Universidade Federal do Rio de Janeiro
Instituto de Matemática
Departamento de Ciência da Computação
Grupo de Resposta a Incidentes de Segurança

Rio de Janeiro, RJ - Brasil

Rootkits

GRIS-2011-A-001

Pedro Philippe Costa Rosanes

A versão mais recente deste documento pode ser obtida na página oficial do GRIS: http://www.gris.dcc.ufrj.br.

GRIS - Grupo de Resposta a Incidentes de Segurança
Av. Brigadeiro Trompowski, s/n°
CCMN – Bloco F1 - Decania
Cidade Universitária - Rio de Janeiro/RJ
CEP: 21949-900
Telefone: +55 (21) 2598-9491

Última atualização em: 21 de fevereiro de 2011

Sumário

Resumo

Nesse artigo farei uma descrição introdutória dos rootkits, passando pelas técnicas utilizadas por estes para se esconder no sistema. Descreverei um pouco da história desse tipo de ataque e citarei formas de detecção e prevenção, além de algumas ferramentas para isso. Depois mostrarei exemplos de rootkits famosos para exemplificar o conteúdo apresentado.

1 Introdução

Na maioria dos ataques a sistemas de informação, o objetivo do invasor é ter acesso ao nível administrativo. Uma vez transposta a barreira para conseguir esse privilégio, o novo objetivo é manter o poder e apagar os indícios da invasão.

Um rootkit é um conjunto de programas utilizados para impedir a detecção de atividades maliciosas no sistema, como a presença de usuários não autorizados. Costumam ser usados por invasores de sistemas para manterem acesso após um ataque bem sucedido, sem que precisem subverter o sistema novamente.

Cada rootkit possui diferentes características, mas em geral realiza alguns procedimentos como:

- Esconder informações sobre os processos referentes;
- Esconder seus arquivos;
- Esconder sockets criados para comunicação em rede;
- Modificar ou restringir o acesso aos arquivos de log;

O termo "rootkit" vem da junção de "root" e "kit". "Root", representa o chamado super-usuário, ou usuário root, em sistemas UNIX-like. Este usuário tem poder completo sobre o sistema. No Windows é conhecido como Administrator / Administrador. O termo "kit", vem do conjunto de programas que compõem o rootkit.

2 História dos rootkits

Começaram a aparecer para o público no fim dos anos 80, início dos 90. Em meados da década de 90, administradores de sistemas UNIX começaram a perceber comportamentos estranhos no computador: Espaço em disco utilizado, porém não identificado; conexões de rede que não eram listadas; uso da CPU acima do normal. Os rootkits eram bem simples, mas a partir de então passaram a evoluir muito, se instalando em lugares cada vez mais internos ao sistema, como no kernel e no bootloader.

Um caso polêmico e bastante divulgado na mídia envolvendo rootkits aconteceu em 2005, quando a SONY/BMG criou um programa para tentar impedir a pirataria de músicas e CDs. Ao inserir o CD no computador, era pedida uma permissão para instalar uma atualização do tocador de música. Porém um rootkit também era instalado. O código alterava o funcionamento do Windows, dificultando sua detecção. Então impedia-se que qualquer software diferente dos permitidos pela SONY pudesse acessar as músicas do CD.

Na listagem a seguir, estão alguns eventos importantes, em ordem cronológica:

- 1989: Primeiro alterador de logs é encontrado em sistemas corrompidos.
- 1994: Primeiros rootkits em sistemas operacionais da Sun são detectados.
- 1996: Primeiro rootkit para Linux aparece publicamente.
- 1997: Rootkits a nível de kernel (através de módulos carregáveis pelo kernel) são propostos na "Phrack".
- 1998: Rootkits que se instalam em níveis mais próximos ao hardware são propostos por Silvio Cesare.
- 2000: Rootkit a nível de biblioteca é lançado.

- 2002: Funções de “sniffers” começam a ser introduzidas em rootkits.
- 2005: SONY/BMG causam escândalos ao incluir rootkit anti-pirataria em seus CDs.
- 2006: A pesquisadora de segurança Joanna Rutkowska cria o Blue Pill, um rootkit a nível de maquina virtual em hardware.
- 2007: Mebroot, um rootkit a nível de boot, é descoberto pela empresa de segurança iDefense.

