Capítulo 1

Forense Computacional: fundamentos, tecnologias e desafios atuais

Evandro Pereira[1], Leonardo Lemes Fagundes[2], Paulo Neukamp[2], Glauco Ludwig[1], Marlom Konrath[2]

Universidade do Vale do Rio dos Sinos - Unisinos

[1]{evandrodvp, llemes, glaucol}@unisinos.br; [2]{pneukamp, marlomk}@gmail.com

Abstract

This chapter presents a very current theme that has received substantial attention from both academic and industrial communities: the computer forensics. The scientific and systematic inspection of computational environments, with the goal of figuring out and reconstruct events, still has lot of open research topics. On industry, the interest on this subject is supported by an increasing amount of criminal investigations having digital data as its main evidences. In this chapter, we present basic notions of computer forensics and tools that can be used to collect, keep and analyze evidences. The anatomy of some malicious codes and case studies are also used as a complement on the subject and as a way of increasing its comprehension.

Resumo

Este capítulo trata de um tema bastante atual e que tem recebido significativa atenção tanto da comunidade científica quanto da indústria: a forense computacional. A inspeção científica e sistemática em ambientes computacionais com o objetivo de tentar reconstituir eventos apresenta ainda vários tópicos de pesquisa em aberto. Na indústria, o interesse justifica-se pela grande quantidade de investigações criminais, cujas principais evidências estão armazenadas em formato digital. São apresentados neste capítulo noções de forense computacional e ferramentas que podem ser utilizadas para auxiliar na coleta, manutenção e análise de evidências. A anatomia de alguns códigos maliciosos e estudos de caso complementam o tema e auxiliam no entendimento do assunto.

1.1 Introdução

O *cyber crime* diferencia-se dos crimes tradicionais em função do seu modo de operação, pois envolve a utilização de dispositivos eletrônicos, de computadores e da Internet para a execução de ação ou omissão, típica, antijurídica e culpável [Chawki 2005]. Entretanto, tanto quanto os autores dos atos ilícitos convencionais – aqueles cometidos sem o uso de computadores – os responsáveis por crimes virtuais devem ser identificados, julgados e penalizados. Contudo, essa é uma tarefa complexa devido à possibilidade de anonimato dos contraventores e ao fato de que as evidências do crime podem estar distribuídas em diversos servidores espalhados pela Internet, possivelmente em computadores localizados em regiões distantes daquelas onde as vítimas se encontram.

Conforme estatísticas apresentadas no *Internet Crime Report*, um relatório anual elaborado conjuntamente pelo *National White Collar Crime Center* (NWCCC) e pelo *Federal Bureau of Investigation* (FBI), que reúne diversas informações sobre tendências e padrões observados nos crimes praticados via Internet, somente em 2006, aproximadamente U$ 200.000.000,00 (duzentos milhões de dólares) foram perdidos em conseqüência de diferentes tipos de fraudes [NWCCC and FBI 2006]. Entre os crimes apresentados pode-se citar o caso *Nigerian Letter Fraud*, em que uma vítima é induzida a auxiliar um agente de governo estrangeiro a movimentar grandes somas de dinheiro para fora do seu país em troca de uma generosa comissão. Após fornecer os dados da conta bancária, ao invés de ter a comissão depositada, as vítimas são roubadas. A pesquisa indica ainda que, assim como neste golpe, aproximadamente 76% dos casos reportados tiveram como meio de comunicação com a vítima o correio eletrônico.

Entre as ocorrências mais comuns estão a calúnia, difamação e injúria[1] via e-mail, o roubo de informações confidenciais e a remoção de arquivos. Essas ações são motivadas pelo interesse de causar constrangimento ou algum tipo de perda à vítima e, normalmente, são protagonizadas por colaboradores insatisfeitos ou por concorrentes de um determinado segmento de mercado. Além disso, crimes como pedofilia, fraudes e o tráfico de drogas via Internet também são atos ilícitos constantemente realizados com o apoio de computadores. Institutos de pesquisa como o WebSense indicam que, em 2007, o crime organizado se unirá à *crackers* para comprar, vender e negociar *commodities*, como *kits* de ferramentas prontas para ataques virtuais e golpes utilizando vulnerabilidades recém descobertas [Bessa 2006].

Com a finalidade de auxiliar na investigação de tais crimes, se faz necessário o uso da Forense Computacional. De acordo com [Palmer and Corporation 2001] a Forense Computacional pode ser definida como a inspeção científica e sistemática em ambientes computacionais, com a finalidade de angariar evidências derivadas de fontes digitais, tendo como objetivo, promover a reconstituição dos eventos encontrados (podendo assim, determinar se o ambiente em análise foi utilizado na realização de atividades ilegais ou não autorizadas).

---

[1] **Calúnia:** consiste em atribuir, falsamente, a alguém a responsabilidade pela prática de um fato determinado definido como crime. **Difamação:** consiste em atribuir a alguém fato determinado ofensivo à sua reputação. **Injúria:** consiste em atribuir a alguém qualidade negativa, que ofenda sua dignidade ou decoro.

Este capítulo trata de um tema bastante atual e que tem recebido significativa atenção tanto da comunidade científica quanto da indústria: a forense computacional. No âmbito acadêmico, o interesse deve-se à quantidade de questões de pesquisa em aberto, fruto do constante surgimento de novas tecnologias, de inúmeras vulnerabilidades e de ameaças cada vez mais sofisticadas [Richard and Roussev 2006]. Já no contexto da indústria, tal interesse justifica-se pela grande quantidade de investigações criminais, cujas principais evidências estão armazenadas em formato digital, e pela carência de profissionais especializados para realizar o processo de investigação de maneira precisa a fim de que o resultado obtido possa ser aceito por juízes nos tribunais, tanto como peça de acusação quanto de defesa [Viotto 2007].

O presente capítulo está organizado da seguinte forma. A Seção 1.2 apresenta análises referentes à anatomia de diversos tipos de códigos maliciosos (*malwares*) e um cenário de ataque, cujas evidências deixadas por *worms* e *rootkits* serão detalhadamente descritas. Na Seção 1.3 é realizada uma introdução à Forense Computacional. Já na Seção 1.4 são realizadas demonstrações sobre a utilização de um conjunto de ferramentas que oferecem suporte a cada uma das etapas do processo de investigação forense, bem como são mencionadas técnicas anti-forense normalmente empregadas por invasores no intuito de ocultar as evidências das ações realizadas. A Seção 1.5 apresenta quatro estudos de caso que consolidam os conceitos estudados nos capítulos anteriores e permitem ao leitor um primeiro contato com a prática forense. A Seção 1.6 demonstra os desafios atuais em forense digital e este trabalho se encerra apresentando as considerações finais na Seção 1.7.

1.2 Códigos Maliciosos

Segundo [Skoudis and Zeltser 2003], códigos maliciosos são conjuntos de instruções executadas em um computador e que fazem o sistema realizar algo que um atacante deseja. Esta definição é bastante genérica, buscando assim abordar todas as categorias de *software* que são comumente consideradas quando se fala de *malware*.

Nos Estados Unidos, a maioria das ações legais contra *malware* é tomada pela *Federal Trade Commission* (FTC), sendo que o seu foco principal é companhias que produzem ou distribuem *spyware* [Payton 2006]. Como na maioria dos casos estas companhias estão fora dos EUA, a FTC publicou em junho de 2005 uma recomendação para o congresso de mudança na lei para permitir cooperação com órgãos internacionais [FTC 2005]. No entanto, este tipo de iniciativa ainda é bastante incomum, o que facilita a ação de atacantes que se utilizam de servidores geograficamente dispersos.

Esta seção aborda as principais categorias de *malware* existentes. A classificação aqui apresentada é fortemente influenciada por [Skoudis and Zeltser 2003].

1.2.1 Vírus

Os primeiros códigos com capacidade de se auto-replicarem de que se têm notícia surgiram em 1962, nos laboratórios Bell, com o jogo Darwin, onde programas “lutavam” entre si para sobreviverem [Aleph-Null 1971]. No entanto, a utilização do termo vírus para referenciar-se a programas com capacidade de se auto-replicarem só surgiu em 1984, com o artigo “Computer Viruses – Theory and Experiments” de Fred

Cohen [Cohen 1987]. Nele, o autor demonstra ainda que não há algoritmo capaz de detectar todos os possíveis vírus de computador.

Embora recentemente novas definições estejam sendo propostas [Bonfante et al. 2007, Case and Moelius 2007], a definição de vírus aqui apresentada segue o modelo utilizado por [Adleman 1990] em que um vírus é considerado uma função computável que infecta qualquer programa. Um programa infectado pode realizar então três ações disparadas conforme as entradas: (a) executar no programa hospedeiro e propagar a infecção, (b) danificar o sistema ou (c) imitar o programa hospedeiro.

Uma das características de um vírus é a necessidade de anexar-se a um "programa hospedeiro" para funcionar [Skoudis and Zeltser 2003]. O alvo neste caso pode ser um arquivo executável, o setor de inicialização, um documento que suporte macros ou um arquivo de script. Os vírus propagam-se normalmente através de mídias removíveis, e-mails, downloads e diretórios compartilhados. Os tipos abaixo apresentados seguem a classificação apresentada em [Kaspersky 2007]:

Os chamados "vírus acompanhantes" (*companion viruses*) são aqueles nos quais o vírus não infecta um arquivo executável, mas utiliza o mesmo nome de um arquivo existente, com uma extensão diferente, que é preferida pelo Sistema Operacional [Sophos 2001]. Na Figura 1 um exemplo desta técnica é mostrada. Se o usuário clicar em Iniciar => Executar e digitar "notepad", sem aspas, o arquivo notepad.com é quem será executado.

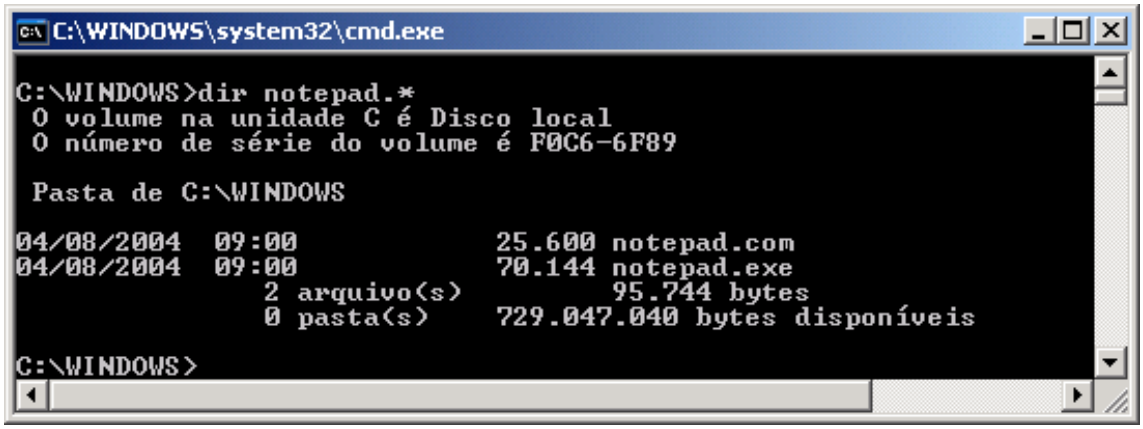

Figura 1. Exemplo de Vírus Acompanhante

No caso de vírus que infectam um arquivo executável existente, modificando o seu código, duas técnicas de infecção são possíveis: (a) infectar o início do arquivo e (b) infectar o final do arquivo. Na primeira técnica, o vírus irá adicionar seu código antes do início do código do programa hospedeiro. Um exemplo de aplicação de tal procedimento é o vírus Nimda. Já ao infectar o final do arquivo, como o próprio nome diz, o código do vírus é adicionado no final do arquivo. Para ser executado, parte do início do arquivo original é sobrescrita com instruções que realizam um salto de execução para o início do código do vírus. Ao final deste último, a parte sobrescrita do arquivo original é novamente adicionada e um segundo salto desviará o fluxo de execução para o código original. A maioria dos vírus utiliza esta técnica, pois a mesma

possui implementação mais fácil, embora vírus mais robustos geralmente utilizem mais de uma técnica de infecção.

Além de arquivos executáveis, alguns vírus podem instalar-se nos primeiros setores lidos durante a inicialização de um Sistema Operacional. Estes setores são tipicamente os primeiros setores de um disco rígido e é aonde a máquina irá inicialmente procurar pelo código a ser executado quando a mesma é ligada. O setor mestre de inicialização (também chamado Master Boot Record ou MBR) é o setor mais comumente utilizado, embora os primeiros setores de uma partição também possam ser utilizados. Um dos vírus mais conhecidos que infectava a MBR foi o Michelangelo [IBM 2007]. No entanto, infectar a MBR a partir do Windows NT 4 tornou-se inócuo, pois este SO e seus derivados acessam diretamente o hardware [Symantec 2007a].

Vírus de macro, por sua vez, são possíveis graças à capacidade de alguns *softwares* de interpretar código presente dentro de arquivos de dados. Os exemplos mais clássicos desta classe são os vírus de macro do Microsoft Word. Por possuir a capacidade de interpretar código em uma linguagem denominada Visual Basic for Applications, vários vírus desta categoria surgiram no Word no final da década passada. Um dos mais conhecidos foi o vírus Melissa [CERT 1999], que era capaz de enviar e-mails para a lista de contados do usuário, contendo como anexo um documento infectado, gerado pelo vírus ou do próprio usuário.

Por último, podem existir vírus com outros alvos em específico, como o vírus de scripts, que se espalham através de arquivos que permitem a execução de códigos em scripts [Kaspersky 2007]. O vírus PHP.Pirus, por exemplo, foi o primeiro vírus escrito na linguagem PHP [Symantec 2007b] e pode residir em servidores que possuem suporte a esta linguagem.

Em [Subramanya and Lakshminarasimhan 2001] os autores classificam os vírus em 5 gerações:

1. Vírus Simples: não faziam nada muito significativo além de replicar-se. Freqüentemente consumiam toda a memória por replicarem-se indiscriminadamente.
2. Vírus com auto-reconhecimento: detectam um sistema já infectado através de alguma assinatura, não gerando duplicidade de infecção.
3. Vírus invisíveis (*stealth*): interceptam chamadas do sistema para tentar esconder a sua presença.
4. Vírus com arsenal (*armored*): empregam técnicas para dificultar a análise de seu código e podem realizar ataques diretos a antivírus presentes no sistema.
5. Vírus polimórficos: também chamados de auto-mutantes, este tipo de vírus infecta novos alvos com versões modificadas ou criptografadas de si mesmo.

1.2.2 Backdoor

Segundo [Skoudis and Zeltser 2003], um *software* que permite uma “entrada pelos fundos” (*backdoor*) é um programa que habilita um atacante a furar os controles normais de segurança de um sistema, ganhando acesso ao mesmo através de um caminho alternativo. Segundo [Zhang and Paxson 2000] normalmente um *backdoor*

opera sobre o protocolo Telnet, Rlogin ou SSH e tipicamente fornece uma das seguintes funcionalidades ao atacante:

1. Aumento dos Privilégios Locais (*Local Escalation of Privileges*): permite que um usuário normal execute programas com privilégios de superusuário.
2. Execução Remota de Comandos: permite que o atacante envie comandos para a máquina alvo e obtenha as respostas geradas pela execução dos mesmos.
3. Acesso Remoto à Linha de Comando: permite que o atacante utilize um *shell* remoto na máquina alvo, de onde poderá realizar qualquer operação como se estivesse utilizando o teclado em frente à máquina real.
4. Controle Remoto da Interface Gráfica: permite ao atacante observar e interferir na interface gráfica à qual o usuário local está conectado, fornecendo assim acesso pleno à máquina.

Um dos *backdoors* mais conhecidos é o netcat [Giacobbi 2007], também chamado de "canivete suíço". Este apelido se deve ao fato de o netcat poder executar praticamente qualquer comando em uma máquina e desviar a entrada e/ou saída padrão para uma conexão de rede. Assim, pode-se programar este *software* para ficar escutando uma porta TCP ou UDP e esperando por conexões em uma máquina alvo. Ao estabelecer uma conexão, o netcat executa um comando e passa a desviar toda a saída para a conexão estabelecida. Na outra ponta, um atacante executa o programa numa versão cliente e passa a desviar a entrada padrão para o programa executando na máquina alvo e obtendo os resultados gerados nesta.

A Figura 2 ilustra um exemplo de como o netcat pode ser utilizado para fornecer acesso remoto à linha de comando. Outros exemplos de programas que podem ser utilizados como *backdoors* são Virtual Network Computing (VNC) [Cambridge 2007] e Loki [Phrack 2007]. O primeiro permite que se capture e manipule a interface gráfica do usuário e o segundo é um *backdoor* que utiliza o protocolo ICMP, não necessitando portanto, abrir portas TCP ou UDP.

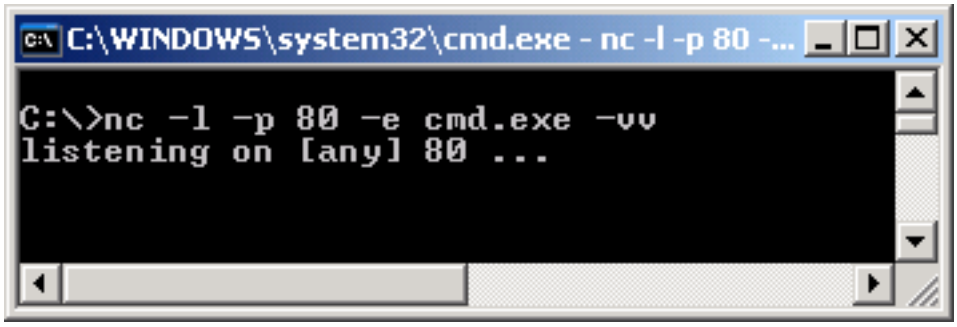

Figura 2. Exemplo de utilização do Netcat

1.2.3 Cavalos de Tróia

Segundo [Kolter and Maloof 2006], um cavalo de tróia (*Trojan Horse*) é um programa que se mascara ou aparenta possuir um propósito benigno, mas que arbitrariamente realiza funções maliciosas. Normalmente, um cavalo de tróia não é capaz de replicar-se

automaticamente [Haagman and Ghavalas 2005]. Segundo [Newman 2006], quando um cavalo de tróia é ativado os resultados podem variar, mas freqüentemente estes programas criam *backdoors* (veja Seção 1.2.2) permitindo acesso remoto ou vazamento de informações.

Para não ser descoberto, é comum que cavalos de tróia tentem esconder seus rastros, bem como disfarçar-se de programas legítimos. Para disfarçar-se de programa legítimo um cavalo de tróia muda seu nome para nomes comuns de serem encontrados no SO hospedeiro, como explorer.exe, iexplore.exe, svchost.exe, csrss.exe*, services.exe*, smss.exe*, spoolsv.exe*, System*, System Idle Process* ou winlogon.exe* no Windows ou init, cron ou httpd em sistemas *nix.

