LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO RIO GRANDE DO NORTE
CURSO BACHARELADO EM SISTEMAS DE INFORMAÇÃO

LUCIANO ARAÚJO DE MEDEIROS

COMPUTAÇÃO FORENSE EM SISTEMAS GNU/LINUX UTILIZANDO O UBUNTU

NATAL/RN
2011

LUCIANO ARAÚJO DE MEDEIROS

COMPUTAÇÃO FORENSE EM SISTEMAS GNU/LINUX UTILIZANDO O UBUNTU

Monografia apresentada ao curso Bacharelado em Sistemas de Informação da Faculdade Natalense para o desenvolvimento do Rio Grande do Norte, como requisito obrigatório para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Profa. Msc. Joseane Alves Pinheiro.

NATAL/RN
2011

Catalogação na Publicação – Biblioteca da FARN
Setor de Processos Técnicos

Medeiros, Luciano Araújo de
Computação Forense em Sistemas GNU/Linux utilizando o UBUNTU / Luciano Araújo de Medeiros. – Natal, 2011.
65 f.

Orientador: Joseane Alves Pinheiro. (Msc.)
Monografia (Bacharel em Sistemas de Informação) – Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte.

1. Sistemas de Informação - Monografia. 2. Computação Forense – Monografia. 3. Segurança de Redes de Computadores – Monografia.
I. Título.

RN/FARN/BC CDU 004.49

LUCIANO ARAÚJO DE MEDEIROS

COMPUTAÇÃO FORENSE EM SISTEMAS GNU/LINUX UTILIZANDO O UBUNTU

Monografia apresentada ao curso Bacharelado em Sistemas de Informação da Faculdade Natalense para o desenvolvimento do Rio Grande do Norte, como requisito obrigatório para obtenção do grau de Bacharel em Sistemas de Informação.

Aprovado Em: ___/___/____

BANCA EXAMINADORA

Joseane Alves Pinheiro

Joseane Alves Pinheiro, Msc.
ORIENTADORA

Aluízio F. Rocha

Aluízio Ferreira da Rocha Neto, Msc.
MEMBRO DA BANCA

Rômulo Fagundes Cantanhede, Esp.
MEMBRO DA BANCA

Dedico esta monografia aos meus pais, que com todas as dificuldades encontradas na vida conseguiram me educar, mostrando-me sempre o melhor caminho a ser percorrido; e a FARN, pelas oportunidades oferecidas; Essa vitória não é minha, é de vocês, que sempre acreditaram em mim.

AGRADECIMENTOS

Primeiramente a Deus, pela saúde e proteção.

Aos familiares e amigos, por me apoiarem e me incentivarem, especialmente ao meu padrinho Milton e ao amigo Vicente Pascaretta Júnior, sem vocês nada disso seria possível.

A Professora Joseane pela oportunidade e pela orientação para que este trabalho fosse concluído.

Aos demais professores pela dedicação e contribuição com o meu conhecimento.

E a todos que, direta ou indiretamente, contribuíram para a realização deste trabalho.

"As pessoas que são loucas o suficiente para achar que podem mudar o mundo são aquelas que o mudam."

Steve Jobs

RESUMO

Este trabalho faz uma breve explanação acerca dos problemas e soluções atualmente existentes nas empresas relacionados à segurança de sistemas computacionais, destacando-se a importância da Forense Computacional no combate aos crimes eletrônicos. São apresentadas as principais vulnerabilidades e ameaças as quais estão sujeitas, procedimentos e formas de minimizá-las. Os principais tipos de atacantes, técnicas e ferramentas para combatê-los. Essa pesquisa também destaca a importância do profissional estar apto a resolver problemas de Incidentes de Segurança utilizando tais técnicas e ferramentas. Por este motivo foi desenvolvido uma prática com as ferramentas comumente utilizadas na computação forense em sistemas GNU/Linux utilizando o Sistema Operacional Ubuntu, as quais foram aplicadas numa perícia em um ambiente simulado com ênfase nas Forenses In Vivo e Post Mortem, a fim de gerar conhecimento para os profissionais de TI que estejam iniciando ou que já possuam algum conhecimento em computação forense, e demais profissionais que desejem agregar valor aos seus conhecimentos em segurança computacional.

Palavras-chave: Crimes Eletrônicos. Ferramentas de Analise Pericial. Forense Computacional. Software Livre. Ubuntu.

ABSTRACT

This paper is a brief explanation about the problems and solutions that currently exist in companies related to the security of computer systems, highlighting the importance of Computer Forensics in the fight against electronic crimes. Presents the main vulnerabilities and threats which are subject, procedures and ways to minimize them. The main types of attackers, techniques and tools to fight them. This research also highlights the importance of the professional being able to solve problems of Security Incidents using these techniques and tools. For this reason it was developed a practice with the tools commonly used in forensic computing systems GNU/Linux using the Operating System Ubuntu, which were applied in a skill in a simulated environment with emphasis on Forensic In Vivo and Post Mortem, in order to generate knowledge for IT professionals who are starting or already have some knowledge of computer forensics, and other professionals wishing to add value to their expertise in computer security.

Keywords: Electronic Crimes. Tools of Pericial Analysis. Forensic computer. Free Software. Ubuntu.

LISTA DE ILUSTRAÇÕES

Figura 1 - Atividades operacionais da Computação Forense 15
Figura 2 - Ciclo da Perícia Forense 16
Figura 3 - Nível de conhecimento demandado por um invasor 21
Figura 4 - *Modus Operandi* - visão macro 26
Figura 5 - Ações do invasor após a intrusão do sistema 29
Figura 6 - Processo de Resposta a Incidente 30
Figura 7 - Perícia Forense inserida no Processo de Resposta a Incidente 31
Figura 8 - Volatilidade das evidências 33
Figura 9 - Lendo pacotes com a ferramenta tcpdump 40
Figura 10 - Obtendo informações do *Pen-Drive* com o comando dmesg 40
Figura 11 - Obtendo informações da versão do *kernel* 40
Figura 12 – Gerando o *hash* dos dados capturados 46
Figura 13 - Desligando a máquina periciada no VirtualBox 47
Figura 14 - Abrindo o programa criador de discos de inicialização no Ubuntu 48
Figura 15 - Instalando o Ubuntu no *Pen-Drive* 48
Figura 16 - Configurando a BIOS para dar *boot* pelo *Pen-Drive* 49
Figura 17 - Tela inicial com as opções do Ubuntu 49
Figura 18 - Interface do sistema operacional Ubuntu 50
Figura 19 - Obtendo informações do HD Externo e da máquina invadida 50
Figura 20 - Usando o comando sfdisk para exibir a estrutura do HD periciado 51
Figura 21 - Gerando o *hash* das imagens criadas 52
Figura 22 - Montando o HD Externo na estação forense 52
Figura 23 - Exibindo informações dos dispositivos montados 53
Figura 24 - Usando o comando ls para exibir arquivos do diretório var 53
Figura 25 - Utilizando a ferramenta strings na imagem da máquina invadida 54
Figura 26 - Código fonte da página web modificada 54
Figura 27 - Realizando uma busca por comandos com a ferramenta strings na memória 55
Figura 28 - Realizando uma busca pela data com a ferramenta strings na memória 55
Figura 29 - Lendo o arquivo lastlog com a ferramenta cat 56
Figura 30 - Lendo o histórico de comandos com a ferramenta cat 56

LISTA DE SIGLAS

ABNT:	Associação Brasileira de Normas Técnicas
ARP:	Address Resolution Protocol - Protocolo de Resolução de Endereço
BIOS:	Basic Input/Output System - Sistema Básico de Entrada/Saída
CERT-BR:	Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil
DNS:	Domain Name System - Sistema de Nomes de Domínios
HD:	Hard Disk - Disco Rígido
IDS:	Intrusion Detection System - Sistema de Detecção de Intrusos
IP:	Internet Protocol - Protocolo de Internet
IPS:	Intrusion Prevention Systems - Sistema de Prevenção de Intrusos
ISO:	International Standards Organization - Organização Internacional de Normalização
MAC:	Media Access Control - Controle de Acesso ao Meio
OOV:	Order of Volatility - Ordem de Volatilidade
PID:	Process Identifier - Identificador de Processo
RAM:	Random Access Memory - Memória de Acesso Aleatório
TCP:	Transmission Control Protocol - Protocolo de Controle de Transmissão
TI:	Tecnologia da Informação
UDP:	User Datagram Protocol - Protocolo de Datagrama do Usuário

SUMÁRIO

1 INTRODUÇÃO 11

1.1 OBJETIVOS 12
1.1.1 Geral 12
1.1.2 Específicos 12
1.2 JUSTIFICATIVA 13

2 REFERENCIAL TEÓRICO 14

2.1 COMPUTAÇÃO FORENSE 14
2.2 COLETA DE EVIDÊNCIAS DIGITAIS 15
2.3 ANÁLISE FORENSE COMPUTACIONAL 16
2.3.1 Aquisição 17
2.3.2 Identificação 17
2.3.3 Avaliação 18
2.3.4 Apresentação 18
2.4 TIPOS DE ATACANTES 19
2.4.1 Hackers 19
2.4.2 Crackers 20
2.4.3 Script Kiddies 20
2.4.4 Insiders 21
2.4.5 Cyberpunks 23
2.4.6 White Hats 23
2.4.7 Gray Hats 24
2.4.8 Cyberterroristas 24
2.5 COMO AGEM OS ATACANTES 25
2.6 RESPOSTA A INCIDENTES DE SEGURANÇA 30
2.7 FORENSE IN VIVO 33
2.8 FORENSE DE REDE 34
2.9 FORENSE POST MORTEM 35
2.10 FERRAMENTAS FORENSES 35

3 METODOLOGIA 37

4 ANÁLISE DE RESULTADOS 38

4.1 Aplicação de ferramentas na Forense In Vivo 39
4.2 Aplicação de ferramentas na Forense Post Mortem 47
4.3 Elaborando o Laudo Pericial 58

5 CONCLUSÃO 59

REFERÊNCIAS 60

APÊNDICE 62

1 INTRODUÇÃO

Vivemos em um mundo globalizado fortemente articulado pelas redes onde a informação, independente do seu formato, é um dos maiores patrimônios de uma empresa moderna. Considerada um ativo importantíssimo para a realização do negócio, a informação deve ser protegida e gerenciada.

Nos últimos anos as tecnologias de informação e comunicação têm evoluído de forma rápida, fazendo com que as organizações tenham maior eficiência e rapidez nas tomadas de decisão. Portanto as chances de uma empresa não usar sistemas de informação tornou-se praticamente nula.

No passado a questão segurança da informação era muito mais simples, pois os arquivos contendo inúmeros papéis podiam ser trancados fisicamente. Porém com a chegada das tecnologias da informação e comunicação a questão ficou bem mais complexa, já que hoje a maioria dos computadores está conectada a *Internet* e os dados estão em formato digital, fazendo com que estes ativos tornassem atrativos para *cybercriminosos*.

Neste contexto surge a necessidade das empresas legais atuarem no combate aos crimes eletrônicos, dando origem a Forense Computacional.

Segundo Mati (2004 apud Melo, 2009), a Forense Computacional é um instrumento que possibilita a posterior reconstrução de eventos criminais, ou detecção de ações não autorizadas ou anômalas a comportamentos operacionais esperados ou planejados em um contexto computacional caracterizando um incidente de segurança, que depois poderão vir a se transformar em um laudo e/ou ação judicial, mas para isso é necessário o uso de ferramentas e técnicas específicas.

A falta de regulamentação das leis que punem os crimes cometidos no meio computacional e de controle da grande rede mundial de computadores (*Internet*) contribui bastante para ao aumento do número de incidentes de segurança que ocorrem a todo instante no mundo.

Existem hoje no mercado muitas ferramentas proprietárias para a análise destes incidentes, mas o seu alto custo acaba sendo um grande obstáculo para os profissionais de TI. Além das ferramentas proprietárias, existem também as ferramentas de *software* Livre, que são disponibilizadas gratuitamente na *Internet*. Elas podem vir isoladamente, ou agrupadas em Live CD, com dezenas ou até centenas de ferramentas para os mais diferentes tipos de

análises, e tudo isso a custo zero. No entanto, algumas delas, exigem-se conhecimentos sólidos em sistemas GNU/Linux para sua manipulação e o domínio do inglês.

Este trabalho mostrará passo-a-passo como realizar uma perícia em um servidor invadido utilizando o sistema operacional Ubuntu, não pela questão do custo, mas por ser o sistema de código aberto mais popular do mundo e baseado no Debian – ótima plataforma para criação de estações forenses, com suporte a múltiplos sistemas de arquivos e uma grande variedade de ferramentas de código aberto que devidamente combinadas reúnem recursos poderosos para qualquer perícia, além de ser base também para implementações de alguns projetos de Live CDs de segurança como Helix, Auditor, entre outros.

1.1 OBJETIVOS

1.1.1 Geral

Descrever de forma teórica e prática como realizar uma forense em sistemas GNU/Linux utilizando o sistema Operacional Ubuntu.

1.1.2 Específicos

- Estudar as técnicas de coleta Forense;
- Estudar as técnicas de análise Forense;
- Estudar os tipos de invasores;
- Estudar a atuação dos invasores;
- Estudar o Processo de Resposta a Incidentes;
- Estudar a Ordem de Volatilidade das Evidências;
- Estudar as ferramentas forenses;
- Analisar as ferramentas tecnicamente;

- Descrever todo o processo de uma perícia com ênfase na Forense In Vivo e na Forense Post Mortem;
- Descrever como elaborar o Laudo Pericial.

1.2 JUSTIFICATIVA

O crescimento e a popularização da *Internet* fizeram com que o número de tutoriais práticos para invasão de computadores crescesse bastante nos últimos anos.

Segundo a CERT-BR (2011) – organização ligada ao Comitê Gestor da *Internet* no Brasil, órgão que coordena todas as iniciativas de serviços da *Internet* no País e um dos principais Centros de Resposta a Incidentes de Segurança da *Internet* brasileira, o aumento dos ataques ocorre em uma proporção maior que o crescimento da *Internet* no País.

Investigar estes ataques é o trabalho do perito computacional que se utiliza da Perícia Forense para descobrir fraudes, rastros, evidências que possam levar ao autor que cometeu o delito.

As estatísticas, presentes no site da CERT-BR (2011), mostram o crescimento gradual das ameaças digitais nos últimos anos no Brasil, onde em 1999 foi o ano em que ocorreram menos notificações, apenas 3.107 e em 2009 atingiu um número recorde com 358.343 notificações de incidentes de segurança. Em 2010 este número, comparado com o ano de 2009, diminuiu para menos da metade para 142.844 notificações, e em Junho deste ano já atingiu 217.840, superando o ano de 2010. Estes números continuam aumentando e a previsão é de que supere o ano de 2009.

Segundo Melo (2009), no Brasil faltam pessoas capacitadas para trabalhar nessa área, embora a demanda seja cada vez maior. No âmbito nacional opera sob a coordenação da Polícia Federal e em alguns países existem empresas especializadas em investigar crimes digitais.

Infelizmente a Computação Forense é um assunto novo e pouco explorado no Brasil. Apesar de existir vasta documentação que pode ser encontrada na *Internet*, ela se encontra em muitos casos desestruturada e confusa, e a maioria dos livros desta área está em língua estrangeira, o que dificulta a leitura dos mesmos, justificando o desenvolvimento deste trabalho.

2 REFERENCIAL TEÓRICO

2.1 COMPUTAÇÃO FORENSE

Segundo Melo (2009), a Computação Forense é um campo de pesquisa relativamente novo no mundo e foi criada pela necessidade das instituições legais atuarem no combate aos crimes eletrônicos.

Entende-se por Crime Eletrônico ou Informático, qualquer tipo de crime que seja praticado ou provido através dos meios informáticos e/ou *Internet*. (NEVES, 2009)

De acordo com Machado; Eleutério (2011), a Computação Forense é responsável por determinar à dinâmica, a materialidade e a autoria de ilícitos ligados à área de informática, identificando e processando as evidências digitais em provas materiais de crime, com a utilização de métodos que visam à preservação e integridade da prova, conferindo-lhes validade probatória em juízo.

Segundo Melo (2009), por sua demanda ser muito recente, falta uma legislação específica. Portanto procura-se adaptar leis do mundo real para solucionar casos do mundo virtual.

A falta da existência de leis específicas para crimes virtuais, e o fato do invasor não estar fisicamente no local do crime, gera a sensação de anonimato, tornando os crimes virtuais cada vez mais comuns e elaborados.

De acordo com Casey (2004 apud Lima, 2009), um princípio fundamental na ciência forense é o Princípio da Troca de Locard. Esse princípio enuncia que "qualquer um, ou qualquer coisa, que entra em um local de crime leva consigo algo do local e deixa alguma coisa para trás quando parte". Numa cena de crime "tradicional", as evidências deixadas podem ser: impressões digitais, material orgânico contendo DNA, dentre outras. No caso da computação, os vestígios de um crime são digitais, uma vez que toda informação armazenada dentro desses equipamentos computacionais é composta por bits, (números zeros e uns), em uma sequencia lógica.

De acordo com Machado; Eleutério (2011), quem realiza o trabalho de identificar as evidências digitais de delitos e crimes de forma oficial no âmbito criminal é o Perito Criminal em Informática. Entretanto, diversos outros profissionais podem ter a necessidade de realizar exames em computação. São eles: peritos particulares, auditores de sistemas, profissionais de

TI e outros. Além disso, juízes, advogados, delegados, promotores e demais profissionais da área de direito também devem conhecer como evidências e provas digitais devem ser corretamente coletadas, apuradas e apresentadas em juízo.