3 Tipos

Cada rootkit pode se instalar em diferentes níveis do sistema. Para cada nível, o rootkit se estabelece de uma forma diferente e requer, portanto, estratégias diferenciadas para ser detectado. À medida em que ficamos mais próximos do hardware, aumentamos o poder de controle e a complexidade da detecção.

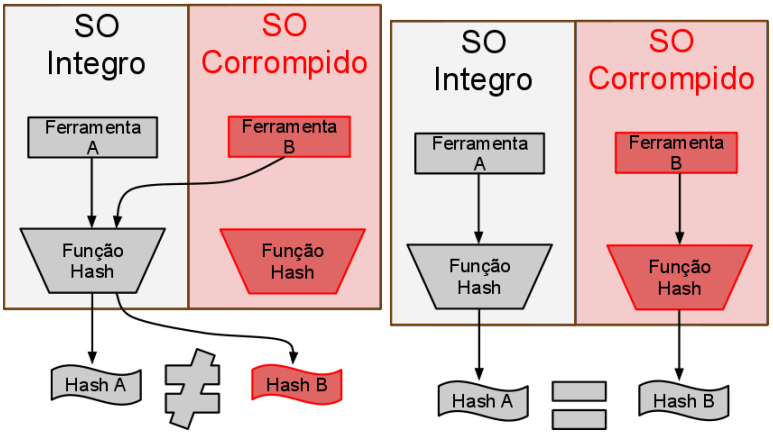
3.1 Rootkit de aplicativo (Modo usuário)

Esse tipo modifica arquivos binários básicos do sistema operacional como, por exemplo, os responsáveis pela listagem de arquivos num diretório, de processos, de conexões de rede ativas. Sua forma de instalação é trivial. Uma vez que o invasor obteve acesso privilegiado à máquina, os binários dos aplicativos a serem modificados são baixados e substituem os originais. Então ao chamar um programa como o ‘ls’ do GNU/Linux, o modificado seria executado e esta versão nova, não listaria arquivos relacionados ao rootkit.

Exemplos de binários que podem ser alterados em sistemas GNU/Linux com um determinado propósito:

- Esconder processos. Arquivo ‘ps’.
- Esconder arquivos e diretórios maliciosos. Arquivos ‘ls’, ‘dir’.
- Esconder usuários. Arquivos ‘w’, ‘who’.
- Esconder conexões de rede. Arquivo ‘netstat’.
- Permitir o aumento de privilégio no sistema. Arquivos ‘/bin/login’, ‘su’.
- Permitir acesso remoto. Arquivos ‘ssh’, ‘telnetd’.

Para detectar esses rootkits, basta comparar os aplicativos supostamente alterados com seus respectivos originais. Isso pode ser feito através do hash do seu arquivo com a de um arquivo não modificado. É importante que o arquivo original e o cálculo das hashs sejam provenientes de um sistema íntegro. Para isso, é muito comum usar um live CD da distribuição GNU/Linux em questão.

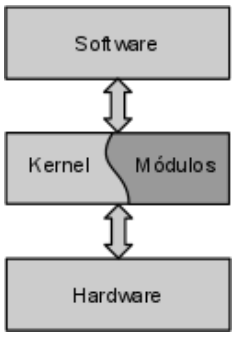


3.2 Rootkit em modo kernel (LKM/drivers)

Antes de falar sobre esse tipo de rootkit, é preciso introduzir o conceito de Kernel, Loadable Kernel Module (LKM) e drivers.

Kernel é o núcleo do sistema operacional, a camada que liga os aplicativos ao 'hardware'. É ele que controla a CPU, a memória, e os dispositivos de entrada e saída, além de realizar funções básicas de sobrevivência e manutenção do sistema, como paginação, gerenciamento de memória, escalonamento de processos e comunicação em rede. Quando um aplicativo pede (faz requisição) para imprimir um texto no terminal, por exemplo, é o kernel o responsável por enviar esses dados à placa gráfica e indicá-la que estes dados devem ser impressos na tela.

