I. PROLOGO 1

II. INTRODUCCIÓN 2

2.1. ¿Que es una intrusión? 2

2.2. Tipos de intrusiones 3

2.3. Quién es un intruso y tipos 4

2.4. ¿Cómo intentan entrar los intrusos en los sistemas? 4

2.5. Estadísticas de Intrusiones 8

III. DESCUBRIENDO AL INTRUSO 9

3.1. Detección en sistemas UNIX/Linux. 10

3.1.1. Cómo saber si hay un intruso “actualmente” en el sistema. 10

3.1.1.1. Visualización de los usuarios “logged” en el sistema. 10

3.1.1.2. Visualización de los procesos activos. 13

3.1.2. Cómo detectar que “ya ha ocurrido” una intrusión 17

3.1.2.1. Examinar los archivos log. 17

3.1.2.2. Buscar archivos setuid y setgid. 27

3.1.2.3. Chequear los archivos binarios del sistema. 29

3.1.2.4. Comprobar puertos abiertos. 30

3.1.2.5. Chequear si hay sniffers. 32

3.1.2.6. Examinar archivos que estén ejecutándose como cron y at. 34

3.1.2.7. Chequear si hay servicios no autorizados. 34

3.1.2.8. Examinar el archivo /etc/passwd. 35

3.1.2.9. Chequear la configuración del sistema y la red. 36

3.1.2.10. Buscar todos lados archivos escondidos o inusuales. 36

3.1.2.11. Examinar todas las máquinas en la red local. 38

3.1.3. ¿Qué hacer cuando se detecta un intruso? 38

3.1.4. Que herramientas podemos usar para detectar intrusos. 40

3.1.4.1. Detectores de Sniffers. 41

3.1.4.2. Detectores de troyanos. 44

3.1.4.3. Detectores de zapper's. 52

3.1.4.4. Herramientas de Análisis. 53

3.1.4.5. Crackeadores de passwords 55

3.2. Detección de Intrusos en Windows NT 57

3.2.1. Examinar los archivos de registro 58

3.2.2. Verificar si existen cuentas y grupos de usuarios desconocidas 58

3.2.3. Buscar membresías de grupo incorrectas. 58

3.2.4. Buscar derechos de usuario incorrectos. 59

3.2.5. Verificar si existen aplicaciones desautorizadas de inicio. 59

3.2.6. Verificar los sistemas binarios 61

3.2.7. Verificar la configuración y actividad de la red. 62

3.2.8. Verificar si existen partes desautorizadas. 63

3.2.9. Examinar trabajos ejecutados por el servicio del scheduler. 64

3.2.10. Verificar si existen procesos desautorizados. 64

3.2.11. Buscar por todas partes archivos inusuales u ocultos 65

3.2.12. Verificar si existen permisos alterados en archivos o en claves de registro. 65

3.2.13. Verificar si existen cambios en políticas del usuario o de la computadora. 65

3.2.14. Asegurar que el sistema no se ha movido a un Workgroup o Dominio diferente. 66

3.2.15. Examinar todas las máquinas en la red local. 66

IV. SISTEMAS DE DETECCIÓN DE INTRUSOS 67

4.1. IDS basado en red (NIDS) 69

4.2. IDS basado en host 71

4.3. Ventajas y desventajas de ambos tipos de IDS's 72

V. CONCLUSIONES 75

5.1. Conclusiones Prácticas 75

5.2. Ventajas 76

5.3. Inconvenientes 78

5.4. Posibles Ampliaciones 78

VI. BIBLIOGRAFÍA 79

I. PRÓLOGO

En este documento se trata de realizar un recorrido por diversos temas relacionados con la detección de intrusos. Primeramente se introducen conceptos generales, posteriormente se tratan los modos de obrar para llevar a cabo dicha tarea en distintos sistemas operativos y, finalmente, se centra la atención en diversas herramientas comerciales conocidas como IDS's (Intrusion Detection system).

Como una introducción a la detección de intrusos se describirá en qué consiste una intrusión, los tipos de atacantes que existen, los pasos más habituales que suele realizar un intruso para entrar en un sistema informático y los métodos que utiliza para poder entrar de nuevo en el futuro.

Una vez que se poseen conceptos generales, se profundiza en los pasos a seguir para detectar intrusos en los sistemas operativos UNIX/Linux y WindowsNT en los que se comentarán comandos útiles, diversas herramientas (como detectores e sniffers, detectores de zappers, etc.) y en general, modos de obrar recomendados para llevar a cabo dicha tarea.

En la parte final del trabajo, se realiza una introducción a los Sistemas de Detección de Intrusos (IDS) describiendo en qué consisten, características generales y los tipos de IDS's que se pueden encontrar en la actualidad.

II. INTRODUCCIÓN

La eclosión en los últimos años de las redes informáticas y fundamentalmente de Internet, ha sido el factor fundamental que ha hecho que la Seguridad Informática cobrase una importancia vital en el uso de sistemas informáticos conectados. Desde el momento en que nuestro ordenador se conecta a Internet, se abren ante nosotros toda una nueva serie de posibilidades, sin embargo éstas traen consigo toda una serie de nuevos y, en ocasiones complejos, tipos de ataque. Mientras en un ordenador aislado el posible origen de los ataques es bastante restringido, al conectarnos a Internet, cualquier usuario de cualquier parte del mundo puede considerar nuestro sistema un objetivo apetecible.

La manera en que los ataques contra infraestructuras de computadoras han crecido en los últimos 12 años ha sido espectacular, convirtiéndose así en un problema muy serio a tener en cuenta, puesto que todo sistema informático es susceptible de ser víctima de uno de estos ataques.

Es por esta razón que es necesario incluir en las redes los mecanismos de seguridad diseñados para regular y proteger la confidencialidad de los datos aunque pensar en la prevención total puede sonar poco realista. Sin embargo, se debe intentar detectar tentativas de intrusión para así poder llevar a cabo la reparación del daño más adelante. Este campo de la investigación se llama Detección de Intrusos.

2.1. ¿QUE ES UNA INTRUSIÓN?

Una intrusión puede ser definida como un conjunto de acciones que intentan comprometer o poner en peligro la integridad, la confidencialidad o la disponibilidad, aunque de forma más sencilla

se podría definir como cualquier actividad inadecuada, incorrecta, o anómala que detectemos en nuestro sistema.

2.2. TIPOS DE INTRUSIONES

Si se tiene en cuenta la naturaleza de la intrusión se puede hacer una primera clasificación o categorización de la siguiente manera:

1. Intrusiones de uso erróneo. Se definen como ataques bien definidos contra puntos débiles sabidos de un sistema. Este tipo de intrusiones pueden ser detectadas observando ciertas acciones que son llevadas a cabo sobre ciertos objetos de dicho sistema.

2. Intrusiones de anomalía. Se podrían definir como desviaciones de los patrones normales de uso del sistema. Pueden ser detectadas guardando y revisando periódicamente un perfil del sistema, en el cual se detectan desviaciones o alteraciones significativas.

Independientemente de si la intrusión está clasificada como una intrusión anómala o como de uso erróneo, existen diferentes maneras primarias en que los intrusos pueden acceder a un sistema informático, en base a las cuales se puede establecer una segunda clasificación para intrusiones:

1. Intrusión física. En este caso el intruso tiene acceso físico a la máquina (puede utilizar el teclado, etc...)

2. Intrusión del sistema. El intruso tiene una cuenta de usuario en el sistema con pocos privilegios pero puede llevar a cabo estrategias para que le sean asignados privilegios administrativos adicionales.

3. Intrusión alejada. Tentativa de penetrar un sistema remotamente, a través de la red.

2.3. QUIÉN ES UN INTRUSO Y TIPOS

Los intrusos son usuarios no autorizados en un sistema.

Si queremos diferenciar tipos de intrusos, una posible clasificación sería la siguiente:

1. Externos. Este tipo de usuarios no está autorizado para usar ningún recurso del sistema. Comúnmente son denominados intrusos (aunque intrusos lo son todos) y son el objetivo central de la seguridad física y de las técnicas de cortafuegos, por ejemplo.

2. Interno. A diferencia de los usuarios externos, este tipo de usuarios están autorizados para usar solamente algunos de los recursos del sistema. A su vez, podemos dividirlos en:

   2.1. "Enmascarados": Imitan o se hacen pasar por otros usuarios.

   2.2. Clandestinos: Evaden todo tipo de control y constituyen, sobre todo, una amenaza para sistemas débiles y sistemas mal manejados.

3. "Misfeasors": Este tipo de usuarios incluye a aquellos que emplean mal los privilegios que tienen asignados.

Todo usuario de un sistema constituye una amenaza potencial para el mismo, independientemente de su origen o de la forma en que se hayan autentificado.

2.4. ¿CÓMO INTENTAN ENTRAR LOS INTRUSOS EN LOS SISTEMAS?

Un intruso suele seguir unos pasos para entrar en el sistema. Primero recopila información general de fallos de seguridad (bugs) y de mensajes oficiales que muestran los pasos que hay que dar para

aprovechar un determinado fallo de seguridad, incluyendo los programas necesarios (exploits). Dichos fallos se aprovechan para conseguir introducirse en el sistema y están basados casi siempre en los protocolos TCP/IP, en servicios de red como NFS o NIS, o en los comandos remotos UNIX. Los protocolos basados en TCP/IP que se suelen aprovechar son TELNET, FTP, TFTP, SMTP, HTTP, etc. Cada uno de ellos tiene sus propios agujeros de seguridad que se van parcheando con nuevas versiones, aunque siempre aparecen nuevos bugs.

Toda esa información está en Internet y sólo es necesario saber buscarla. Por lo tanto, el proceso de hacking sigue las siguientes etapas:

- Obtención de la información del equipo a atacar.
- Entrada en el equipo.
- Obtención de la cuenta de root.
- Mantener los privilegios de root.
- Borrar las huellas.

Generalmente la información que se recopila del equipo a atacar será:

•El tipo de sistema operativo a atacar.

•La versión de Sendmail usada, información que se consigue tecleando telnet <equipo> 25. El número 25 es el número de puerto que utiliza normalmente dicho daemon. Una vez conectados para salir, basta utilizar QUIT o, para la obtención de ayuda, HELP. Para evitar esto, basta configurar el enrutador de manera que todas las conexiones procedentes de fuera pasen a un equipo central y que sea desde éste desde donde se distribuya el correo internamente.

•Qué servicios RPC tiene, para lo que basta con escribir rpcinfo -p <equipo>.

•Información de todo el dominio, es decir, de los equipos que lo integran. Normalmente se usa WHOIS para descubrir cual es el dominio.

•Login de los usuarios que tienen acceso al equipo. Muchas veces esto se obtiene a través del servicio FINGER si el host atacado tiene este servicio disponible. Otra manera es encontrar direcciones de correo electrónico que apunten a esa máquina o usar mecanismos de ingeniería social.

En cuanto a la penetración en el sistema podemos diferenciar dos formas básicas de introducirse:

• Entrar directamente, sin necesidad de poseer una cuenta en el sistema. Una opción es hacerlo como se detallaba al principio, con los comandos remotos.

•Conseguir el fichero de contraseñas del equipo y crackearlo. Para crackearlo existen varios programas, tanto para UNIX como para Windows.

Una vez introducidos en el equipo, los hackers intentarán obtener privilegios de root y para ello explotarán los bugs encontrados para el sistema en el primer paso. Lo que también hacen es intentar explotar bugs que afecten a sistemas UNIX en general. Si siguen sin funcionar, explotarán el sistema (hasta donde le permitan sus privilegios) para tener una visión general de cómo está protegido, por ejemplo, viendo si los usuarios tienen ficheros .rhosts, si determinados ficheros tienen permisos SUID qué usuario tiene determinados ficheros, etc. Y a partir de ahí existirán dos opciones principalmente: la primera es que se olviden durante unos días del equipo para poder recopilar más información sobre bugs actualizados y la segunda es la de hackear otra máquina del mismo dominio, que sea algo más insegura. Una vez hackeado el equipo inseguro, colocarán un sniffer para conseguir una cuenta para el otro equipo.

Un sniffer no es más que un programa que captura todo lo que pasa por la red, poniendo al equipo en modo promiscuo. La obtención de un sniffer es tan sencilla como navegar por Internet, pero incluso programas como Etherfind, Tcpdump o Ethereal pueden ser usados

para este fin, aunque no hayan sido concebidos para ello. La manera de comprobar si un sistema está en modo promiscuo es tecleando ifconfig -a. Una manera de evitar los sniffers es mediante switches en la redes de acceso general del resto de la red.

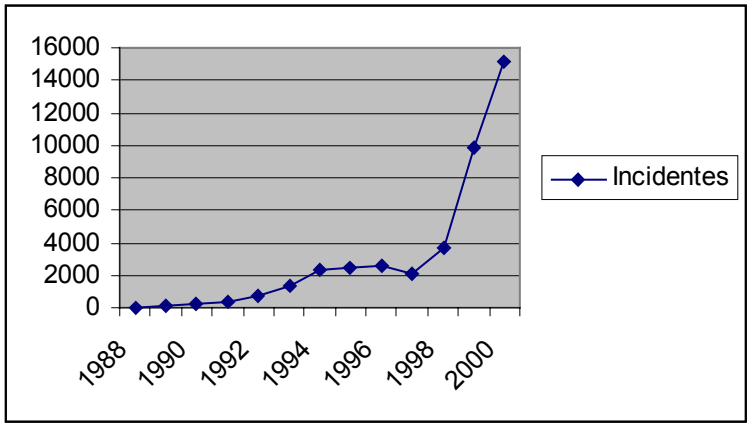
Una vez que los intrusos consiguen privilegios de root deben conseguir mantenerlos. Existen diversas formas de conseguirlo, es decir, asegurar que la próxima vez que los hackers entren en el sistema con la cuenta de un usuario que posee privilegios normales, puedan conseguir los privilegios de root de forma más fácil y sin complicaciones. Para ello, la forma más empleada es el sushi (set-uid-shell), más conocida como huevo. El sushi consiste en copiar un shell a un directorio público, en el que un usuario normal pueda ejecutar los ficheros, y cambiar el nombre al que ellos quieran. Hay que asegurarse de que el shell copiado tenga como propietario al root y, posteriormente cambiar los permisos del fichero con las cifras 4755. El 4 significa que cualquier usuario que ejecute dicho fichero lo estará ejecutando con los privilegios del propietario. Como en este caso el propietario es root y el fichero en cuestión es un shell, el sistema les abrirá a los hackers un shell con privilegios de root. Con esta operación, la próxima vez que accedan al sistema con la cuenta de un usuario normal, sólo tendrán que ejecutar el shell antes mencionado y se convertirán en root.