A mostra um exemplo de como se pode renomear um programa para escondê-lo entre os processos normais do sistema. Nela, primeiramente o nome do arquivo executável netcat.exe é trocado para explorer.exe. Em seguida o programa é executado e pode-se notar que seu nome aparece na lista de tarefas do Windows e que este programa passa a escutar a porta 80.

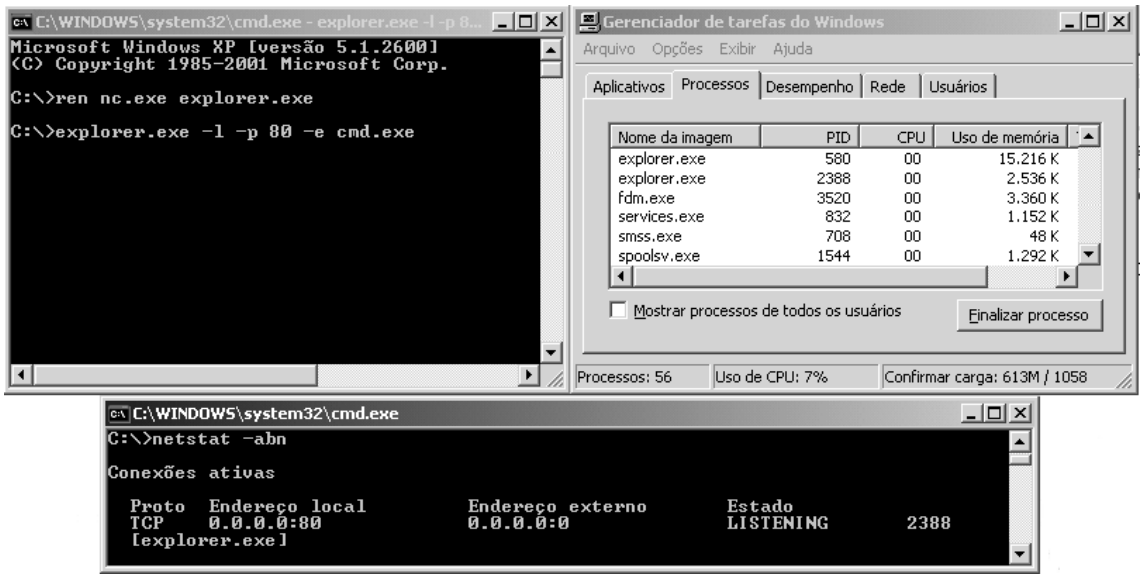

Figura 3. Exemplo de como disfarçar um Cavalo de Tróia

Segundo [Skoudis and Zeltser 2003] outra prática comum é um programa cavalo de tróia ser combinado com um programa executável normal em um único executável. Programas que juntam dois ou mais executáveis são chamados de *wrappers* e o uso destes *softwares* habilita um atacante a mesclar praticamente qualquer cavalo de tróia com qualquer programa legítimo. Como a maioria dos cavalos de tróia não irá gerar saídas para a tela, o programa combinado aparenta ser apenas o outro, escondendo sua atividade do usuário.

Um caso famoso de um cavalo de tróia foi reportado em 2002, quando um atacante invadiu o site oficial do *software tcpdump* e adicionou códigos nos scripts de instalação que permitiriam um atacante executar remotamente comandos em uma

máquina Linux [CERT 2002b]. Em ambiente Windows, programas comumente utilizados como cavalos de tróia são sub7 [Crapanzano 2003] e BO2K [Dildog 2007].

1.2.4 Spyware

De acordo com [Payton 2006], um *spyware* é um *software* que auxilia a coleta de informações sobre uma pessoa ou organização sem o seu conhecimento, que pode enviar tais dados para outra identidade sem o seu consentimento ou que toma controle do computador sem que o usuário saiba disso. De todos os tipos de *malware* existentes *spyware* é o mais comumente encontrado. Segundo [Weiss 2005], uma pesquisa conduzida pela Dell em setembro de 2004 estimou que aproximadamente 90% dos PCs com Windows possuíam no mínimo um *spyware*. Este tipo de *software* foi responsável por metade das falhas em ambientes Windows reportados por usuários da Microsoft [Shukla and Nah 2005]. Outro estudo apontou uma média de 25 *spywares* por PC [Sipior et al. 2005].

Segundo [Solove and Rotenberg 2003], Xupiter e Gator eram as empresas líder na disseminação de programas tidos como *spywares* em 2003. Estimava-se que neste período aproximadamente 35 milhões de PCs nos EUA possuíam algum *software* da Gator Companion. Atualmente não há um consenso quanto às classificações e tipos de *spyware* existentes. Algumas proposições englobam inclusive categorias de *software* que são nesta seção apresentados em separado, como *keyloggers* por exemplo. Em [Payton 2006], *spywares* são classificados quanto ao seu tipo em:

- *Advertising displays*: códigos que mostram anúncios de vários tipos no PC do usuário.
- *Automatic download software*: instalam outros *softwares* sem o conhecimento e consentimento do usuário.
- *Autonomous spyware*: programas que são executados fora de um navegador web, normalmente sendo executados na inicialização e permanecendo indetectáveis pelo usuário.
- *Tracking software*: programas que monitoram os hábitos de um usuário ou os dados fornecidos por ele em páginas web e enviam estas informações pela Internet para algum servidor remoto.

Para reduzir a presença deste tipo de código malicioso, diversas companhias desenvolveram produtos *anti-spyware*. Entre as soluções mais famosas estão: *Spybot Search and Destroy* [Kolla 2007], *Ad-Aware* [Lavasoft 2007], *Pest Patrol* [Etrust 2007] e *Microsoft Windows Defender* [Microsoft 2007a]. Estas ferramentas geralmente já possuem ferramentas para automaticamente eliminar os *spywares* encontrados. Recentemente os *softwares* de antivírus começaram também a detectar e remover este tipo de código malicioso.

1.2.5 Worms

Os *worms* são caracterizados como programas que se auto-propagam por meio de uma rede de computadores explorando vulnerabilidades em serviços usados em larga escala (como programas de e-mail, programas de mensagens instantâneas e compartilhamentos

de rede) [Zou et al. 2005]. Um *worm* diferencia-se de um vírus pela sua característica de auto-replicação, ou seja, não necessita de intervenção humana para se disseminar.

A primeira implementação de um *worm* foi realizada em 1978 por John Shock e Jon Hupp, pesquisadores da Xerox. Na época, a intenção dos autores foi desenvolver um *software* capaz de encontrar processadores ociosos disponíveis na rede da Xerox e, assim, designar tarefas para esses processadores computarem. Contudo, a partir de 1980, o conceito de *worms* passou a ser utilizado por atacantes para espalhar *malwares* de forma rápida e abrangente. Atualmente, os *worms* são classificados como uma das maiores, ameaças virtuais, chegando a atingir 65% dos incidentes de segurança reportados ao Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil (CERT.br) no período de Janeiro a Março de 2007 [CERT.br 2007].

Comumente, os *worms* executam uma varredura em busca de sistemas e serviços vulneráveis. Quando máquinas-alvo são identificadas, os *worms* exploram as vulnerabilidades encontradas e se propagam infectando essas máquinas. Após, a procura por novos alvos recomeça [Klaus and Nelson 2001]. Um *worm* pode ter as mais diversas finalidades, como simplesmente espalhar-se consumindo largura de banda, causar ataques de negação de serviço, apagar arquivos, enviar arquivos por e-mail e, principalmente, instalar outros *malwares* como *keyloggers*, *rootkits* e *backdoors*.

No decorrer dos últimos anos, vários *worms* tornaram-se conhecidos por suas habilidades de rápida disseminação e pelos danos que causaram. Como exemplo, pode-se citar o *Code Red Worm* [Tham 2001]. O *Code Red* foi detectado em 12 de julho de 2001 pela empresa *eEye Digital Security* [eEye 2007], explorando uma vulnerabilidade já conhecida [CERT 2002a] em servidores *Internet Information Service* (IIS) da *Microsoft*. A vulnerabilidade havia sido reportada um mês antes de o *Code Red* entrar em ação, indicando o despreparo de muitos administradores de redes. De acordo com o CERT, há uma estimativa de que mais de 250.000 servidores foram infectados em apenas 9 horas.

A vulnerabilidade explorada pelo *Code Red* está baseada no fato de que quando um servidor ISS é instalado, diversas extensões do ISAPI (*Internet Services Application Programming Interface*) são instaladas automaticamente. O ISAPI permite aos programadores estender as potencialidades de um servidor ISS utilizando bibliotecas DLLs. A biblioteca “idq.dll”, utilizada pelo serviço de indexação do ISS, continha erros de programação, pois não realizava a checagem de strings longas de entrada. Assim, permitindo que atacantes ocasionassem um buffer overflow ao enviar dados a essa DLL. Esta vulnerabilidade em específico dava acesso remoto de super-usuário ao sistema. Todas as versões do ISS que acompanhavam por padrão os sistemas Windows NT 4.0, Windows 2000 e Windows XP beta estavam vulneráveis [CERT 2002a].

As operações realizadas pelo Code Red podem ser classificadas em 6 passos [Tham 2001]:

1. O *worm* tenta conectar-se na porta TCP 80 de máquinas selecionados randomicamente, assumindo que um servidor web será encontrado. No caso de obter conexão, o atacante envia uma requisição HTTP GET para o alvo. A presença da seguinte string em um *log* de um servidor web pode indicar o comprometimento do servidor por parte do Code Red:

```
/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u9
090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u90
90%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a
```

2. Uma vez executado, o *worm* faz uma busca pelo arquivo c:\notworm. Caso esse arquivo seja encontrado, a execução do Code Red é cancelada. Caso contrário, o *worm* gera 100 threads;
3. Se a data do sistema invadido foi anterior ao dia 20 do mês, 99 threads são usadas para a tentativa de invadir mais sistemas;
4. A centésima thread é responsável por modificar a página principal do servidor web (caso a linguagem padrão do sistema o seja o inglês), a qual passará a exibir a mensagem: "*HELLO! Welcome to http://www.worm.com! Hacked by Chinese!"*. Essas alterações são realizadas sem modificar o arquivo da página no disco físico, mas sim, na memória;
5. Caso a data do sistema for entre os dias 20 e 28, o *worm* tenta então disparar um ataque de negação de serviço ao domínio www.whitehouse.com;
6. Caso a data seja superior ao dia 28, a execução do *worm* é cancelada.

Ao longo dos tempos, algumas variantes do *Code Red* acabaram sendo desenvolvidas (ex: *Code Red II, Code Blue, Code Green*). Mais detalhes sobre as mesmas podem ser encontradas em [Tham 2001].

1.2.6 Key Loggers

Os *keyloggers* são definidos como um tipo de spyware cuja finalidade é capturar tudo o que for digitado em um determinado computador [CERT.br 2007]. As intenções no uso de um *keylogger* podem ser as mais diversas, como, por exemplo, monitorar as atividades dos funcionários de uma determinada empresa, capturar conversas em programas de mensagens instantâneas e, inclusive, capturar informações e senhas bancárias. As informações obtidas podem então ser enviadas pela Internet para o gerente da empresa ou para um atacante, tudo sem o consentimento da vítima que faz uso do computador infectado.

De acordo com [SecurityFocus 2007], os *keyloggers* podem ser classificados em três diferentes tipos:

Hardware keylogger: trata-se de um dispositivo físico posicionado entre o teclado e o computador da vítima. Apesar de serem rápidos (por possuírem um hardware dedicado para executar sua função) e por não serem detectados por mecanismos como anti-vírus e anti-spyware, os mesmos podem ser detectados

visualmente por uma pessoa. Além disso, possuem espaço de armazenamento limitado e necessitam de acesso físico a máquina vítima para serem instalados;

Sofware keylogger usando um mecanismo de *hooking*: um *hook* trata-se de uma rotina que tem como objetivo "ficar no meio do caminho" do tratamento normal da execução de informações do Sistema Operacional (SO). Para isso, os programadores utilizam funções disponibilizadas pela API (*Aplication Program Interface*) do SO. Essas funções são responsáveis por capturar as mensagens do sistema (assim como as teclas que são pressionadas) antes que as mesmas sejam tratadas pelas devidas rotinas de tratamento. *Keyloggers* desse tipo normalmente possuem um módulo executável, que dispara a execução do aplicativo, e uma biblioteca que contém as rotinas para a captura das informações desejadas. Esses *keyloggers* podem ser instalados remotamente, no entanto, são os mais lentos e facilmente detectáveis por programas como anti-vírus e anti-spywares;

Kernel keylogger: este tipo de *keylogger* trabalha no nível do kernel e usa suas próprias rotinas para receber os dados diretamente dos dispositivos de entrada (no caso, o teclado). É o método mais difícil de ser desenvolvido (por exigir um elevado conhecimento de programação) e também de ser detectado (por substituir as rotinas padrão do SO e serem inicializados como parte do próprio sistema). Pelo fato de trabalharem no núcleo do sistema, não são capazes de capturar informações que são trocadas diretamente no nível de aplicações (ex: operações de copiar e colar e operações de autocompletar).

Atualmente, um dos *keyloggers* que mais se destaca é o *Perfect Keylogger* (BPK), da *Blazing Tools Software* [BlazingTools 2007]. O BPK trata-se de um *keylogger* baseado em *hooking* que pode ser facilmente instalado em sistemas operacionais Windows. Por possuir um processo de instalação bastante simplificado e uma interface gráfica amigável, qualquer usuário, mesmo sem experiência em segurança de computadores, é capaz de utilizar o programa e monitorar atividades alheias.

A versão completa do BPK pode ser obtida por um preço acessível (U$ 34,95) via o site do seu fabricante. Uma versão de avaliação do produto também é disponibilizada, o que facilita ainda mais a disseminação do seu uso. A Figura 4 mostra a interface de configurações gerais do BPK. Entre suas principais funcionalidades pode-se citar:

- capturar tudo o que for digitado no computador;
- trabalhar em modo invisível (ocultando o processo em execução do gerenciador de tarefas, tornando o programa invisível à lista de startup e removendo o programa do menu de inicialização e da lista de desinstalação de programas do Windows);
- renomear os arquivos que compõem o aplicativo para que tenham um nome qualquer (dificultando a busca no sistema de arquivos pelo nome original do *keylogger*);
- possibilitar que os processos de instalação, atualização e desinstalação possam ser realizados remotamente;
- registrar em arquivos de log os sites que foram visitados;

- enviar as informações capturadas para um determinado e-mail.

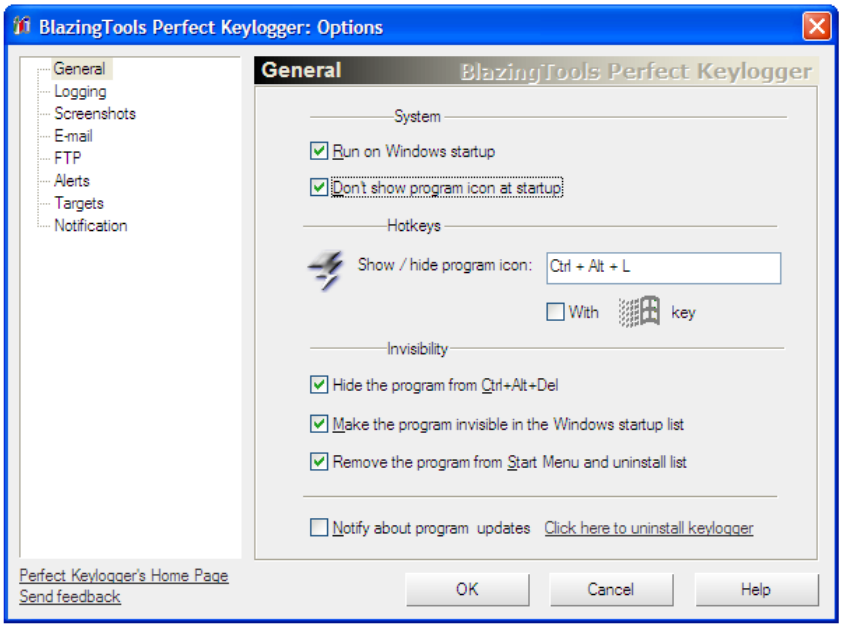

Figura 4. Tela de configurações gerais do *Perfect Keylogger*

Pelo fato do BPK ter a possibilidade de rodar em modo background e ficar invisível ao gerenciador de tarefas do Windows, é preciso do auxílio de uma ferramenta como a *SysInternals' Process Explorer* [Microsoft 2007b] para poder visualizar seu processo em execução. A Figura 5 apresenta o processo spyware executando em *background* (representado por bkp.exe).

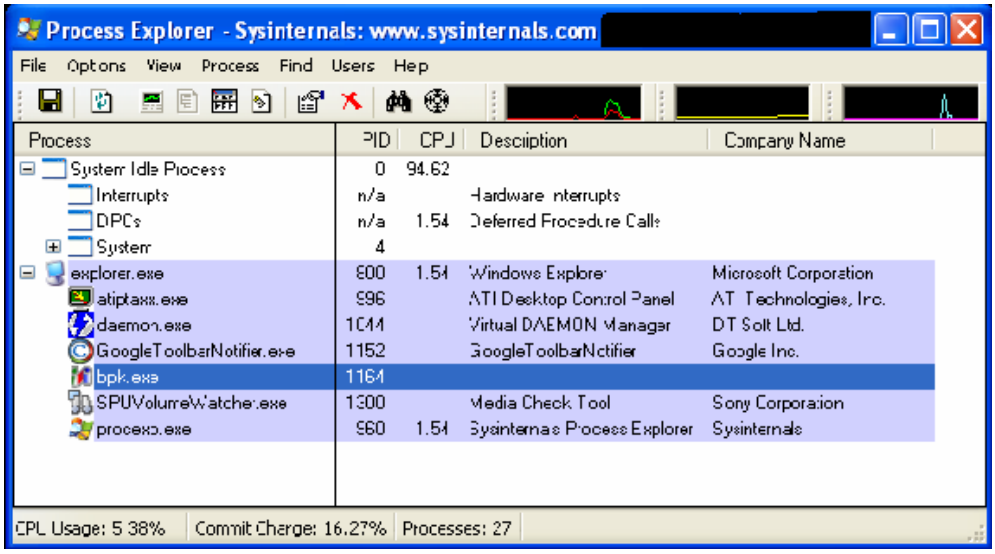

Figura 5. A ferramenta *SysInternals' Process Explorer* exibindo a execução do BPK em *modo background*

Nos últimos anos, foi desenvolvido também o conceito de *screenlogger* [CERT.br 2007]. Um *screenlogger* tem a finalidade de obter uma cópia (*screenshot*) da tela do computador da vítima assim que um clicar do mouse for efetuado. Os *screenloggers* foram uma alternativa utilizada por atacantes para que os mesmos pudessem capturar senhas bancárias quando organizações passaram a utilizar o conceito de teclados virtuais (modelo de teclado que permite que as senhas sejam informadas utilizando um mouse). Essa funcionalidade também está incluída nas versões mais recentes do BPK.