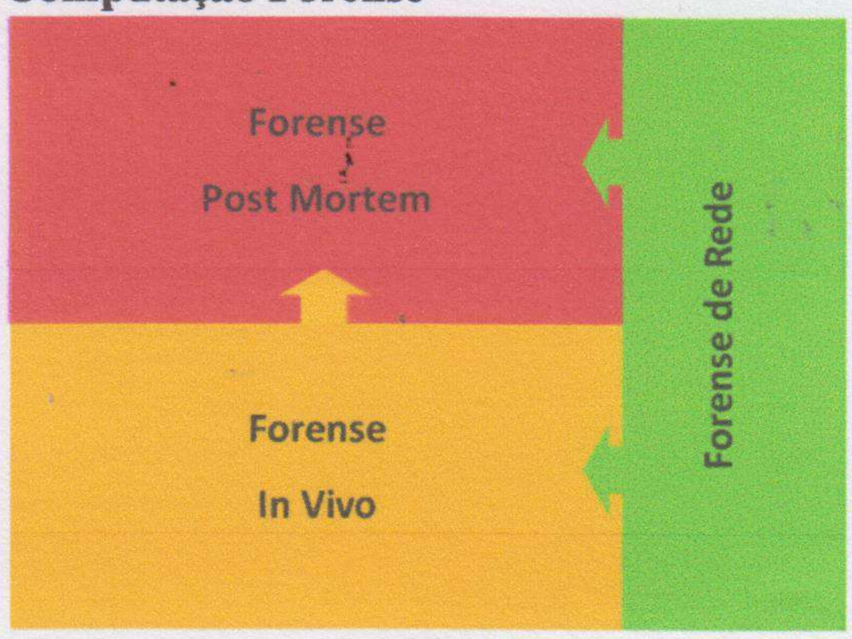
Geralmente o trabalho do Perito Criminal, segundo Machado; Eleutério (2011) inicia-se com o cumprimento de mandados de busca e apreensão ou exames em locais de crime. Em ambos os casos, cabe ao Perito à identificação e posterior coleta das provas, tomando-se o extremo cuidado com a preservação e a cadeia de custódia (relatório), que é o processo de garantia de proteção à prova, cujo objetivo é assegurar sua idoneidade, a fim de evitar questionamentos quanto à sua origem ou seu estado inicial. Para isso, devem ser registrados todos os caminhos percorridos pela prova desde a sua coleta até o trânsito em julgado da ação.

2.2 COLETA DE EVIDÊNCIAS DIGITAIS

De acordo com Santos (2008), antes de iniciar uma análise, o perito deverá avaliar qual procedimento utilizar para a coleta dos dados. A escolha deverá levar em conta o cenário e as circunstâncias em que se encontram o sistema a ser analisado. Alguns questionamentos devem ser feitos, como por exemplo: O que deverá ser coletado? O que coletar primeiro? Vale a pena coletar tudo? Será que não serão colocadas em risco todas as outras evidências só porque se pretende coletar algumas evidências em tempo de execução?

De acordo com Melo (2009), a coleta de evidências pode ser dividida em três etapas, conforme ilustra a Figura 1.

Figura 1 - Atividades operacionais da Computação Forense



Fonte: MELO (2009, p. 13)

A primeira etapa (Forense In Vivo) consiste na coleta de evidências antes do desligamento do sistema da fonte elétrica e tem por objetivo registrar o estado do sistema.

Já a segunda etapa, Forense de Rede, consiste na coleta e analise das informações de atividade da rede, tanto do servidor em questão, como dos demais ativos de redes que tenham informações pertinentes.

Por último, a etapa em que todas as informações reunidas nas etapas anteriores são cruzadas com as informações identificadas na análise das imagens de mídias coletadas. Essa última fase é tecnicamente conhecida como Post Mortem.

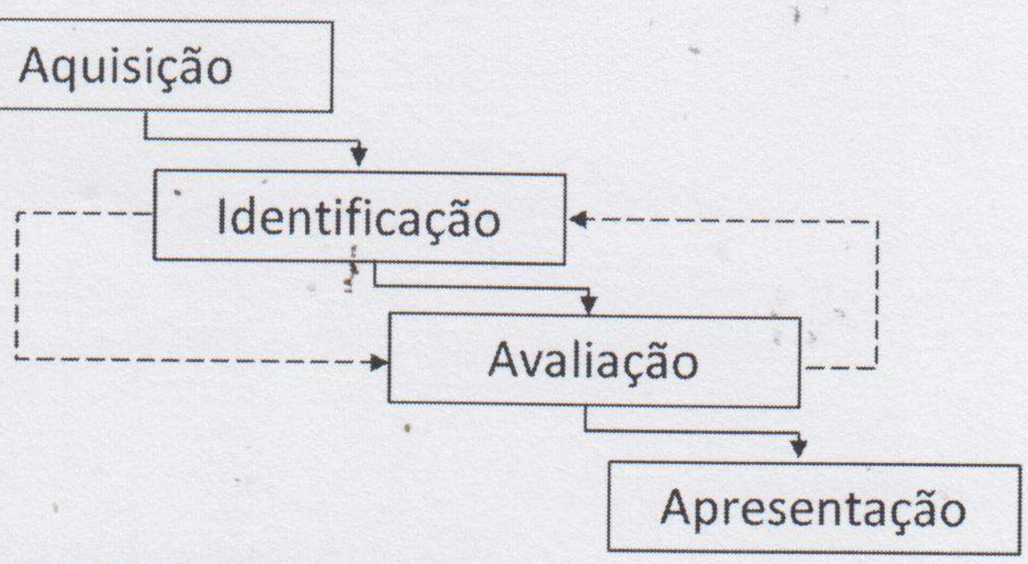
Tanto na Forense In Vivo, na Forense de Rede como na Forense Post Mortem, o objetivo é a coleta do máximo de evidências digitais. São etapas distintas, mas em momento algum independentes, pois se complementam e concorrem para a criação do Laudo Pericial.

De acordo com Freitas (2006 apud Paiva, 2009), um Laudo Pericial é um relatório técnico que aponta os resultados obtidos a partir das provas, no qual é avaliado pela justiça. No entanto, todos os procedimentos da forense computacional devem estar baseados nas legislações locais.

2.3 ANÁLISE FORENSE COMPUTACIONAL

Segundo Melo (2009), do ponto de vista de organização de processos, a Computação Forense pode ser dividida em quatro fases, conforme ilustra a Figura 2.

Figura 2 - Ciclo da Perícia Forense



Fonte: MELO (2009, p. 15)

2.3.1 Aquisição

A aquisição é a fase inicial, propicia para a aplicação da Forense In Vivo. Consiste em reunir o máximo possível de provas (artefatos a serem analisados).

De acordo com Melo (2009, p.15), "O sucesso de uma análise pericial depende fundamentalmente da qualidade do material coletado e da validade dos procedimentos adotados",

O momento da aquisição de provas é crucial e o Perito deve possuir capacitação técnica que lhe possibilite realizar a perícia nos mais variados contextos, desde uma coleta de dados em um computador pessoal, um celular, uma câmera digital, até em uma rede de computadores composta por servidores de sistemas operacionais variados, para cada uma delas existem metodologias e técnicas apropriadas para manipulação dos artefatos.

Segundo Espíndula (2005 apud Melo, 2009), a forma como os dados são armazenados demanda técnicas específicas, visto que a coleta de dados em um ambiente Linux, por exemplo, é feita por ferramentas diferentes da coleta em outro ambiente, embora os métodos e objetivos sejam similares.

2.3.2 Identificação

Segundo Melo (2009), a fase de identificação consiste na análise pericial que objetiva identificar e organizar os artefatos encontrados qualificáveis como vestígios, evidências ou provas, englobando tanto os artefatos identificados no processo, antes do desligamento do abrupto do equipamento denominado Forense In Vivo, como depois do desligamento, denominado Forense Post Mortem.

De acordo com Espíndula (2005 apud Melo, 2009), o Perito deve manter a mesma postura adotada na fase anterior e continuar sendo claro e metódico no processo de identificação de cada artefato digital (*trojan*, *backdoors*, *rootkits*). A documentação clara referente a cada artefato é crucial para a elaboração do Laudo Pericial. Dependendo do propósito da perícia, os dados disponíveis em um objeto periciado podem expor fatores determinantes para uma investigação e estabelecer nexos, mesmo que tal objeto esteja

desvinculado do dispositivo que desencadeou a resposta de incidente do sistema ligado à *Internet*.

2.3.3 Avaliação

Segundo Melo (2009), a fase de avaliação consiste na análise mais pontual dos artefatos e da enumeração dos eventos em uma linha de tempo. É a fase em que o Perito Forense demonstra suas principais habilidades, embora sua participação em todas as partes do processo seja extremamente relevante.

É difícil dimensionar a duração dessas atividades em quantidade de horas, pois cada Perícia Forense é única e vinculada ao contexto de como ocorreu o incidente. Ela pode ter por objetivo coletar dados periciais para fins de provas de crimes não necessariamente vinculados apenas à invasão ou ao uso abusivo de um servidor em uma rede, mas executados por meios eletrônicos ou computacionais como fraudes financeiras, pedofilia e assédio em ambiente de trabalho.

Um Perito deve ter uma postura criteriosa na análise, sem imediatismos, para não elaborar o Laudo Pericial com conclusões prematuras.

2.3.4 Apresentação

Segundo Melo (2009), a fase de apresentação tem o objetivo de desvendar o caso em todos os seus detalhes. Ao perito cabe apresentar um relatório de tudo que é analisado e, se possível, identificar o que ocorreu na Perícia Forense decorrente de um Incidente de Segurança Computacional. Pode-se dizer também que o relatório reconstrói o *Modus Operandi* do potencial atacante ao sistema.

O Laudo Pericial de Incidente de Segurança pode no seu conteúdo sugerir a origem e a categoria do Incidente. Deve ser acompanhado de dados periciais para sua fundamentação.

Dois cuidados devem ser tomados com o Laudo Pericial: o original deve ser apresentado como prova e apresentar cópias somente se não estiver disponível o original, pois quando se trata de dados digitais, eles podem ser reproduzidos com plena integridade. Por

último o Perito deve ser capaz de defender a metodologia utilizada para garantir a veracidade dos dados periciais identificados mediante a análise de cada artefato.

2.4 TIPOS DE ATACANTES

Segundo Tanenbaum (2003, p. 767)

> A maior parte dos problemas de segurança é causada intencionalmente por pessoas maliciosas que tentam obter algum benefício, chamar a atenção ou prejudicar alguém. [...] Para tornar uma rede segura, com frequência é necessário lidar com adversários inteligentes, dedicados e, às vezes, muito bem subsidiados.

Hacker é o termo normalmente utilizado para designar àquelas pessoas que realizam ataques a sistemas de computação. Isso acabou associando a imagem do *hacker* ao dos *cybercriminosos* e àqueles que fazem o mau uso da tecnologia. Na verdade quem pratica estes atos recebem outra nomenclatura conforme o seu perfil e forma de ataque. Essa confusão não é à toa, já que é comum nas mídias de TV, *Internet* e Jornais este termo ser bastante utilizado. Existem classificações para cada tipo de atacante, que também são *hackers*, mas com objetivos e filosofias diferentes.

A seguir será apresentada a definição e a classificação de alguns tipos de *hackers*:

2.4.1 Hackers

Segundo Oliveira (2008 apud Câmara, 2010), *hackers* são indivíduos especialistas em segurança e auditoria em sistemas de computação, que, através de algumas técnicas estudadas e outras próprias, procuram falhas nesses sistemas. Utilizam o conhecimento que possuem para invadir sistemas, não com a intenção de causar danos às vítimas, mas como uma forma de desafio às suas habilidades. Podem invadir redes de computadores ou sistemas, acessar ou alterar arquivos para provar o quanto são capazes de realizar essas ações e para, depois, mostrar suas façanhas para os companheiros. Não querem prejudicar, mas somente provar quanto conhecimento e poder possuem. Os *hackers* não gostam de ser confundidos com os

crackers, são excelentes programadores e conhecem muito bem os segredos das redes e dos computadores.

2.4.2 Crackers

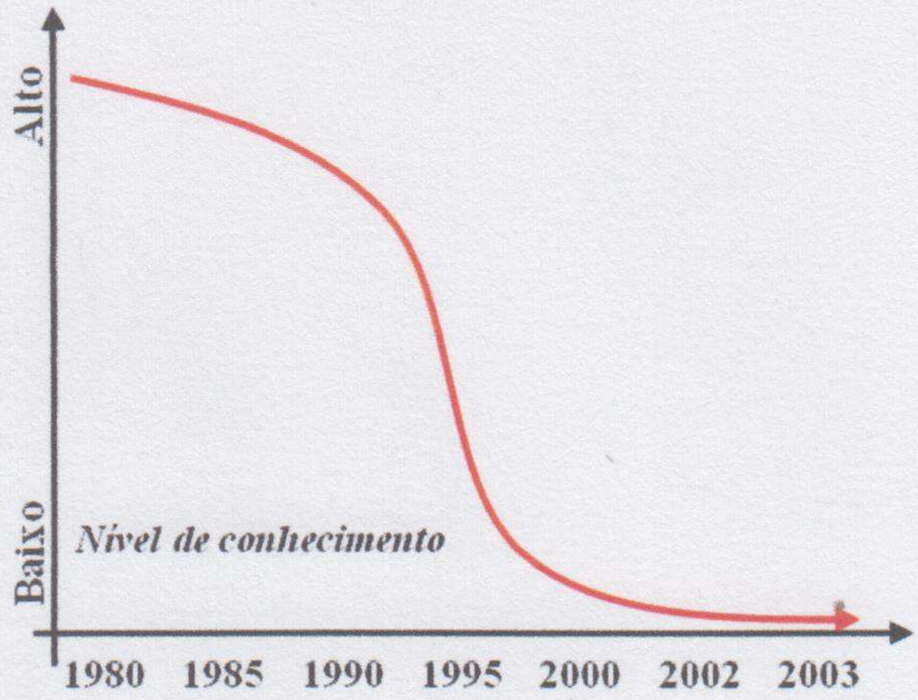
De acordo com Oliveira (2008 apud Câmara, 2010), *crackers* ou *Black Hats*, são indivíduos que invadem sistemas para subtrair informações indevidamente e produzir danos às vítimas. Além disso, decifram códigos e destroem proteções de programas de computador. São os verdadeiros terroristas da *Internet* e, sem medir as consequências de seus atos, fazem o que for preciso para conseguir o que querem. Os *crackers* são, até mesmo, desprezados pelos próprios *hackers*. Às vezes, tentam vender as informações roubadas para a própria vítima, ameaçando torná-las públicas se a empresa ou usuário não pagar o valor pedido por eles.

2.4.3 Script Kiddies

Conforme Oliveira (2008 apud Câmara, 2010), os *script kiddies* são indivíduos principiantes e inexperientes com conhecimentos básicos de computação, mas que podem acarretar diversos problemas para as corporações. Geralmente buscam informações e ferramentas encontradas prontas na *Internet* e as utilizam sem saber o que estão fazendo. Essas ferramentas são bem fáceis de serem obtidas, por isso os *script kiddies* podem trazer perigo a uma grande quantidade de empresas, principalmente as que não possuem uma política de segurança definida adequadamente.

O gráfico da Figura 3 mostra o nível de conhecimento demandado por um invasor, que dos anos 80 era alto e foi diminuindo gradativamente até que em 2003 caiu bastante, exigindo pouco conhecimento do atacante, a este se dá o nome de *script kiddie*.

Figura 3 - Nível de conhecimento demandado por um invasor



Fonte: MELO (2009, p. 27)

Analisando a Figura 3 é possível perceber o perigo que as corporações, organizações militares e órgãos, tanto governamentais como não governamentais, estão assumindo ao colocarem servidores conectados à *Internet*, já que qualquer usuário com conhecimentos básicos de computação pode gerar um incidente de segurança de grandes proporções, dependendo do contexto em que venha a ocorrer o fato. Tal constatação motiva investimentos em segurança não somente no contexto computacional, mas também no de Segurança da Informação.

2.4.4 Insiders

Segundo Oliveira (2008 apud Câmara, 2010), *Insiders* são pessoas que trabalham dentro da própria organização. São considerados os causadores dos incidentes de segurança mais graves.

De acordo com Mitnick (2003 apud Câmara, 2010), os próprios funcionários das empresas são considerados as maiores ameaças:

> A grande maioria dos empregados que são transferidos, demitidos ou rebaixados nunca causa problemas. Mesmo assim é preciso apenas um deles para fazer uma empresa perceber tarde demais as medidas que poderiam ser tomadas para evitar o desastre. A experiência e as estatísticas têm mostrado claramente que a maior ameaça para a empresa vem de dentro. São as pessoas que estão dentro que têm um conhecimento grande do lugar onde

> ficam as informações valiosas e de onde a empresa pode ser atingida para causar o maior dano.