Por último, un intruso con unos conocimientos mínimos siempre intentará eliminar sus huellas. El sistema operativo guarda varios registros de las conexiones de los usuarios al equipo, por lo que los hackers intentarán eliminarlos. Existen varios modos de borrar sus huellas en estos ficheros. La primera es que, como la mayoría no son ficheros de texto, no podrán editarlo con un editor de texto, pero si existen programas conocidos con el nombre de zappers (los más habituales son los siguientes: marry.c, zap.c, zap2.c, remove.c, cloak.c, ...), que pueden borrar los datos relativos a un usuario en particular dejando el resto de la información intacta. La segunda manera es mucho más radical, que consiste en dejar el fichero con cero bytes o

incluso borrarlo. Esta manera sólo se utiliza como último recurso, ya que suscita muchas sospechas por parte de los administradores.

2.5. ESTADÍSTICAS DE INTRUSIONES

En este apartado se presentan datos estadísticos del número de intrusiones detectadas y reportadas al CERT.org desde 1988 hasta la fecha, las cuales nos indican la necesidad de cuidar los aspectos de seguridad informática de nuestros sistemas.



III. DESCUBRIENDO AL INTRUSO

Ante la sospecha de que nuestro sistema haya sido objeto de un ataque, se ha de determinar lo siguiente:

- Si realmente el sistema ha sido atacado.
- Si el ataque ha tenido éxito.
- En qué grado se ha comprometido nuestro sistema en caso de que haya sido atacado.

La tarea de detectar posibles intrusos será más o menos fácil en función del sistema operativo del que dispongamos puesto que algunos sistemas operativos modernos son complejos y poseen numerosos "sitios" en los cuales los intrusos pueden ocultar sus actividades. La mayor parte de los intrusos dejan señales de sus actividades en el sistema.

En principio, si estamos al día en materia de seguridad, así como de fallos que van surgiendo, no tendremos problemas de que un intruso nos entre en nuestro sistema. Realmente con un poco de esfuerzo podemos tener un servidor altamente seguro que nos evitara alrededor del 85% de los intentos de acceso no autorizados a nuestro sistema, pero en muchas ocasiones el peligro viene de los propios usuarios internos del sistema, los cuales presentan un gran riesgo debido a que ya tienen acceso al sistema, pero como siempre existen métodos de seguridad para controlar a los usuarios legítimos.

Lo fundamental es descubrir si realmente ha entrado un intruso, ya que en muchas ocasiones pensamos que ha entrado alguien pero no es cierto. Por eso, ante todo calma, esto es lo más importante para un buen administrador.

3.1. DETECCIÓN EN SISTEMAS UNIX/LINUX.

3.1.1. CÓMO SABER SI HAY UN INTRUSO "ACTUALMENTE" EN EL SISTEMA.

Cuando sospechamos que un intruso puede que se encuentre actualmente en el sistema debemos realizar dos pasos fundamentales:

1. Comprobar si los usuarios que se encuentran actualmente en el sistema son sospechosos.

2. Comprobar que procesos se están ejecutando y quién los ejecuta.

Las sospechas de que un intruso se encuentra en nuestro sistema pueden venir fundamentadas porque en el intento de comprobar si dicho intruso ha atacado el sistema (apartado 2.1.3.) nos damos cuenta, por ejemplo en las fechas de los log's o en las fechas de procesos (o ficheros), que existe una gran posibilidad que se encuentre en él en ese mismo instante. Por ello a continuación se va a explicar los dos pasos fundamentales comentados anteriormente.

3.1.1.1. *Visualización de los usuarios "logged" en el sistema.*

Si creemos que hay intrusos en nuestro sistema, lo primero a determinar es dónde están y qué están haciendo. Existen diversos comandos que permiten conocer los usuarios que están actualmente en el sistema:

• Comando "w".

Este comando muestra una visión general de todos los usuarios que se encuentran en el sistema así como los programas que están ejecutando. A continuación se muestra un ejemplo:

```
$ w
3:47pm up 18 days, 3:02, 7 users, load average: 0.02, 0.00, 0.00
User tty login@ idle JCPU PCPU what
user1 ttyp0 25Mar94 2:08 39:15 4 -tcsh
user2 ttyp1 5Apr94 8 5:51 5:28 emacs
user2 ttyp2 3:46pm w
user3 ttyp3 Mon 2pm 2:04 1 -csh
user3 ttyp4 Mon 3pm 41 21 -csh
user2 ttyp6 5Apr94 3 1:38 6 -tcsh
user2 ttyp7 Wed 2pm 5:31 17 1 –tcsh
```

La primera línea muestra información general del sistema (la hora actual, el número de usuarios conectados,...). El resto de la salida muestra los usuarios que están actualmente en el sistema, el terminal en el que están conectados y lo que están haciendo.

Ante esta información se ha de prestar atención a los siguientes aspectos:

- ✓ Validar si todos los usuarios son válidos.
- ✓ Validar que no llevan conectados en el sistema una cantidad de tiempo excesiva
- ✓ Asegurarnos de que los usuarios no estén ejecutando programas que puedan resultar sospechosos.

- **Comando “finger”.**

Este es un comando similar al anterior. Un ejemplo de su ejecución es el siguiente:

```
$ finger
Login Name TTY Idle When Where
user1 user name p0 26 Fri 11:46 host1.sub.domain
user2 user name p1 34 Tue 10:42 host2.sub.domain
user4 user name p2 Mon 14:04 host3.sub.domain
user3 user name p3 44 Mon 14:06 host5.sub.domain
```

```
user2 user name p4 Mon 16:43 host4.sub.domain
user2 user name p6 3:45 Tue 11:06 host2.sub.domain
user2 user name p7 1 Wed 14:47 host2.sub.domain
user3 user name p8 3:04 Thu 11:04 host5.sub.domain
user3 user name p9 1:02 Fri 13:52 host5.sub.domain
```

Al igual que "**w", "finger"** muestra los usuarios que están en el sistema, el terminal en el que se encuentran y el tiempo que llevan conectados. Además, muestra también desde dónde se han conectado estos. También podemos hacer un **"finger"** que nos indique quién está conectado en un ordenador remoto, esta petición se realiza por el puerto 79 (se explicará en el siguiente apartado como se puede hacer esto). Al igual que con la salida del comando **"w"** una vez visto el resultado de la ejecución de **"finger"** se ha de determinar:

- ✓ Que todos los usuarios son válidos.
- ✓ Que los usuarios no están conectados en el sistema una cantidad de tiempo excesiva
- ✓ Y a diferencia de **"w"** también se puede determinar que los usuarios se han conectado desde una localización válida

- **Comando "who".**

El comando **who** muestra información almacenada en el fichero /etc/utmp. Su salida es muy similar a la de **finger** y por lo tanto se han de verificar los mismos puntos expuestos para dicho comando. Un ejemplo de salida es el siguiente:

```
$ who
user1 ttyp0 Mar 25 11:46 (host1.sub.domain)
user2 ttyp1 Apr 5 10:42 (host2.sub.domain)
```

```
user4 ttyp2 Apr 18 14:04 (host3.sub.domain)
user3 ttyp3 Apr 11 14:06 (host5.sub.domain)
user2 ttyp4 Apr 18 16:43 (host4.sub.domain)
user2 ttyp6 Apr 5 11:06 (host2.sub.domain)
user2 ttyp7 Apr 6 14:47 (host2.sub.domain)
user3 ttyp8 Apr 14 11:04 (host5.sub.domain)
user3 ttyp9 Apr 15 13:52 (host5.sub.domain)
```

Nota: Los comandos anteriores pueden ser modificados por los intrusos de manera que quede oculta su presencia en el sistema. Por lo que debemos estar seguros que los comandos no han sido sustituidos por troyanos con el mismo nombre.

3.1.1.2. *Visualización de los procesos activos.*

Un intruso puede dejar una tarea ejecutándose en el sistema sin haber estado un tiempo excesivo en el mismo, de modo que puede haber pasado desapercibido por alguien que haya querido detectar intrusos mediante los comandos citados en el apartado anterior. Incluso el intruso podría estar en este mismo instante ejecutando procesos del sistema. Para descubrir procesos que puedan realizar tareas que atenten contra el sistema se pueden emplear los siguientes comandos:

- **Comando "ps".**

Este comando muestra los procesos que actualmente se están ejecutando en el sistema y posee diversas opciones:

- ✓ *-a*: muestra una lista de todos los procesos que ejecuta el sistema (no sólo los propios).
- ✓ -u: muestra los procesos pertenecientes a un determinado usuario.

- ✓ -x: muestra procesos que no han sido ejecutados desde un terminal.

Un ejemplo de salida para este comando es el que se muestra a continuación.

```
$ ps –aux

USER PID %CPU %MEM SZ RSS TT STAT START TIME COMMAND
user5 28206 8.1 0.4 48 280 p4 S 13:55 0:00 man inetd.conf
user5 28208 3.9 0.5 56 312 p4 S 13:55 0:00 more -s /usr/man/cat5/in
root 2 0.0 0.0 0 0 ? D Mar 25 0:02 pagedaemon
root 87 0.0 0.0 176 0 ? IW Mar 25 0:16 sendmail: accepting conn
root 1 0.0 0.0 56 0 ? IW Mar 25 0:04 /sbin/init -user3
15547 0.0 0.0 88 0 ? IW Apr 5 0:00 selection_svc
user1 184 0.0 0.0 192 0 p0 IW Mar 25 0:06 -tcsh (tcsh)
user2 28209 0.0 0.8 208 520 p5 R 13:55 0:00 ps -agux
user2 21674 0.0 0.4 112 248 p5 S 16:24 0:00 -tcsh (tcsh)
user3 16834 0.0 0.0 88 0 ? IW Apr 5 0:00 selection_svc
user3 27350 0.0 0.0 112 0 p3 IW Apr 11 0:01 -csh (csh)
user4 23846 0.0 0.0 80 0 pa IW 11:12 0:00 -csh (csh)
user3 23801 0.0 0.0 80 0 p8 IW 11:04 0:00 -csh (csh)
user2 18590 0.0 0.0 120 0 p7 IW Apr 6 0:01 -tcsh (tcsh)
user2 15591 0.0 0.0 120 0 p6 IW Apr 5 0:06 -tcsh (tcsh)
user2 15588 0.0 0.1 9288 72 p1 I Apr 5 7:08 emacs
user2 15496 0.0 0.0 112 0 p1 IW Apr 5 0:00 -tcsh (tcsh)
```

Una vez ejecutado el comando y visualizada su salida, debemos fijarnos en los siguientes aspectos:

- ✓ Hay que prestarle especial atención a los procesos que se ejecutan durante un período largo de tiempo.
- ✓ También suelen ser sospechosos los procesos que comienzan a ejecutarse a horas inusuales (por ejemplo, de madrugada).
- ✓ Como no se debe desconfiar de los procesos con nombres que pueden resultar extraños.

- ✓ Procesos que consumen un porcentaje elevado de CPU(esto probablemente indique la existencia de un programa sniffer en nuestro sistema).
- ✓ Procesos que no se ejecutan desde un terminal (en estos casos, en la columna TT de la salida se visualiza el símbolo **"?"**).

Puede darse el caso de que un sistema contenga una versión modificada del comando **"ps"**, para que no se visualicen procesos intrusos. También puede darse que un proceso intruso se esté ejecutando bajo el nombre de un proceso válido; en este caso resultaría difícil identificarlo como proceso sospechoso. Como ejemplo, decir que algunos intrusos, a menudo, ejecutan programas sniffers bajo nombres tan comunes como puede ser "sendmail" o "inetd".

- **Comando "crash"**

Análogamente al comando **"ps"**, el comando **"crash"** permite visualizar una lista de todos los procesos que se están ejecutando en un momento dado en nuestro sistema, aunque la ejecución de **"crash"** puede considerarse como un "cross-check" (una forma de chequear) contra **"ps"**, de manera que podemos comprobar si algún proceso de los que se visualizan con "crash" no se visualiza con "ps".

Una vez que se ejecuta "crash", aparece en el prompt del sistema el signo ">" seguido de **proc**, después se muestra la respuesta y finalmente vuelve a aparecer ">" seguido de **quit**, lo que indica que ha terminado la ejecución del comando, como se puede observar en el siguiente ejemplo:

```
$ crash

dumpfile = /dev/mem, namelist = /vmunix, outfile = stdout
```

```
> proc
PROC TABLE SIZE = 522
SLOT ST PID PPID PGRP UID PRI CPU EVENT NAME FLAGS
0 s 0 0 0 0 0 0 f8172698 load sys
1 s 1 0 0 0 30 0 f82b5494 init load pagi
2 s 2 0 0 0 1 0 f82b5550 load sys
3 s 965 141 965 0 26 0 f81721f8 in.rlogind swapped pagi
4 s 56 1 56 0 26 0 f81721f8 portmap swapped pagi
6 s 59 1 42 0 26 0 f81721f8 keyserv swapped pagi
7 s 11039 1 11039 0 28 0 ff12a2d0 getty swapped pagi
8 s 73 1 73 0 26 0 f81721f8 in.named load pagi
9 s 76 1 75 0 26 0 f8152d2c biod load pagi
10 s 77 1 75 0 26 0 f8152d2c biod load pagi
11 s 78 1 75 0 26 0 f8152d2c biod load pagi
12 s 79 1 75 0 26 0 f8152d2c biod load pagi
13 s 90 1 90 0 26 0 f81721f8 syslogd load pagi
14 s 98 1 98 0 26 0 ff648d2e sendmail load pagi
> quit
```

A la vista de la salida del comando **"crash"** debemos centrar nuestra atención en los siguientes aspectos, los cuales nos pueden indicar la presencia de actividad "extraña" en nuestro sistema:

- ✓ Una situación, que nos informaría rotundamente de que nuestro sistema está siendo atacado, es encontrarnos con procesos que no aparecen reflejados en la salida del comando **"ps"** (usando el PID para identificarlos).
- ✓ Al igual que con **"ps"** hay que desconfiar de los procesos que consumen un porcentaje elevado de CPU.
- ✓ También tener en cuenta los nombres de comandos inusuales (fijándonos en la columna NAME de la salida).

Nota: La salida de este comando, al igual que la de los anteriores puede ser modificada a favor del intruso.

3.1.2. CÓMO DETECTAR QUE "YA HA OCURRIDO" UNA INTRUSIÓN.

Este apartado solo tratará el punto de vista de cuando un intruso ya ha invadido nuestro sistema Unix/Linux. La utilización de los comandos y consejos a los que se hace referencia a continuación es aconsejable ante la sospecha de que un intruso haya estado en nuestro sistema pero que sabemos que ya lo ha abandonado.

Ante dicha sospecha debemos buscar una serie de señales que nos permitan encontrar huellas de que el intruso haya dejado tras de sí en el sistema. Estas señales se pueden enumerar en una serie de pasos como:

1. Examinar los archivos log.
2. Buscar archivos setuid y setgid.
3. Chequear los archivos binarios del sistema.
4. Comprobar puertos abiertos.
5. Chequear si hay sniffers.
6. Examinar archivos que estén ejecutándose como 'cron' y 'at'.
7. Chequear si hay servicios no autorizados.
8. Examinar el archivo /etc/passwd.
9. Chequear la configuración del sistema y la red.
10. Buscar todos lados archivos escondidos o inusuales.
11. Examinar todas las máquinas en la red local.

