

JOSÉ GERALDO POPOLIN

ANÁLISE DE FERRAMENTAS PARA COMPUTAÇÃO
FORENSE EM SISTEMAS NTFS

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras, como parte das exigências do curso de Pós-Graduação Lato Sensu em Administração em Redes Linux para obtenção do título de Especialização.

Orientador

Prof. D.Sc. Joaquim Quinteiro Uchôa

LAVRAS
MINAS GERAIS – BRASIL

2011

JOSÉ GERALDO POPOLIN

ANÁLISE DE FERRAMENTAS PARA COMPUTAÇÃO
FORENSE EM SISTEMAS NTFS

Monografia apresentada ao Departamento de Ciência da
Computação da Universidade Federal de Lavras, como
parte das exigências do curso de Pós-Graduação Lato
Sensu em Administração em Redes Linux para obtenção
do título de Especialização.

Aprovada em ____ de _____ de ____

Prof. _____

Prof. _____

Prof. _____

(Orientador)

LAVRAS
MINAS GERAIS – BRASIL

2011

Dedico esta monografia à minha esposa Lucilene,
pela paciência e apoio no decorrer do trabalho.

Agradecimentos

Ao Professor Joaquim pela oportunidade e pela orientação para que este trabalho fosse concluído.

A todos os professores do curso ARL pela dedicação e orientação durante todo o curso.

SUMÁRIO

1 Introdução.....	1
1.1 Metodologia.....	2
1.2 Organização do trabalho.....	3
2 Forense Computacional.....	4
2.1 Identificação.....	5
2.2 Preservação.....	6
2.3 Análise	6
2.4 Apresentação.....	7
2.5 Comentários finais.....	7
3 Coleta de Evidências digitais.....	8
3.1 Coleta de provas com a RFC 3227.....	8
3.2 Ordem de Volatilidade dos dispositivos	10
3.2.1 Dispositivos de armazenagem da CPU.....	11
3.2.2 Memória de periféricos.....	11
3.2.3 Memória principal do sistema.....	11
3.2.4 Tráfego de rede.....	12
3.2.5 Estado do sistema operacional.....	12
3.2.6 Estado da rede.....	12
3.2.7 Sistema de arquivos	13
3.3 Considerações finais.....	14
4 Sistemas NTFS.....	15
4.1 Características do NTFS.....	16
4.2 Estrutura do NTFS.....	16
4.3 <i>MACTimes</i>	19
4.4 <i>Alternate Streams</i>	20
4.5 Considerações finais.....	21
5 Ferramentas para Forense em ambiente NTFS.....	22
5.1 Ferramentas para a coleta de evidências para	25
<i>Post Mortem Analysis</i>	25
5.1.1 <i>Dd, Dcfldd e Dc3dd</i>	25
5.1.2 <i>Guymager</i>	29
5.1.3 <i>AIR</i>	31

5.2 Ferramentas para análise de evidências	32
<i>Post Mortem Analysis</i>	32
5.2.1 <i>The Sleuth kit</i>	32
5.2.2 <i>Stegdetect</i>	34
5.2.3 <i>Foremost e Scalpel</i>	35
5.2.4 <i>Strings e Grep</i>	36
5.3 Ferramentas para a coleta de evidências em.....	37
<i>Live Forensic Analysis</i>	37
5.3.1 Ferramentas usadas para aquisição da imagem.....	38
de memória RAM.....	38
5.3.2 Ferramentas para análise da rede.....	42
5.3.3 Ferramentas para análise de arquivos.....	45
5.3.4 Ferramentas para análise de arquivos de <i>browser</i>	49
5.3.5 Ferramentas para recuperação de arquivos apagados....	52
5.4 Considerações finais	54
6 Aplicação das ferramentas em um incidente	55
6.1 Aplicação de ferramentas para <i>Live Forensic Analysis</i>	55
6.2 Aplicação de ferramentas para <i>Post Mortem Analysis</i>	58
6.3 Considerações finais.....	61
7 Conclusão	62
8 Referências Bibliográficas	63

ÍNDICE DE FIGURAS

Figura 1: Estrutura do MTF (NTFS.COM, 2010).....	17
Figura 2: CAINE- <i>Computer Aided INvestigative Environment</i>	23
Figura 3: Interface gráfica do <i>WinTaylor 2.1</i>	24
Figura 4: Aquisição de uma imagem de dispositivo com o <i>dd</i>	26
Figura 5: Aquisição de uma imagem de dispositivo com o <i>dcfldd</i>	27
Figura 6: Aquisição de uma imagem de dispositivo com o <i>dc3dd</i>	29
Figura 7: Aquisição de uma imagem pelo <i>guymager</i>	30
Figura 8: Aquisição de uma imagem de um dispositivo pelo AIR.	32
Figura 9: Análise de uma imagem pelo <i>Autopsy</i>	33
Figura 10: MDD adquirindo uma imagem da memória RAM.....	39
Figura 11: <i>Win32dd</i> adquirindo uma imagem da memória RAM.....	40
Figura 12: <i>Winen</i> adquirindo uma imagem da memória RAM.....	41
Figura 13: <i>CurrPorts</i> analisando a rede.....	44
Figura 14: <i>Advanced LAN Scanner</i> analisando a rede local.....	45
Figura 15: Auditoria realizada pelo <i>Winaudit</i>	47
Figura 16: <i>RootkitRevelear</i> analisando um sistema.....	49
Figura 17: <i>PasswordFox</i> revelando uma senha armazenada.....	51
Figura 18: <i>PhotoRec</i> recuperando arquivos.....	53
Figura 19: Aquisição da imagem com o <i>FTK Imager</i>	56
Figura 20: Recuperação de arquivo deletado com o <i>Recuva</i>	57
Figura 21: Ferramenta AIR adquirindo uma imagem do SO.....	58
Figura 22: Ferramentas <i>strings</i> e <i>greps</i> procurando por evidências.....	59
Figura 23: <i>Stegdetect</i> analisando a imagem recuperada.....	60

Resumo

O presente trabalho tem como objetivo mostrar como é relativamente fácil analisar um sistema operacional suspeito de invasão, usando ferramentas gratuitas que podem ser encontradas na internet, individualmente ou agrupadas em um *LiveCD*. Essas ferramentas são usadas para análise de sistemas operacionais, principalmente na área da forense computacional, mas devido a sua facilidade de uso, nada impede o seu uso pelo administrador de redes, ou até mesmo por usuários, para se proceder a análise de algum equipamento suspeito de violação. Neste trabalho foram realizados experimentos com ferramentas para a extração e análise de dados tendo como foco uma máquina com o sistema de arquivos NTFS instalado, mas essas ferramentas também podem ser usadas em sistemas de arquivos FAT32. Também há ferramentas que trabalham em Linux e podem ser usadas para analisar uma imagem obtida de um sistema FATx ou NTFS. Todos os testes foram realizados segundo os padrões estabelecidos pela RFC3227.

Palavras-Chave: Perícia Forense; *LiveCD*; NTFS; Linux; invasão.

Capítulo 1

1 Introdução

Atualmente, tanto o computador quanto a Internet são indispensáveis para o contínuo desenvolvimento da humanidade, e a convivência sem eles seria inimaginável.

Essa mudança de estilo de vida e comportamento gerou novos tipos de serviços, onde cada vez mais pessoas e instituições se tornaram totalmente dependentes dos computadores, a ponto de serem considerados itens essenciais da sociedade atual.

Essa mudança também ocasionou o surgimento de novos tipos de incidentes, explorando justamente a facilidade e fragilidade que esses novos hábitos trouxeram.

Para analisar e corrigir esses incidentes também surgiram ferramentas que ainda hoje são pouco conhecidas na maioria das organizações.

Existem no mercado muitas ferramentas proprietárias para a análise de incidentes, mas o custo para sua aquisição é um obstáculo para os profissionais de TI, pois existem os problemas legais com o pagamento das licenças para seu uso, o que torna o aperfeiçoamento nessa área oneroso.

Existem também, além das ferramentas proprietárias, as ferramentas *software* livre e ferramentas gratuitas que são disponibilizadas gratuitamente na internet. Elas podem vir isoladamente, ou agrupadas em *LiveCD*, com dezenas ou até centenas de ferramentas para os mais diferentes tipos de análises a custo zero para o profissional de TI.

Entende-se como *software* Livre, qualquer programa de computador independente do sistema operacional utilizado, que possa ser utilizado, reproduzido, ou redistribuído sem restrições.

Usualmente é anexado com a distribuição do *software* livre uma licença específica. Dependendo da licença, o *software* livre pode ser disponibilizado com o código fonte, sendo possível sua modificação e comercialização, mas na maioria das vezes como é o caso das ferramentas gratuitas, ele está disponível gratuitamente apenas para uso.

1.1 Metodologia

Neste trabalho é mostrado o uso de algumas dessas ferramentas e seus procedimentos básicos, para a realização de uma análise quando ocorrerem incidentes em um sistema operacional, com enfoque para o seu uso em um sistema de arquivos NTFS, mas a maioria delas podem ser usadas em outros sistemas de arquivos como FATx, ext2, ext3.

Os meios de pesquisa utilizados para a concretização deste trabalho foram baseados em livros existentes que abordam temas da área, documentação existente de trabalhos realizados em monografias, teses e dissertações em que o tema da perícia forense estava presente.

Os testes realizados neste trabalho com as ferramentas foram realizados, seguindo-se a metodologia estabelecida pela RFC 3227 (RFC3227, 2002).

Para o presente trabalho foi usada uma imagem de um sistema operacional *Windows XP- Service Pack 3*, onde ocorreu uma falha na rede.

Para a análise, a imagem foi transferida para uma estação forense e

montada em uma máquina virtual onde foram utilizadas algumas ferramentas que estão inseridas no CAINE- *Computer Aided INvestigative Environment* (www.caine-live.net). A máquina virtual utilizada foi o *VirtualBox* (www.virtualbox.org), instalada em um sistema operacional Linux *Mandriva 2010* (www.mandriva.com).

1.2 Organização do trabalho

Para a apresentação deste trabalho, o texto se encontra organizado como se segue: O Capítulo 2 faz uma abordagem geral dos fundamentos da Computação Forense, as etapas de investigação com a *identificação, preservação, análise e apresentação* dos dados. O Capítulo 3 trata da metodologia usada para a aquisição das evidências, o uso da RCF 3227, e a *ordem de volatilidade* para a aquisição de dados. O Capítulo 4 trata do surgimento do sistema NTFS, suas características técnicas e estrutura. O Capítulo 5 faz uma breve análise de algumas ferramentas forenses usadas para uma análise forense. O Capítulo 6 mostra a aplicação dessas ferramentas em um caso simulado. E por fim no Capítulo 7 são apresentadas as principais conclusões do trabalho e perspectiva de trabalhos futuros.

Capítulo 2

2 Forense Computacional

Um dos principais fundamentos da forense computacional é o da Teoria de Locard. Segundo Venema (VENEMA, 2007), esse princípio diz que qualquer um ou qualquer coisa que entra no local do crime leva consigo algo do local e deixa algum rastro quando sai.

No mundo virtual dos computadores, o Princípio da Teoria de Locard ainda é válido, pois qualquer violação em um sistema operacional deixa rastros. Tais rastros com maior ou menor dificuldade poderão ser identificados e seguidos. Nesses casos o processo de análise forense pode se tornar extremamente complexo e demorado, necessitando do desenvolvimento de novas tecnologias para a procura de evidências.

Essa procura de evidências é necessária para examinar e encontrar vestígios de alguma prova de invasão, alteração em arquivos, ou nos sistemas de computadores, fazendo uso de *software* específico tais como: analisadores de discos, analisadores de pacotes, ferramentas de clonagem, analisadores de *logs*, *scanners*.

Na maioria das vezes, esse *software* está presente em um sistema operacional montado especificamente para análise forense. Com esse aparato podem ser realizados diversos procedimentos como: análise de memórias que contenham instruções de endereços em hexadecimal e de interrupções de processos (*dumpers*), análise dos códigos dos programas (*debuggers*) ou *logs* de arquivos de sistemas, e as mais variadas partes que compõem um sistema operacional, seja ele Linux, Unix, ou Windows.

O perito de computação forense necessita de um conhecimento

especializado para a utilização de técnicas e métodos, para que o resultado obtido seja o mais próximo possível do que realmente aconteceu.

A correta utilização desses métodos científicos para: preservar, coletar, restaurar, identificar, documentar e apresentar as evidências digitais, determinam com exatidão se um sistema computacional sofreu algum tipo de violação ou não.

Os procedimentos adotados na coleta de dados devem ser formais, seguindo-se uma metodologia para um correto procedimento na obtenção das provas de acordo com as normas legais.

O caminho ideal para a coleta nessa etapa de investigação começaria pela obtenção e coleta dos dados, sua *Identificação*, sua *Preservação*, a *Análise* e *Apresentação* dos dados, mostrados nas seções seguintes.

2.1 Identificação

Dentre os vários fatores envolvidos no caso é necessário estabelecer com clareza quais são as conexões relevantes como as datas, nomes de pessoas, empresas, órgãos públicos, autarquias, instituições, etc, dentre as quais foi estabelecida a comunicação eletrônica.

Discos rígidos em computadores podem trazer sua origem após os processos de recuperação de dados.

Segundo Melo (MELO, 2009), a etapa de identificação consiste na análise pericial que visa organizar os artefatos encontrados, englobando todos os artefatos de identificação do processo, antes e depois do desligamento da máquina.

2.2 Preservação

Todas as evidências encontradas precisam obrigatoriamente ser legítimas, para terem sua posterior validade jurídica. Sendo assim, todo o processo relativo à obtenção e coleta das mesmas, seja no elemento físico ou lógico, deve seguir normas legais.

Parte-se sempre do princípio que a outra parte envolvida no caso poderá e deverá pedir a contraprova, sobre os mesmos elementos físicos. Consequentemente, o zelo do profissional na obtenção desses dados começa seguindo rigorosamente normas estabelecidas, para não haver uma possível invalidação da prova.

Um perito em Forense Computacional experiente terá que ter certeza de que uma evidência extraída deverá ser adequadamente manuseada e protegida, para se assegurar de que nenhuma evidência seja danificada, destruída ou mesmo comprometida pelos maus procedimentos usados na investigação, e que nenhum vírus ou código malicioso seja introduzido em um computador durante a análise forense.

