1.496-4.1.497-4.1.498-4.1.499-4.1.500-4.1.501-4.1.502-4.1.503-4.1.504-4.1.505-4.1.506-4.1.507-4.1.508-4.1.509-4.1.510-4.1.511-4.1.512-4.1.513-4.1.514-4.1.515-4.1.516-4.1.517-4.1.518-4.1.519-4.1.520-4.1.521-4.1.522-4.1.523-4.1.524-4.1.525-4.1.526-4.1.527-4.1.528-4.1.529-4.1.530-4.1.531-4.1.532-4.1.533-4.1.534-4.1.535-4.1.536-4.1.537-4.1.538-4.1.539-4.1.540-4.1.541-4.1.542-4.1.543-4.1.544-4.1.545-4.1.546-4.1.547-4.1.548-4.1.549-4.1.550-4.1.551-4.1.552-4.1.553-4.1.554-4.1.555-4.1.556-4.1.557-4.1.558-4.1.559-4.1.560-4.1.561-4.1.562-4.1.563-4.1.564-4.1.565-4.1.566-4.1.567-4.1.568-4.1.569-4.1.570-4.1.571-4.1.572-4.1.573-4.1.574-4.1.575-4.1.576-4.1.577-4.1.578-4.1.579-4.1.580-4.1.581-4.1.582-4.1.583-4.1.584-4.1.585-4.1.586-4.1.587-4.1.588-4.1.589-4.1.590-4.1.591-4.1.592-4.1.593-4.1.594-4.1.595-4.1.596-4.1.597-4.1.598-4.1.599-4.1.600-4.1.601-4.1.602-4.1.603-4.1.604-4.1.605-4.1.606-4.1.607-4.1.608-4.1.609-4.1.610-4.1.611-4.1.612-4.1.613-4.1.614-4.1.615-4.1.616-4.1.617-4.1.618-4.1.619-4.1.620-4.1.621-4.1.622-4.1.623-4.1.624-4.1.625-4.1.626-4.1.627-4.1.628-4.1.629-4.1.630-4.1.631-4.1.632-4.1.633-4.1.634-4.1.635-4.1.636-4.1.637-4.1.638-4.1.639-4.1.640-4.1.641-4.1.642-4.1.643-4.1.644-4.1.645-4.1.646-4.1.647-4.1.648-4.1.649-4.1.650-4.1.651-4.1.652-4.1.653-4.1.654-4.1.655-4.1.656-4.1.657-4.1.658-4.1.659-4.1.660-4.1.661-4.1.662-4.1.663-4.1.664-4.1.665-4.1.666-4.1.667-4.1.668-4.1.669-4.1.670-4.1.671-4.1.672-4.1.673-4.1.674-4.1.675-4.1.676-4.1.677-4.1.678-4.1.679-4.1.680-4.1.681-4.1.682-4.1.683-4.1.684-4.1.685-4.1.686-4.1.687-4.1.688-4.1.689-4.1.690-4.1.691-4.1.692-4.1.693-4.1.694-4.1.695-4.1.696-4.1.697-4.1.698-4.1.699-4.1.700-4.1.701-4.1.702-4.1.703-4.1.704-4.1.705-4.1.706-4.1.707-4.1.708-4.1.709-4.1.710-4.1.711-4.1.712-4.1.713-4.1.714-4.1.715-4.1.716-4.1.717-4.1.718-4.1.719-4.1.720-4.1.721-4.1.722-4.1.723-4.1.724-4.1.725-4.1.726-4.1.727-4.1.728-4.1.729-4.1.730-4.1.731-4.1.732-4.1.733-4.1.734-4.1.735-4.1.736-4.1.737-4.1.738-4.1.739-4.1.740-4.1.741-4.1.742-4.1.743-4.1.744-4.1.745-4.1.746-4.1.747-4.1.748-4.1.749-4.1.750-4.1.751-4.1.752-4.1.753-4.1.754-4.1.755-4.1.756-4.1.757-4.1.758-4.1.759-4.1.760-4.1.761-4.1.762-4.1.763-4.1.764-4.1.765-4.1.766-4.1.767-4.1.768-4.1.769-4.1.770-4.1.771-4.1.772-4.1.773-4.1.774-4.1.775-4.1.776-4.1.777-4.1.778-4.1.779-4.1.780-4.1.781-4.1.782-4.1.783-4.1.784-4.1.785-4.1.786-4.1.787-4.1.788-4.1.789-4.1.790-4.1.791-4.1.792-4.1.793-4.1.794-4.1.795-4.1.796-4.1.797-4.1.798-4.1.799-4.1.800-4.1.801-4.1.802-4.1.803-4.1.804-4.1.805-4.1.806-4.1.807-4.1.808-4.1.809-4.1.810-4.1.811-4.1.812-4.1.813-4.1.814-4.1.815-4.1.816-4.1.817-4.1.818-4.1.819-4.1.820-4.1.821-4.1.822-4.1.823-4.1.824-4.1.825-4.1.826-4.1.827-4.1.828-4.1.829-4.1.830-4.1.831-4.1.832-4.1.833-4.1.834-4.1.835-4.1.836-4.1.837-4.1.838-4.1.839-4.1.840-4.1.841-4.1.842-4.1.843-4.1.844-4.1.845-4.1.846-4.1.847-4.1.848-4.1.849-4.1.850-4.1.851-4.1.852-4.1.853-4.1.854-4.1.855-4.1.856-4.1.857-4.1.858-4.1.859-4.1.860-4.1.861-4.1.862-4.1.863-4.1.864-4.1.865-4.1.866-4.1.867-4.1.868-4.1.869-4.1.870-4.1.871-4.1.872-4.1.873-4.1.874-4.1.875-4.1.876-4.1.877-4.1.878-4.1.879-4.1.880-4.1.881-4.1.882-4.1.883-4.1.884-4.1.885-4.1.886-4.1.887-4.1.888-4.1.889-4.1.890-4.1.891-4.1.892-4.1.893-4.1.894-4.1.895-4.1.896-4.1.897-4.1.898-4.1.899-4.1.900-4.1.901-4.1.902-4.1.903-4.1.904-4.1.905-4.1.906-4.1.907-4.1.908-4.1.909-4.1.910-4.1.911-4.1.912-4.1.913-4.1.914-4.1.915-4.1.916-4.1.917-4.1.918-4.1.919-4.1.920-4.1.921-4.1.922-4.1.923-4.1.924-4.1.925-4.1.926-4.1.927-4.1.928-4.1.929-4.1.930-4.1.931-4.1.932-4.1.933-4.1.934-4.1.935-4.1.936-4.1.937-4.1.938-4.1.939-4.1.940-4.1.941-4.1.942-4.1.943-4.1.944-4.1.945-4.1.946-4.1.947-4.1.948-4.1.949-4.1.950-4.1.951-4.1.952-4.1.953-4.1.954-4.1.955-4.1.956-4.1.957-4.1.958-4.1.959-4.1.960-4.1.961-4.1.962-4.1.963-4.1.964-4.1.965-4.1.966-4.1.967-4.1.968-4.1.969-4.1.970-4.1.971-4.1.972-4.1.973-4.1.974-4.1.975-4.1.976-4.1.977-4.1.978-4.1.979-4.1.980-4.1.981-4.1.982-4.1.983-4.1.984-4.1.985-4.1.986-4.1.987-4.1.988-4.1.989-4.1.990-4.1.991-4.1.992-4.1.993-4.1.994-4.1.995-4.1.996-4.1.997-4.1.998-4.1.999-4.1.1000-4.1.1001-4.1.1002-4.1.1003-4.1.1004-4.1.1005-4.1.1006-4.1.1007-4.1.1008-4.1.1009-4.1.1010-4.1.1011-4.1.1012-4.1.1013-4.1.1014-4.1.1015-4.1.1016-4.1.1017-4.1.1018-4.1.1019-4.1.1020-4.1.1021-4.1.1022-4.1.1023-4.1.1024-4.1.1025-4.1.1026-4.1.1027-4.1.1028-4.1.1029-4.1.1030-4.1.1031-4.1.1032-4.1.1033-4.1.1034-4.1.1035-4.1.1036-4.1.1037-4.1.1038-4.1.1039-4.1.1040-4.1.1041-4.1.1042-4.1.1043-4.1.1044-4.1.1045-4.1.1046-4.1.1047-4.1.1048-4.1.1049-4.1.1050-4.1.1051-4.1.1052-4.1.1053-4.1.1054-4.1.1055-4.1.1056-4.1.1057-4.1.1058-4.1.1059-4.1.1060-4.1.1061-4.1.1062-4.1.1063-4.1.1064-4.1.1065-4.1.1066-4.1.1067-4.1.1068-4.1.1069-4.1.1070-4.1.1071-4.1.1072-4.1.1073-4.1.1074-4.1.1075-4.1.1076-4.1.1077-4.1.1078-4.1.1079-4.1.1080-4.1.1081-4.1.1082-4.1.1083-4.1.1084-4.1.1085-4.1.1086-4.1.1087-4.1.1088-4.1.1089-4.1.1090-4.1.1091-4.1.1092-4.1.1093-4.1.1094-4.1.1095-4.1.1096-4.1.1097-4.1.1098-4.1.1099-4.1.1100-4.1.1101-4.1.1102-4.1.1103-4.1.1104-4.1.1105-4.1.1106-4.1.1107-4.1.1108-4.1.1109-4.1.1110-4.1.1111-4.1.1112-4.1.1113-4.1.1114-4.1.1115-4.1.1116-4.1.1117-4.1.1118-4.1.1119-4.1.1120-4.1.1121-4.1.1122-4.1.1123-4.1.1124-4.1.1125-4.1.1126-4.1.1127-4.1.1128-4.1.1129-4.1.1130-4.1.1131-4.1.1132-4.1.1133-4.1.1134-4.1.1135-4.1.1136-4.1.1137-4.1.1138-4.1.1139-4.1.1140-4.1.1141-4.1.1142-4.1.1143-4.1.1144-4.1.1145-4.1.1146-4.1.1147-4.1.1148-4.1.1149-4.1.1150-4.1.1151-4.1.1152-4.1.1153-4.1.1154-4.1.1155-4.1.1156-4.1.1157-4.1.1158-4.1.1159-4.1.1160-4.1.1161-4.1.1162-4.1.1163-4.1.1164-4.1.1165-4.1.1166-4.1.1167-4.1.1168-4.1.1169-4.1.1170-4.1.1171-4.1.1172-4.1.1173-4.1.1174-4.1.1175-4.1.1176-4.1.1177-4.1.1178-4.1.1179-4.1.1180-4.1.1181-4.1.1182-4.1.1183-4.1.1184-4.1.1185-4.1.1186-4.1.1187-4.1.1188-4.1.1189-4.1.1190-4.1.1191-4.1.1192-4.1.1193-4.1.1194-4.1.1195-4.1.1196-4.1.1197-4.1.1198-4.1.1199-4.1.1200-4.1.1201-4.1.1202-4.1.1203-4.1.1204-4.1.1205-4.1.1206-4.1.1207-4.1.1208-4.1.1209-4.1.1210-4.1.1211-4.1.1212-4.1.1213-4.1.1214-4.1.1215-4.1.1216-4.1.1217-4.1.1218-4.1.1219-4.1.1220-4.1.1221-4.1.1222-4.1.1223-4.1.1224-4.1.1225-4.1.1226-4.1.1227-4.1.1228-4.1.1229-4.1.1230-4.1.1231-4.1.1232-4.1.1233-4.1.1234-4.1.1235-4.1.1236-4.1.1237-4.1.1238-4.1.1239-4.1.1240-4.1.1241-4.1.1242-4.1.1243-4.1.1244-4.1.1245-4.1.1246-4.1.1247-4.1.1248-4.1.1249-4.1.1250-4.1.1251-4.1.1252-4.1.1253-4.1.1254-4.1.1255-4.1.1256-4.1.1257-4.1.1258-4.1.1259-4.1.1260-4.1.1261-4.1.1262-4.1.1263-4.1.1264-4.1.1265-4.1.1266-4.1.1267-4.1.1268-4.1.1269-4.1.1270-4.1.1271-4.1.1272-4.1.1273-4.1.1274-4.1.1275-4.1.1276-4.1.1277-4.1.1278-4.1.1279-4.1.1280-4.1.1281-4.1.1282-4.1.1283-4.1.1284-4.1.1285-4.1.1286-4.1.1287-4.1.1288-4.1.1289-4.1.1290-4.1.1291-4.1.1292-4.1.1293-4.1.1294-4.1.1295-4.1.1296-4.1.1297-4.1.1298-4.1.1299-4.1.1300-4.1.1301-4.1.1302-4.1.1303-4.1.1304-4.1.1305-4.1.1306-4.1.1307-4.1.1308-4.1.1309-4.1.1310-4.1.1311-4.1.1312-4.1.1313-4.1.1314-4.1.1315-4.1.1316-4.1.1317-4.1.1318-4.1.1319-4.1.1320-4.1.1321-4.1.1322-4.1.1323-4.1.1324-4.1.1325-4.1.1326-4.1.1327-4.1.1328-4.1.1329-4.1.1330-4.1.1331-4.1.1332-4.1.1333-4.1.1334-4.1.1335-4.1.1336-4.1.1337-4.1.1338-4.1.1339-4.1.1340-4.1.1341-4.1.1342-4.1.1343-4.1.1344-4.1.1345-4.1.1346-4.1.1347-4.1.1348-4.1.1349-4.1.1350-4.1.1351-4.1.1352-4.1.1353-4.1.1354-4.1.1355-4.1.1356-4.1.1357-4.1.1358-4.1.1359-4.1.1360-4.1.1361-4.1.1362-4.1.1363-4.1.1364-4.1.1365-4.1.1366-4.1.1367-4.1.1368-4.1.1369-4.1.1370-4.1.1371-4.1.1372-4.1.1373-4.1.1374-4.1.1375-4.1.1376-4.1.1377-4.1.1378-4.1.1379-4.1.1380-4.1.1381-4.1.1382-4.1.1383-4.1.1384-4.1.1385-4.1.1386-4.1.1387-4.1.1388-4.1.1389-4.1.1390-4.1.1391-4.1.1392-4.1.1393-4.1.1394-4.1.1395-4.1.1396-4.1.1397-4.1.1398-4.1.1399-4.1.1400-4.1.1401-4.1.1402-4.1.1403-4.1.1404-4.1.1405-4.1.1406-4.1.1407-4.1.1408-4.1.1409-4.1.1410-4.1.1411-4.1.1412-4.1.1413-4.1.1414-4.1.1415-4.1.1416-4.1.1417-4.1.1418-4.1.1419-4.1.1420-4.1.1421-4.1.1422-4.1.1423-4.1.1424-4.1.1425-4.1.1426-4.1.1427-4.1.1428-4.1.1429-4.1.1430-4.1.1431-4.1.1432-4.1.1433-4.1.1434-4.1.1435-4.1.1436-4.1.1437-4.1.1438-4.1.1439-4.1.1440-4.1.1441-4.1.1442-4.1.1443-4.1.1444-4.1.1445-4.1.1446-4.1.1447-4.1.1448-4.1.1449-4.1.1450-4.1.1451-4.1.1452-4.1.1453-4.1.1454-4.1.1455-4.1.1456-4.1.1457-4.1.1458-4.1.1459-4.1.1460-4.1.1461-4.1.1462-4.1.1463-4.1.1464-4.1.1465-4.1.1466-4.1.1467-4.1.1468-4.1.1469-4.1.1470-4.1.1471-4.1.1472-4.1.1473-4.1.1474-4.1.14			