3.1.2.1. *Examinar los archivos log.*

Lo primero que se debe de hacer siempre que se tenga la sospecha de que el sistema ha sido atacado (y lo más importante) es examinar los archivos log a conexiones de lugares inusuales u otra actividad inusual. Por ejemplo, se debe buscar el último acceso al sistema de un usuario, el conteo de procesos, todos los accesos generados por syslog y otros accesos de seguridad. Hay que tener en cuenta que esto no es infalible

ya que muchos intrusos modifican los archivos log para esconder su actividad.

A continuación se hará un listado de los principales log's que se deben revisar, de herramientas que muestran algunos logs e incluso de cómo un intruso podría modificarlos para borrar sus huellas:

- **xferlog**

Si el sistema comprometido tiene servicio FTP, este fichero contiene el loggeo de todos los procesos del FTP y su localización suele ser el directorio /var/adm/. Podemos examinar que tipo de herramientas a subido el intruso y que ficheros ha bajado de nuestro servidor. Suele ser bastante interesante revisar este log ya que un intruso puede usar carpetas ocultas del directorio del FTP para guardar la información y aplicaciones que necesite para atacar el sistema.

La información que almacena este log suele ser la siguiente:

- ✓ La hora y la fecha a la que se transfiere.
- ✓ Nombre del host remoto que inicia la transferencia.
- ✓ Tamaño de fichero transferido.
- ✓ Nombre del fichero transferido.
- ✓ Modo en que el archivo fue transferido (ASCII o binary).
- ✓ Flags especiales (C para comprimidos, U para descomprimidos, T para un archivo *tar*).
- ✓ Dirección de transferencia.
- ✓ El tipo de usuario que entró en el servicio (**a** para un usuario anónimo, **g** para un invitado y **r** para un usuario local).

Un ejemplo del fichero puede ser:

```
Sat Mar 11 20:40:14 1995 329 CU-DIALUP-0525.CIT.CORNELL.EDU 426098
 /pub/simson/scans/91.Globe.Arch.ps.gz b _ o a ckline@tc.cornell.edu ftp 0*
Mon Mar 13 01:32:29 1995 9 slip-2-36.ots.utexas.edu 14355
 /pub/simson/clips/95.Globe.IW.txt a _ o a mediaman@mail.utexas.edu ftp 0 *
Mon Mar 13 23:30:42 1995 1 mac 52387 /u/beth/.newsrc a _ o r bethftp 0 *
```

```
Tue Mar 14 00:04:10 1995 1 mac 52488 /u/beth/.newsrc a _ i r bethftp 0 *
```

- **secure**

Algunos sistemas Unix loggean mensajes al fichero secure, ya que utilizan algún software de seguridad para ello, como el TCP Wrapper. En todo momento una conexión establecida con uno de los servicios que se están ejecutando bajo **inetd** (ahora, **xinetd**) y que usan TCP Wrappers, un mensaje de logeo es añadido a al fichero **"secure"** que se suele encontrar en "/var/secure". Cuando examinemos el fichero log, debemos buscar anomalías tales como servicios a los que se accedió por un método no habitual y desde host desconocidos.

Un ejemplo del fichero sería:

```
# Registro de las conexiones aceptadas.
Mar 14 23:03:11 mardg1 in.telnetd[25178]: connect from marad1.in2p3.fr
Mar 14 23:03:21 mardg1 in.telnetd[25179]: connect from marad1.in2p3.fr

# Registro de las conexiones rechazadas.
Mar 12 00:41:01 mardg1 in.ftpd[6801]: refused connect from 64.224.121.65
Mar 12 08:41:55 mardg1 in.ftpd[21508]: refused connect from 137.138.33.48
```

- **wtmp**

Guarda un log cada vez que un usuario se introduce en el equipo, sale de él o la máquina resetea. Dicho fichero se ubica normalmente en /etc/wtmp, /var/log/wtmp ó /var/adm/wtmp y contiene la información en formato usuario con la hora de conexión, IP origen del usuario, … por lo que podemos averiguar de donde provino el intruso.

Éste puede ser mostrado con el comando **"who <localización del fichero>"**, con lo que se obtendrá una salida parecida a la siguiente:

```
esper ttyp3 Mar 26 12:00 (afrodita.ei.uvigo.es) ttyp3 Mar 26 12:10
esper ttyp3 Mar 26 12:10 (afrodita.ei.uvigo.es) ttyp3 Mar 26 13:00
pepe ttyp2 Mar 30 17:00 (atenea.ei.uvigo.es) ttyp2 Mar 30 17:59
```

También puede obtenerse información este log con el comando **"last"** que permite conocer el tiempo durante el cual el intruso ha estado en el sistema y el momento en el que lo ha abandonado. La opción "-n" nos permite personalizar la salida asociada al comando, de manera que se visualizarán únicamente las últimas n entradas en el */var/adm/wtmp.*

Un ejemplo de salida para este comando podría ser la siguiente:

```
$ last –20

user1 ftp host1.sub.domain Fri Apr15 15:09 - 15:10 (00:00)
user3 ttyp9 host5.sub.domain Fri Apr 15 13:52 still logged in
user6 ttyp2 host7.sub.domain Fri Apr 15 13:45 - 14:1 (00:26)
user6 ttyp2 host7.sub.domain Fri Apr 15 10:34 - 10:34 (00:00)
user6 ftp host7.sub.domain Fri Apr 15 10:32 - 10:33 (00:01)
user4 ttyp4 host3.sub.domain Fri Apr 15 10:17 still logged in
user5 ttyp2 host6.sub.domain Fri Apr 15 09:20 - 10:29 (01:09)
user1 ttyh1 Thu Apr 14 20:33 - 22:00 (01:26)
user4 ftp host3.sub.domain Thu Apr 14 14:21 - 14:22 (00:01)
user4 ttyp2 host3.sub.domain Thu Apr 14 14:01 - 16:36 (02:35)
user4 ftp host3.sub.domain Thu Apr 14 13:43 - 13:44 (00:00)
user5 ttyp4 host6.sub.domain Thu Apr 14 13:38 - 14:56 (01:18)
user4 ttyp2 host3.sub.domain Thu Apr 14 13:37 - 13:47 (00:10)
user4 ftp host3.sub.domain Thu Apr 14 13:16 - 13:18 (00:01)
user4 ttyp2 host3.sub.domain Thu Apr 14 13:12 - 13:18 (00:05)
```

```
user4 ttypa host3.sub.domain Thu Apr 14 11:13 - 15:05 (03:52)
user4 ttyp9 host3.sub.domain Thu Apr 14 11:12 - 13:08 (01:55)
user3 ttyp8 host5.sub.domain Thu Apr 14 11:04 still logged in
user1 ftp host1.sub.domain Thu Apr 14 11:01 - 11:02 (00:00)
```

El formato de la salida es el siguiente:

La primera columna muestra el login asociado a cada usuario, seguida del terminal desde el cual dicho usuario se ha conectado. En el caso de que la conexión haya sido establecida mediante un dispositivo de red, se visualiza el nombre del sistema remoto. Finalmente se visualiza información del momento en que se produce la entrada y la salida (en caso de que ya haya abandonado el sistema) del usuario en el sistema.

Una vez visualizada la salida tras la ejecución se debe:

- ✓ Examinar las entradas registradas alrededor de la hora en la que se sospecha que el sistema pudo ser atacado, las que tienen un login que no resulta familiar, los logins de lugares inusuales, etc.
- ✓ Examinar la posibilidad de que se perdiera el fichero /var/adm/wtmp o que fuera cambiado por uno con agujeros en su salida con la finalidad de ocultar la presencia del intruso.

- **utmp**

Guarda un registro de los usuarios que están utilizando el equipo mientras están conectados a él. Este fichero se encuentra en: /var/log/utmp, /var/adm/utmp o /etc/utmp. Para mostrar la información de este fichero basta con teclear **"who"** y saldrá algo como lo siguiente:

```
esper tty0c Mar 13 12:31 pepe tty03 Mar 12 12:00 jlrivas ttyp2 Mar 1 03:01
```

- **lastlog**

En él se encuentra el momento exacto en que entró el usuario en el equipo por última vez. En algunas versiones de Unix también almacena el último acceso fallido en la cuenta de un usuario. Se ubica en /var/log/lastlog o en /var/adm/lastlog y su contenido suele ser visualizado cada vez que se entra en el sistema:

```
login: jb
password: jb
Last login: Tue May 12 07:49:59 on tty01
```

También se puede visualizar su contenido con el comando **"finger"**:

```
$ finger tim

Login name: tim          In real life: Tim Hack
Directory: /Users/tim    Shell: /bin/csh
Last login Tue Jul 12 07:49:59 on tty01
No unread mail
No Plan.
$
```

- **acct o pacct**

Registra todos los comandos ejecutados por cada usuario, pero no sus argumentos. Se encuentra en: /var/adm/acct ó /var/log/acct. Para mostrar la información, hay que teclear el comando "**lastcomm"**, de manera que se obtendrá algo parecido a esto:

```
$ lastcomm
```

```
nroff user1 ttyp1 0.39 secs Thu Sep 8 12:31
man user1 ttyp1 0.00 secs Thu Sep 8 12:31
sh user1 ttyp1 0.00 secs Thu Sep 8 12:31
page user1 ttyp1 0.03 secs Thu Sep 8 12:31
col user1 ttyp1 0.02 secs Thu Sep 8 12:31
tbl user1 ttyp1 0.02 secs Thu Sep 8 12:31
head user1 ttyp1 0.00 secs Thu Sep 8 12:31
lastcomm X user1 ttyp1 0.06 secs Thu Sep 8 12:31
lastcomm X user1 ttyp1 0.05 secs Thu Sep 8 12:31
csh F user1 ttyp1 0.00 secs Thu Sep 8 12:31
lastcomm X user1 ttyp1 2.97 secs Thu Sep 8 12:28
sh root __ 0.00 secs Thu Sep 8 12:30
atrun root __ 0.00 secs Thu Sep 8 12:30
cron F root __ 0.00 secs Thu Sep 8 12:30
sh root __ 0.00 secs Thu Sep 8 12:15
atrun root __ 0.00 secs Thu Sep 8 12:15
cron F root __ 0.00 secs Thu Sep 8 12:15
sh root __ 0.00 secs Thu Sep 8 12:00
atrun root __ 0.00 secs Thu Sep 8 12:00
cron F root __ 0.00 secs Thu Sep 8 12:00
```

Es importante tener en cuenta que **"lastcomm"** registra los comandos que son ejecutados pero no las acciones que se desencadenan desde esos comandos (por ejemplo, si ejecutamos un editor no se registra que archivos se abrieron).

Borrar las huellas con el accounting activado es mucho más difícil para el hacker, aunque hay que tener en cuenta que lo que pueden hacer es reducir la información de su presencia en el sistema, empleando dos métodos diferentes. El primero es que, nada más entrar en el sistema, pueden copiar el fichero acct a otro fichero y, antes de abandonar el equipo, sólo tendrían que copiar dicho archivo de nuevo al acct. Así, todos los comandos ejecutados durante la sesión no aparecen en el fichero acct. Un administrador puede darse cuenta de esta situación porque su entrada queda registrada en el sistema, así como

las dos copias. El segundo método que podrían emplear sería hacerse con un editor para el fichero acct que borrara los datos correspondientes al usuario, dejando intactos los del resto. Pero otra vez el administrador puede darse cuenta si el intruso realizó esta operación, ya que la ejecución del programa editor que borra sus huellas quedaría registrado como comando por su usuario. La última opción que puede emplear un hacker sería dejar el fichero acct con cero bytes lo cual llamaría la atención del personal de seguridad y tomarían medidas.

- **Syslog**

Esto no es un log sino una aplicación que viene con el sistema operativo UNIX. Dicha aplicación genera mensajes que son enviados a determinados ficheros donde quedan registrados. Estos mensajes son generados cuando se dan unas determinadas condiciones relativas a seguridad, información, etc. Los mensajes de errores típicos están ubicados en /var/log/messages, /usr/adm/messages, /var/adm/messages o incluso /var/syslog.

Un fichero típico, por ejemplo /var/syslog, sería:

```
Apr 20 13:04:22 host8 sendmail[15026]:
NAA15025:to=user8@sub.domain,user7@sub.domain,user3@sub.domain,
delay=00:00:02, mailer=smtp, relay=computer.sub.domain. [128.xxx.xx.xx], stat=Sent
(Mail accepted)

Apr 20 13:04:23 host8 sendmail[15026]: NAA15025:
to=user5,user2, delay=00:00:03, mailer=local, stat=Sent
Apr 20 13:04:23 host8 sendmail[15026]: NAA15025:
to=user1@host1.sub.domain, delay=00:00:03, mailer=smtp,
relay=host1.sub.domain. [198.128.36.1], stat=Sent (Ok)
Apr 20 13:06:20 host8 in.telnetd[15032]: connect from
```

```
computer.sub.domain (198.xxx.xx.xx)
```

De esta información se puede obtener lo siguiente:

• Información de dirección de E-mail que provengan de hosts sospechosos. Pues esto puede indicar que un intruso está enviando información a tu sistema desde otro remoto.

• Conexiones vía Telnet, tanto de entrada como de salida, deberían ser examinadas.

• Un pequeño archivo puede ser sospechoso si en el log se indica que el archivo ha sido editado o borrado.

En muchos casos, los logs del syslog guardan información que puede ser sospechosa y realmente no los sea. Esto nos resta mucho tiempo a la hora de examinar lo logs. Además estos logs suelen ser muy largos y examinarlos puede resultar complicado.

Otro fichero podría ser /var/adm/messages:

```
Mar 21 10:36:04 host8 su: 'su root' failed for user1 on /dev/ttyp2
Mar 21 10:36:08 host8 su: 'su aaa' succeeded for user1 on /dev/ttyp2
Mar 21 16:00:59 host8 xntpd[121]: Previous time adjustment didn't complete
Mar 24 15:01:44 host8 login: REPEATED LOGIN FAILURES ON console, user3
Mar 25 11:42:51 host8 shutdown: reboot by user1
Mar 25 11:42:53 host8 syslogd: going down on signal 15
Mar 25 11:48:04 host8 su: 'su aaa' succeeded for user1 on /dev/ttyp0
Mar 28 15:47:19 host8 login: ROOT LOGIN REFUSED ON ttyp3 FROM machine.sub.domain
Mar 28 16:12:12 host8 login: ROOT LOGIN console Apr 13 15:58:35 host8 su: 'su aaa' failed for user1 on /dev/ttyp0
Apr 13 15:58:55 host8 su: 'su aaa' succeeded for user1 on /dev/ttyp0
Apr 15 08:48:22 host8 named[2682]: starting. named 4.9.2 Wed
Nov 17 13:17:49 PST 1993
Apr 15 08:48:22 host8 named[2683]: Ready to answer queries.
```

En este fichero se puede encontrar actividades no deseadas:

• una autorización de entrada en un directorio de root no autorizado.

• intento de entrar como root en el sistema (mediante "su") o en una cuenta privilegiada.