2.3 Análise

Na análise dos dados, a separação do que realmente interessa será feita após o perito estudar todos os tipos de arquivos, ou partes de arquivos que foram deixados intactos, programas suspeitos, registros de *logs*, fotos, etc.

Essa fase é muito importante e deve ser realizada rigorosamente nos padrões estabelecidos, para não haver comprometimento dos arquivos

estudados, que poderão servir de prova legítima em algum processo, pois qualquer descuido pode provocar sua alteração, e conseqüentemente sua invalidação.

2.4 Apresentação

A apresentação consiste em adequar as provas obtidas na análise realizada, para seu enquadramento no padrão de leis existentes no local em que foi realizado.

Com isso ela poderá ser usada como uma prova em um processo no âmbito cível ou criminal. Mas é muito importante que a fase de análise tenha sido muito bem elaborada, pois qualquer indício que possa colocar em dúvida sua veracidade, pode tornar todo o procedimento realizado nulo.

2.5 Comentários finais

O objetivo desse capítulo foi mostrar que na coleta de uma prova ou de evidências para uma futura análise, o mais importante antes do uso de um *software* específico é realizar todo o procedimento na coleta dessas evidências, com extrema cautela e cuidado, para não alterar seu estado original, o que fatalmente tornariam as evidências inválidas para o uso como prova, ou poderia causar dúvida sobre sua veracidade.

Capítulo 3

3 Coleta de Evidências digitais

Evidência digital se refere a toda e qualquer informação digital capaz de determinar se realmente ocorreu um incidente. Uma característica da evidência digital é que ela pode ser armazenada em outro dispositivo sem nenhuma alteração, para posterior análise.

A captura de evidências em um sistema computacional, constitui-se de uma varredura minuciosa em busca por informações. Essas informações podem estar em arquivos, na memória, podem ter sido excluídas, criptografadas ou podem ser apenas parte de um arquivo ou ainda estarem em arquivos danificados.

O tempo de vida de uma evidência digital varia de acordo com o local onde ela está armazenada. Quanto maior a volatilidade de uma informação, mais difícil se torna sua extração e menos tempo há para sua captura.

Apesar do estágio atual das pesquisas no campo da forense computacional, ainda existe muita carência de metodologias para o manuseio dessas evidências. Tal carência pode ser explicada pelo fato de existirem inúmeras mídias e sistemas operacionais diferentes, além de não haver um padrão definido para a coleta das mesmas.

3.1 Coleta de provas com a RFC 3227

Segundo Toscano (TOSCANO, 2009), o objetivo da RFC3227 (RCF, 2002) é mostrar aos administradores de sistemas ou peritos forenses, qual a

forma mais segura para coletar e arquivar provas relevantes. Segundo a RCF 3227 (RCF, 2002), que norteia a sequência padrão na coleta de provas, um "incidente de segurança" é a violação de um sistema de segurança, onde a política de segurança é de alguma forma infringida.

Sua finalidade é fornecer orientações sobre como coletar e arquivar as provas relevantes, com a garantia de que essas provas serão uma cópia fiel do ocorrido e que poderão ser aceitas e usadas para a análise do caso, ou aceitas como provas no caso de uma ação judicial.

Os itens constantes nesse documento para a aquisição das provas seguem a seguinte sequência:

- Manusear as provas conforme a lei local determina, para não haver contestação quanto ao resultado final do trabalho.
- Sempre que possível, fazer uma cópia exata do sistema a ser analisado, ou o mais próximo disso.
- Fazer um relatório detalhado, datado e assinado, com todo o procedimento executado e os resultados alcançados.
- Observar a diferença entre o relógio do sistema e a *Universal Time Coordinated* (UTC). Para cada *timestamp* fornecido indicar se a UTC ou hora local é utilizada.
- Estar preparado para testemunhar em juízo em alguma fase processual, descrevendo todas as ações realizadas. Por isso é importante detalhar todos os procedimentos realizados.
- Minimizar alterações, isto é evitar atualizações de arquivos e até de diretórios.
- Remover conexões externas quando houver necessidade de alterações.
- Sempre coletar os dados, copiar e depois analisar.

- Ser metódico. Mesmo não havendo problemas, os procedimentos devem ser testados e seguidos para garantir a viabilidade em uma situação de crise. Se possível, os procedimentos devem ser automatizados por razões de rapidez e precisão.
- Para cada dispositivo, uma abordagem metódica deverá ser adotada, seguindo-se as orientações estabelecidas. No quesito velocidade, os dispositivos devem trabalhar em paralelo para recolher provas. Porém, em um sistema a coleta deve ser feita passo a passo.
- Em casos forenses, geralmente será feita uma imagem fiel do original (cópia *bit a bit*).
- Executar os procedimentos seguindo uma ordem de volatilidade, começando-se pelos dados mais voláteis aos mais duradouros.

Todos esses procedimentos descritos devem ser realizados com cautela, pois qualquer descuido nessa fase poderá colocar todo o resultado final sob suspeita. Se a máquina ainda estiver ativa, os procedimentos para as evidências mais voláteis devem ter prioridade na realização, seguindo-se a ordem de volatilidade.

3.2 Ordem de Volatilidade dos dispositivos

O fator mais importante na aquisição de evidências é saber por qual dispositivo se deve começar o trabalho para a obtenção das provas. Dependendo do dispositivo, o tempo em que o conteúdo de uma instrução permanece armazenado é que vai definir a prioridade de análise no caso de um incidente de segurança. Os procedimentos descritos nas subseções seguintes seguem a ordem de volatilidade.

3.2.1 Dispositivos de armazenagem da CPU

As informações contidas nos registradores da CPU são de mínima utilidade. A captura de informações tanto dos registradores quanto das memórias *caches* são impraticáveis, devido ao tempo de vida muito curto dos mesmos.

3.2.2 Memória de periféricos

Muitos dispositivos como *modems*, *paggers*, aparelhos de *fax* e impressoras, possuem memórias que podem ser acessadas e seus dados salvos. Nelas podem estar armazenadas informações que não mais residem no sistema analisado, como documentos e mensagens de texto ou números de *fax* e telefone.

A memória de vídeo também pode prover informações úteis no caso do invasor estar utilizando um console ou terminal gráfico, de modo que a tela corrente pode ser capturada e reproduzida.

3.2.3 Memória principal do sistema

Uma análise da memória principal de um sistema ativo pode revelar muitas evidências de fatos ocorridos. Esse tipo de memória chamada de RAM é uma memória do tipo volátil, pois quando se desliga a máquina, todos os dados presentes na mesma são perdidos.

Quando um programa é executado ou alguma outra operação é realizada, os dados que estão sendo manipulados ficam residentes nessa

memória, até que sejam salvos em algum arquivo ou eliminados.

Quando alguma falha acontece no sistema ainda ativo, essa memória pode ainda estar armazenando as informações responsáveis pelo ocorrido. Essas informações podem ser capturadas e posteriormente analisadas.

3.2.4 Tráfego de rede

O tráfego de rede entre o invasor e a máquina alvo pode ser reconstituído a partir dos datagramas capturados, estabelecendo-se uma sequência de eventos para comparação com outras evidências encontradas.

Para a captura do tráfego da rede existem vários programas conhecidos como *sniffers*, que além da captura dos dados, conseguem decodificar e mostrar o resultado em uma forma humana de entendimento.

3.2.5 Estado do sistema operacional

O estado em que foi encontrado o sistema operacional no momento do ocorrido, pode fornecer informações importantes quanto a um possível ataque.

Essas informações podem conter dados sobre processos que se encontravam em execução, dados sobre portas e conexões abertas, usuários *logados*, instruções em tabelas e *caches*, e outros que são perdidos ao se desligar a máquina.

3.2.6 Estado da rede

O estado da rede mostra informações valiosas sobre as conexões

estabelecidas. Podem-se obter informações sobre as portas com conexões que estão ativas, ou as portas que estão aguardando conexões.

Com essas informações pode ser possível também determinar se foi instalada ou ativada uma *backdoor*, ou se há alguma conexão em andamento que não foi autorizada.

Uma análise nas interfaces da rede pode revelar a existência de um *sniffer* no sistema, bem como uma tentativa de isolar a máquina mudando o endereço do IP das interfaces, ou ainda se ela está em modo promíscuo.

3.2.7 Sistema de arquivos

Os arquivos de configuração, arquivos de *logs* e os arquivos temporários guardam muitas informações valiosas sobre o sistema operacional. Os arquivos de configuração são os responsáveis pelas permissões de acesso ao sistema. Qualquer alteração que passar despercebida pode ocasionar sérios problemas de segurança para o sistema envolvido.

Quando de uma invasão, ou da instalação de executáveis suspeitos em uma máquina é bem provável que o diretório específico para os arquivos temporários tenha alguma cópia desses arquivos ou de algum executável, pois a maioria deles é armazenada nesse diretório.

Os arquivos de *logs* são muito importantes para se conhecer todo o histórico do sistema operacional, pois eles podem registrar tudo. A análise desses arquivos pode revelar o que realmente aconteceu com um determinado programa em algum momento.

Os arquivos de *logs* podem registrar se o programa foi executado indevidamente ou se algum outro programa suspeito estava em atividade, ou

até mesmo encontrar uma suposta falha que ficou registrada.

Alguns arquivos de sistemas podem ainda ser renomeados com nomes bem parecidos com o original e serem usados para esconder arquivos suspeitos.

Esses arquivos podem ter como conteúdo dados originários de uma espionagem industrial, ou um conteúdo pornográfico, como em um caso de pedofilia, o que torna bem difícil o trabalho de localização dos mesmos.

O conhecimento sobre a função dos arquivos de sistemas, sua localização e seu diretório são fatores importantes para elevar o nível de segurança do sistema operacional, devido às informações que são armazenadas nesse diretório.

3.3 Considerações finais

A coleta de evidências é a parte mais crítica de todo o processo em uma análise de um sistema operacional, independentemente se ela for usada para confirmar uma possível invasão, ou como uma prova judicial, ou detectar uma possível falha no sistema operacional analisado. A coleta de provas segundo a RCF 3227 (RCF, 2002) torna bem mais seguro o resultado final.

Capítulo 4

4 Sistemas NTFS

O sistema NTFS (*New Technology File System*) foi desenvolvido face aos problemas de segurança e as limitações que o sistema FAT apresentava, principalmente para o uso em servidores e aplicações críticas.

Ele foi desenvolvido para ser um sistema de arquivos flexível, adaptável, altamente seguro e confiável, sendo que seus conceitos funcionais foram herdados do sistema de arquivos HPFS (*High Performance File System*).

Segundo o artigo do Suporte da *Microsoft* (MICROSOFT, 2005), o HPFS é o sistema de arquivos utilizado pelo OS/2 da IBM, com recursos que se aproximam muito dos permitidos pelo NTFS, como nome de arquivos com até 254 caracteres incluindo espaços, partições de até 512 GB e unidades de alocação de 512 *bytes*.

O sistema de arquivos HPFS foi primeiro introduzido com OS/2 1.2, para permitir um acesso mais abrangente aos discos rígidos maiores que apareceram no mercado. Além disso, foi necessário para um novo sistema de arquivos estender o sistema de nomes, a organização, e a segurança, para a crescente demanda do mercado de servidores de rede.

O sistema de arquivos HPFS mantém a organização de diretórios do FAT, mas adiciona uma classificação automática de diretórios baseada nos nomes de arquivos. Os nomes de arquivos têm até 254 caracteres de dois *bytes*.

Esse sistema também permite que um arquivo seja composto de dados e

atributos especiais, que concedem uma flexibilidade elevada em função do suporte de outras convenções de nomenclatura e de segurança.

Além disso, a unidade de alocação é alterada dos *clusters* para os setores físicos (512 *bytes*), o que reduz a perda de espaço no disco. Embora muito eficiente, esse sistema de arquivos caiu em desuso juntamente com o OS/2, sendo suportado atualmente somente pelo Linux.

4.1 Características do NTFS

O NTFS possui algumas características importantes como:

- *Confiança*: que permite o sistema operacional se recuperar de problemas sem perder informações, fazendo-o ser tolerante a falhas.
- *Segurança*: onde é possível ter um controle de acesso preciso e ter aplicações que rodem em rede, fazendo com que seja possível o gerenciamento de usuários, incluindo suas permissões de acesso e escrita de dados.
- *Armazenamento*: onde é possível trabalhar com uma grande quantidade de dados, permitindo-se inclusive o uso de *arrays* RAID.
- *Rede*: fazendo do sistema plenamente funcional para o trabalho e o fluxo de dados em rede.

4.2 Estrutura do NTFS

O NTFS (*New Technology File System*), embora seja um sistema de arquivos nativo do *Windows* NT/2K, continua mantendo suporte ao sistema

FAT originário do DOS.

Esse sistema foi desenvolvido com o objetivo de suprir as necessidades do mercado corporativo, tais como: maior capacidade de endereçamento, suporte a critérios de segurança aplicáveis a cada arquivo individualmente, e cifragem de dados entre outros.

Segundo o artigo da www.NTFS.com (NTFS, 2010), a formatação de uma partição NTFS resulta na criação da *Master File Table* (MFT) e de diversos arquivos de sistema. A MFT contém informações sobre todos os arquivos e diretórios de uma partição NTFS.

Além da MFT, o processo de formatação ainda cria um conjunto de arquivos que contêm meta informações usadas para implementar a estrutura do sistema de arquivos.

A Figura 1 mostra a estrutura do MTF. Tais arquivos são mapeados nos primeiros registros da MFT, inclusive a própria.