1.2.7 Rootkits

Um *rootkit* pode ser definido como um programa - ou um conjunto de programas - usado por um atacante para que o mesmo consiga ocultar sua presença em um determinado sistema e, ainda, para permitir acesso futuro a esse sistema (ex: por meio da instalação de *backdoors*) [Klaus and Nelson 2001]. Percebe-se que o termo *rootkit* refere-se a um conjunto de ferramentas utilizadas pelo atacante não para obter privilégios de super-usuário (como no caso do uso de um *exploit* que dá acesso *root* a um sistema), mas sim, para manter esses privilégios ocultos em seus acessos futuros [Microsoft 2007b]. Os *rootkits* podem ser classificas em duas categorias [Klaus and Nelson 2001]: os *rootkits* tradicionais e os *rootkits* baseados em LKMs (*Loadable Kernel Modules*).

Os *rootkits* tradicionais começaram a ser desenvolvidos em meados de 1994 e são caracterizados por versões modificadas de comandos do sistema, como *ls* (usado para listar arquivos), *ps* (usado para listar processos), *ifconfig* (usado para configurar dispositivos de rede) e *netstat* (usado para exibir conexões de rede). Esses comandos passaram a ser programados para ocultarem do administrador do sistema os processos, os arquivos e as conexões utilizadas pelo atacante. No caso do *ifconfig*, por exemplo, o programa original é modificado e substituído por uma versão maliciosa que oculta o fato de uma determinada interface de rede estar sendo executada em modo promíscuo[2], dando a ilusão ao administrador da máquina de que tudo está ocorrendo normalmente em seu sistema.

Os *rootkits* dessa geração podem ser neutralizados facilmente. Para isso, é preciso fazer uso de programas específicos para monitorar o sistema original e armazenar as informações obtidas (como tamanho do arquivo e data de criação) em bases de dados. Após, caso algum arquivo seja alterado, o programa identifica essa alteração e aponta a possibilidade do sistema estar comprometido.

Os *rootkits* baseados em LKM, por sua vez, começaram a ser publicados a partir de 1997. Esses códigos maliciosos funcionam alterando as chamadas do sistema (*system calls*). Os módulos do kernel são componentes que podem ser carregados de forma dinâmica, modificando a funcionalidade de um sistema mesmo sem a necessidade de uma reinicialização. Esse tipo de *rootkit* normalmente altera também as chamadas do sistema que permitem listar os módulos de kernel instalados. O processo de detecção desses *malwares* é muito mais difícil comparado ao processo de detecção dos *rootkits*

[2] Uma interface de rede sendo executada em modo promíscuo passa a aceitar pacotes de maneira passiva, mesmo que esses pacotes não sejam endereçadas para essa interface.

tradicionais, pois os comandos do sistema continuam inalterados e o próprio kernel responderá às requisições.

1.2.8 Bots

Segundo [Holz 2005] há três atributos que caracterizam um *bot* (nome derivado de {Robot}): (a) a existência de um controle remoto, (b) a implementação de vários comando e (c) um mecanismo de espalhamento, que permite ao *bot* espalhar-se ainda mais. De acordo com [Ramachandran and Feamster 2006], acredita-se que a maior parte dos *spams* é enviado por *botnes*, ora partindo diretamente destes, ora os mesmos sendo utilizados como *relay*.

A família de *bots* mais conhecida é provavelmente a família *Agobot* (também conhecida como Gaobot). O código foi escrito em *C++* com suporte multi-plataforma. Segundo a Sophos [Sophos 2007] há mais de mil variantes do Agobot conhecidas. Ao ser iniciado, o *bot* tenta conectar-se com alguns endereços previamente conhecidos e realizar um teste de velocidade, o que torna fácil a contabilização do número de infecções [Holz 2005].

Tipicamente um *bot* conecta-se a uma rede IRC (*Internet Relay Chat*) e fica esperando por comandos em um canal específico. Ao identificar seu mestre, o *bot* irá realizar o que lhe for ordenado através de comandos. Um conjunto de *bots* é chamado de uma *botnet*, *bot-network* ou mesmo *zombie drones*. Em [Mclaughlin 2004], os autores apontam um estudo que estimava que, em 2004, o Phatbot possuía uma rede aproximadamente 400.000 *bots*.

1.3 Forense Computacional

Nessa seção será apresentado uma breve resumo sobre a história da ciência forense, em seguida serão descritas as etapas do processo de investigação e os desafios inerentes a realização das técnicas da Forense Computacional em ambientes de produção ou que não podem ser desconectados (*live systems*).

1.3.1 Uma Breve Incursão pela História da Ciência Forense

A Forense Computacional tem como objetivo, a partir de métodos científicos e sistemáticos, reconstruir as ações executadas nos diversos ativos de tecnologia utilizados em cyber crimes. Embora as aplicações de tais métodos no contexto que envolve a tecnologia seja algo recente, o mesmo não se pode afirmar da ciência forense como um todo, pois ao longo da história podem ser observados muitos casos de aplicação de métodos científicos para fins de comprovação de fraudes e reconstrução de eventos.

Um dos primeiros casos de descoberta de fraudes a partir de experimentos científicos é relatado pelo historiador romano Virtrúvio, segundo o qual Arquimedes foi chamado pelo rei Hieron para atestar que a coroa encomendada junto a um artesão local não era composta pela quantidade de ouro combinada previamente entre as partes. Embora existisse essa suspeita, o rei Hieron não tinha evidências que lhe permitissem acusar o fraudador e, portanto, atribuiu a tarefa de investigação sobre o caso a Arquimedes que, depois de algum tempo e quase que por acaso, formulou a teoria do

peso específico dos corpos [Inman and Rudin 2000]. A partir dessa nova descoberta Arquimedes comprovou que parte da estrutura da coroa havia sido composta de prata e, portanto, não se tratava de uma peça totalmente de ouro.

Outro fato que demonstra que há muito tempo a sociedade faz uso da forense para fins da lei e para atribuir responsabilidades a determinados indivíduos, data do século VII. Nessa época, já eram utilizadas impressões digitais para determinar as identidades dos devedores. As impressões digitais dos cidadãos eram anexadas às contas que ficavam em poder dos credores. Essas contas eram legalmente reconhecidas como prova válida do débito. Essa mesma técnica também era empregada pelos chineses para identificar a autoria de documentos e de obras de arte. Em 1823, John Evangelist Purkinji, um professor de anatomia da Universidade de Breslau, Czecheslovakia, publicou o primeiro artigo sobre a impressão digital e sugeriu um sistema de classificação baseado em padrões [Inman and Rudin 2000].

Já no século XX a evolução da ciência forense pode ser observada a partir de pesquisas que conduziram, por exemplo, à identificação do tipo sanguíneo e a análise e interpretação do DNA. Durante este período foram publicados os principais estudos referentes a aplicação de métodos e técnicas utilizadas na investigação de crimes e, também, foi criado *The Federal Bureau of Investigation* (FBI) – uma referência no que tange a investigação de crimes e a utilização de técnicas forense em diversas áreas [Inman and Rudin 2000].

Atualmente, existem peritos especializados em diversas áreas ou disciplinas como por exemplo: análise de documentos, antropologia, balística, criminalística, genética, odontologia, patologia, psiquiatria, química e toxicologia. De maneira formal, afirma-se que a Forense Computacional é uma sub-área da Forense Digital voltada a análise de evidências em computadores isolados e também em computadores em rede, embora ambas possam ser definidas como a ciência que estuda a aquisição, a preservação, a recuperação e a análise de dados que estão em formato eletrônico, a Forense Digital possui um escopo mais abrangente pois engloba evidências armazenadas e processadas por qualquer tipo de dispositivo eletrônico, por exemplo celulares, máquinas fotográficas digitais e computadores [Kruse and Heiser 2001]. Na subseção seguinte serão apresentadas as etapas do processo de investigação.

1.3.2 O Processo de Investigação Forense

Conforme mencionado na subseção 1.1, a Forense computacional é empregada em diversos cenários tanto para fins legais (por exemplo: investigar casos de espionagem industrial) quanto para o exercício de ações disciplinares internas (por exemplo: uso indevido de recursos da instituição) - em ambos os casos o intuito é obter evidências relacionadas à realização desses eventos.

As evidências são peças utilizadas por advogados nos tribunais e cortes do mundo inteiro, mas para que sejam consideradas provas válidas é muito importante que o perito realize o processo de investigação de maneira cuidadosa e sistemática, para que entre outras coisas todas as evidências sejam preservadas e detalhadamente documentadas. De acordo com [Kent et al. 2006] e [Kruse and Heiser 2001] as fases de um processo de investigação são:

- Coleta dos dados: nessa fase os dados relacionados a um evento devem ser coletados e a integridade dos mesmos deve ser preservada, posteriormente, os equipamentos devem ser identificados, devidamente embalados, etiquetados e suas identificações registradas;
- Exame dos dados: nessa segunda fase são selecionadas e utilizadas ferramentas e técnicas apropriadas a cada tipo de dado coletado, a fim de identificar e extrair as informações relevantes ao caso que está sendo investigado, mas sempre com a preocupação de manter a integridade dos dados;
- Análise das informações: a terceira etapa refere-se à análise dos dados filtrados na etapa anterior, cujo objetivo é obter informações úteis e relevantes que possam responder às perguntas que deram origem à investigação;
- Interpretação dos resultados: na última fase do processo de investigação gera-se um relatório no qual deve estar descrito os procedimentos realizados e os resultados obtidos.

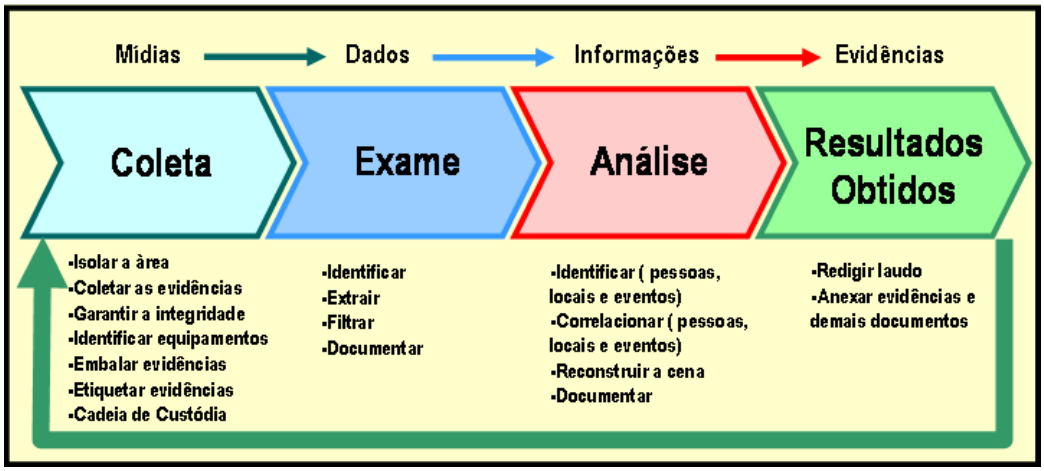

Figura 6. Fases do processo de investigação

A Figura 6 ilustra as quatro fases mencionadas acima, a seqüência em que devem ser realizadas e os procedimentos relacionados a cada uma das etapas. Além disso, é possível se observar as transformações ocorridas durante o processo forense. Por exemplo, a investigação começa a partir da apreensão dos dispositivos (computadores, meios de armazenamento e outras fontes de informações), em seguida os dados armazenados são coletados e passam pela fase de exame, essa é a primeira transformação pois a partir desse momento o perito utiliza ferramentas que lhe permitam separar apenas os dados relevantes ao caso investigado. As outras transformações que se observam estão relacionadas com as etapas de análise, da qual se obtêm a partir da correlação de eventos informações importantes, e de elaboração do relatório da investigação, que tem como resultado um laudo técnico no qual as evidências são claramente indicadas e descritas. A seguir as fases do processo forense serão descritas em detalhes.

Coleta dos Dados

Conforme menção anterior, a primeira etapa do processo forense é identificar possíveis fontes de dados. As fontes de dados mais comuns são computadores pessoais, laptops e dispositivos de armazenamento em rede. Esses sistemas normalmente possuem diversos tipos de conexões que possibilitam o acesso a outras fontes de informações, tais como: CD's e DVD's. Os equipamentos podem ainda possuir algumas portas de comunicação de vários tipos, como: *USB, Firewire, Flash card e PCMCIA* em que outras mídias e dispositivos externos de armazenamento de dados podem estar conectados [Kent et al. 2006].

Os dados também podem estar armazenados em locais fora de domínios físicos da cena investigada, como provedores de Internet, servidores FTP (*File Transfer Protocol*) e servidores coorporativos. Nesses casos, a coleta dos dados armazenados em um provedor de Internet, por exemplo, somente será possível mediante ordem judicial. Após a identificação das possíveis origens dos dados, o perito necessita adquiri-los. Para a aquisição dos dados, é utilizado um processo composto por três etapas:

1. o perito deve estabelecer a ordem (prioridade) na qual os dados devem ser coletados. Os fatores importantes na priorização dos dados são [Kent et al. 2006]:

   - **Volatilidade:** os dados voláteis representam informações que serão perdidas, caso o sistema seja desligado e, portanto, devem ser imediatamente coletados pelo perito. Por exemplo, o estado das conexões de rede e o conteúdo da memória.

   - **Esforço:** o esforço necessário para coletar dados de diferentes origens pode variar. O esforço envolve não somente o tempo gasto pelo perito, mas também o custo dos equipamentos e serviços de terceiros, caso sejam necessários. Por exemplo, coletar os dados de um roteador da rede local necessita menor esforço do que coletar os dados de um provedor de Internet.

   - **Valor estimado:** baseado na percepção do perito sobre a situação do ambiente e nas experiências anteriores semelhantes de investigação, ele deve estimar um valor relativo para cada provável fonte de dados, para, assim, poder definir a seqüência na qual as fontes de dados serão investigadas.

   Utilizando esses três fatores para cada provável fonte de dados, o perito poderá definir qual será a prioridade a ser adotada para a aquisição e quais dados serão coletados. Por exemplo, em uma investigação de invasão da rede, o perito deve preocupar-se, primeiramente, com os dados voláteis, como as conexões da rede, o estado das portas TCP e UDP e quais programas estão em execução. Em seguida, devem ser coletados os dados contidos na memória, as configurações da rede, informações sobre quais programas estão em execução para, então, iniciar a coleta dos dados não-voláteis.

2. **Copiar dados:** o processo de cópia dos dados envolve a utilização de ferramentas adequadas para a duplicação dos dados, por exemplo, o utilitário dd, encontrado na maioria das distribuições Linux, que pode ser usado para coletar os dados voláteis como os contidos na memória, e duplicação das fontes de dados não-voláteis, garantindo a integridade e segurança dos dados.
3. **Garantir e preservar a integridade dos dados:** após a coleta dos dados, o perito deve garantir e preservar a integridade dos mesmos, pois, caso isso não ocorra, eles poderão ser invalidados como provas perante a justiça. A garantia da integridade das evidências consiste na utilização de ferramentas que aplicam algum tipo de algoritmo *hash*. Esse procedimento deve ser executado nos dados originais e nas cópias, e as strings resultantes devem ser comparadas, para certificar-se de que são idênticas, garantindo, assim, a integridade dos dados.

A exemplo dos demais objetos apreendidos na cena do crime, os materiais de informática apreendidos deverão ter anotado em seu relatório de apreensão, conhecido como cadeia de custódia, o nome de todas as pessoas que estejam de posse dos mesmos e a situação envolvendo o referido material.

Exame dos dados

O exame dos dados tem a finalidade de avaliar e extrair somente as informações relevantes à investigação, o que representa uma tarefa muito trabalhosa visto a grande capacidade de armazenamento dos dispositivos atuais e a quantidade de diferentes formatos de arquivos existentes, entre eles: imagens, áudio, arquivos criptografados e compactados.

Em meio aos dados recuperados podem estar informações irrelevantes e que devem ser filtradas. Por exemplo, o arquivo de log do sistema de um servidor pode conter milhares de entradas, porém somente algumas delas podem interessar à investigação. Além disso, são muitos os formatos de arquivos que possibilitam o uso de esteganografia para ocultar dados, o que exige que o perito esteja atento e apto a identificar e recuperar esses dados.

A correta aplicação das diversas ferramentas e técnicas disponíveis, atualmente, pode reduzir muito a quantidade de dados que necessitam de um exame minucioso. A utilização de determinados filtros como palavras-chave ou tipos de arquivos nas pesquisas podem agilizar a localização das informações, tais como encontrar documentos que mencionem um determinado assunto, pessoa em particular ou ainda identificar entradas entre os registros de e-mail para um endereço específico.

Outra prática vantajosa é utilizar ferramentas e fontes de dados que possam determinar padrões para cada tipo de arquivo como texto, imagem, música, vídeos, entre outros. Por exemplo, o projeto denominado *National Software Reference Library* (NSRL), contêm uma coleção de assinaturas digitais referentes a milhares de arquivos o que pode ser usado para identificar e filtrar, por exemplo, arquivos que tenham sido manipulados por ferramentas de esteganografia [Kruse and Heiser 2001] e [Farmer and

Venema 2006]. As técnicas e ferramentas que podem ser utilizadas nesta fase da investigação serão descritas na próxima seção.

Análise das informações

Uma vez que as informações relevantes foram extraídas dos dados coletados, o perito deve concentrar suas habilidades e conhecimentos na etapa de análise e interpretação das informações. A etapa de análise tem a finalidade de identificar pessoas, locais e eventos, determinando como esses elementos estão inter-relacionados, pois, dessa maneira, será possível realizar uma descrição precisa e conclusiva da investigação [Kent et al. 2006] e [Farmer and Venema 2006].

Normalmente, nessa etapa, é necessário correlacionar informações de várias fontes de dados. Por exemplo, alguém tenta realizar um acesso não autorizado a um determinado servidor, através da análise dos eventos registrados nos arquivos de log do sistema, é possível identificar o endereço IP, utilizado pelo equipamento de onde a tentativa de acesso não autorizado. Além disso, os registros gerados pelos *firewalls*, sistemas de detecção de intrusão (tanto de rede quanto de host) e demais aplicações são extremamente importantes nesta etapa do processo.

Essa é uma fase que além de consumir muito tempo, esta muito suscetível a equívocos pois depende muito da experiência e do conhecimento dos peritos, já que são poucas as ferramentas que realizam esse tipo de análise com precisão [Casey 2006].

Interpretação dos Resultados

A interpretação dos resultados obtidos é a etapa conclusiva da investigação onde o perito constrói um laudo pericial que deve ser escrito de forma clara e concisa, elencando todas as evidências localizadas e analisadas, com base em todas as etapas anteriores da investigação.