Si un hacker intenta borrar las huellas que deja dicho daemon, necesita tener privilegios de root, por lo que los intrusos mirarán el fichero de configuración /etc/syslogd.conf para saber en qué ficheros están guardando la información. Cuando lo averigüen, los visualizarán y buscarán algún mensaje de la intromisión en el equipo, de la forma Login: ROOT LOGIN REFUSED on ttya. Cuando los encuentran, los borran y cambian la fecha del fichero con el comando "**touch**", de forma que coincida con la fecha del último mensaje con la fecha del fichero, ya que si no lo hacen, al comprobar las fechas éstas no coincidirían y se deducirá que alguien ha modificado el fichero.

Por lo dicho en este último párrafo y en algunos de los superiores, un administrador que intente detectar una intrusión no debe pensar que la revisión de los log's es definitiva para determinar si el sistema ha sido comprometido, aunque si que es aconsejable revisarlos todos.

Precisamente para evitar una modificación de los log's se pueden usar anti-zappers (ver apartado de "herramientas") o seguir alguno de los siguientes consejos:

- Mandar la parte más sensible del registro a una impresora, de forma que al intruso le sería imposible borrar estas entradas. Aunque si se da cuenta del truco, puede colapsar la impresora mandándole imprimir basura.

- Utilizar otra máquina como registro, necesitará atacar esta otra máquina para eliminar todas sus huellas.

- Mandar los logs por correo electrónico.

- Cualquier otra posibilidad que se os ocurra. Existen varios sistemas de registro de log, mucho más potentes y seguros que el syslog típico de Linux, como por ejemplo **Modular Syslog** de la empresa Core.

Por culpa de que los logs no son definitivos en la detección de intrusos se recomienda seguir los pasos que vienen a continuación.

3.1.2.2. *Buscar archivos setuid y setgid.*

Los sistemas Unix permiten a los usuarios elevar temporalmente sus privilegios a través de un mecanismo llamado **setuid**. Cuando un archivo con el atributo setuid es ejecutado por un usuario, el programa se va a ejecutar con los permisos del propietario del mismo.

Por ejemplo, el programa **"login"** es un programa con el atributo setuid y propiedad del root. Cuando un usuario lo invoca se habilita el acceso al sistema con privilegios de superusuario en lugar de los del propio usuario.

Los ficheros setuid aparecen en el listado de directorios con una "s" en lugar de una "x" en la posición correspondiente al bit de ejecución. Por ejemplo, la salida de **"ls -l .sh"** sería:

```
$ ls –l .sh

-r-sr-xr-x 1 root other 86012 Jun 2 01:09 .sh
```

Los intrusos frecuentemente dejan copias setuid de /bin/sh o /bin/time para que así les sea autorizado el acceso como root en una ocasión posterior. Para la búsqueda de este tipo de archivos, tenemos disponible el comando **find** (el comando **find** puede ser sustituido por un troyano para esconder ficheros del intruso, por lo que no es totalmente fiable). Un ejemplo de uso de find puede ser el siguiente:

```
find / -user root -perm -4000 -print
find / -group kmem -perm -2000 –print
```

Estos dos comandos se emplean para encontrar archivos *setuid root* y archivos *setgid kmem* en el sistema de archivos completo.

Muchas veces no interesa que los comandos anteriores busquen en el directorio completo incluyendo NFS montados en el sistema de archivos. Algunos comandos find soportan la opción "`-xdev`" para evitar buscar esas jerarquías. Por ejemplo:

```
find / -user root -perm -4000 -print –xdev
```

Existen más modos para buscar archivos setuid. Uno es usar el comando **ncheck** en cada partición de disco. Un ejemplo de uso de este comando es el siguiente:

```
ncheck -s /dev/rsd0g

# Busca archivos setuid en la partición de disco /dev/rsd0g.
```

Otra manera de detectar los cambios en los ficheros del equipo sería tecleando el comando:

```
ls -aslgR /bin /etc /usr >ListaPrincipal
```

El archivo ListaPrincipal deberá estar en alguna ubicación que no pueda ser detectada por el intruso. Después se deben ejecutar los comandos:

```
ls -aslgR /bin /etc /usr > ListaActual diff ListaPrincipal ListaActual
```

Con lo que saldrá un informe. Las líneas que sólo estén en la ListaPrincipal saldrán precedidas por el carácter <, mientras que las que sólo están en ListaActual irán precedidas por >.

3.1.2.3. *Chequear los archivos binarios del sistema.*

En ocasiones, los intrusos modifican los programas del sistema para ocultar su intrusión. Es aconsejable revisar los archivos binarios del sistema para asegurarnos que no han sido modificados. Algunos administradores con experiencia ya han descubierto numerosos programas como por ejemplo: ps, login, su, telnet, netstat, ifconfig, ls, find, du, df, libc, sync, archivos binarios referentes a /etc/inetd.conf, otros programas críticos de sistema y de la red, y librerías de objetos compartidas han sido sustituidas por troyanos. Existen varias herramientas conocidas como **RootKit** que permite a un intruso cambiar los binarios del sistema por troyanos que son copias exactas de los originales.

Los programas troyanos pueden producir el mismo checksum y timestamp estándar como la versión legítima. Debido a esto, el comando estándar "**sum"** de UNIX y los **timestamps** asociados con los

programas no son suficientes para determinar si han sido reemplazados. El uso de herramientas checksum como **cmp**, **MD5**, **tripwire** y otras herramientas checksum criptográficas son suficientes para detectar estos programas troyanos. Adicionalmente, se puede considerar usar una herramienta (PGP por ejemplo) para "firmar" la salida generada por MD5 o Tripwire, para futura referencia.

Además también se puede comparar con las copias de seguridad aunque puede que estas copias también hayan sido sustituidas por un troyano.

Por ejemplo, los binarios encontrados en localizaciones inusuales pueden ser comparados mediante el ejecutable **"cmp"**:

```
$ cmp /home/jdoe/sed /usr/bin/sed
```

3.1.2.4. *Comprobar puertos abiertos.*

Un intruso que ha atacado nuestro sistema puedo haber dejado puertos o conexiones abiertas de procesos. Para poder comprobar esto se puede usar el comando **"netstat"**, que principalmente nos da información de las conexiones abiertas.

Lo que se debería hacer es comparar la salida de este comando con la de **"last -n"** para poder comprobar si existe relación entre los usuarios que se conectaron al sistema y las conexiones abiertas.

Además el comando **"netstat"** tiene el parámetro **"-a"** que es usado para mostrar el estado de todos los sockets del equipo.

Un ejemplo de la salida de este comando puede ser:

$ netstat -a

```
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address Foreign Address (state)
tcp 0 0 host6.sub.domain.pop host1.sub.domain.1809 TIME_WAIT
tcp 0 78 host6.sub.domain.telne host9.sub.domain.54641 ESTABLISHED
tcp 0 0 host6.sub.domain.telne host7.sub.domain.1434 ESTABLISHED
tcp 0 0 host6.sub.domain.login host3.sub.domain.1022 ESTABLISHED
tcp 0 0 host6.sub.domain.login host3.sub.domain.1023 ESTABLISHED
tcp 0 0 host6.sub.domain.login host5.sub.domain.1021 ESTABLISHED
tcp 0 0 host6.sub.domain.telne host5.sub.domain.1957 ESTABLISHED
tcp 0 0 host6.sub.domain.login host2.sub.domain.1023 ESTABLISHED
tcp 0 0 *.printer *.* LISTEN
tcp 0 0 *.731 *.* LISTEN
tcp 0 0 *.pop *.* LISTEN
tcp 0 0 *.chargen *.* LISTEN
tcp 0 0 *.daytime *.* LISTEN
tcp 0 0 *.discard *.* LISTEN
tcp 0 0 *.echo *.* LISTEN
tcp 0 0 *.time *.* LISTEN
tcp 0 0 *.finger *.* LISTEN
udp 0 0 *.1022 *.*
udp 0 0 *.1023 *.*
udp 0 0 *.16517 *.*
udp 0 0 *.16516 *.*
udp 0 0 *.16515 *.*
udp 0 0 *.772 *.*
udp 0 0 *.16514 *.*
udp 0 0 *.16513 *.*
Active UNIX domain sockets
Address Type Recv-Q Send-Q Vnode Conn Refs Nextref Addr
ff65340c dgram 0 0 0 0 0 0
ff653e8c dgram 0 0 0 0 0 0
ff64978c dgram 0 0 0 0 0 0
ff648d8c dgram 0 0 ff151508 0 0 0 /dev/log
ff64920c dgram 0 0 0 0 0 0
ff64808c dgram 0 0 0 0 0 0
```

Leyendo esta salida se puede obtener información de:

• Si se tiene una conexión telnet que no está relacionada con la salida de los comandos **"who"** o **"w"**.

• Conexiones a otras redes.

En algunos casos, en los sistemas comprometidos se ha introducido una versión troyana de **"netstat"** que no enseña las conexiones hacia o desde el origen del intruso.

3.1.2.5. *Chequear si hay sniffers.*

Es importante comprobar que en los sistemas no existe el uso no autorizado de un programa de monitoreo de red, comúnmente llamado como sniffer. Los intrusos pueden usar un sniffer para capturar información de la cuenta y password de un usuario. Los sniffers son la mayor fuente de ataques hoy en día.

Muchos adaptadores ethernet están configurados (y deben estarlo) para aceptar solo mensajes que son pedidos por ellos mismos. Un atacante o un sniffer puede establecer su adaptador de red en "modo promiscuo" para escuchar todas las tramas de ethernet, de esta forma TODOS los paquetes de datos que llegan a la placa son aceptados (no sólo los destinados a esa PC) de modo que se puede "espiar" las "conversaciones", passwords, …

Normalmente cuando la interfaz pasa a modo promiscuo, queda reflejado en el fichero de logs, tal y como podemos ver aquí.

```
# cat /var/log/messages
Nov 20 08:51:20 maquineta          /kernel: fxp0: promiscuous mode enabled
```

Uno de los comandos que se puede usar para determinar si un sniffer fue instalado en nuestro sistema es **"ifconfig"** (existen otros como ifstatus o cpm). Dicho comando muestra la configuración actual de nuestra interfaz de red.

. Una salida simple de un sistema en modo promiscuo puede ser:

```
$ ifconfig –a

ie0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING,PROMISC>
inet 987.654.32.1 netmask ffffff00 broadcast 987.654.32.255
lo0: flags=49<UP,LOOPBACK,RUNNING>
inet 127.0.0.1 netmask ff000000
```

El modo promiscuo se puede ver activado en como el último parámetro de la descripción de flag's.

Como otros comandos **"ifconfig"** también puede ser un troyano. Si realmente tenemos sospechas de que un sniffer ha sido instalado se debería instalar la herramienta **cpm** y ejecutarla. Esta herramienta testeará la interfaz de red directamente y nos mostrará un informe si está en "modo promiscuo".

Otras posibles medidas, además de los programas antes mencionados, para detectar el sniffer son:

- Controlar y detectar los logs que genera el sniffer.
- Controlar las conexiones al exterior, por ejemplo, el envío sospechoso de e-mail a cuentas extrañas.
- Utilizar la herramienta **"lsof"** (LiSt Open Files), de forma que tengamos monitorizados los programas que acceden al dispositivo de red.

Por otra parte, en caso de que no podamos acceder y consultar el estado de las interfaces de red, puesto que el sniffer no esta en nuestra máquina sino que se encuentra en alguna otra máquina de la red. Lo que tendremos que hacer, es utilizar algún defecto en la implementación concreta del protocolo TCP/IP por algún programa/comando (tal y como hace el programa **neped** respecto al *arp*) o ingeniárnoslas para averiguar de alguna forma si tenemos algún sniffer corriendo en la red. Por ejemplo, una de las posibles técnicas, consiste en enviar paquetes a una máquina inexistente y cuya dirección no está dada de alta en el servidor de nombres. Sabremos que tenemos un sniffer en nuestra red si posteriormente detectamos cualquier intento de acceso a la máquina ficticia.

3.1.2.6. *Examinar archivos que estén ejecutándose como* **cron** *y* **at.**

Otro paso a seguir es examinar todos los archivos que están siendo ejecutados por **cron** y **at**. Se sabe que los intrusos dejan puertas traseras en archivos corriendo como **cron** o como **at**. Estas técnicas pueden permitir a un intruso volver a entrar en el sistema aunque lo hayamos echado. Además se debe verificar que todos los archivos o programas relacionados con tareas del **cron** y **at**, y las tareas se archiven por si mismas sean nuestras y que no tienen permiso de escritura.

3.1.2.7. *Chequear si hay servicios no autorizados.*

Es interesante chequear si hay servicios no autorizados dados de alta en el sistema. Se debe inspeccionar **/etc/inetd.conf** por si se le ha añadido algún servicio sin nuestra autorización o cualquier tipo de cambio. En particular hay que buscar entradas que ejecuten un programa shell (por ejemplo /bin/sh o /bin/csh) y chequear todos los

programas que estén especificados en **/etc/inetd.conf** para verificar que son correctos y que no han sido reemplazados por troyanos.

Además se debe comprobar la legitimidad de los servicios que nosotros mismos hemos dado de alta en el archivo /etc/inetd.conf. Los intrusos pueden habilitar un servicio que pensemos que previamente lo habíamos deshabilitado, o reemplazar el programa **inetd** con un programa troyano.

3.1.2.8. *Examinar el archivo /etc/passwd.*

Otro de los pasos a seguir en la detección de intrusos es chequear el archivo /etc/passwd en el sistema y buscar posibles modificaciones que se pudieran realizar en el mismo. En particular debemos buscar:

- ✓ la creación no autorizada de nuevas cuentas.
- ✓ cuentas sin password.
- ✓ cambios de UID (específicamente UID 0 que es el "root") a cuentas existentes.
- ✓ Una entrada como "+::".

El formato del fichero **"/etc/passwd"**:

```
root:x:0:0:root,,,:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
gogoju1:x:5015:500: GOMEZ JUAN ANTONIO:/correo/gogoju1:/bin/bash
roaldi1:x:5016:500: ALVAREZ DIEGO:/correo/roaldi1:/bin/bash
peblfe1:x:5017:500: BLANCO FERNANDO:/correo/peblfe1:/bin/bash
pirovi1:x:5018:500: RODRIGUEZ VICTOR MANUEL:/correo/pirovi1:/bin/bash
```

Actualmente las contraseñas no se guardan en este fichero sino que se hace en otro llamado **"/etc/shadow"**.

A continuación se muestra un script que imprime en la salida por defecto del sistema la lista de todos los usuarios del /etc/passwd:

```
$ cat /etc/passwd | awk -F':' '{print $1}'
```

3.1.2.9. *Chequear la configuración del sistema y la red.*

Otro paso es examinar las entradas no autorizadas en los archivos de configuración de nuestro sistema y de nuestra red. En particular hay que buscar entradas con signo '+' y nombres de host no locales inapropiados en **/etc/hosts.equiv**, **/etc/hosts.lpd** y en todos los archivos **.rhosts** (especialmente **root**, **uucp**, **ftp**, ...) del sistema. Estos ficheros no deberían tener atributo de escritura para todo el mundo.