LKM são módulos conhecidos como extensões ao kernel do linux. E drivers são usados para permitir o uso de um novo 'hardware' ou sistema de arquivos. Uma vantagem é que eles só são carregados se necessário.

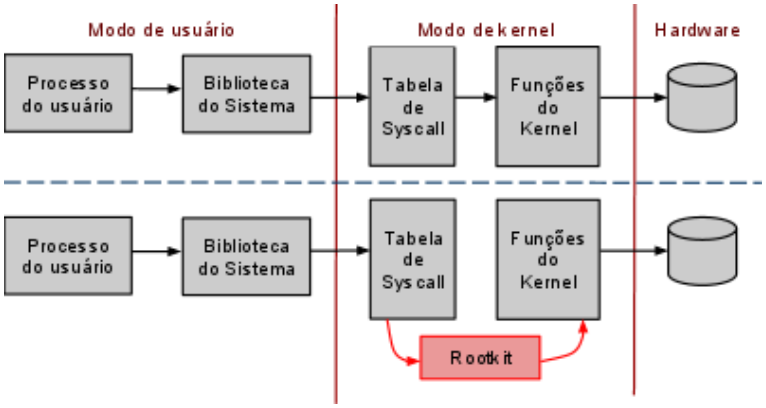


Rootkits de kernel modificam as funções de chamada ao sistema operacional (syscalls) que os aplicativos utilizam. Logo, ao invés de alterar os programas como 'ls' ou 'ps', as chamadas feitas é que são modificadas. Por exemplo, quando um programa pede para ler um arquivo, a syscall 'open()' é usada, e então o kernel permite que o processo acesse-o. O código dessa chamada está dentro do kernel, e um LKM tem a capacidade de alterá-lo, como veremos a seguir.

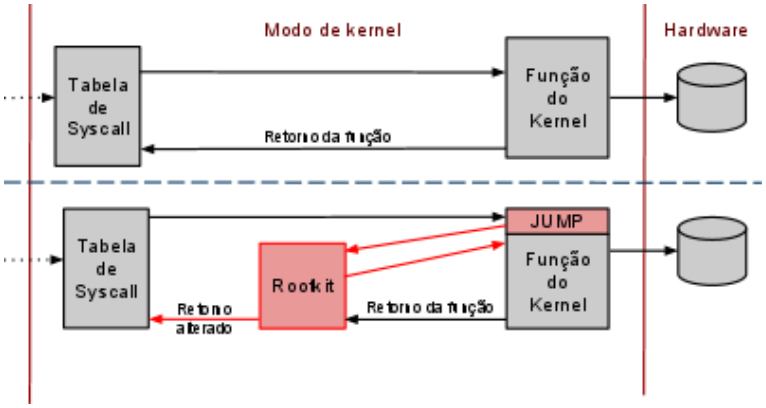
Portanto, é possível manter a integridade dos arquivos do sistema operacional, que continuam funcionando do mesmo modo. O que muda é a 'syscall'. Por exemplo, o 'ls' ao listar os arquivos, tem de acessar o disco. Ao fazer a requisição, a chamada pode ter sido alterada para não retornar arquivos e diretórios referentes ao rootkit. O problema não está mais nos programas, e sim na informação modificada que o kernel repassa.

Para que os processos se comuniquem de forma segura com as chamadas do sistema, é preciso passar pelas bibliotecas do sistema, e então são essas que fazem as 'syscall's. Para isso, elas usam uma tabela (syscall table) que mapeia o código de cada chamada.

O modo mais simples de modificar uma chamada é alterando o mapeamento da tabela. Para detectar essa alteração, pode-se executar um programa passo a passo e comparar com um arquivo criado na compilação do kernel para facilitar a depuração de processos, chamado System.map. Outra forma, é executar a ferramenta 'ktraq', que restaura todos os ponteiros da tabela.