#	Title	Description	PoC/Exploit	References
31	<a href="#">"Unspecified Kernel Remote Fun"</a>	<a href="#">Pull the plug</a> , beware of evil RF.	<a href="#">Coming soon</a>	<a href="#">CVE-2007-0586</a>
30	<a href="#">Multiple Apple Software Format String Vulnerabilities</a>	Apple Help Viewer, Safari, iMovie and iPhoto are affected by multiple format string vulnerabilities, related to certain functions from AppKit that have been documented in previous releases.	Not required.	<a href="#">CVE-NO-NAME</a>
29	<a href="#">Apple iChat Bonjour Multiple Denial of Service Vulnerabilities</a>	Apple <a href="#">iChat Bonjour</a> functionality is affected by several remotely exploitable denial of service flaws which can be triggered via advertising presence services over multicast DNS.	<a href="#">MQAB-29-01-2007.rb</a>	<a href="#">CVE-NO-NAME</a>
28	<a href="#">Apple crashdump Privilege Escalation Vulnerability</a>	crashdump follows symlinks within the /Library/Logs/CrashReporter/ directory, allowing admin-group users to execute arbitrary code and overwrite files with elevated privileges. In couple with a specially crafted Mach-O binary, this can be used to write a malicious crontab entry, which will run with root privileges.	<a href="#">MQAB-28-01-2007.rb</a>	<a href="#">CVE-2007-0467</a>
27	<a href="#">Telestream Flip4Mac WMV Parsing Memory Corruption Vulnerability</a>	Flip4Mac fails to properly handle WMV files with a crafted ASF_File_Properties_Object size field, leading to an exploitable memory corruption condition, which can be abused remotely for arbitrary code execution.	<a href="#">MQAB-27-01-2007.wmv</a>	<a href="#">CVE-2007-0466</a>
26	<a href="#">Apple Installer Package Filename Format String Vulnerability</a>	Apple Installer fails to properly handle package filename strings. It's affected by a typical format string vulnerability, which can lead to a denial of service condition or arbitrary code execution.	Not necessary	<a href="#">CVE-2007-0465</a>

