Rootkits: jugando a las escondidas

Autor: Cristian Borghello, Technical & Educational Manager de ESET para Latinoamérica

Fecha: Martes 29 de abril del 2008

ESET, LLC 610 West Ash Street, Suite 1900
phone: (619) 876 – 5400, fax: (619) 437 – 7045
sales@eset.com, www.eset.com

Introducción

Actualmente, el malware contempla una gran cantidad de tipos de códigos maliciosos, diseñados específicamente para realizar una tarea puntual en el equipo infectado.

Dentro de esta categorización se suele ubicar a los rootkits como códigos dañinos capaces de ocultar acciones al sistema y/o al usuario. Sin embargo, esta definición se ajusta tan sólo en parte al concepto y el objetivo del presente es integrar los conocimientos actuales sobre este tipo de herramientas y la capacidad de los antivirus para lidiar con ellas.

Reseña histórica y definición

Antes de comenzar con los detalles técnicos, es apropiado realizar una síntesis de la evolución de los rootkits y más precisamente, del por qué de su denominación y un detalle de sus funcionalidades.

Típicamente, se llama root al usuario con mayores privilegios dentro del sistema operativo Unix o derivados. Cualquier usuario distinto de root tiene menores privilegios. Por este motivo, los permisos de este usuario son los que cualquier persona o proceso ajeno al sistema desea poder obtener para realizar acciones no deseadas.

Por esto, originalmente el término root-kit correspondía a un conjunto de herramientas que permiten el control del usuario root y de los procesos del sistema operativo, a la vez que estas aplicaciones permanecen indetectables para el mismo y por ende, para el usuario.

La "necesidad" de autodefensa de los programas dañinos llevó a que los mismos evolucionen e incorporen técnicas de ocultación conocidas como steatlth, las cuales han sido utilizadas desde siempre en otros sistemas operativos como DOS y las primeras versiones de Windows.

Por citar un caso, en los '90 un programa dañino denominado Whale ya era capaz de interceptar servicios de DOS y entregar al usuario datos falsificados de algún área del sistema.

Esta evolución de las aplicaciones dañinas llevó a que el término rootkit también fuera aplicado a aquellas versiones de Windows que permiten diferentes grados de acceso, aquellos con kernel NT (Windows NT, Windows 2000, Windows XP y Windows Vista), y no solo a los sistemas Unix.

Funcionalidades

Si bien con cambios de diseño propios de cada sistema operativo, actualmente se acepta como rootkit a cualquier aplicación que desarrolle alguna de las siguientes actividades en el sistema:

- pueda lograr acceso al sistema y mantener un control privilegiado sobre el mismo
- sea capaz de ocultar procesos, threads, directorios, archivos, handles, claves de registro, puertos, comunicaciones, etc.
- pueda tener acceso privilegiado a funciones o áreas críticas del sistema
- sea capaz de mostrar al sistema operativo una "imagen limpia" de sí mismo, creada para pasar desapercibido.

Nota: Es necesario remarcar que un rootkit en sí mismo no necesariamente tiene que ser dañino o causar daño en el sistema huésped sino que puede ser empleado por otros programas maliciosos para ocultar su presencia y así pasar mayor tiempo desapercibido, logrando sus objetivos (sean cuales fuesen los mismos).

Tipos de rootkits

Se distinguen dos tipos de rootkits: modo usuario y modo kernel. La diferencia radica en el nivel de privilegio en el que el código de cada uno es ejecutado, y del punto en el que se realiza la inyección en el sistema operativo afectado.

Los rootkits de modo usuario reemplazan o modifican programas -normalmente a través de procedimientos denominados hooking y patching- que se ejecutan en modo usuario (anillo 3) y, por ejecutarse en este nivel, son más sencillos de detectar. Son capaces de ocultar sus procesos por ejemplo, al Administrador de Tareas.

Los rootkits de modo kernel, en cambio, involucran la modificación de componentes del kernel del sistema operativo (anillo 0) que normalmente no puedan ser modificados o sobrescritos mediante un proceso sin privilegios.

Las ventajas de este último tipo de rootkit son evidentes debido a que, al ejecutarse con el mismo privilegio que las herramientas de detección, el rootkit puede utilizar mecanismos defensivos contra las herramientas de seguridad, a menos que las mismas estén preparadas para detectar y neutralizar estas acciones.