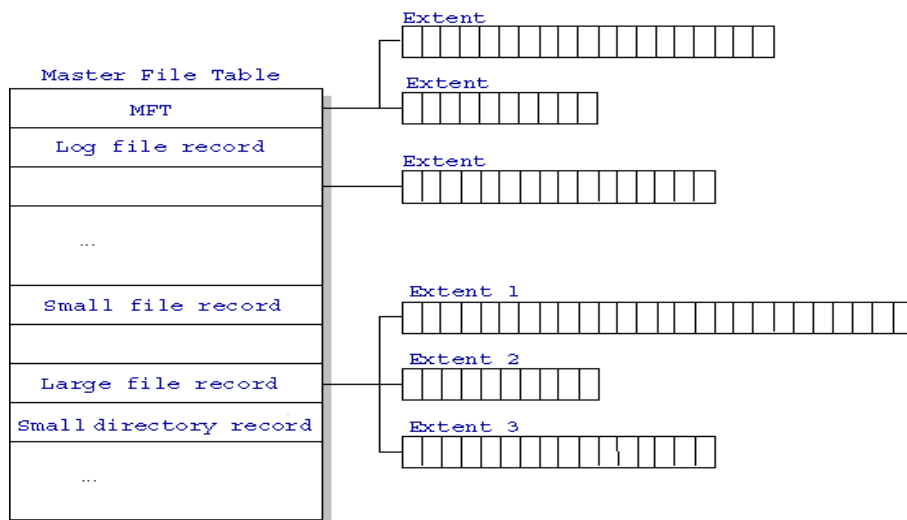


Figura 1: Estrutura do MTF (NTFS.COM, 2010).

Segundo o artigo da *www.NTFS.com* (NTFS, 2010), cada registro do MTF é composto por um pequeno cabeçalho que contém informações básicas descrevendo o próprio registro do MTF, tais informações são listadas abaixo:

- Números de sequência usados para verificação de integridade.
- Ponteiro para o primeiro atributo do registro.
- Ponteiro para o primeiro byte livre no registro.
- Número do registro em relação ao registro base da MFT, caso não seja o primeiro.

O cabeçalho inicial do MTF é seguido por um ou mais atributos que descrevem as características do arquivo. Cada atributo é dividido em dois componentes, sendo um cabeçalho que guarda o tipo do atributo, nome, *flags* e a localização da parte de dados do registro, e uma parte de dados onde é armazenada a informação do registro.

Cada arquivo em um volume NTFS é representado por um registro em um arquivo especial chamado tabela de arquivos mestre (MFT). O NTFS reserva os primeiros 16 registros da tabela de informações específicas.

O primeiro registro da tabela descreve a tabela de arquivos mestre em si seguido de um registro espelho MFT. Se o primeiro registro MFT estiver corrompido, o NTFS lê o segundo registro para localizar o arquivo espelho MFT, cujo primeiro registro é idêntico ao primeiro registro da MFT. Os locais dos segmentos de dados para o espelho MFT e MFT arquivo são registrados no setor de inicialização.

Para obter informações sobre o volume NTFS e o MTF há uma pequena ferramenta de linha de comando chamada *ntfsinfo* disponibilizada gratuitamente em (www.sysinternals.com).

4.3 *MACTimes*

Quando um arquivo é criado, acessado ou modificado, essa operação fica registrada. Esse registro é chamado de *MACTimes*. Segundo Oliveira (OLIVEIRA, 2001), uma vez compreendida a estrutura básica de um volume NTFS, pode-se então partir para uma análise mais detalhada sobre o *MACTimes* do ambiente Windows.

Um primeiro problema que aflige a grande maioria dos sistemas operacionais é a falta de um registro histórico, uma vez que os tempos registrados nos arquivos dizem respeito apenas a última modificação, o que torna impossível a obtenção dos acessos anteriores, utilizando-se apenas o sistema de arquivos.

Uma solução para esse problema poderia ser a habilitação de um *log* que registraria os acessos aos arquivos críticos do sistema, o que além de fornecer um histórico dos acessos, ainda forneceria outros dados como o usuário que fez o acesso.

Porém, tal solução poderia ter efeitos colaterais como o possível excesso de *logs*, além do fato de que seria necessário um estudo para se descobrir quais arquivos monitorar.

Outro problema seria que para efeitos de desempenho, o *LastAccessTime* tem resolução de uma hora, logo, todos os acessos a um arquivo em um intervalo menor de tempo aparentemente não seriam registrados.

A análise do *MACTimes* no Windows ainda reserva uma série de anomalias. Quando se copia um arquivo para um outro, de nome diferente, a data da última modificação continua igual a do arquivo original, enquanto as datas do último acesso e criação se comportam normalmente, dando a

impressão que o arquivo foi modificado antes de ser criado.

4.4 Alternate Streams

Segundo Oliveira (OLIVEIRA, 2001), as *alternate streams* são um mecanismo para embutir um arquivo dentro de outro, sem que seu conteúdo ou tamanho seja alterado.

Todo arquivo NTFS possui um outro arquivo sem nome embutido, o qual se chama *default stream* ou *unnamed stream*, onde os dados convencionais, como texto e programas são armazenados. Os arquivos embutidos criados com nomes diferentes são chamados *alternate streams*.

Essa funcionalidade foi desenvolvida para tornar o NT um servidor de arquivos para computadores que utilizassem o Mac OS.

O objetivo era simular o *resource forks* do HFS (*Hierarchical File System*), que é usado para armazenar dados como ícones e outros tipos de metainformação.

Um grande perigo dos arquivos que possuem as *alternate streams* é que existe a possibilidade de se ocultar nesses arquivos um programa malicioso ou um executável, sem que os mesmos sejam notados, pois mesmo alterando as *alternate streams*, não é modificada a assinatura digital do arquivo principal, e os arquivos que possuem *alternate streams*, só podem ser descobertos com programas específicos para isso.

Para obter informações sobre arquivos que contenham *streams*, há uma pequena ferramenta de linha de comando chamada *streams* disponibilizada gratuitamente em (www.sysinternals.com).

4.5 Considerações finais

O NTFS se tornou o sistema de arquivos padrão da plataforma Windows e vem evoluindo a cada nova versão, mas por ser um sistema proprietário, a sua documentação é escassa, o que torna seu entendimento pouco acessível, bem diferente de sistemas Linux, que por serem de código aberto e bem documentados, o seu entendimento se torna bem acessível.

O profissional que for realizar uma análise em um sistema de arquivos NTFS, deve estar sempre atualizado quanto as inovações e modificações desse sistema.

Isso é essencial para uma correta análise e avaliação de algum problema ocorrido. Também é importante saber os procedimentos legais vigentes, para a validação dos dados obtidos. As ações deverão ser feitas com a máxima cautela, para não haver alteração das evidências, que devem ser preservadas como foram encontradas.

Capítulo 5

5 Ferramentas para Forense em ambiente NTFS

Para a realização deste trabalho, foi usado o *LiveCD* CAINE-*Computer Aided INvestigative Environment* (www.caine-live.net). CAINE é uma distribuição GNU/Linux que oferece um ambiente forense com as ferramentas mais utilizadas para análise em sistemas Unix, Linux, e também em sistemas Windows.

Esse *LiveCD* foi desenvolvido para integrar ferramentas de *software* existentes que são acessados a partir de uma interface gráfica amigável.

Os principais objetivos do projeto do CAINE são o de garantir um ambiente interoperável para o perito digital durante as quatro fases da investigação digital, uma interface gráfica amigável, e uma compilação semi-automatizada do relatório final.

A Figura 2 mostra a interface principal do CAINE. Sua versão até a conclusão deste trabalho é a 2.0.

Ele foi remasterizado em uma distribuição UBUNTU 10.4, sendo que já existe repositório para a instalação dessas ferramentas em um sistema operacional com UBUNTU já instalado, acessível pelo *link*: (http://www.soluzioni.org/caine/howto/caine-from-deb_0.1.6a_i386.deb).



Figura 2: CAINE- *Computer Aided INvestigative Environment*

CAINE possui uma gama muito grande de ferramentas conhecidas e que são utilizadas no sistema Linux, mas também podem ser utilizadas para a análise de uma imagem adquirida de um sistema Windows (*Post Mortem Analisys*).

Para a análise de um sistema Windows que esteja ativo (*Live Forensic Analysis*) foi desenvolvida uma interface com um conjunto de ferramentas denominada *WinTaylor*. Até a versão 1.5 do CAINE, essa interface estava inclusa no *LiveCD*, mas ficou fora na versão 2.0. No link (<http://www.caine-live.net/Downloads/wintaylor2.1.zip>) é possível fazer o *download* da interface.

Na imagem do CAINE para *pendrive* ou para o uso em *netbook*

(<http://www.caine-live.net/Downloads/nbcaine2.0.dd.gz>) o *Wintaylor* está incluído. A Figura 3 mostra a interface do *WinTaylor*.

Para seu uso foi disponibilizada uma interface gráfica, onde existem *links* para rápido acesso às ferramentas mais utilizadas, e um *link* de acesso as demais ferramentas (*More tools*). Até a conclusão deste trabalho sua versão é a 2.1. A maioria das ferramentas inclusas no *WinTaylor* podem trabalhar tanto em sistemas FATx quanto em NTFS.

As ferramentas inclusas no *Wintaylor* podem ser acessadas a partir de um *pendrive*, ou de um CD, não sendo necessária sua instalação. As ferramentas disponíveis estão agrupadas em pastas, tendo algumas dessas ferramentas licenças GPL, outras tem licenças proprietárias, mas todas elas são gratuitas tanto para *download* como para uso.

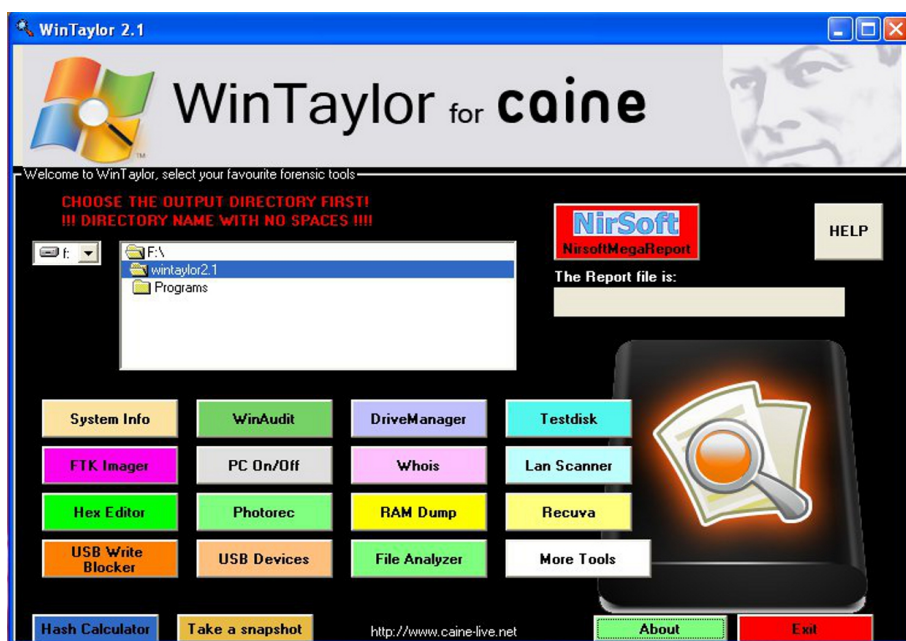


Figura 3: Interface gráfica do *WinTaylor 2.1*.

5.1 Ferramentas para a coleta de evidências para Post Mortem Analysis

A criação da imagem de um sistema a ser analisado e que já esteja desligado, seja por alguma falha do sistema operacional ou por algum acesso indevido, sempre que possível deverá ser o primeiro procedimento a ser realizado pelo profissional.

Existem diferentes maneiras para a criação de uma imagem. Pode-se retirar o HD, conectá-lo em uma estação forense onde será realizada a cópia, ou pode ser usado um *LiveCD* ou *pendrive* com programas para a obtenção da imagem e salvá-la em algum outro dispositivo, ou ainda a imagem pode ser transferida através da rede para uma estação forense.

O ideal é sempre trabalhar com uma cópia da imagem, pois para qualquer problema ocorrido durante os testes, sempre haverá a imagem original para dirimir qualquer dúvida.

Para a obtenção da imagem existem várias ferramentas, as mais frequentemente usadas e que estão disponíveis no CAINE são o *dd* e suas variantes, o *guymager* e o *AIR*.

5.1.1 *Dd, Dcfldd e Dc3dd*

Dd é uma ferramenta de linha de comando usada para aquisição de imagem, trabalha em Linux, tem versão para Windows (<http://www.chrysocome.net/dd>), sendo bem simples o seu uso e requer mínimos recursos para funcionar. Porém, não possui alguns recursos que são encontrados em ferramentas mais atuais para aquisição de imagens, como

recursos para coleta de metadados e recursos para correções de erros. A ferramenta *dd* gera arquivos de imagem tipo RAW, que podem ser lidos por muitos outros programas. A Figura 4 mostra a aquisição de uma imagem de dispositivo com o *dd*.

```
root@gerald-desktop:/home/caine# dd if=/dev/sdb1
of=imagem.img
15635532+0 records in
15635532+0 records out
8005392384 bytes (8.0 GB) copied, 1624.15 s, 4.9 MB/s
```

Figura 4: Aquisição de uma imagem de dispositivo com o *dd*.

Sua sintaxe básica é “*dd if=/dev/hda of=imagem.img bs=65536 conv=noerror,sync*”, onde “*if/dev/hda*” é o dispositivo que será copiado; “*of=imagem.img*” é a imagem que será criada; *bs* é tamanho do bloco em *bytes*, e caso não seja especificado será adotado o tamanho de bloco padrão de 512 *bytes*; *conv* são as opções de conversão a serem feitas. As páginas *man* ou *info* da ferramenta trazem inúmeras opções para seu uso.

A partir do *dd* surgiram alguns *forks* que foram aperfeiçoados inclusive para uso forense como o *dcfldd*, *sdd*, *dd_rescue*, *ddrescue*, *dccidd*.

Dcfldd (<http://dcfldd.sourceforge.net/>) é uma ferramenta aprimorada do *dd*, trabalha em Linux e foi desenvolvida por Nicholas Harbour quando trabalhava no *U.S. Department of Defense Computer Forensics Lab*. A sua última versão é a 1.3.4 datada de 12/02/2006.

Essa ferramenta possui algumas características úteis para os investigadores forenses, tais como:

- On-the-fly hashing- Calcula hashes dos dados de entrada enquanto

estão sendo copiados, garantindo a sua integridade.

- Mostra a quantidade de dados que já foram enviados, inclusive o tempo restante.
- Pode ser usada para zerar (*wipe*)¹ dispositivos de armazenagem.
- Verificação de que a imagem é idêntica a unidade original, *bit a bit*.
- A saída pode ser direcionada simultaneamente para mais de um arquivo/disco.
- A saída pode ser dividida em vários arquivos.
- *Logs* e dados podem ser canalizados para aplicações externas.