Um estudo conduzido pela consultoria norte-americana Verizon, mostra que ataques virtuais conduzidos por pessoas de dentro da própria empresa – os chamados *Insiders* – respondem por importante parte do total de ocorrências. Em 2009, os ataques internos chegaram a 48% dos casos estudados. Em 2010, o número caiu para 17%, embora a Verizon acredite que não houve, necessariamente, queda do número deste tipo de ocorrência, mas sim um grande aumento no número de ataques externos. (BORBA, 2011)

Várias são as razões que levam um funcionário a prejudicar as redes das empresas onde trabalham. Há as pessoas que agem de forma inadequada, mas não maliciosa; os que agem involuntariamente, sem malícia; e, por fim, os que agem deliberada e maliciosamente – é o caso, por exemplo, de funcionários que cometem fraudes para obter algum tipo de lucro ou de funcionários demitidos que invadem as redes em busca de vingança. O 12º Estudo Global de Segurança da Informação, realizado em 2010 pela Ernst & Young, aponta que as represálias de ex-funcionários, aliás, são o maior risco para a área de TI das empresas. Eles apagam dados de servidores, enviam *e-mails* em nome de outras pessoas – muitas vezes do presidente da empresa – ou mesmo roubam dados de clientes.

Apesar da dificuldade em se identificar os *Insiders*, geralmente são funcionários descontentes com a empresa e acham que suas atividades não são reconhecidas pelos chefes. Muitas vezes, são destratados e querem mostrar seu valor fazendo algo que façam sentirem-se importantes. Os concorrentes podem facilmente manipular esses indivíduos, persuadindo-os e sabendo que não se encontram em uma condição muito confortável dentro da empresa.

As organizações devem prestar bastante atenção aos *Insiders*, pois podem estar perdendo mercado e espaço para seus concorrentes sem saber por que isso está acontecendo. Os concorrentes podem ter conseguido acesso a dados através do roubo de informações causados pelos *Insiders*.

Segundo Oliveira (2008 apud Câmara, 2010), a segurança é antes de tudo um problema social e não apenas um problema tecnológico. Dessa forma, as organizações não podem esquecer os aspectos humanos, pessoais e sociais no processo de elaboração da estratégia de segurança.

2.4.5 Cyberpunks

De acordo com Oliveira (2008 apud Câmara, 2010), *Cyberpunks* são aquelas pessoas que se dedicam a invadir sistemas apenas pelo desafio ou por simples divertimento. A grande preocupação deles é com o governo que pode estar acessando dados privados das pessoas. Têm muito conhecimento e obsessão pela privacidade de suas informações, fazendo com que todas as suas comunicações sejam realizadas com criptografia. Muitas vezes, são os *Cyberpunks* que acham vulnerabilidades em sistemas, serviços ou protocolos, fazendo um grande favor às corporações, divulgando as falhas encontradas. Isso contribui para que os fabricantes de *softwares* corrijam seus programas e passem a desenvolvê-los dando maior importância à segurança. Entretanto, os fabricantes de programas ainda têm preferência por corrigir seus programas a escolher um método de desenvolvimento dos *softwares* com mais enfoque na segurança. Isso pode ser constatado através do grande número de falhas que ainda aparecem nos sistemas.

2.4.6 White Hats

Conforme Oliveira (2008 apud Câmara, 2010), os *White Hats* usam suas habilidades para encontrar falhas de segurança nos sistemas e realizar as correções necessárias, trabalhando profissionalmente e legalmente nas empresas. Também são conhecidos como "*hackers* do bem", "*hackers* éticos", samurais ou *sneakers*. São tidos como guerreiros que tem a função de proteger os sistemas de computação das corporações e, para isso, realizam testes de invasões, simulam ataques para verificar o nível de segurança da rede e efetuam diversas análises necessárias para proteger os dados das empresas.

Porém, deve-se tomar uma série de cuidados antes de contratar os serviços de um *White Hat*. É preciso determinar limites na simulação de ataques para prevenir uma exposição de dados confidenciais. Outro ponto importante é colocar de forma clara no contrato que os dados obtidos permanecerão sob sigilo e a garantia se que as correções necessárias devem ser executadas. Um *White Hat* pode ser importante para a segurança de uma empresa, mas limites precisam ser impostos para que uma situação indesejada não ocorra no futuro. Eles podem

achar diversas vulnerabilidades no sistema e depois querer cobrar para efetuar as correções necessárias.

Para Mitnick (2003 apud Câmara, 2010), a segurança é um aspecto multidisciplinar que abrange diversos fatores diferentes e caso as pessoas não possuam o conhecimento necessário sobre os riscos a que estão expostas, podem achar que estão seguras. Ou seja, não há como proteger-se contra o que não se conhece. Isso torna o conhecimento importantíssimo para uma proteção apropriada.

2.4.7 Gray Hats

De acordo com Oliveira (2008 apud Câmara, 2010), *Gray Hats* são *Black Hats* trabalhando na área de segurança fazendo o papel de *White Hats*. Muitas empresas contrata-os para fazer análises de segurança, mas vários problemas já aconteceram e mostraram que o nível de confiança necessário para a execução dessas atividades tão críticas e estratégicas não é atingida através da contratação do serviço dessas pessoas.

A verdade é que utilizar os serviços de um *Gray Hat* para cuidar da segurança de uma instituição pode ser extremamente perigoso, principalmente por causa da própria natureza desses indivíduos.

2.4.8 Cyberterroristas

Segundo Oliveira (2008 apud Câmara, 2010), *cyberterroristas* são aqueles que realizam ataques com a intenção de divulgar mensagens religiosas ou políticas, danificar a estrutura de comunicações ou conseguir informações para comprometer a segurança de algum país. Realizam pichações em sites, alterando uma página do site para difundir informações falsas e mensagens religiosas ou políticas, além de ataques de negação de serviços distribuídos e ataques a sistemas para conseguir dados confidenciais. Esses ataques podem também ter cunho pessoal, transformando-se em uma espécie de troféu para o autor - um prêmio pela sua capacidade de penetrar na segurança de um determinado sistema.

O ato pode ser analogamente comparado à pichação de muros e paredes. O autor do *defacement* geralmente é referenciado como *defacer* ou também pichador.

O *Defacement* (ou simplesmente *deface*) é um termo de origem inglesa para o ato de modificar ou danificar a superfície ou aparência de algum objeto e é empregado comumente na Segurança da informação para categorizar os ataques realizados por *defacers* e *script kiddies* para modificar a página de um site na *Internet*.

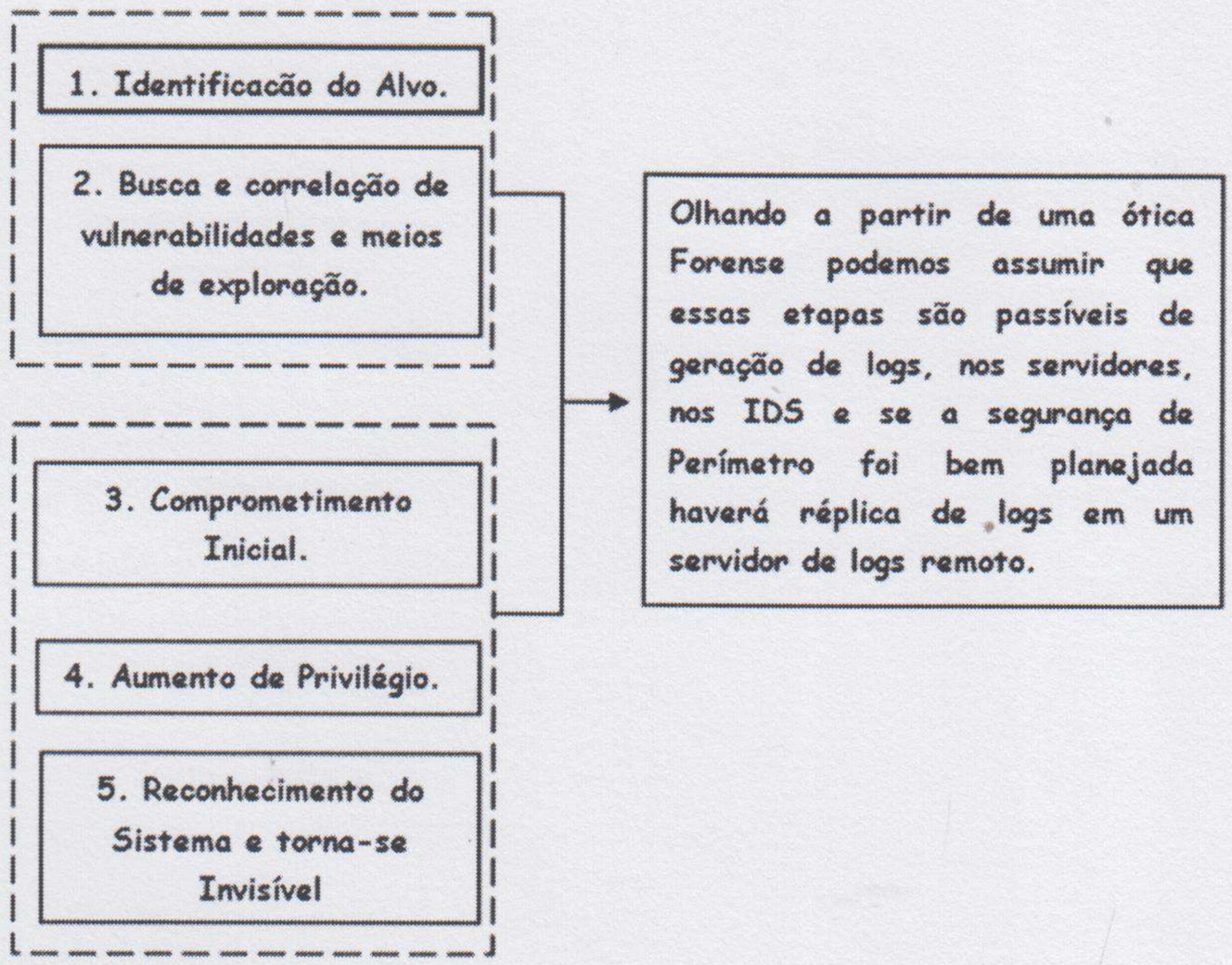
2.5 COMO AGEM OS ATACANTES

Conhecer como os atacantes agem possibilita uma visão macro dos principais pontos do processo da invasão a um sistema.

Segundo Melo (2009), possibilita antecipar onde, potencialmente, serão gerados dados para uma Perícia Forense Computacional, melhorando a atuação do perito durante a resposta a um incidente de Segurança.

A Figura 4 ilustra uma visão macro de como age um invasor, ou seja, seu *Modus Operandi*, que primeiramente identifica o alvo e depois busca por vulnerabilidades e meios de exploração. Após identificar a vulnerabilidade consegue acesso à máquina e aumento de privilégio. Por fim faz um reconhecimento do sistema e torna-se invisível para não ser detectado por administradores ou sistemas de detecção de intrusão (IDS) que possam estar monitorando a rede.

Figura 4 - *Modus Operandi* - visão macro



Fonte: MELO (2009, p. 21)

O fluxo de ações do processo de uma invasão de sistema conforme ilustrado na Figura 4 geram dados periciais digitais, que podem ser classificado como:

- Vestígio: É um sinal, resquício, indício de uma ação que se torna um elemento que demanda investigação para que seja ratificada futuramente sua relevância como uma potencial evidência ou mesmo prova;
- Evidência: Um dado mais apurado, uma certeza manifesta que tem a qualidade ou condição do que é evidente e que pode se tornar um fato notório. Ela poderá ser conclusiva ou circunstancial;
- Prova: Aquilo que demonstra ou estabelece a verdade de um fato ou testemunho. Uma evidência validade dentro do processo investigatório.

Segundo Melo (2009), o contexto é fator determinante para a perícia, pois a forma como o servidor foi instalado, os ativos de redes disponíveis como IDS, *Firewall*, Servidores de Registro (*logs*) são importantes para o Perito Forense Computacional no momento de reconstruir os passos dos invasores. O invasor pode ser engenhoso a ponto de utilizar técnicas

furtivas e anti-forense buscando eliminar seus rastros. Informações levantadas nos ativos de redes tornam-se determinantes na análise da invasão.

Segundo Melo (2009), uma invasão de sistema pode ocorrer de forma clássica onde a possibilidade de criação de dados periciais é bem maior. A seguir é apresentado um processo de invasão onde o criminoso utiliza a forma clássica:

1. O invasor inicia pelo levantamento de dados que pode partir de engenharia social, seguida do levantamento de informações básicas, que é denominado *Footprint*. Em seguida, busca informações disponíveis no registro *Whois*, base DNS e do sistema operacional remoto, a partir de análise dos pacotes montados pela pilha TCP/IP do servidor alvo, seja de forma ativa, por meio de *scanners* de *Fingerprint*, ou de forma passiva, por meio de *software* de *Fingerprint* que captura pacotes e os analisa;
2. Após o *Footprint*, inicia-se o levantamento dos serviços de redes ativos, que demanda o uso de *Port Scanners*, podendo disparar alertas em IDS ou mesmo no servidor alvo, caso haja mecanismos para detecção de varreduras;
3. Por outro lado, existe a possibilidade do invasor em cada ação evitar a execução das técnicas de varredura que possam chamar atenção;
4. O próximo passo é a enumeração dos serviços ativos identificados no servidor alvo, o invasor pode utilizar técnicas de identificação de vulnerabilidades.

Até esse momento, a intrusão não ocorreu efetivamente, mas, todas as técnicas citadas, caracterizam um processo de invasão em curso, haja vista que, para o invasor ter acesso ao sistema, demanda-se a identificação de pelo menos uma vulnerabilidade que pode ser uma falha em uma aplicação, ou mesmo na configuração de um ativo de rede.

Quando uma rede é projetada para responder a um Incidente de Segurança, deve estar apta a reter o maior número possível de informações sobre potenciais eventos. Pois essas informações podem se tornar evidências ou provas importantes durante a Perícia.

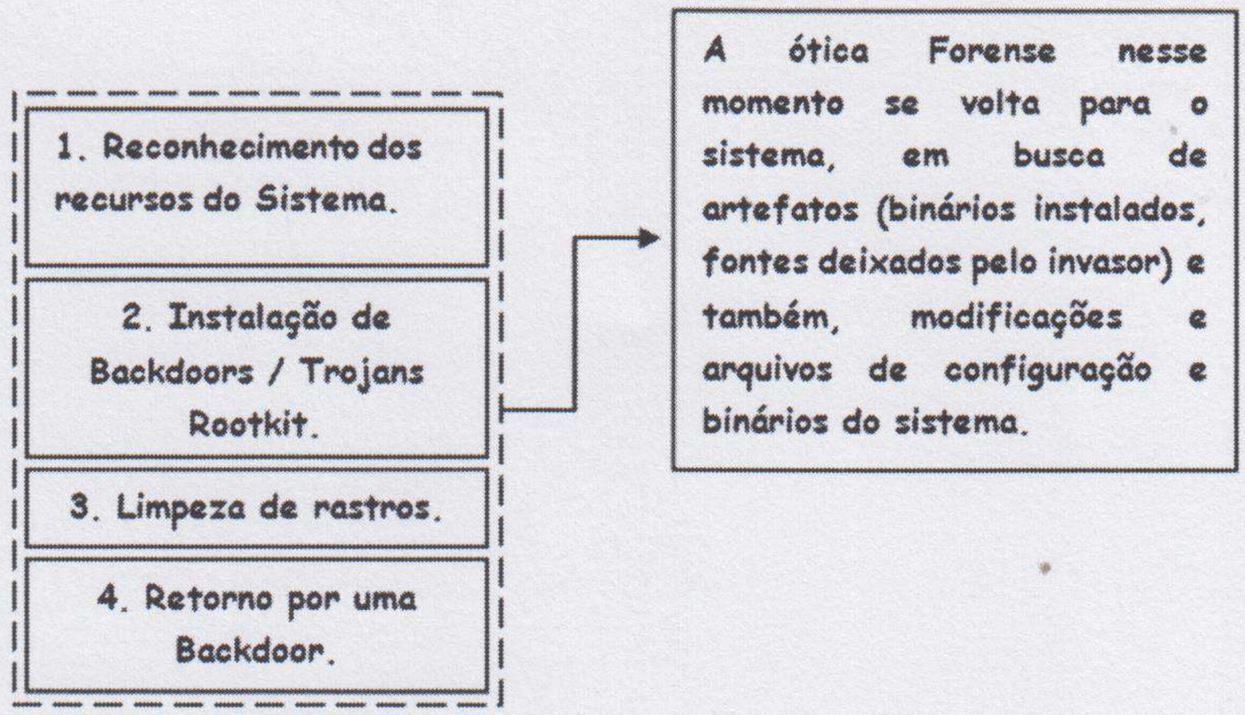
O conhecimento necessário para tratar de segurança de redes é bastante amplo e o profissional designado para cuidar desse importante segmento deve ser bem treinado visto que suas falhas podem representar prejuízos imensuráveis ao patrimônio da empresa. Segundo Melo (2009), o treinamento deve suprir pelo menos três tipos de capacidades:

- A habilidade de projetar segurança de redes de servidores, utilizando princípios básicos como: Menor privilégio, *Hardening*, Segurança de Perímetro, Segurança em Camadas, entre outros;
- A habilidade de testar segurança de redes, por meio de testes de segurança que avaliam ativos de redes e que possibilitam quantificar níveis de segurança. No caso especial de ativos de redes destinados à segurança, como *Firewalls* e IDS, identificar até onde suas respectivas tecnologias são efetivas diante de reais ameaças dentro de um contexto controlado;
- A habilidade de responder a Incidentes de Segurança utilizando os procedimentos e metodologias do Perito Computacional Forense, mesmo que o objetivo não seja a elaboração de provas para uma Ação Judicial. De fato, o profissional de TI não é isento de responder aos incidentes em juízo e, sendo assim, é preferível adotar, como regra, o uso de metodologias, técnicas e ferramentas forenses.

