UNAM – FACULTAD DE INGENIERÍA

SISTEMAS OPERATIVOS

2018-1

ROOTKITS: ESCONDIÉNDOSE DEL ADMINISTRADOR

Expositor:

Victor Eduardo Valdez Isidro

Profesor:

Ing. Gunnar Eyal Wolf Iszaevich

Presentación

El contenido escrito aquí es una recopilación de sitios web especializados en el tema de seguridad digital. Se agradece a los autores mantener al alcance de muchas personas esta valiosa información ☺ (Véase al final de este trabajo escrito las referencias al contenido original).

¿Qué son los *rootkits*?

Los *rootkits* son una colección de herramientas utilizadas por intrusos para mantener tanto a usuarios legítimos y administradores de un sistema comprometidos, en cuanto a seguridad y privacidad se refiere, y casi siempre se desconoce su presencia.

¿Para qué sirven?

Estas herramientas sirven para esconder procesos, y archivos que permiten al intruso mantener el acceso al sistema con fines maliciosos, esconder conexiones de red, así como accesos a *root* (control total del sistema operativo); básicamente es engañar a las propias herramientas del sistema operativo para que piensen que cierto recurso (archivo, proceso, conexión) no existe.

¿Cómo perjudican?

1) Puerta trasera o *backdoor*. Por ejemplo, si en el sistema hay una puerta trasera para llevar a cabo tareas de espionaje, el *rootkit* ocultará los puertos abiertos que delaten la comunicación; o si hay un sistema para enviar *spam*, ocultará la actividad del sistema de correo.

2) Información falsa. Los *rootkits*, al estar diseñados para pasar desapercibidos, no pueden ser detectados. Si un usuario intenta analizar el sistema para ver qué procesos están ejecutándose, el *rootkit* mostrará información falsa, mostrando todos los procesos excepto él mismo y los que está ocultando.