Imagen: Últimos días en el mes de los fallos de Apple

## Test de Intrusion (IV de VI)

<http://elladodelmal.blogspot.com/2007/04/test-de-intusin-iv-de-vi.html>

\*\*\*\*\*

Artículo publicado en PCWorld (Marzo 2007)

\*\*\*\*\*

### Expedientes de Seguridad

Cuando un bug es descubierto debe documentarse correctamente. Para ello cada fabricante de software suele mantener su propia forma de codificar los expedientes de seguridad, pero han surgido en Internet muchas empresas que se han dedicado a mantener sus propias bases de datos de expedientes de seguridad. El trabajo es arduo si pensamos en la cantidad de proyectos de software y lo rápido que se descubren nuevos bugs, con lo cual es imposible decir que una base de datos tiene todos los fallos conocidos. Lo que si está claro es que algunas se han convertido en un estándar de facto. Las dos bases de datos más utilizadas por la gente que se dedica a seguridad son Bugtraq de SecurityFocus [\[http://www.securityfocus.com/bid\]](http://www.securityfocus.com/bid) y CVE (Common Vulnerabilities and Exposures) [\[http://cve.mitre.org/cve\]](http://cve.mitre.org/cve). Ambas bases de datos codifican cada fallo con un identificador y a partir de ahí se analiza el software vulnerable, el software no vulnerable que podría ser susceptible de serlo por ser parte de la familia de productos o versiones, se discute sobre cual podría ser el alcance de dicho bug, si existe o no el exploit disponible para ese fallo y cual es la solución ofrecida por el fabricante. La ventaja de estas bases de datos, es que en cualquier momento se pueden consultar todos los expedientes de seguridad relativos a un determinado producto y versión para ver cuales son las soluciones a aplicar.

