

# Forense Computacional

0 0

October 2019 PDF TXT

**Bookmark** **Embed** **Share** **Print**

**SAVE THIS DOCUMENT**

This document was uploaded by user and they confirmed that they have the permission to share it. If you are author or own the copyright of this book, please report to us by using this DMCA report form. **Report DMCA**

## Overview

Download & View **Forense Computacional** as PDF for free.

## More details

- **Words:** 8,128
- **Pages:** 29

[Preview](#) [Full text](#)

Forense Computacional Jer nimo Pellegrini Jo o Eduardo Ferreira Bertacchi Jo o Paulo Rechi Vita 22 de Junho de 2005

Resumo Este trabalho   uma breve introdu  o ao estudo de t cnicas forenses computacionais, mantendo o foco principalmente em sistemas do tipo Unix. Algumas ferramentas s o apresentadas, sem a pretens o de oferecer uma cobertura completa do assunto.

Sum rio 1 Introdu  o 1.1 Ci ncia Forense . . . . . 1.1.1 Breve Hist rico . . . . . 1.2 Forense Computacional . . . . .

2 2 2 3

2 Aspectos Legais 2.1 Privacidade . . . . . 2.2 Legisla  o . . . . .

4 4 5

3 Metodologia 6 3.1 Etapa Pr e-Incidente . . . . . 6 3.1.1 Configura  o de hosts e infraestrutura de rede . . . . . 6 3.1.2 Time de Resposta . . . . . 7 3.1.3 Defini  o de Procedimentos . . . . . 7 3.1.4 Pol tica de Utiliza  o dos Recursos de TI . . . . . 8 3.2 Etapa P os-Incidente . . . . . 8 3.2.1 Procedimentos para coleta e preserva  o de evid ncias 8 3.2.2 Procedimentos para an lise de evid ncias . . . . . 16

1

Cap tulo 1 Introdu  o 1.1

Ci ncia Forense

A ci ncia forense trata de quest es de interesse do sistema legal. Diversas  reas da ci ncia podem estar relacionadas com essas quest es, como por exemplo a medicina para determinar a causa de uma morte, ou a qu mica para detectar res duos de determinados compostos utilizados em um crime[WPF].

1.1.1

## Breve Histórico

O primeiro caso de uso da ciência forense pode ser considerado a prova feita por Arquimedes de que uma coroa não era feita de ouro, como era declarada, através da verificação da sua densidade [WPA]. O primeiro relato que se tem do uso de impressões digitais para determinar identidades foi durante o século VII, onde as impressões digitais de devedores eram anexadas a contas, que ficavam com os credores. Essas contas eram legalmente reconhecidas como prova da validade do débito. O primeiro registro do uso da entomologia para desvendar crimes está no livro Xi Yuan Ji Lu (Coleção de Casos de Injustiça Corrigida) de Song Ci, escrito em 1248. Um dos casos conta sobre um assassinato executado com uma foice, que foi resolvido por um investigador que instruiu que todos levassem suas foices a um mesmo local. Moscas, atraídas pelo cheiro de sangue, rodearam apenas uma das foices. Diante deste fato o assassino confessou o crime. Durante o século XVI foram feitos vários estudos e avanços na medicina forense. Já em 1775 Karl Wilhelm, um químico sueco, desenvolveu um método para detectar a presença de arsênico em grandes quantidades em cadáveres. A cada dia surgem novas tecnologias nas mais diversas áreas, e essas tecnologias também são utilizadas para a execução de atividades ilícitas. Em 2

consequência disto, novas necessidades surgem na área de ciência forense, o que faz com que esteja em desenvolvimento até hoje. Algumas especialidades dentro da ciência forense, dentre várias outras, são: • Balística • Criminalística • Antropologia • Química • Odontologia • Patologia • Toxicologia • Genética • Computacional • Psiquiatria • Análise de documentos

## 1.2

### Forense Computacional

Forense computacional é o processo de investigação em equipamentos de processamento de dados—geralmente um computador pessoal, laptop, servidor, estação de trabalho ou mídia eletrônica—para determinar se o equipamento foi utilizado para atividades ilegais ou não autorizadas. Também engloba monitoramento de redes com o mesmo propósito [WPC]. Muito mais informações são mantidas em um computador do que as pessoas imaginam, o que torna muito mais difícil do que as pessoas acham remover essas informações. Por essas razões, dentre outras, a forense computacional consegue com frequência achar evidências de, ou até recuperar completamente, informações perdidas ou intencionalmente apagadas. São atribuições dos especialistas análise forense identificar suspeitos e fontes de evidências, obter e preservar evidências digitais, analisar essas evidências e apresentar um relatório com as conclusões da análise. Isto deve ser feito utilizando procedimentos padronizados e aceitos pela comunidade científica, para que as evidências sejam aceitas em uma corte. 3

## Capítulo 2 Aspectos Legais 2.1

### Privacidade

Sempre deve-se tomar muito cuidado ao acessar dados privados durante uma investigação, já que o direito a privacidade é garantido por nossa constituição federal. Principalmente se a coleta de evidência for acontecer em um servidor de arquivos de uma instituição qualquer. Neste caso, mesmo tendo-se certeza da ocorrência de um delito, existem muitos outros usuários que poder vir a armazenar informações pessoais no mesmo servidor e certamente não têm nenhuma ligação com o incidente. Devemos evitar ao máximo o recolhimento de dados pessoais que não têm ligação com a investigação. Além disso, temos que cuidar para que as informações colhidas somente estejam acessíveis ao pessoal ligado a investigação, já que todo este material, como logs e arquivos pessoais, pode ser capaz descrever com grande perfeição os hábitos das pessoas que se utilizam do servidor. Como forma de minimizar os problemas legais que a observação de dados pessoais pode vir a causar, torna-se necessário o estabelecimento de políticas de segurança bem claras que prevejam o vasculhamento de dados pessoais dos usuários em casos de incidentes de segurança. Segue abaixo uma lista de regras a serem seguidas antes de começar qualquer coleta de dados: 1. Tome cuidado de respeitar as regras de privacidade da empresa em que o sistema se encontra assim como as leis vigentes no local. Tenha certeza de que as informações coletadas não estejam disponíveis para pessoas que normalmente não teriam acesso a elas. Isto inclui arquivos de log e arquivos com dados pessoais; 2. Não invada a privacidade de alguém sem uma forte justificativa. Especialmente, não colete informações de locais que você normalmente não 4

teria razões para acessar a menos que haja indicação suficiente de que realmente ocorreu um incidente de segurança; 3. Certifique-se de que os procedimentos da empresa são compatíveis com os passos a serem adotados na coleta de evidências durante um incidente.