Dcfldd tem muitas opções para a aquisição de imagens. As páginas *man* ou *info* da ferramenta trazem inúmeras opções para seu uso. A Figura 5 mostra a ferramenta na aquisição da imagem de um dispositivo e calculando o *hash* da imagem gerada.

```
root@gerald-desktop:/home/caine# dcfldd if=/dev/sdb1
of=imagem hash=sha1
244224      blocks      (7632Mb)      written.Total      (sha1):
99ec83bafcab2a1a0a22790506330b97f7ef0cb0

244305+1 records in
244305+1 records out
```

Figura 5: Aquisição de uma imagem de dispositivo com o *dcfldd*.

1 - O termo zerar (*wipe*) tem o significado de regravar todo o disco rígido ou outro dispositivo de armazenagem com apenas zeros, apagando todos os dados existentes no mesmo. Isso pode ser realizado no *dcfldd* com o comando: *dcfldd if=/dev/zero of=/dev/nome_do_dispositivo*

Dc3dd (<http://dc3dd.sourceforge.net/>) é outra ferramenta usada para auxiliar em uma análise forense. Ela está licenciada sob a versão 3 da Licença Pública Geral (GPL).

Essa ferramenta e suas atualizações são mantidas pelo *DoD Cyber Crime Center* (<http://www.dc3.mil/>), que é uma agência do Departamento de Defesa do Governo dos Estados Unidos. Sua última versão é 7.0.0 de 19/08/2010. Entre suas características estão:

- Pode escrever um único valor hexadecimal ou uma sequência de texto para zerar (*wipe*) dispositivos de armazenagem.
- Suporta *hashes* MD5, SHA-1, SHA-256 e SHA-512.
- Agrupamento de erros. Substitui várias mensagens de erro iguais por apenas uma indicando o que aconteceu.
- Cria *logs* para *hashes* e erros.
- Tem a capacidade de dividir a saída em pedaços com extensões numéricas ou alfabéticas.
- Indicador de progresso. Indica o percentual real do trabalho sendo executado.
- Trabalha em Linux.

A Figura 6 mostra o *dc3dd* adquirindo uma imagem. Como o *dd* e o *dcfldd*, o *dc3dd* também tem várias opções para a aquisição da imagem.

As páginas *man* ou *info* da ferramenta trazem inúmeras opções para seu uso.

```
root@gerald:~/home/caine# dc3dd progress=on bs=512
if=/dev/sdb1 of=imagem
warning: sector size not probed, assuming 512
dc3dd 6.12.3 started at 2010-11-22 15:30:24 +0000
command line: dc3dd progress=on bs=512 if=/dev/sdb1
of=imagem
compiled options: DEFAULT BLOCKSIZE=32768
sector size: 512 (assumed)
2457882+0 sectors in
2457882+0 sectors out
1258435584 bytes (1.2G) copied (??%), 399,832 s, 3 M/s
dc3dd completed at 2010-11-22 15:37:04 +0000
```

Figura 6: Aquisição de uma imagem de dispositivo com o *dc3dd*.

5.1.2 Guymager

Guymager (www.guymager.sourceforge.net) é uma ferramenta geradora de imagens forenses, tendo como principais características:

- A interface de usuário é muito amigável e pode ser configurada para vários idiomas e a plataforma de trabalho é o Linux.
- Pode ser usada em máquinas com multi-processador.
- Pode gerar imagens flat (dd), EWF (E01) e imagens AFF. O formato EWF (Expert Witness, ou E01) é um formato proprietário para o armazenamento de imagens de disco usado no EnCase². O Advanced Forensics Format (AFF) é um formato aberto para o armazenamento de imagens de disco e metadados.

O seu uso e sua interpretação são bem simples. Na interface GUI, os

2- O *EnCase* é um *software* proprietário utilizado para perícia forense produzido pela *Guidance Software*. (<http://www.guidancesoftware.com>)

dispositivos de armazenamento que estão conectados aparecem em uma lista na parte superior da GUI.

Novos dispositivos podem ser conectados a qualquer momento. Ao se pressionar o botão Rescan, os novos dispositivos conectados serão mostrados na listagem. O dispositivo que está sendo adquirido aparecerá com uma referência na coluna *State* em azul.

Os dispositivos marcados na coluna *State* na cor vermelha são discos rígidos locais. Eles não podem ser adquiridos, evitando-se assim adquirir a imagem do disco errado.

O discos rígidos locais são reconhecidos por seus números de série que podem ser inseridos no arquivo de configuração.

A parte inferior mostra informações mais detalhadas sobre a aquisição do dispositivo selecionado pelo cursor. A Figura 7 mostra o *guymager* em funcionamento. Sua versão atual é *guymager-0.5.7beta1* (07/09/2010).

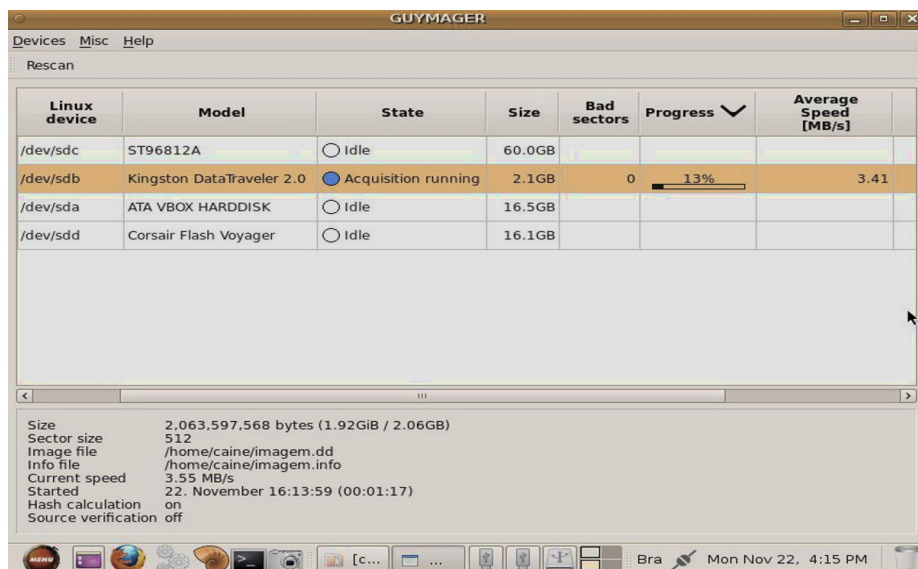


Figura 7: Aquisição de uma imagem pelo *guymager*.

5.1.3 AIR

AIR - *Automated Image and Restore* (<http://air-imager.sourceforge.net/>) é uma interface gráfica criada para gerar imagens forenses, trabalha em Linux, sendo bem simples e intuitivo seu uso no processo de aquisição de uma imagem. Entre as suas principais características estão:

- Detecta automaticamente unidades IDE e SCSI, CD-ROMs e *drives* de fitas.
- Cria imagens usando as ferramentas *dd* ou *dc3dd*.
- Verificação da imagem entre o original e a cópia usando *hash* MD5 ou SHA1/256/384/512.
- A compressão/descompressão da imagem é realizada pelas ferramentas *gzip/bzip2*.
- A imagem adquirida poder ser enviada através de uma rede TCP/IP via *netcat/cryptcat*.
- Suporta *drives* de fita SCSI.
- Zera (*wipe*) dispositivos de armazenagem ou partições.
- A imagem pode ser dividida em vários segmentos.
- Gera um *log* detalhado com data/tempo e todos comandos utilizados.

AIR seleciona automaticamente os dispositivos presentes na máquina analisada, como mostra a Figura 8. Todas as opções de configuração para a aquisição da imagem estão presentes na interface gráfica.

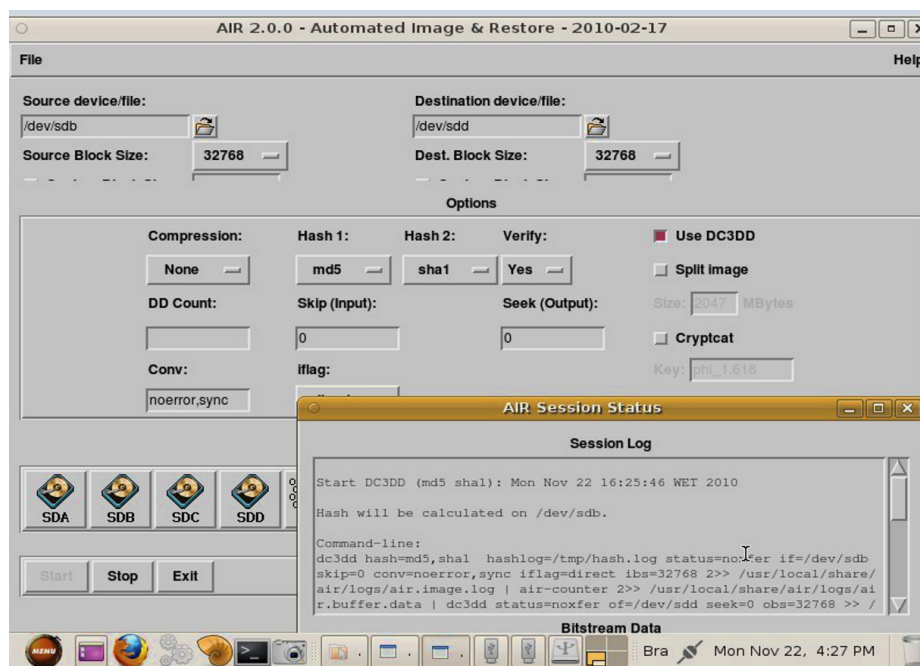


Figura 8: Aquisição de uma imagem de um dispositivo pelo AIR.

5.2 Ferramentas para análise de evidências Post Mortem Analysis

5.2.1 *The Sleuth kit*

The Sleuth Kit-TSK (<http://www.sleuthkit.org>) desenvolvido por Brian Carrier é um conjunto de ferramentas *open source* de linha de comando usadas para perícia digital forense. Sua versão em 28/10/2010 é a 3.2.0.

Essas ferramentas são utilizadas na análise de sistemas operacionais UNIX, Linux, OS X, FreeBSD, OpenBSD, Windows, sistemas de arquivos NTFS, FAT, UFS, ext2, ext3.

The Sleuth Kit-TSK executa a análise de um sistema de uma forma não

intrusiva, e integra muitos *LiveCDs* usados em perícia forense como o Helix, BackTrack, FCCU, FIRE, CAINE.

The Sleuth Kit-TSK é organizado em camadas, e cada uma delas possui ferramentas específicas para a análise de determinado setor da imagem.

A análise realizada na imagem apresenta dados relativos quanto ao volume e funcionalidade do sistema de arquivos, aos dados de sistemas de arquivos, aos dados das estruturas de nome de arquivo, aos metadados, às unidades de dados, ao *journal*, aos volumes de sistemas e também em arquivos de imagem nos formatos jpeg, gif e outros.

The Sleuth Kit normalmente é usado com uma interface gráfica em HTML chamada *Autopsy Forensic Browser*.

A Figura 9 mostra a análise de uma imagem realizada pela interface *Autopsy Forensic Browser*. Ela é uma ferramenta de código aberto e gratuito, também desenvolvida por Brian Carrier.

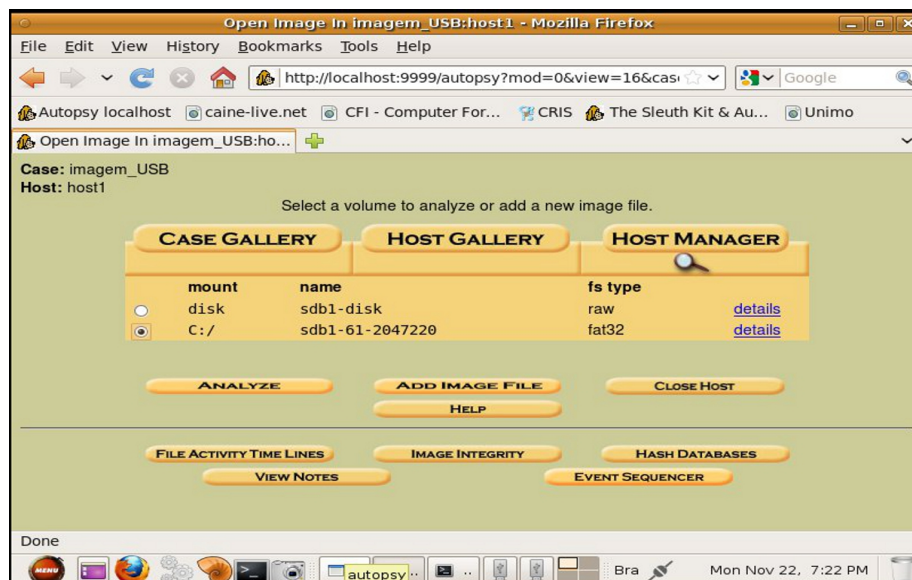


Figura 9: Análise de uma imagem pelo *Autopsy*.

O acesso à interface do *Autopsy* se dá por um navegador *Web*. *Autopsy* cria um servidor *Web* e seus *scripts* geram as páginas da interface.

Essa interface funciona como um gerenciador de arquivos, onde existem várias opções (*links*) que auxiliam na visualização de detalhes sobre dados apagados, arquivos, imagens, e estruturas do sistema de arquivos, facilitando a análise dos dados existentes.

Também é possível buscar por palavras-chave e expressões regulares nas imagens, bem como criar uma linha de tempo contendo os *mactimes* dos arquivos e diretórios.

Autopsy e *The Sleuth Kit* podem ser executados a partir de um *LiveCD* ou um *pendrive*. Esse procedimento é frequentemente utilizado durante a resposta a incidentes, enquanto o incidente está sendo confirmado. Depois que for confirmado, pode-se adquirir uma imagem do sistema para uma análise mais pormenorizada.

5.2.2 *Stegdetect*

Stegdetect é uma ferramenta *open source* de linha de comando usada para esteganografia, sendo utilizada para verificar se existem informações escondidas em imagens JPEG. Essa ferramenta trabalha em Linux e é capaz de detectar vários métodos diferentes de esteganografia usadas para inserir informações em imagens JPEG. Atualmente, os sistemas detectáveis são *jsteg*, *jphide* (*Unix and Windows*), *invisible secrets*, *outguess 01.3b*, *F5* (*header analysis*), *appendX* and *camouflage*.