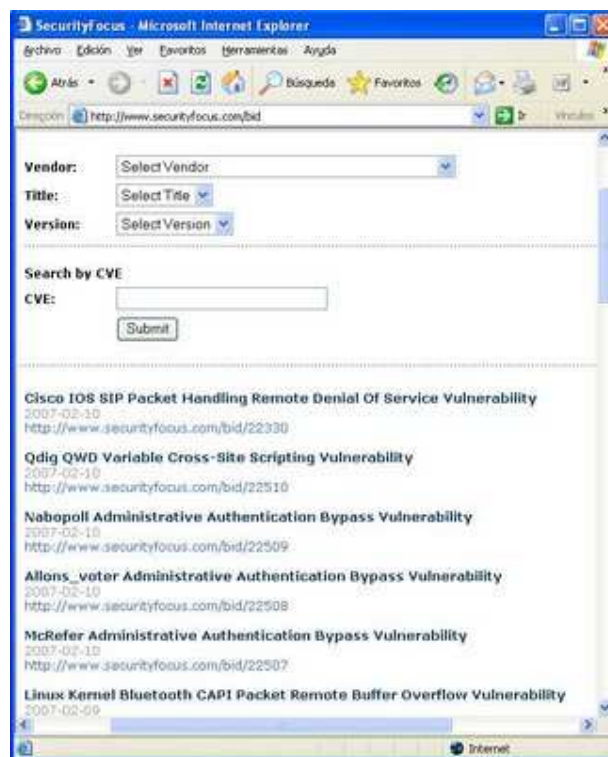


Imagen: Últimos expedientes en Bugtraq



Imagen: Expediente de Seguridad de un Bug

Como se puede ver en la imagen, los expedientes de seguridad almacenan los exploits asociados a estos fallos, luego, si el servidor tiene un fallo sin arreglar y el exploit es público nuestro servidor está en un claro riesgo, pero, aunque el exploit no sea público se deben aplicar las medidas de solución que recomiende el fabricante ya que, muchos exploits, debido a su importancia o impacto son de pago y puede que existan y estén circulando por Internet. Hay que recordar que los rumores hablan de precios desorbitados para exploits de productos populares o de alto nivel de criticidad. En la imagen se puede apreciar como security focus, relaciona su codificación con la codificación CVE.

### Scanners de Explotación

Una de las herramientas que suelen aparecer justo después de que se haya hecho público el bug y el exploit asociado son scanners que implementan motores de búsqueda de servidores vulnerables a ese fallo y un mecanismo para ejecutar dicho exploit. Están preparados para lanzarlos de forma cómoda, ya que la mayoría de las veces se requiere hacer modificaciones en las POC para que estas funcionen en diferentes entornos, es decir, se requiere cierto nivel técnico para poder utilizarlos, mientras que con estas herramientas suele ser tan fácil como aplicar botones. Algunos ejemplos de estas herramientas en las siguientes capturas:

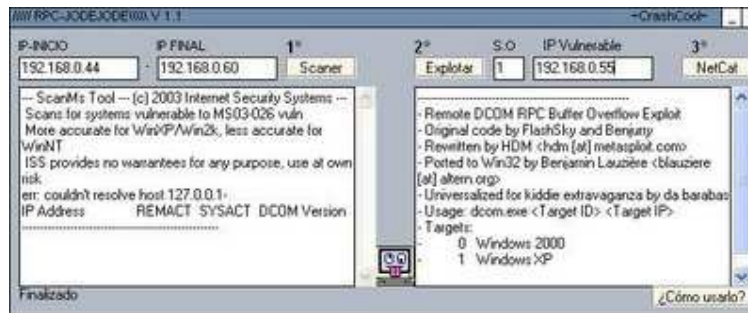


Imagen: "RPC Jode Jode". Antiguo scanner para un fallo RPC.



Imagen: WebDAV Scanner. Para detectar Servidores con WebDAV activado.



Imagen: WebDAV Exploit. Asociado al WebDAV Scanner

## Actualizaciones de Seguridad o Parches

La mayoría de las soluciones que se aplican a los fallos de seguridad del software suelen ser Actualizaciones de Seguridad o también llamados parches que nos ofrece el propio fabricante. Estos parches deben ser aplicados lo antes posible, tras haber realizado previamente las comprobaciones de que estos no afectan al funcionamiento normal de nuestra infraestructura por supuesto. La premura en la aplicación de los parches se debe a que en el momento en que se publica un parche se está acelerando la creación de los exploit. ¿Por qué? Pues porque el contar con el software sin parchear y con el software parcheado permite a los creadores de exploits realizar ingeniería inversa localizada. La idea es sencilla, se realiza una imagen del sistema sin parchear y luego otra imagen con el sistema parcheado, se comparan y se buscan las diferencias. Cuando se tienen localizados los ficheros modificados se realizan comparaciones de esos ficheros y mediante las técnicas de debugging, similares a las utilizadas para la creación cracks, se busca el punto de



fallo en el programa. Digamos que publicar un parche es marcar el sitio del problema por lo que es muy importante actualizar lo antes posible. Esto ha hecho que ahora se hable de que el segundo martes de cada mes, día que Microsoft hace públicas sus actualizaciones de seguridad sea el día del parche y que el segundo miércoles de cada mes sea el día del exploit.

## Microsoft Baseline Security Analyzer

Lo primera herramienta que se debe utilizar para realizar una auditoría de caja blanca en mi organización es MBSA, que va a permitirnros comprobar si a algún equipo o servidor de nuestra organización le falta alguna actualización de seguridad. Esta herramienta es de Microsoft y por tanto solo nos va a comprobar las actualizaciones de seguridad de los productos Microsoft, por lo que si tenemos software de otras compañías, lo primero es ver como se distribuyen, se actualiza y se comprueba si nuestros productos tienen o no todas las actualizaciones de seguridad aplicadas.