## 2.2

### Legislação

Atualmente, no Brasil, inexistem normas que tratem da perícia computacional e visem guiar o trabalho de um perito na área. Ao mesmo tempo, para que qualquer trabalho de perícia possa ser realizado, seja na área computacional ou não, é necessário que o perito siga uma rígida cadeia de ações que vise dar credibilidade às informações levantadas. Então cria-se a dúvida de como deve agir um perito que trabalha com computação, já que não há nenhuma norma específica que se aplique a este caso. A solução temporária encontrada foi a utilização de

normas gerais, que se aplicam a todo tipo de perícia. São feitas apenas pequenas modificações, para adequá-las a realidade do mundo computacional. A título de exemplificação, utilizaremos dois artigos extraídos do Código de Processo Penal. Art. 170 - Nas perícias de laboratório, os peritos guardarão material suficiente para a eventualidade de nova perícia. Sempre que conveniente, os laudos serão ilustrados com provas fotográficas, ou microfotográficas, desenhos ou esquemas. Deste artigo, podemos inferir que qualquer manipulação das provas, não deve modificar seu conteúdo, permitindo que uma futura análise possa ser realizada. Para que isso seja obtido, o usual é que se trabalhe sempre com uma cópia da evidência original. Outra atitude utilizada, é a obtenção de hashes criptográficos das evidências, buscando mostrar que elas não sofreram nenhum tipo de alteração desde sua coleta. Art. 171 - Nos crimes cometidos com destruição ou rompimento de obstáculo a subtração da coisa, ou por meio de escalada, os peritos, além de descrever os vestígios, indicarão com que instrumentos, por que meios e em que época presumem ter sido o fato praticado. Deste outro, notamos a importância de se coletar informações como logs e tabelas de cache do kernel. Com o auxílio destes, seremos capazes de traçar toda a linha de ação tomada pelos invasores do sistema computacional. Análises semelhantes devem ser realizadas para cada artigo, buscando garantir que as provas obtidas terão validade em um tribunal.

5

Capítulo 3 Metodologia Assim como qualquer tipo de evidência, as evidências digitais devem ser manipuladas com extremo cuidado, para que não sejam danificadas ou destruídas, o que pode ser causado por vírus, campos eletromagnéticos, choques mecânicos, eletricidade estática etc. Além disso, os procedimentos utilizados durante a análise devem ser bem documentados e reproduzíveis a partir de uma cópia das evidências originais, para que possuam valor legal. Nesta seção serão descritos diversos procedimentos de análise forense para coleta, preservação e análise dos dados, além da pré-configuração do ambiente para possibilitar uma boa análise forense.

### 3.1

#### Etapa Pré-Incidente

A preparação para resposta a um incidente de segurança deve começar bem antes da ocorrência do evento propriamente dito [dSO02] Para propiciar uma análise rápida, completa e confiável, deve existir toda uma infra-estrutura previamente configurada. A seguir serão apresentados alguns pontos que devem ser levados em conta na preparação dessa infra-estrutura.

#### 3.1.1

##### Configuração de hosts e infraestrutura de rede

Devem ser tomados cuidados especiais na configuração de serviços de rede, não apenas pensando em torná-los mais robustos, mas também fazendo com que eles passem a gerar logs e eventos passíveis de monitoramento. Se possível esses logs não devem ficar armazenados na mesma máquina que oferece o serviço, para que não possam ser alterados. O ideal é ter uma máquina responsável por receber os logs de todos os serviços, facilitando assim o correlacionamento dos mesmos. Devem ser instalados programas para analisar

esses logs, de preferência em tempo real, além de uma estrutura de monitoramento de alertas (IDS). Também é interessante ter softwares de estatísticas de uso, pois um uso incomum de banda de rede, CPU ou memória podem ser sinais de problemas. Além disso estes softwares são bastante úteis para a administração da rede, podendo ressaltar necessidades de remodelagem de elementos da rede ou da própria topologia. Outro ponto importante é a utilização de ferramentas que garantam a integridade de arquivos críticos do sistema. Isso é feito gerando-se um resumo criptográfico dos arquivos logo após a instalação, e posteriormente verificando a consistência do resumo dos arquivos do sistema em produção.

#### 3.1.2

##### Time de Resposta

Um componente importantíssimo em qualquer cenário de segurança é o Time de Resposta, responsável pela coordenação das atividades antes, durante e após um incidente de segurança. Deve ser formado por uma ou mais pessoas com conhecimentos específicos de segurança e suas responsabilidades são as seguintes:

- Participar da elaboração de uma política de segurança
- Elaborar procedimentos de resposta a incidentes
- Compreender as prioridades da organização, de forma a adaptar seus procedimentos
- Efetuar simulações de incidentes
- Treinar pessoal para reação correta em caso de incidentes, preservando assim o maior número de evidências possível
- Coordenar as atividades no caso de um incidente real

#### 3.1.3

##### Definição de Procedimentos

A definição prévia dos procedimentos para coleta, preservação e análise das máquinas afetadas é de extrema importância para a credibilidade

e rapidez da investigação. Não há tempo para que os procedimentos sejam testados, deve-se possuir um conjunto de ações pré-definidas que tentem englobar a maioria dos casos que podem ser encontrados pela frente

7

Estes procedimentos são definidos em documentos conhecidos como SOP's (Standard Operating Procedures), que descrevem como o Time de Resposta deve agir quando ocorrer um incidente. Além disso, deve ser previamente instalado e configurado um kit de resposta: ferramentas para que seja possível realizar a coleta de evidências, de acordo com os procedimentos definidos nas SOP's. Sem isso é impossível coletar dados relativos ao estado atual da máquina (memória, processos etc).

### 3.1.4

#### Política de Utilização dos Recursos de TI

Por último mas não menos importante, deve ser elaborada uma política de utilização dos recursos de TI, que aborde questões como o que é permitido e o que não é permitido a cada usuário do sistema fazer, quebra de privacidade, monitoramento de atividades etc, que deve ser assinada por todos os usuários do sistema. Essa política deve estar de acordo com a filosofia da organização. Para auxiliar nessa tarefa diversos modelos de políticas podem ser encontrados no site da SANS1.

### 3.2 3.2.1

#### Etapas Pós-Incidente Procedimentos para coleta e preservação de evidências

Agora que temos uma estrutura previamente preparada, podemos lidar corretamente com a maioria dos incidentes de segurança. Mas somente as ferramentas corretas não bastam. Devemos ter, também, bem definidos, todos os procedimentos a serem adotados. Isto porque durante um incidente, dificilmente conseguimos determinar, devido à pressão natural em tal situação, qual a melhor atitude a ser tomada, já que as ações devem ser decididas rapidamente. Temos que ter em mente que qualquer procedimento executado de forma errada pode destruir as provas obtidas ou invalidá-las perante um tribunal. Para guiar-nos neste processo, foi elaborada uma RFC que descreve uma sequência de passos a serem executados de forma a obter evidências da forma adequada. Esta seção visa cobrir os aspectos levantados por este documento, a RFC3227, dando um maior nível de detalhes. Antes de começarmos a falar sobre a coleta de dados propriamente dita, precisamos esclarecer alguns termos a serem utilizados e justificar o grande cuidado descrito no procedimento. Inicialmente começaremos falando sobre o efeito do aspecto legal sobre a aquisição de provas. 1

