Integridad de los ficheros del sistema

Antonia Estero Botaro y Antonio García Domínguez

Seguridad y Competencias Profesionales
Curso 2009-2010
29 octubre 2009

Aunque el mantenimiento de los ficheros de registro y las herramientas de análisis pueden ayudar a capturar intrusos potenciales, no son técnicas infalibles. A veces un intruso experto puede llegar a tener un acceso suficiente al sistema y los ficheros de registro no indicar nada acerca del problema. Para tener evidencias de este tipo de ataque más sofisticado, se necesita seguir la pista del estado de ficheros importantes del sistema por si sufren cambios inesperados.

Por ejemplo, suponga que un intruso ha reemplazado la utilidad *ls* con una nueva versión que falla al listar cualquier fichero que comience con un determinado código. Esto permitiría al intruso almacenar en el disco duro ficheros de configuración que podría utilizar para irrumpir en el sistema, sin que fuésemos capaces de verlos.

1. Búsqueda de rootkits

Una vez que un intruso ha tenido acceso como **root** a tu sistema, quiere mantenerlo. Normalmente los intrusos introducen este tipo de programas utilizando un *rootkit*, una colección de programas y *scripts* diseñado para permitir al intruso tener acceso de forma continuada, incluso si se descubre la entrada inicial. Es decir, el *rootkit* le proporciona varios métodos de acceso, de forma que si se elimina uno de ellos le queden más para poder acceder. Por ejemplo, el *rootkit* denominado *lrk4*, incluye los programas modificados que se muestran en la tabla 1 en la página siguiente.

1.1. Medidas preventivas

Una buena forma de prepararse para un ataque de *rootkit* sería la siguiente:

- Hacer una copia de las utilidades críticas del sistema tales como: *ls*, *ps*, *login*, *inetd* y *find*, colocándolas en un disquete o CD. Si sospecha de su sistema, puede usar estas utilidades para explorar el sistema, en vez de las del disco duro que han podido ser modificadas.

Programa	Alteración
crontab	Oculta ciertas tareas planificadas de su salida.
du	No incluye el tamaño de ciertos ficheros ocultos cuando muestra la información sobre el uso del disco.
find	No lista ciertos ficheros ocultos.
ifconfig	No muestra información de red que podría revelar la actividad del intruso.
inetd	Permite el acceso automático a ciertos puertos de red.
killall	No mata ciertos procesos que utiliza el intruso para mantener el acceso.
login	Permite login remoto sin contraseña para el usuario **root**.
ls	No muestra ciertos ficheros ocultos.
ps	No muestra ciertos procesos creados por el intruso.

Cuadro 1: Posibles objetivos para un *rootkit*

- También es conveniente considerar la creación de copias de estas utilidades que estén enlazadas estáticamente. Es decir, que sean programas autocontenidos que no dependan de otros componentes del sistema (bibliotecas compartidas) para funcionar. La razón de este procedimiento es que algunos *rootkits* pueden modificar las bibliotecas compartidas. Si éstas han sido modificadas cualquier programa que las utilice puede no funcionar de forma adecuada. Por ejemplo, suponga que las órdenes *ls* y *find* operan solicitando una serie de ficheros de un componente de una biblioteca compartida. Si ésta ha sido alterada, las órdenes *ls* y *find* no devolverán resultados exactos, incluso si los programas no han sido alterados.

1.2. Medidas de detección

Según el sistema operativo y su versión utilizada, el conjunto de herramientas cambia considerablemente. En este apartado se mencionarán algunas de las más conocidas para distribuciones recientes de GNU/Linux.

En general, el uso de un disco de arranque de una distribución GNU/Linux o un conjunto mínimo de utilidades es una buena forma de preparar un análisis de emergencia sin necesidad de recompilar software.

chkrootkit

El paquete *chkrootkit* permite comprobar la existencia de *rootkits* en el sistema. Este paquete incluye un *script* que funciona como un antivirus, y puede informar de la presencia de un *rootkit* en nuestro sistema, aunque no puede eliminarlo. Examina los ficheros binarios del sistema para detectar unos 30 *rootkits* diferentes. Este paquete se puede obtener en `rpmfind.net`. También puede ser muy útil visitar `www.chkrootkit.org`, bajarse el paquete, y revisar algunos de los enlaces para ver cómo los intrusos pueden explotar *rootkits* para mantener su acceso no autorizado al sistema.

Gestor de paquetes RPM

Se puede utilizar la orden *rpm* para verificar la integridad de los ficheros de un paquete.

```
$ rpm −V paquete
```

Esta orden puede comprobar diferentes aspectos de los ficheros en el paquete original y los ficheros instalados en el sistema. Los aspectos que compara son:

- Tamaño del fichero.
- Permisos y tipo de fichero.
- Suma MD5.
- Números de dispositivo mayor y menor.
- Propietario.
- Grupo.
- Fecha y hora de modificación.

Cualquier cambio en algunos de estos aspectos del fichero es comunicado, aunque también nos podríamos plantear que el programa *rpm* podría haber sido modificado por el *rootkit*.