O profissional de segurança com sólidos conhecimentos em forense computacional reúne as qualidades necessárias para atuar de forma mais ampla dentro de sua especialidade e com uma visão mais coesa de como ataques podem ser executados e dos impactos que as ameaças digitais podem causar. Ele adquire maior capacidade de projetar a segurança dos servidores e da própria rede, introduzindo recursos eficazes na contenção, mas preparando-se para o pior dos cenários: quando o incidente já ocorreu.

A Figura 5 ilustra as ações de um invasor após a concretização da invasão do sistema, onde o invasor primeiramente instala na máquina invadida um conjunto de *softwares* maliciosos para estabelecer pleno controle do sistema, ocultar suas atividades, apagar seus rastros (registros de *logs*), capturar mais informações e, ainda em alguns casos, utilizar técnicas anti-forenses.

Figura 5 - Ações do invasor após a intrusão do sistema



Fonte: MELO (2009, p. 24)

Na Figura 5, apesar de o invasor ter instalado na máquina invadida ferramentas para tentar apagar seus rastros, o Princípio da Troca de Locard, explicado em capítulos anteriores, diz que isso é praticamente impossível, já que segundo ele "qualquer um, ou qualquer coisa, que entra em um local de crime leva consigo algo do local e deixa alguma coisa para trás quando parte".

De acordo com Melo (2009), modificar arquivos e instalá-los significa alterar os tempos registrados no *MACtime* (termo técnico para os três horários: *mtime*, *atime*, *ctime*).

Segundo Farmer e Venemà (2007), os *MACtimes* podem fornecer uma excelente visão sobre a atividade passada em uma máquina. O atributo *atime* refere-se à última data e hora em que o arquivo ou diretório foi acessado (lido). Já o atributo *mtime* data e hora da última vez em que o conteúdo de um arquivo foi modificado. E por último o atributo *ctime*, que exibe a data e hora da última modificação das propriedades do arquivo como direitos e atributos.

De acordo com Melo (2009), existem alguns modos de ataque que não necessitam da instalação de qualquer programa e que deixam poucos dados periciais, tornando-se um grande desafio para o Perito devido a sua volatilidade. Um exemplo é quando um invasor utiliza técnicas como transferência de instruções maliciosas direto para a memória RAM da máquina invadida.

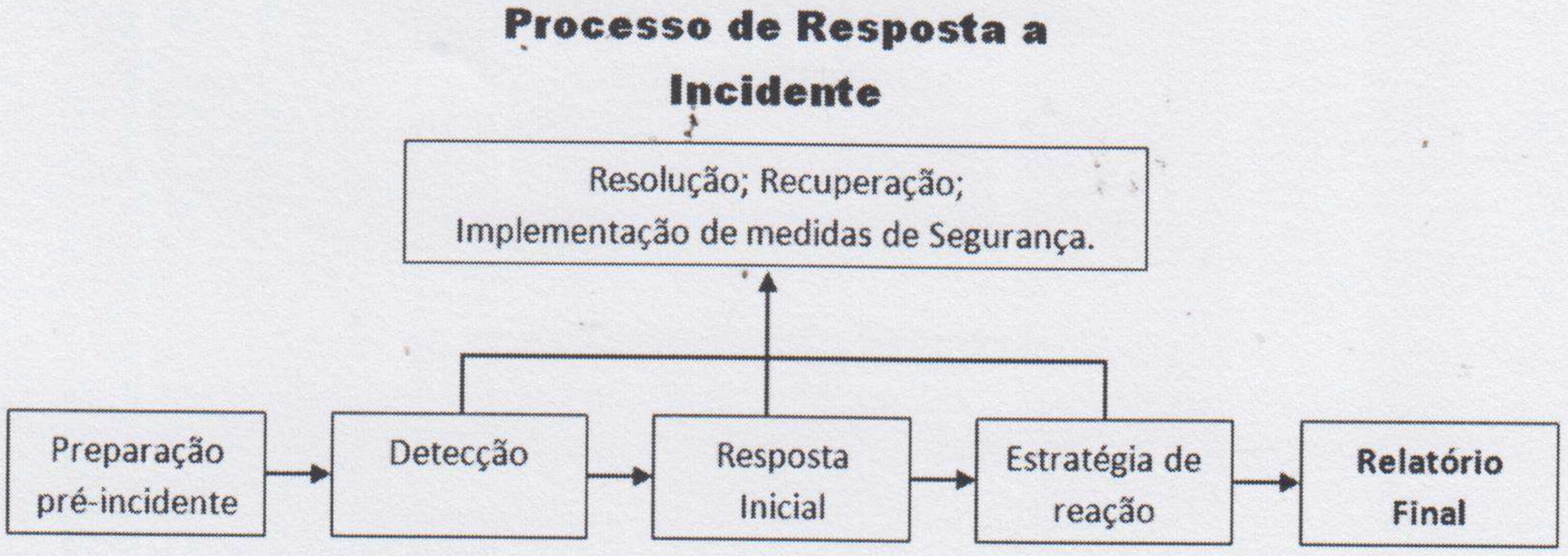
2.6 RESPOSTA A INCIDENTES DE SEGURANÇA

Uma vez conhecido o *Modus Operandi* dos invasores, é possível desenvolver tanto ferramentas como técnicas para responder aos ataques, ou seja, responder a um Incidente de Segurança de forma concisa e efetiva.

Tradicionalmente, o processo de resposta a Incidente de Segurança segue o fluxo ilustrado na Figura 6, onde:

- Pré-incidente: são realizadas ações para preparar a organização e a equipe de segurança antes que o incidente ocorra;
- Detecção: identificam um incidente de segurança em potencial;
- Resposta inicial: é uma investigação inicial, um registro de detalhes básicos que envolvem o incidente, configuração de uma equipe de resposta a incidentes, notificação de pessoas envolvidas no incidente;
- Estratégia de reação: nessa etapa, uma vez fundamentado os resultados de todos os fatos conhecidos, são determinadas as melhores respostas. Uma vez obtida aprovação gerencial, serão determinadas quais ações são cabíveis perante conclusões da investigação;
- Relatório final: são confeccionados em formato que seja útil para gerentes. Finalmente, a resolução que emprega medidas de segurança e mudanças de procedimento, registra as lições apreendidas e desenvolve soluções para problemas identificados.

Figura 6 - Processo de Resposta a Incidente



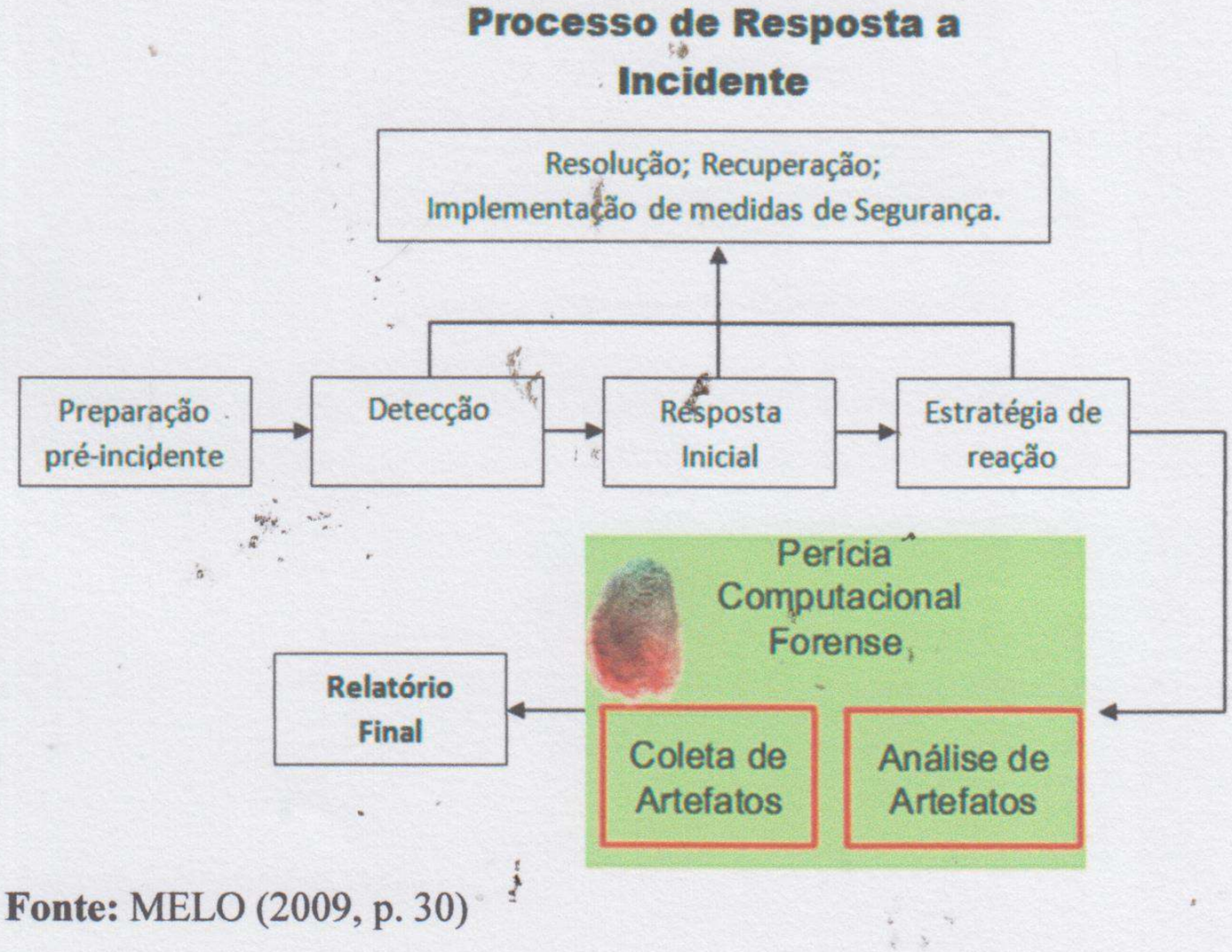
Fonte: MELO (2009, p. 29)

Pressupõe-se que exista uma pessoa ou equipe capacitada em atuar, se necessário, em Resposta a Incidentes de Segurança. Normalmente nas empresas, a equipe responsável pela segurança assume essa responsabilidade.

De acordo com Melo (2009), devido ao fato dos Incidentes de Segurança estarem mais relacionados a crimes por meios eletrônicos houve uma mudança no perfil do profissional de segurança: ele deve ser capaz de converter dados periciais gerados nos incidentes em provas válidas para uma ação judicial. Embora as empresas normalmente prefiram resolver problemas de Incidente de Segurança sem disparar um processo legal, este profissional deve estar apto para isso, quando necessário. O domínio das técnicas, ferramentas e metodologia de Perícia Forense Computacional possibilita ao profissional de segurança agir de forma metódica e propícia para que as informações coletadas tornem-se válidas em um tribunal.

A Figura 7 ilustra a Perícia Forense Computacional como parte do processo de Resposta a Incidente de Segurança, explicado anteriormente na Figura 6.

Figura 7 - Perícia Forense inserida no Processo de Resposta a Incidente



Fonte: MELO (2009, p. 30)

Segundo a ABNT (2005 apud Melo, 2009), o profissional de segurança ao responder a um Incidente de Segurança deve ser capaz de identificar:

- Origem: se o Incidente é interno ou externo;
- Motivação: se o ataque busca trocar uma página ou há um dano maior, como a total sabotagem dos dados;
- Classificação: se o ataque é um crime em andamento;
- Tecnologia: se constitui um ataque automatizado sendo executado por *Malware*;
- Meta: se perpetra violações de confidencialidade, disponibilidade e/ou integridade da informação;
- Danos: se houve uso impróprio de sistemas de informação;
- Robustez da infra-estrutura: se houve violações de acesso;
- Foco da exploração: se houve mau funcionamento de *Hardware* ou de *Software*;
- Grau de resistência: se houve violação da politica de Segurança Física;
- Extensão: quais serviços se tornaram temporariamente inativos.

O profissional ou a equipe de segurança devem ser ágeis a ponto de impedir que algum ativo importante da rede fique indisponível e gere prejuízos à organização.

Segundo Melo (2009), a preocupação com a coerência da resposta a Incidentes de Segurança garantiu um capítulo na NBR ISSO/IEC 17799:2005, uma das ISO's mais conceituadas e utilizadas no segmento mundial. Assim, é uma boa prática desenhar uma política de segurança interna que atenda no mínimo às recomendações da NBR ISSO/IEC 17799:2005 e fundamental ter um plano de ações devidamente estruturado em que todos os envolvidos saibam como reportar e documentar o Incidente.

De fato, o uso das técnicas, ferramentas e metodologia Forense tornam o profissional mais preparado para atuar em reposta a um Incidente de Segurança. Segundo Freitas (2003 apud Melo, 2009), a própria norma NBR ISSO/IEC 17799:2005 esclarece que devam existir controles para coletas que preservem a qualidade e a integridade do dado pericial e sua admissibilidade, mantendo controle, inclusive da cadeia de custódia. A conformidade com a NBR ISSO/IEC 17799:2005 somente é possível se os procedimentos usuais da Perícia Forense Computacional forem empregados de forma correta.

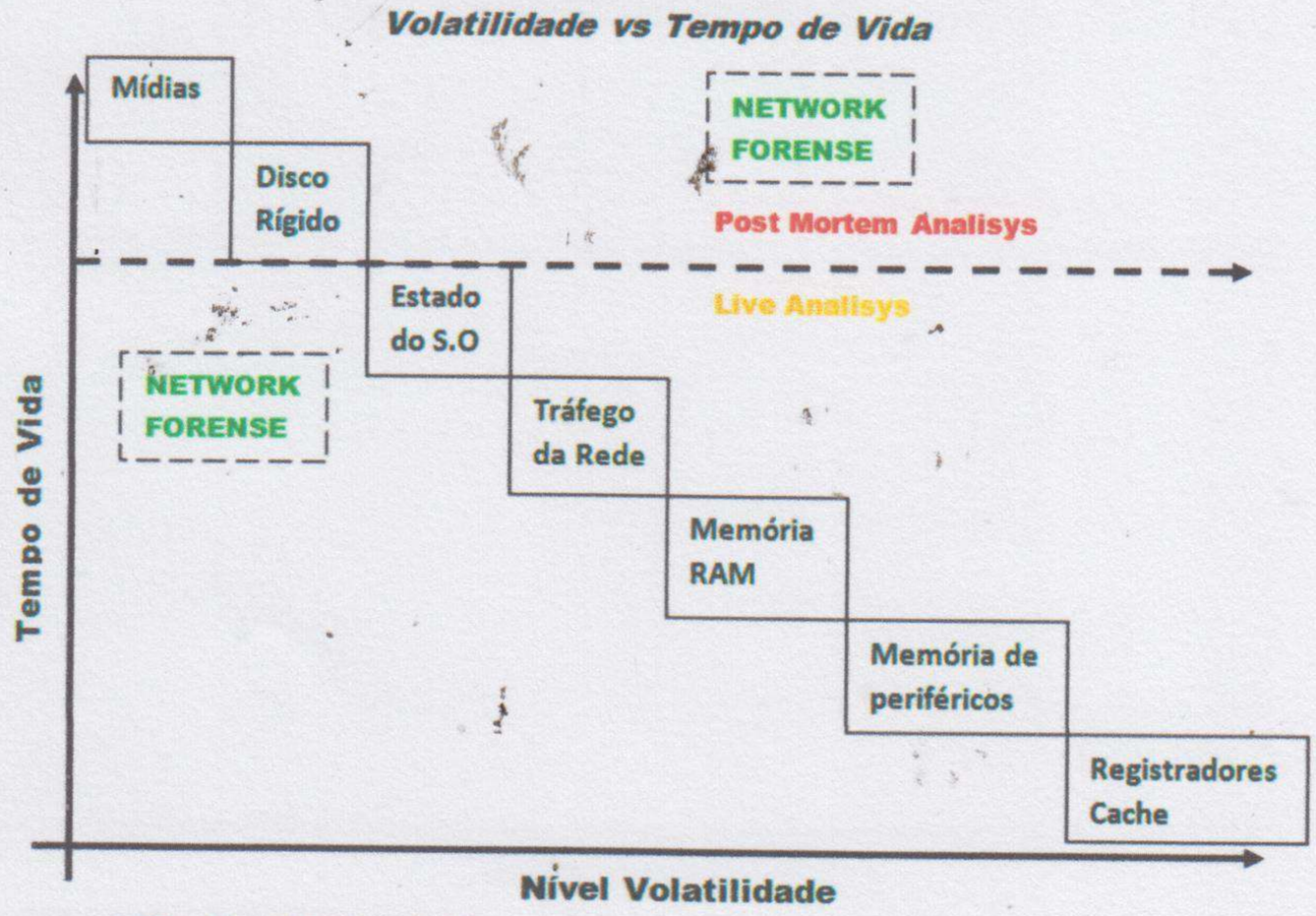
2.7 FORENSE IN VIVO

A Forense In Vivo, conforme descrito anteriormente, consiste num conjunto de procedimentos que são utilizados para a coleta de dados sem que o sistema seja desligado. De acordo com Santos (2008), nesta é priorizada a coleta dos dados mais voláteis para os menos voláteis, pois em sua maioria, os dados mais voláteis serão perdidos caso o sistema venha a ser deligado.