Más específicamente, el fichero **.rhosts** es empleado para permitir el acceso remoto a un sistema y en algunas ocasiones es usado por los intruso como puertas traseras. Si el fichero fue modificado recientemente puede que se haya usado para sabotear el sistema. Inicialmente y periódicamente debemos verificar que el host remoto y el nombre de los usuarios en dichos ficheros son consistentes.

3.1.2.10. *Buscar todos lados archivos escondidos o inusuales.*

A menudo, la mejor manera de averiguar si nuestro sistema ha sido o no comprometido es a través del chequeo de los archivos del mismo. Los intrusos suelen ocultar su presencia en un sistema usando archivos o directorios ocultos o inusuales (ficheros que empiezan por un

'**.**' (punto), que no salen con un simple **ls**), ya que pueden ser usados para esconder herramientas que le permitan romper la seguridad del sistema o incluso pueden contener el /etc/passwd del sistema o de otros sistemas a los cuales ha entrado el intruso.

Muchos intrusos suelen crear directorios ocultos utilizando nombres como '...' (punto-punto-punto), '..' (punto-punto), '..^g' (punto-punto control+G). En algunos casos un intruso ha utilizado nombres como '.x' o '.hacker' o incluso '.mail' (algunos de ellos pueden parecer normales para el usuario o el administrador).

De nuevo el comando **find** puede ser usado para buscar archivos ocultos, por ejemplo:

```
find / -name ".. " -print -xdev
find / -name ".*" -print -xdev | cat –v
```

También puede usarse para listar todos los ficheros que fueron *"modificados"* en los últimos n días:

```
$ find / -mtime -ndays –ls
```

Este comando es también interesante para localizar fichero que han cambiado de *"inode"* en los últimos n días:

```
$ find / -ctime -ndays -ls.
```

También deberíamos examinar los directorios **~/ftp** los cuales pueden ser escritos por usuarios anónimos (que sean intrusos) que almacenen y cambien ficheros.

3.1.2.11. *Examinar todas las máquinas en la red local.*

Se debe examinar cuidadosamente todos los ordenadores de nuestra red local en busca de indicios de que nuestra red ha sido comprometida. En particular, aquellos sistemas que compartan **NIS+** o **NFS**, o aquellos sistemas listados en el **/etc/hosts.equiv**. Lógicamente también hay que revisar los sistemas informáticos que los usuarios comparten mediante el acceso del **.rhost**.

Lógicamente si nuestro ordenador ha sido atacado, probablemente los sistemas de nuestra red también hayan sido atacados.

3.1.3. ¿QUÉ HACER CUANDO SE DETECTA UN INTRUSO?

Una vez vistas diferentes técnicas para detectar que hemos sido atacados por un intruso vamos a establecer los pasos que se deben seguir cuando hemos ya estamos seguros de que estamos siendo atacados.

Si hemos pillado al intruso en el momento, tenemos varias opciones:

- ✓ Hablar con el, usando el comando **talk**, aunque debemos tener en cuenta que puede contestar de forma amistosa (ayudándonos con la seguridad del sistema) o agresiva (borrando el sistema para no dejar rastro).
- ✓ Desconectarle del sistema, usando el comando **kill**, pero para evitarnos que vuelva a entrar, antes de usar **kill**, usaremos el comando **passwd** para cambiar el password de la cuenta por la cual el intruso entró, por ejemplo, podemos ejecutar los siguientes comandos:

```
# Averiguamos los procesos del intruso en cuestión.
$ ps –aux
```

USER	PID	%CPU	%MEN	VSIZE	RSIZE	TT	STAT	TIME	COMMAND
ROOT	1434	20.1	1.4	968K	224K	01	R	0:00	ps aux
intruso	147	1.1	1.9	1.02K	304k	p3	S	0:07	~ (csh)
intruso	321	10.0	8.7	104k	104k	P3	S	0:09	cat /etc/passwd
intruso	339	8.0	3.7	2.05K	456k	P3	S	0:09	crack

 # Cambiamos la password del intruso.
$ passwd intruso

Changing password for intruso
New password: noentrasmas
Retype new password: noentrasmas

 # Matamos los procesos pertenecientes al intruso.
 # kill -9 147 321 339

Debemos utilizar las utilidades del sistema para recopilar información del intruso, en caso de denuncia será necesaria. Por lo que trataremos de tracearle, usando los siguientes comandos: who, w, last, lastcomm, netstat, snmpnetstat, ….

También sería interesante examinar los ficheros history del shell, como el .history, .rchist y ficheros similares.

Además ejecutando el comando **"finger"**, intentaremos sacar información del host de donde provino el ataque, como por ejemplo:

$ finger @intruso.es → **Comprueba los usuarios conectados en una máquina.**
$ finger intruso@intruso.es → **Comprueba un usuario conectado a una máquina.**

También podemos dirigirnos a Internic (http://www.internic.net) donde podemos pedir información de cualquier servidor del mundo, siempre y cuando no sea militar. Allí ponemos el dominio del servidor donde provino el ataque, y podremos ver con quien debemos ponernos en contacto con el servidor atacante. Si existe un teléfono de contacto,

lo mejor sería llamar a la persona encargada, ya que si enviamos un mail informándole que tiene un pirata puede que el intruso intercepte el mensaje y se haga pasar por el administrador. A parte de Internic también podemos conectar vía Telnet para solicitar información de un servidor.

Ahora que tenemos bastante información del atacante, lo mejor sería desconectar nuestro servidor de Internet y dedicarnos unos días a repasar cuidadosamente lo sucedido.

Deberíamos hacer una copia de seguridad, por lo que ejecutaremos la siguiente sentencia:

```
# dd if=/dev/sda of=/dev/sdb
```

Por último, tan solo nos queda realizar un exhaustivo análisis mediante los pasos descritos en el apartado anterior.

3.1.4. QUE HERRAMIENTAS PODEMOS USAR PARA DETECTAR INTRUSOS.

Ahora se explicarán algunas de las diferentes herramientas que están disponibles a lo largo de Internet, y además casi todas son freeware, por lo que no existe excusa alguna para no usarlas. Además usando habitualmente estas herramientas mantendremos nuestro sistema seguro.

Las herramientas que se describirán a lo largo de este apartado, van desde anti-zapper's, detectores de sniffers, detectores de troyanos, así como algunas herramientas que también utilizan los intrusos y que nosotros emplearemos para nuestro propio beneficio. Algunas de estas herramientas ya se han ido explicando a lo largo de este documento y debido a la gran cantidad de herramientas disponibles no se explican aquí todas las que existen.

3.1.4.1. *Detectores de Sniffers.*

A la hora de detectar un sniffer se pueden emplear numerosos programas. Un sniffer se encarga de monitorear la red y su explicación ya fue introducida en el apartado 2.1.3.5. Los intrusos pueden usar un sniffer para capturar información de la cuenta y password de un usuario.

Uno de los comandos que se pueden emplear para detectar si un intruso está usando un sniffer es **"netstat"** pero no es 100% fiable. Otro es el comando **"ifconfig"** (ambos comandos fueron comentados con anterioridad).

También podemos usar otros comandos y herramientas como las que se detallan a continuación:

- **cpm**

La herramienta **cpm** (Check Promiscuous Mode), cuya funcionalidad es similar a la de **ifstatus** permite testear el estado de una interfaz de red y determinar si está en modo promiscuo (como ya comentamos anteriormente, esto es un claro síntoma de detección de sniffers), un síntoma de una posible actividad de un sniffer en el sistema. Un salida de **cpm** puede ser:

```
$ cpm
    8 network interfaces found:
   eth0:7: Normal
   eth0:6: Normal
   eth0:5: Normal
   eth0:3: Normal
   eth0:2: Normal
   eth0:1: Normal
   eth0: *** IN PROMISCUOUS MODE ***
```

```
lo: Normal
```

Como se puede ver en el cuadro una de las interfaces está en modo promiscuo.

- **ifstatus**

El programa ifstatus puede correrse en los sistemas UNIX para identificar interfaces de red que estén en depuración o en modo promiscuo. Las interfaces de red en estos modos pueden ser una señal que un intruso está supervisando la red para robar passwords y otra información.

Sería interesante que se ejecutara el **"ifstatus"** con el cron una vez por hora mas o menos. Además, si se tiene un cron moderno que manda por correo el rendimiento de trabajos del cron a su propietario.

El programa no imprime ninguna salida (con el argumento -v si muestra un salida siempre) a menos que encuentra las interfaces en "malos" modos.

Empleando el argumento **–v** y suponiendo que la interfaz no se encuentra en modo promiscuo, la salida del programa sería la siguiente:

```
checking interface le0... flags = 0x863
checking interface le1... flags = 0x862
checking interface le2... flags = 0x862
```

Ahora bien, en el caso de que la interfaz se encuentre en modo promiscuo o si dichas interfaces están en depuración, la salida con la opción **–v** sería:

```
checking interface le0... flags = 0x963
WARNING: COMP1.SMPL.COM INTERFACE le0 IS IN PROMISCUOUS MODE.
checking interface le1... flags = 0x866
WARNING: COMP1.SMPL.COM INTERFACE le1 IS IN DEBUG MODE.
checking interface le2... flags = 0x966
WARNING: COMP1.SMPL.COM INTERFACE le0 IS IN PROMISCUOUS MODE.
WARNING: COMP1.SMPL.COM INTERFACE le1 IS IN DEBUG MODE.
```

Como se puede ver **"ifstatus"** en este último caso alerta de una situación anómala.

- **Antisniff**

Es una de las mejores herramientas de detección de sniffer de forma remota, aunque quizás este un poco obsoleto, sobretodo porque no contempla la nueva generación de sniffers.

Antisniff escanea una red y detecta si sus ordenadores están en modo promiscuo. Con antisniff se puede saber quién está escuchando el tráfico de nuestra red.

- **Sentinel**

Es otra interesante herramienta, cuyo objetivo principal es la detección remota de sniffers. Utiliza las librerías **libcap** y **libnet** y tenemos el código fuente disponible.

- **Otros programas para detectar sniffers:**

✓ promisc.c: Es un programa escrito en lenguaje C.

✓ NePED

Debido a que muchos sniffers logean las conexiones de la misma forma, la cual es la siguiente:

```
TCP/IP LOG -- TM: Tue Nov 15 15:12:29 –
PATH: not_at_risk.domain.com(1567) => at_risk.domain.com(telnet)
```

podemos fácilmente podemos escribir un pequeño shell script que busque ficheros de sniffers:

```
$ grep PATH: $sniffer_log_file | awk '{print $4}' | \ awk -F\( '{print $1}'| sort –u
```

lógicamente debemos ajustar este script a nuestras necesidades.

3.1.4.2. *Detectores de troyanos.*

Un troyano es una aplicación que consta de dos módulos, uno servidor y otro cliente, el atacante tiene la parte cliente y la víctima el módulo servidor, de esta manera el intruso ejecuta una orden sobre su modulo, y este hace un "eco" al otro modulo (el de la víctima), y lo ejecuta en local, de esa manera se puede ejecutar cualquier comando (que te permita la aplicación) si necesidad de tener permisos en la máquina víctima.

Se sabe de casos en los que los intrusos han reemplazado los siguientes programas por un caballo de troya para facilitar su ataque: /usr/etc/in.telnetd, /bin/login, ps, netstat, ifconfig, su,... y así una larga lista de ellos. Ya que los intrusos pueden instalar variaciones de diferente comandos de Unix se pueden utilizar algunos de los siguientes programas para detectarlos:

- ✓ Podemos usar el comando **sum** pero tampoco es 100% fiable.

- ✓ También podemos usar el comando **cmp**, pero lo mismo que el comando anterior.

Otra opción es verificar los programas usando información de chequeo criptográfico:

- **Tripwire**

Tripwire es una herramienta que verifica la integridad de un conjunto de archivos y directorios seleccionados por el usuario. Los archivos y directorios actuales se comparan con la información almacenada en una base de datos previamente generada. Cualquier diferencia es señalada por medio de una bandera y se registra, incluyendo entradas agregadas o suprimidas. Cuando se ejecuta contra los archivos del sistema sobre una base regular, Tripwire nos permite descubrir los cambios en los archivos del sistema críticos y toma inmediatamente medidas apropiadas de los daños. Además la herramienta detecta e informa al administrador del sistema de los posibles cambios, adiciones o borrados de dichos ficheros.

Una vez inicializada su base de datos, puede ejecutarse la herramienta y en consecuencia generará los siguientes informes según los eventos que ocurrieran en el sistema:

a) Informe que se muestra cuando **no han ocurrido cambios**:

La salida que se muestra a continuación indica que no han ocurrido cambios en los archivos y directorios.

En la siguiente salida se habilita el modo "verbose" **–v** para mostrar el escaneo realizado por **"tripwire"** de cada directorio y fichero. Además, se configuró para que solo se escanearan los ficheros de */etc*.

```
# ./tripwire -v -c ./tw.config -d ./databases/tw.db_<the local system's host name>
```

```
### Phase 1: Reading configuration file
### Phase 2: Generating file list
./tripwire: /etc/dfs/sharetab: No such file or directory
./tripwire: /etc/hosts.equiv:  No such file or directory
./tripwire: /etc/named.boot: No such file or directory
./tripwire: /etc/rmtab: No such file or directory
### Phase 3: Creating file information database
scanning: /etc scanning: /etc/TIMEZONE
scanning: /etc/aliases
scanning: /etc/autopush
scanning: /etc/clri

 (... resto de la salida borrada ...)

scanning: /etc/printers.conf
scanning: /etc/shells
scanning: /etc/one-time.sh
scanning: /etc/mkgroup
scanning: /etc/mkpasswd
scanning: /etc/mkshadow

### Phase 4: Searching for inconsistencies
### ### Total files scanned:     461
### Files added:                 0
### Files deleted:               4
### Files changed:               454
###
### After applying rules:
### Changes discarded:           454
### Changes remaining:             0
```

En la fase 4 **"Tripwire"** contabiliza sus búsquedas. La salida muestra 454 archivos que han cambiado, pero según las reglas que han sido definidas, ninguno de los cambios ha sido considerado como inesperado. Si realmente las modificaciones sobre ficheros fueran inesperadas, el informe sería como el del siguiene punto.

b) Informe que se muestra cuando **han ocurrido cambios**:

A continuación se muestra la salida de **"Tripwire"** después de que se haya añadido un nuevo usuario en el sistema. En la fase 4 se puede ver como **tripwire** muestra 5 cambios inesperados. En la siguiente salida aparece una nueva fase 5, esta fase nos aporta información sobre las modificaciones detectadas.

```
# ./tripwire -c ./tw.config -d ./databases/tw.db_<the local system's host name>
### Phase 1: Reading configuration file
### Phase 2: Generating file list
./tripwire: /etc/dfs/sharetab: No such file or directory
./tripwire: /etc/hosts.equiv: No such file or directory
./tripwire: /etc/named.boot: No such file or directory
./tripwire: /etc/rmtab: No such file or directory
### Phase 3: Creating file information database
### Phase 4: Searching for inconsistencies
###
### Total files scanned:   463
### Files added:           2
### Files deleted:         4
### Files changed:         456
###
### After applying rules:
### Changes discarded:     455
### Changes remaining:     5
###
added: -rw------- root   0 Apr 19 16:21:14 1997 /etc/.pwd.lock
added: -rw------- root   0 Apr 19 16:21:13 1997 /etc/.group.lock
changed: drwxr-xr-x root 3584 Apr 19 16:21:14 1997 /etc
changed: -r--r--r-- root 6982 Apr 19 16:20:58 1997 /etc/passwd
changed: -r-------- root 1571 Apr 19 16:20:59 1997 /etc/shadow
### Phase 5: Generating observed/expected pairs for changed files
###
### Attr Observed (what it is) Expected (what it should be)
### ============================================================
/etc
  st_mtime: Sat Apr 19 16:21:14 1997   Sat Apr 19 14:00:09 1997
  st_ctime: Sat Apr 19 16:21:14 1997   Sat Apr 19 14:00:09 1997
/etc/passwd
```

```
st_ino: 74339                74305
st_size: 6982                6932
st_mtime: Sat Apr 19 16:20:58 1997   Sat Apr 19 14:00:08 1997
st_ctime: Sat Apr 19 16:20:58 1997   Sat Apr 19 14:00:08 1997
md5 (sig1): 1k0No.UNC9mCJglGMtPk4O    0D5zziXYpwvXxOzy.DrYCx
snefru (sig2): 0ENvBIiHddw3tJMRQUMFHy 3ilRZc:dY2NFYLu3CAEXVI
/etc/shadow
st_ino: 74305 74338
```

Las dos primeras líneas de la fase 4 muestran el número de ficheros creados cuando la herramienta de administración añade el nuevo usuario. Las tres líneas siguientes muestran los ficheros y directorios que ya existían previamente y que han sido cambiados como resultado de la adición. Ya en la fase 5 se visualizan las siguientes modificaciones:

- ✓ Las marcas de tiempo de la creación de un inode en /etc
- ✓ Los atributos del fichero /etc/passwd
- ✓ El número de inode del fichero /etc/shadow

Es evidente que el contenido de /etc/passwd ha sido modificado ya que como se puede ver en el informe, su tamaño y el checksum criptográfico (md5) han cambiado.

c) Informe que se muestra cuando **se pierden ficheros**:

En los dos ejemplos anteriores se muestra, en la fase 2, que se perdieron 4 ficheros. Esto ocurre porque los ficheros estaban especificados en el archivo de configuración de **"Tripwire"** y por el contrario no se encuentran en la base de datos que ha generado antes de realizar el informe. Si los ficheros fueron borrados, se especifica en la fase 4 junto con la cantidad de ficheros:

```
### Files deleted: 4
```

Similarmente, también se muestra un línea en la que se indica cual es el fichero que se borró:

```
deleted: -rw-r----- 0 36 Jul 3 10:48:26 1997 /etc/testdir/testfile
```

d) Informe que se muestra los **nuevos ficheros y directorios**:

Si se añaden ficheros o directorios se especifica, junto con el número de ellos, en una línea de la fase 4 como la siguiente:

```
### Files added: 2
```

Como con los ficheros perdidos o cambiados, sus nombres y atributos son también listados.

```
added: -rw------- root 0 Apr 19 16:21:14 1997 /etc/.pwd.lock
added: -rw------- root 0 Apr 19 16:21:13 1997 /etc/.group.lock
```

- **MD5**

MD5 es el programa de checksum criptográfico, más popular y más aconsejable, que toma como entrada un mensaje de longitud arbitraria y produce como salida una "huella digital" de 128 bits o un "mensaje asimilado" de la entrada.

- **chkrootkit**

"chkrootkit" es otra herramienta que nos permite verificar que tenemos la versión original de nuestros ficheros y no la versión troyanizada. En su última versión disponible detecta troyanos en todos estos **ficheros**: aliens, asp, bindshell, lkm, rexedcs, sniffer, wted, z2,

amd, basename, biff, chfn, chsh, cron, date, du, dirname, echo, egrep, env, find, fingerd, gpm, grep, hdparm, su, ifconfig, inetd, inetdconf, identd, killall, login, ls, mail, mingetty, netstat, named, passwd, pidof, pop2, pop3, ps, pstree, rpcinfo, rlogind, rshd, slogin, sendmail, sshd, syslogd, tar, tcpd, top, telnetd, timed, traceroute, write.

Siendo capaz de detectar los siguientes **RootKits**: Solaris rootkit, FreeBSD rootkit, lrk3, lrk4, lrk5, lrk6, t0rn (and t0rn v8), some lrk variants, Ambient's Rootkit for Linux (ARK), Ramen Worm, rh[67]-shaper, RSHA, Romanian rootkit, RK17, Lion Worm, Adore Worm, LPD Worm, kenny-rk, Adore LKM, ShitC Worm, Omega Worm, Wormkit Worm, dsc-rootkit.

Una vez compilados los programas (chkwtmp, chklastlog, chkproc, chkwtmp, ifpromisc) que utiliza el chkrootkit para realizar parte de sus trabajos (chkrootkit en sí mismo es un shell script), la utilización del mismo es como se puede observar a continuación:

```
# ./chkrootkit
ROOTDIR is `/'
Checking `amd'... not found
Checking `basename'... not infected
Checking `biff'... not infected
Checking `chfn'... not infected
Checking `chsh'... not infected
Checking `cron'... not infected
Checking `date'... not infected
Checking `du'... not infected
Checking `dirname'... not infected
Checking `echo'... not infected
Checking `egrep'... not infected
Checking `env'... not infected
Checking `find'... not infected
Checking `fingerd'... not found
Checking `gpm'... not infected
Checking `grep'... not infected
Checking `hdparm'... not infected
Checking `su'... not infected
Checking `ifconfig'... not infected
Checking `inetd'...

....
```

Solo puede darse un problema y es que al ser un shell script, se basa en las siguientes herramientas: awk, cut, echo, egrep, find, head, id, ls, netstat, ps, strings, sed, uname y estas pueden contener algún troyano. La solución nos la da el propio programa, ya que por medio de un simple parámetro, le indicaremos donde debe de buscar y ejecutar esos programas que necesita:

- ✓ Si los tenemos un CD-ROM:

```
./chkrootkit -p /cdrom/bin
```

- ✓ En caso de tener los binarios originales de la distribución en un diskette:

```
./chkrootkit -p /floppy
```

También podemos verificar algún determinado fichero indicándolo directamente:

```
# ./chkrootkit ps login sendmail
ROOTDIR is /'
Checking 's'... not infected
Checking 'login'... not infected
Checking 'sendmail'... not infected
```

"Chkrootkit" también permite verificar la alteración de los logs mediante los programas chkwtmp y chklastlog:

```
# ./chkwtmp
# ./chklastlog
```

3.1.4.3. *Detectores de zapper's.*

Un atacante, cuando termine su intrusión, querrá borrar sus huellas e intentará acceder a los logs mediante unas utilidades para eliminar la parte correspondiente a su ataque. Estas utilidades se llaman **zappers**.

Para descubrir el uso de los zappers existen unos programas (antizappers) que los detectan. Los más nombrados en la bibliografía son:

- ✓ Antizap.c
- ✓ Antizap2.c

Aunque estos dos programas se mencionan numerosas veces, en la bibliografía, es complicado encontrarlos, e incluso en diversos documentos se llega a indicar tal circunstancia.

Pero también podemos tomar medidas para complicar a los intrusos la modificación de los log's. Por ejemplo, en el sistema operativo Linux y concretamente en su sistema de archivos, ext2 (Second Extended File System) y ext3, existen unos bits especiales (a parte del setuid, setguid, x [ejecución], w [escritura] y r [lectura]) e inexistentes en otros sistemas Unix para proteger aún más los ficheros. Uno de ellos es el **bit a**, este bit indica que sólo se podrá acceder al fichero en modo escritura para añadir datos, nunca para borrar. Este bit es extremadamente útil para los ficheros de log, ya que cuando un atacante quiera borrar sus huellas de los logs no le estará permitido. Obviamente, si el usuario ha conseguido privilegios de root, podrá resetear este bit, pero no olvidemos que la mayoría de los usuarios que acceden a los sistemas de forma fortuita, son gente sin experiencia, sin conocimientos de Unix y posiblemente, utilizando alguna utilidad que ni siquiera sabe cómo funciona, de modo que con este bit activado, tenemos una posibilidad más de que se queden registrados su actos.

3.1.4.4. *Herramientas de Análisis.*

Continuando con la detección de intrusos también existen una serie de herramientas de análisis que pueden resultar bastante útiles. Algunas de las más importantes se nombran a continuación:

- **Satan**

Posiblemente la herramienta mas conocida. SATAN es una herramienta de prueba y reporte que colecciona una gran variedad de información sobre los hosts conectados a una red de computadoras.

SATAN registra información sobre unos hosts y redes específicas mediante la recolección de información de los servicios de red (por ejemplo: finger, NFS, NIS, ftp, …). Puede realizar informes con los resultados de su propia investigación sobre los posibles problemas de seguridad.

SATAN también puede ser usado para detectar vulnerabilidades en host. Esto puede ser usado tanto en la detección de intrusos como para realizar ataques.

SATAN está disponible de muchos sitios, incluyendo

ftp://ftp.win.tue.nl/pub/security/satan_doc.tar.Z

- **Lsof (List Open Files)**

Es un programa que lista todos los ficheros abiertos, incluidos los sockets abiertos.

Con el uso de **"lsof**", podemos detectar la presencia de un sniffer. No encuadramos esta herramienta dentro de los detectores de sniffers porque en realidad ese no es su cometido, aunque se pueda usar para intentar localizarlos.

Un sniffer, que se encuentra instalado en nuestro sistema, guarda en un archivo log la información que va recolectando y además puede tener conexiones de red abiertas. Con **"lsof"** podemos visualizar todos los ficheros abiertos, incluyendo los sockets, lo cual nos será muy útil para detectar a dicho sniffer. El comando muestra por la salida por defecto la siguiente información:

```
# lsof
COMMAND    PID  USER   FD   TYPE    DEVICE   SIZE/OFF   NODE NAME
Init         1  root  cwd    DIR      3,1       1024      2 /
Init         1  root  rtd    DIR      3,1       1024      2 e
Kerneld     20  root   1u   CHR       4,0        0t0  10174 /dev/console
Kerneld     20  root   2u   CHR       4,0        0t0  10174 /dev/console
```

Podemos obtener esta herramienta de la siguiente URL:

ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/

- **Otras herramientas:**

✓ TCP_Wrapper: Es un conjunto de utilidades para controlar nuestro servidor.

✓ Netcat 1.10: Para saber por donde nos puede entrar un intruso, ya que este programa es capaz de crear cualquier tipo de conexión.

✓ COPS: Otro conjunto de herramientas que se puede encontrar en ftp://info.cert.org/pub/tools/cops.

✓ Roses Software Check Tool V.1.2.2: Interesante herramienta de análisis para servidores Linux.

✓ Stalker Audit-Trail Tool: Herramienta para auditar los log's.

- ✓ IDES/NIDES (Intrusion-Detection Expert System/Next-Generation IDES): Una herramienta de detección de intrusos en tiempo real.
- ✓ Saint (Security Analisys INtegration Tool): Herramienta en español para auditar.
- ✓ NetSuite Professional Audit: Herramienta profesional para auditar.
- ✓ tcplist: Lista todos los puertos abiertos que tenemos, además de diversa información más. Se puede conseguir en ftp://ftp.cdf.toronto.edu/pub/tcplist.

3.1.4.5. *Crackeadores de passwords.*

- **Crack V5**

Posiblemente el crakeador mas conocido. Crack es un programa libremente disponible diseñado para identificación. En UNIX, el algoritmo DES encripta passwords que se pueden encontrar en diccionarios extensamente disponibles. Las técnicas de encriptación están descritas en la documentación del Crack.

Muchos administradores del sistema ejecutan el Crack como un sistema regular de procedimiento de administración y notifica a dueños de cuentas a quienes les han "crackeado" passwords. El Crack está disponible en:

ftp://coast.cs.purdue.edu/pub/tools/unix/crack/.

- **John The Ripper**

Es un crackeador de passwords que un administrador puede usar para ver si una password de un usuario es fácilmente crackeable.

3.2. DETECCIÓN DE INTRUSOS EN WINDOWS NT

Como ya se ha comentado anteriormente este trabajo está más centrado en la detección de intrusos en Linux/Unix debido a la mayor facilidad que aporta este sistema operativo a la hora de obtener información y debido a que su empleo en sistemas de seguridad está mucho más valorado que el de Windows NT. Por lo tanto lo que se hace a continuación es una pequeña referencia a lo posibles pasos a seguir para la detección de intrusos en un sistema operativo como Windows NT.

Estos pasos son los siguientes:

1. Examinar los archivos de registro.
2. Verificar si existen cuentas y grupos de usuarios desconocidos.
3. Buscar membresías de grupo incorrectas.
4. Buscar derechos de usuario incorrectos.
5. Verificar si existen aplicaciones desautorizadas de inicio.
6. Verificar los sistemas binarios.
7. Verificar la configuración y actividad de la red.
8. Verificar si existen partes desautorizadas.
9. Examinar trabajos ejecutados por el servicio del scheduler.
10. Verificar si exiten procesos desautorizados.
11. Buscar por todas partes archivos inususales u ocultos.
12. Verificar si existen permisos alterados en archivos o en claves de registro.
13. Verificar si exiten cambios en políticas del usuario o de la computadora.
14. Asegurar que el sistema no se ha movido a un Workgroup o Dominio diferente.
15. Examinar todas las máquinas en la red local.

3.2.1. EXAMINAR LOS ARCHIVOS DE REGISTRO.

El primer paso a llevar a cabo a la hora de detectar una intrusión en sistemas NT es examinar los archivos de registro provenientes de conexiones establecidas desde localizaciones no habituales o de otro tipo de actividad inusual. En este sentido, se puede hacer uso del "Visor de Eventos" (Panel de Control) para verificar entradas de logon desconocidas, fallos de servicios, o reinicios. Si nuestro firewall o enrutador escribe registros a una distinta localización a la del sistema comprometido, debemos recordar verificar estos registros también.

Pero hay que tener en cuenta que esto no es seguro a menos que se registre para añadir sólo medios de comunicación; muchos intrusos corrigen archivos de registro en una tentativa de ocultar su actividad.

3.2.2. VERIFICAR SI EXISTEN CUENTAS Y GRUPOS DE USUARIOS DESCONOCIDAS.

Verificar si existen cuentas y grupos de usuarios desconocidas. Puede utilizar la herramienta "User Manager" o los comandos **"net user"**, **"net group"** y **"net localgroup"** en la línea de comando. Asegúrese que la cuenta incorporada *"Invitado"* no está disponible si el sistema no requiere el acceso del huésped.