<http://www.sans.org/resources/policies/>

8

Todo e qualquer procedimento realizado no computador comprometido deve ser acompanhado por uma ou mais testemunhas. Deve ser criado um resumo criptográfico de todos os arquivos extraídos do computador. Deve-se evitar ao máximo fazer modificações no computador afetado. As evidências coletadas devem ser mantidas em um local seguro, com acesso controlado e deve-se poder provar que apenas um pequeno grupo de pessoas tem acesso a elas. Outra característica dos sistemas computacionais que deve ser observada antes de se iniciar o processo de obtenção de evidências é a ordem de volatilidade de cada meio em que elas se encontram. A coleta de informações deve seguir a risca a ordem de volatilidade em que os dados estão armazenados, indo dos dispositivos mais voláteis para os menos voláteis. Caso esta ordem não seja respeitada, corremos o risco de perder para sempre informações importantes para a investigação. A ordem a ser seguida é:

- Registradores e memória cache
- Tabela de roteamento, tabela arp, tabela de processos, estatísticas do kernel e memória
- Sistemas de arquivos temporários
- Disco rígido
- Log remoto e monitoramento dos dados relevantes para o sistema comprometido
- Configuração física e topologia da rede
- Mídia de backup

Devemos observar que mesmo que esta ordem seja seguida à risca, ainda é impraticável anaassim algumas informações se perderão invariavelmente. E lidar um determinado estado do computador sem fazer com que este estado se altere. Veja, por exemplo, o estado em que a memória do computador se encontra após a ocorrência de um incidente. Para podermos obter o conteúdo desta memória, por ela ser volátil, não podemos desligar a máquina. Então a única maneira acaba sendo executar um programa que leia o conteúdo desta memória. Acontece que pela simples execução de um programa, o estado da memória irá se alterar e, certamente, uma parte de seu conteúdo será sobrescrito. Enquanto uns podem imaginar isto como sendo uma razão para que tanto rigor seja desnecessário, prefiro encarar a situação da maneira inversa. Justamente por termos esta grande dificuldade é que devemos ser

extremamente cautelosos visando minimizar ao máximo o estrago causado pela obtenção destas evidências. Elas podem ser de importância fundamental para a investigação. Note que a reflexão está relacionada a algumas arquiteturas de hardware apenas. No caso de processadores Sparc, por exemplo, existe um dispositivo de hardware chamado OpenBoot firmware. Este possibilita que se faça uma dump da memória sem alterar seus dados. Visando atender as restrições acima, foi elaborada uma lista de atitudes que devem ser evitadas. Elas são:

1. Não desligar o computador até que todas as evidências tenham sido coletadas. Ao desligarmos o computador, uma boa parte das evidências pode ser perdida e, além disso, o invasor pode ter alterados scripts e serviços de inicialização e desligamento para destruir as evidências;
2. Não confiar nos

programas instalados no sistema. Eles podem ter sido modificados pelo invasor e comportar-se de forma diferente da esperada. Sempre utilize seu kit de ferramentas pré-produzido e armazenado em uma mídia não regravável; 3. Não executar programas que alterem a data de acesso aos arquivos do sistema comprometido; 4. A remoção das portas de entrada para o sistema comprometido pode acionar armadilhas deixadas pelo invasor que façam com que as evidências sejam limpas. Além disso, temos que tomar alguns cuidados quanto a violação de privacidade. Segue abaixo uma lista de regras a serem seguidas antes de coletar qualquer coleta de dados: 1. Respeitar as regras de privacidade da empresa em que o sistema se encontra assim como as leis vigentes no local. Ter certeza de que as informações coletadas não estejam disponíveis para pessoas que normalmente não teriam acesso a elas. Isto inclui arquivos de log e arquivos com dados pessoais; 2. Não invadir a privacidade de alguém sem uma forte justificativa. Especialmente, não coletar informações de locais que normalmente não teria razões para acessar a menos que haja indicação suficiente de que realmente ocorreu um incidente de segurança; 3. Certificar-se de que os procedimentos da empresa são compatíveis com os passos a serem adotados na coleta de evidências durante um incidente. 10

**Coleta dos dados** Esta seção explica o processo de obtenção de evidências em um computador que não foi desligado após o incidente. Faremos isto mostrando uma sequência de passos a serem seguidos. Anote meticulosamente cada atitude que tomar durante a coleta de evidências e tenha cuidado para que elas sejam transparentes o suficientes para poderem ser reproduzidas por outras pessoas. Conheça profundamente os processos adotados. Não se esqueça de que qualquer procedimento deve ter como premissa a menor alteração possível do sistema afetado. Visando isto, o melhor a se fazer é conectar um notebook na rede do computador afetado e redirecionar para para ele a saída dos comandos executados utilizando-se da combinação de pipe e netcat. Vejamos as etapas de forma mais detalhada: **Fotografia da tela do sistema comprometido** Esta é provavelmente a medida mais simples a ser tomada, porém não a menos importante. Fotografe a tela do sistema. Isto pode servir como prova em um tribunal. **Montagem da mídia que contém o kit de ferramentas** Esta medida é a mais problemática durante todo o processo de coleta de informações. Isto porque a única maneira de montar sua mídia no sistema comprometido será através do comando mount não confiável do sistema afetado. Um atacante pode claramente modificar este programa para que, quando ele entrar em execução, apague todas as evidências de qualquer ataque. Por simplicidade, vamos supor que não é este o caso. Podem existir outros problemas. Podemos estar sem acesso a um shell. Neste caso devemos inicialmente abrir um novo shell antes de efetuar a montagem do dispositivo. Novamente cáimos no mesmo problema descrito para o mount de se confiar em algo que não se controla. Podemos, também, não saber a senha do administrador (alterada pelo atacante, por exemplo), alguma mídia pode já estar montada no dispositivo que você deseja utilizar, a inserção de uma mídia pode ativar um processo de montagem automático, etc, e estas são somente algumas possibilidades da infinidade de casos possíveis. Quando a dificuldade para se obter evidências na fase pré-desligamento for muito grande, deve-se pesar as consequências da execução de um arquivo não confiável e do desligamento sem a aquisição inicial de dados. Pode ser que o risco de se perder informações importantes seja tão grande que seja melhor desligar o computador e trabalhar somente com a análise dos dados não voláteis. Além do risco de se executar algo não confiável, o processo de montagem também afeta o estado atual do sistema. Ele envolve a alteração de diversos dados, 11

tanto em memória quanto em disco. Abaixo temos uma lista dos arquivos modificados pela execução do comando mount, obtida através do comando `# strace /bin/mount /mnt/cdrom /etc/ld.so.cache atime /lib/tls/libc.so.6 atime /usr/lib/locale/locale-archive atime /etc/fstab atime /etc/mtab* atime, mtime, ctime /dev/cdrom atime /bin/mount atime`