Las API, dueñas del sistema operativo

Una API (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) es el conjunto de funciones, procedimientos y bibliotecas que ofrecen acceso a ciertos servicios de propósito general. [5]

La ventaja de las API radica en que evitan al programador de nuevas aplicaciones, el desarrollo de funciones típicas del sistema operativo y que son realizadas continuamente por el mismo.

Por supuesto, estas funciones también son el blanco perfecto para cualquiera que desee controlar el sistema operativo (como los rootkits), debido a que cada vez que se invoque a una API, se estaría invocando al proceso dañino deseado.

Este proceso puede verse claramente en el caso de un keylogger en donde las APIs de manejo de teclado son interceptadas, la tecla presionada es almacenada y el usuario no nota dicha intercepción:

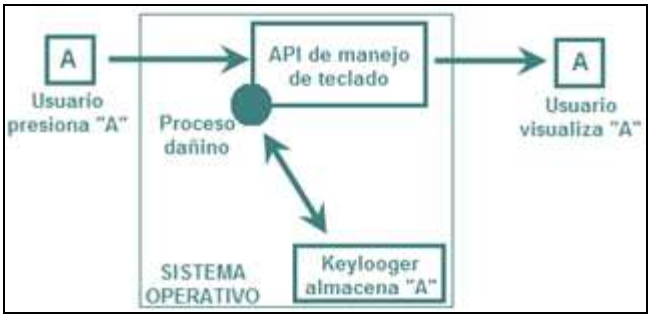

Imagen 1 – Proceso de intercepción de una API

Entonces, si bien puede parecer correcto utilizar el nombre genérico rootkit para cualquier programa dañino que se oculte, esto no es así y se debería hablar de troyanos, gusanos, adware y spyware con capacidades de ocultamiento.

De hecho, técnicas comúnmente utilizadas por los rootkits son empleadas por empresas para control de Gestión de Derechos Digitales (DRM) [1], software comercial [2] y por otras empresas para proteger sus aplicaciones [3].

Es por ello que en los últimos tiempos se prefiere llamar Programas Potencialmente Indeseables (PUP - Potentially Unwanted Programs- por sus siglas en inglés) a este tipo de aplicaciones.

Para más información sobre el origen del término, se puede consultar "Rootkits, la raíz de todos los males" [4].

La acción de tomar el control de las funciones del sistema operativo se puede realizar en modo usuario o modo kernel y abarca dos mecanismos denominados:

1. Hooking, un mecanismo por el que una función del rootkit se "engancha" con una función original del sistema operativo, redireccionando o interviniendo el flujo de operación normal [6].

Nota: Se requiere aquí la misma aclaración realizada para los rootkits, ya que "enganchar" una función no necesariamente debe hacerse con un fin dañino. Este procedimiento también es utilizado por drivers del sistema, herramientas de monitoreo y seguridad. Por ejemplo, un antivirus puede utilizar hooking para analizar cada archivo abierto por el usuario y verificar si el mismo posee capacidades dañinas o no.

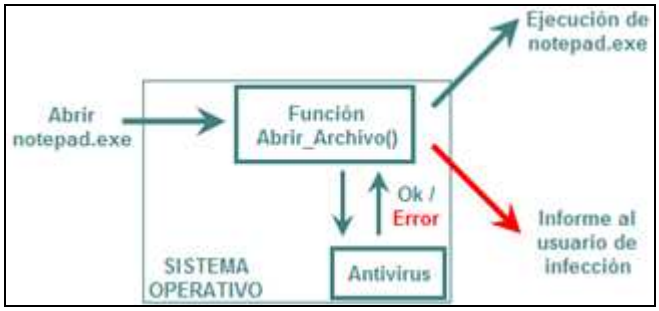

Imagen 2 – Hooking en el caso de un antivirus

2. Patching, la modificación de pequeñas partes de código de las funciones originales con el objeto de llamar a las funciones del programa intruso.

En funcionamiento

Para demostrar el funcionamiento de un rootkit se analizará el difundido Hacker Defender creado por Holy_father para sistemas basados en tecnología NT y muy utilizado por el malware actual. Dicho rootkit se compone de un archivo ejecutable desarrollado en Delphi que se ejecuta en modo usuario, y de un driver programado en C y Assembler que realiza sus funciones en modo kernel [7].