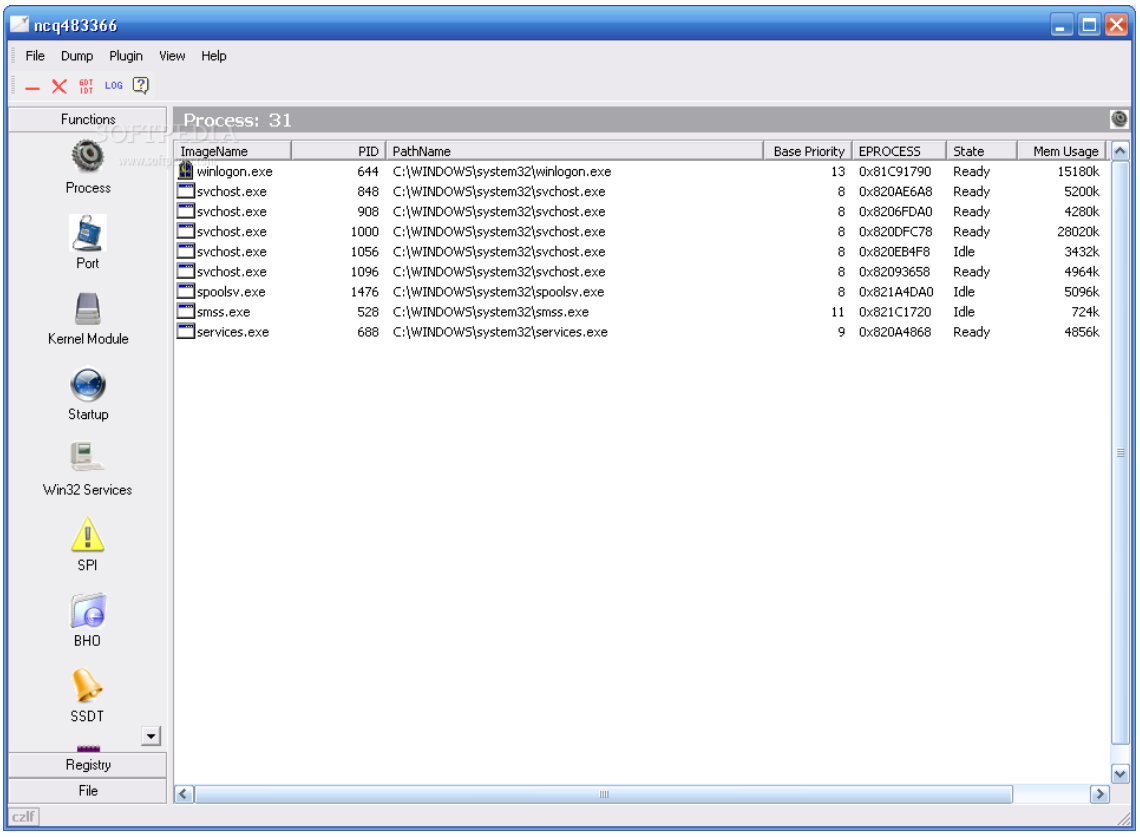

Figura. Captura de pantalla ejecutando el software Icesword Anti-Rootkit (para Microsoft Windows)

3) Camuflaje ante el software antivirus. Cuando el software antivirus haga una llamada al sistema operativo (*syscall)* para comprobar qué archivos hay, o cuando intente averiguar qué procesos están en ejecución, el *rootkit* falseará los datos y el antivirus no podrá recibir la información correcta para llevar a cabo la desinfección del sistema.

¿Cómo infecta al equipo?

Los *rootkits* se pueden instalar siguiendo varios métodos, pero el más común es aprovechando una vulnerabilidad en el sistema operativo o en una aplicación del equipo. Los ataques se dirigen contra vulnerabilidades conocidas y desconocidas en el sistema operativo y aplicaciones; usando un *exploit* que controle la computadora. Luego, se instala el *rootkit* y se configuran unos componentes que proporcionan

acceso remoto al PC. Los *exploit* se suelen alojar en una *website*, hackeada previamente. Otra forma de infección son las memorias flash USB. Los atacantes dejan memorias flash USB infectados en algún sitio donde una víctima los vea y los recoja: edificios de oficinas, cafeterías o centros de convenciones. En algunos casos, se realiza la instalación mediante vulnerabilidades de seguridad, pero en otros, se instala a partir de una aplicación o un archivo legal de un USB.

¿Cómo prevenirse de los *rootkits*?

Es necesario un sistema que vigile no únicamente la actividad de los archivos en el disco, sino que vaya más allá. En lugar de analizar los archivos byte a byte, debe vigilarse lo que hacen al ejecutarse.