Surge então o conceito denominado ordem de volatilidade (OOV), introduzido por Dan Farmer e Wietse Venema, que segundo Melo (2009), é justamente a razão da importância da informação versus o tempo que ela fica imutável.

Este conceito, ilustrado na Figura 8, determina o tempo de vida de uma evidência digital, que pode variar de acordo com o local em que ela está armazenada, quanto maior a volatilidade de uma informação, mais difícil é a sua extração e menos tempo há para capturá-la. Algumas informações como as dos *caches* e registradores são tão voláteis que acabam não sendo relevantes como evidência digital, uma vez que o simples ato de coletar informações altera o estado delas.

Figura 8 - Volatilidade das evidências



Fonte: MELO (2009, p.35)

Entretanto, mesmo com alta volatilidade, algumas informações voláteis como o conteúdo da memória principal do sistema, o tráfego da rede e o estado do sistema

operacional podem ser capturados com relativa facilidade e podem conter informações relevantes a respeito das intrusões em andamento que devem ser coletadas no processo de Forense In Vivo.

Segundo Farmer e Venema (2007), seguindo essa ordem de volatilidade há uma maior probabilidade de se preservar os detalhes mais efêmeros que uma mera coleta de dados pode destruir, e ela permite que você capture dados sobre o incidente em questão, em vez de simplesmente capturar os efeitos colaterais da sua sessão de coleta de dados.

Em geral, coletar dados de acordo com a OOV ajuda a conservar em vez de destruir.

2.8 FORENSE DE REDE

A Forense de Rede consiste em coletar dados dos demais ativos de redes envolvidos em um Incidente de Segurança para que durante a Forense Post Mortem, esses dados correlacionados com demais evidências coletadas na Forense In Vivo sejam argumentos de apoio a conclusões quanto à ação do invasor.

Segundo Melo (2009), as evidências em uma Forense de Rede podem ser pouco esclarecedoras ou muito difíceis de serem levantadas. Portanto é crucial para a investigação coletar dados de todos os ativos disponíveis como: IDS, IPS, Servidor de *Logs*, entre outros, objetivando obter informações acerca do ocorrido que possam confirmar e responder a questionamentos quanto à autonomia da intrusão tais como: O invasor utilizou alguma técnica de levantamento de dados, como *footprint*, *fingerprint* e *port scanner*? É possível identificar quais as técnicas utilizadas? Ocorreu uma invasão ou foi uma ação resultante da atividade de um *Malware* automatizado, ou seja, um novo *worm*?

Em um contexto em que tenha ocorrido um caso de invasão: Por meio de qual serviço? Qual vulnerabilidade foi explorada? A ameaça constituída foi a partir de uma vulnerabilidade do programa utilizado ou por uma falha operacional? O acesso inicial foi a partir de um usuário comum ou de um usuário de Sistema? Ocorreu escala de privilégio? Ou foi uma invasão por meio de uma falha de segurança que possibilitou nível de administrador? Após ganhar acesso ao sistema, o invasor instalou algum *backdoor* ou mesmo um *rootkit*? É possível recuperar os códigos fonte ou mesmo os binários que o invasor possa ter instalado? Quais os artefatos recuperados? É possível identificar alguma ação de tentativa de eliminação de rastros (*cleanlogs*), ou mesmo o uso de alguma técnica anti-forense?

Dependendo do contexto do Incidente de Segurança muitas das ações cometidas poderão ser reconstruídas com riqueza de detalhes.

2.9 FORENSE POST MORTEM

A Forense Post Mortem é a etapa da Perícia onde o alvo principal são as imagens das mídias. Segundo Machado; Eleutério (2011), a manipulação dos dados contidos em mídias de armazenamento computacional deve ser realizada com toda atenção possível, pois a prova não pode ter seu estado inicial alterada, ou seja, nenhum bit pode ser modificado. Isso garante a validade da prova em juízo. Para se ter uma ideia da sensibilidade das evidencias digitais, apenas ao ligar um computador e aguardar seu sistema operacional ser inicializado, dados contidos no disco rígido já são alterados, mesmo sem qualquer intervenção direta do usuário. Portanto o primeiro passo e cuidado que se deve ter nesta etapa é não utilizar o disco rígido da máquina envolvida no Incidente de Segurança para que seu estado inicial não seja alterado. O correto é criar uma cópia bloco a bloco, ou seja, fiel e idêntica do disco rígido investigado e fazer toda a análise nela, buscando informações qualificáveis como vestígios, evidências e potenciais provas.

Existem grandes desafios no momento da geração da imagem do disco rígido, um deles é o fato de haver possibilidade da presença de grande volume de informações no disco, uma vez que, a cada dia, discos com maior capacidade são colocados no mercado.

Segundo Melo (2009, p. 56):

> A Forense Post Mortem é a etapa mais demorada de uma Perícia Forense Computacional, pois em sua essência ela é a consolidação do cruzamento de tudo que foi colhido na Forense In Vivo com o que é identificado durante a análise do disco e com possíveis informações adquiridas nos demais ativos de redes durante a Forense de Rede.

2.10 FERRAMENTAS FORENSES

Para encontrar vestígios, evidências ou alguma prova de invasão em sistemas computacionais, é necessário o uso de *softwares* específicos tais como: analisadores de discos,

analisadores de pacotes, ferramentas de clonagem, analisadores de *logs*, *scanners*, entre outros.

De acordo com Melo (2009), o mercado de TI oferece equipamentos e softwares especializados para Computação Forense. Todavia, no universo do software livre, encontram-se ótimas opções de ferramentas para realização de Perícia Forense Computacional.

Segundo Ponpolin (2011), na maioria das vezes, esses softwares estão presentes no próprio sistema operacional. Com eles podem ser realizados diversos procedimentos tais como: análise de memórias que contenham instruções de endereços em hexadecimal e de interrupções de processos (*dumpers*), análise dos códigos dos programas (*debugers*) ou *logs* de arquivos de sistemas, e as mais variadas partes que compõem um sistema operacional, seja ele Linux, Unix, ou Windows.

> Há um grande número de projetos disponibilizados em Live CDs, 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.

De acordo com Pereira (2007), a correta aplicação das diversas ferramentas e técnicas disponíveis pode reduzir muito a quantidade de dados que necessitam de um exame minucioso.

Nota-se que existe uma grande variedade de ferramentas forenses tanto para coleta, como para análise dos dados em sistemas computacionais. É importante que o profissional de segurança tenha conhecimento sobre algumas delas e saiba como utilizá-las, pois em um incidente de segurança, elas podem ser bastante úteis.

3 METODOLOGIA

Nessa monografia realiza-se um estudo descritivo-exploratório, com a finalidade de descrever e gerar conhecimento sobre como realizar uma análise forense usando o sistema operacional Ubuntu em incidentes com servidores GNU/Linux, com enfoque para o seu uso nas Forenses In Vivo e Post Mortem.

Iniciou-se com uma pesquisa bibliográfica 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.

Para realizar a parte prática utilizaram-se duas máquinas virtuais e dois Pen-Drives. Em uma das máquinas virtuais, é instalado o Sistema Operacional Debian versão 6.0.3 e o apache, onde via SSH, a página principal é modificada em uma ação chamada *Defacement* ou desfiguração - técnica antiga também conhecida como pichação. Em um dos Pen-Drives, são copiadas/adicionadas algumas ferramentas do Sistema Operacional Ubuntu 11.10, para coletar as evidências, conforme a Ordem de Volatilidade, enquanto esta permanece ligada (Forense In Vivo). Já no outro, é instalado Sistema Operacional Ubuntu versão 11.10, o qual é utilizado para fazer a cópia do disco após esta ter sido desligada (Forense Post Mortem). Também é instalado o mesmo sistema na outra máquina virtual (estação forense), a qual é utilizada para fazer a análise das evidências coletadas, tanto nas Forenses In Vivo e Post Mortem, utilizando ferramentas nativas do próprio sistema e outras de terceiros.

O software utilizado para virtualizar as máquinas é o VirtualBox, instalado em um sistema operacional Windows 7. O ambiente virtual é escolhido devido à facilidade de coletar as telas para a demonstração deste trabalho.

Optou-se por usar as ferramentas nativas do Sistema Operacional Ubuntu 11.10 devido estarem na versão mais recente, diferentemente dos Live CDs, que nem sempre estão. Os dois são importantes, mas em cenários diferentes. Os Live CDs são mais indicados para procedimentos de coleta remota e só. A perícia e a análise das evidências são mais adequadas em estações periciais, devido à possibilidade de utilização do sistema operacional e de aplicativos atualizados e estáveis, além da possibilidade de aumento de hardware e software.

4 ANÁLISE DE RESULTADOS

Este capítulo descreverá como age um Perito Computacional Forense em uma perícia num servidor web de uma empresa, que sofreu um ataque simulado via SSH, onde a página principal foi modificada, e como são aplicadas as ferramentas nativas do Ubuntu e outras de terceiros, enquanto a máquina está ligada (Forenses In Vivo) e após o seu desligamento (Post Mortem). Em seguida será descrito como elaborar o Laudo Pericial.

O desafio das empresas, grandes ou pequenas, públicas ou privadas, é o mesmo: lidar de forma eficiente com os constantes incidentes de segurança, violações de políticas de uso e com o aumento expressivo de crimes e fraudes relacionados a computadores e redes. Para tanto, elas precisam dos serviços de um Perito Computacional Forense.

O que a empresa deve fazer logo após perceber que o seu servidor foi invadido? Depende da situação. Alguns peritos defendem que se deve desconectar imediatamente o cabo de rede, a não ser que haja algum motivo para não fazer isso, como é o caso de empresas que disponibilizam algum serviço que em hipótese alguma pode ser parado, sob a pena de serem multadas. O problema de manter o cabo conectado é que o ataque pode se perpetuar e o atacante dê um jeito de remover as evidências. Outros peritos são contra esse procedimento, pois dizem que o atacante pode deixar algum artefato na máquina que fique monitorando o estado da rede, e quando percebam que a rede foi desconectada produza alguma ação danosa que é justamente o que está se tentando evitar retirando o cabo de rede. A escolha entre retirar ou não o cabo de rede, como dito anteriormente, vai depender muito da situação ou da metodologia adotada pelo Perito.

É muito importante que se mantenha a máquina ligada para que seja possível coletar evidências que estão na memória do computador, já que tudo que é executado na máquina primeiro vai para a memória para depois ser processada, então posso ter dentro da memória comandos, códigos utilizados e injetados que o invasor utilizou, pois uma hora eles rodaram e tiveram que ir para a memória para serem executados.

Portanto é muito importante a preservação da memória e não mexer na máquina, nem mesmo fazer *login*, pois isso pode gerar *logs* e acabar apagando ou sobrescrevendo algum dado importante que um atacante deixou para trás e está na memória.

O principal problema ao se coletar dados em uma situação em que a máquina está ligada (Forense In Vivo) é que nunca se tem certeza do grau de comprometimento da mesma,

que pode variar de uma simples invasão a um ataque complexo, com modificação de arquivos e programas críticos, como do próprio sistema operacional.

Uma prática comum dos invasores é instalarem *rootkits* em ambientes comprometidos - suítes de programas com versões modificadas de binários do sistema como "ls" e "ps" para ocultar a exibição de arquivos e processos de bibliotecas comuns compartilhadas por programas. Uma das formas de identificar facilmente estas questões é montando o sistema comprometido numa partição separada e somente leitura em cima de um ambiente limpo e controlado. A análise nunca deve ser feita dentro do sistema comprometido para que se garanta a integridade e a confiabilidade das respostas em um tribunal.

4.1 Aplicação de ferramentas na Forense In Vivo

O horário de início da perícia deve ser registrado assim que o Perito chegar ao local em que ocorreu o incidente de segurança. Em seguida é necessário colher e anotar algumas informações relevantes, como por exemplo: Do que se trata o incidente? O dia e a hora em que foi percebido o incidente? Quem foi a última pessoa que utilizou a máquina? Entre outras. Após isso, deve-se verificar o estado da máquina, que nesta prática, encontra-se ligada e com o cabo de rede conectado, o que possibilita monitorar o tráfego de entrada e saída. Isso ajuda a identificar eventuais atividades maliciosas ocorrendo naquele exato momento. Para isso a estação forense, que deve ser conectada a mesma rede e usada alguma ferramenta para capturar todo o tráfego originado ou destinado a máquina invadida, que nesse caso foi o tcpdump, conforme comando abaixo:

```
root@caine:~# tcpdump -i eth0 -s 1518 -w ./trafego.dump src or dst host 192.168.2.1
```

O comando acima, executado na estação forense, faz a captura via rede de todos os pacotes originados ou destinados ao servidor comprometido, cujo IP é 192.168.2.1, escrevendo o resultado da captura no arquivo trafego.dump para posterior análise, que pode ser com o próprio tcpdump utilizando o parâmetro -r ou com outras ferramentas de análise de tráfego, como o popular wireshark. Neste caso a ferramenta utilizada para ler o arquivo é o tcpdump e seu conteúdo é exibido na Figura 9.

Figura 9 - Lendo pacotes com a ferramenta tcpdump

```
root@caine:~# tcpdump -r trafego.dump
reading from file trafego.dump, link-type EN10MB (Ethernet)
16:18:24.907599 ARP, Request who-has 192.168.2.1 tell 192.168.2.14, length 46
```

Fonte: O autor.

Analisando a Figura 9 o que se vê são apenas respostas a requisições a máquina.

Após essa análise inicial o próximo passo ao entrar em contato com a máquina atacada é logar-se como root e montar algum dispositivo externo, que pode ser um *Pen-Drive* ou um HD Externo, que será usado para salvar tudo que será capturado. O Perito também pode optar em enviar estes dados via rede para a estação forense. A Figura 10 ilustra o uso da ferramenta dmesg, que serve para descobrir como os dispositivos externos são reconhecidos pelo sistema. Algumas linhas foram omitidas para facilitar a visualização.

Figura 10 - Obtendo informações do *Pen-Drive* com o comando dmesg

```
[ 1071.381165] scsi 3:0:0:0: Direct-Access     SanDisk  Cruzer           8.01 PQ: 0 ANSI: 0 CCS
[ 1071.388177] sd 3:0:0:0: Attached scsi generic sg2 type 0
[ 1071.401183] sd 3:0:0:0: [sdb] 15731711 512-byte logical blocks: (8.05 GB/7.50 GiB)
```

Fonte: O autor

De acordo com a Figura 10 o dispositivo externo foi reconhecido como sdb. Em seguida a ferramenta mount é usada para montá-la (tornar acessível no sistema), conforme o comando abaixo:

```
root@srv-web:~# mount /dev/sdb1 /mnt/
```

Após a preparação do ambiente, é importantíssimo que a coleta das informações do sistema comprometido se dê seguindo-se a ordem de volatilidade dos mesmos para que não se perca nenhuma evidência. De acordo com a Figura 9 é a memória, mas antes é necessário saber qual a versão o *kernel* do sistema para saber se o mesmo possui restrição de acesso a memória. Para isso utiliza-se o comando uname junto com o parâmetro -a, conforme ilustra a Figura 11.

Figura 11 - Obtendo informações da versão do *kernel*

```
root@srv-web:~# uname -a
Linux srv-web 2.6.32-5-686 #1 SMP Thu Nov 3 04:23:54 UTC 2011 i686 GNU/Linux
```

Fonte: O autor

A versão do *kernel* ilustrada na Figura 11 possui restrição de acesso a memória. Para conseguir fazer o *dump* da memória é necessário usar o módulo fmem, que deve ser baixado e compilado na máquina invadida para retirar à restrição de acesso a memória do *kernel*, os passos necessários são descritos abaixo:

1. root@srv-web:~# wget http://hysteria.sk/~niekt0/foriana/foriana_current.tgz
2. root@srv-web:~# tar -vzxf foriana_current.tgz
3. root@srv-web:~# cd fmem_1.6-1/
4. root@srv-web:~/fmem_1.6-1# make
5. root@srv-web:~/fmem_1.6-1# ./run.sh

O comando wget acima faz o download do programa no endereço especificado. Como o arquivo está compactado, pode-se usar o comando tar com os parâmetros -vzxf para poder descompactá-lo. O comando cd é utilizado para adentrar o diretório /fmen_1.6-1/ onde se encontra o módulo, a execução dos arquivos make e ./run.sh realizam o processo de compilação. Após estes procedimentos consegue-se realizar o *dump* da memória com o comando abaixo:

root@srv-web:~# dd if=/dev/fmem of=/mnt/forense/memoria.dd bs=1M count=384

Os comandos anteriores são necessários apenas em sistemas com versões de *kernel* 2.6.27 ou posteriores. Em versões inferiores basta apenas utilizar o comando abaixo:

root@srv-web:~# dd if=/dev/mem of=/mnt/forense/memoria.dd

Para que o ambiente de comando esteja minimamente protegido da influência externa, pode-se usar uma nova *shell* confiável, que pode estar no dispositivo externo ou numa mídia de CD. Neste caso a ferramenta encontra-se no diretório /mnt/ferramentas e para usá-la basta utilizar os comandos abaixo:

root@srv-web:~# cd /mnt/ferramentas/
root@srv-web: /mnt/ferramentas# ./bash --norc
bash-4.2#

O parâmetro "--norc" acima evita que a *shell* execute o arquivo "~/.bashrc", que pode conter instruções maliciosas. Caso fosse uma *shell* de *login*, teria que passar também o parâmetro "--noprofile" para evitar a execução dos arquivos "/etc/profile" e "~/bash_profile".