Para esta demostración se utiliza un proceso "notepad.exe" que será ocultado por el rootkit mientras su ejecución continúa normalmente. Además, se ocultarán ciertos puertos UDP (445, 500, 1900, 4500) y TCP (135, 445). En la siguiente imagen pueden apreciarse los procesos antes de la ejecución del rootkit:

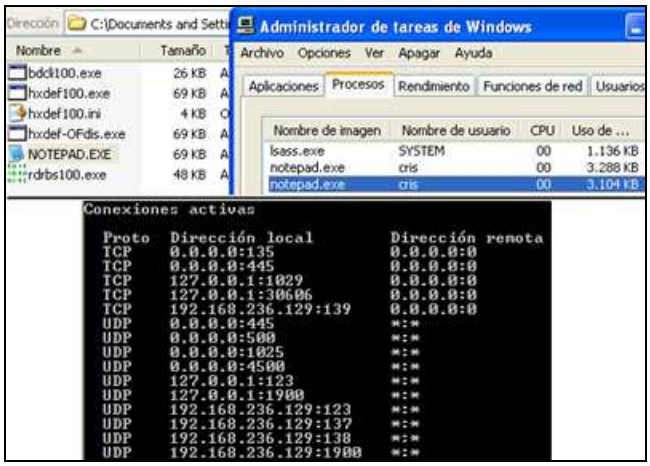

Imagen 3 – Procesos, archivos y puertos antes de la ejecución del rootkit

Luego de la ejecución del rootkit y de la configuración del mismo para ocultar el proceso "notepad.exe" y los puertos deseados, el estado del sistema es el siguiente:

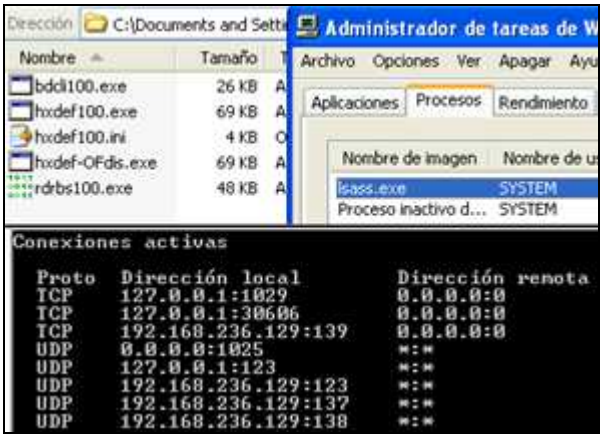

Imagen 4 - Procesos, archivos y puertos luego de la ejecución del rootkit

Como puede verse, el proceso "notepad.exe" y las conexiones a los puertos configurados han sido ocultados.

El procedimiento descripto puede ser realizado para cualquier programa y es el utilizado por el malware que desea ocultarse en el sistema.

Para hallar las actividades realizadas por el rootkit o del malware con propiedades de ocultamiento, es necesario poseer herramientas para tal fin o incluso, soluciones de seguridad con Tecnología Anti-Stealth como ESET NOD32 Antivirus y ESET Smart Security.

Para más detalle sobre el funcionamiento de este rootkit, puede consultarse el video de realizado por ESET Latinoamérica [14].

Técnicas alternativas

Así como las técnicas descriptas son utilizadas masivamente por los rootkits actuales, existen otras tecnologías que permiten el ocultamiento y camuflaje de archivos en sistemas operativos Windows.

Una de ellas es la manipulación de Flujos Alternativos de Datos (ADS, Alternate Data Streams por sus siglas en inglés) del sistema de archivos NTFS y que son utilizados para mantener información asociada con el archivo. El uso de esta técnica, por algunos virus y gusanos, data del 2000. Hablando con rigor, esta técnica no es un método para engañar al sistema operativo sino al uso de funciones APIs no documentadas y poco conocidas del mismo. De hecho cualquier usuario puede hacer uso de esta metodología para ocultar archivos [8].

Otra técnica se refiere a BootRoot, un rootkit del sector de arranque (MBR, Master Boot Record por sus siglas en inglés) y creado para BlackHat 2005 por eEye, con fines de investigación [9].

Si bien se han volcado mares de tinta a su alrededor, la utilidad práctica de esta aplicación no ha pasado a mayores y de hecho los antivirus detectan dicha amenaza desde hace tiempo bajo el nombre de Sinowal [10].