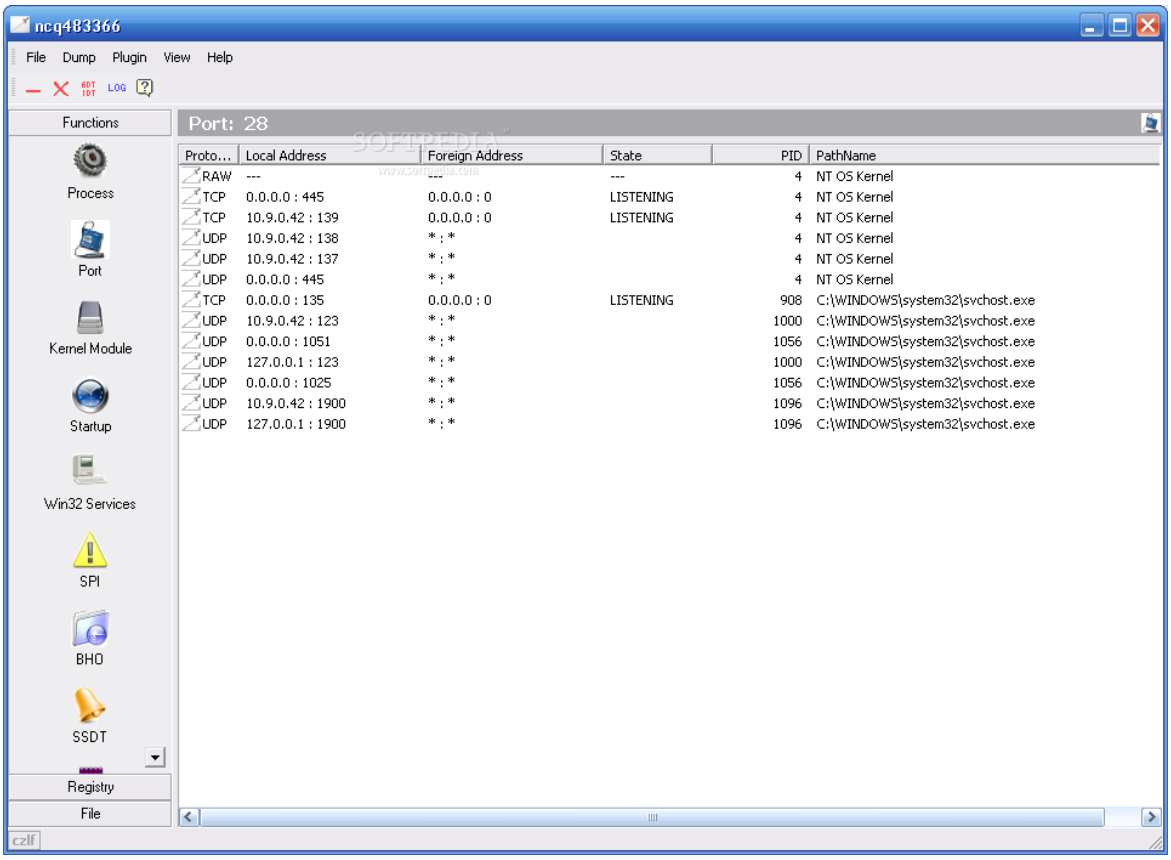

Figura. Captura de pantalla ejecutando el software Icesword Anti-Rootkit (para Microsoft Windows). Se puede ver que tenemos al proceso svchost.exe y el *status* de sus puertos en modo *listening* (ejecutandóse).

Un *rootkit* necesita llevar a cabo algunas tareas que se podrían considerar "típicas", como adquirir derechos de *superusuario*, modificar llamadas básicas al sistema operativo, falsear sistemas de reporte de datos del sistema, etcétera.

Todas estas tareas, una a una, entrañan poco peligro. Pero todas ellas, juntas y en el mismo momento, llevadas a cabo por el mismo programa, proporcionan información clara de que algo extraño está pasando en la computadora. Si las soluciones antivirus fracasan definitivamente a la hora de detectar un *rootkit*, las nuevas tecnologías de detección de amenazas por comportamiento tienen su mejor prueba de eficacia en la detección y bloqueo de *rootkits*.

Estas tecnologías no basan su funcionamiento en condicionantes previamente aprendidos sobre patrones cerrados de identificación de amenazas. Su éxito se basa en la investigación inteligente y automática de la situación de un proceso en una computadora.

Cuando una serie de acciones se llevan a cabo sobre el sistema y todas ellas (o, al menos, alguna) pueden suponer un riesgo para la integridad de la información o el correcto funcionamiento de la computadora, se evalúan una serie de factores que sirven para calificar la peligrosidad de esa tarea. Por ejemplo, que un proceso quiera tomar derechos de administración en un sistema puede ser más o menos habitual. Y tiene un cierto riesgo, sin duda, pero no hay que alertar por ello. Por ejemplo, un simple instalador para un juego puede necesitar tener derechos de administrador para poder llevar a cabo las modificaciones necesarias y poder ejecutarse correctamente.

O por ejemplo, es posible que un determinado proceso deba permanecer oculto, ya que no existe posibilidad de interacción, o que un determinado proceso abra un puerto en concreto para comunicarse, o que registre pulsaciones de teclas. Pero todas esas características juntas hacen que el proceso se pueda considerar como una amenaza y sea necesario un análisis en profundidad para poder autorizar la ejecución de manera segura.

¿Se puede eliminar un rootkit?

Los *rootkits* pueden eliminarse (aunque no tan fácilmente). Estos programas se protegen de manera automática escondiéndose y evitando que ningún otro proceso, por ejemplo, como un antivirus pueda detectarlos. Pero para que ese proceso pueda ocultarse, debe estar en funcionamiento y activado en memoria.

La mejor manera de evitar que el proceso entre en acción, es evitar el arranque del sistema operativo en el disco en el que se encuentra el *rootkit*, utilizando un disco diferente al del sistema infectado; como puede ser un *Live CD*. Así, si el *rootkit* es conocido, podrá eliminarse.

Figura. Captura de pantalla ejecutando el software Gparted (Distro GNU/Linux cargado a memoria RAM desde un *LiveCD*)

Sin embargo, si el *rootkit* no es conocido (es decir, que ha sido desarrollado específicamente para un sistema en concreto), cualquier antivirus fracasará.