Imagen: Microsoft Baseline Security Analyzer

MBSA no es solo una herramienta para buscar parches no instalados, sino que además es lo que denominamos una herramienta de Auditoría de caja blanca, porque nos va a recoger información sobre el sistema relativa a seguridad, desde la política de contraseñas, la configuración de seguridad del servidor Web o los servicios corriendo. Es de caja blanca porque exige la utilización de las credenciales de un usuario y es lo primero que se debe realizar en cualquier auditoría de seguridad de servidores Microsoft. Más información sobre MBSA en la siguiente URL: <http://www.microsoft.com/technet/security/tools/mbsahome.mspx>

En la misma línea que MBSA, Microsoft ofrece dos herramientas más, todas gratuitas, que son *MS Exchange Best Practices Analyzer* y *MS SQL Server Best Practices Analyzer*. Estas herramientas no se quedan solo en la configuración de seguridad sino que además ayudan a ajustar los servidores para poder aplicar medias de defensa en profundidad o ajuste del rendimiento. Si vamos a analizar la seguridad e un servidor los informes ofrecidos por estas herramientas son importantes y a tener muy en cuenta.



## Test de Intusión (V de VI)

<http://elladodelmal.blogspot.com/2007/05/test-de-intusin-v-de-vi.html>

\*\*\*\*\*

Artículo Publicado en PCWorld Abril 2007

\*\*\*\*\*

Durante estos meses hemos estado recorriendo las herramientas que van desde la recogida de la información que ofrece un sistema informático hasta como funciona el ciclo de bug, exploits, parche, scanner de vulnerabilidades. Esta vez nos centramos en las herramientas principales para la realización de un test de intrusión.

### Los Escáneres de Vulnerabilidades

Este tipo de herramientas son la pieza principal cuando se va a realizar una auditoría de seguridad, tanto de caja blanca, como de caja negra. Engloban el conjunto de acciones necesarias para identificar las IPs activas, los puertos y servicios ofrecidos, la identificación del sistema operativo, el nivel de actualizaciones de seguridad aplicadas e incluso la detección y explotación de las vulnerabilidades encontradas. Dentro de los múltiples escáneres de vulnerabilidades los hay más o menos amigables, es decir, los hay que no ejecutan nunca la fase de intento de explotación de los fallos encontrados y otros que los prueban como forma normal de trabajo. Esto puede llevar a que la realización de una auditoría con un escáner poco "friendly" pueda "tumbar" algún servicio o servidor.

Si estas herramientas realizan todas las fases del proceso, ¿esto quiere decir que todas las herramientas anteriores no son necesarias? No, las herramientas anteriores son mucho más específicas en su función y permiten ser afinadas mucho más allá de lo que permitirá una escáner de vulnerabilidades completo.

Otra de las características de los escáneres de vulnerabilidades es que no son todos iguales, ni en la detección, ni en la evaluación de los riesgos, ni en la profundidad de los análisis, así que siempre es recomendable la utilización de, al menos, un par de ellos en un buen estudio.

### Satán, Santa y Nessus

Aunque previamente habían aparecido muchos escáneres de una vulnerabilidad o un exploit, quizá el primer escáner de vulnerabilidades completo que se creo fue S.A.T.A.N. (Security Administrator Tool for Analyzing Networks). Su nombre creo mucha controversia así que realizaron apareció la versión [SAINT](#) "SANTA" (Security Administrator's Integrated Network Tool) que hoy en día se comercializa.

El relevo de estos lo cogió Nessus, apareciendo en el año 1998, de la mano de Renaud Deraison, la primera versión pública del producto. En su origen y hasta el año 2005 todas las versiones han salido bajo la licencia GPL, pero a finales del año 2005 anunció que la versión 3 sería gratuita pero no GPL. Las últimas versiones de Nessus se pueden obtener de la web <http://www.nessus.org> pero pertenecen a la empresa **Tenable Network Security**, la empresa que Renaud creo en el año 2002.

### Nessus

Nessus es una de las herramientas preferidas por todos a la hora de realizar un test de penetración en una red debido a algunas de sus características.

- Actualización y funcionamiento mediante Plugins: Todos los escáneres de vulnerabilidades deben ofrecer un sistema de actualizaciones rápido para detectar las nuevas vulnerabilidades, que aparecen constantemente. En Nessus funciona mediante plugins, bajo demanda podemos actualizar la base de datos de conocimiento antes de cada ejecución para tener siempre actualizado el repositorio. Para hacer el sistema mucho más flexible en la creación de plugins y para poder ser alimentado con conocimiento propio podemos crear nuestros propios añadidos utilizando el lenguaje NASL (Nessus Attack Scripting Language).



Imagen: Detección de Plugins



Imagen: Actualización de Plugins

- Arquitectura cliente/servidor: El sistema está diseñado para funcionar desde distintos clientes contra distintos objetivos, así, Nessus corre como servicio en la máquina que desees y puede ser utilizado desde cualquier cliente. Esta arquitectura es independiente de plataforma y permite instalar, tanto los clientes como los servidores en arquitecturas Microsoft y \*NIX. Esto va a permitir hacer esfuerzos económicos para tener un mejor servidor que va a ser productivo para muchos clientes. Además, los objetivos pueden ser casi de cualquier tipo ya que su base de conocimiento detecta vulnerabilidades en Servidores, clientes y dispositivos de red corriendo Windows, \*NIX o MacOS.

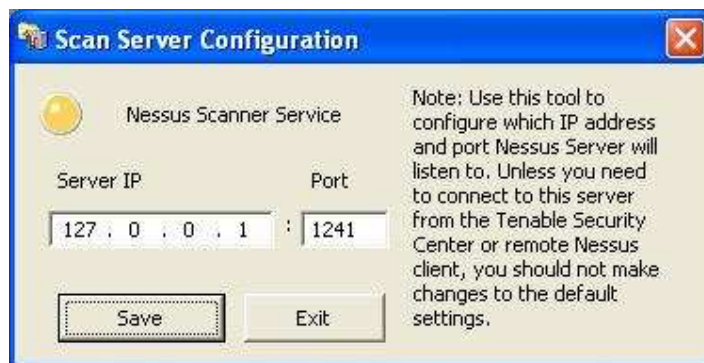


Imagen: Configuración de Servicio Servidor

- Políticas de auditoría: Es importante que cualquier escáner de vulnerabilidades permita afinar la política a aplicar, poder elegir los servicios y las vulnerabilidades a auditar (no es lo mismo auditar un servidor de correo que un servidor FTP), que se integre con los distintos protocolos de comunicaciones (algunos escáneres tienen problemas con los protocolos SSL) y que se pueda decidir si queremos un escaneo amigable o uno hasta las últimas consecuencias, es decir, que pruebe todo aun asumiendo que podemos realizar una denegación de servicio en algunos servicios o en el sistema. Para ello en Nessus, cuando creamos o editamos una política definiremos si queremos una Segura o no y después los plugings que queremos que pruebe en el proceso de auditoría.