Outro elemento mais volátil residente na memória é o estado da rede. Por isso, as tabelas de *cache* representando esses estados são as primeiras a serem obtidas. Portanto deve-se começar pela tabela do Protocolo de Resolução de Endereços (ARP), que aponta as associações entre endereços IP do segmento da rede que se conectaram recentemente a máquina comprometida e os endereços MAC das respectivas placas de redes. Os parâmetros usados para a ferramenta arp são: "-n", para que ele mostre endereços em formato numérico, sem tentativa de resolução e "-a" para exibir todas as entradas existentes na tabela. O comando abaixo mostra seu uso:

```
bash-4.2# ./arp -na > /mnt/forense/tabela_arp
```

Depois se repete o processo com a tabela de rotas IP do sistema, contendo indicações de rotas a serem seguidas para alcançar *hosts* e redes específicas. A ferramenta utilizada é o route, conforme abaixo:

```
bash-4.2# ./route -Cnee > /mnt/forense/tabela_rotas
```

A opção "-C" opera diretamente no *cache* do kernel, o "-n" novamente mantém os resultados numéricos, enquanto o "-ee" mostra todos os parâmetros da tabela de rotas.

Os valores contidos nessas tabelas podem acusar rotas maliciosas, conexões recentes não autorizadas do sistema comprometido com outras máquinas do segmento ou mesmo ataques mais elaborados de ARP *Spoofing* ou *Cache Poisoning*, em que o atacante manipula requisições ARP para fingir ser outra máquina na rede.

A preocupação depois é com conexões ativas e portas abertas no sistema, saber que processos estão escutando. O netstat é uma poderosa ferramenta para essa tarefa, com diversas opções de filtros de exibição. O comando abaixo mostra seu uso:

```
bash-4.2# ./netstat -tupan > /mnt/forense/conexoes
```

O parâmetro "-n" é para exibição de endereços numéricos sem resoluções. "-t" e "-u" para mostrar respectivamente conexões TCP e UDP. "-p" para exibir o PID (Program ID) e

nome do processo a que o *socket* encontrado pertence e "-a" para listar todos os *socket* abertos. Uma maneira fácil de lembrar-se dos mesmos é associa-los ao poderoso Deus Tupi/Guarani com a combinação conjunta "-tupan" que agrega todos os parâmetros acima.

Por já ter capturado os dados das tabelas de rede do sistema e monitorado o tráfego, a próxima ferramenta a ser utilizada é o unhide-tcp, sua função é varrer as portas TCP/UDP e conferir se existem portas ocultas. O comando abaixo mostra seu uso:

```
bash-4.2# ./unhide-tcp > /mnt/forense/conexoes_ocultas
```

Outra ação muito comum de invasores é instalar *sniffers* para coletar todo o tráfego da rede em busca de senhas ou informações relevantes, em processo semelhante ao que foi feito no início da análise para saber quem se comunicava com a máquina invadida. Porém, para o funcionamento efetivo do processo, a interface de rede precisa ser colocada em modo promíscuo, observando e registrando tudo que passa pelo cabo e não apenas o que tem seu endereço como destino. A primeira coisa é saber se as interfaces foram maliciosamente configuradas em modo promíscuo. Para conferir pode-se usar a ferramenta ifconfig, conforme abaixo:

```
bash-4.2# ./ifconfig eth0 > /mnt/forense/interface
```

A análise dos processos em execução no ambiente é de grande importância, pois indica seu comportamento naquele momento. Como mencionado anteriormente, é possível que um invasor desvirtue a execução de um processo para garantir ou aumentar seus privilégios no sistema. Se os processos não estiverem funcionando corretamente, o sistema também não estará.

Existem várias formas de se obter informações relacionadas aos processos em execução, como: comando executado, arquivos abertos, consumo de recursos, entre outras, que revelam evidencias de atividades não autorizadas. A ferramenta ps junto com parâmetros "auxww", por exemplo, é uma maneira de obter a listagem completa dos processos em execução no sistema. O comando abaixo mostra seu uso:

```
bash-4.2# ps auxww > /mnt/forense/processos
```

Os parâmetros “a” e “x”, quando usados em conjunto, fazem o comando exibir todos os processos. Já o “u” formata a tabela de saída agrupada por usuário. O “w” por sua vez garante que a saída do comando não será truncada devido à largura do terminal, sendo utilizado duas vezes “ww” para considerar largura limitada.

Um erro comum entre administradores de sistemas é utilizar, no ps que vem com a maioria das distribuições Linux, a sintaxe de parâmetros do UNIX, colocando um hífen (-) antes dos parâmetros “auxww”. Nesse caso, o parâmetro “u” para exibição em formato orientado a usuários, seria transformado em “-u”, exibindo apenas os processos do usuário que vem a seguir (o possivelmente inexistente usuário “xww”) e dos processos definidos pelo “-a” também diferente do “a” sem hífen. Embora o ps exiba um aviso de erro e assuma que o usuário digitou um hífen a mais caso não encontre tal usuário, convém evitar esse tipo de vício.

Uma forma de se obter informações sobre os processos que estão ocultos é usando a ferramenta unhide. Os comandos abaixo mostram seu uso:

```
bash-4.2# ./unhide-linux26 proc > /mnt/forense/processos_ocultos.proc
bash-4.2# ./unhide-linux26 sys > /mnt/forense/processos_ocultos.sys
bash-4.2# ./unhide-linux26 brute > /mnt/forense/processos_ocultos.brute
```

Outro dado importante que deve ser capturado são as informações como data e hora do *login* dos usuários. Para saber qual a data e hora que os usuários fizeram *login* na máquina basta usar a ferramenta lastlog, conforme abaixo:

```
bash-4.2# ./lastlog > /mnt/forense/lastlog
```

Para obter informações como: a quantidade memória usada e quanto tempo à máquina está ligada, deve-se utilizar as ferramentas free e uptime, conforme abaixo:

```
bash-4.2# ./free -m > /mnt/forense/info_memoria
bash-4.2# ./uptime > /mnt/forense/uptime
```

Sabendo quanto tempo esta máquina está ligada é possível descobrir se a mesma foi reiniciada pelo atacante.

A relação de pacotes instalados no Debian é possível obter com a ferramenta dpkg, conforme abaixo:

bash-4.2# COLUMNS=110 ./dpkg -l > /mnt/forense/pacotes

O COLUMNS é para aumentar a largura da coluna, pois o Debian mostra o nome do pacote cortado em determinado ponto, com ele a informação aparece completa.

Outros dados muito importantes são a data e a hora do sistema, a ferramenta date obtém essa informação, conforme abaixo:

bash-4.2# ./date > /mnt/forense/date

Se no momento da captura houver diferenças da hora do relógio, isso deverá constar no Laudo Pericial.

Uma série de informações do disco como: utilização dos discos, detalhes sobre dispositivos montados e o esquema de particionamento, podem ser capturados, usando-se as ferramentas df, mount e fdisk, conforme abaixo:

bash-4.2# ./df -hT > /mnt/forense/utilizacao_discos
bash-4.2# mount > /mnt/forense/dispositivos_montados
bash-4.2# ./fdisk -l > /mnt/forense/particionamento_discos

O parâmetro T, descrito na primeira linha dos comandos anteriores, vai acrescentar a informação do tipo de *filesystem* que a máquina está usando.

Por último podem ser obtidas as informações dos módulos de *kernel*, com a ferramenta lsmod, conforme abaixo:

bash-4.2# ./lsmod > /mnt/forense/modulos_kernel

Terminado essa fase de captura deve-se verificar se realmente todos os arquivos foram gravados no dispositivo externo e calcular o *hash* de todos eles para garantir que os mesmos permanecerão íntegros durante a fase de análise, conforme ilustra a Figura 12.

Figura 12 – Gerando o *hash* dos dados capturados

```
bash-4.2# ./sha256sum /mnt/forense/*
5cc2a6b2c196ff025e7299372ed4b4456eee201452eb69a09fc78de64e891c0f  /mnt/forense/conexoes
1e7a4886354e807241ebd10fd9cfbad3f45c9c4fb3153696418b8818a76ff86d  /mnt/forense/conexoes_ocultas
8f6611e52623096c7d95996b84e92aea2f9c67d0cce699b29fda8d105e067d93  /mnt/forense/date
2df870427a9533c30b1b082720dd71d046b98f85994fd529a611658a0d420c99  /mnt/forense/dispositivos_montados
e3c73fdcf3e84e460443147ea30818a44573ed46b1f859a155f339e4f46703a4  /mnt/forense/historico_comandos
9b9abf407ee3b51e2e9cf7db61022f37d0e24916af640517ff874161e04a1946  /mnt/forense/info_memoria
85f6f8adeef169aef0d3952ce1545801eb34fa5d16c2d474dd1665fc90f4b968  /mnt/forense/interface
bdd6b48b979b5027439eb86d67ea127da9304ea32172afbb8342bec70df30bb3  /mnt/forense/memoria.dd
b78c4486e51dd6614b10272cfc0c439d1d97d8c8173a2af5406fb2f877bcc5bc  /mnt/forense/modulos_kernel
1d2f525c291e2ee2f7c7251b67f7fc9ad8fa5c18cbbf0a11e03ab818da33c3c1  /mnt/forense/pacotes
8e6c9c8d6f2ab1ccf7d79c9c1df2f518da88451dfd6de553021ae790999514d5  /mnt/forense/particionamento_discos
d422a311184ffdc0d6c8846c7d832c9143185a57751c797210d1c7796b606423  /mnt/forense/processos
80afc9c6db660191e8ca49c5b4b551c77acec256f55763c597a1c47831c8b535  /mnt/forense/processos_ocultos.brute
1bb1b6a2dc1b4a07fd1e49dddcfc4a80d6b1e2d1942e4bb4b8a4bf2bfdf118f8  /mnt/forense/processos_ocultos.proc
660be52f11e04db56f3b1dfc959ef9de67c9eb1be47a504202762f210f533802  /mnt/forense/processos_ocultos.sys
c9e511cd87227b34ed8f358b271e66e650732098410c9be2988def5db0239621  /mnt/forense/tabela_arp
b00c4d7e6b3fb0f4bebcfd8e30a1abdda5f00cacd6fb42a787af73750a939998  /mnt/forense/tabela_rotas
f34b1677b63ccadb5549dcb10263b0c1365e87e8290f15cf51ff64213dd5beea  /mnt/forense/uptime
85f1ee49173d71ad54f6be983424e9b20cfc7330a2de9f7999044994caf6a3b2  /mnt/forense/usuarios
adc0d4196563b9582be49add96d2819c3c8a782adc208a2ebc302f39b4bd75d6  /mnt/forense/utilizacao_discos
```

Fonte: O autor

Os *hashes* são algoritmos faz um cálculo matemático em cima de um determinado conteúdo de arquivos. Ele produz quantidade fixa de *bytes* como resultado em hexadecimal. O SHA1sum, gera um *hash* de 160 *bits* (20 *bytes*), que convertidos para hexadecimal dão 40 caracteres, e convertidos para base-64 dão 28 caracteres. Já o SHA256sum gera um *hash* de 256 *bits* (32 bytes), que convertidos para hexadecimal dão 64 caracteres, e convertidos para base-64 dão 44 caracteres.

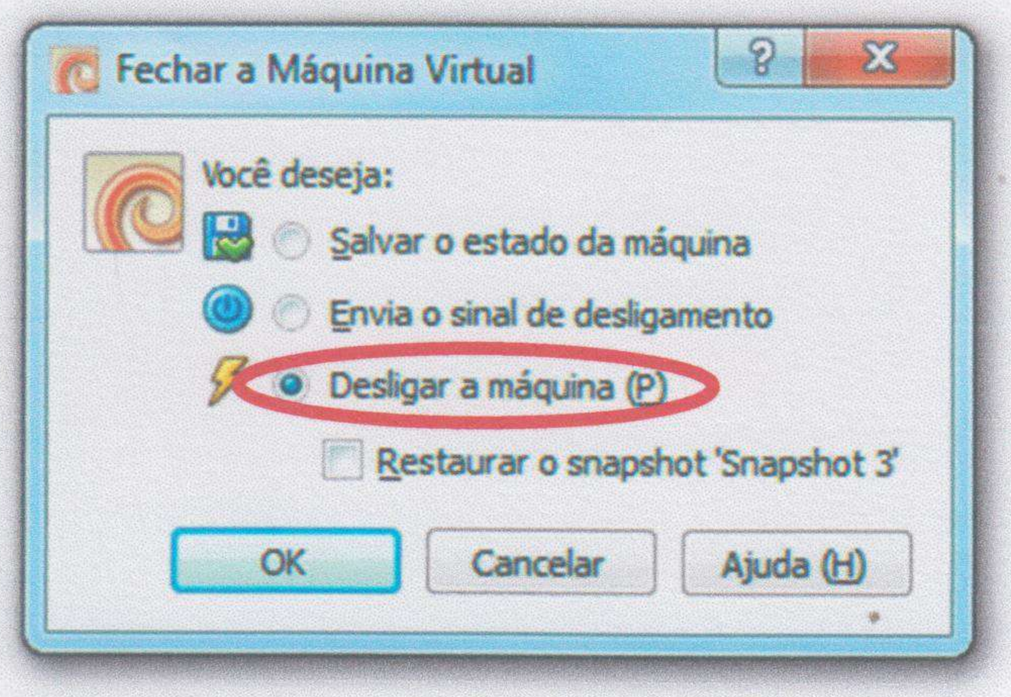
Hash só trabalha com conteúdo de arquivo, se eu mudar um *bit* no conteúdo de um arquivo, algum caractere, por exemplo, o *hash* tem que mudar. Não existem dois conteúdos diferentes que produzam o mesmo *hash*. Quando isso ocorre se diz que houve uma colisão, que ele foi quebrado. O SHA1sum e o MD5sum já foram quebrados e deve-se evitar utilizá-los.

Gerados os *hashes*, deve-se desmontar o dispositivo externo usando a ferramenta umount, conforme abaixo:

```
bash-4.2# exit
root@srv-web:/mnt/ferramentas# cd
root@srv-web:~# umount /mnt
```

Feito isso a máquina pode ser desligada, mas com o cuidado de não deixar que a mesma grave dados no disco. Para isso deve ser desligada retirando-se o cabo de força da energia elétrica. Como nesta prática está se usando máquinas virtuais deve-se desligar conforme o procedimento da Figura 13.

Figura 13 - Desligando a máquina periciada no VirtualBox



Fonte: O autor

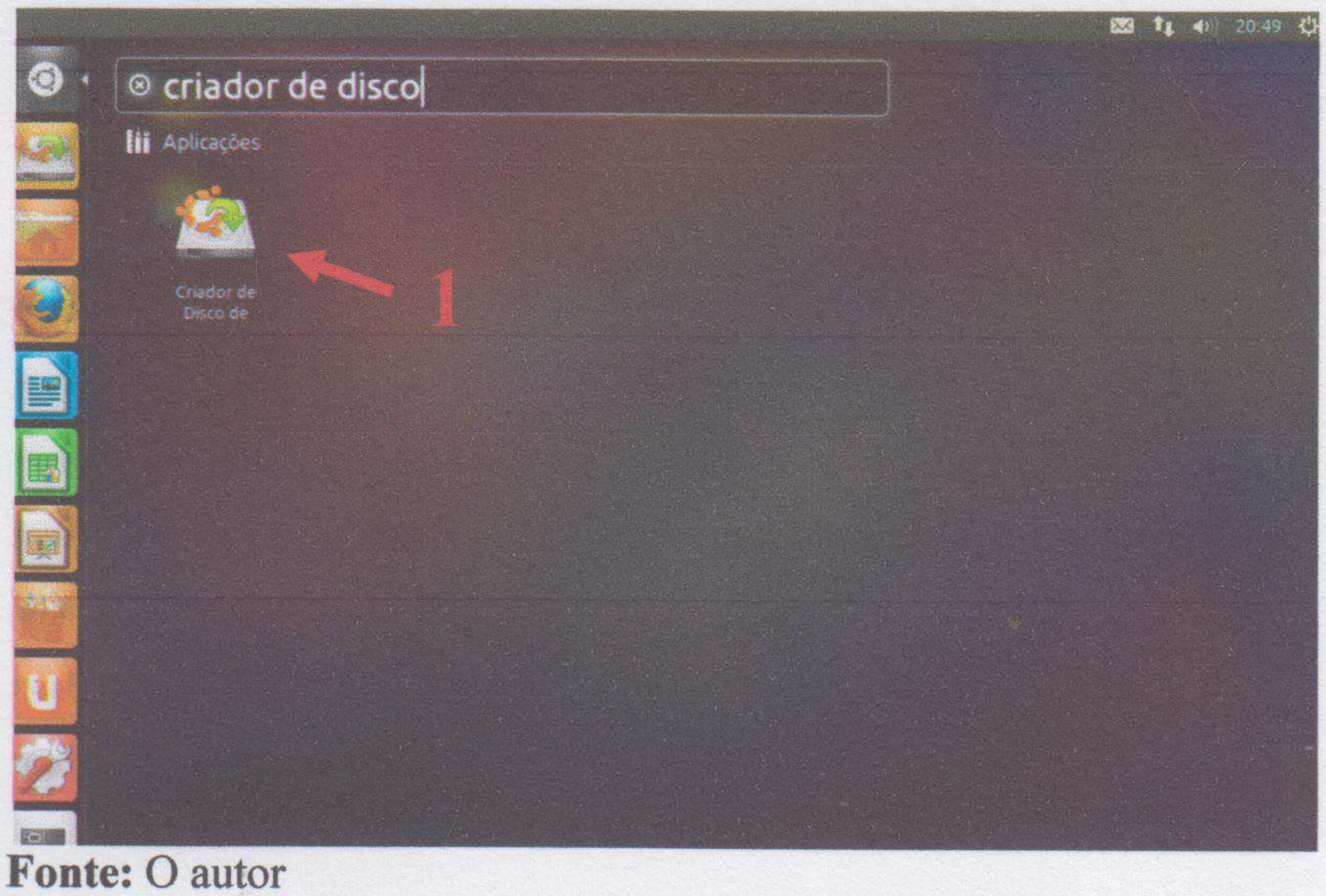
Todos os procedimentos descritos até o momento corresponde a Forense In Vivo, ou seja, só servem para as máquinas atacadas que permaneceram ligadas, as etapas seguintes serão mostrados os procedimentos da Forense Post Mortem, quando a máquina foi desligada.