Stegdetect (<http://www.outguess.org/>) funciona através de uma sintaxe simples que é “*#stegdetect /diretório/*.jpeg*”. Essa ferramenta faz uma

varredura no diretório selecionado e analisa os arquivos JPEG encontrados.

O *Xsteg* é uma interface gráfica para o *Stegdetect*. Sua última versão é 0.6(2004). Essa ferramenta foi desenvolvida por Niels Provos. (<http://www.citi.umich.edu/u/provos/>). As páginas *man* ou *info* da ferramenta trazem inúmeras opções para seu uso.

5.2.3 Foremost e Scalpel

Foremost (<http://foremost.sourceforge.net/>) é um programa de linha de comando usado para recuperar arquivos (*file carver*) com base em seus cabeçalhos, rodapés e estruturas internas de dados. Trabalha em Linux.

Ele pode trabalhar em arquivos de imagem, como aqueles gerados pelo *dd*, *SafeBack*, *Encase*, ou diretamente na unidade a ser analisada. Os cabeçalhos e rodapés podem ser especificados por um arquivo de configuração, ou pode ser usado uma linha de comando específica.

Foremost foi desenvolvido por Jesse Kornblum e Kris Kendall, quando trabalhavam pelo Escritório de Investigações Especiais da Força Aérea dos Estados Unidos e do Centro de Estudos de Segurança de Sistemas de Informação e Investigação, e posteriormente foi aberto ao uso público.

Foremost pode ser usado na recuperação de dados presentes em uma imagem. Os arquivos encontrados são salvos em pastas nomeadas com o nome da respectiva extensão do arquivo, no diretório de saída.

Sua última versão é *foremost-1.5.7.tar.gz*. A sintaxe básica usada no *foremost* é “**#foremost -i arquivo_de_entrada -o diretório_de_saída**”. As páginas *man* ou *info* da ferramenta trazem inúmeras opções para seu uso.

Scalpel (<http://www.digitalforensicssolutions.com/Scalpel/>) é um

programa de linha de comando para recuperação de arquivos (*file carver*).

Essa ferramenta lê as definições de um banco de dados de cabeçalho e rodapé, e extrai os arquivos de um conjunto de arquivos de imagem ou de imagens brutas.

Scalpel funciona independente do sistema de arquivos e pode recuperar arquivos em sistemas FATx, NTFS, ext2, ext3, ou partições brutas.

Pode ser útil tanto para a investigação forense digital, como para recuperação de arquivos apagados. *Scalpel* é resultado de uma reescrita completa do *Foremost 0.69*, e foi desenvolvida por Golden G. Richard III com a finalidade de melhorar o desempenho e diminuir o uso de memória.

Sua última versão é *Scalpel 1.60*. A sintaxe básica para o uso do *Scalpel* é “*#scalpel arquivo_de_entrada -o diretório_de_saída*”. As páginas *man* ou *info* da ferramenta trazem inúmeras opções para seu uso.

5.2.4 Strings e Grep

Essas ferramentas de linha de comando trabalham em Linux e são usadas para pesquisar o conteúdo da memória volátil de uma máquina. Segundo Santos (SANTOS, 2008) a ferramenta *strings* retorna o conteúdo texto do arquivo e a ferramenta *grep* faz uma varredura nos arquivos procurando pelo termo pesquisado e imprime as linhas do arquivo que contenham o termo. *Strings* tem uma versão para Windows (www.sysinternals.com) e a ferramenta *grep* para Windows pode ser encontrada em (<http://gnuwin32.sourceforge.net/packages/grep.htm>).

Elas são muito úteis na análise de uma imagem adquirida de uma memória RAM. Com o uso conjunto dessas ferramentas é possível encontrar

desde simples caracteres, até endereços URLs que estão na memória no momento da aquisição.

A sintaxe básica da ferramenta *strings* trabalhando em conjunto com a ferramenta *grep* é “*#strings -n imagem.img | grep <palavra>*”, sendo o *-n* o número mínimo de caracteres a serem pesquisados; *imagem.img* é o nome do arquivo de imagem a ser pesquisado; *<palavra>* é a palavra ou caractere a ser procurado. As páginas *man* ou *info* das ferramentas trazem inúmeras opções para seu uso.

5.3 Ferramentas para a coleta de evidências em *Live Forensic Analysis*

No caso do computador estar ativo, podem existir informações e dados voláteis que contenham elementos importantes para auxiliar na análise a ser realizada, como por exemplo, dados residentes na memória, conexões ou portas abertas, e processos em execução.

A coleta dos dados voláteis deve seguir a ordem de volatilidade, pois o tempo de vida de uma evidência pode ser diferente, dependendo de sua localização.

Enquanto a máquina estiver ativa é importante coletar o maior número de informações possíveis, pois ao ser deligada, os dados e as informações importantes do ocorrido presentes na memória RAM serão perdidos.

Se ocorrer uma falha ou uma possível invasão, o primeiro procedimento para uma posterior análise é fazer uma cópia (*dump*) da memória principal.

A memória RAM contém diversas informações voláteis do sistema, como por exemplo, dados sobre os processos que estão em execução, dados

que ainda estão sendo manipulados e não foram gravados no disco rígido.

Na análise da memória principal podemos identificar aplicações maliciosas como: *rootkits* injetados diretamente na memória, restos de *trojans*, endereços IP, conexões TCP/UDP, caracteres ASCII no qual possam conter indícios de *passwords*, comandos executados, etc.

5.3.1 Ferramentas usadas para aquisição da imagem de memória RAM

Existem vários programas para geração de imagem de uma memória RAM. As ferramentas MDD, *win32dd* e *winen* são ferramentas de linha de comando, utilizadas para aquisição de imagens da memória volátil e *FTKImager* é uma ferramenta com interface gráfica.

Essas quatro ferramentas foram incluídas no *LiveCD CAINE*, e podem ser acessadas através da interface gráfica *WinTaylor*.

Elas podem ser usadas para análises em sistemas NTFS, e também podem ser usadas para análises em sistemas de arquivos em que esteja instalado o sistema FAT32. A única restrição encontrada no uso das ferramentas é a necessidade da permissão de administrador para sua execução.

MDD (*MDD ManTech Physical Memory Dump Utility*) é uma ferramenta que gera uma imagem RAW e pode ser executada a partir de um drive USB ou um *LiveCD*. Essa ferramenta possui a licença GPL e é mantida gratuitamente por Mantech Security (<http://www.mantech.com>).

MDD é capaz de adquirir imagens da memória no Win2000, XP, Vista e Windows Server 2003 e Windows Server 2008.

Há versões para processadores de 32 bits, bem como para 64 bits. O seu

uso é bem simples. Para mais algumas informações sobre opções de uso, consultar o arquivo *readme* da ferramenta. A Figura 10 mostra sua execução e o resultado obtido a partir de uma janela do DOS.

O *MDD* é executado a partir do diretório onde está o executável. Sua sintaxe básica é “>*mdd -o <diretório armazenagem da imagem + nome do arquivo >*”.

```
D:\wintaylor2.1\Programs\ram\mdd>mdd -o D:\imagem_dd
-> mdd
-> ManTech Physical Memory Dump Utility
Copyright (C) 2008 ManTech Security & Mission Assurance
-> This program comes with ABSOLUTELY NO WARRANTY; for
details use option '-w'
This is free software, and you are welcome to redistribute
it under certain conditions; use option '-c' for details.
-> Dumping 1199.48 MB of physical memory to file
D:\imagem_dd'.
```

Figura 10: *MDD* adquirindo uma imagem da memória RAM.

Win32dd é uma ferramenta de linha de comando que pode ser executada a partir de um *drive* USB ou um *LiveCD*, e pode adquirir imagens de memória do Win2000, XP, Vista, Seven e Windows Server, e pode ser executada em sistemas de arquivos FAT32 e NTFS.

Imagens do tipo *Microsoft hibernation files* e *Microsoft memory crash dump files* (BSOD) também podem ser adquiridas. Esses tipos de arquivos são gerados pelo Windows quando ocorre algum tipo de evento ou falha.

Essa ferramenta é mantida por Matthieu Suiche (<http://msuiche.net>) e MoonSols (<http://moonsols.com>), e a versão disponibilizada gratuitamente é a *Community Edition*. Há versões para processadores de 32 bits e 64 bits.

A Figura 11 mostra o resultado obtido da execução do *win32dd* a partir

de uma janela do DOS. Sua sintaxe básica é “>*win32dd /f <diretório + nome do arquivo>*”, onde *diretório* é o local a ser armazenada a imagem e *nome do arquivo* é o nome da imagem. As várias opções para se uso podem ser acessadas, executando-se o comando *win32dd*.

```
D:\wintaylor2.1\Programs\ram\win32dd>win32dd /f D:\imagem
win32dd - 1.3.1.20100417 - (Community Edition)
Kernel land physical memory acquisition
Copyright (C) 2007 - 2010,Matthieu Suiche <http://www.msuiche.net>
Copyright (C) 2009 - 2010, MoonSols <http://www.moonsols.com>
Name                               Value
----                               -
File type:                          Raw memory dump file
Acquisition method:                 PFN Mapping
Content:                             Memory manager physical memory block
Destination path:                   D:\imagem
O.S. Version: Microsoft Windows XP Professional Service Pack3
(build 2600)
Computer name:                      GERALDO
Physical memory in use:              20%
Physical memory size:                1228272 Kb ( 1199 Mb)
Physical memory available:           981428 Kb ( 958 Mb)
Paging file size:                   2935180 Kb ( 2866 Mb)
Paging file available:               2823588 Kb ( 2757 Mb)
Virtual memory size:                 2097024 Kb ( 2047 Mb)
Virtual memory available:            2083412 Kb ( 2034 Mb)
Extented memory available:            0 Kb ( 0 Mb)
Physical page size:                  4096 bytes
Minimum physical address:             0x00000000000001000
Maximum physical address:             0x000000004AFEF000
Address space size:                   1258225664 bytes (1228736 Kb)
--> Are you sure you want to continue? [y/n] y
Aquisition started at: [29/03/2011 <DD/MM/YYYY> 1:14:46 <UTC>]
Processing...Done.
Aquisition finished at [29/03/2011 <DD/MM/YYYY> 1:21:50 <UTC>]
Time elapsed:                         7:04 minutes:seconds <424 secs>
```

Figura 11: Win32dd adquirindo uma imagem da memória RAM.

Winen é uma ferramenta de linha de comando que pode ser executada a partir de um dispositivo USB ou um *LiveCD*.

Sua sintaxe é bem simples, sendo necessário apenas executar o comando *winen*, e o programa irá interagir com o usuário e solicitar algumas informações para dar início ao processo da aquisição da imagem.

A imagem será salva no mesmo diretório em que se encontra o arquivo executável. Para mais algumas informações sobre opções de uso, consultar o arquivo *readme* da ferramenta. Funciona em sistemas FAT32 e NTFS.

A Figura 12 mostra o funcionamento do *winen*. Ele gera um arquivo tipo *EWF*. O formato *EWF* (*Expert Witness*, ou E01) é um formato para o armazenamento de imagens de disco usado no *Encase* (<http://www.guidancesoftware.com>), que é um *software* proprietário.

Para obter imagens tipo RAW ou *dd*, após ser adquirida a imagem com o *winen*, terá que ser feita uma conversão do formato. No *FTK Imager* existem ferramentas para isso.

```
D:\wintaylor2.1\Programs\ram\winen>winen

Please enter a value for the option "EvidencePath":

01

Please enter a value for the option "EvidenceName":

imagem

Please enter a value for the option "CaseNumber":

01

Please enter a value for the option "Examiner":

gerald

Please enter a value for the option "EvidenceNumber":

01
```

Figura 12: *Winen* adquirindo uma imagem da memória RAM.

FTK Imager é uma ferramenta com uma interface gráfica para aquisição de imagens de dispositivos de armazenagem, como disco rígido, pendrives, mídias removíveis, ou da memória RAM. Pode-se também adquirir dados ou arquivos específicos de uma imagem em análise e exportar o conteúdo para outro dispositivo ou diretório. Essa ferramenta funciona em sistemas FAT32 e NTFS.

Tem versões em linha de comando para *Linux (Fedora, HedHat e Debian)*. A ferramenta *FTK Imager* pode ser adquirida gratuitamente no site da Acessdata (<http://www.acessdata.com>).

5.3.2 Ferramentas para análise da rede

A partir do tráfego de rede é possível analisar toda a comunicação entre atacante e máquina invadida, estabelecendo-se uma sequência de eventos e comparando-as com as outras evidências encontradas.

A análise da interface de rede pode fornecer muitos detalhes do sistema para verificação sobre uma possível invasão, ou alguma conexão não autorizada sendo utilizada no sistema.

A análise das portas que estiverem abertas pode revelar valiosas informações quanto aos programas que estão trafegando por elas, se são autorizados ou não, ou se existe algo suspeito.

Segundo Melo (MELO, 2006), a visão de um administrador sobre as técnicas de varreduras usadas por script kiddies e crackers são importantíssimas para se conhecer o estado atual de segurança da sua rede.

Uma varredura completa em uma rede interna ou externa, ou ainda em determinada faixa de endereços, também mostrará inúmeras informações

sobre as condições gerais da mesma.

Pode-se saber quais as portas TCP/IP e UDP estão abertas e com os respectivos programas em uso, processos com o PID, nome do processo e seu *path*, etc.

Para complementar a coleta de dados, pode-se ter uma visão instantânea da rede, onde se pode observar as portas que estão sendo usadas, os respectivos protocolos e programas, ou ainda se existem programas desconhecidos ativos que possam estar comprometendo o sistema com conexões estranhas ou IPs suspeitos.

A Ferramenta *Currport* analisa as portas TCP/IP e UDP que estão abertas na máquina em análise, mostrando os processos em andamento e informações sobre os mesmos, como o nome, se é executável, seus desenvolvedores, endereço IP, data em que foi criado, além de mais alguns itens.

Essa ferramenta permite ainda fechar portas que estão abertas sem permissão, ou matar (*kill*) algum processo suspeito, realçando as portas abertas que estão sendo executadas e sem identificação.

O resultado obtido pode ser salvo, pois essa ferramenta possui a opção de gerar um relatório em HTML, XML ou texto. A Figura 13 mostra a ferramenta em operação.