### **GfI LanGuard Network Security Scanner**

La empresa GFI tiene una amplia gama de herramientas de seguridad, desde herramientas para la protección de servidores, herramientas para la gestión y correlación de eventos de seguridad y como no un escáner de vulnerabilidades, LanGuard Network Security Scanner. Esta solución ha sido tradicionalmente una de las más utilizadas en entornos Microsoft y ha tenido una gran aceptación en España. El contar con la solución en castellano y con soporte en castellano para los clientes ofrecido desde España hizo de esta una de las opciones más utilizadas, además de una política de precios muy asequible. El producto no solo es un escáner de vulnerabilidades sino que además es una solución que permite desplegar parches. Al igual que Nessus y la mayoría de las soluciones equivalentes permite el perfilado de las políticas. Para escribir este artículo se ha utilizado la versión 8 del producto que actualmente está en versión beta. A la hora de escanear y auditar una plataforma Microsoft esta es una de las mejores soluciones, es la solución que más información ofrece y mejores resultados muestra. En las políticas por defecto expande algunas vulnerabilidades, es decir, las aprovecha, siempre que no sean dañinas. Muy recomendada, puedes bajarte una versión de evaluación totalmente funcional de la web <http://www.gfihispana.com>

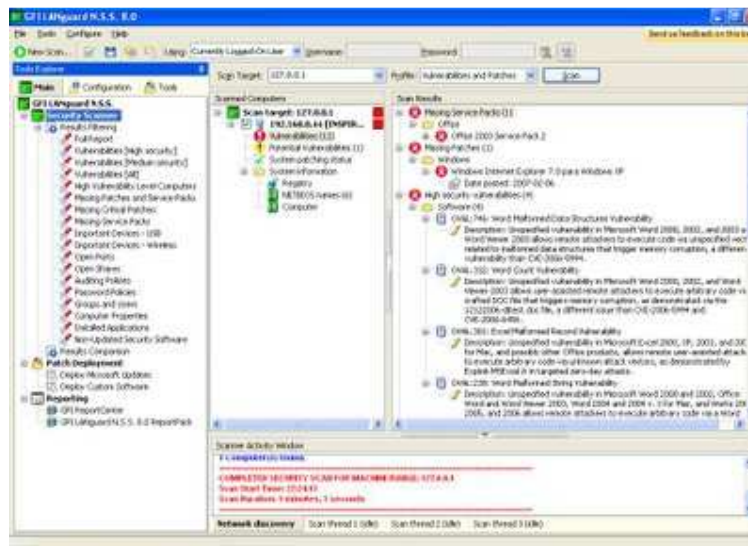


Imagen: GFI Languard Network Security Scanner

### Shadow Security Scanner

Esta herramienta, de Safety-Lab, ha sido una de mis preferidas durante mucho tiempo porque era "poco friendly", de hecho la llamábamos cariñosamente "la bestia parda". Esta forma de trabajar la herramienta le venía heredado de su predecesor Shadow Scanner, una herramienta que no se vendía y que estaba pensada no como Scanner de Vulnerabilidades sino como un Scanner para atacar, con opciones de bombardeo inclusive. Este perfil de la compañía se nota con otras herramientas como Shadow Instant Message, herramienta que se usa para "auditar la seguridad" de las conexiones de los sistemas de mensajería instantánea como MSN Messenger o Yahoo Messenger. Todavía es posible encontrar la primera herramienta en los "mentideros" de Internet. A día de hoy el mantenimiento y actualización de Shadow Security Scanner ha decaído ligeramente y personalmente creo que está por detrás de sus competidoras. No tiene soporte en castellano aunque sí en catalán. Puedes descargar una versión de evaluación completamente funcional durante 15 días desde la web de la compañía <http://www.safety-lab.com>.



Imagen: Shadow Security Scanner

## E-eye Retina

La empresa E-eye, famosa por su sniffer IRIS, tiene la solución Retina para la auditoría de vulnerabilidades en software. Al igual que GFI, cuenta con presencia y soporte en España lo que hace de ella una buena alternativa. Se puede conseguir una versión totalmente funcional de 15 días de la web de la empresa:

<http://www.e-eye.com>

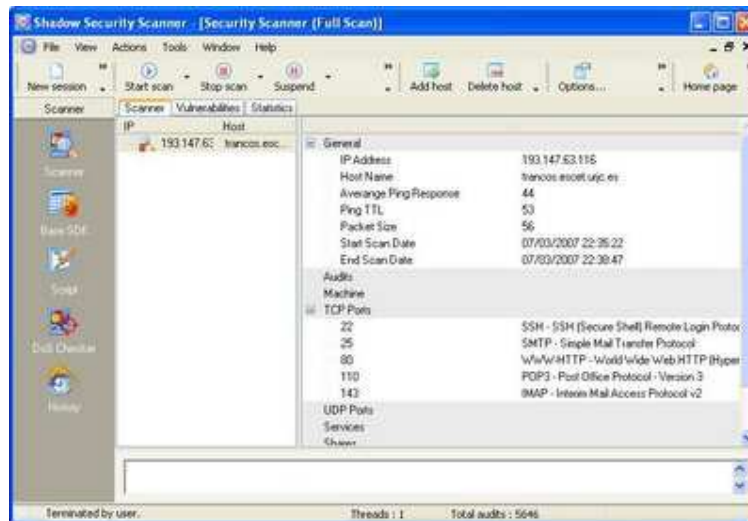


Imagen: E-eye Retina Network Security Scanner

## Otros Escáneres

Existen otros muchos escáneres de vulnerabilidades, tanto de pago, como realizados por comunidades de desarrolladores, quizás algunos echéis en falta ISS (Internet Security Scanner), NetBrute o XScan o cualquier otro, pero al final las ideas son similares. Mi recomendación como siempre es contar con un par de ellos como mínimo en cualquier test de intrusión que se vaya a realizar.

## Resumen del proceso

Si tuviéramos que resumir brevemente cual es el proceso que hay que seguir para la realización de un test de intrusión sería el siguiente:

- 1.- Identifica que quieres auditar: Utilizando las técnicas de Footprinting y FingerPrinting vistas en la primera parte.
- 2.- Busca las vulnerabilidades de esos productos: Utilizando los expedientes de seguridad o los escáneres de vulnerabilidades vistos en la segunda y tercera parte.
- 3.- Busca los exploits para esas vulnerabilidades como vimos en la segunda parte de este artículo.
- 4.- Parchea para dejar el sistema corregido tal y como recomienda el fabricante del software en los expedientes de seguridad de la vulnerabilidad.