Relación con el Sistema Operativo

Dominios de protección jerárquica (anillos de protección)

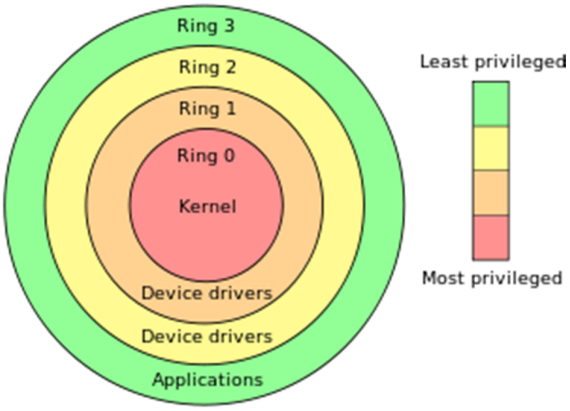

Figura. Anillos de privilegio para el procesador x86

(Según la Wikipedia) Los sistemas operativos proporcionan diferentes niveles de acceso a los recursos. Un anillo de protección es uno de dos o más niveles jerárquicos o capas de privilegios dentro de la arquitectura de un sistema de computación. Esto es generalmente impuesto por el hardware por algunas arquitecturas de CPU.

Los anillos están dispuestos en una jerarquía desde los más privilegiados (de más confianza), usualmente numerado cero, hasta el menos privilegiado (de menos confianza), usualmente con el mayor número de anillo. En la mayoría de sistemas operativos, el anillo 0 (Ring-0) es el nivel con la mayoría de los privilegios e interactúa más directamente con el hardware físico, como el CPU y la memoria.

Se proporcionan puertas especiales entre los anillos para permitir a un anillo exterior acceder a los recursos de un anillo interior de una manera predefinida, en vez de permitir un uso arbitrario. El correcto acceso por puertas entre los anillos puede mejorar la seguridad previniendo que los programas de un anillo o nivel de privilegio, usen de mala manera los recursos destinados a los programas en otro anillo.

Por ejemplo, se debe evitar que el *spyware* corriendo como un programa de usuario en el anillo 3 (Ring-3) encienda una cámara web sin informar al usuario, puesto que el acceso al hardware debe ser una función reservada del anillo 1 (Ring-1) para los controladores de dispositivos. Los programas como navegadores web corriendo en los anillos de números más altos, deben solicitar el acceso a la red, un recurso restringido a un anillo de numeración inferior.

Tipos de *rootkits*

***Rootkits* en Espacio de Usuario.** Estos se ejecutan en el anillo 3 (Ring-3), y modifican librerías, o archivos de configuración, e inclusive ejecutables (ls y ps en GNU/Linux).

Rootkits* en Espacio de *Kernel. Estos se ejecutan en el anillo 0 (Ring-0), y modifican estructuras del núcleo del sistema operativo o *kernel* y atrapan llamadas al sistema operativo (*hijacking syscall-table)*. Podemos tenerlos como LKM´s o como parches al núcleo *patch to kernel* ejecutando: */dev/kmem* en GNU/Linux.

Referencias

1. - URL: http://www.guadalajaracon.org/taller-de-desarrollo-de-rootkits-en-linux/

2. - URL: http://www.guadalajaracon.org/ponentes-2013/marcos-schejtman/

GuadalajaraCON 2013. Evento de seguridad digital realizado en Abril de 2013. Conferencia-taller de diseño de módulos de *kernel* para *rootkits* impartido por Marcos Ricardo Schejtman Rubio.

Ingeniero en Sistemas Computacionales egresado de la ESCOM-IPN. Su *hobbie* es la programación, es amante del software libre y apasionado de la seguridad. Se ha dedicado a la seguridad informática desde hace 6 años, y a sistemas reactivos desde hace 4 años.

Twitter: @NataSHell666

3. - URL: https://www.infospyware.com/articulos/que-son-los-rootkits/

4. - URL: https://mvp.microsoft.com/es-es/PublicProfile/4033573?fullName=Marcelo%20Rivero

Artículo realizado por Marcelo Rivero.

Ingeniero especializado en Malware. Ha impartido diferentes conferencias en España y Latinoamérica.

Twitter: @MarceloRivero

5. - URL: https://es.wikipedia.org/wiki/Anillo_(seguridad_inform%C3%A1tica)