Currport suporta os sistemas Windows NT, Windows 2000, Windows XP, Windows Server 2003, Windows Server 2008, Windows Vista, e Windows 7. Há versões para 32 e 64 bits e funciona em sistemas FAT32 e NTFS. Pode ser adquirida gratuitamente em (<http://www.nirsoft.net>).

The image shows a screenshot of the CurrPorts application window. The window title is 'CurrPorts' and it has a menu bar with 'File', 'Edit', 'View', 'Options', and 'Help'. Below the menu bar is a toolbar with various icons. The main area contains a table with the following columns: Process, P... (Port), Pro... (Protocol), Local... (Local Port), Local Po... (Local Port), Local ... (Local IP), Remote ... (Remote IP), Re... (Remote Port), State, and Process Path. The table lists several processes and their network connections.

Process	P...	Pro...	Local...	Local Po...	Local ...	Remote ...	Re...	State	Process Path
alg.exe	428	TCP	1030		127.0.0.1	0.0.0.0		Listening	C:\WINDOWS\System32\alg.exe
LEXPPS.EXE	1...	TCP	1025		0.0.0.0	0.0.0.0		Listening	C:\WINDOWS\system32\LEXPPS.EXE
lsass.exe	680	UDP	500	isakmp	0.0.0.0				C:\WINDOWS\system32\lsass.exe
lsass.exe	680	UDP	4500		0.0.0.0				C:\WINDOWS\system32\lsass.exe
mDNSResp...	1...	TCP	5354		127.0.0.1	0.0.0.0		Listening	C:\Arquivos de programas\Microsoft\Windows\mDNSResp...
mDNSResp...	1...	UDP	5353		10.0.2.15				C:\Arquivos de programas\Microsoft\Windows\mDNSResp...
mDNSResp...	1...	UDP	1026		0.0.0.0				C:\Arquivos de programas\Microsoft\Windows\mDNSResp...
svchost.exe	1...	TCP	1209		10.0.2.15	205.128....		Establish...	C:\WINDOWS\System32\svchost.exe
svchost.exe	940	TCP	135	epmap	0.0.0.0	0.0.0.0		Listening	C:\WINDOWS\System32\svchost.exe

Figura 13: CurrPorts analisando a rede.

Advanced LAN Scanner (<http://www.radmin.com>) é um scanner de rede muito rápido e altamente configurável para Windows, além de gratuito. Essa ferramenta pode escanear todas as 65.536 portas em menos de um minuto e funciona em sistemas FAT32 e NTFS.

Ao escanear as portas de uma rede interna, ou uma faixa de IPs para qual for configurada, pode extrair dados como os nomes dos usuários, endereço analisado, serviços, compartilhamentos e muitas outras informações úteis.

Seu funcionamento é simples, bastando para tanto, se conectar a uma máquina alvo ou uma faixa de IPs, digitando o IP ou a faixa de IPs desejados, como usuário ou se necessário, especificando um *login* e uma senha para isso.

A Figura 14 mostra o resultado obtido em uma operação de escaneamento de uma rede interna, e esse resultado pode ser salvo em um relatório gerado em arquivo texto. *Advanced LAN Scanner* tem uma interface bem amigável e fácil de ser usada.

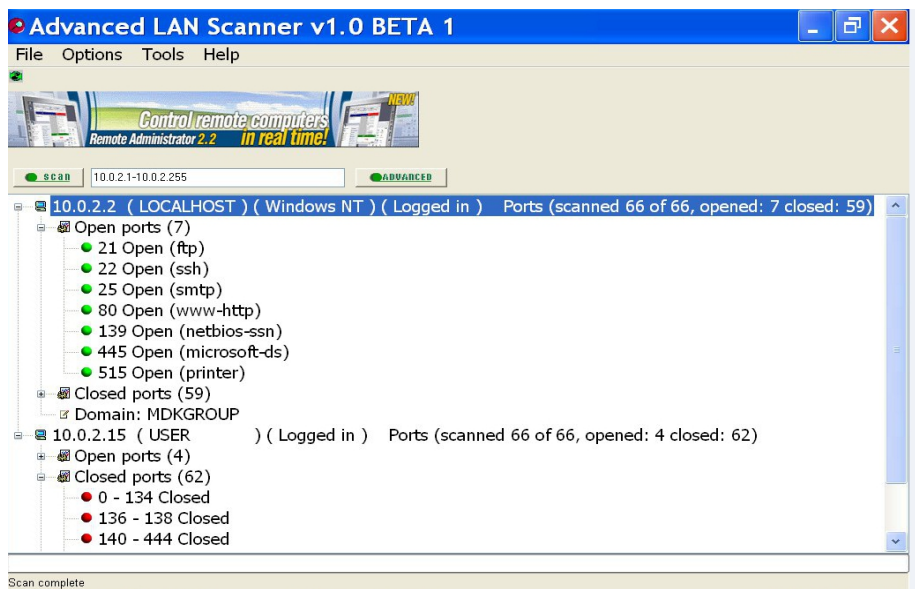


Figura 14: *Advanced LAN Scanner* analisando a rede local.

5.3.3 Ferramentas para análise de arquivos

Os serviços que estão ativos em uma determinada máquina também são fundamentais em uma análise. Programas maliciosos tendem a executar seus processos juntamente com os habituais, mudando o nome, ou com um nome bem parecido do original, com o intuito de dificultar sua descoberta. Um bom conhecimento de processos dos sistemas básicos que estão ativos na máquina é imprescindível em uma análise desse tipo.

Segundo Uchôa (UCHÔA, 2005) por mais cuidados que se tenha em uma invasão, ela deixa rastros. Do mesmo modo que não existe sistema totalmente seguro, não existe uma invasão perfeita. Verificar com uma certa frequência os arquivos de registros pode evitar surpresas extremamente desagradáveis.

Arquivos de *logs* registram, por exemplo, as atividades dos usuários, processos e conexões entre outros. Esses arquivos possuem um papel crucial na análise do sistema de arquivos, pois permitem a reconstituição de fatos que ocorreram no sistema.

Os arquivos de *logs* são uma das fontes mais importantes para a análise de ocorrências em qualquer sistema operacional. Qualquer modificação, instalação autorizada ou não de algum dispositivo, falhas de sistemas, conexões com outras máquinas, etc, criam algum *log* em algum arquivo específico para a ocorrência.

Para a obtenção desses dados, que podem ser usados na complementação da análise a ser realizada, há a ferramenta *WinAudit Freeware* (<http://www.pxserver.com/WinAudit.htm>) Sua atual versão é *WinAuditFreeware v 2.28.2*. Essa ferramenta é de propriedade de *Parmavex Services*, mas é disponibilizada gratuitamente. Ela funciona em sistemas de arquivos FAT32 e NTFS.

Essa ferramenta gera um relatório bem detalhado sobre o estado geral da máquina, contendo dados de *hardware*, *software*, programas ativos, programas instalados, portas abertas, *logs*, etc. Esse relatório pode ser gravado em vários formatos de documentos conhecidos. Sua interface é bem simples e pode ser configurada para ser visualizada em vários idiomas.

O programa é muito simples de se usar e não requer instalação. A Figura 15 mostra a sua operação, que requer apenas um clique para acionar um botão (Auditar), sendo que a análise completa da máquina é realizada rapidamente, e também podem ser selecionados os itens a serem auditados, através do menu (Opções). Essa ferramenta pode ser executada a partir de qualquer dispositivo removível como um disquete, unidade USB ou CD.

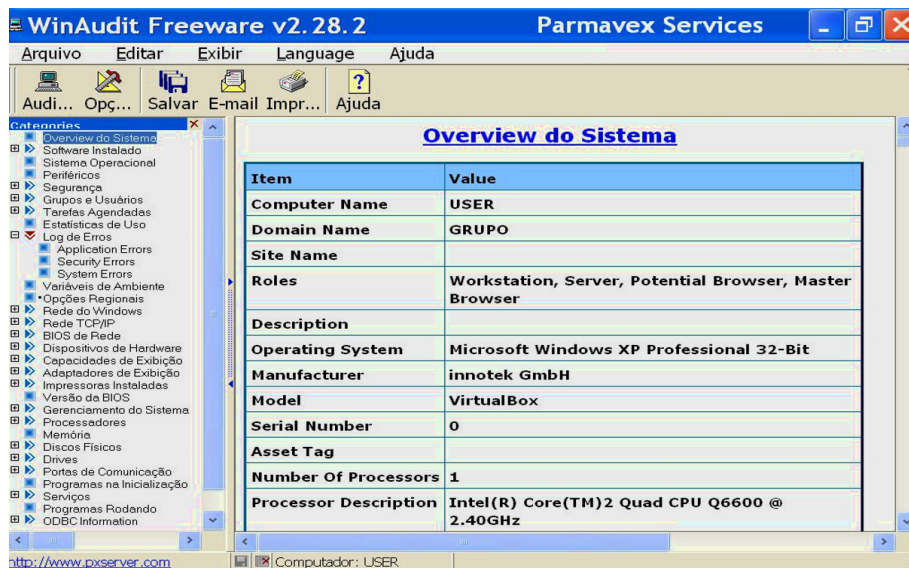


Figura 15: Auditoria realizada pelo Winaudit.

A verificação dos arquivos de configuração permite obter informações importantes para saber se houve alguma modificação no sistema. O invasor pode alterar esses arquivos para criar permissões especiais para acesso e manuseio do sistema, ou ocultar a sua presença.

Uma minuciosa análise desses arquivos pode revelar preciosas pistas para o esclarecimento de possíveis ataques ou tentativas de invasão do sistema, ou a presença de *rootkits*.

Segundo Teixeira (TEIXEIRA, 2005), os *rootkits* são um conjunto de ferramentas utilizadas para dar acesso total a um sistema. Como nos sistemas operacionais Unix e Unix Like o usuário *root* é quem tem acesso irrestrito ao sistema, o nome *rootkit* denomina uma ferramenta que dá acesso ao invasor com esses privilégios.

O termo *rootkit* é usado para descrever os mecanismos e técnicas em que o *malware*, incluindo *vírus*, *spyware* e *trojans*, tentam esconder sua

presença dos *anti-spywares*, *antivírus* e utilitários de gerenciamento do sistema.

Para o uso no sistema operacional Windows, a maioria das ferramentas existentes para a análise de arquivos infectados por *rootkits* são proprietárias, normalmente elas estão vinculadas a *anti-vírus*, com licenças que vão desde *freeware*, *trial* ou *shareware*.

Apesar disso a grande maioria é disponibilizada gratuitamente para os usuários. Neste trabalho foi utilizada a ferramenta *RootkitRevealer* que apesar de ser proprietária é disponibilizada gratuitamente em (www.sysinternals.com). Ela funciona em sistemas de arquivos FAT32 e NTFS.

RootkitRevealer é um utilitário para a detecção de *rootkits*. Ele suporta o Windows NT e versões superiores, mostrando na sua saída, uma lista de registros e discrepâncias no sistema API, que podem indicar a presença de algum tipo de *rootkit*.

Segundo o artigo de suporte da *Microsoft* (WINDOWS SYSINTERNAL, 2006), os *rootkits* persistentes funcionam alterando os resultados da API, para que se tenha uma visão do sistema usando APIs, diferentes da visão real de armazenamento.

RootkitRevealer compara os resultados de um sistema de digitalização do mais alto nível com o nível mais baixo.

A partir de uma listagem do diretório, por exemplo, o *rootkit* será visto pelo *RootkitRevealer* como uma discrepância entre as informações retornadas pela API do Windows e a que foi vista na verificação bruta de um volume FAT ou NTFS, das estruturas do sistema de arquivos.

A Figura 16 mostra o *RootkitRevealer* analisando e mostrando arquivos suspeitos.

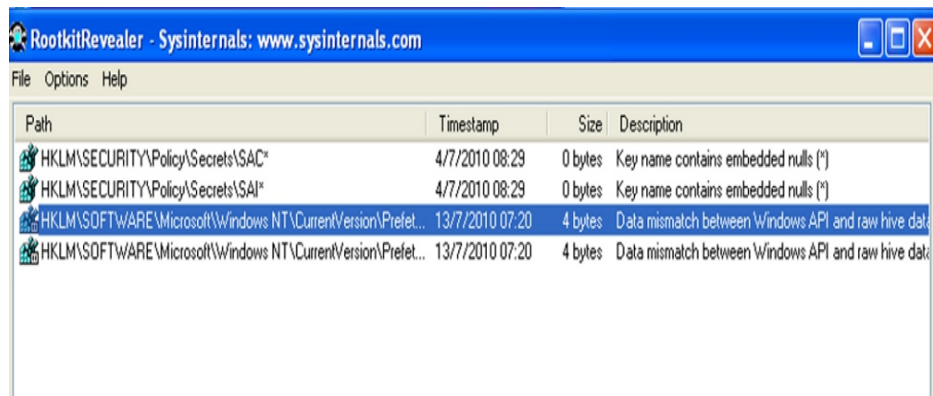


Figura 16: *RootkitRevealer* analisando um sistema.

5.3.4 Ferramentas para análise de arquivos de *browser*

Os arquivos dos *browsers* armazenam dados que podem ser de grande importância para análise de uma possível invasão, ou até para recuperação de dados.

Neles ficam registrados informações que podem revelar um histórico dos sites acessados, senhas gravadas, e *cookies* entre outros. Existem ferramentas simples e apropriadas para a extração desses dados e que podem fornecer uma infinidade de informações.

A ferramenta *IEHistoryView* (<http://www.nirsoft.net/>) lista todas as URLs visitadas que ficaram armazenadas no *Internet Explorer*. Pode-se também selecionar uma ou várias URLs, salvando-as em um arquivo texto HTML ou XML que poderão ser usados para a reconstrução da atividade *Web*. Essa ferramenta pode trabalhar em sistemas de arquivo FAT32 ou NTFS.

Esse utilitário lê todas as informações do arquivo *history* no computador, e exibe a lista de todas as URLs visitadas nos últimos dias. Ele

também permite selecionar um ou mais endereços URLs, removê-los do histórico do arquivo, ou salvá-los em texto HTML ou arquivo XML.

Além disso, essa ferramenta dá ao usuário permissão para ver a lista de URLs visitadas no perfil de usuário, e até mesmo acessar a lista de URLs visitadas em um computador remoto, desde que se tenha permissão para acessar a pasta *history*. *MozillaHistoryView* é a versão da ferramenta para o *Mozilla* (<http://www.nirsoft.net/>).