Podemos evitar a modificação deste arquivo utilizando a opção -n Lembre-se que nenhum arquivo deve ser gravado no computador comprometido e para isso iremos utilizar o netcat e nosso notebook conectado à rede. (host comprometido) # mount -n /mnt/cdrom

Com este primeiro comando, montamos a unidade de cdrom que contém ` partir deste momento, mais nenhum arquivo do nosso kit de ferramentas. A computador comprometido deve ser executado. Usaremos apenas as ferramentas disponíveis em nossa mídia. Documentação da data e hora iniciais do procedimento de coleta Utilizaremos a data no formato UTC por esta depender apenas do tempo e não da localidade em que se está. Ela é, portanto, independente de fuso horário. (remoto)# nc -l -p porta > date\_compromised (comprometido)# /mnt/cdrom/date -u | /mnt/cdrom/nc porta (remoto) (remoto)# md5sum date\_compromised > date\_compromised.md5

Tabelas do S.O. armazenadas em memória Iniciamos coletando as informações contidas nas tabelas cache do sistema operacional por estas permanecerem imutáveis por um tempo curto. Neste exemplo são armazenadas as tabelas arp e de roteamento. Tabela de Mac address (endereços de enlace): (remoto)# nc -l -p porta > arp\_compromised (comprometido)# /mnt/cdrom/arp -an | /mnt/cdrom/nc porta (remoto) (remoto)# md5sum arp\_compromised > arp\_compromised.md5

Tabela de roteamento do Kernel: (remoto)# nc -l -p porta > route\_compromised (comprometido) # /mnt/cdrom/route -Cn | /mnt/cdrom/nc porta (remoto) (remoto)#md5sum route\_compromised > route\_compromised.md5

12

Conexões estabelecidas e pendentes e portas TCP/UDP abertas Neste passo, obtemos as portas TCP e UDP abertas no host comprometido. Note que poderíamos utilizar o comando cat em /proc/net/tcp e /proc/net/udp em vez do netstat. (remoto)#nc -l -p porta > connections\_compromised (comprometido)# /mnt/cdrom/netstat -an | /mnt/cdrom/nc porta (remoto) (remoto)#md5sum connections\_compromised > connections\_compromised.md5

Imagem da memória Nesta etapa iremos obter uma imagem completa da memória no instante de execução dos comandos. Podemos fazer isto de duas maneiras diferentes. Uma é através do dispositivo /dev/mem e outra através da sua representação ELF em um arquivo kcore. A vantagem do arquivo kcore é a de permitir uma análise das evidências mais fácil, já que podemos utilizar a ferramenta gdb neste processo. Podem ser necessárias outras informações, caso seja necessário analisar a memória como um todo. Podemos precisar de alguns dados relativos a tabela de páginas tais como a estrutura que mapeia a memória virtual na memória física e o tamanho de cada página. Aqui iremos fazer uma cópia do arquivo kcore de forma a facilitar uma futura análise. Nesta cópia, tanto o conteúdo da memória alocada quanto o da memória não utilizada serão gravados em um arquivo no computador remoto. (remoto)#nc -l -p porta > kcore\_compromised (comprometido)##mnt/cdrom/dd < /proc/kcore | /mnt/cdrom/nc porta (remoto) (remoto)#md5sum kcore\_compromised > kcore\_compromised.md5

Não se esqueça que esta não é uma cópia fiel da memória, já que a simples execução do software altera o estado da memória, possivelmente sobrescrevendo evidências. Existem soluções de hardware que permitem obter uma cópia fiel da memória [CG04]. Lista de módulos do kernel carregados A partir desta etapa, os procedimentos adotados procuram garantir que os dados obtidos nas etapas anteriores são confiáveis. Imagine, por exemplo, que o invasor alterou o comportamento do kernel para que seus rastros não possam ser detectados através do netstat. Precisamos ter certeza de que os programas executados anteriormente apresentaram o resultado correto e para isto iremos obter uma lista dos módulos carregados pelo kernel. Com esta lista, tentaremos identificar algum módulo suspeito, ou então, o ocultamento de algum módulo. A maneira padrão de obter esses módulos é através dos comandos abaixo. 13

```
(remoto)# nc -l -p porta > lkms_compromised (comprometido)##mnt/cdrom/cat /proc/modules | /mnt/cdrom/nc porta (remoto) (remoto)# nc -l -p porta >
lkms_compromised.md5 (comprometido)##mnt/cdrom/md5sum /proc/modules | /mnt/cdrom/nc porta (remoto)
```

Porém, pode acontecer do módulo estar escondido e não aparecer na listagem acima. Para contornarmos este problema, nos utilizamos de um pequeno truque. Iremos nos utilizar de um módulo construído para verificar a cadeia de módulos carregada no kernel. Este módulo se chama hunter.o [hun]. (comprometido)##mnt/cdrom/insmod -f /mnt/cdrom/hunter.o

A opção -f é utilizada caso o módulo não tenha sido compilado especificamente para o kernel utilizado. A opção ideal seria compilar o módulo para cada kernel a ser avaliado. Após carregarmos o módulo hunter.o, vamos salvar as informações que ele disponibiliza para comparar com os dados obtidos da maneira convencional. Esta comparação pode ser muito importante para detectar módulos escondidos. Devemos nos atentar também para o tamanho dos módulos, já que alguns códigos maliciosos se juntam a módulos já carregados. (remoto)#nc -l -p porta > modules\_hunter\_compromised (comprometido)##mnt/cdrom/cat /proc/showmodules && /mnt/cdrom/dmesg | /mnt/cdrom/nc porta (remoto) (remoto)#md5sum modules\_hunter\_compromised > modules\_hunter\_compromised.md5

Por último, vamos obter uma cópia dos símbolos exportados pelos módulos do kernel. Rootkits do tipo LKM (Loadable Kernel Module) conseguem explorar estes símbolos e, através de uma análise deles, é possível determinar a presença de tais ameaças. (remoto)#nc -l -p porta > ksyms\_compromised (comprometido)#/mnt/cdrom/cat /proc/ksyms | /mnt/cdrom/nc porta (remoto) (remoto)# nc -l -p porta > ksyms\_compromised.md5 (comprometido)#/mnt/cdrom/md5sum /proc/ksyms | /mnt/cdrom/nc porta (remoto)

**Lista de processos ativos** Nesta etapa iremos utilizar uma ferramenta chamada lsof que nos possibilita obter informações de todos os processos, portas e arquivos abertos. Evidentemente ele só é útil caso o computador comprometido esteja livre de rootkits do tipo LKM. Obteremos a lista de processos sendo executados e, analisando-a, tentaremos descobrir algum processo suspeito para ser copiado da memória no passo seguinte. (remote)#nc -l -p port > lsof\_compromised (comprometido)#/mnt/cdrom/lsof -n -P -l | /mnt/cdrom/nc (remote) port (remote)#md5sum lsof\_compromised > lsof\_compromised.md5

Algumas dicas para identificar processos suspeitos são: 14

- Processos que escutam numa porta TCP/UDP atípica ou num raw socket aberto;
- Processos que tem uma conexão ativa com uma máquina remota;
- Um programa que foi apagado depois de executado;
- Um processo que apagou algum arquivo (por exemplo, um log);
- Um processo com nome estranho;
- Um processo iniciado por um usuário que não existe ou sem os privilégios necessários.