3.2.3. BUSCAR MEMBRESÍAS DE GRUPO INCORRECTAS.

Se debe verificar en todos los grupos si existen membresías de usuario inválidas. Algunos de los grupos NT predefinidos dan privilegios especiales a los miembros de esos grupos. Los miembros del grupo de "Administradores" pueden hacer cualquier cosa en el sistema local. Los "operadores auxiliares" pueden leer cualquier archivo en el sistema y los "PowerUsers" pueden crear partes.

3.2.4. BUSCAR DERECHOS DE USUARIO INCORRECTOS.

Buscar derechos de usuario incorrectos. Para examinar los derechos de usuario utilice la herramienta "User Manager" y dentro de ella "políticas" y "derechos de Usuario". Hay 27 derechos diferentes que puede asignarse a usuarios o grupos. Generalmente la configuración predefinida para estos derechos está segura.

3.2.5. VERIFICAR SI EXISTEN APLICACIONES DESAUTORIZADAS DE INICIO.

Otro paso en la búsqueda de posibles intrusiones es chequear si aplicaciones desautorizadas están iniciándose. Hay un número de diversos métodos que un intruso podría utilizar para iniciar un programa de puerta trasera. Debemos asegurarnos de:

- ✓ Verificar las carpetas de inicio: se debe verificar todos los items en las carpetas c:\winnt\profiles*\start menu\programs\startup. También se puede examinar todos los accesos directos seleccionando Start, Programs, Startup. Hay que tener en cuenta que existen dos carpetas de inicio, una para el usuario local y una para todos los usuarios. Cuando un usuario entra, todas las aplicaciones en ambos lugares ("Todos los Usuarios" y las carpetas de inicio de usuario) se inician. Debido a esto es importante verificar todos las carpetas de inicio para saber si existen aplicaciones sospechosas.

- ✓ Verificar el registro: Las localizaciones más comunes para las aplicaciones de inicio a través del registro son:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SessionManager\KnownDLLs
HKEY_LOCAL_MACHINE\System\ControlSet001\Control\SessionManager\KnownDLLs
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Current Version\Run
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Current Version\RunOnce
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Current Version\RunOnceEx

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows
("run=" line)
HKEY_CURRENT_USER\Software\Microsoft\Windows\Current Version\Run
HKEY_CURRENT_USER\Software\Microsoft\Windows\Current Version\RunOnce
HKEY_CURRENT_USER\Software\Microsoft\Windows\Current Version\RunOnceEx
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices
HKEY_CURRENT_USER\Software\Microsoft\WindowsNT\CurrentVersion\Windows
("run=" value)
```

- ✓ Verificar si existen servicios inválidos: Algunos programas "puertas traseras" se instalarán por si mismos como un servicio que se inicia cuando el sistema inicia. Los servicios pueden entonces ejecutarse con el "Logon as Service" de los derechos del usuario. Se debe verificar los servicios que son iniciados automáticamente y tenemos que asegurarnos que son necesarios. También hay que verificar que los servicios no ejecuten un programa caballo de Troya o clandestino.

- ✓ El siguiente archivo por lotes ayudará a recopilar la información sobre los servicios NT que se ejecutan en un sistema de registro. La salida listará la clave del servicio, el valor inicial, y el archivo ejecutado. Este archivo por lotes utiliza el comando REG.EXE, que es parte del Kit de Recursos de NT. Los archivos y el registro no son modificados con este archivo por lotes.

```
@echo off
REM The 'delims' parameter of PULLINFO1 and PULLINFO2 should be a single TAB.

for /f "tokens=1 delims=[]" %%I in
('reg query HKLM\SYSTEM\CurrentControlSet\Services')
do
call :PULLINFO1 %%I
set START_TYPE=
goto :EOF
```

```
:PULLINFO1
for /f "tokens=3 delims= " %%I in
('reg query HKLM\SYSTEM\CurrentControlSet\Services\%1 ^| findstr "Start" ')
do
call :PULLINFO2 %1 %%I
oto :EOF

:PULLINFO2
for /f "tokens=3,4 delims= " %%I in
('reg query HKLM\SYSTEM\CurrentControlSet\Services\%1 ^| findstr "ImagePath" ')
do
call :SHOWINFO %1 %2 %%I %%J
goto :EOF

:SHOWINFO
if /i {%2}=={0} set START_TYPE=Boot
if /i {%2}=={1} set START_TYPE=System
if /i {%2}=={2} set START_TYPE=Automatic
if /i {%2}=={3} set START_TYPE=Disabled
if not "%4" == "" (echo %1 -%START_TYPE%- %3\%4)
else (echo %1 -%START_TYPE%- %3)
goto :EOF
```

3.2.6. VERIFICAR LOS SISTEMAS BINARIOS.

Este apartado se refiere a verificar nuestros sistemas binarios para saber si existen alteraciones. Hay que comparar las versiones en los sistemas con las copias buenas conocidas, como aquellas de la instalación inicial de los medios de comunicación. Tenga cuidado de confiar en copias de seguridad; sus copias de seguridad pueden también contener caballos de Troya.

Los programas caballo de Troya pueden producir el mismo tamaño y timestamp del archivo como en la versión legítima. Debido a esto, verificar solamente las propiedades de los archivos y los timestamps

asociados a los programas no es suficiente para determinar cuales de los programas han sido reemplazados, como en el caso de Unix se puede utilizar MD5, Tripwire, y otras herramientas criptográficas del checksum para detectar estos programas caballo de Troya, (suministrando las herramientas del checksum se mantiene la seguridad y no están disponibles para ser modificadas por el intruso). Además, se puede considerar el utilizar una herramienta (PGP, por ejemplo) para "señalar" la salida generada por MD5 o Tripwire, para una referencia futura.

Utilizar un software antivirus también le ayudará a verificar los virus de computadora, puertas traseras, y programas caballos de Troya.

3.2.7. VERIFICAR LA CONFIGURACIÓN Y ACTIVIDAD DE LA RED.

Verificar un sistema y su configuración de red para entradas no autorizadas es otro de los puntos importantes para detectar un intrusos. Lo que se debe hacer es buscar las entradas inválidas para configurarlas como WINS, DNS, envíos IP, etc ya que estas configuraciones pueden ser verificadas usando la herramienta “Propiedades de Red” o usando el comando "**ipconfig /all**" en la línea de comando.

Debemos asegurarnos de que sólo los “Servicios de Red” que quiera ejecutar en su sistema estén listados en la configuración de los Servicios de Red.

Además también hay que verificar si existen accesos desconocidos que escuchan conexiones de otros hosts utilizando el comando "**netstat -an**" cuya salida es similar al comando de Unix (Ver apartados anteriores).

A continuación se va a mostrar un archivo por lotes (**.bat**) que se encarga de analiza puertos que están en un estado de escuchar y muestra que servicio puede ser ejecutado en ese puerto. Este archivo

por lotes utiliza números de puertos bien conocidos y además hay que asegurarse de reemplazar la palabra "TAB" con un tabulador real. Este archivo no modifica o escribe a ningún archivo. Requiere de un archivo llamado "port-numbert.txt." Este archivo lista los números de puertos así como también los posibles servicios que se escuchan en ese puerto.

```
@echo off
for /f "tokens=1,2 delims=:" %%I in ( 'netstat -an ^| findstr "0.0.0.0:[1-9]"' ) do
call :CLEAN %%I %%J
goto :EOF

:CLEAN
set X=0
for /f "tokens=1,2,3 delims=TAB " %%A in ('findstr /I "\<%3/%1\>" port-numbers.txt')
do call :SETUP %%A %%C %3 %1
if %X% == 0 echo %3/%1 ***UNKNOWN***
goto :EOF