Um método mais sofisticado é adicionar um desvio no começo do processo legítimo. O rootkit chama a função do sistema, recebe sua resposta e a altera. Assim, a modificação está no processo e não na tabela de chamadas, inutilizando a ferramenta 'ktraq'. Para detectá-lo é preciso procurar por instruções de desvio (jump) no começo das funções do kernel, mas deve-se tomar cuidado com falso-positivos.



É importante notar que os métodos de sequestro discutidos acima também podem ser aplicados às bibliotecas dinâmicas. Os programas têm tabelas com os endereços das funções externas, portanto o ataque seria análogo.

3.3 Rootkit em Master Boot Record

Master Boot Record (MBR) é um setor reservado do disco rígido, usado para carregar o sistema operacional, manter informações sobre o disco e suas partições. Ao ligar o computador, a BIOS executa o código que está nesta área e carrega um pequeno programa para por o sistema operacional na memória.

A máquina normalmente é comprometida através de uma atualização do sistema. O usuário baixa de um site não confiável um update modificado.

Durante a instalação o MBR é reescrito com o código malícioso, o original é guardado e o rootkit instalado em um setor do disco. Na próxima inicialização do computador, certas interrupções que permitem controlar tudo que é carregado pelo sistema serão feitas. Com isso o malware estará do lado de fora do sistema operacional. Proporcionando vantagens para o atacante, como:

- Controle total sobre o processo de boot. O código é executado antes mesmo do sistema operacional carregar.
- Não é preciso esconder processos, ou entradas de registro. O rootkit é carregado pelo código do Master Boot Record, fora do sistema operacional.
- Para esconder seu arquivo, só é preciso controlar setores específicos do disco.

Para verificar a integridade do código usado na inicialização do computador seria necessário acessar o modificado. Porém ao tentarmos vê-lo, o rootkit cria um desvio para o original. Uma forma de detecção é procurar por comportamentos anômalos, verificando na memória se existem processos desconhecidos. Para prevenção analise as atualizações de seu sistema antes de executá-las, e nunca baixe de sites desconhecidos.

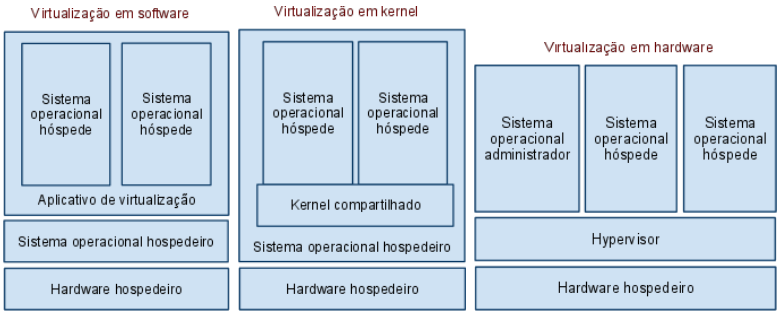
Para retirar a infecção basta executar uma recuperação de sistemas, como o "Windows Recovery Console", ou reescrever o MBR através de um LiveCD.

3.4 Rootkit em Máquina Virtual assistida por Hardware (HVM)

Máquina virtual pode ser definida como a duplicação eficiente de uma máquina real. Existem três tipos:

- Virtualização em software: um aplicativo emula todo o hardware usado pelo hóspede;

- Virtualização em kernel: os hóspedes compartilham um kernel;
- Virtualização assistida por hardware: o processador tem instruções que ajudam na simulação de uma máquina, eliminando a sobrecarga nas operações de entrada e saída.



Um rootkit que usa HVM cria um hypervisor (software que permite a virtualização) muito leve, permitindo assim diminuir os problemas de performance. Isso torna muito difícil a detecção. Ele é capaz de se instalar e migrar para a máquina virtual enquanto o sistema operacional sequestrado ainda está rodando.

Para impedir a detecção duas atitudes importantes são tomadas: Evitar a perda de eficiência em relação ao sistema original. Para isso é necessário ficar a maior parte do tempo "dormindo", e só "acordar" quando um evento interessante acontecer. Não se instalar no HD.