Coleta de processos suspeitos Será utilizado o utilitário pcat para fazer a cópia de toda memória alocada por um processo escolhido. (remoto)#nc -l -p porta > proc\_id\_compromised (comprometido)#/mnt/cdrom/pcat proc\_id | /mnt/cdrom/nc porta (remoto) (remoto)#md5 proc\_ip\_compromised > proc\_ip\_compromised.md5

**Informações úteis sobre o sistema comprometido** Além de todos os dados colhidos, podemos precisar de algumas informações extras sobre o sistema comprometido. Estas informações serão necessárias para preparar uma descrição adequada do incidente e para fazer a cópia de todos os arquivos de sistema. Elas podem ser obtidas através dos comandos abaixo. Não se esqueça de direcionar a saída de cada comando para o computador remoto através no netcat. /mnt/cdrom/cat /proc/version Versão do sistema operacional /mnt/cdrom/cat /proc/sys/kernel/name Hostname /mnt/cdrom/cat /proc/sys/kernel/domainname Domínio /mnt/cdrom/cat /proc/cpuinfo Informações sobre hardware /mnt/cdrom/cat /proc/swaps Partições swap mnt/cdrom/cat /proc/partitions Sistemas de arquivos locais /mnt/cdrom/cat /proc/self/mounts Sistemas de arquivos montados mnt/cdrom/cat /proc/uptime Uptime -h à quanto tempo o sistema esta executando

**Documentação da data e hora finais do procedimento de coleta** Esta é a última etapa antes de podermos desligar o computador. Obtemos no computador comprometido a data e hora em que a coleta de informações se encerrou e enviamos para o computador remoto. (remoto)#nc -l -p porta > end\_time (comprometido)# /mnt/cdrom/date | /mnt/cdrom/nc porta (remoto)

15

**Imagens dos sistemas de arquivos** Para criarmos uma imagem do disco rígido, a melhor opção é reiniciar a máquina comprometida e fazer o boot a partir do cdrom previamente preparado. As partições dos discos rígidos comprometidos devem ser montadas no modo somente leitura pelo seu live cd para evitar que os dados sejam alterados. Neste exemplo, estamos criando a imagem do disco principal do barramento ide0. Novamente utilizamos o utilitário md5sum para gerar o hash do arquivo. (remoto)#nc -l -p porta > image\_hda (comprometido)#dd < /dev/hda | nc porta (remoto) (remoto)#md5sum image\_hda > image\_hda.md5

**Coleta de logs remotos** Caso haja logs da máquina comprometida feitos em uma máquina remota, deve-se copiar todos os arquivos de log daquela máquina. Este procedimento, por ser executado em máquina não comprometida, pode ser executado de forma mais simples, logando na máquina e copiando os arquivos (também é necessário que resumos criptográficos sejam gerados para estes logs). **Armazenamento das evidências** As evidências precisam ser guardadas com segurança e não podem ser manipuladas por qualquer um. Deve-se dar preferência a um meio de armazenamento comumente utilizado a uma forma de armazenamento pouco difundida. O acesso às evidências deve ser extremamente restrito e deve ser documentado além de ser possível detectar acesso não autorizado a elas. Deve ser feita uma documentação detalhada sobre onde, quando e por quem as evidências foram descobertas e coletadas. Deve-se também, documentar onde, quando e por quem as evidências foram manipuladas ou examinadas. Outra informação importante é quem tem a custódia das evidências, durante qual período e quando elas mudarem



de custódia, quando e como esta mudança ocorreu.

### 3.2.2

#### Procedimentos para análise de evidências

As ferramentas de análise forense tentam reconstruir os eventos que compõem um incidente através de diversas técnicas, que são mescladas de forma a se obter o resultado desejado. Durante o processo de análise, é importante que as evidências não sejam alteradas. Sempre que possível opera-se sobre uma cópia, e não sobre a evidência original. Algumas das técnicas empregadas são listadas a seguir: 16

- Análise de logs: os logs podem ajudar muito na reconstrução do incidente; é particularmente interessante que cada máquina faça seus logs remotamente em outra, para que, caso a máquina seja comprometida, não tenha seus logs apagados ou alterados. Se a máquina tem um sistema de detecção de intrusão, a análise de seus logs será ainda mais útil;
- Análise de sistema de arquivos: recuperação de partições e arquivos deletados; de acordo com o tempo de acesso dos arquivos encontrados, pode-se criar uma linha do tempo que será útil na análise;
- Criptanálise e esteganálise: informações cifradas ou escondidas em arquivos de forma a não levantar suspeitas podem ser recuperadas;
- Busca por rootkits: verificar a integridade do kernel e de programas do sistema;
- Dump e análise de processos rodando na máquina: é possível fazer um snapshot de um processo que esteja rodando e obter uma série de informações importantes sobre ele;
- Engenharia reversa de programas e processos: apesar de escondidos, arquivos executáveis podem ser encontrados no sistema de arquivos. Após encontrá-los, pode-se tentar convertê-los para assembly para tentar determinar qual era o objetivo do executável. As técnicas acima são comuns em análise forense. Além destas, há diversas outras técnicas usadas (análise de caixas de email, busca por vírus e outras). Recuperação de arquivos Determinar quais arquivos existiram em um sistema de arquivo e foram removidos ou escondidos é fundamental para o processo de análise forense. Recuperar arquivos pode envolver diversos passos:
- Recuperar a tabela de partições: há utilitários, como o gpart, que fazem uma varredura de um dispositivo procurando por sequências de bytes que correspondam a partições. Com isso pode-se recuperar uma tabela de partições de um disco rígido, e a partir dela tentar recuperar é importante notar que partições e outras áreas do disco arquivos. E podem ser escondidas, dependendo do disco e da habilidade do usuário;

### 17

- Desfazer remoções: muitos sistemas de arquivos tornam bastante fácil recuperar arquivos, uma vez que estes são apenas marcados como apagados, mas ainda estão presentes no diretório de arquivos. Se os setores ocupados pelo arquivo no disco ainda não foram sobrescritos por outros, é possível simplesmente remover o flag de "apagado" do diretório e recuperar o arquivo, ainda com seu nome e tipo originais;
- Varrer a imagem: o passo anterior não é suficiente para recuperar toda a informação que estava presente no sistema de arquivos. Deve-se também varrê-lo, procurando sinais de arquivos escondidos ou deletados. As ferramentas que realizam este tipo de trabalho procuram por assinaturas de tipos específicos de arquivos (normalmente cabeçalhos e rodapés), e os recuperam (sem o nome e tipo do arquivo originais). Determinação de tempos de acesso Para se reconstruir um incidente, é necessário que tenhamos uma linha temporal que servirá de base para o trabalho. Isso pode ser feito por ferramentas que procuram todos os arquivos do sistema, verificam seus tempos de criação, acesso e modificação (mac times), e os ordenam. Estes tempos podem ser alterados pelo invasor de forma a cobrir seus rastros, mas se não tiverem sido, podem ser úteis. Por exemplo, poderíamos usar o utilitário mac-robber [Car] para extrair os mac times dos arquivos de um diretório "test": # mac-robber-1.00/mac-robber test > test.mac