A ferramenta *IECookiesView* mostra todos os *cookies* que o *browser Internet Explorer* armazena em seu computador. Com essa ferramenta é possível traçar um roteiro de todos os sites visitados pelo usuário. *MozillaCookiesView* é a ferramenta similar para o *Mozilla*. Elas podem trabalhar em sistemas de arquivo FAT32 ou NTFS. Essas ferramentas podem ser encontrados em (<http://www.nirsoft.net/>).

A comodidade de deixar a senha de *e-mails*, *sites* e outros programas que necessitam de *login* para serem acessados, gravados no *Internet Explorer* ou outro navegador, pode ser uma porta aberta para o invasor ter acesso a esses dados.

Com ferramentas apropriadas, pode-se obter o *login* e senha de acesso do usuário.

As ferramentas *IepassView*, *ChromePass*, *PasswordFox* e *OperaPassView*, todas encontradas no link (<http://www.nirsoft.net/>), informam rapidamente se há alguma senha armazenada nos seus respectivos *browsers*, e o que pode ser visto como uma comodidade, pode se tornar um grande problema para o usuário no caso de uma invasão. Elas podem trabalhar em sistemas de arquivo FAT32 ou NTFS.

A Figura 17 mostra a ferramenta *PasswordFox* revelando uma senha armazenada no *browser Firefox*.

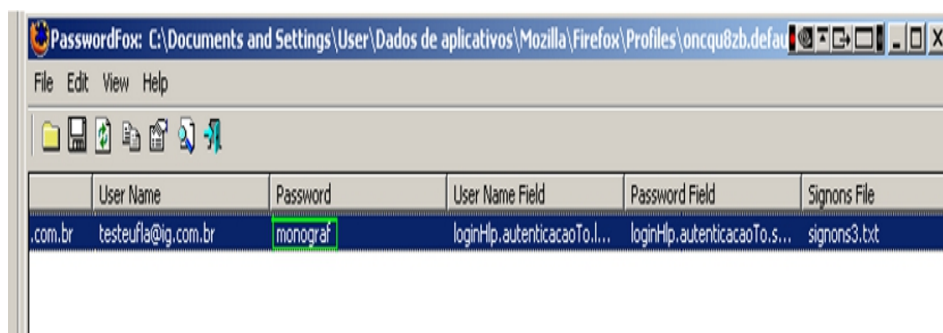


Figura 17: PasswordFox revelando uma senha armazenada.

Existem várias formas para a análise de um *e-mail*. O *e-mail* pode ser visualizado como um texto simples, contendo apenas a mensagem, ou como uma mensagem completa incluindo o cabeçalho com todos os dados. Isso pode ser muito útil para se descobrir a origem do mesmo.

Fazendo uma análise mais minuciosa do cabeçalho, pode-se identificar quem o enviou, o IP de quem enviou, o destinatário, dia e horário que foi enviado, etc.

Isso pode ser útil em uma investigação, pois se o *e-mail* é suspeito, pode-se cruzar informações, por exemplo, se no momento de uma invasão foi o mesmo *e-mail* no que foi aberto algum arquivo anexo.

IPNetInfo (<http://www.nirsoft.net/>) é uma ferramenta que permite que se encontre várias informações disponíveis sobre um endereço IP, como por exemplo, o dono do endereço IP, o país, nome do estado, intervalo de endereços IP, informações de contato (morada, telefone, *fax*, *e-mail*) e mais. Pode trabalhar com sistemas de arquivo FAT32 ou NTFS.

Esse utilitário pode ser muito útil para encontrar a origem de mensagens não solicitadas. Pode-se simplesmente copiar os cabeçalhos das

mensagens de texto do *e-mail* e colá-los em *IPNetInfo*.

Ele extrai automaticamente todos os endereços IPs presentes nos cabeçalhos das mensagens, e exibe as informações sobre esses endereços, que podem ser salvas em formato HTML.

Existem várias ferramentas para a análise de dados em aplicativos como o *Internet Explorer*, *Outlook Express*, *MSN*, etc, as quais podem revelar senhas arquivadas, *cookies*, URLs visitados, *e-mails*, conexão de *pendrives*, etc, dando ao investigador muitos dados para uma avaliação do que realmente pode estar acontecendo no sistema.

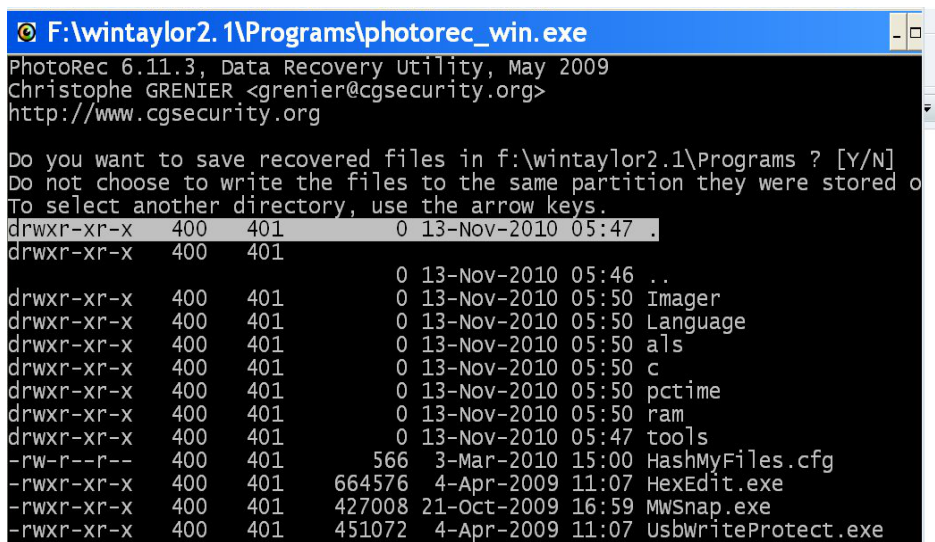
5.3.5 Ferramentas para recuperação de arquivos apagados

O Windows usa o *Master File Table* (MFT) como um índice para os arquivos que são armazenados no disco rígido. Quando um arquivo é apagado, torna-se mais prático para o Windows marcar sua entrada no MFT como excluído, do que apagar o arquivo em si, de modo que o arquivo é deixado no disco rígido. No processo de recuperação, quando é apagado um arquivo, o Windows não substitui a entrada MFT até que ele precise ser reutilizado.

PhotoRec (<http://www.cgsecurity.org>) é uma ferramenta de linha de comando, gratuita, utilizada para recuperar arquivos apagados. Essa ferramenta foi desenvolvida por Christophe Grenier e sua última versão é a 6.11.3, sendo bem simples o seu uso. Para seu funcionamento é necessário estar instalado o *CygWin* no Windows. Pode trabalhar com sistemas de arquivo FAT32 ou NTFS.

PhotoRec localiza os dispositivos de armazenagem presentes no

sistema, e após selecionar um dispositivo, apresenta os arquivos presentes no mesmo conforme mostra a Figura 18, e solicita o local para a armazenagem dos dados a serem salvos.



```
F:\wintaylor2.1\Programs\photorec_win.exe
PhotoRec 6.11.3, Data Recovery Utility, May 2009
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org

Do you want to save recovered files in f:\wintaylor2.1\Programs ? [Y/N]
Do not choose to write the files to the same partition they were stored on
To select another directory, use the arrow keys.
drwxr-xr-x  400  401      0 13-Nov-2010 05:47 .
drwxr-xr-x  400  401      0 13-Nov-2010 05:46 ..
drwxr-xr-x  400  401      0 13-Nov-2010 05:50 Imager
drwxr-xr-x  400  401      0 13-Nov-2010 05:50 Language
drwxr-xr-x  400  401      0 13-Nov-2010 05:50 als
drwxr-xr-x  400  401      0 13-Nov-2010 05:50 c
drwxr-xr-x  400  401      0 13-Nov-2010 05:50 pctime
drwxr-xr-x  400  401      0 13-Nov-2010 05:50 ram
drwxr-xr-x  400  401      0 13-Nov-2010 05:47 tools
-rw-r--r--  400  401      566  3-Mar-2010 15:00 HashMyFiles.cfg
-rwxr-xr-x  400  401  664576  4-Apr-2009 11:07 HexEdit.exe
-rwxr-xr-x  400  401  427008 21-Oct-2009 16:59 MWSnap.exe
-rwxr-xr-x  400  401  451072  4-Apr-2009 11:07 UsbwriteProtect.exe
```

Figura 18: *PhotoRec* recuperando arquivos.

A ferramenta *Recuva* (<http://www.piriform.com/recuva/download>) é um *software* para recuperação, que permite recuperar arquivos apagados acidentalmente ou propositadamente, arquivos enviados para a lixeira, ou arquivos deletados que estejam no disco rígido ou em outro dispositivo de armazenagem.

Com essa ferramenta, pode-se ter uma lista completa de arquivos que ainda estejam presentes no disco rígido, e que poderão ser úteis para a elucidação de algum incidente ocorrido.

Recuva tem uma interface gráfica bem amigável e o menu é interativo. Para seu funcionamento é necessário selecionar o tipo de arquivo objeto da pesquisa, e o diretório a ser pesquisado. Ela pode recuperar arquivos em

sistemas FAT32 ou NTFS.

Possui ainda algumas opções extras, que mostram os arquivos que não foram deletados, no caso de imagens jpg ou outra, pode-se ver a imagem (*preview*), pode-se ainda mostrar o cabeçalhos dos arquivos (*headers*). Para a recuperação é só selecionar o arquivo e escolher o local onde será copiado.

5.4 Considerações finais

Existe um grande número de ferramentas para coleta e análise de elementos de um sistema operacional, podendo essas ferramentas serem proprietárias, *open source*, *freeware*, etc. O importante é o administrador ou gerente de redes ter um conhecimento mais profundo sobre elas e sua utilização. Assim, no caso de surgir um imprevisto, ele pode conseguir elementos suficientes para descobrir o que ocasionou o incidente e obter a solução.

Capítulo 6

6 Aplicação das ferramentas em um incidente

Para demonstrar a utilização de algumas ferramentas foi feita uma simulação de um caso ocorrido, onde toda a extração das evidências foi feita segundo a RF3227.

Primeiramente, foi realizado a análise e extração das evidências com a máquina em funcionamento (*Live Forensic Analysis*). Posteriormente, os dados coletados foram analisados em uma estação forense montado com o *LiveCD CAINE* (*Post Mortem Analysis*).

O problema ocorreu em um equipamento onde estava instalado o sistema operacional Windows XP- *Service Pack3*, que começou a apresentar sinais de anormalidade na rede. A conexão ficava lenta, e após um certo tempo o equipamento era desconectado da rede, sendo preciso reiniciar o mesmo para a conexão voltar a funcionar, mas decorrido certo tempo de funcionamento o problema retornava, dificultando o uso desse equipamento, e causando transtornos.

6.1 Aplicação de ferramentas para *Live Forensic Analysis*

Ao chegar no local, foi constatado que o equipamento estava em funcionamento. O primeiro procedimento após proceder a abertura de um relatório com todos os dados do local e do equipamento a ser analisado, seguindo a ordem de volatilidade, foi a aquisição de uma imagem da memória RAM.

Foi adquirida uma imagem da memória RAM com a máquina funcionando perfeitamente, e após algum tempo, quando ocorreu a falha na conexão da rede,

foi adquirida nova imagem da memória RAM. A Figura 19 mostra a ferramenta *FTK Imager*, adquirindo a imagem da memória RAM.

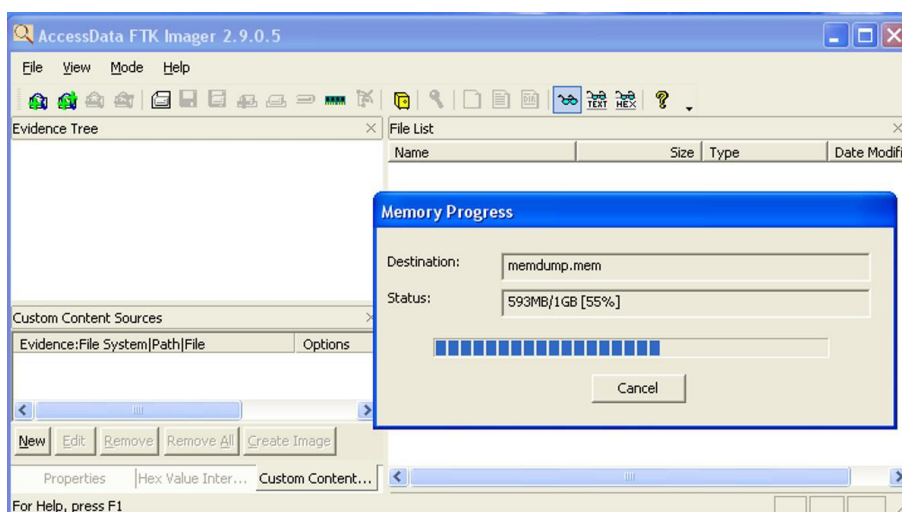


Figura 19: Aquisição da imagem com o *FTK Imager*.

Após, foi feita uma inspeção na rede para averiguar se havia algum possível programa desconhecido funcionando, o que não foi encontrado. Um rápido exame com o *CurrPort* não mostrou nenhum programa estranho rodando na máquina.

Em contato informal com os usuários, foi constatado que o problema começou após ter sido feito um *download* de um papel de parede. O mesmo foi aberto para verificação e foi enviado para a lixeira.

Arquivos que contenham imagens ou algo semelhante são muito usados para ocultar programas executáveis com a técnica de esteganografia. Esses arquivos podem conter *vírus*, *trojans*, ou algum cavalo de tróia, que irão se instalar automaticamente, quando forem abertos os arquivos hospedeiros.

Primeiramente, foi realizada uma análise para constatar a existência de possíveis *rootkits* na máquina. Com a ferramenta *Rootkit Revealer* foi verificado o registro do *Windows* a procura de alguma modificação ou falha, que poderia acusar a presença de algum *rootkit* no sistema, e nada foi encontrado de anormal.

Em seguida, foi realizada uma busca por arquivos deletados com a ferramenta *Recuva*, na tentativa de localizar o arquivo que poderia ter sido usado para inserir um programa malicioso no sistema.