O laudo pericial deve apresentar uma conclusão imparcial e final a respeito da investigação. Para que o laudo pericial se torne um documento de fácil interpretação por qualquer pessoa, seja ela do meio jurídico ou técnico, é indicado que o mesmo seja organizado em seções como: finalidade da investigação, autor do laudo, resumo do incidente, relação de evidências analisadas e seus detalhes, conclusão, anexos e glossário [Kent et al. 2006].

Nesse documento deve constar informações sobre a metodologia utilizada durante a realização do processo, as técnicas, os *softwares* e os equipamentos empregados, isso para que se necessário as fases da investigação possam ser reproduzidas. Na subseção 1.4.1 serão apresentadas mais informações sobre a elaboração do relatório final de uma investigação. A seção seguinte descreve as vantagens existentes e os cuidados que devem ser tomados durante a execução da investigação em dispositivos conectados a redes coorporativas e à Internet.

1.3.3 *Live Forensics*: Diagnóstico de sistemas *on-line*

Em muitos casos os profissionais de forense computacional estão mediante uma difícil tomada de decisão, desligar os equipamentos ou mantê-los operando a fim de executar

os procedimentos de uma investigação. Por exemplo, o sistema de detecção de intrusão gera alertas que indicam que o servidor web de uma organização esta sob um determinado ataque, o que pode ser um falso positivo, nesse momento a equipe de resposta a incidentes é acionada e tem que decidir entre a parada do servidor, o que pode representar a perda de dinheiro para a instituição, mas garante o tempo e as condições necessárias para que os peritos realizem as suas atividades, ou mantê-lo *on-line* o que permite ao investigador coletar dados voláteis - que são de grande importância para o entendimento e reconstrução dos eventos realizados - mas mediante qualquer descuido existe a possibilidade de haver a contaminação das evidências.

De acordo com [Carrier 2006] e [Adelstein 2006] o processo de investigação forense envolve basicamente dois tipos de técnicas: *post-mortem* e *live analysis*. A abordagem tradicional da Forense Computacional (post-mortem) tem como premissa a preservação de todas as evidências armazenadas nos discos rígidos e outras mídias, enquanto que a abordagem denominada de *live computer forensics* tem como objetivo obter o máximo de informações relacionadas ao contexto (por exemplo: estado das conexões, conteúdo da memória e dados referentes aos processos em execução), algo como uma fotografia da cena do crime. Essas técnicas quando realizadas de forma correta, claramente, se complementam e contribuem para que o resultado da investigação seja conclusivo e preciso.

Quando o processo forense é realizado nas mídias apreendidas (tais como discos rígidos, CDs e DVDs) o desafio, conforme mencionado anteriormente, é localizar entre um grande volume de dados, aqueles que são pertinentes ao caso investigado. Nesse tipo de cenário, a única e principal fonte de informação é o conteúdo gravado nos meios de armazenamento não voláteis. Esses dados podem ser obtidos a partir de ferramentas desenvolvidas ou instaladas e compiladas pela própria equipe de investigação e, portanto a priori confiáveis. Entretanto, quando se trata de cenários em que os dispositivos não foram desligados é importante que o perito certifique-se que as ferramentas utilizadas para executar os procedimentos de coleta, exame e análise dos dados não foram comprometidas a fim de gerar dados falsos ou omitir informações, como por exemplo ocultar processos em execução no sistema operacional ou não mostrar determinadas entradas do arquivo de log do servidor [Skoudis and Zeltser 2003].

Conforme [Skoudis and Zeltser 2003] e [Carrier 2006] os *rootkits* são, entre todos os códigos maliciosos, os principais causadores ou fontes de dados falsos, pois podem modificar as ferramentas (comandos) do sistema operacional e assim permanecerem com acesso ao host e não serem identificados. Os *rootkits* inserem filtros em determinadas partes do sistema operacional que impedem a visualização de algumas informações. Por exemplo, a Figura 2 mostra um filtro que impede que o arquivo passwds.txt, mesmo existindo no sistema de arquivos, seja exibido com a saída de um comando para listar o conteúdo de diretórios.

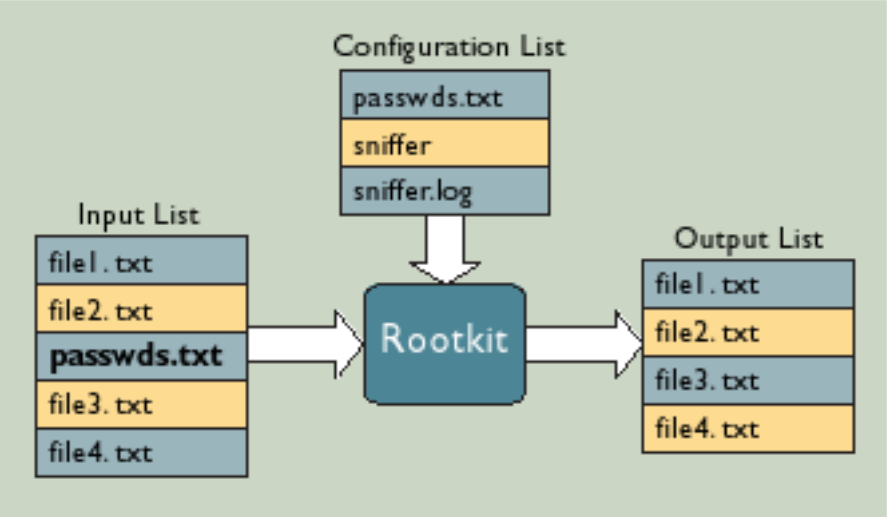

Figura 7. Exemplo de um filtro utilizado por *rootkits* [Carrier 2006]

Os *rootkits*, conforme visto na subseção 1.2.7, podem realizar modificações em diversas camadas do sistema operacional, ou seja, (a) podem atuar diretamente na aplicação, (b) redirecionar as chamadas de sistema, (c) substituir as bibliotecas compartilhadas e (d) subverter o kernel do sistema operacional [Hoglund and Butler 2005].

A fim de mitigar os riscos mencionados, sugere-se que o perito utilize um *live CD* com um conjunto de ferramentas apropriadas e que sejam confiáveis, pois isso servirá como contramedida para *rootkits* que atuam no nível da aplicação e de bibliotecas. Atualmente, existem algumas distribuições Linux voltadas a Forense Computacional que podem auxiliar no processo de investigação forense, entre elas sugere-se a utilização o projeto denominado FDTK [Neukamp 2007]. Contornar os problemas com *rootkits* que atuam no nível do kernel é um pouco mais complicado porque não há como acessar a memória ou o hardware sem passar pelo kernel. Neste caso a principal recomendação ainda é utilizar os detectores de *rootkits*, mas esse tipo de ferramenta pode interferir no sistema de alguma forma [Carrier 2006] e [Adelstein 2006].

Além disso, é importante relembrar que o perito deve (a) manter uma lista contendo a *hash* de todas as evidências coletadas, para que se necessário posteriormente possa demonstrar que nada foi alterado, e (b) ter em mente que alguns dados são mais efêmeros do que outros e, portanto, a ordem de volatilidade deve ser considerada no momento da coleta dos dados [Farmer and Venema 2006].

Em suma, além dos tipos de dados que podem ser coletados a diferença mais significativa entre *live* e *post-mortem analysis* é o grau de confiança existente nos resultados obtidos. Isso porque as ferramentas e comandos instalados no sistema investigado podem ter sido modificados para produzir dados falsos e também porque qualquer erro do perito pode contaminar ou alterar os dados existentes [Carrier 2006].

1.4 Técnicas e Ferramentas Forenses

Na seção anterior, foram apresentados alguns dos conceitos básicos sobre Forense Computacional e descritas as etapas em uma investigação forense. Essa seção retomará

este assunto, mas agora sob uma perspectiva mais técnica, com o intuito de aprofundar a compreensão do leitor sobre cada uma das etapas. Ao final, esta seção apresentará ainda uma introdução ao assunto das técnicas anti-forense.

1.4.1 Técnicas Forenses

Antes de iniciar a descrição sobre as técnicas e os procedimentos a serem adotados pelo perito em cada uma das fases do processo de investigação, é importante mencionar algumas das boas práticas que antecedem a coleta dos dados. Por exemplo [Farmer and Venema 2006]:

1. esterilizar todas as mídias que serão utilizadas ou usar mídias novas a cada investigação;
2. certificar-se de que todas as ferramentas (*softwares*) que serão utilizadas estão devidamente licenciadas e prontas para utilização;
3. verificar se todos os equipamentos e materiais necessários (por exemplo, a estação forense, as mídias para coleta dos dados, etc.) estão a disposição;
4. quando chegar ao local da investigação, o perito deve providenciar para que nada seja tocado sem seu consentimento, com o objetivo de proteger e coletar todos os tipos de evidências;
5. os investigadores devem filmar ou fotografar o ambiente e registrar detalhes sobre os equipamentos como: marca, modelo, números de série, componentes internos, periféricos, etc.
6. manter a cadeia de custódia.

Uma vez tomados esses cuidados, o perito poderá dar início a coleta de dados junto aos dispositivos eletrônicos apreendidos.

Coleta dos Dados

Uma vez que os equipamentos estejam protegidos e devidamente registrados, o perito poderá dar início à coleta dos dados. A primeira ação a ser tomada é manter o estado do equipamento, ou seja, se o equipamento estiver ligado, o mesmo não deve ser desligado e, se o equipamento estiver desligado, o mesmo não deve ser ligado, pois dessa forma não haverá modificações nas evidências.

Conforme mencionado na subseção 1.3.3, o estado no qual os equipamentos se encontram é muito importante, pois determinará a prioridade durante a coleta dos dados – que são classificados em voláteis e não-voláteis. A lista abaixo apresenta um conjunto de dados voláteis organizada pela ordem recomendada para coleta, segundo [Kent et al. 2006]:

- **Conexões de rede:** os sistemas operacionais oferecem recursos que permitem visualizar informações sobre as conexões de rede atuais. Por exemplo, os endereços IP de origem e destino, o estado das conexões e o programa associado a cada uma das portas. Além disso, a lista de sistemas de arquivos montados remotamente e o estado da interface de

rede também são dados relevantes e que podem auxiliar a análise dos dados;

- **Sessões de *Login*:** dados como a lista dos usuários atualmente conectados, o horário em que a conexão foi realizada e o endereço de rede de onde partiu essas conexões, quando correlacionados com outras informações como aquelas obtidas através das conexões de redes podem auxiliar, por exemplo, na identificação (a) dos usuários, (b) das ações realizadas e (c) do horário em que essas atividades foram executadas, o que permite a reconstrução dos fatos segundo a ordem cronológica dos eventos ocorridos;
- **Conteúdo da memória:** o espaço de troca e a memória principal, normalmente, contém os dados acessados recentemente tais como: senhas e os últimos comandos executados. Além disso, como em um sistema de arquivos, a memória pode conter resíduos de dados nos espaços livres ou que não estão em utilização, por exemplo: partes ou até mesmo arquivos inteiros que foram manipulados [Farmer and Venema 2006];
- **Processos em execução:** a lista e o estado de cada um dos processos do sistema são dados importantes, pois possibilitam identificar quais os programas que estão sendo executados;
- **Arquivos abertos:** comandos como o *lsof* presente em sistemas operacionais Linux geram uma lista contendo o nome de todos os arquivos que estão abertos no momento – essa informação pode ser um indicador para o perito do que deve ser coletado e, posteriormente analisado;
- **Configuração de rede:** as configurações da rede incluem informações como o nome da máquina, o endereço *IP* e o *MAC Address* (*Media Access Control*) de cada uma das interfaces de rede;
- **Data hora do sistema operacional:** a data e hora atual do sistema e as configurações de fuso horário – esses dados são importantes para reconstruir os eventos segundo a ordem cronológica de realização dos eventos.

Ao contrário dos dados voláteis, os dados não-voláteis são menos sensíveis à manipulação e podem ser coletados após o equipamento ser desligado, pois não sofrem alterações. Para realizar a cópia dos dados existem pelo menos dois métodos:

- **Cópia lógica (Backup):** as cópias lógicas gravam o conteúdo dos diretórios e os arquivos de um volume lógico. Não capturam outros dados que possam estar nas mídias, tais como os arquivos deletados ou fragmentos de dados armazenados nos espaços não utilizados, mas alocados por arquivos.
- **Imagem:** a imagem do disco, ou imagem *bit-a-bit* dos dados das mídias, inclui os espaços livres e os espaços não utilizados. As imagens bit-a-bit dos dados necessitam mais espaço de armazenamento e consomem

muito mais tempo para serem realizadas, porém permite ao investigador realizar as etapas de exame e análise com base em uma cenário mais próximo do real, pois possibilita por exemplo a recuperação de arquivos excluídos já que se trata de uma imagem da mídia apreendida.

A principal fonte de dados não-voláteis é o sistema de arquivos que armazena diversos tipos de dados, entre eles [Kent et al. 2006] e [Farmer and Venema 2006]:

- **Arquivos temporários:** durante a instalação e execução das aplicações são gerados arquivos temporários – que nem sempre são excluídos ao desligar os equipamentos. Esse tipo de arquivo pode conter dados relevantes como cópias de arquivos do sistema, dados sobre as aplicações e outras evidências;
- **Arquivos de Configuração:** esse tipo de arquivo fornece uma série de informações, como por exemplo: a lista dos serviços que devem ser ativados durante o processo de inicialização, a localização de arquivos de log, a relação de grupos e usuários do sistema e também os arquivos de senha e de agendamento de tarefas;
- **Arquivos de Swap:** os arquivos de swap (ou de troca) quando utilizados fornecem dados sobre aplicações, nome e senha de usuários, entre outros tipos de dados;
- **Arquivos de Dados:** são aqueles arquivos gerados por *softwares* como editores de texto, planilhas, agendas, etc.;
- **Arquivos de Hibernação:** arquivos de hibernação são criados para preservar o estado do sistema e contêm dados sobre a memória do dispositivo e os arquivos em uso – esses arquivos são utilizados para restaurar o sistema;
- **Arquivos de Log:** normalmente os sistemas operacionais registram diversos eventos relacionados ao sistema. Além disso, as aplicações também geram os seus próprios arquivos de log, nos quais são registrados dados como horário de acesso e de inicialização de serviços e transações, entre outros.

Durante a aquisição dos dados mencionados acima é muito importante manter a integridade dos atributos de tempo *mtime* (*modification time*), *atime* (*access time*) e *ctime* (*creation time*) – denominados de MAC *Times* - que estão relacionados aos arquivos e diretórios. [Farmer and Venema 2006]. Segue abaixo uma breve descrição destes atributos:

- **Modificação:** registro da data e hora em que ocorreu a última alteração no arquivo;
- **Acesso:** registro da data e hora em que ocorreu o último acesso ao arquivo;
- **Criação:** registro da data e hora em que o arquivo foi criado, entretanto, quando um arquivo é copiado de um local para outro em um sistema, o registro de criação assume a data e hora do destino e as informações de modificação permanecem inalteradas.

Essas informações são úteis para identificar o que ocorreu em um incidente, mas também são muito suscetíveis a alterações. Por exemplo, o simples acesso a um diretório altera o atributo atime e pode induzir a equívocos durante as fases seguintes.

1.5 Exame dos Dados

Após a restauração da cópia dos dados, o perito inicia o exame dos dados coletados e faz uma avaliação dos dados encontrados, incluindo os arquivos que haviam sido removidos e foram recuperados, arquivos ocultos e fragmentos de arquivos encontrados nas áreas livres ou nas áreas não utilizadas das mídias. Esse exame minucioso dos dados coletados tem como finalidade localizar, filtrar e extrair somente as informações que possam de alguma maneira, contribuir para a reconstrução dos eventos que deram origem à investigação. A seguir, serão descritos as técnicas envolvidas nesta fase do processo.

Extração dos dados

A extração manual dos dados é um processo difícil e demorado, pois exige do perito conhecimento aprofundado, principalmente, sobre o sistema de arquivos. Entretanto, existem algumas ferramentas disponíveis que podem automatizar o processo de extração dos dados, bem como na recuperação dos arquivos deletados.

Localização de arquivos

A tarefa de localização e identificação do conteúdo dos diversos tipos de arquivos, com os quais o perito irá se deparar durante a investigação, pode ser facilitada se o mesmo possuir um bom conhecimento dos diversos formatos de arquivos existentes, por exemplo, uma extensão JPG identifica um arquivo gráfico, uma extensão mp3 identifica um arquivo de áudio. Mas, os usuários podem alterar a extensão de qualquer tipo de arquivo, por exemplo, renomear um arquivo de texto para a extensão mp3. Além disso, esses arquivos podem armazenar outros dados, vide aplicação de técnicas de esteganografia em áudio, vídeo e arquivos de imagem.

Os dados armazenados nos arquivos podem ser identificados com maior precisão, utilizando ferramentas de análise de cabeçalhos. O cabeçalho de um arquivo contém assinaturas particulares que possibilitam identificar qual o tipo de dado que o arquivo contém, podendo também indicar se ele foi cifrado. Uma prática comum utilizada pelos atacantes é renomear a extensão dos arquivos, entretanto comandos como o file permite identificar o tipo de arquivos independentemente do tipo de tipo de extensão.

A criptografia está freqüentemente presente entre os desafios enfrentados pelos peritos. Os usuários podem cifrar arquivos, pastas, volumes ou partições para que outras pessoas não possam acessar o seu conteúdo sem conhecer a chave ou a senha. Em alguns casos, não é possível decifrar esses arquivos, pois, mesmo com a ajuda de ferramentas como *John the Ripper*, esta tarefa pode exigir um tempo excessivo para a descoberta da senha [Farmer and Venema 2006].

Já para identificar e localizar arquivos que tenham sido submetidos à esteganografia, normalmente, a procura se dá nos registros dos metadados, através de

histogramas. Outra evidência é a presença de programas de esteganografia armazenados no equipamento [Kent et al. 2006]. Uma vez determinada a presença de arquivos que tenham sido submetidos à esteganografia, é importante empregar técnicas de esteganoanálise a fim de recuperar os dados ocultados.

Análise dos Dados

A etapa de análise das informações, muitas vezes, ocorre paralelo à etapa de exame, pois, conforme as evidências vão sendo identificadas e extraídas dos dados, o perito tem condições de efetuar um cruzamento e correlacionamento entre as mesmas, a fim de estabelecer e recriar o(s) evento(s) que estão sendo investigado(s). A correlação das evidências tem o propósito de responder às perguntas-chave que normalmente dão origem a uma investigação: quando e como um fato ocorreu e quem é o responsável pelos mesmo.