E em seguida ordenar os arquivos com o utilitário mactime (parte do SleuthKit [sle]): mactime -b test.mac 01/01/2001 > test\_timeline.01-01-2001

O resultado é mostrado a seguir: Tue Jun 21 2005 13:08:21 Tue Jun 21 2005 13:08:26 Tue Jun 21 2005 13:08:30

```
3 mac -rw-r--r-- 1001 5 mac -rw-r--r-- 1001 2 mac -rw-r--r-- 1001
```

```
1001 1001 1001
```

```
117850748 /home/x/test/b 117850749 /home/x/test/subdir/a 117850747 /home/x/test/c
```

Isso permite determinar mais claramente quais foram os passos de um invasor, e em que ordem aconteceram.

### 18

Detecção de rootkits Os utilitários de detecção de rootkits usam diversas técnicas para determinar se um sistema foi comprometido. Algumas delas são:

- Nomes de arquivos suspeitos: é comum rootkits esconderem seus arquivos de forma a não chamar a atenção. Por exemplo, em sistemas Unix, nomes arquivos começando com um ponto dentro de /usr/ e subdiretórios são considerados suspeitos. Outros nomes de arquivos e diretórios, que já se sabe serem instalados por rootkits também são verificados. Sempre que um arquivo com tais nomes for encontrado, o detector de rootkits gera um aviso;
- Assinaturas de rootkits: sequências de bytes conhecidas podem ser encontradas em arquivos, indicando que um rootkit está instalado. Esta técnica é semelhante às técnicas usadas por detectores de vírus: basta varrer os arquivos, procurando pelas

assinaturas em quest~ao; • Verifica, c~ao de interfaces de rede: uma interface de rede em modo promíscuo é forte evidência da presença de algum programa de captura de tráfego na máquina; • Altera, c~ao de logs do sistema: é possível determinar quando há um gap nos logs, e outras alterações. Por exemplo: – Um gap nos dados em wtmp em sistemas Unix, por exemplo, como se o tempo tivesse dado um “salto”; – O arquivo /var/log/wtmp registra os logins e duração da sessão de cada usuário. Um atacante pode tentar esconder seus rastros modificando este arquivo—mas, a não ser que o arquivo seja reconstruído, é possível verificar inconsistências (como entradas de duração zero, por exemplo). Análise de logs Os logs presentes em uma máquina comprometida nem sempre são confiáveis, uma vez que o atacante pode ter alterado os logs de forma a cobrir seus rastros, mas ainda assim é importante submetê-los a análise. Quando os logs estão em uma máquina diferente da que foi comprometida, têm muito mais valor. Esta análise é feita usando ferramentas de software, e há ferramentas específicas para cada tipo de log. Devido à diversidade dos tipos de log, não podemos descrever as ferramentas de análise para todos eles. 19

Esteganálise é possível esconder dados em arquivos de imagens. Por exemplo, o utilitário E steghide [steb] usa uma sequência de bits da imagem para guardar uma mensagem (a sequência de bits é determinada por um gerador de números pseudo-aleatórios, e a semente é justamente a senha que deve ser usada para decifrar a mensagem). A mensagem é incluída alterando pixels da imagem de forma não visualmente perceptível. O processo de esteganálise tenta determinar se é provável ou não que haja dados escondidos em uma imagem, usando diversos testes estatísticos. Há utilitários como o stegdetect, [stea] que realizam este tipo de análise. Criptanálise Alguém que pretenda esconder informações, inclusive do perito forense, muito provavelmente tratará de cifrar os arquivos que façam quaisquer referências às tais informações. No entanto, dependendo de quão importante as informações são, de quais recursos estão à disposição, e de como as informações foram cifradas, pode ser possível recuperá-las através de criptanálise (ou usando força bruta). A Criptanálise é algo demasiado extenso, e está fora do escopo deste trabalho. Análise de processos “vivos” Para capturar um processo em execução, uma boa ferramenta primeiro deverá evitar temporariamente que ele seja executado (por exemplo, em sistemas Unix pode-se enviar o sinal SIGSTOP), para que os dados extraídos sejam consistentes. Procede-se então à coleta de dados: • Ambiente: quais eram as variáveis de ambiente e quais eram seus valores quando o executável foi chamado? Qual é exatamente a string da linha de comando usada? Isso permite determinar se um daemon ou outro executável do sistema foi chamado com alguma opção que acarrete uma diminuição de segurança (por exemplo, desligando opções de controle de acesso, ou iniciando o daemon sem restrições de chroot) • Bibliotecas compartilhadas: quais são, em que diretório estavam quando foram ligadas com o programa, e o dump binário delas (para ser comparado com as bibliotecas legítimas); • File descriptors: determina-se quais arquivos, pipes, sockets, ou quaisquer outros streams que o executável esteja usando. Verifica-se qual é o caminho deles no sistema de arquivos, se houver. 20

• Mapa de memória: determina-se, do ponto de vista do processo, onde estão os endereços das bibliotecas compartilhadas, do seu próprio código executável, da pilha, e da área de dados. • Dump de memória – além do mapa de memória, pode-se fazer um dump de toda a memória que pode ser “vista” pelo processo; • Raiz: em sistemas Unix, é importante saber qual é a raiz do processo, para verificar se ele está restrito a uma região específica do sistema de arquivos ou não (pode-se assim saber se um daemon quebrou as restrições impostas pela chamada de sistema). Por exemplo, a ferramenta Cryogenic [Bre00] realiza parte das tarefas descritas acima. Podemos usá-la para entender o que um processo (neste caso o gkrellm) está fazendo: \$ ps aux|grep gkrellm user 7711 1.3 0.7 23972

8268 ?