A busca obteve um retorno positivo e foi localizado o arquivo *stego_img.jpg*, o qual foi alvo de uma análise mais pormenorizada com a ferramenta *stegdetect*. A Figura 20 mostra o *Recuva* no momento da localização do arquivo.

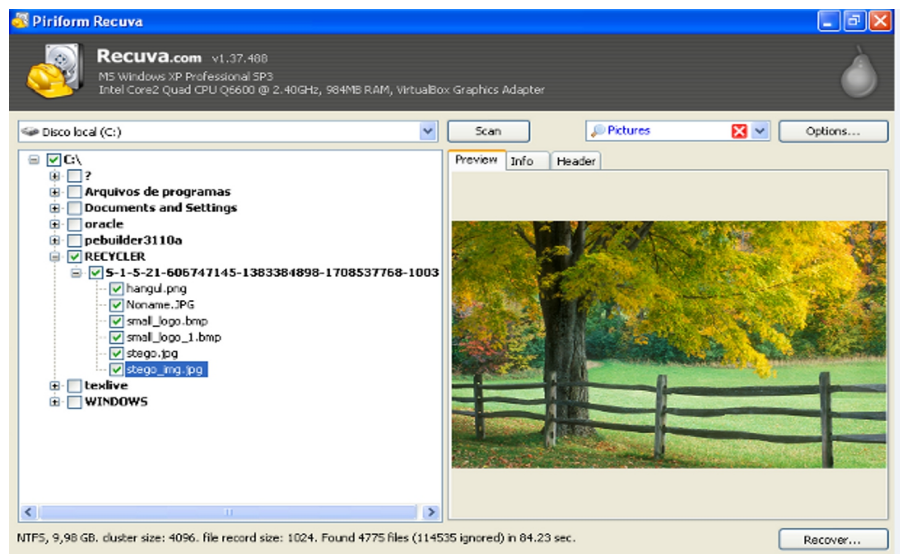


Figura 20: Recuperação de arquivo deletado com o *Recuva*.

Para completar a análise com a máquina ainda ativa foi usada a ferramenta *WinAudit*, que gerou um relatório pormenorizado de todo o sistema operacional.

6.2 Aplicação de ferramentas para *Post Mortem Analysis*

Após recolhidas as informações mais voláteis, a máquina foi desligada e foi dado *boot* pelo *LiveCD* CAINE, para a aquisição de uma imagem completa do sistema, como precaução para uma futura análise, se necessário. A Figura 21, mostra o AIR adquirindo a imagem.

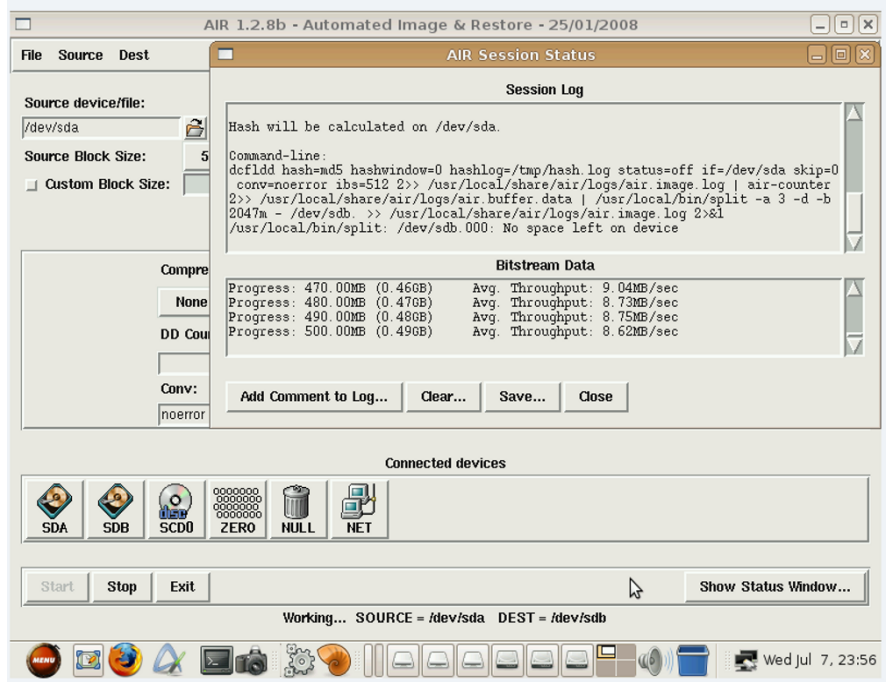


Figura 21: Ferramenta AIR adquirindo uma imagem do SO.

Para a *Post Mortem Analysis* foi criada uma estação de trabalho com o CAINE instalado em uma máquina virtual. Com essa estação foi possível analisar as imagens e arquivos coletado na primeira fase da perícia realizada no sistema.

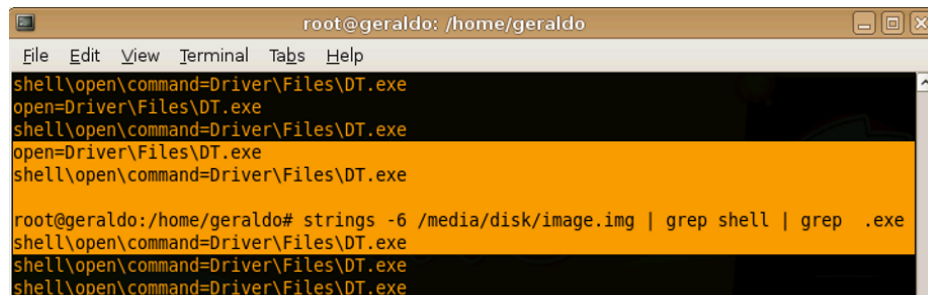
O primeiro procedimento foi fazer uma análise da imagem da memória RAM, adquirida após ter ocorrido o incidente, a procura de indícios da instalação de algum programa indevido.

Foi usada a ferramenta *strings* em conjunto com a ferramenta *grep* para a análise da imagem obtida da memória, a procura de caracteres que indicassem comandos digitados, programas iniciados, etc, e que pudessem ajudar na investigação.

Como a suspeita era de algum programa executável, foi feita uma procura por indícios, que poderiam trazer algum dado que ficou residente na memória. A sintaxe usada foi “*#strings -2 imagem.img | grep .exe*”.

A Figura 22 mostra o resultado, no qual há indícios de que realmente foi instalado algum executável.

As linhas *shell\open\command=Driver\Files\DT.exe*, *open=Driver\Files\DT.exe*, são linhas de comando encontradas em arquivos de extensão *.ini*, que são usados para execução automática de executáveis no Windows, trazendo também o nome do arquivo, que no caso é *DT.exe*.



```
root@gerald: /home/gerald
File Edit View Terminal Tabs Help
shell\open\command=Driver\Files\DT.exe
open=Driver\Files\DT.exe
shell\open\command=Driver\Files\DT.exe
open=Driver\Files\DT.exe
shell\open\command=Driver\Files\DT.exe
root@gerald:/home/gerald# strings -6 /media/disk/image.img | grep shell | grep .exe
shell\open\command=Driver\Files\DT.exe
shell\open\command=Driver\Files\DT.exe
shell\open\command=Driver\Files\DT.exe
```

Figura 22: Ferramentas *strings* e *greps* procurando por evidências.

A próxima ação foi usar o *Stegdetect* para analisar a imagem recuperada da lixeira. A Figura 23 mostra que a análise da imagem *jpg*, traz indícios de que foi usado o programa *jphide* para a ocultação de um arquivo.

No presente trabalho foi possível descobrir o nome do executável, nomeado como *DT.exe*, e não foi preciso executar o programa *Stegbreak* (www.outguess.org/download.php), para tentar descobrir a senha que ocultou o executável na imagem.

Quando um programa é ocultado em outro, pode-se usar uma senha para dificultar a sua extração no arquivo hospedeiro. O programa *Stegbreak* tenta por força bruta descobrir a senha e extrair o programa inserido.

A terminal window titled 'root@gerald: /home/gerald' with a menu bar containing 'File Edit View Terminal Tabs Help'. The terminal output shows the command 'stegdetect /media/GERALDO-8-1/Driver1/Files/image.jpg' being executed, resulting in the output '/media/GERALDO-8-1/Driver1/Files/image.jpg : jphide(*)'. The prompt 'root@gerald:/home/gerald#' is visible at the end of the line.

```
root@gerald: /home/gerald
File Edit View Terminal Tabs Help
root@gerald:/home/gerald# stegdetect /media/GERALDO-8-1/Driver1/Files/image.jpg
/media/GERALDO-8-1/Driver1/Files/image.jpg : jphide(*)
root@gerald:/home/gerald#
```

Figura 23: *Stegdetect* analisando a imagem recuperada

Com os dados obtidos nessa análise, foi realizada uma pesquisa sobre o executável descoberto. Esse executável alterava o funcionamento da rede, tornando-a lenta e derrubando a conexão. Pesquisando em sites especializados em segurança, foi descoberto que o executável além de alterar o seu nome, alterava o registro do Windows, e ocasionava todo o problema na rede. Feitas as retificações indicadas, o sistema voltou a funcionar normalmente.

6.3 Considerações finais

Esse caso descrito foi uma simulação adaptada de um caso real. A intenção foi demonstrar ao administrador de sistemas, ou ao usuário, a utilidade de ferramentas projetadas para uma investigação forense.

Elas foram usadas para analisar e descobrir alterações em um sistema operacional, que acontecem com frequência, decorrentes de situações comuns e que não podem ser evitadas.

Talvez a facilidade de interpretação dos resultados dessas ferramentas não seja tão simples, pois demanda experiência, mas o importante é se ter consciência de que a cada dia surgem mais ferramentas para esse fim, e suas qualidades também se aprimoram, e o mais importante é o custo, que atualmente é zero.

Capítulo 7

7 Conclusão

Com este trabalho procurei mostrar que existem muitas ferramentas gratuitas para auxiliar na análise forense de sistemas operacionais supostamente invadidos ou que apresentem anomalias. Nesse caso foi focado o seu uso em sistemas operacionais com sistema de arquivos NTFS.

Também, procurei mostrar como é fácil sua obtenção e a facilidade do uso dessas ferramentas gratuitas, pois além de serem usadas em uma investigação forense, nada impede que sejam usadas para a análise de uma suposta anomalia de uma máquina, ou de varias delas, evitando futuros aborrecimentos.

Há um grande número de projetos disponibilizados em *LiveCDs*, que podem ser obtidos sem custo algum. Pode-se ter em mãos, ferramentas gratuitas e de qualidade para análise de qualquer sistema operacional, nas mais diversas circunstâncias, e sendo essa análise realizada de acordo com padrões estabelecidos e reconhecidos internacionalmente, o resultado obtido será de alta qualidade.

Um trabalho futuro, incluindo a análise e os procedimentos para o uso dessas ferramentas, disponibilizando o resultado e a documentação desse trabalho para todos os que tem interesse nesses tipos de ferramentas, facilitaria e aumentaria a procura pelas mesmas, pois a compreensão de seu uso, ajudaria a tornar a administração ou gerência de uma rede mais segura.

8 Referências Bibliográficas

MELO Sandro- Computação Forense com Software Livre. Conceitos, técnicas, Ferramentas e Estudos de Casos. Página 16. Editora AltaBooks 2ª edição 2009.

MELO Sandro- Exploração de Vulnerabilidades em redes TCP/IP. Página 52. Editora AltaBooks 2ª edição 2006.

MICROSOFT- Visão geral dos sistemas de arquivos FAT, HPFS e NTFS (2005). Disponível na Internet via [www. url: http://support.microsoft.com/kb/100108/pt-br](http://support.microsoft.com/kb/100108/pt-br). Acessado em 20/10/2010.

MICROSOFT- WINDOWS SYSINTERNAL- (2006). Disponível na Internet via [www. url: http://technet.microsoft.com/en-us/sysinternals/bb897445](http://technet.microsoft.com/en-us/sysinternals/bb897445). Acessado em 20/10/2010.

NTFS.COM, (2010)- NTFS Master File Table (MFT)- Disponível na internet via [www. url: http://www.ntfs.com/ntfs-mft.htm](http://www.ntfs.com/ntfs-mft.htm). Acessado em 20/10/2010.

OLIVEIRA Flávio de Souza- Metodologias de Análise Forense para Ambientes Baseados em NTFS- 2001. Disponível na internet via [www. url: http://www.las.ic.unicamp.br/paulo/papers/2001-SSI-flavio.oliveira-forense.ntfs.pdf](http://www.las.ic.unicamp.br/paulo/papers/2001-SSI-flavio.oliveira-forense.ntfs.pdf). Acessado em 20/10/2010.

RFC3227 - Guidelines for Evidence Collection and Archiving- Network Working Group D. Brezinski Category: Best Current Practice neart.org February 2002 Disponível na internet via www. url: <http://www.faqs.org/rfcs/rfc3227.html>. Acessado em 20/10/2010.

SANTOS Laudenino Azeredo dos- Computação Forense em Sistemas GNU/Linux. Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras- Conclusão do Curso de Pós- Graduação em Administração de Redes Linux- 2008.

SWGDE- *Scientific Working Group on Digital Evidence*- Disponível na Internet em www. url: <http://www.swgde.org>. Acessado em 20/10/2010.

TEIXEIRA Ataliba de Oliveira. Uma Visão Forense dos Rootkits em Sistemas Linux. Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras- Conclusão do Curso de Pós- Graduação em Administração de Redes Linux- 2005.

TOSCANO Prof.Wagner- Auditoria Forense Computacional. Norma RFC3227. Coleta e arquivamento de provas– Prof. Wagner Toscano PMR/POLI/USP. São Paulo, Brasil. Disponível na internet via www.url: <http://wagnertoscano.eti.br/Pool/%5BAUF%5DNormaRFC.pdf>. Acessado em 20/10/2010.

UCHÔA Joaquim Quinteiro- Textos Acadêmicos- Segurança Computacional. Página 44. Curso de Pós- Graduação “*Latu Sensu*” em Administração de Redes Linux- Universidade Federal de Lavras- Ufla- FAEPE. 2ª edição 2005.

VENEMA, D. F. W. Perícia Forense Computacional – Teoria e Prática Aplicada. [S.l.]: Pearson Prentice Hall, 2007.