4.2 Aplicação de ferramentas na Forense Post Mortem

É nesta etapa que se faz a cópia bloco a bloco do disco da máquina investigada. Para isso pode-se usar qualquer sistema operacional bootável, de preferência que seja alguma distribuição forense. Deve-se evitar o uso de Live CDs, pois estes utilizam as áreas de swap do disco e montam mídias automaticamente.

Para fazer a coleta optou-se por usar o sistema operacional Ubuntu versão 11.10, instalado em um *Pen-Drive*. Os três passos necessários para instalar o Ubuntu no *Pen-Drive* são: abrir o programa "Criador de discos de inicialização" presente no próprio sistema Ubuntu, conforme ilustra a Figura 14.

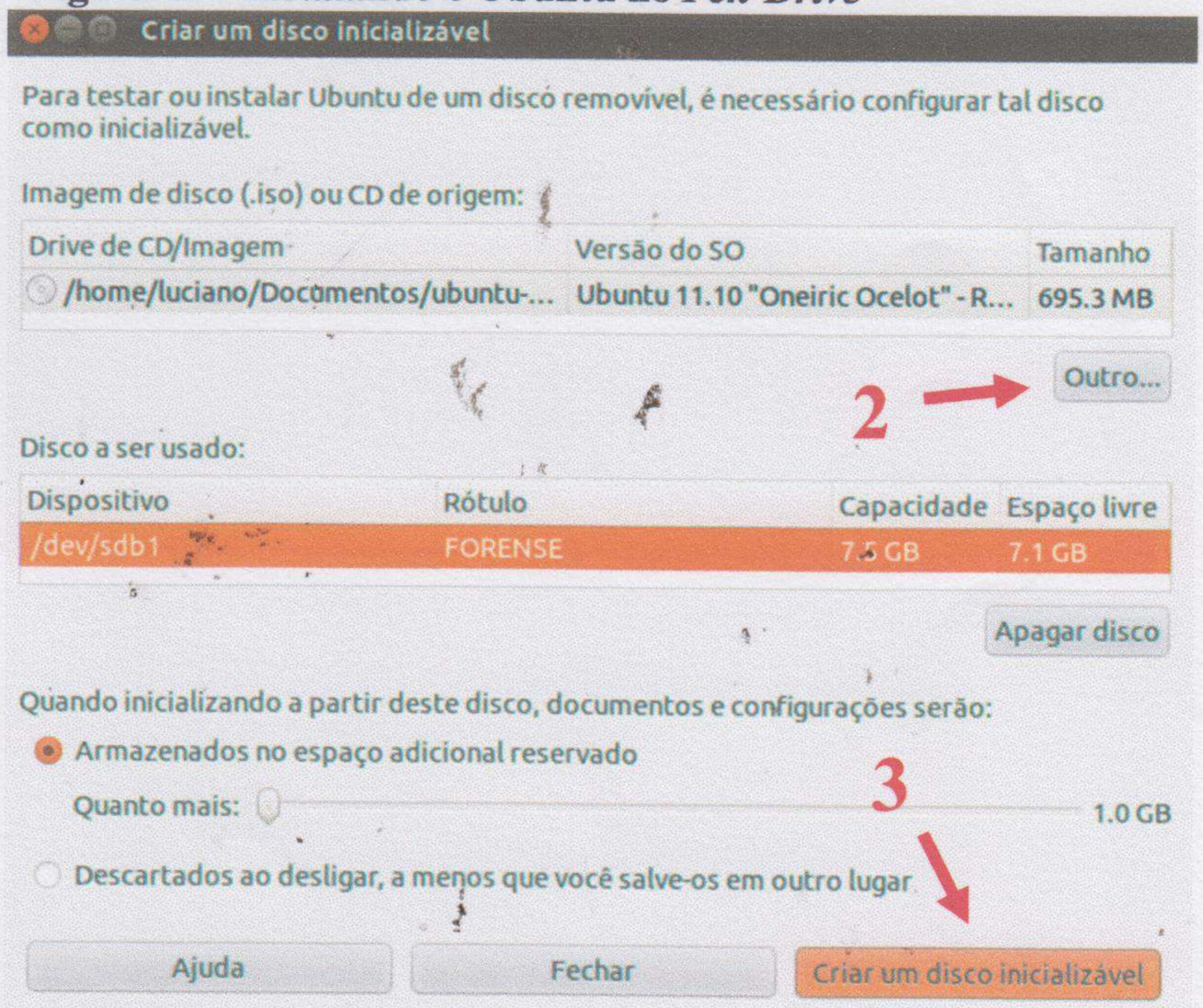
Figura 14 - Abrindo o programa criador de discos de inicialização no Ubuntu



Fonte: O autor

Ao abrir o programa irá aparecer uma janela como a ilustrada na Figura 15.

Figura 15 - Instalando o Ubuntu no *Pen-Drive*

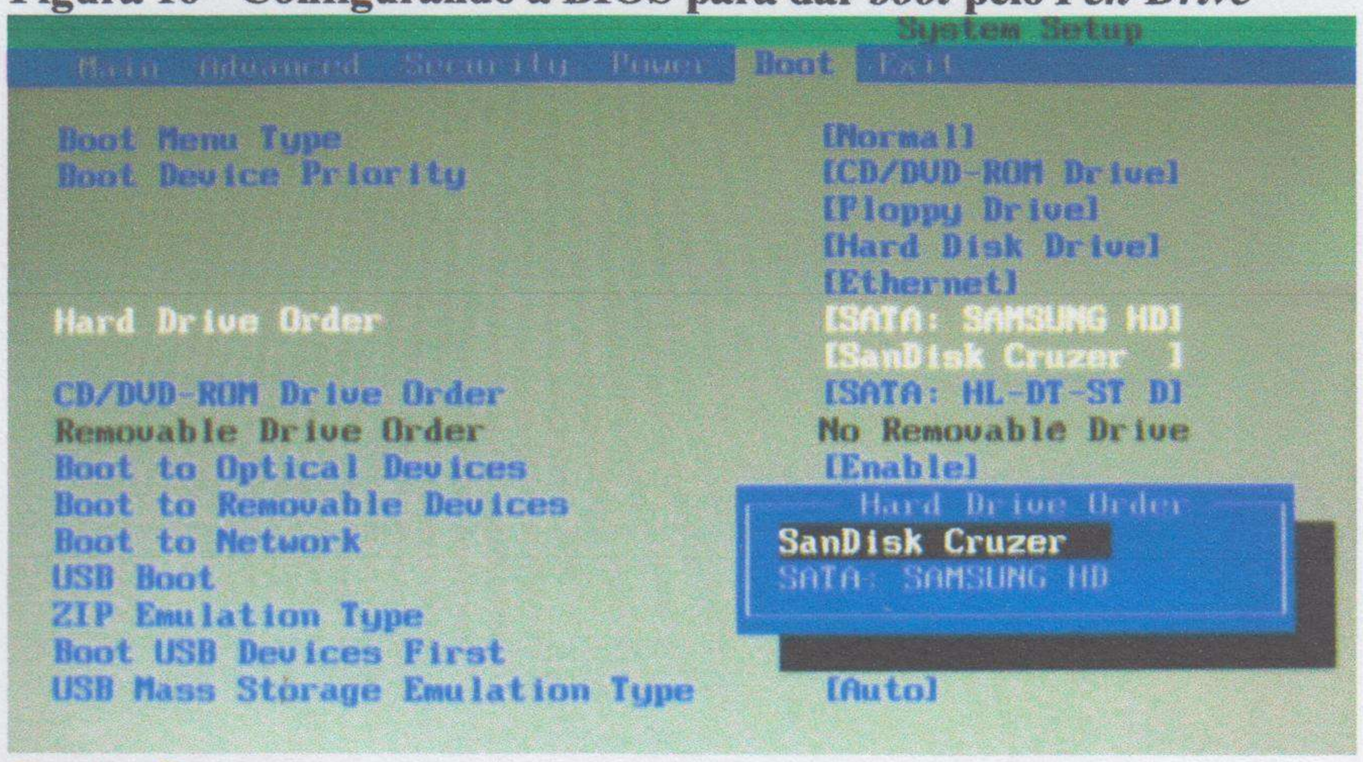


Fonte: O autor

O segundo passo é clicar em "Outro" e selecionar a imagem do Ubuntu, e o terceiro passo clicar em "Criar um disco inicializável". Após alguns minutos surgirá uma janela informando que o processo foi concluído com sucesso.

Antes de ligar a máquina invadida, a BIOS deve ser configurada para dar *boot* no *Pen-Drive*, conforme ilustra Figura 16.

Figura 16 - Configurando a BIOS para dar *boot* pelo *Pen-Drive*

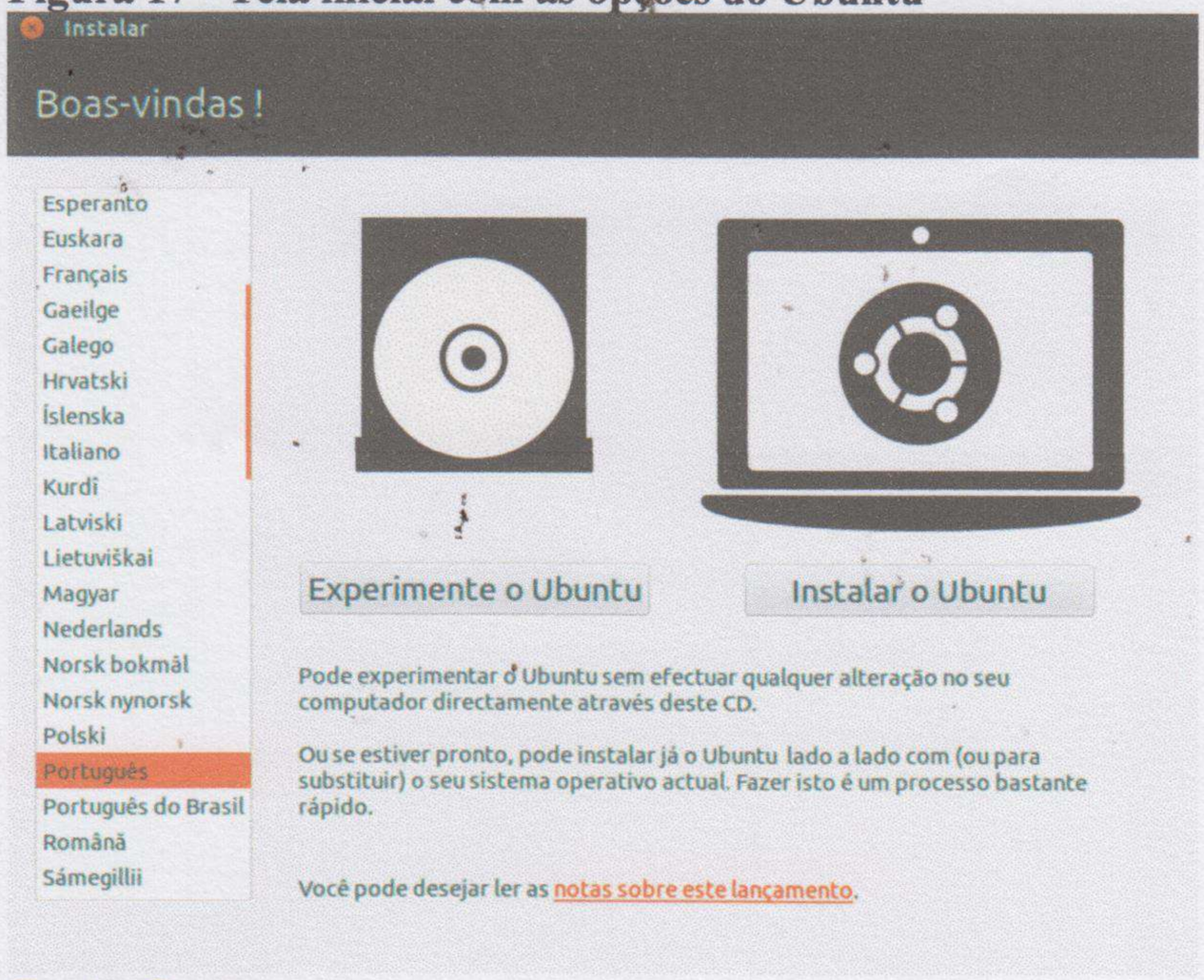


Fonte: O autor

Para saber como acessar as configurações da BIOS do computador basta consultar o manual do fabricante da placa mãe. Na maioria dos casos utiliza-se a tecla *delete* ou F2 do teclado.

Após salvar as configurações da BIOS a máquina será reinicializada e aparecerá uma tela como a ilustrada na Figura 17.

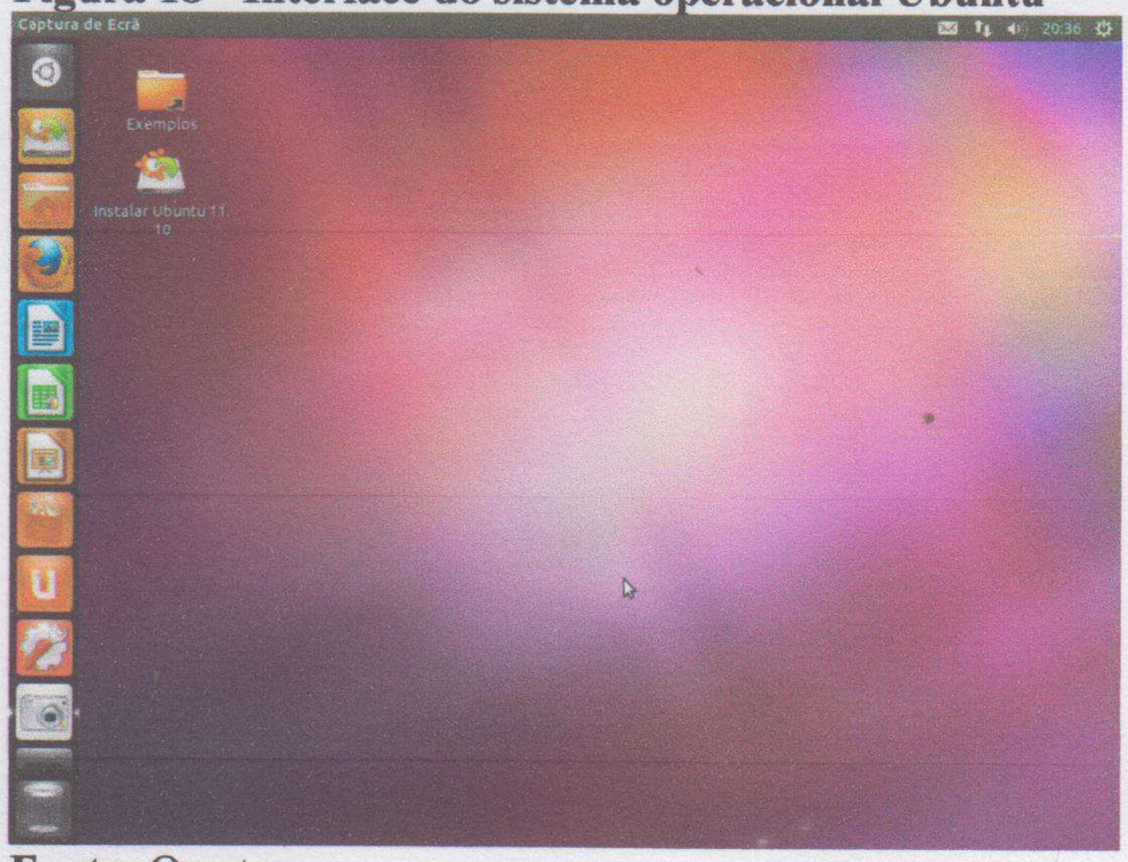
Figura 17 - Tela inicial com as opções do Ubuntu



Fonte: O autor

Deve-se escolher o idioma de sua preferência e em seguida clicar em "Experimente o Ubuntu" para que o sistema seja inicializado como Live, conforme Figura 18.