Por causa da segunda atitude, o rootkit não persiste se o computador for desligado. Porém, levando em consideração que os grandes servidores não são desligados com tanta frequência, a desvantagem não é tão grande. E para evitar isto, já estão tentando emular um desligamento. Isso também impede a detecção através de um sistema integro.

Blue Pill foi o primeiro rootkit em máquina virtual de hardware efetivo, criado por Joana Rutkowska como uma prova de conceito. Ou seja, demonstrou que é possivel implementar o conceito de rootkit em máquinas virtuais.

Detectar esse tipo de rookit por métodos convencionais é quase impossível. Porém como qualquer máquina virtual cria uma perda de eficiência, é possível fazer uma análise do tempo de execução de certas tarefas que "acordam" o rootkit. Mas cuidado, pois o rootkit pode facilmente alterar o software de análise para parecer que não houve mudança nas contagens. Seria necessário usar um contador de tempo externo ao computador analisado.

Como o Blue Pill não tenta esconder seu código na memória, se você souber o que está procurando, pode-se fazer uma análise na memória.

3.5 Rootkit em Firmware

Firmware é um pequeno código estático que roda em aparelhos eletrônicos, e executa tarefas bem específicas. Normalmente ficam armazenados em uma memória ROM ou flash. Nos computadores, por exemplo, a BIOS (Basic Input/Output System) deve inicializar o sistema e reconhecer os periféricos.

Esse tipo de programa é raramente modificado e checagens de integridade não são feitas com frequência. Portanto, esse é um bom local para instalar um rootkit.

Ele é instalado de forma semelhante aos que comprometem o Master Boot Record: através de um update não-legítimo, baixado de um site não confiável.

O grande problema é que o malware não é apagado mesmo com a re-instalação do sistema operacional ou com a formatação do disco.

Para certificar-se da integridade de um firmware basta fazer uma verificação da hash, tomando os cuidados apontados na secção 3.1 deste artigo. E para remover o rootkit, restaure a BIOS.

4 Rootkits conhecidos

Nesta sessão, rootkits conhecidos serão descritos e classificados de acordo com os tipos explicados nesse texto.

eEye BootRoot
Criado pela empresa eEye para mostrar o conceito de rootkit a nível de boot, tem controle sobre todo o código que carrega o Windows. Também manipula acessos ao disco, feitos pelo sistema. Altera o driver de rede NDIS.sys e pode monitorar os pacotes transferidos.

FU
Criado pelo hacker "Fuzen op", atua a nível de kernel, alterando a forma como os processos são vistos pelo Task Manager. Cada processo é representado por um objeto que contem suas características e dois 'links' para seus vizinhos. O rootkit altera essa lista encadeada. Apesar dessa mudança o processo ainda é executado, porém de forma invisível.

SubVirt
Criado por pesquisadores da Microsoft e da Universidade de Michigan para mostrar o conceito de rootkit a nível de máquina virtual. Além de controlar todo acesso ao hardware, processos maliciosos, alheios ao sistema operacional, podem ser executados na própria máquina virtual.

System Management Mode Based Rootkit
Criado por pesquisadores da "University of Central Florida" para mostrar o conceito de rootkit a nível de firmware. Altera um firmware de processadores Intel usado para controle de hardware. Esse código tem um espaço de memória reservado, e geralmente é invisível ao sistema operacional. Ele cria um keylogger e envia o que foi lido para um computador remoto, através do protocolo UDP. Não é preciso fazer nenhuma mudança no sistema operacional.

5 Ferramentas de detecção

A forma mais simples de achar um rootkit é através de ferramentas automáticas de detecção. Para UNIX existem 'chkrootkit', 'rkhunter', 'OSSEC', entre outras. Em Windows temos 'Microsoft Sysinternals Rootkit Revealer', 'avast! antivirus', 'Sophos Anti-Rootkit', 'F-Secure Blacklight', 'Radix'.

Eventualmente é preciso fazer verificações manuais, como procurar por processos ou arquivos estranhos. Mas é necessário que as ferramentas utilizadas estejam em locais seguros, como em um CD, ou que a verificação seja feita através de outro sistema operacional.