Otra situación semejante se vivió en Windows Vista cuando Joanna Rutkowska [11] presentó en BlackHat 2006 BluePill [12], un rootkit conceptual para ese sistema operativo [13].

Aún así, con respecto a los rootkits actuales, la estructura en capas y el contexto no privilegiado de Windows Vista dificulta que los rootkits tradicionales puedan ejecutar drivers a nivel de kernel.

Detección

Debido a que las técnicas utilizadas se perfeccionan día a día y se aprovechan vulnerabilidades o funciones distintas, la detección de rootkits es siempre compleja. Por ello, no es tarea trivial y

generalmente debe hacerse con herramientas diseñadas para tal fin (Anti-Rootkits) o antivirus con capacidades Anti-Stealth que permitan su detección y remoción [14].

Los mecanismos utilizados por estas herramientas consisten en analizar el sistema a un nivel muy bajo, sin utilizar las llamadas a funciones APIs mencionadas y luego hacer la misma operación utilizándolas, comparando los resultados. Si hay discrepancias, existe una aplicación que hace que estas diferencias existan, muy probablemente un rootkit.

Debido a que este método no es infalible, aquí debe seguirse un principio básico de la seguridad antivirus: la detección debe realizarse cuando el sistema operativo afectado no se encuentra en ejecución, debido a que sino no se puede asegurar que el rootkit no se encuentra bloqueando y engañando al análisis.

Conclusiones

Desde siempre, el malware ha buscado nuevas metodologías de propagación y, como en este caso, de autodefensa contra las tecnologías antimalware.

Lamentablemente, los rootkits actuales son muchos y variados y todos ellos hacen uso de diferentes metodologías e incluso se ponen públicamente a disposición de terceros que pueden intentar utilizarlos con fines delictivos.

El malware se aprovecha de cualquier ventaja que deje disponible el usuario o el sistema. En el caso particular de los rootkits, el sistema operativo es afectado por una aplicación que puede ser capaz de ejecutarse a su mismo nivel y por ende, engañar al usuario todo el tiempo necesario para realizar otras acciones dañinas.

Si bien los motivos económicos detrás del malware hacen que el mismo se perfeccione, las herramientas de detección y remoción también han evolucionado y son capaces de controlarlos.

La utilización de soluciones de seguridad con capacidades de detección de herramientas de tipo stealth y de sistemas operativos con diseño seguro y con capacidades de protección proactiva es la mejor opción a la hora de prevenir.

Más Información:

[1] "Sony, Rootkits and Digital Rights Management Gone Too Far". Mark Russinovich.
http://www.sysinternals.com/blog/2005/10/sony-rootkits-and-digital-rights.html

[2] Rootkit en software comercial
http://www.microsoft.com/technet/sysinternals/blog/2006/01/rootkits-in-commercial-software.html

[3] Symantec Admits Rootkit Usage in SystemWorks
http://www.realtechnews.com/posts/2478
http://www.eweek.com/c/a/Security/Symantec-Caught-in-Norton-Rootkit-Flap/

[4] Rootkits, la raíz de todos los males
http://www.eset-la.com/threat-center/1520--la-raiz-todos-los-males-rootkits-revelados

[5] API: Application Programming Interface
http://es.wikipedia.org/wiki/Application_Programming_Interface

[6] API Hooking Revealed
http://www.codeguru.com/Cpp/W-P/system/misc/article.php/c5667

Invisibility on NT boxes
http://vx.netlux.org/lib/vhf00.html

[7] Rootkit Hacker Defender
http://es.wikipedia.org/wiki/Hacker_Defender

[8] ADS
http://prog-asm.blogspot.com/2007/10/alternate-data-stream-caracteristicas.html

[9] BootRoot
http://research.eeye.com/html/tools/RT20060801-7.html
http://www2.gmer.net/mbr/

[10] Rootkit en el sector de arranque ¿otra vez?
http://blogs.eset-la.com/laboratorio/2008/01/14/rootkit-sector-arranque/

[11] Sitio personal de Joanna Rutkowska
http://invisiblethings.org/

[12] BluePill
http://www.bluepillproject.org/

[13] Vos también ¡Hackea Windows Vista!
http://www.segu-info.com.ar/terceros/?titulo=hackea

[14] Video detección de Rootkits
http://www.eset-la.com/threat-center/videos_educativos.php