A escolha das ferramentas a serem utilizadas nesta fase depende de cada caso. Por exemplo, para investigar ataques ou tentativas de invasão em sistemas informatizados, serão necessárias ferramentas que auxiliem na identificação da origem do ataque. Uma vez determinada a origem dos ataques, através do endereço IP utilizado no ataque por exemplo, é necessária a identificação do responsável. Esta última pode exigir a utilização de outras ferramentas. No caso do responsável pelo endereço do atacante ser um ISP (*Internet Service Provider*), será necessária a solicitação de um mandado judicial, solicitando ao ISP informações a respeito do seu cliente que utilizava o endereço IP identificado no início da investigação.

Todas as etapas e conclusões sobre as análises realizadas devem ser devidamente registradas e ao final anexadas ao laudo pericial.

Interpretação dos Dados

Durante as etapas iniciais de uma investigação, são gerados documentos específicos referentes às atividades realizadas em cada fase. Ao longo dessa documentação, será necessário identificar somente as informações que sejam especificamente relevantes à investigação e organizá-las em categorias. A seguir, serão descritos alguns procedimentos que podem ser benéficos à organização da documentação necessária para a confecção do laudo pericial [Kent et al. 2006].

- Reunir todas as documentações e anotações geradas nas etapas de coleta, exame e análise dos dados, incluindo as conclusões prévias já alcançadas;
- Identificar os fatos que fornecerão suporte às conclusões descritas no laudo pericial;
- Criar uma lista de todas as evidências analisadas, para que as mesmas sejam enumeradas no laudo pericial;
- Listar as conclusões que devem ser relatadas no laudo pericial;
- Organizar e classificar as informações recolhidas para garantir a redação de um laudo conciso e inquestionável.

Redação do Laudo

Posteriormente à organização devida de todas as informações, inicia-se a redação do laudo pericial. É imprescindível que resultado da investigação seja registrado de forma clara e concisa, evitando a utilização de termos técnicos complexos ou expressões somente conhecidas por pessoas ligadas à tecnologia. A seguir, são referidas algumas das seções e informações que podem auxiliar na redação do laudo pericial [Kent et al. 2006].

- Finalidade do relatório: Explicar claramente os objetivos do laudo;
- Autor do relatório: listar todos os autores e co-autores do relatório, incluindo suas especialidades e responsabilidades, durante a investigação, e informações para contato;
- Resumo do incidente: síntese, explicando o incidente investigado e suas conseqüências. O resumo deve ser redigido de forma que uma pessoa não-técnica, como um juiz ou um júri, compreenda como e quais os fatos que ocorreram e estão sob investigação.
- Evidências: fornecer descrições sobre o estado das evidências: como, quando e por quem elas foram adquiridas no decorrer das investigações.
- Detalhes: fornecer uma descrição detalhada de quais evidências foram analisadas, quais os métodos utilizados e quais as conclusões alcançadas, descrevendo os procedimentos e as técnicas adotados, durante a investigação.
- Conclusões: na conclusão, os resultados da investigação devem ser somente descritos, citando especificamente as evidências que comprovem as conclusões, evitando pormenores excessivos sobre como as evidências foram obtidas, pois essas informações já foram descritas na seção detalhes. A conclusão deve ser clara e não oferecer dupla interpretação.
- Anexos: todas as documentações, referentes à investigação, devem ser anexadas, ao final do laudo, tais como: diagramas da rede, formulários descritivos dos procedimentos utilizados, formulário de cadeia de custódia e informações gerais sobre as tecnologias envolvidas na investigação, para que, em caso de necessidade, possam ser consultadas. Outro detalhe significativo é referente aos anexos. Eles devem fornecer todas as informações complementares ao laudo, para que o leitor compreenda completamente o incidente investigado.

Outro aspecto relevante à redação do laudo refere-se ao glossário. O perito, sempre que possível, precisa adicionar um glossário dos termos utilizados no laudo, que poderá esclarecer muitas dúvidas que possam surgir durante a leitura do juiz e/ou dos jurados.

Concluídas todas as etapas de uma investigação, vale lembrar que, durante o decorrer do processo, o perito manterá contato com informações que podem ser sigilosas (por exemplo: segredos industriais ou de justiça). Sendo assim, é necessário que o perito entenda a importância e as suas responsabilidades no que se refere a preservação dos dados.

1.5.1 Ferramentas

Na etapa de coleta dos dados serão abordadas ferramentas utilizadas a fim de salvaguardar os dados contidos no equipamento suspeito, para posterior análise.

Entre as ferramentas mais conhecidas para coleta de dados estão o dd [OpenGroup 2007] (*Disk Definition*) e o dcfldd [DCFL 2007] (*Department of Defense Computer Forensics Lab Disk Definition*). O segundo é uma versão aprimorada do primeiro, criado pelo Laboratório Forense do Departamento de Defesa Americano, na qual foram adicionadas funcionalidades como a geração do *hash* dos dados durante a cópia dos mesmos, visualização do processo de geração da imagem e divisão de uma imagem em partes, a fim de facilitar o armazenamento e o transporte desta.

Visando facilitar a utilização destas duas ferramentas, um *frontend*, chamado *Automated Image & Restore* (AIR) [Gibson 2007], foi desenvolvido. O AIR é uma interface gráfica para os comandos dd/dcfldd que auxilia na criação ou restauração de imagens dos dados (evidências), tanto das mídias conectadas fisicamente ao equipamento, quanto imagens geradas através de uma rede. Além da criação de imagens, o AIR gera e compara automaticamente *hashes* MD5 ou SHA e produz um relatório contendo todos os comandos utilizados durante a sua execução. Uma das grandes funções deste utilitário é eliminar o risco da utilização de parâmetros errados por usuários menos capacitados. Entretanto, a utilização do AIR não elimina a necessidade do perito conhecer basicamente como os utilitários dd ou dcfldd funcionam. A Figura 8 ilustra a tela principal da ferramenta AIR.

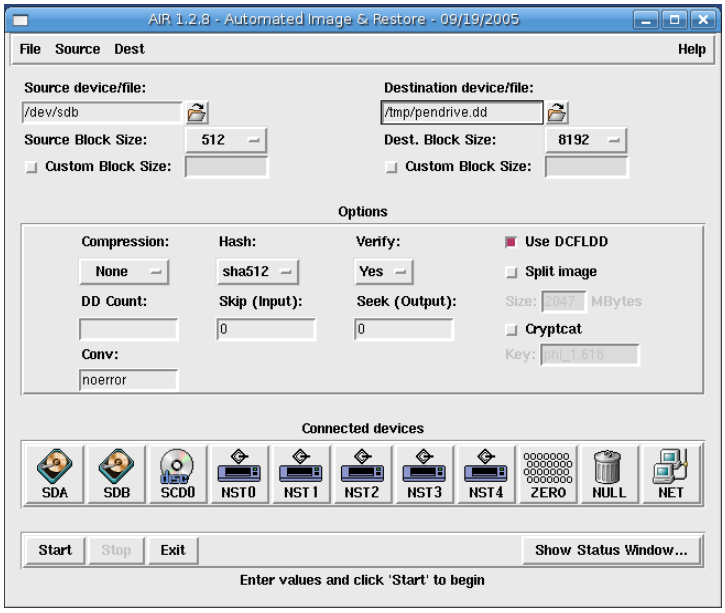

Figura 8. Imagem do utilitário AIR

Ainda dentro da etapa de coleta dos dados existe uma ferramenta, chamada *aimage*, que faz parte da biblioteca de ferramentas AFFLIB [Garfinkel 2007] (*Advanced Forensic Format Library*) e que oferece uma série de benefícios ao perito. Dentre estes benefícios estão: (a) a facilidade de utilização, (b) a automação na geração de hashes dos dados, (c) a redução de 30 a 50% no tamanho dos arquivos de imagem gerados

(através de compactação LZMA (*Lempel-Ziv-Markov Chain-Algorithm*) [Pavlov 2007] e (d) a possibilidade de extração de informações contidas nas imagens sem a necessidade de descompactá-las.

Exame dos Dados

A etapa de exame dos dados pode se tornar muito cansativa para o perito, caso o mesmo não utilize um conjunto de ferramentas adequadas que possibilite a filtragem e o foco de suas habilidades nos dados mais importantes da investigação. Diante deste cenário, o *National Institute of Standards and Technology* (NIST) [NIST 2007] mantém um projeto denominado *National Software Reference Library* (NSRL) [NSRL 2007], o qual disponibiliza uma coleção de assinaturas digitais (*hashes*) de aplicações e arquivos conhecidos. Disponíveis no formato ISO, estas coleções de assinaturas permitem que o perito elimine um conjunto de arquivos coletados e que não sofreram modificações, reduzindo assim a superfície de análise. Por exemplo, em uma investigação suponha que seja gerada a imagem de um disco com 240GB de dados. Dentre os arquivos existentes provavelmente vários deles serão parte dos arquivos do sistema operacional sendo utilizado, além de um conjunto de arquivos pertencentes aos programas instalados no equipamento. A utilização destas bases permite que o perito elimine os arquivos cujos *hashes* casem com os presentes na base de dados.

Atualmente, diversas ferramentas disponíveis, tanto proprietárias, quanto baseadas em código aberto, já permitem a utilização dos bancos de dados supracitados. Entre elas, pode-se citar o *Encase* [Guidance 2007], a *Autopsy* [Carrier 2007a , Carrier 2007c] e o *pyFLAG* [Collett and Cohen 2007]. A Figura 9 ilustra um exemplo da tela principal da ferramenta Autopsy. Como se pode notar, a interação dá-se através de uma interface web, o que dispensa a necessidade de se instalar um software específico para esta finalidade.

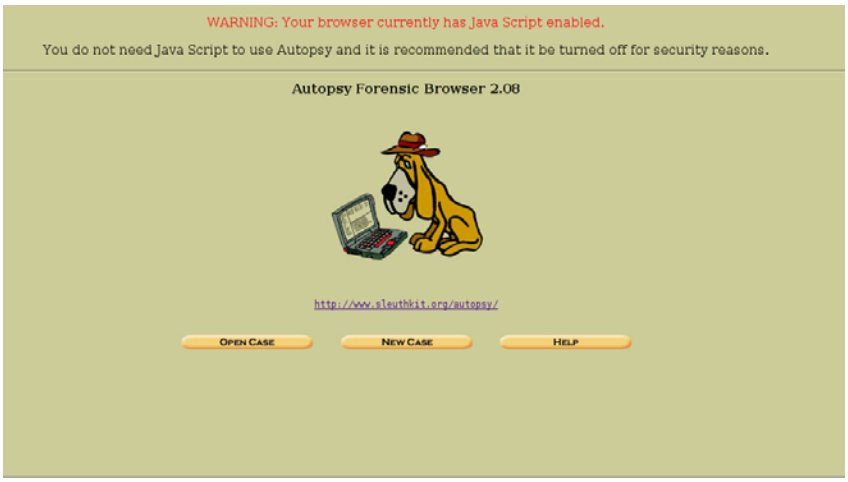

Figura 9. Imagem do utilitário *Autopsy*

Análise dos Dados

Dentre as ferramentas utilizadas na etapa de análise dos dados, é importante ressaltar os utilitários para construção da linha de tempo dos eventos. Nesta categoria, uma

ferramenta particularmente interessante é o *mactime* [Carrier 2007b], que permite que a partir das informações contidas nos metadados dos arquivos e diretórios, os mesmos possam ser classificados de acordo com a sua data de criação ou modificação, fornecendo assim uma visão cronológica dos acontecimentos.

Muitos arquivos importantes que fazem parte dos sistemas operacionais da família Windows não possuem uma clara explicação de suas estruturas dificultando assim o acesso e a compreensão do conteúdo dos mesmos. Diante desta constatação foram desenvolvidas algumas ferramentas capazes de amenizar algumas destas dificuldades e que podem ser verificadas na pagina da *Foundstone* [Foundstone 2007]. Dentre as ferramentas disponíveis neste site pode-se ressaltar: Pasco [Jones 2007b] e Galleta [Jones 2007a].

O utilitário Pasco foi criado com a finalidade de analisar os índices dos arquivos do Internet Explorer. Esta ferramenta analisa gramaticalmente as informações contidas nos arquivos index.dat, exportando os resultados em um formato de texto padrão, inteligível por humanos e que utiliza como delimitador de campos o caractere |. Dessa forma, pode-se analisar as informações do arquivo índex.dat com a ajuda de outros softwares. Após a importação destas informações o perito poderá facilmente verificar a ocorrência de situações como, por exemplo, o acesso a sites ou conteúdos proibidos, ou até mesmo a utilização dos recursos da empresa em benefício próprio durante o horário de trabalho. A Figura 10 ilustra o local onde o arquivo index.dat fica armazenado, dependendo do sistema operacional sendo utilizado.

Operating System	*File Path(s)*
Windows 95/98/Me	\Windows\Temporary Internet Files\Content.IE5\ \Windows\Cookies\ \Windows\History\History.IE5\
Windows NT	\Winnt\Profiles\<username>\Local Settings\Temporary Internet Files\Content.IE5\ \Winnt\Profiles\<username>\Cookies\ \Winnt\Profiles\<username>\Local Settings\History\History.IE5\
Windows 2K/XP	\Documents and Settings\<username>\Local Settings\Temporary Internet Files\Content.IE5\ \Documents and Settings\<username>\Cookies\ \Document and Settings\<username>\Local Settings\History\History.IE5\

Figura 10. Localização do arquivo *index.dat* do Internet Explorer

Em algumas situações o perito necessita reconstruir a seqüência das ações realizadas via web por um suspeito, na qual os arquivos *cookies* do Internet Explorer poderão fornecer valiosas informações. Neste caso, a ferramenta Galleta [Jones 2007a] é capaz de analisar os *cookies* existente em uma máquina e separar as informações úteis em campos para que possam ser manipuladas por outros programas. A Figura 11mostra onde os *cookies* são armazenados, dependendo da versão do Windows sendo utilizada.

Operating System	*Cookie File Location*
Windows 2000/XP	C:\Documents and Settings\<*username*>\Cookies
Windows 95/98/ME	C:\Windows\Cookies

Figura 11. Localização dos *Cookies*

1.5.2 Métodos e Ferramentas Anti-Forense

Os métodos anti-forenses tem como objetivo deliberadamente destruir, ocultar ou modificar as evidências existentes em um sistema a fim de dificultar o trabalho realizado pelos investigadores [Harris 2006]. A seguir uma breve descrição destes métodos.

Destruição dos Dados

Para impedir, ou pelo menos dificultar, a recuperação dos dados os atacantes utilizam ferramentas (conhecidas como *wiping tools*) para remoção dos dados, tais como: *wipe*, *secure-delete*, *pgp wipe* e *The Defiler's Toolkit*. Essa categoria de ferramentas emprega uma variedade de técnicas para sobrescrever o conteúdo dos arquivos, por exemplo, gravar dados de forma randômica e sobrepor o conteúdo dos arquivos com bytes nulos. Essas ferramentas também alteram o *inode* dos arquivos o que torna a tarefa de recuperação dos arquivos ainda muito complexa, embora seja possível [Harris 2006]. Além da destruição lógica, o atacante, em alguns casos, pode danificar fisicamente as mídias utilizadas – o que dificulta e muitas vezes impossibilita a recuperação dos dados.

Ocultação dos Dados

Os dados de um arquivo podem ser escondidos pelo menos de duas formas: (a) fragmentando um arquivo e armazenando esses fragmentos em espaços não alocados ou naqueles marcados como *bad blocks* e (b) utilizando recursos como *Alternate Data Stream* (ADS), existente em sistemas de arquivos NTFS, que possibilitam esconder arquivos que não serão visualizados por comandos de listagem de conteúdo de diretórios dentro de outros arquivos, por exemplo executáveis [Zadjmool 2004].

Além desses métodos, a aplicação de criptografia e esteganografia em arquivos de texto, imagem, vídeo e áudio representam uma barreira difícil de ser superada, pois exigem tempo e recursos nem sempre disponíveis para identificação dos dados ocultos. Por exemplo, utilizar ferramentas de esteganoanálise em uma mídia de 80GB requer muito tempo e na prática nem sempre é algo viável de se realizar. O mesmo ocorre quando se trata de arquivos criptografados [Harris 2006].

Os *rootkits*, conforme já mencionados neste capítulo, implementam métodos eficientes para ocultar informações como arquivos e dados sobre os processos em execução no sistema operacional.

Modificação dos Dados

Os métodos mais comuns para realizar a modificação dos dados são: alterar a extensão e o conteúdo do cabeçalho dos arquivos [Harris 2006]. A troca da extensão pode ser facilmente detectável por comandos dos sistema operacional como file, já a modificação do cabeçalho altera a assinatura do arquivo e, portanto, impede que as ferramentas associem o conteúdo do cabeçalho a um determinado tipo de arquivo.

Outros métodos de modificação incluem a alteração dos atributos de tempo, através de ferramentas como *touch* e *timestamp*, e ataques de colisão em *hash* do tipo MD5, que podem ser utilizados por atacantes para criar arquivos com valores de *hash* idênticos, o que permite substituir arquivos legítimos por arquivos com códigos maliciosos, entre outras ameaças [Wang and Yu 2005].

Os métodos anti-forense mencionados nesta seção são implementados em diversas ferramentas tais como *Metasploit Anti-Forensic Investigation Arsenal* (MAFIA) e *Windows Memory Forensic Toolkit* [Harris 2006].

1.6 Estudos de Caso

Com base em investigações realizadas pelos autores, quatro estudos de caso serão apresentados nesta seção. Estes casos são verídicos, entretanto, algumas informações são omitidas com o objetivo de preservar a identidade das partes envolvidas. No primeiro caso será mostrado um acesso remoto de uma empresa à outra, considerado pela denunciante como indevido (sem permissão). Nesse processo, serão explicadas passo a passo todas as etapas da forense computacional. No segundo caso será mostrada uma investigação de um possível roubo de informações, mais especificamente, de dados bancários (onde o uso de um *malware* pode ter sido utilizado). Já no terceiro e no quarto caso serão mostradas investigações para a descoberta da origem de e-mails. A diferença entre os dois últimos casos é que no terceiro não se tem o cabeçalho do e-mail para análise, mas há um computador suspeito. Já no quarto caso, o perito possui o computador da vítima para análise, logo pode analisar o cabeçalho do e-mail.

1.6.1 Acesso Indevido

O primeiro caso trata do acesso indevido a um sistema, envolvendo a empresa acusada, denominada “invasora”, a empresa que realizou a denúncia, denominada como “vítima” e um sistema, denominado “sistema X”. O sistema X possui informações fundamentais para determinado ramo empresarial. Assim, o acesso ao mesmo é controlado e existe a cobrança de uma mensalidade. Para evitar que mais de uma empresa utilize o sistema com o pagamento de apenas uma mensalidade, a autenticação é baseada no endereço IP.

Em determinado momento os responsáveis pela auditoria do sistema X verificaram que havia mais de uma empresa acessando o sistema, sendo que aquela que não possuía cadastro estava acessando através daquela que possuía. Ou seja, uma empresa estava conectando-se àquela que possuía acesso ao sistema X e a partir dela acessava o sistema. Neste momento, o acesso foi bloqueado, ocasionando grande prejuízo à empresa “invadida”, visto que ela possuía clientes que dependiam de informações acessadas no sistema X.