Gestor de paquetes Debian y debsums

En las distribuciones basadas en Debian, todos los paquetes incluyen resúmenes criptográficos MD5, y más recientemente, SHA-1 e incluso SHA-256. Estos resúmenes se incluyen en la lista de paquetes que publica el servidor. La propia lista tiene un resumen criptográfico, listado en la lista de listas de paquetes global del servidor. Esta lista de lista de paquetes se halla firmada con una clave GPG, para certificar que proviene de una fuente oficial. Los usuarios reciben prominentes avisos en caso de que vayan a instalar software cuyo resumen no se halle apoyado por una firma GPG del anillo de confianza, y no se aceptan los paquetes descargados automáticamente si su resumen criptográfico varía.

Para comprobar si los ficheros instalados se corresponden con los de la última versión de sus paquetes Debian, puede utilizarse *debsums*:

```
$ debsums −c
```

También podríamos ver los paquetes a los que les faltan resúmenes MD5 con:

```
$ debsums −l
```

1.3. Medidas de recuperación

Si descubre un *rootkit* en su sistema los pasos a dar serían los siguientes:

- Si es posible, desconecte el servidor de la red hasta que se haya solucionado el problema.
- Haga una copia de seguridad del sistema completo, incluyendo los ficheros del sistema operativo y los ficheros de datos. Éstos podrían ser revisados posteriormente para seguir la pista al intruso y perseguirlo si es posible.
- Reconstruya el sistema, actualizando los paquetes que se hayan dañado, o reinstalando el sistema operativo completo si fuera necesario.

2. Utilización de comprobadores de integridad

Aunque la verificación del sistema buscando un *rootkit* es una buena idea se se sospecha que alguien ha comprometido la seguridad del sistema, una forma más amplia de abordar el problema es comprobar la integridad de los ficheros del sistema. Utilidades tales como *md5sum* y *gpg*, además de la opción `-checksig` de la utilidad *rpm*, nos sirven para verificar la integridad.

2.1. Tripwire

El verificador de integridad más conocido es Tripwire, cuya versión libre se incluye en muchas distribuciones de Linux. Para utilizarlo, debemos tener un sistema del que estemos totalmente seguros, por ejemplo, uno que acabemos de instalar de CD's, antes de conectarlo a la red. Tripwire crea un especie de foto de los ficheros críticos del sistema de acuerdo con una configuración que podemos realizar.

Una vez hecho esto, ejecutaremos Tripwire cada cierto tiempo, para ver si se han producido cambios en el estado del sistema. En algunos casos, los cambios serán esperados, por ejemplo si hemos modificado el fichero `/etc/passwd` porque hemos añadidos nuevos usuarios. Cuando los cambios son normales, podemos actualizar la foto que mantiene Tripwire, de forma que estos cambios no aparezcan como problemas potenciales. Sin embargo, si se detecta un cambio inesperado, podemos ver inmediatamente qué ficheros están afectados y tomar las acciones adecuadas. En este caso, la información que proporciona Tripwire puede mostrarnos las diferencias entre el sistema original y el sistema actual. Esto puede ayudarnos a determinar cuánto daño se ha hecho y cuál es la mejor forma de recuperar el sistema.

Los ficheros de configuración de Tripwire están protegidos por una firma criptográfica basada en una frase de paso que se le proporciona durante la instalación inicial. El cifrado previene que los intrusos puedan modificar la configuración de Tripwire, o sus informes para ocultar sus actividades.

En resumen: podemos decir que Tripwire genera una base de datos, controlada por un fichero de configuración, de todos los ficheros que hay en el sistema, su suma de comprobación, etc. Si posteriormente detecta cambios en los ficheros, avisará.

Si en su sistema no está disponible esta herramienta puede conseguirla de `http://sourceforge.net/projects/tripwire/`. En 1999, Tripwire se dividió en dos proyectos: un proyecto de software propietario, dirigido por Tripwire Incorporated, y un proyecto de software de código abierto, Open Source Tripwire.

2.2. Samhain

Otro comprobador de integridad es Samhaim. Es similar a Tripwire, pero presenta varias ventajas:

- Se ejecuta como un demonio, por lo que puede informar inmediatamente de los cambios.
- Puede detectar módulos de núcleo que hayan sido cargados como parte de un *rootkit*.
- Puede operar en un ambiente cliente/servidor para proporcionar una monitorización centralizada de varios sistemas desde un puesto.
- La base de datos y los ficheros de configuración están firmados para prevenir que nadie los cambie.

Samhain está disponible en `la-samhna.de/samhain/`. Se ejecuta en muchas distribuciones de Linux y UNIX, por lo que puede tener que instalar el código fuente y compilarlo.

Para concluir con el tema de la integridad de los ficheros, vamos a comentar el paquete *binutils*. Contiene un conjunto de utilidades que nos permiten explorar el contenido de los ficheros. Por ejemplo, la orden *objdump* permite examinar el contenido de un fichero objeto byte a byte. La orden *strings* lista todas las cadenas de texto dentro de un fichero binario, incluyendo la utilidades del sistema y las bibliotecas compartidas.