:SETUP
echo %3/%4 %1 %2
set X=1;
goto :EOF
```

3.2.8. VERIFICAR SI EXISTEN PARTES DESAUTORIZADAS.

Podemos usar el comando "**net share**" en la línea de comando para utilizar la herramienta “Server Manager” y de esta manera listar todas las partes de nuestro sistema. NT proporciona una manera de mostrar las partes ocultas agregando un '**$**' al final del nombre de la parte. Hay pocos nombres de partes predefinidos que NT utiliza (como PRINT$), pero si no está compartiendo una impresora con otros usuarios, se debe verificar para ver por qué esta parte fue creada.

Si se advierte un nombre sospechoso de parte las herramientas le mostrarán la localización real en el sistema que está siendo compartido. Un drive o directorio puede tener múltiples nombres de parte. Cada una de estas partes puede tener diferentes permisos asociados a ellas.

3.2.9. EXAMINAR TRABAJOS EJECUTADOS POR EL SERVICIO DEL SCHEDULER.

Hay que verificar si existen algunos trabajos programados en ejecución. Los intrusos pueden dejar puertas traseras en archivos que son programados para ejecutarse en un tiempo futuro. Esta técnica puede permitirle a un intruso regresar al sistema (incluso después de que cree que había direccionado el compromiso original). También, se debe comprobar que todos los archivos/programas referenciados (directamente o indirectamente) por el scheduler (planificador) y el trabajo de los mismos, tengan permisos de escritura para todos los usuarios.

Para verificar si exiten trabajos pendientes actualmente utilice el comando **"at**" o la herramienta **WINAT** del kit de recursos de NT.

3.2.10. VERIFICAR SI EXISTEN PROCESOS DESAUTORIZADOS.

Podemos utilizar la herramienta **Task Manager** o el comando **pulist.exe** del kit de recursos NT en la línea de comando para recopilar la información acerca de los procesos que se ejecutan en su sistema. **pulist.exe** y **tlist.exe** están incluidos en el kit de recursos NT.

Con el comando **pulist**, puede ver quién empezó cada proceso. Los servicios son normalmente asociados con la cuenta del SISTEMA.

El comando **tlist** con el flag -t le mostrará que procesos iniciaron procesos pequeños.

3.2.11. BUSCAR POR TODAS PARTES ARCHIVOS INUSUALES U OCULTOS.

Los archivos inusuales o ocultos se pueden utilizar para ocultar las herramientas y la información (programas que crakean passwords, archivos de password de otros sistemas, etc.). Los archivos ocultos se pueden ver con el explorador de NT. Seleccione *View*, *Options*, *Show all Files*. Para ver los archivos ocultos escriba en la línea de comando **"dir /ah"**.

3.2.12. VERIFICAR SI EXISTEN PERMISOS ALTERADOS EN ARCHIVOS O EN CLAVES DE REGISTRO.

La parte de asegurar propiamente un sistema NT es para establecer los permisos apropiados en los archivos y claves de registro de modo que los usuarios desautorizados (backdoors o keyloggers) no puedan cambiar archivos del sistema. Para controlar varios archivos a través de su árbol de directorio puede utilizar el programa **XCACLS.EXE** que es parte del kit de recursos de NT. El NT "Security Configuration Manager" puede ser utilizado para analizar su sistema contra una configuración que ha definido previamente. Esto ayudaría a determinar que pudo haber sido modificado.

3.2.13. VERIFICAR SI EXISTEN CAMBIOS EN POLÍTICAS DEL USUARIO O DE LA COMPUTADORA.

Las políticas se utilizan en sistemas NT para definir una amplia variedad de configuraciones y se pueden utilizar para controlar que pueden y que no pueden hacer los usuarios. Puesto que un número de ítems se configuran en el editor de política (**poledit.exe**) se recomienda guardar una copia actual de las políticas que usted crea, en caso de que se alteren y se necesite determinar cual fue cambiada.

3.2.14. ASEGURAR QUE EL SISTEMA NO SE HA MOVIDO A UN WORKGROUP O DOMINIO DIFERENTE.

Debemos asegurarnos de que el sistema no se ha movido a un Workgroup o Dominio diferente. Un intruso puede intentar tener el acceso del Administrador de Dominio de una estación de trabajo cambiando el dominio actual a un dominio en el que el intruso tiene el control.

3.2.15. EXAMINAR TODAS LAS MÁQUINAS EN LA RED LOCAL.

Cuando se investigue las señales de intrusión, se debe examinar todas las máquinas en la red local. La mayoría de las veces, si un host ha sido comprometido, otros en la red también lo están.

IV. SISTEMAS DE DETECCIÓN DE INTRUSOS

La detección de intrusos es una tecnología que intenta identificar intrusiones que sean realizadas contra una red de computadoras. Para descubrir posibles intrusiones en su sistema informático, un administrador puede hacer uso de un sistema de detección de intrusos (IDS's).

Para realizar su labor, muchos IDS's basan sus operaciones en el análisis de un seguimiento realizado sobre el sistema operativo. Los datos así obtenidos constituyen una "huella" del uso del sistema a lo largo del tiempo. A partir de esta información, los IDS's calculan métricas sobre el estado global del sistema y deciden si en un determinado momento el sistema está sufriendo algún tipo de intrusión. Los IDS's también pueden realizar su propio sistema de monitoreo, manteniendo un conjunto de estadísticas que ofrecen un perfil del uso del sistema. Las estadísticas citadas pueden ser obtenidas de varias fuentes como pueden ser: el uso de la CPU, las entradas y salidas a disco, el uso de memoria, las actividades realizadas por los usuarios, el número de logins intentados, etc. Estos datos deben ser actualizados continuamente para reflejar el estado actual del sistema y, a partir de un modelo interno, el IDS determinará si una serie de acciones constituyen una intrusión o un intento de intrusión. El modelo interno mencionado puede describir un conjunto de escenarios de intrusión o posibles perfiles de un sistema sin intrusiones.

Una posible clasificación de los IDS's existentes hoy en día podría ser la siguiente:

- ✓ **IDS basados en hosts:** Este tipo de IDS's monitorizan log's de eventos de actividades sospechosas procedentes de diversas fuentes. Estas herramientas son especialmente útiles para detectar intrusiones iniciadas por usuarios habituales en el sistema o

usuarios que se infiltran a través de la red. Los fallos son detectados muy rápidamente lo que hace a estas herramientas de detección muy populares. Abacus Project, Kane Secure Enterprise KSE, RealSecure OS Sensor o Intruder Alert son ejemplos de productos de este tipo.

- ✓ **IDS basado en red (NIDS):** Básicamente es un sniffer –monitoriza el tráfico de la red- y además detecta tráfico no deseable (actuando en consecuencia). Algunos productos de este tipo son snort, Defense Worx IDS, Network Flight Recorder, RealSecure, SHADOW.

- ✓ **IDS híbrido:** Consiste en la combinación de los dos anteriores proporcionando una máxima cobertura, no obstante, esto supone un gasto importante, por lo que se suele reservar para servidores críticos. En el futuro se estima que serán los utilizados. Pertenecen a este tipo productos como CentraxICE, CyberCop Monitor o RealSecure Server Sensor.

- ✓ **Honeypots:** Un honeypot es un sistema que simula uno o varios host's vulnerables, por lo que tales host's resultarán un objetivo apetecible para cualquier atacante. De este modo, si poseemos un punto vulnerable en nuestro sistema, el intruso perderá tiempo en el honeypot dándonos un margen de tiempo para resolver la parte del sistema que realmente se encuentre en estado crítico. Algunos productos son BackOfficer Friendly, Spectre, CyberCop Sting o Mantrap.

- ✓ **Verificadores de integridad de ficheros:** Cuando un sistema es atacado, el intruso a menudo alterará ciertos ficheros clave para poder acceder posteriormente y evitar que sea detectado. Ante la modificación de uno de estos ficheros, el sistema de detección de intrusos genera una alarma; después del ataque se podrá comprobar la integridad de los ficheros para conocer la extensión del dicho ataque. De este tipo de IDS's existen Tripwire, Veracity, Fcheck o chrootkit entre otros.

- ✓ **Identificación de puertos:** Examina los puertos en activo y para cada nuevo puerto encontrado, determina si es peligroso. Algunos productos de este tipo son Intrusion Vision, Intruder Alert o NetForensics

Actualmente existen sobre 88 productos IDS distintos, por lo tanto, a continuación se introducirán funciones y características generales de los dos tipos de IDS's más populares: los basados en host y los basados en red.

4.1. IDS BASADO EN RED (NIDS)

Para realizar su labor, NIDS analiza paquetes de red en bruto buscando una firma procedente de algún atacante. Para determinar si una firma pertenece a un atacante se compara con un modelo de posibles atacantes. Este reconocimiento se realiza en tiempo real y normalmente se realiza con un módulo de reconocimiento de ataques IDS. Para realizar este reconocimiento se usan cuatro técnicas que son las siguientes:

- Mediante patrones o emparejamiento de expresiones.
- Frecuencia o sobrepaso de un determinado umbral.
- Relación mutua entre pequeños eventos.
- Detección de estadísticas "anormales".

De todas ellas, la técnica más usada es la basada en patrones, también denominada *análisis de firmas* o *detección de abusos*. Dicha técnica está programada para interpretar una serie de paquetes como

un ataque. El IDS busca una subcadena dentro del flujo de datos que llega a través de los paquetes de red y cuando encuentra una cadena que se corresponde con ella, advierte de una intrusión. Por ello es de gran importancia disponer de una lista de firmas de ataques totalmente actualizada.

En el momento que descubre un ataque, reacciona ante el mismo del modo que su capacidad (variará de unos productos IDS a otros) y la política de administración del sistema le permita. Las capacidades de IDS incluyen acciones como enviar alertas a la consola, registrar eventos, enviar e-mails, eliminar conexiones (TCP reset), reconfigurar un firewall o un enrutador, o usar una "trampa" SNMP.

La mayor parte de los sistemas IDS basados en red sitúan un agente o sensor en el segmento de la red que desea ser monitorizada. Este agente enviará de vuelta los datos recogidos a la consola de la computadora encargada de monitorizar todos los agentes utilizados. Los datos intercambiados entre la consola y los agentes deben estar encriptados para prevenir un ataque de intercepción o de inhabilitación de un agente (por ejemplo un intruso podría hacerse pasar por la consola anulando la labor del agente).

La computadora que se use como IDS ha de cumplir una serie de requisitos como son tener instalado una tarjeta de red para monitorizar la red o el segmento de red que se desee mantener seguro, rapidez de procesador para procesar los datos enviados por todos los agentes en un tiempo corto y capacidad de almacenamiento en disco para mantener los log's y datos recibidos de los agentes –además del paquete de software IDS-. Las tarjetas de interfaz de red de los agentes (NIC) deben correr en modo promiscuo. Esto permite a las computadoras agentes ver todo el tráfico de paquetes en el segmento de red en el que se encuentre.

Nota: La consola y el agente pueden correr en la misma máquina, no obstante, numerosos vendedores no recomiendan esto debido a que se

puede sobrecargar los recursos del ordenador, especialmente en redes muy ocupadas.

Los agentes (o sensores) IDS deben ser situados en el/los lugar/es más adecuado/s para la detección de intrusiones en la red, pero no existe unanimidad de opinión sobre la localización más adecuada de los agentes. En determinadas empresas se prefiere la ubicación de los agentes dentro del firewall debido a que se deben detectar los ataques que entran por el firewall. En otras en cambio, se decide poner fuera del firewall (en lo que se conoce como zona delimitada o DMZ) para detectar los ataques contra el propio firewall; por supuesto, esto deja al agente en situación de poder ser atacado y su mantenimiento será probablemente muy tedioso. Muchos firewalls permiten crear túneles seguros o redes virtuales privadas (VPN) que permiten acceso seguro al agente. Normalmente habrá que configurar un túnel por el número de puerto que el IDS use para la comunicación entre agente y consola.

Otra opción puede consistir en poner un agente fuera y otro dentro de la zona custodiada por el firewall, con lo que se obtiene una gran capacidad de detección; esto permite tener pleno conocimiento de los dos “lados” del firewall. Además, también se podría saber si existe alguien fuera que no logra entrar debido a una mala configuración del firewall.

4.2. IDS BASADO EN HOST

Surgieron como una manera automatizada de examinar los log’s de los ordenadores, lo cual resulta en numerosas ocasiones definitivo para detectar hackers, de hecho, el famoso hacker kevin Mitnick fue cazado por un ingeniero que chequeó los log’s de su ordenador; sus

ataques hubiesen sido detectados tanto por el IDS basado en host's como por el basado en red.

En sistemas que corren con Windows NT, los IDS's basados en host's monitorizan el sistema, los eventos y los log's de seguridad que posee el sistema; en sistemas con Unix, monitorizan el *syslog*. Cuando se añade una entrada en el log, el IDS comprueba que dicha entrada no coincida con un patrón de ataque. Algunos IDS's también comprueban si ha habido algún cambio en determinados ficheros del sistema usando checksum's, y si es así envían una notificación. Adicionalmente, también pueden monitorizar los puertos del ordenador en el que están, tomando medidas cuando hay accesos a determinados puertos. Cuando detectan una intrusión pueden hacer varias acciones: registrar el evento causante, enviar un aviso a la consola, iniciar una "trampa" SNMP, finalizar el login del usuario o inhabilitar la cuenta del usuario.

4.3. VENTAJAS Y DESVENTAJAS DE AMBOS TIPOS DE IDS'S

Económicamente, puede resultar más conveniente IDS basado en red, puesto que una consola y un agente estratégicamente colocados pueden cubrir una red entera o el segmento de red más vulnerable. Cubrir una red con IDS basado en host implicaría instalar dicho software en todas las máquinas de la red. Esto es importante ya que los productos IDS no suelen ser nada baratos, por ejemplo Centrax para CyberSafe (para red o host) vale sobre 3000 euros, RealSecure para Internet Security System (tanto para red como host) vale unos 9000 euros.

Los IDS's basados en red pueden detectar intrusiones que pasarían inadvertidas por los IDS de host ya que los primeros examinan el tráfico de paquetes buscando actividades sospechosas (como un servicio denegado, por ejemplo), mientras que los segundos detectarían

determinadas intrusiones después de que el sistema ya hubiese sido atacado.

Los IDS's de red son más eficientes a la hora de seguir el rastro de un hacker puesto que al detectar intrusiones, las notificaciones o acciones tomadas se realizan en tiempo real. En un sistema basado en host, un intruso podría corromper los log's eliminando las entradas que registren su actividad e impidiendo así su seguimiento.

Además, los IDS's basados en host, normalmente sólo detectan las intrusiones que tienen éxito, mientras que los de red también detectan las intrusiones fallidas.

Una última ventaja de los IDS de red es que son independientes respecto a los sistemas operativos, los de host, en cambio, corren sobre la máquina que están cubriendo, lo que los hace dependientes del sistema operativo que tengan instalado.

Por otra banda, los IDS de red pueden avisar de intrusiones que en realidad no lo son, mientras que los de host sólo avisan de una intrusión cuando realmente ocurren.

Los IDS de host pueden monitorizar actividades del sistema más específicas que las monitorizadas por los de red, lo que permite detectar acceso a ficheros, intentos de instalación de ejecutables (como podrían ser caballos de Troya) o conocer las tareas de los usuarios conectados a la máquina (local o remotamente).

Además, para este tipo de IDS no es necesario ningún tipo de hardware sino que es suficiente un paquete de software que corra en el ordenador.

Por último, una característica muy importante de los IDS's de host consiste en que desempeñan bien su trabajo en redes de alta velocidad (al contrario que los IDS de red) dado que el IDS de host

reside en el ordenador y es totalmente independiente de la topología o protocolo de la red que se use.

V. CONCLUSIONES

En un principio, al comenzar a elaborar este tema nos vimos desbordados puesto que nos parecía un tema muy amplio y genérico en el que se podían tratar muchos y diversos temas relacionados. A esto hay que añadir que ignorábamos todo lo referente a seguridad en sistemas informáticos.

En esta situación, el desarrollo del trabajo se podía plantear de dos formas totalmente opuestas: profundizar en un determinado tema de la detección de intrusos que fuese de gran interés, como por ejemplo centrarnos en la explicación del uso de una herramienta IDS (Intrusion Detection System), o por el contrario, abarcar de una forma generalizada un amplio ámbito de conceptos, modos de obrar, herramientas, etc. relacionados con la detección de intrusos, sin caer, no obstante, en el error de hacer una visión superficial poco útil. Elegimos esta segunda opción puesto que la consideramos la más conveniente para nuestro aprendizaje sobre el tema. Además, creemos que es una buena manera para que cualquiera sin un gran conocimiento en detección de intrusos pueda usar este documento como punto de partida.

A lo largo del trabajo hemos introducido numerosos ejemplos de scripts y comandos de Unix/Linux, tratando siempre de explicar su utilidad de cara a la detección de intrusos, pero sin entrar en detalles del manejo de este sistema operativo puesto que se asume que el lector tiene un conocimiento básico del mismo.

5.1. CONCLUSIONES PRÁCTICAS

Las conclusiones prácticas que hemos obtenido son:

- ✓ Que Unix/Linux es un sistema operativo rico en utilidades y herramientas para conseguir un sistema seguro (aunque la seguridad total es una utopía), tanto en la integridad de su sistema de ficheros, como en la utilización de sus recursos o la gestión de sus usuarios.

- ✓ Es muy importante mantener actualizadas las herramientas de protección y detección ante posibles ataques, así como realizar con frecuencia la mayoría de los pasos indicados, hacer copias de seguridad, etc. ya que cada día surgen nuevos métodos de ataque mediante vías que uno no espera.

- ✓ Aunque pueda resultar un poco tedioso, con poco esfuerzo se puede salvar el sistema de un alto porcentaje de los posibles ataques.

- ✓ Referente a los IDS's se centró la atención en los de red (NIDS) y en los de hosts (HIDS), por ser los más populares y se indicó que el uso simultáneo de los dos ofrecía unas prestaciones de seguridad muy altas, pero que suponían un gran coste, a menudo, difícil de aceptar. No obstante, consideramos que con la disminución de coste y la mayor importancia que cada vez más adquiere la seguridad, en un futuro serán los sistemas que se impongan como sistema de detección de intrusos.

5.2. VENTAJAS

El campo de la detección de intrusos ofrece numerosas ventajas puesto que el detectar intrusos que **intentan** atacar el sistema, nos ayuda a garantizar:

- ✓ El mantenimiento de la integridad de los datos en el sistema. Esto es fundamental ya que en ocasiones los datos almacenados en el sistema pueden poseer una gran relevancia.

- ✓ El correcto funcionamiento del sistema tanto a nivel de software como de hardware.
- ✓ Disponibilidad de los recursos del sistema para todos los usuarios registrados en el mismo, puesto que se pueden detectar ataques que saturen los mismos.
- ✓ Una mayor facilidad en el mantenimiento del sistema.
- ✓ Confidencialidad de los datos. Mediante la detección de intrusos se puede asegurar con un nivel bastante alto que determinados datos sólo serán accedidos por aquellos usuarios que pueden hacerlo.
- ✓ En la actualidad existen herramientas que hacen la labor de la detección de intrusos de manera automatizada de modo que resulta muy sencillo para los administradores.

La detección de intrusos también ofrece ventajas en el caso de que se detecten intrusos cuando estos ya han atacado el sistema puesto que posibilita:

- ✓ El seguimiento de las tareas realizadas por el intruso dentro del sistema lo cual nos permite llevar el sistema a un estado seguro.
- ✓ Mediante el registro de los pasos seguidos en el sistema por el intruso se pueden descubrir posibles agujeros de seguridad posibilitando la incorporación mayor robusted al sistema.
- ✓ El descubrimiento del origen del ataque permitiéndonos tomar las medidas que se consideren oportunas (denuncia, denegación de acceso al origen del ataque, etc.).

5.3. INCONVENIENTES

Entre los inconvenientes que trae consigo la detección de intrusos se podrían citar las siguientes:

- ✓ A menudo resulta pesado realizar actividades de mantenimiento.
- ✓ Puede resultar complejo para usuarios inexpertos.
- ✓ Existen productos de IDS que pueden resultar muy caros para los interesados en su uso.
- ✓ La existencia de tantos sistemas de detección de intrusos puede dificultar la elección del IDS que mejor se adapte a las necesidades específicas de cada usuario.
- ✓ La necesidad de mantener actualizados los mecanismos de detección de intrusos puesto que de lo contrario perderán gran parte de su cometido.
- ✓ La imposibilidad de detectar el 100% de las intrusiones cometidas contra el sistema. Los intrusos pueden eliminar sus huellas de los log's, falsificar su identidad, modificar programas de detección sustituyéndolos por troyanos, etc.

5.4. POSIBLES AMPLIACIONES

Creemos que este documento constituye una buena base para introducirse en el campo de la detección de intrusos, conocer en qué consiste, métodos de llevarla a cabo, sus ventajas e inconvenientes, etc. A partir de aquí, se podrían desarrollar con una mayor profundidad cada uno de los temas tratados, en especial, en el uso concreto de varios IDS's.

VI. BIBLIOGRAFÍA

- **URL's:**

- http://bulmalug.net/: Descripción de algunas herramientas y pequeña introducción de la detección de intrusos.
- http://www.cert.org: Bastante información sobre seguridad en general.
- http://unixbsd.org/: Información de herramientas IDS.
- http://readires.es: Información sobre seguimiento de una intrusión.
- http://acm.org: Introducción a temas relacionados con los IDS.
- http://www.it-sec.de/mirrors/ids/network-intrusion-detection.html: FAQ sobre IDS's.
- http://www.informaticadelcastillo.com: Documentación sobre detección de intrusos en general.
- http://www.redhat.com/docs/manuals/linux: Algunas herramientas de detección.
- http://www.megaglobal.net/docs/sniffing/html/: Más herramientas.
- http://www.cerias.purdue.edu/coast/intrusion-detection/: Información sobre los IDS's.
- http://www.networkintrusion.co.uk/: IDS's.
- http://rr.sans.org/intrusion/: Mucha información sobre IDS.
- https://www.securehq.com/: Más sobre IDS.

- http://www.ebone.at/books/programmers/sonstiges/oreillybookself/tcpip/puis/index: Un índice de seguridad en UNIX.

- http://www.linuxsecurity.com: Más sobre detección.

- **Libros:**

- Proyecto Académicamente Dirigido: "Seguridad en servidores de Internet: Estudio de herramientas del hacker y del administrador". D. José Ramón Méndez Reboredo.