A empresa vítima verificou em seus *logs* o acesso remoto da empresa invasora, os imprimiu e levou às autoridades para investigação. Contudo, *logs* impressos ou capturados na empresa invadida não são consideradas evidências consistentes, visto que *logs* podem ser alterados facilmente com os devidos cuidados (datas de criação/alteração de arquivo), podendo burlar qualquer situação.

Diante dos fatos, as autoridades decidiram apreender os computadores da empresa invasora, o que pode ser considerado por muitos uma decisão arriscada, visto que a empresa dependia do uso dos mesmos para trabalhar e o backup do sistema e informações poderiam estar nelas. Tratava-se de dez computadores e havia certa urgência na conclusão da investigação. Então, foi decidido que cada perito realizaria a perícia em um conjunto de máquinas (em paralelo) e, toda informação considerada importante deveria ser compartilhada com todos para facilitar a correlação de eventos.

A metodologia aplicada na perícia foi a *post mortem forensic*, visto que os computadores foram enviados aos peritos. Caso algum perito fosse requisitado para comparecer no local da apreensão, a metodologia *live forensic* poderia ser aplicada, podendo inclusive detectar o flagrante de conexões efetuadas entre as empresas envolvidas.

Seguindo as etapas explicadas na seção 3, primeiramente todos os computadores foram fotografados, identificados e seus componentes foram descritos. Todas as mídias encontradas foram associadas ao computador em que se encontravam conectadas. Por exemplo, se dois discos rígidos (HDs) foram encontrados no computador identificado como “CPU01”, os HDs podem ser identificados como “HD01-CPU01” e “HD02-CPU01”. Cabe salientar que as fotografias devem ser tiradas também do interior dos computadores, identificando os componentes que o compõem (mesmo que a descrição também seja feita) e, como estava o estado das conexões (ex: se todos HDs estavam conectados, se um drive de CD-ROM ou DVD-ROM estava conectado, entre outros). As fotografias são importantes no sentido de documentar o passo a passo realizado pelo perito e ajudar a entender algumas situações. Por exemplo, pode-se encontrar conteúdo totalmente fora de contexto em um HD. Já em outro HD, no mesmo computador, pode-se encontrar conteúdo coerente com a investigação. Muitas vezes, analisando as fotografias realizadas do interior do computador é possível verificar que um HD estava desconectado, talvez por ser um HD antigo e tiver sido desativado.

Se tivesse sido solicitada a presença do perito no local da apreensão, o ambiente onde os equipamentos se encontravam instalados, os próprios computadores, dispositivos de conexão, cabos, mídias e tudo o que pudesse ajudar na montagem do cenário poderia ser fotografado. Além das fotografias, seria possível ainda a identificação dos computadores de acordo com a localização. Por exemplo, se foi encontrado na recepção (possivelmente seria o computador da secretária), na sala com identificação de “Gerência”, “Contabilidade”, em um armário de telecomunicações (possivelmente um servidor), etc.

Depois de realizada a identificação do material questionado, foi realizada a coleta do conteúdo de cada mídia. Para isto foi utilizada a ferramenta Encase, uma das mais utilizadas na área de forense computacional. Contudo, poderiam ser utilizadas ferramentas baseadas em *software* livre. Mas, como mencionado na Seção 1.4, não

existe, ou não existia no momento da perícia, uma distribuição Linux contendo um bom conjunto de ferramentas para forense computacional.

Com um disco de boot do Encase é possível realizar uma cópia da imagem (bit-a-bit) de uma mídia para outra, sem contaminar a mídia questionada. Para certificar-se de que a integridade não foi afetada, a opção de geração de *hash* foi ativada no momento da geração da cópia. O *hash* é gerado na mídia questionada e, depois de realizada a cópia, é gerada também nesta, para que seja possível o confronto dos dois códigos gerados e assim garantir que não houve escrita na mídia questionada. Há todo este cuidado para que no caso de um pedido de contra-prova, pela parte da defesa, em um julgamento, não seja constatada uma possível escrita na mídia questionada após a apreensão da mesma. Assim, a defesa não pode alegar que alguém possa ter colocado dados que incriminem o acusado durante a cadeia de custódia (após o desligamento dos computadores até a realização da perícia).

Depois de realizada a coleta dos dados, foi possível realizar o exame dos mesmos. Como o objetivo da perícia era verificar o acesso da empresa “invasora” à empresa “invadida” e os peritos possuíam em mãos impressões de *logs*, o primeiro passo foi procurar em diretórios onde geralmente existem *logs*, típico em sistemas Linux. No entanto, a maioria dos computadores possuía sistema operacional da família Windows. Em um primeiro momento, poderia-se pensar em não analisar estes computadores e partir apenas para os que possuíam sistema Linux. Entretanto, isto seria uma irresponsabilidade dos peritos (deduzir onde tem e onde não tem evidências). Os computadores com sistema Windows possuíam diversos e-mails comerciais, documentos, entre outros. Durante este exame, os peritos verificaram dentro da imagem gerada pelo Encase (semelhante ao Windows Explorer) o que poderia conter evidências, olhando rapidamente alguns arquivos. Contudo, a visualização de muitos tipos de arquivos é prejudicada dentro desta ferramenta, então todos diretórios e arquivos candidatos a conterem dados importantes para investigação foram selecionados e extraídos. Fora da imagem é possível analisar com mais detalhes os arquivos, utilizando ferramentas específicas, como editores de texto e planilhas eletrônicas, visualizadores de imagens, e-mails e históricos de acesso a páginas Web.

Com a análise dos dados extraídos, foi verificado que havia uma troca de e-mails entre as duas empresas. Ainda, foi encontrado um contrato onde constava um acordo comercial de acesso ao sistema X pelas duas. Com a análise mais aprofundada dos e-mails e documentos encontrados foi possível elaborar uma lista de palavras-chave, contendo nomes de funcionários, endereços de e-mail, nomes de empresas, entre outros. Estas chaves foram colocadas no Encase e uma busca foi realizada. Foram encontrados arquivos excluídos, trechos de texto encontrados em parte do disco não alocada pelo sistema de arquivos e trechos de texto encontrados em unidades de alocação onde os arquivos não utilizavam todo o espaço destinado a ela. Ou seja, havia e-mails e documentos excluídos, sendo que alguns puderam ser totalmente recuperados (pois não havia escrita de outros arquivos na mesma porção do disco). Já outros arquivos foram sobrescritos em partes do disco e, desta forma, apenas alguns trechos puderam ser recuperados.

Além das constatações já relacionadas, ainda foi possível verificar em alguns casos evidências de formatação de discos e redimensionamento de algumas partições. Quanto mais se encontrava evidências de comunicação entre as empresas, mais se

encontrava palavras-chave. Assim, mais buscas eram realizadas. Contudo, nenhum indício de acesso entre as empresas havia sido encontrado até então. Tudo indicava que as máquinas com sistema Windows eram apenas utilizadas por pessoas da área administrativa/comercial. Porém, estas informações foram úteis para mostrar que se houve acesso entre elas, tudo indicava que era um acesso lícito, pois havia inclusive um contrato entre elas. Continuando as buscas, acabou sendo encontrado um desentendimento (e-mails e documentos de texto) e a parceria teria sido rompida.

Continuando com a perícia em máquinas com sistema Linux, foi verificado que essas não possuíam e-mails ou dados comerciais, indicando que se tratavam de servidores (foram identificados serviços típicos de um provedor de acesso Internet - ramo da empresa acusada). Foram realizadas buscas por comandos e endereços IP que poderiam gerar os *logs* em formato impresso, conforme mencionado anteriormente. Para o desfecho do caso, finalmente foi encontrado um comando de acesso ao endereço IP da empresa vítima. No entanto, em nenhum momento foi encontrado algum indício de invasão propriamente dita, e sim, um acesso remoto com um usuário e senha legítimos.

Após montar a cronologia dos eventos, reunir todos os arquivos, trechos de texto e outras informações consideradas relevantes para a investigação, iniciou-se então a elaboração do laudo técnico. No laudo foi relatado tudo o que foi mencionado anteriormente, informando a metodologia adotada, como foi realizada a cópia dos dados com a garantia de integridade, a análise do conteúdo e a conclusão. Basicamente, foi informado no laudo que:

- havia um contrato de parceria entre as empresas envolvidas, constatado através de trocas de mensagens e documentos;
- foi constatado certo desentendimento entre as empresas e a parceria teria sido interrompida;
- foram verificadas seqüências de comandos compatíveis com a geração dos *logs* impressos pela empresa vítima, sendo as datas posteriores ao possível desentendimento relatado;
- não foi constatado nenhum tipo de ataque, entretanto, foram constatados acessos legítimos utilizando credenciais autênticas e válidas.

Para complementar, todos os arquivos e dados considerados relevantes para a elaboração do laudo foram gravados em CD e anexados ao mesmo para que a investigação e as partes envolvidas pudessem analisar e comparar com as informações que foram relatadas. Ainda, para garantir a integridade do CD foi gerado um código *hash* para cada arquivo, esses códigos foram gravados em um arquivo. Um novo *hash* foi gerado, no arquivo de *hashes*. Este “ *hash* dos *hashes*” foi adicionado ao laudo juntamente com a explicação de como comprovar a integridade dos arquivos. No CD foi gravado um *software* livre que gera códigos *hash* utilizando o algoritmo MD5 e instruções de como utilizá-lo. Este cuidado é tomado porque durante um processo judicial advogados de defesa podem solicitar uma cópia do laudo para analisar e, após um período determinado pela Justiça, o devolver. Uma cópia do laudo (impressa) pode ser facilmente realizada e entregue ao advogado, mas uma cópia do CD não é feita, sendo assim este pode ser levado. Uma cópia alterada pode ser facilmente criada contendo rótulo (etiqueta) falsificado e entregue novamente à Justiça. Se isto for

realizado, a qualquer momento o teste da integridade dos arquivos pode comprovar a fraude. Por este motivo, é fundamental ter uma cópia, para no caso de constatação de fraude, enviar um novo CD anexo.

1.6.2 Malware

O segundo caso trata de uma suspeita de fraude bancária. Um funcionário de confiança de uma empresa possuía os dados da conta bancária e senha da mesma. Após a conferência de um extrato bancário, foi constatada a transferência de uma grande quantia de dinheiro para outra conta. Como apenas este funcionário e o dono da empresa sabiam a senha da conta bancária e ambos garantiam que a transferência não tinha sido feito por eles, foi instaurado inquérito policial.

Após investigações, foi constatado que o dono da conta para a qual foi feita a transferência era uma pessoa com poucos recursos, semi-analfabeto, e que a conta havia sido aberto há poucos dias. Este foi interrogado e falou que um desconhecido lhe ofereceu uma pequena quantia em dinheiro para que ele abrisse uma conta em determinado banco e que entregasse o cartão eletrônico ao tal desconhecido. Depois de ter entregado o cartão, o cidadão interrogado afirma que nunca mais viu ou teve contato com o desconhecido. Apenas soube descrever algumas características físicas do mesmo, o que não ajudou muito para a polícia. Este é um típico caso de fraude bancária, onde um “laranja” com pouco grau de instrução é utilizado para abrir uma conta. Os fraudadores transferem então quantias monetárias para essa conta e, de posse do cartão da vítima, retiram o dinheiro.

Neste caso, a empresa vítima solicitou ao banco o ressarcimento do valor transferido, alegando que alguma fraude eletrônica deveria ter ocorrido (já que ninguém mais saberia a senha da conta além das duas pessoas mencionadas anteriormente). O banco concordou, entretanto, a fraude deveria ser provada. Por se tratar de suspeita de crime e não haver nenhum suspeito, a solução adotada foi apreender (ou neste caso, solicitar) o computador utilizado pela vítima na empresa. A suspeita era que o computador estaria infectado por algum *malware*, situação comum na época (atualmente ainda é).

Seguindo as etapas já descritas neste capítulo, primeiramente foi realizada a coleta dos dados com uma cópia bit-a-bit e garantia de integridade. Como o objetivo estava bem definido, começou-se a análise das caixas de e-mail. Após algum tempo de análise, foi encontrada uma mensagem de *phishing scam* solicitando ao usuário para clicar em um determinado link. Para verificar se o link ainda estava ativo, foi clicado no mesmo. O link já não estava mais ativo. No entanto, era possível verificar o nome do arquivo que seria baixado (exemplo: fotos.exe). Procurou-se então pelo arquivo fotos.exe na imagem (cópia bit-a-bit) e o mesmo foi encontrado. Alguns *softwares* antivírus foram executados e constatou-se que o arquivo continha um *keylogger*, o BPK, descrito na seção 2. Uma pesquisa foi realizada e foi constatado que o BPK foi desenvolvido para monitorar, entre outros, filhos e esposo(a), segundo o site do fabricante [BlazingTools 2007]. Aparentemente, seria um *software* para uso legítimo, muito utilizado para monitorar funcionários de uma empresa, encaminhando *logs* a um determinado e-mail ou por FTP, por exemplo. Para o caso de monitoramento de funcionários no Brasil, o uso deste *software* seria possível, conforme notícias do

Tribunal Superior do Trabalho [TST 2005] e o processo E-ED-RR - 613/2000-013-10-00.7 do Tribunal Superior do Trabalho [TST 2006].

Toda a explicação do parágrafo anterior serve para mostrar que mesmo um *software* desenvolvido para o uso legítimo pode ser utilizado de forma ilegítima, ou seja, sem o consentimento do usuário. Isto porque o BPK pode ser instalado de forma "camuflada", sem que se perceba. Então, muitos fraudadores utilizam-se desta característica para que usuários instalem tal ferramenta em seu computador e envie dados para algum destino. Para isto, o fraudador deve realizar algumas configurações, como por exemplo, capturar teclas digitadas e cliques do mouse sempre que o usuário entrar em determinado site, enviando os dados capturados para um determinado servidor de FTP a cada intervalo de tempo ou volume de dados.

Voltando ao caso da empresa infectada, depois de detectado o *keylogger*, este foi extraído para análise. Nas configurações dele, foram encontrados:

- armazenamento de *logs* (teclas digitadas) e telas (região de uma quantidade pixels ao redor do clique do mouse) a partir do acesso a seis sites de bancos pré-definidos;
- envio dos arquivos gerados para um servidor FTP, com usuário e senha pré-definidos.

A partir do momento que foi verificado que o computador estava realmente infectado e que a instalação dele foi realizada momentos depois do recebimento do e-mail segundo a data e hora analisados (indicando que o e-mail induziu à execução do *malware*), partiu-se para uma nova busca. O que os peritos estavam a procura neste momento era se os dados relativos à conta bancária realmente teriam sido enviados ao servidor de FTP previamente mencionado. Foi realizada então uma busca por palavras-chave sem a utilização de expressões regulares, pois se procuravam palavras específicas, incluindo usuário e endereço IP do servidor ig.com. Para a satisfação dos peritos, foram encontrados indícios na memória virtual, indicando conexões realizadas ao servidor de FTP mencionado e com o usuário configurado. Inclusive a senha pôde ser verificada, visto que não havia criptografia. Por se tratar de memória virtual, não foi possível extrair grande quantidade de informação útil, sem haver "sujeira" entre um comando e outro. Contudo, os fragmentos encontrados puderam indicar a conexão com o servidor e algumas datas e horários puderam ser coletadas em texto claro.

O servidor FTP encontrava-se em outro país e possuía a modalidade de contas gratuitas. Um teste foi realizado no site da empresa provedora do serviço e uma conta foi criada sem informação de dados pessoais validados, ou seja, foram digitados caracteres aleatórios com um e-mail gratuito e a conta foi criada sem dificuldades.

Todas as informações mostradas neste caso foram relatadas no laudo pericial, indicando inclusive o endereço IP do servidor FTP, caso a investigação tivesse algum acesso a *logs* do servidor para uma nova perícia. No laudo não foi possível concluir que os dados da conta foram enviados a algum destinatário, porque os arquivos de log gerados pelo BPK eram excluídos de tempos em tempos. Mesmo com uma busca por arquivos excluídos, não foram encontrados dados relativos à conta. No entanto, foi possível relatar que o computador estava infectado, que foi infectado logo após o

recebimento da mensagem com *phishing scam*, e que foram realizadas conexões com um servidor FTP previamente configurado no *keylogger*.

1.6.3 Ameaça por E-mail (sem cabeçalho)

O terceiro caso trata de uma ameaça realizada por e-mail a um diretor de uma empresa, com cópia a diversos membros da diretoria. O diretor imprimiu o e-mail e o entregou à polícia, realizando o boletim de ocorrência. Contudo, o cabeçalho do e-mail não foi impresso e o computador contendo o e-mail recebido não foi entregue para a perícia. O conteúdo do e-mail mencionava fatos ocorridos na empresa, citando que uma categoria de funcionários estava correndo perigo e o diretor, além de não tomar providências, ainda protegia os possíveis ameaçadores. Para não ficar vaga a explicação, um exemplo será dado a seguir, lembrando que o exemplo é fictício e serve apenas para ilustrar a situação.

A rodoviária de uma cidade sofre diversos atentados por vândalos, que realizam furtos, são violentos com os cidadãos que transitam por ela, agridem os vigias e ameaçam os mesmos. A polícia alega não ter gente suficiente para garantir a segurança da rodoviária 24h. Os vigias não podem portar e utilizar armas. Um dos vigias, indignado, cria uma conta de e-mail gratuita e envia um e-mail para o responsável da rodoviária, relatando que os vândalos fazem o que querem com os vigias, que ele é um incompetente e outras coisas mais (palavras de baixo calão). Por fim, o ameaça e a sua família também.

Após a análise do e-mail impresso e com depoimento do responsável pela rodoviária, deduz-se que o e-mail foi criado por um funcionário, mais especificamente, um vigia. Sabe-se que os vigias utilizam um computador com acesso à Internet, disponível para os funcionários durante o horário de descanso. A polícia decide, então, apreender este computador e enviar juntamente com o e-mail impresso, à perícia.

Estava traçado então o objetivo da perícia, ou seja, saber se aquele e-mail impresso foi originado na máquina enviada para a investigação. Depois de realizada a cópia bit-a-bit do disco rígido questionado, começou-se a busca. Por se tratar de um e-mail enviado a partir de um site (Webmail), foram realizadas duas atividades em paralelo. Foram selecionadas palavras-chave para busca na imagem. Foram selecionadas palavras com erros de grafia encontradas no e-mail impresso e, também, palavras muito específicas (que raramente seriam encontradas em outro arquivo ou trecho do disco). Enquanto essa busca era processada, foram verificados os arquivos encontrados na pasta de arquivos temporários da Internet.