Este proceso debe ser parte del plan de mantenimiento de los servicios/servidores y debe estar igual de planificado como el proceso de copia de seguridad o de actualización de software, sin embargo, a día de hoy, esto lo realizan pocas empresas o no tantas como debieran en un claro síntoma de descuido hacia su sistema informático.

**¿Eso es todo?**

Pues no, en primer lugar hay que tener en cuenta que un test de intrusión realizado por dos auditores distintos puede obtener resultados dispares dependiendo de la destreza de un auditor a la hora de configurar las políticas de los escáneres para saltarse los mecanismos de seguridad intermedios o afinar los plugins a ejecutar. Además de todo esto hay que tener en cuenta que una auditoría de seguridad solo tiene validez para el punto de ejecución desde donde se ejecuta, es decir, imaginemos que realizamos un test de intrusión desde Internet y hay una vulnerabilidad que está siendo protegida por el firewall de perímetro a nivel de aplicación. Desde Internet podremos tener un sistema que aparentemente no es vulnerable, mientras que un acceso desde la intranet o desde una conexión VPN puede detectar la vulnerabilidad.



## Test de Intrusión (VI de VI)

<http://elladodelmal.blogspot.com/2007/05/test-de-intrusin-vi-de-vi.html>

\*\*\*\*\*  
Artículo Publicado en PCWorld Abril 2007  
\*\*\*\*\*

### Realización de una auditoría con Tenable Nessus 3

Tras haber descargado, activada e instalado el producto lo primero que tenemos que hacer es actualizar la base de datos de plugins del producto. Si es la primera vez que lo arrancamos lo preguntará él mismo, si no, tenemos una herramienta para invocar la actualización en cualquier momento. Después configuramos el servidor, eligiendo la dirección IP y el puerto por el que va a escuchar el servicio de Nessus. Eventualmente, para conexiones remotas, deberíamos generar la lista de usuarios a los que se les permite la conexión con este servidor.



Imagen: Gestión de usuarios en Nessus

Una vez creados los usuarios tendremos que crear las políticas de auditoría ajustadas a nuestros entornos, para ello arrancamos la herramienta de Scanner de Vulnerabilidad y se selecciona la opción de "Manage Policies" y se crea una nueva. Una vez creada tendremos dos opciones de configuración principal. En primer lugar deberemos seleccionar las opciones de configuración generales de la política. En esa lista se configurará, en primera instancia si es una política Segura o no. Si decidimos que la política sea segura ya no se lanzará ningún plugin que pueda dañar el servicio. Además, es importante configurar en estas opciones las propiedades de las credenciales a utilizar, la carga que se va a realizar del servidor y las opciones específicas de rutas, ubicaciones y características que se conozcan del servidor para poder afinar el uso de los plugins. Es aquí donde se nota la destreza o no de un buen auditor de seguridad.

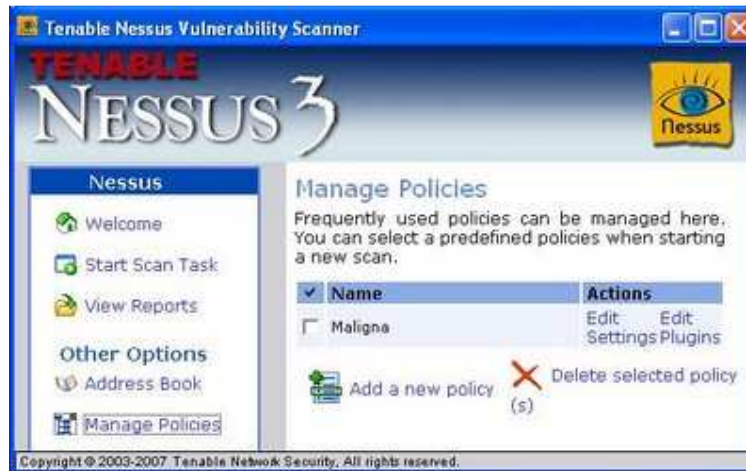


Imagen: Gestión de políticas



Imagen: Configuración de Settings de una política

La otra parte a afinar son directamente los plugins, para ello, cuando creamos una política deberemos elegir que es lo que queremos buscar. No tiene sentido realizar búsqueda de fallos locales en Gentoo, cuando estamos auditando en remoto un Windows Server 2003 R2. Para facilitar esta gestión todos los plugins están agrupados en categorías y podremos añadir o quitar categorías o directamente plugins. La ejecución de muchos de los plugins se realizará teniendo en cuenta las configuraciones definidas previamente en la política.



Imagen: Configuración de Plugins

De todos y cada uno de los plugins que acompañan Nessus hay una ficha de información accesible en el programa y que se mantiene online en el sitio web de la compañía. Con simplemente hacer clic sobre el plugin podremos saber que es lo que mira, cual es el factor de riesgo y si el plugin puede afectar o no a nuestro servidor.

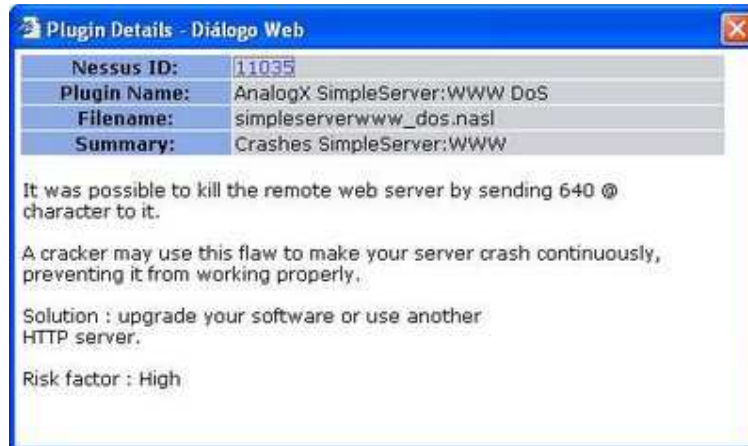


Imagen: Información de un Plugin

Una vez definida las políticas de auditoría podemos proceder ya a realizar el escaneo del servidor. Para ello seleccionamos comenzar una tarea de escaneo, elegimos la política y el motor Nessus desde el que deseamos que se realice, no hay que olvidar que la arquitectura de Nessus es cliente/servidor, gracias a lo que podremos configurar múltiples auditorías desde múltiples servidores.