S

Jun17

87:15 gkrellm

Agora usamos o cryogenic: \$ cryogenic -p 7711 output \$ cd output/7711 \$ ls -l total 812K -rw----- 1 user user 8 -rw----- 1 user user 464 -rw----- 1 user user 431 -rw----- 1 user user 782K -rw----- 1 user user 9.3K -rw----- 1 user user 597

2005-06-21 2005-06-21 2005-06-21 2005-06-21 2005-06-21 2005-06-21

13:29 13:29 13:29 13:29 13:29 13:29

cmdline dirs\_and\_descriptors environ exe maps status

A única coisa que não foi produzida é o dump de memória do processo. O arquivo exe é o executável, reconstruído a partir do código em memória. O arquivo dirs\_and\_descriptors mostra os diretórios e descritores de arquivos: \$ cat dirs\_and\_descriptors /proc/7711/cwd -> /home/user /proc/7711/root -> / /proc/7711/fd/0 -> /dev/null /proc/7711/fd/1 -> pipe:[11700] /proc/7711/fd/2 -> pipe:[11700] /proc/7711/fd/3 -> socket:[11751] /proc/7711/fd/4 -> pipe:[11753] /proc/7711/fd/5 -> pipe:[11753] /proc/7711/fd/6 -> /proc/diskstats /proc/7711/fd/7 -> /proc/net/dev /proc/7711/fd/8 -> /proc/vmstat /proc/7711/fd/9 -> /home/user/.gkrellm2/lock\_:0 /proc/7711/fd/11 -> /proc/stat /proc/7711/fd/12 -> /proc/net/route

O arquivo maps mostra o mapa de memória do processo (neste caso grande demais para ser listado por inteiro neste trabalho): \$ cat maps  
08048000-080fa000 080fa000-0810c000 0810c000-08280000 b6a9d000-b6a9e000 b6a9e000-b6a9f000 b6a9f000-b729e000 b729e000-b72a1000  
b72a1000-b72b2000 b72b2000-b72b4000 (...)

r-xp rw-p rw-p rw-p ---p rwxp rw-p r--p r-xp

00000000 000b2000 0810c000 b6a9d000 b6a9e000 b6a9f000 b729e000 00000000 00000000

03:07 03:07 00:00 00:00 00:00 00:00 00:00 03:07 03:07

197248 197248 0 0 0 0 479325 81781

/usr/bin/gkrellm /usr/bin/gkrellm

/usr/share/fonts/truetype/ttf-bitstream-vera/Vera.ttf /usr/lib/pango/1.4.0/modules/pango-basic-fc.so

O arquivo status mostra o estado do processo, a quantidade de memória usada por ele, e outras informações: \$ cat status Name: gkrellm State: S (sleeping) SleepAVG: 98% Tgid: 7711 Pid: 7711 PPid: 7660 TracerPid: 0 Uid: 1001 1001 1001 1001 Gid: 1001 1001 1001 1001 FDSize: 256 Groups: 4 5 6 7 8 20 21 24 25 29 30 44 46 60 100 105 112 116 1001 1002 1012 VmSize: 23972 kB VmLck: 0 kB VmRSS: 8268 kB VmData: 9788 kB VmStk: 84 kB VmExe: 712 kB VmLib: 10036 kB VmPTE: 32 kB Threads: 1 SigPnd: 0000000000000000 ShdPnd: 0000000000000000 SigBlk: 0000000000000000 Sigln: 0000000020001000 SigCgt: 00000000800044a6 CapInh: 0000000000000000 CapPrm: 0000000000000000 CapEff: 0000000000000000

Dumps de memória Para capturar a memória de um sistema que ainda esteja rodando sem que os processos hosts na máquina tomem conhecimento de que foram descobertos, pode ser necessário o uso de hardware específico que faça um dump de toda a memória [CG04].

22

Análise de binários De posse de um binário (encontrado no sistema de arquivos ou extraído diretamente de um processo que estava executando), pode-se usar engenharia reversa para tentar determinar o que o executável em questão faz. Live CDs Tem se tornado comum a construção de live CDs para diversos objetivos específicos, e o mesmo acontece com forense computacional. Uma série de CDs com ferramentas para análise forense existe hoje. Juntando tudo De posse de todas estas ferramentas, um investigador deve tentar reconstruir o incidente. Isso implica em cuidadosa análise de logs, verificação de grande quantidade de arquivos, e muita atenção à relação causal e ordenação de fatos. Vejamos agora, de forma resumida, alguns exemplos (hipotéticos) de procedimento de análise forense: • Uma máquina é e conhecidamente comprometida. O perito vai até o local e acessa o sistema usando a conta de um usuário que sempre costuma usar aquela máquina. Tendo feito isso, ganha acesso de superusuário, faz um dump dos processos diretamente pela rede usando netcat, para uma outra máquina. Depois disso, desliga a máquina sem realizar o procedimento de shutdown (para não perder os dados na área de swap e não disparar armadilhas postas nos scripts de shutdown e inicialização); captura uma imagem do sistema de arquivos e o leva para a mesma máquina onde fará a análise. Por último, copia os logs da máquina onde eles são feitos (que é diferente da primeira). A análise do sistema de arquivos mostrou que diversos programas utilitários do sistema foram substituídos. O perito então verifica que outros arquivos foram modificados perto deste intervalo de tempo, e nota que há um arquivo suspeito, deletado pouco antes dos utilitários serem substituídos. Notando que o nome deste arquivo é o mesmo de um dos processos de que fez dump, ele volta sua atenção ao dito processo. Toma o executável (que foi conseguido diretamente da memória), e usa o utilitário biew para verificar o código assembly, confirmando que é um programa hostil. Uma análise mais detalhada do executável, seus descritores de arquivo e do resto do sistema de arquivos pode ajudar a determinar os efeitos dele no sistema, e a análise de logs pode ajudar a identificar 23

algum endereço IP com o qual este programa tenha se comunicado (de onde o atacante provavelmente coleta informações ou envia comandos ao seu programa). • Algum tipo de mídia (CDs, por exemplo) é obtido em uma batida policial. O sistema de arquivos é verificado: primeiro checando o diretório de arquivos, e depois através de uma varredura (por exemplo, com o foremost). Ao descobrir alguns arquivos escondidos, o perito os examina, e descobre que um deles é uma imagem que traz conteúdo escondido por esteganografia. O fato é comunicado a outros investigadores, que mostram ao perito outros objetos encontrados no mesmo local. Entre eles, há algumas folhas de papel com algumas frases aparentemente inocentes. O perito usa estas frases no que agora já é um processo de criptanálise, e descobre que a senha para obter o conteúdo dentro da imagem é composta pelas letras ímpares da primeira frase. Encontrando um rootkit Como um breve estudo de caso, apresentaremos o desafio número 15 do honeynet.org. De posse da partição raiz de um sistema Red Hat Linux 6.2 Server que foi comprometido, deve-se determinar como o atacante conseguiu invadir o sistema, e coletar o máximo de informações relacionadas ao ataque. De posse da imagem da partição raiz do sistema, usamos uma ferramenta como o utilitário dls do SleuthKit. Logo de início, notamos que um arquivo chamado lk.tgz foi deletado da partição. Podemos recuperá-lo facilmente com o autopsy (na verdade, o browser do autopsy mostra o arquivo com uma flad de removido, e só precisamos pedir para "baixar" o arquivo. O conteúdo do arquivo encontrado é o seguinte: drwxr-xr-x -rwxr-xr-x -rw-r--r--rwx-----rwx-----rwxr-xr-x -rw-r--r--rwxr-xr-x -rw-r--r--rwxr-xr-x -rw-r--r--rw-----rw-r--r--rw-----rw-r--r--rwx-----rwxr-xr-x -rwxr-xr-x -rwxr-xr-x -rw-r--r--rwxr-xr-x -rw-r--r--rwxr-xr-x -rw-r--r--rw-----rw-r--r--rw-----rw-r--r--rwx-----rwxr-xr-x -rwxr-xr-x -rwxr-xr-x