Diversos acessos a sites de Webmail foram encontrados, com um total de seis usuários diferentes. No entanto, na pasta de arquivos temporários não foi encontrado o e-mail com as ameaças nem mesmo os dados da conta de e-mail que foi utilizada. Antes de continuar a análise “manual”, o processamento da busca foi concluído, mostrando algumas ocorrências para as palavras-chave selecionadas. Analisando as ocorrências, foi encontrado a maior parte do e-mail dentro do arquivo de memória virtual (plataforma Windows), indicando que algum arquivo ou mensagem na Internet foi visualizado no computador. Antes do início da mensagem havia o seguinte metadado:

<?xml:namespace prefix = o ns = "urn:schemas-microsoft-com:office:office"

Este metadado indica a utilização de um dos aplicativos do Microsoft Office. Neste caso, há o indício de que a mensagem teria sido digitada no aplicativo de edição de texto do pacote Office, antes de ser enviado por e-mail. Além destas informações, nada mais foi encontrado. Logo, não houve uma conclusão de que a mensagem foi originada na máquina analisada. Porém foram relatados os indícios encontrados, podendo ajudar na investigação.

1.6.4 Injúria por E-mail (com cabeçalho)

O quarto caso trata de uma mensagem de e-mail contendo injúria, mais especificamente, racismo. O e-mail foi enviado para a amiga da vítima, sendo que a amiga mostrou para a vítima e esta decidiu registrar ocorrência, levando consigo o e-mail impresso. O delegado decidiu então solicitar o computador da amiga para enviar à perícia. Neste caso, foi possível analisar o cabeçalho do e-mail, ilustrado na Figura 12. Alguns dados foram propositalmente alterados para evitar a identificação dos envolvidos.

Figura 12. Exemplo de um cabeçalho de e-mail

Analisando os cabeçalhos (em negrito) de baixo para cima, é possível verificar por quais servidores passou o e-mail questionado. O primeiro servidor é denominado “svr-int.empresa.com.br”, a máquina que enviou o e-mail possui nome “01gecad” e endereço IP privado 192.168.1.22. Na segunda parte do cabeçalho (de baixo para cima) é possível verificar que o servidor “email-2.ig.com.br” recebeu diretamente do servidor de e-mail da empresa e que o endereço IP do servidor de e-mail da empresa é 210.218.115.13. Após, na última parte (a do topo) é possível ver um encaminhamento de um servidor para outro no mesmo domínio “ig.com.br”.

Foi possível consultar na base de dados de domínios brasileiros [NIC.br 2007] e os dados da empresa, como endereço, telefone do responsável, entre outros, puderam ser coletados. Atualmente os dados não são acessíveis por qualquer usuário, porém na época em que foi realizada a perícia, estes dados eram acessíveis. Estes dados foram

colocados no laudo, indicando a origem do e-mail sendo a empresa descrita como detentora do domínio “empresa.com.br”, localizada na mesma cidade onde morava a vítima e o acusado, e, segundo informações que a investigação passou para os peritos posteriormente, esta seria a empresa onde o acusado trabalhava.

1.7 Desafios Atuais em Forense Computacional

Essa seção apresenta as oportunidades de pesquisa identificadas pelos autores durante o processo de revisão bibliográfica que antecedeu a proposta do minicurso em questão. As questões de pesquisa mencionadas estão organizadas de acordo com o processo de investigação apresentado na Seção 1.3.

1.7.1 Coleta dos Dados

Conforme discutido ao longo das seções anteriores o sucesso de uma investigação depende muito da coleta e da preservação da integridade dos dados, portanto esta etapa do processo forense deve ser realizada de maneira sistemática e tão logo as fontes de evidências sejam localizadas. As principais fontes de evidências existentes são: os discos rígidos e a memória física dos *hosts*, os *logs* dos diversos serviços em execução e o tráfego da rede capturado.

Embora existam ferramentas especializadas para coletar e preservar esses tipos de fontes de dados, essa é uma etapa que ainda envolve muito tempo até ser finalizada, pois além do deslocamento do perito até o local da cena do crime é necessário considerar o tempo gasto para coletar os dados de cada um dos hosts apreendidos, o que pode demandar muitas horas, principalmente, se considerarmos que as estações e os servidores podem estar ligados, o que requer uma série de cuidados extras para que não haja a contaminação dos dados coletados (vide seção 1.3.3).

Para facilitar e agilizar a aquisição das evidências em *hosts* que estejam conectados a rede, foram desenvolvidas ferramentas para forense remota que a partir de uma console central são capazes de coletar simultaneamente dados de diversos *hosts*. Entretanto, este tipo de aplicação não é capaz de realizar a cópia completa da memória ou inspecionar as áreas de memória alocadas por um determinado processo [Casey 2006]. Sendo assim, com este tipo de ferramenta não é possível, por exemplo, adquirir evidências localizadas em arquivos abertos ou referentes a processos em execução.

Ainda no que diz respeito a coleta de informações em sistemas ativos, [Carrier 2006] afirma que os *rootkits* representam a principal ameaça e indica como contramedida o uso de *hardware* especializado para realizar a cópia do conteúdo da memória. Por exemplo, o sistema *Tribble* [Carrier 2004] é um cartão PCI que pode realizar a cópia da memória física usando requisições DMA, sem a mediação do *kernel*. Portanto, ainda que o *kernel* do *host* apreendido esteja comprometido, é possível obter a imagem da memória para ser examinada em uma estação forense confiável.

Além das questões recém mencionadas outro tópico que tem recebido a atenção dos pesquisadores refere-se a padronização dos formatos utilizados para armazenar a imagem dos discos rígidos. Segundo [Hosmer 2006] e [Garfinkel 2007] o padrão de fato para copiar informações de uma mídia é o formado denominado “*raw format*”, aquele no qual os dados são copiados setor por setor. Entretanto, nesse caso não são coletados metadados, tais como: o número serial dos dispositivos, a data e o local em que os

dados foram adquiridos e nem mesmo a assinatura digital para garantir a integridade dos dados. Além disso, este formato não diferencia setores em branco daqueles que não são acessíveis e, ainda, não oferece suporte a compactação, o que resulta no desperdício do espaço em disco (aquele para o qual os dados estão sendo copiados).

Segundo publicações do *The Common Digital Evidence Storage Format Working Group* (CDESF) alguns formatos proprietários resolvem parte das questões supracitadas, contudo criam outras limitações, entre elas:

- falta de compatibilidade entre os padrões o que, em uma tentativa de conversão entre os formatos, pode resultar em dados incorretos, perda de metadados e de tempo;
- para os investigadores que não tiverem acesso ao software de leitura do padrão gerado não há como recuperar os dados o que impossibilita o exame e análise dos dados;
- em alguns casos juízes ou advogados podem rejeitar evidências em formatos proprietários, com base na alegação de que esses padrões estão sujeitos ou violam algumas patentes.

O desenvolvimento de padrões abertos bem documentados, que contemplem acesso aos metadados, que garantam a integridade dos dados e cujas evidências possam ser examinadas e analisadas por múltiplas ferramentas, é um desafio em aberto e que representa o foco das pesquisas realizadas pelo grupo de trabalho CDESF, anteriormente mencionado. Outras características desejáveis são:

- a definição da estrutura e implementação de trilhas de auditoria que gerem a cadeia de custódia, dessa maneira seria registrado de forma segura e automática todas as ações relacionadas a aquisição e alteração dos dados adquiridos;
- a implementação das seguintes características de segurança: mecanismos de autenticação, verificação de integridade, controle de acesso as evidências e não-repúdio;

Em [Hosmer 2006] é apresentado o conceito de *Digital Evidence Bag* (DEB) uma espécie de container digital no qual as evidências coletadas são armazenadas. As diferenças existentes entre um container do mundo real e o DEB são as seguintes: o container digital permite a duplicação, cópia, e compartilhamento do seu conteúdo de maneira segura, pois prevê as características de segurança descritas acima, muito embora se reconheça que ainda é necessário continuar pesquisando e aprimorando o conceito de DEB até que seja possível implementar um protótipo com todos os atributos aqui elencados.

Ainda sobre o desenvolvimento de formatos para gerar e armazenar imagens de discos rígidos, cabe mencionar o *Advanced Forensics Format* (AFF), um formato aberto e flexível que armazena a imagem do disco em um conjunto de páginas, o que permite compactar a imagem gerada [Garfinkel 2007]. Além disso, (a) este formato oferece a possibilidade de armazenar os metadados na própria imagem ou em um arquivo separado, (b) está livre de patentes e segredos comerciais e (c) os desenvolvedores disponibilizaram um biblioteca e um conjunto de ferramentas para gerar imagem de

discos e converter formatos de imagens para o padrão AFF. Sendo assim, existe a possibilidade de integrar e desenvolver ferramentas para coleta de dados que ofereçam suporte a este padrão, inclusive este é o objetivo do projeto de acordo com [Garfinkel 2007].

Além das características mencionadas anteriormente é desejável que um formato aberto para armazenamento dos dados seja flexível e aplicável à diferentes formas de evidências digitais (tráfego de rede e *dumps* de memória), ou seja, não esteja restrito somente a dados armazenados nos discos rígidos.

1.7.2 Exame dos Dados

Nessa etapa o primeiro desafio é identificar entre todos os dados armazenados nas mídias coletadas junto as estações e servidores, quais são relevantes para auxiliar na elucidação dos fatos investigados. Essa tarefa é complexa devido, sobretudo, aos seguintes aspectos:

1. a capacidade cada vez maior de armazenamento dos dispositivos;
2. os arquivos podem ter sido criptografados ou esteganografado;
3. a presença de *rootkits* induz o perito a erros no momento de avaliar quais as informações que serão filtradas e encaminhadas para a etapa seguinte do processo de investigação;
4. um arquivo pode ter sido fragmentado e armazenado em espaços não alocados do disco ou ainda marcados, indevidamente, como bad blocks.

Essa série de barreiras faz com que seja necessário muitas horas (ou até mesmo dias) até que seja possível identificar os dados pertinentes ao caso em questão. Com o intuito de localizar tais dados, são utilizadas ferramentas que implementam técnicas de pesquisa baseada na assinaturas dos arquivos. Essas aplicações varrem o disco rígido relacionando as assinaturas dos arquivos do sistema com a sua base de dados, dessa forma é possível identificar: arquivos comprometidos por códigos maliciosos ou com ADS (vide subseção 1.4.3), arquivos criptografados, esteganografados ou protegidos por senhas.

Já para identificar a presença de *rootkits* sem comprometer os dados ou perturbar o ambiente investigado, [Carrier 2006] recomenda o desenvolvimento e utilização de mecanismos de detecção de rootkits baseado em hardware, como o *Copilot* proposto em [Nick 2004].

De acordo com [Casey 2006], atualmente não existem ferramentas para inspecionar e interpretar, de maneira ágil, as estruturas de dados da memória virtual. O desenvolvimento de tais aplicações pode auxiliar o investigador a encontrar senhas de usuários, fragmentos de arquivos visualizados pelo intruso e até mesmo senhas de arquivos criptografados em um curto espaço de tempo.

Outra questão em aberto e que esta relacionada ao exame dos dados, refere-se ao desenvolvimento de métodos que possibilitem identificar a presença de arquivos, cujos fragmentados foram gravados em espaços não alocados do disco – técnica utilizada por algumas ferramentas anti-forense.

Portanto, o desenvolvimento de técnicas e ferramentas projetadas para solucionar as questões mencionados nessa subseção, no menor tempo possível e com baixas taxas de falsos positivos, representam questões de pesquisas relevantes e cujos avanços podem representar um passo significativo para acelerar o processo de exame dos dados e garantir que as informações enviadas à análise sejam confiáveis.

1.7.3 Análise dos Dados

Após conseguir extrair das mídias somente os dados que são pertinentes a investigação o perito ainda tem pela frente outro grande desafio, identificar, entender e reconstruir os fatos ocorridos. Para tal, é necessário desenvolver e aprimorar (a) os métodos de redução de dados, (b) os mecanismos de reconhecimento de padrões de comportamento e (c) as técnicas de correlacionamento de alertas [Forte 2004].

Segundo [Casey 2006] ferramentas proprietárias voltadas à gerência de segurança, tais como: *CS-MARS* e *nFX*, embora não tenham sido projetadas com o propósito de serem utilizada para fins de análise forense, fornecem alguns dos recursos mencionados por [Forte 2004]. Portanto, o desenvolvimento de sistemas especialistas em forense que possua em suas bases de dados informações sobre como: agregar as diversas entradas dos arquivos de log distribuídos pelos ativos da organização em um único evento, correlacionar eventos de forma automatizada e que forneçam informações detalhadas sobre o ocorrido, permitem reduzir o tempo gasto na análise dos dados coletados, pois além de tornar o processo automatizado torna-o menos suscetível a erros, uma vez que não há mais a dependência unicamente do conhecimento do investigador para analisar os dados.

Uma vez que tenham sido empregadas técnicas adequadas para redução e correlacionamentos das evidências, o perito necessita realizar a análise desses dados, o que pode ser feito de forma rápida e fácil através de técnicas de visualização hierárquicas e não hierárquicas ou como uso de diagramas de link.

Técnicas de visualização não hierárquicas mostram dados estatísticos sobre os arquivos existentes em um diretório ou subdiretório sem nenhuma informação sobre o relacionamento entre arquivos e diretórios. Nesse tipo de técnica os arquivos são representados por quadrados, cujas cores indicam o tamanho dos arquivos, tonalidades escuras representam arquivos de tamanho menores, enquanto as tonalidades claras representam os arquivos maiores. Quando o parâmetro utilizado para consulta for um atributo de tempo, os quadrados de cor mais clara representam os arquivos acessados recentemente. Além do tamanho e dos atributos de tempo podem ser utilizados outros parâmetros (por exemplo, o formato do arquivo), o que permite analisar diferentes cenários de forma rápida e agradável [Teelink and Erbacher 2006].

As técnicas de visualização hierárquicas apresentam os dados em uma estrutura de árvore, como ilustrado na Figura 13, mantendo a relação entre os arquivos e os diretórios e possibilitando diferentes métodos de análise, por exemplo, cada arquivo é representado por um quadrado sombreado, cujo tamanho informa, entre outras coisas, qual o percentual do espaço total do diretório é ocupado por este arquivo. O *popup* indica uma discrepância entre o nome e o tipo de arquivo, o que indica uma tentativa de ocultar dados. Além disso, os atributos relacionados a data e hora do arquivo também são exibidos [Teelink and Erbacher 2006].

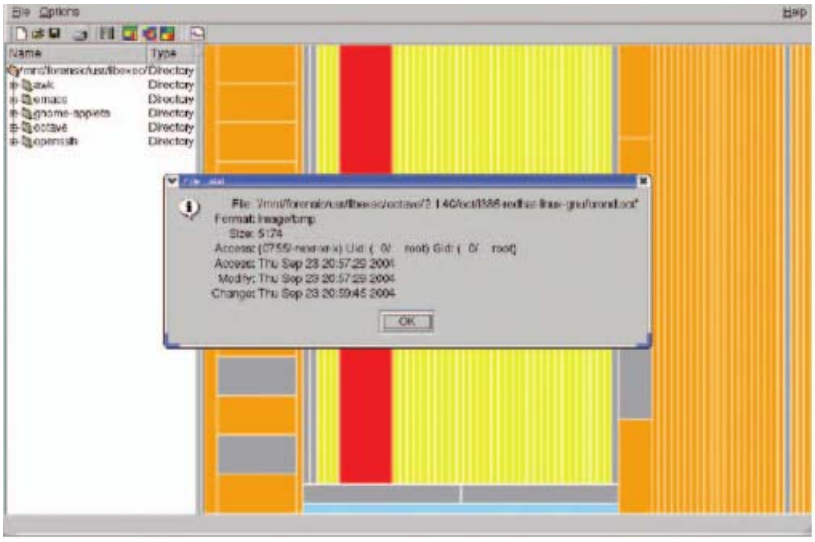

Figura 13. Diagrama hierárquico em estrutura de árvore

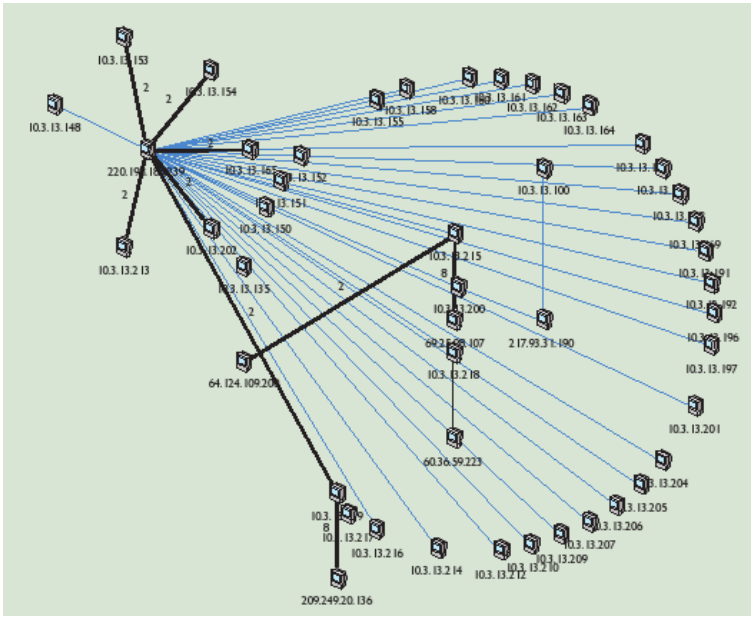

Figura 14. Análise através de diagramas de *link*

O diagrama de link é uma técnica que permite gerar filtros em função das comunicações estabelecidas e visualizar graficamente as informações existentes. No exemplo da Figura 14, os *logs* indicam que o *host*, cujo endereço IP é 220.198.186.239 (no canto superior esquerdo) acessou ou comprometeu as demais estações.

Outra maneira interessante de analisar os dados é exibindo-os ao longo da linha do tempo, o que permite reconstruir os fatos de maneira cronológica, recurso que os autores desse capítulo encontraram apenas na ferramenta proprietária *Analysts Notebook*[3], conforme ilustrado na Figura 15.

