

Pega ladrão!

Não é preciso ter caras ferramentas proprietárias para praticar a arte da computação forense.
por Nils Magnus, Achim Leitner e Joe Casad

Cena do crime: a sala dos servidores... O ladrão não precisa de um cartão para entrar nela, nem mesmo da proteção das sombras – o invasor pode usar a Internet para ir e vir. Porém, apesar da entrada secreta, o agressor ainda deixa vestígios que podem desmascará-lo. Encontrar e interpretar essas evidências é a maior prioridade dos investigadores criminais.

O tema de capa deste mês explora o mundo da computação forense. Mostraremos algumas ferramentas usadas pelos especialistas para encontrar pistas, recuperar arquivos apagados e desenterrar provas escondidas. Começaremos com um estudo das ferramentas forenses do *Sleuth Kit*. Em seguida, mostraremos o *Foremost* e o *Scalpel*, duas ferramentas para encontrar e recuperar arquivos deletados. Ensinaresmos ainda a examinar discos de sistemas Windows com ferramentas do Linux, abordando, ao fim, a *Open Computer Forensics Architecture*, uma coleção gratuita de bibliotecas e ferramentas forenses desenvolvidas pela polícia holandesa.

Porém, se você não vai enfrentar um julgamento e deseja apenas capturar o invasor do sistema, talvez não seja necessário realizar uma perícia completa. As seções a seguir descrevem algumas ferramentas para encontrar invasores no sistema usando utilitários padrão do Linux.

Bem debaixo do seu nariz?

Uma das primeiras perguntas que um investigador forense deve perguntar é se a investigação deve ser realizada abertamente – o que significa que também será visível para o agressor – ou se o invasor não deve saber que está sendo investigado.

Um computador sob investigação forense é bem semelhante a uma partícula na mecânica quântica: simplesmente olhar para ela já altera seu estado.

Um agressor poderia ver o comando `ps` e, rodando o `find` no disco rígido, sobrescrever os valores de *atime* dos objetos do sistema de arquivos, eliminando provas do último acesso de um usuário.

Apesar das possíveis complicações de se trabalhar abertamente, a necessidade de chegar à raiz de atividades ilícitas às vezes é mais importante do que usar técnicas elaboradas para evitar ser notado.

Além disso, lembre-se de que a maioria dos ataques são disparados por meio de scripts e programas au-

tomatizados e que, portanto, não é comum capturar um agressor em flagrante no console. As dicas a seguir são destinadas principalmente a casos em que não seja fundamental esconder suas atividades ou deixar uma trilha de papel.

Para evitar dar pouca atenção aos detalhes, uma abordagem sistemática é muito útil. A idéia de seguir um rastro ainda quente é sempre muito sedutora, mas se não levar o investigador a algum lugar, será decepcionante.

Por exemplo, se você investigar uma lista de processos com o comando:

```
ps gauwww
```

você pode guardar a lista primeiro e consultá-la depois. O comando exibe todos os processos ativos e seus argumentos de linha de comando, com todas as opções usadas.

Obviamente, se o sistema em questão tiver sido comprometido, o invasor poderá ter instalado versões alteradas (com cavalos de tróia, por exemplo) dos utilitários do sistema, como o próprio `ps`, para esconder seus atos.

Um pequeno script de shell pode fazer o mesmo lendo dados do diretório `/proc/`.

Extensões individuais podem ser facilmente adicionadas a um script como esse e podem ser particularmente úteis se o `ps` não for mais confiável.

Para um bom teste de sanidade, é necessário verificar os resultados usando

Índice das matérias de capa

- ▶ BackTrack e Sleuth Kit 30
- ▶ Recuperação de arquivos apagados 34
- ▶ Investigação de sistemas Windows 38
- ▶ OCFA 44

uma ferramenta semelhante ao popular `pstree`. Investigadores forenses também lembram que os programas podem mudar a lista de argumentos.