1031/users 1031/users 1031/users 1031/users 1031/users 1031/users 1031/users 1031/users 1031/users 1031/users 1031/users 1031/users  
 1031/users 1031/users 1031/users 1031/users 1031/users 1031/users 1031/users 1031/users

0 611931 1 3713 7165 1345 3278 79 11407 4060 880 540 344 512 688 8268 4620 33280 35300

2001-02-26 2002-02-08 2001-02-26 2001-03-02 2001-02-26 1999-09-09 2001-01-27 2001-02-26 2001-01-27 2001-02-26 2000-10-22 2000-10-22  
 2000-10-22 2000-10-22 2001-02-26 2001-02-26 2001-02-26 2001-02-26 2001-02-26

14:40:30 07:08:13 09:29:58 21:08:37 09:22:50 10:57:11 09:11:32 09:28:40 09:11:44 09:22:55 14:29:44 14:29:44 14:29:44 14:29:44 09:29:51 09:22:59  
 09:23:10 09:23:33 09:23:42

24

last/ last/ssh last/pidfile last/install last/linsniffer last/cleaner last/inetd.conf last/lsattr last/services last/sense last/ssh\_config last/ssh\_host\_key  
 last/ssh\_host\_key.pub last/ssh\_random\_seed last/sshd\_config last/sl2 last/last.cgi last/ps last/netstat

-rwxr-xr-x -rwxr-xr-x -rwx-----rw-r--r--rwxr-xr-x

1031/users 1031/users 1031/users root/root 1031/users

19840 53588 75 708 632066

2001-02-26 2001-02-26 2001-02-26 2001-03-02 2001-02-26

09:23:47 09:23:55 09:24:03 21:05:12 08:46:04

last/ifconfig last/top last/logclear last/s last/mkxfs

Uma análise detalhada dos arquivos permite inferir o comportamento do rootkit. Neste caso, um servidor ssh e um sniffer foram instalados na máquina, além de programas para enviar emails para o atacante e um script cgi que permitia ao atacante ter acesso a informações remotamente. O kit também substitui alguns binários que poderiam detectá-lo, como ps, netcat. Nem sempre a análise é tão simples. Pode ser que um arquivo com o rootkit completo não tenha sido transferido para o servidor, ou pode ser que o rootkit seja uma pista falsa. Dificultando a análise forense As ações tomadas pelos invasores para dificultar a análise forense variam em complexidade e eficácia; listamos aqui algumas delas: • Inclusão de pistas falsas no sistema: evidentemente, a inclusão de arquivos suspeitos, módulos de kernel, e outras pistas falsas podem desviar a atenção do analista. O atacante poderia mesmo realizar um ataque real, comprometendo uma máquina, apenas para desviar a atenção de algo que esteja realizando; • Cifragem de dados e logs: os malfatores que pretendam ocultar suas atividades podem usar de criptografia: sistemas de arquivos inteiros podem ser facilmente cifrados, e simplesmente desligar o computador é o suficiente para dificultar enormemente a análise forense (que ainda pode fazer uso de informações sobre tráfego de rede e outros fatores externos ao sistema de arquivos). Vale notar que algoritmos “caseiros” tem pouca chance de resistir aos esforços de um criptanalista experiente; • Cifragem de executáveis: tanto no caso de sistemas pertencentes a malfatores como no de sistemas alvo de executáveis hostis, é possível usar de técnicas de cifragem ou obscurecimento de código executável. No entanto, é necessário que o código seja decifrado antes de ser executado, e isso é o suficiente para que um analista determinado decifre o executável. Esta técnica, no entanto, pode atrasar consideravelmente a análise; • Disk wiping utilities: é possível recuperar os dados de um disco rígido, mesmo após a gravação de zeros em todos os seus setores [Gut96]. 25

Os utilitários de remoção segura de dados gravam dados aleatórios diversas vezes, tornando muito difícil (se não impossível) recuperar qualquer informação do disco rígido.

26

Referências Bibliográficas [Bre00] Dominique Brezinski. Cryogenic, 2000. <http://staff.washington.edu/dittrich/talks/blackhat/blackhat/>. [Car]

Brian Carrier. Mac-robber. <http://www.digital-evidence.org/>.

[CG04] Brian Carrier and Jow Grand. A hardware-based memory acquisition procedure for digital investigations. Digital Investigation Journal, 1(1):50–60, 2004. Ver também <http://www.grandideastudio.com/portfolio/index.php?id=1&prod=14>. [dSO02] Flávio de Souza Oliveira. Resposta a incidentes e análise forense para redes baseadas em windows 2000. Master's thesis, IC/Unicamp, Novembro 2002. [Gut96] Peter Gutmann. Secure deletion of data from magnetic and solid-state memory. In Sixth USENIX Security Symposium Proceedings, 1996. [http://www.cs.auckland.ac.nz/pgut001/pubs/secure del.html](http://www.cs.auckland.ac.nz/pgut001/pubs/secure%20del.html). [hun]

The hunter kernel module. [http://www.phrack.org/phrack/61/p610x03 Linenoise.txt](http://www.phrack.org/phrack/61/p610x03%20Linenoise.txt).

[sle]

The sleuth kit. <http://www.sleuthkit.org/>.

[stea]

Stegdetect. <http://www.outguess.org/detection.php>.

[steb]

Steghide. <http://steghide.sourceforge.net/>.

[WPA]

Sobre "archimedes". in <http://en.wikipedia.org/wiki/Archimedes>.

The

Wikipedia.

[WPC] Sobre "computer forensics". in The [http://en.wikipedia.org/wiki/Computer forensics](http://en.wikipedia.org/wiki/Computer_forensics).

Wikipedia.

[WPF]

Wikipedia.


Sobre "forensics". in <http://en.wikipedia.org/wiki/Forensics>. 27

The


## Related Documents

[Forense Computacional](#)


[Computacional Ucsm](#)

 October 2019 44 January 2020 82



### Hidraulica Computacional

 November 2019 46

### Complejidad Computacional

 December 2019 44


### **Our Company**

 2403 Burnside Court 1602-DE-7050 Dinamica De Fluidos Computacional [info@pdfcookie.com](mailto:info@pdfcookie.com)

### **Quick Links**

 December 2019 71

### Analisis Computacional Open Pits.pdf

 October 2019 44[About](#)[Contact](#)[Help / FAQ](#)[Account](#)**Legal**

[Terms of Service](#)

[Privacy Policy](#)

[Cookie Policy](#)

[Disclaimer](#)

## Follow Us



## Mobile Apps



---

Copyright © 2023 PDFCOOKIE.