[3] http://www.i2inc.com/Products/

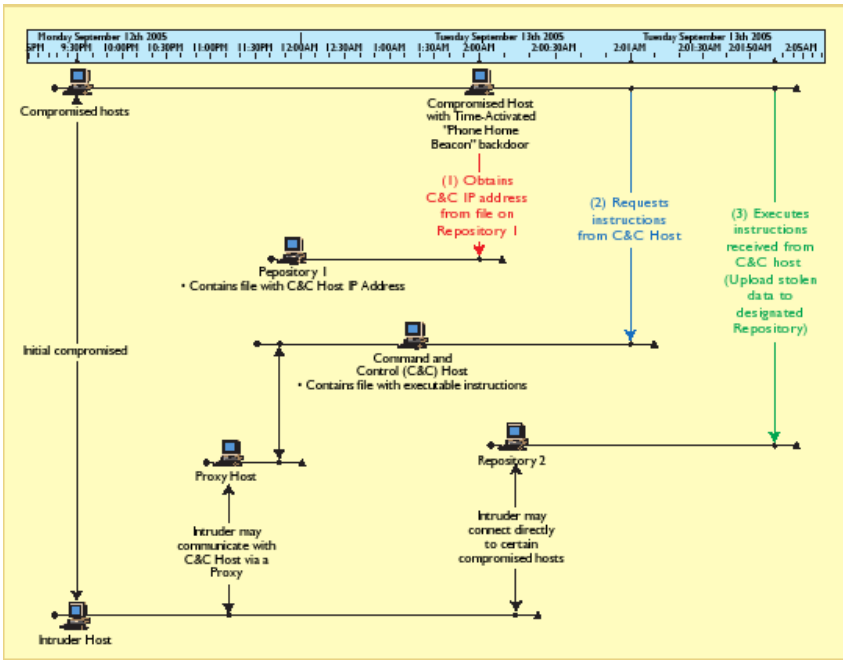

Figura 15. Análise Cronológica de Eventos

1.8 Considerações Finais

Este capítulo apresentou a forense computacional, um tema bastante atual e que tem recebido significativa atenção tanto da comunidade científica quanto da indústria. A seção sobre *malwares* apresentou os principais códigos maliciosos e exemplificou as ações realizadas pelos mesmos.

Na Seção 1.3 foi realizada uma introdução à Forense Computacional e em seguida o processo de investigação forense foi descrito tendo como base as seguintes etapas: coleta e exame dos dados, análise das informações e interpretação dos resultados. Por fim, foram apresentados características e cuidados referentes à *live forensics*. Sobre as quatro etapas mencionadas é importante ressaltar: (a) a importância de salvaguardar e garantir a integridade dos dados coletados, bem como estabelecer e manter a cadeia de custódia do material apreendido, (b) o exame das mídias deve ser realizado com o auxilio de ferramentas que permitam ao perito, no menor intervalo de tempo possível, separar os dados relevantes para (c) realizar a análise do dados, essa etapa requer o correlacionamento das informações geradas em diferentes serviços (Web, FTP e E-Mail) com os alertas gerados pelos mecanismos de proteção (*firewalls* e sistemas de detecção de intrusão) e (d) o processo é concluído com a elaboração de uma laudo técnico que descreve como os procedimentos foram realizados e quais as conclusões obtidas.

No que tange a análise de sistemas ativos, *live analysis*, cabe ressaltar que essa técnica pode fornecer evidências que não estão disponíveis na abordagem tradicional, *post mortem analysis*, contudo, em algumas situações, esse tipo de análise pode gerar resultados pouco confiáveis. Entretanto, casos em que a investigação deve ocorrer em sigilo ou em que é necessário monitorar as ações de colaboradores ou suspeitos, o tipo de abordagem mais recomendado é a *live analysis*.

A Seções subseqüentes apresentaram e discutiram problemas e questões atuais de pesquisa relacionadas a Forense Computacional, algumas totalmente em aberto outras já com alguns trabalhos sendo desenvolvidos, mas ambas extremamente relevantes e com espaço para novos contribuições.

Além dos assuntos mencionados, a Forense Digital – que conforme mencionado na Seção 1.3 possui um escopo mais abrangente do que a Forense computacional no que diz respeito ao tipo dispositivos analisados – também apresenta uma série de questões a serem tratadas, por exemplo: o desenvolvimento e aprimoramento de metodologias, ferramentas e técnicas que ofereçam suporte ao processo de forense em dispositivos móveis como aparelhos celulares e PDAs [Jansen and Ayers 2004].

Referências Bibliográficas

[Adelstein 2006] Adelstein, F. (2006). Live forensics: diagnosing your system without killing it first. Commun. ACM, 49(2):63–66.

[Adleman 1990] Adleman, L. M. (1990). An abstract theory of computer viruses. pages 1–354.

[Aleph-Null 1971] Aleph-Null (1971). Software - practice and experience. volume 1, pages 201–204.

[Bessa 2006] Bessa, L. (2006). Websense revela suas previsões sobre a segurança da internet para 2007. IMS Marketing. Websense, Inc, http://www.websense.com/global/pt/PressRoom/PressReleases/PressReleaseDetail/index.php?Release=0612191332.

[BlazingTools 2007] BlazingTools (2007). Perfect keylogger - easy to use stealth solution for pc and internet surveillance. discover the truth now! http://www.blazingtools.com/bpk.html.

[Bonfante et al. 2007] Bonfante, G., Kaczmarek, M., and Marion, J. Y. (2007) Toward an abstract computer virology.

[Cambridge 2007] Cambridge, A. L. (2007). VNC - Virtual Network Computing from AT&T Laboratories Cambridge.

[Carrier 2007a] Carrier, B. (2007a). Autopsy forensic browser. SourceForge.net, http://www.sleuthkit.org/autopsy/desc.php.

[Carrier 2007b] Carrier, B. (2007b). mactime. SouceForge.net, http://www.sleuthkit.org/sleuthkit/man/mactime.html.

[Carrier 2007c] Carrier, B. (2007c). The sleuth kit. http://www.sleuthkit.org/sleuthkit/desc.php.

[Carrier 2006] Carrier, B. D. (2006). Risks of live digital forensic analysis. Commun. ACM, 49(2):56–61.

[Carrier 2004] Carrier, I. B. (2004). A hardware-based memory acquisition procedure for digital.

[Case and Moelius 2007] Case, J. and Moelius, S. E. (2007). Cautious virus detection in the extreme. In Proceedings of the 2007 workshop on Programming languages and analysis for security (PLAS 2007), pages 47–52, New York, NY, USA. ACM Press.

[Casey 2006] Casey, E. (2006). Investigating sophisticated security breaches. Commun. ACM, 49(2):48–55.

[CERT 1999] CERT (1999). CERT Advisory CA-1999-04 Melissa Macro Virus. CERT Coordination Center (CERT/CC), http://www.cert.org/advisories/CA-1999-04.html.

[CERT 2002a] CERT (2002a). CERT Advisory CA-2001-13 Buffer Overflow In IIS Indexing Service DLL. CERT Coordination Center (CERT/CC), http://www.cert.org/advisories/CA-2001-13.html.

[CERT 2002b] CERT (2002b). CERT advisory CA-2002-30 trojan horse tcpdump and libpcap distributions. CERT Coordination Center (CERT/CC), http://www.cert.org/advisories/CA-2002-30.html.

[CERT.br 2007] CERT.br (2007). Centro de estudos, resposta e tratamento de incidentes de segurança no brasil. Comitê Gestor da Internet no Brasil - CGI.br, http://www.cert.br/.

[Chawki 2005] Chawki, M. (2005). A critical look at the regulation of cybercrime. http://www.crime-research.org/articles/Critical/.

[Cohen 1987] Cohen, F. (1987). Computer viruses: theory and experiments. Comput. Secur., 6(1):22–35.

[Collett and Cohen 2007] Collett, D. and Cohen, M. (2007). Forensic and log analysis gui. SourceForge.net, http://sourceforge.net/projects/pyflag/.

[Crapanzano 2003] Crapanzano, J. (2003). Deconstructing subseven, the trojan horse of choice. Technical report, SANS Institute.

[DCFL 2007] DCFL (2007). dcfldd. Department of Defense Computer Forensics Lab, http://dcfldd.sourceforge.net/.

[Dildog 2007] Dildog (2007). BO2K - Opensource Remote Administration Tool. Cult of the Dead Cow, http://www.bo2k.com/.

[eEye 2007] eEye (2007). eEye Digital Security website. http://www.eeye.com/html/index.html.

[Etrust 2007] Etrust (2007). Pestpatrol anti-spyware. http://www.pestpatrol.com/.

[Farmer and Venema 2006] Farmer, D. and Venema, W. (2006). Perícia Forense Computacional - Teoria e Prática Aplicada. 1 edition.

[Forte 2004] Forte, D. (2004). The art of log correlation. HTCIA Worldwide Conference, http://www.dflabs.com/images/Art_of_correlation_Dario_Forte.pdf.

[Foundstone 2007] Foundstone (2007). Foundstone network security: Risk management. FoundStone. McAfee, Inc, http://www.foundstone.com/us/resources-free-tools.asp.

[FTC 2005] FTC (2005). The US SAFE WEB Act: Protecting Consumers from Spam, Spyware, and Fraud. A Legislative Recommendation to Congress. Federal Trade Commission, http://ftc.gov/reports/ussafeweb/USSAFEWEB.pdf.

[Garfinkel 2007] Garfinkel, S. L. (2007). Advanced forensic format (aff). Simson L. Garfinkel and Basis Technology Corp, http://www.afflib.org/.

[Giacobbi 2007] Giacobbi, G. (2007). The gnu netcat - official homepage. http://netcat.sourceforge.net/.

[Gibson 2007] Gibson, S. (2007). Automated image and restore (air). SourceForge.net, https://sourceforge.net/projects/air-imager/.

[Guidance 2007] Guidance (2007). Encase forensic. Guidance Software, Inc, http://www.guidancesoftware.com/products/ef_index.asp.

[Haagman and Ghavalas 2005] Haagman, D. and Ghavalas, B. (2005). Trojan defence: A forensic view. Digital Investigation, 2(1):23–30.

[Harris 2006] Harris, R. (2006). Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem. In The 6th Annual Digital Forensic Research Workshop (DFRWS 2006).

[Hoglund and Butler 2005] Hoglund, G. and Butler, J. (2005). Rootkits: Subverting the Windows Kernel. Addison-Wesley Professional.

[Holz 2005] Holz, T. (2005). A Short Visit to the Bot Zoo [malicious bots software]. IEEE Security & Privacy Magazine, 3(3):76–79.

[Hosmer 2006] Hosmer, C. (2006). Digital Evidence Bag. Commun. ACM, 49(2):69–70.

[IBM 2007] IBM (2007). Michelangelo madness. IBM Research, http://www.research.ibm.com/antivirus/SciPapers/White/VB95/vb95.distribnode7.html.

[Inman and Rudin 2000] Inman, K. and Rudin, N. (2000). Principles and Practice of Criminalistics: The Profession of Forensic Science (Protocols in Forensic Science). CRC.

[Jansen and Ayers 2004] Jansen, W. and Ayers, R. (2004). Guidelines on PDA Forensics: recommendations of the national institute of standards of and technology. Department of Homeland Security. National Institute of Standards and Technology, http://csrc.nist.gov/publications/nistpubs/800-72/sp800-72.pdf.

[Jones 2007a] Jones, K. J. (2007a). Galleta - an internet explorer cookie forensic analysis tool. Foundstone, Inc, http://www.foundstone.com/us/resources/proddesc/galleta.htm.

[Jones 2007b] Jones, K. J. (2007b). Pasco - an internet explorer activity forensic analysis tool. Foundstone, Inc, http://www.foundstone.com/us/resources/proddesc/pasco.htm.

[Kaspersky 2007] Kaspersky (2007). Malware descriptions: Classic viruses. Kaspersky Lab, http://www.viruslist.com/en/virusesdescribed?chapter=152540474.

[Kent et al. 2006] Kent, K., Chevalier, S., Grance, T., and Dang, H. (2006). Guide to integrating forensic techniques into incident response: Recommendations of the national institute of standards and technology. NIST Special Publication 800-86. National Institute of Standards and Technology (NIST), http://csrc.nist.gov/publications/nistpubs/800-86/SP800-86.pdf.

[Klaus and Nelson 2001] Klaus, S. and Nelson, M. (2001). Métodos para detecção local de rootkits e módulos de kernel maliciosos em sistemas unix. In III Simpósio sobre Segurança em Informática (SSI), São José dos Campos, SP.

[Kolla 2007] Kolla, P. M. (2007). Spybot search and destroy website. http://www.safer-networking.org/pt/index.html.

[Kolter and Maloof 2006] Kolter, Z. J. and Maloof, M. A. (2006). Learning to detect and classify malicious executables in the wild. J. Mach. Learn. Res., 7:2721–2744.

[Kruse and Heiser 2001] Kruse, W. G. and Heiser, J. G. (2001). Computer Forensics : Incident Response Essentials. Addison-Wesley Professional.

[Lavasoft 2007] Lavasoft (2007). Ad-Aware website. Lavasoft AB, http://www.lavasoftusa.com/software/adaware.

[Mclaughlin 2004] Mclaughlin, L. (2004). Bot software spreads, causes new worries. IEEE Distributed Systems Online, 5(6).

[Microsoft 2007a] Microsoft (2007a). Windows defender home. http://www.microsoft.com/athome/security/spyware/software/default.mspx.

[Microsoft 2007b] Microsoft (2007b). Windows sysinternals. Microsoft Technet. Microsoft Corporation, http://www.microsoft.com/technet/sysinternals/default.mspx.

[Neukamp 2007] Neukamp, P. (2007). Fdtk-ubuntubr: Linux forense digital toolkit. http://www.fdtk-ubuntubr.1br.net/.

[Newman 2006] Newman, R. C. (2006). Cybercrime, identity theft, and fraud: practicing safe internet - network security threats and vulnerabilities. In InfoSecCD ’06: Proceedings of the 3rd annual conference on Information security curriculum development, pages 68–78, New York, NY, USA. ACM Press.

[NIC.br 2007] NIC.br (2007). Registro de domínios para a internet no brasil. Núcleo de Informação e Coordenação do Ponto br - NIC.br. Comitê Gestor da Internet no Brasil - CGI.br, http://registro.br/.

[Nick 2004] Nick (2004). Copilot – a coprocessor-based kernel runtime integrity monitor. pages 179–194.

[NIST 2007] NIST (2007). National Institute Of Standards And Technology (NIST). U.S. Commerce Department’s Technology Administration, http://www.nist.gov/.

[NSRL 2007] NSRL (2007). National Software Reference Library (NSRL). National Institute of Standards and Technology (NIST). U.S. Department of Justice’s National Institute of Justice (NIJ), http://www.nsrl.nist.gov/.

[NWCCC and FBI 2006] NWCCC and FBI (2006). Internet crime report. Prepared by the National White Collar Crime Center and Federal Bureau of Investigation. The Internet Crime Complaint Center (IC3), http://www.ic3.gov/media/annualreport/2006_IC3Report.pdf.

[OpenGroup 2007] OpenGroup (2007). dd - convert and copy a file. The Open Group, http://www.opengroup.org/onlinepubs/009695399/utilities/dd.html.

[Palmer and Corporation 2001] Palmer, G. and Corporation, M. (2001). A road map for digital forensic research. Technical report.

[Pavlov 2007] Pavlov, I. (2007). LZMA SDK (Software Development Kit). http://www.7-zip.org/sdk.html.

[Payton 2006] Payton, A. M. (2006). A review of spyware campaigns and strategies to combat them. In InfoSecCD ’06: Proceedings of the 3rd annual conference on Information security curriculum development, pages 136–141, New York, NY, USA. ACM Press.

[Phrack 2007] Phrack (2007). Project loki. http://www.phrack.org/issues.html?issue=49&id=6#article.

[Ramachandran and Feamster 2006] Ramachandran, A. and Feamster, N. (2006). Understanding the network-level behavior of spammers. In Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM 2006), pages 291–302, New York, NY, USA. ACM Press.

[Richard and Roussev 2006] Richard, G. G. and Roussev, V. (2006). Next-generation digital forensics. Communications of the ACM, 49(2):76–80.

[SecurityFocus 2007] SecurityFocus (2007). Securityfocus website. Symantec Corporation, http://www.securityfocus.com/.

[Shukla and Nah 2005] Shukla, S. and Nah, F. F. (2005). Web browsing and spyware intrusion. Commun. ACM, 48(8):85–90.

[Sipior et al. 2005] Sipior, J. C., Ward, B. T., and Roselli, G. R. (2005). A united states perspective on the ethical and legal issues of spyware. In ICEC ’05: Proceedings of the 7th international conference on Electronic commerce, pages 738–743, New York, NY, USA. ACM Press.

[Skoudis and Zeltser 2003] Skoudis, E. and Zeltser, L. (2003). Malware: Fighting Malicious Code. Prentice Hall PTR.

[Solove and Rotenberg 2003] Solove, D. J. and Rotenberg, M. (2003). Information Privacy Law (Aspen Elective Series). Aspen Publishers.

[Sophos 2001] Sophos (2001). Glossary of terms: Companion virus. Sophos Plc., http://www.sophos.com/pressoffice/news/articles/2001/11/va_glossary.html#comp.

[Sophos 2007] Sophos (2007). Sophos - anti-virus and anti-spam software for businesses. http://www.sophos.com/.

[Subramanya and Lakshminarasimhan 2001] Subramanya, S. R. and Lakshminarasimhan, N. (2001). Computer viruses. IEEE Potentials Magazine, 20(4):16–19.

[Symantec 2007a] Symantec (2007a). Understanding Virus Behavior under Windows NT. Symantec AntiVirus Research Center, http://securityresponse.symantec.com/avcenter/reference/virus.behavior.under.win.nt.pdf.

[Symantec 2007b] Symantec (2007b). Php.pirus. Symantec Corporation, http://www.symantec.com/security_response/writeup.jsp?docid=2000-122009-2642-99.

[Teelink and Erbacher 2006] Teelink, S. and Erbacher, R. F. (2006). Improving the computer forensic analysis process through visualization. Commun. ACM, 49(2):71–75.

[Tham 2001] Tham, A. (2001). What is code red worm? As part of the Information Security Reading Room. SANS Institute, http://www.sans.org/reading_room/whitepapers/malicious/45.php.

[TST 2005] TST (2005). TST admite que empresa investigue e-mail de trabalho do empregado. Tribunal Superior do Trabalho, http://ext02.tst.gov.br/pls/no01/no_noticias.Exibe_Noticia?p_cod_noticia=5319&p_cod_area_noticia=ASCS.

[TST 2006] TST (2006). Processo E-ED-RR - 613/2000-013-10-00.7. Tribunal Superior do Trabalho, http://ext02.tst.gov.br/pls/ap01/ap_red100.resumo?num_int=29569&ano_int=2003&qtd_acesso=908889.

[Viotto 2007] Viotto, J. (2007). CSI Digital.

[Wang and Yu 2005] Wang, X. and Yu, H. (2005). How to break md5 and other hash functions. In Eurocrypt 2005, volume 3494, pages 19–35. Lecture Notes in Computer Science.

[Weiss 2005] Weiss, A. (2005). Spyware be gone! netWorker, 9(1):18–25.

[Zadjmool 2004] Zadjmool, R. (2004). Hidden threat: Alternate data streams. Articles :: Windows OS Security. Security Focus, http://www.windowsecurity.com/articles/Alternate_Data_Streams.html.

[Zhang and Paxson 2000] Zhang, Y. and Paxson, V. (2000). Detecting backdoors. In Proc. 9th USENIX Security Symposium, pages 157–170.

[Zou et al. 2005] Zou, C. C., Gong, W., Towsley, D., and Gao, L. (2005). The monitoring and early detection of internet worms. IEEE/ACM Trans. Netw., 13(5):961–974.