6 Conclusão

Rootkits são muito poderosos, além de criarem backdoors para invasões posteriores, têm técnicas incríveis para se esconderem. Estão se instalando em lugares mais diferentes, aumentando seus poderes de controle. O problema fundamental é que não se pode confiar no sistema corrompido para achar o rootkit.

Worms, vírus e outros tipos de malware estão cada vez mais incorporando rootkits para evitar detecção. É preciso estar atento a essa evolução, manter o sistema atualizado e utilizar ferramentas de detecção automatizadas.

7 Agradecimentos

Gostaria de agradecer ao Augusto Santos, Pedro Asad e Caroline Cabral pela grande ajuda na revisão deste artigo. E um agradecimento especial ao Bruno Buss.

8 Referências Bibliográficas

Ultima visita aos sites: 27 de Janeiro de 2011.

Referências

[1] Métodos para Detecção Local de Rootkits e Módulos de Kernel Maliciosos em Sistemas UNIX - http://www.chkrootkit.org/papers/chkrootkit-ssi2001.pdf

[2] Bios Rootkit Attacks: What's the Real Risk ? - http://adventuresinsecurity.com/blog/2006/02/01/bios-rootkit-attacks-whats-the-real-risk/

[3] Master Boot Record Rootkit is here and ITW - http://www.prevx.com/blog/75/Master-Boot-Record-Rootkit-is-here-and-ITW.html

[4] New rootkit hides in hard drive's boot record - http://www.computerworld.com/s/article/9056378/New_rootkit_hides_in_hard_drive_s_boot_record

[5] Is a Master Boot Record (MBR) rootkit completely invisible to the OS? - http://searchsecurity.techtarget.com/expert/KnowledgebaseAnswer/0,289625,sid14_gci1316951,00.html

[6] Blue Pill: The first effective Hypervisor Rootkit - http://blogs.zdnet.com/Ou/index.php?p=295

[7] Detecting the Blue Pill Hypervisor rootkit is possible but not trivial - http://www.zdnet.com/blog/ou/detecting-the-blue-pill-hypervisor-rootkit-is-possible-but-not-trivial/297

[8] Linux Kernel Rootkit - http://rootkitanalytics.com/kernelland/linux-kernel-rootkit.php

[9] There's a rootkit in the closet! - http://www.void.gr/kargig/blog/2009/08/21/theres-a-rootkit-in-the-closet/

[10] What is a Rootkit? - http://www.brighthub.com/computing/smb-security/articles/42780.aspx

[11] Hardware virtualization - http://en.wikipedia.org/wiki/Hardware_virtualization

[12] Virtual Machine - http://en.wikipedia.org/wiki/Virtual_machine

[13] Hardware Virtualization Rootkits - http://www.theta44.org/software/HVM_Rootkits_ddz_bh-usa-06.pdf

[14] Firmware - http://www.rootkitanalytics.com/firmware/

[15] Papers and conference presentations - http://invisiblethings.org/papers.html

[16] SubVirt: Implementing malware with virtual machines - http://research.microsoft.com/pubs/67911/subvirt.pdf

[17] Firmware rootkits are the latest threat - http://www.zdnet.com/blog/btl/firmware-rootkits-are-the-latest-threat/4590

[18] Security Watch: Root Kit 101 - http://reviews.cnet.com/4520-3513_7-6361348-1.html

[19] Procurando por "rootkits"em sistemas GNU/Linux - http://gris.dcc.ufrj.br/bd/tutoriais/GRIS-2005-T-001.pdf

[20] Inside Windows Rootkit - http://madchat.fr/vxdevl/library/Inside%20Windows%20Rootkits.pdf

[21] An Overview of Virtualization and VMware Server 2.0 - http://www.virtuatopia.com/index.php/An_Overview_of_Virtualization_and_VMware_Server_2.0

[22] Introducing Blue Pill - http://theinvisiblethings.blogspot.com/2006/06/introducing-blue-pill.html

[23] Introducing Blue Pill - http://theinvisiblethings.blogspot.com/2006/06/introducing-blue-pill.html