Kernel

Um macete simples, como procurar processos, é inútil contra um rootkit de kernel. Os rootkits modificam o kernel para impedir que ele forneça informações sobre certos processos ao sistema `/proc/` ou outros semelhantes.

Por outro lado, é muito surpreendente como alguns invasores não se dignam a cobrir seus rastros, então pode valer a pena tentar isso.

Conexões de rede

Além dos processos, conexões de rede também podem revelar pistas, tais como o vetor de ataque e o endereço usado pelo invasor para se conectar ao sistema.

O comando `netstat --ip -pan` exibe todos os soquetes IP locais, seus protocolos (TCP ou UDP) e possivelmente os parceiros de comunicação dos soquetes conectados – a menos que o comando ou o kernel tenham sido manipulados.

Usar a opção `-n` no `netstat` impede que o DNS resolva os endereços IP, o que é uma boa idéia para evitar tráfego desnecessário na rede, pois isto levantaria suspeitas no invasor. Se for realmente necessário, sempre se pode resolver os IPs em outro momento.

Os comandos `whois` e `traceroute` exibem mais informações sobre endereços IP, as quais dificilmente podem ser forçadas pelo agressor sem a colaboração de um provedor de acesso.

Origem da conexão

Um último fator que não deve escapar do investigador forense é que a origem das conexões TCP e UDP pode ser diferente da localização real do invasor. Alguns agressores utilizam

sistemas “seqüestrados” como ponto de partida para seus ataques.

Se a conexão for originada num sistema muito próximo ao invadido, é importante tomar cuidado. Uma lista mais curta de pulos na saída do `tcpdump <destino>` será mais informativa. Se for possível eliminar as suspeitas de que se trata de um usuário comum, já haverá provas suficientemente convincentes de que um agressor remoto entrou pela rede. Um invasor próximo é muito mais perigoso do que um mais distante, pois capturar senhas na mesma sub-rede é bem mais fácil do que pela Internet).

Procurando pistas

Se o especialista forense descobrir um processo ou programa desconhecido em execução no sistema, a próxima pergunta será: “o que ele faz?”.

Ele pode estar apenas usando a sua máquina como intermediário para mais ataques, o que significa que um processo desconhecido deve criar uma entrada na lista de soquetes abertos.

Ou então, ele pode estar capturando dados na rede. Se for esse o caso, certamente haverá uma interface de rede em modo promíscuo, que pode ser detectada com o comando `dmesg`, por exemplo.

Mas o que se deve fazer ao ver um processo ativo que faça algo desconhecido? Um bom primeiro passo seria fazer um becape do próprio executável responsável pelo processo, o que é fácil com o comando `ps gauxwww` ou consultando-se o arquivo `/proc/PID/cmdline`.

O que se pode fazer caso o agressor tenha iniciado uma ferramenta, imediatamente apagado-a e sobrescrito os setores do disco? Enquanto o programa estiver em execução, há esperança – o kernel mantém um link simbólico virtual para o executável em `/proc/PID/exe`, mesmo que o agressor o tenha apagado do sistema de arquivos. Se a equipe de recupe-

ração salvar esse arquivo em algum local, provavelmente será possível analisá-lo em outro momento.

Lixo binário

Uma técnica simples, mas efetiva, é analisar o próprio binário. O comando `strings -a binário` procura caracteres imprimíveis no arquivo. Caso o programa malicioso se conecte a um servidor FTP ou Web que exija uma senha, pode ser possível encontrar a senha no código do programa. Porém, será necessária certa intuição para fazer a distinção entre as migalhas digitais e o lixo binário.

Conclusão

As estratégias simples descritas nesta introdução podem ajudar um administrador a flagrar o gatuno. Porém, se o intruso for um “profissional” experiente, ou se for necessário manter um processo formal e documentado para coleta de provas, então precisamos de algo mais.

As matérias a seguir exploram técnicas mais avançadas para quem desejar se aprofundar na área. ■