El proceso completo se puede ver en la imagen siguiente:

Please enter the target you want to scan:

You can specify a single host (hostname or IP), a list of hosts separated by comma, or a range, or a network. Address of a DNS name is entered as a target, it must be resolvable in the system in order to be a valid host. (See Examples)

For Nessus's usual addresses, Nessus host can help you to manage them (previous host entries will show up in the "Hosts" list below).

If you want to skip target from the list, click here.

Please choose the plugins set you want to use:

Nessus uses plugins to do security checks. Most plugins are implemented in Nessus, others, "scripted", are external, and perform a custom security check. By selecting plugins, you can define a specific check to be run on the target.

☐ Enable all but dangerous plugins with default settings (Recommended)

☐ Enable all plugins with default settings (Even dangerous plugins are enabled)

☒ Choose a predefined policy (You should use Manage Policies to create one first)

☐ Define my policy (for advanced user)

When dangerous plugins are enabled, Nessus will attempt to exploit the target. This may cause damage to the target.

Please choose a predefined policy:

Notes: Having policy and mode can help you to simplify the Nessus's used security checks, but it does have one disadvantage: while we are working hard on providing new plugins to detect the newest vulnerabilities, the used plugin set will be outdated.

☐ Default

Choose a Nessus server:

Notes: Nessus has a client-server architecture which allows you to scan from a remote Nessus server. Please indicate if you want to scan from local host. If not, client provides the high performance of the remote Nessus server.

☒ Scan from the local host

☐ Scan from a remote Nessus server

Name:

or IP:

Host:

Username:

Password:

Scan in progress: 0 of 1 host(s) done.

Host Being scanned	Progress	Open Ports	Issues	Warnings	State
192.168.1.100	0%	1	0	0	0

Imagen: Proceso de Escaneo

## Informe de Auditoría

Ahora a recibir los deberes. Cada vez que se termina un escaneo Nessus genera un informe de auditoría completo que se almacena en un fichero xml. Dicho informe permite que se realicen diferentes visualizaciones del mismo para reflejar la información que ha sacado el escaner. En los informes se podrá ver desde los datos que son puramente informativos hasta la información que es sustancialmente importante para la seguridad y debe ser corregida.

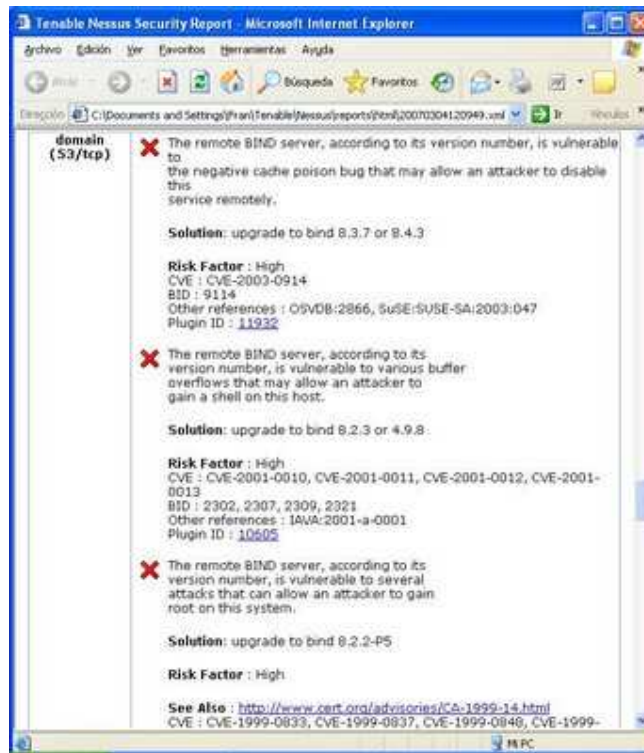


Imagen: Informe de auditoría

Vale, una vez que tienes el informe ¿qué se debe hacer? Bien, en un test de intrusión completo de una compañía se evalúan todas las vulnerabilidades intentando llegar al final, es decir, sí con un exploit se puede conseguir el control de un equipo de la organización, pues seguir adelante para ver hasta que nivel de riesgo se estaría en un caso de una vulnerabilidad similar y realizar un test de intrusión del sistema completo. Esto permitirá descubrir fallos en la política de seguridad de la red.

Si lo que queremos es simplemente corregir el servidor entonces deberemos seguir las recomendaciones para la solución de cada uno de los fallos, estas las vamos a encontrar en los expedientes de seguridad. Una vez aplicadas las medidas de remedio para todas las vulnerabilidades deberemos volver a escanear el mismo servidor y obtener un nuevo informe. El proceso debe repetirse hasta que el informe quede totalmente "limpio". Una de las características de Nessus que podemos utilizar para este proceso es la comparación de informes, con el que podremos comparar los cambios sufridos en la seguridad del servidor en cada cambio que apliquemos al servidor.



## Imagen: Gestión de Informes

Hoy en día Nessus es el escáner de vulnerabilidades más utilizado a nivel mundial aunque no es el único y existen otras alternativas/complementos muy interesantes. Aunque hay bastantes escáneres de vulnerabilidades, en el artículo de hoy vamos a ver solo algunas de las mejores soluciones profesionales. El proceso de auditoría en cualquiera de estas soluciones es similar al explicado con Nessus.

### **Para terminar**

Un último punto sobre el que hay que reflexionar antes de dar por terminado este artículo es la auditoría de las aplicaciones propias, es decir, los desarrollos personales. Una aplicación web puesta de cara a Internet, con https, con su firewall protegiéndola por delante, con su auditoría de seguridad con escáneres de vulnerabilidades limpia puede tener un bonito SQL Injection en un radio button y se acabó el cuento. En el caso de los desarrollos personales es necesario contar con una aproximación diferente, con herramientas distintas y con unos auditores más diestros, ya que no solo deben conocer el uso de las herramientas sino también los fallos en el desarrollo de tecnologías. Para aquellos que les interese este tema el mes que viene vamos a ver como se realiza un proceso de auditoría de una aplicación web y cuales son las herramientas que se pueden utilizar.

Para los impacientes que deseen ir abriendo boca les dejo los dos retos hacking que monté sobre test de intrusión en aplicaciones web. Del primero ya existe un solucionario publicado y del segundo se publicará a finales de Abril.

[Primer Reto Hacking](#)

[Segundo Reto Hacking](#)