Figura 18 - Interface do sistema operacional Ubuntu



Fonte: O autor

Agora o Perito deve fazer uma cópia integral bloco a bloco do HD da máquina a qual está trabalhando para posteriormente trabalhar em cima dela. Para isso deve ser utilizado novamente algum dispositivo externo com capacidade superior ao HD da máquina comprometida. Não é recomendado utilizar um HD de mesmo tamanho de outra marca, pois nem sempre vai ser do mesmo tamanho e vai dar problema no processo da cópia.

A primeira coisa a ser feita é montar o dispositivo externo que irá receber a cópia do disco, mas antes deve-se usar a ferramenta dmesg, ilustrado na Figura 19, para descobrir como ele e o HD da máquina invadida foram reconhecidos no sistema. Algumas linhas foram omitidas para facilitar a visualização.

Figura 19 - Obtendo informações do HD Externo e da máquina invadida

```
[  929.235862] sd 3:0:0:0: [sdb] 976773168 512-byte logical blocks: (500 GB/465 GiB)
[    2.487467] sd 2:0:0:0: [sda] 16777216 512-byte logical blocks: (8.58 GB/8.00 GiB)
```

Fonte: O autor

Após o comando dmesg descobriu-se que o sistema reconheceu o dispositivo externo como sdb e o HD da máquina como sda. Com estas informações o dispositivo externo deve ser montado, conforme comando a seguir:

root@caine:~# mount /dev/sdb1 /mnt

Em seguida a ferramenta dd deve ser utilizada para criar tanto a imagem completa com todas as partições do HD da máquina invadida, como de cada partição. A Figura 41 ilustra a criação da imagem completa. Dependendo da capacidade do HD este processo pode demorar vários minutos ou até horas.

root@caine:~# dd if=/dev/sda of=/mnt/hd_full.dd

Já para se criar as imagens de cada partição primeiramente deve-se usar a ferramenta sfdisk, conforme ilustra a Figura 20.

Figura 20 - Usando o comando sfdisk para exibir a estrutura do HD periciado

```
root@caine:~# sfdisk -luS /dev/sda

Disk /dev/sda: 1044 cylinders, 255 heads, 63 sectors/track
Warning: extended partition does not start at a cylinder boundary.
DOS and Linux will interpret the contents differently.
Units = sectors of 512 bytes, counting from 0

   Device Boot    Start       End   #sectors  Id  System
/dev/sda1   *      2048  15988735   15986688  83  Linux
/dev/sda2      15990782  16775167     784386   5  Extended
/dev/sda3             0         -          0   0  Empty
/dev/sda4             0         -          0   0  Empty
/dev/sda5      15990784  16775167     784384  82  Linux swap / Solaris
```

Fonte: O autor

O comandó sfdisk, ilustrado na Figura 20, informa a estrutura do dispositivo, de acordo com ela o sda1 corresponde à raiz do sistema e o sda5 a memória *swap*. Para gerar as imagens das partições do HD basta usar os comandos abaixo;

root@caine:~# dd if=/dev/sda1 of=/mnt/hd_raiz.dd

root@caine:~# dd if=/dev/sda5 of=/mnt/hd_swap.dd

Após a geração das imagens criou-se o *hash* de cada imagem com o comando SHA256sum, conforme ilustra a Figura 21. Dependendo da capacidade do HD este processo pode demorar vários minutos ou até horas.

Figura 21 - Gerando o *hash* das imagens criadas

```
root@caine:~# sha256sum /mnt/hd_full.dd
b57a6ef1954a214d21e591767021a3f84e192141125aea6d063ff5b47cc0040d  /mnt/hd_full.d
d
root@caine:~# sha256sum /mnt/hd_raiz.dd
9cadf5925aaf2f27dde30ce411f65b61d51d4ff07b1f1c2b65d3c7ca90f2ae5f  /mnt/hd_raiz.d
d
root@caine:~# sha256sum /mnt/hd_swap.dd
671a89e8edc0bcbb78be2c44f41ec158f80388b32d05a421db2051f799bfaa9a  /mnt/hd_swap.d
d
```

Fonte: O autor

O mais importante neste momento é preservar o máximo o disco da máquina comprometida para que nada seja alterado.

Todos esses passos do Perito, desde a abertura da máquina comprometida, criação da imagem, cálculo do *hash* até a assinatura do relatório de tudo que foi realizado, o qual faz parte do Laudo Pericial, deverá ser acompanhado por no mínimo duas testemunhas, que de preferência pertençam à instituição. Tudo deve ser documentado, isso vai resguardar o perito, a Instituição, os dados, a pessoa que está sendo investigada, uma série de coisas, não fazer isso como perito pode ser que, em casos extremos, o perito vire réu em determinado momento. Outro motivo também é por que ninguém é confiável, principalmente quando se vai gerar alguma coisa para a justiça, por isso a importância de ter testemunhas, para evitar que depois de apresentar uma prova qualquer, alguém acuse o Perito de ter manipulado alguma prova para incriminar alguém.

Após a obtenção da imagem do disco e as testemunhas tendo assinado o relatório, o próximo passo é trabalhar com as imagens e com os dados coletados anteriormente na Forense In Vivo, para isso utilizou-se a estação forense - máquina virtual com o Ubuntu 11.10 instalado.

A primeira coisa a fazer é montar o dispositivo externo que contém as imagens que serão periciadas. Usando o comando dmesg, descobriu-se que este foi identificado como sdc, conforme ilustra a Figura 22. Em seguida o mesmo é montado para tornar-se acessível.

Figura 22 - Montando o HD Externo na estação forense

```
[    6.225734] sd 5:0:0:0 [sdc] 976773168 512-byte logical blocks: (500 GB/465 GiB)
```

Fonte: O autor

Os arquivos de imagem podem ser analisados diretamente (exame de superfície), ou poderão ser montadas como *loop* (por ser um arquivo) e *read only* - importante para não

alterar o conteúdo. Nesta prática são utilizadas as duas formas. O comando abaixo faz a montagem da imagem raiz do sistema como *read only*:

```
root@caine:~# mount -o loop,ro /media/sdc1/hd_raiz.dd /mnt
```

Usando-se a ferramenta mount é possível ver o dispositivo externo montado em /media/sdc1 e a imagem em /mnt como *loop*, conforme ilustra a Figura 23.

Figura 23 - Exibindo informações dos dispositivos montados

```
root@caine:~# mount
aufs on / type aufs (rw)
none on /proc type proc (rw,noexec,nosuid,nodev)
none on /sys type sysfs (rw,noexec,nosuid,nodev)
/none on /dev type devtmpfs (rw,mode=0755)
none on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=0620)
/dev/sdb1 on /media/cdrom type vfat (ro,noatime,fmask=0022,dmask=0022,codepage=c
p437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro)
/dev/loop0 on /rofs type squashfs (ro,noatime)
/none on /sys/fs/fuse/connections type fusectl (rw)
/none on /sys/kernel/debug type debugfs (rw)
/none on /sys/kernel/security type securityfs (rw)
none on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev)
none on /var/run type tmpfs (rw,nosuid,mode=0755)
none on /var/lock type tmpfs (rw,noexec,nosuid,nodev)
none on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
/dev/sdc1 on /media/sdc1 type fuseblk (rw,allow_other,blksize=4096)
/dev/loop1 on /mnt type ext3 (ro)
```

Fonte: O autor

Surgiu a seguinte dúvida, por onde começar a análise? Os dados informados no início da perícia são muito importantes nesse momento, como por exemplo, a hora e a data em que foi percebido que a página da empresa foi modificada. Como a máquina investigada sofreu um ataque do tipo *Defacement*, o atacante com toda certeza teve acesso ao diretório /var/www. Por essa razão deve ter alguma informação relevante nos *logs* contidos no diretório /var/log. Através do comando ls constatou-se que os *logs* foram apagados, conforme ilustra a Figura 24.

Figura 24 - Usando o comando ls para exibir arquivos do diretório var

```
root@caine:~# ls /mnt/var/
backups  cache  lib  local  lock  mail  opt  run  spool  tmp  www
```

Fonte: O autor

Segundo mostra a Figura 24 o *log* foi apagado, possa ser que tenha alguma evidência nos dados que foram capturados quando a máquina estava ligada e que os arquivos ainda existam no disco. Para isso é feita uma busca pela data em que ocorreu o *Defacement* na imagem da máquina invadida utilizando a ferramenta strings, conforme ilustra a Figura 25.

Figura 25 - Utilizando a ferramenta strings na imagem da máquina invadida

```
root@caine:~# strings /media/sdc1/hd_raiz.dd |grep 30/Nov/2011
192.168.2.11 - - [30/Nov/2011:18:34:43 -0300] "GET / HTTP/1.1" 200 484 "-" "Mozilla/5.0 (Windows NT 6.1
) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:34:43 -0300] "GET /favicon.ico HTTP/1.1" 404 500 "-" "Mozilla/5.0 (Win
dows NT 6.1) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:43:55 -0300] "GET /favicon.ico HTTP/1.1" 404 501 "-" "Mozilla/5.0 (Win
dows NT 6.1) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:43:57 -0300] "GET / HTTP/1.1" 304 209 "-" "Mozilla/5.0 (Windows NT 6.1
) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:43:57 -0300] "GET /favicon.ico HTTP/1.1" 404 500 "-" "Mozilla/5.0 (Win
dows NT 6.1) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:53:13 -0300] "GET /favicon.ico HTTP/1.1" 404 501 "-" "Mozilla/5.0 (Win
dows NT 6.1) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:53:15 -0300] "GET / HTTP/1.1" 200 641 "-" "Mozilla/5.0 (Windows NT 6.1
) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:53:15 -0300] "GET /pirata.jpg HTTP/1.1" 200 25973 "http://192.168.2.1/
" "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:53:15 -0300] "GET /favicon.ico HTTP/1.1" 404 500 "-" "Mozilla/5.0 (Win
dows NT 6.1) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
10.0.0.1 - - [30/Nov/2011:19:27:08 -0300] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (Windows NT 5.2; rv
:7.0.1) Gecko/20100101 Firefox/7.0.1"
```

Fonte: O autor

A busca ilustrada na Figura 25 retornou exatamente os dados dos *logs* que foram apagados. O que chama atenção nessa imagem é o arquivo chamado pirata.jpg, primeiro pelo nome, já que uma das imagens apresentadas no site modificado da empresa é um pirata, e segundo pela data e hora em que uma máquina com IP 192.168.2.11 acessou a página da empresa e carregou esta imagem. Foram justamente alguns minutos depois deste horário que foi percebido que o site havia sido modificado. A Figura 26 ilustra o código fonte da página modificada, confirmando que essa imagem faz parte do conteúdo dela.

Figura 26 - Código fonte da página web modificada

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http:/
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title></title>
</head>
<body>
<h1>Pagina pichada</h1>
<p>Informações negrejando a imagem da empresa</p>
<img src="pirata.jpg"/>
<h1>ELFOH4CK3R</h1>
</body>
```

Fonte: O autor

O passo seguinte deve ser buscar informações no arquivo de memória capturado (*dump*), por exemplo, o comando usado para apagar o arquivo de *log*. Conforme ilustra a Figura 27.

Figura 27 - Realizando uma busca por comandos com a ferramenta strings na memória

```
root@caine:~# strings /media/sdc1/forense/memoria.dd |grep rm |grep /var/log
rm -r /var/log
```

Fonte: O autor

A Figura 27 acima mostra que o atacante usou o comando rm para apagar o diretório dos *logs*. Logo após é realizada uma nova busca no mesmo arquivo, só que esta vez pela data em que ocorreu o *Defacement*. A Figura 28 exibe a saída do arquivo.

Figura 28 - Realizando uma busca pela data com a ferramenta strings na memória

```
root@caine:~# strings /media/sdc1/forense/memoria.dd |grep 30/Nov/2011
N[30/Nov/2011:18:53:15 -0300]
192.168.2.11 - - [30/Nov/2011:18:53:15 -0300] "GET /favicon.ico HTTP/1.1" 404 500 "-" "Mozilla/5.0 (Windows NT 6.1)
 AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
N[30/Nov/2011:18:53:13 -0300]
192.168.2.11 - - [30/Nov/2011:18:53:13 -0300] "GET /favicon.ico HTTP/1.1" 404 501 "-" "Mozilla/5.0 (Windows NT 6.1)
 AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:53:
N[30/Nov/2011:18:53:15 -0300]XF
N[30/Nov/2011:19:27:08 -0300]
10.0.0.1 - - [30/Nov/2011:19:27:08 -0300] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (Windows NT 5.2; rv:7.0.1) Geck
o/20100101 Firefox/7.0.1"
N[30/Nov/2011:19:27:08 -0300]
10.0.0.1 - - [30/Nov/2011:19:27:08 -0300] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (Windows NT 5.2; rv:7.0.1) Geck
o/20100101 Firefox/7.0.1"
192.168.2.11 - - [30/Nov/2011:18:34:43 -0300] "GET / HTTP/1.1" 200 484 "-" "Mozilla/5.0 (Windows NT 6.1) AppleWebKi
t/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:34:43 -0300] "GET /favicon.ico HTTP/1.1" 404 500 "-" "Mozilla/5.0 (Windows NT 6.1)
 AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:43:55 -0300] "GET /favicon.ico HTTP/1.1" 404 501 "-" "Mozilla/5.0 (Windows NT 6.1)
 AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:43:57 -0300] "GET / HTTP/1.1" 304 209 "-" "Mozilla/5.0 (Windows NT 6.1) AppleWebKi
t/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:43:57 -0300] "GET /favicon.ico HTTP/1.1" 404 500 "-" "Mozilla/5.0 (Windows NT 6.1)
 AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:53:13 -0300] "GET /favicon.ico HTTP/1.1" 404 501 "-" "Mozilla/5.0 (Windows NT 6.1)
 AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:53:15 -0300] "GET / HTTP/1.1" 200 641 "-" "Mozilla/5.0 (Windows NT 6.1) AppleWebKi
t/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:53:15 -0300] "GET /pirata.jpg HTTP/1.1" 200 25973 "http://192.168.2.1/" "Mozilla/5
.0 (Windows NT 6.1) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:53:15 -0300] "GET /favicon.ico HTTP/1.1" 404 500 "-" "Mozilla/5.0 (Windows NT 6.1)
 AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
10.0.0.1 - - [30/Nov/2011:19:27:08 -0300] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (Windows NT 5.2; rv:7.0.1) Geck
```

Fonte: O autor

O resultado da busca ilustrado na Figura 28 também retornou informações dos *logs* que foram apagados. As informações relevantes são as mesmas obtidas na busca realizada na superfície do HD invadido (ver Figura 25).

Através da análise do arquivo lastlog, que contém informações dos usuários que realizaram *login* na máquina, descobriu-se um suspeito, conforme ilustra a Figura 29.

Figura 29 - Lendo o arquivo lastlog com a ferramenta cat

```
root@caine:~# cat /media/sdc1/forense/lastlog
Nome de Usuário         Porta     De             Último
root             pts/0    192.168.2.11     Qua Nov 30 18:47:47 -0300 2011
daemon                                      **Nunca logou**
bin                                         **Nunca logou**
sys                                         **Nunca logou**
sync                                        **Nunca logou**
games                                       **Nunca logou**
man                                         **Nunca logou**
lp                                          **Nunca logou**
mail                                        **Nunca logou**
news                                        **Nunca logou**
uucp                                        **Nunca logou**
proxy                                       **Nunca logou**
www-data                                    **Nunca logou**
backup                                      **Nunca logou**
list                                        **Nunca logou**
irc                                         **Nunca logou**
gnats                                       **Nunca logou**
nobody                                      **Nunca logou**
libuuid                                     **Nunca logou**
Debian-exim                                 **Nunca logou**
statd                                       **Nunca logou**
sshd                                        **Nunca logou**
luciano          pts/0    192.168.2.11     Seg Nov 14 08:26:43 -0300 2011
ronaldo                                     **Nunca logou**
ronaldo-admin    pts/0    192.168.2.11     Qua Nov 30 18:47:47 -0300 2011
```

Fonte: O autor

A Figura 29 mostra que o usuário ronaldo-admin realizou *login* no servidor no mesmo dia da invasão minutos antes da percepção que o site havia sido modificado. Descobriu-se que esta conta de usuário pertencia a um ex-funcionário e administrador de redes da empresa que havia sido demitido há alguns dias, devido ter se envolvido numa briga com o gerente de TI da empresa. O próximo passo após esta descoberta é verificar todos os arquivos referentes a esta conta, de preferência começando pelo histórico de comandos, conforme ilustra a Figura 30.

Figura 30 - Lendo o histórico de comandos com a ferramenta cat

```
root@caine:~# cat /mnt/home/ronaldo-admin/.bash_history
wget http://www.mediafire.com/?hoixtbp1rl1ejcz
ls
rm index.html\?hoixtbp1rl1ejcz
wget http://www.mediafire.com/?hoixtbp1rl1ejcz
rm index.html\?hoixtbp1rl1ejcz
ls
wget http://download744.mediafire.com/1rb0bd8scxig/hoixtbp1rl1ejcz/index.zip
ls
ld
ls
gunzip index.zip
exit
ls
cat index.zip
PuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYP
TYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTT
uTTYPuTTYfd
apt-get install unzip
unzip index.zip
ls
cp index.html /var/www/
cp pirata.jpg /var/www/
rm -r /var/log
ls /var/
exit
```

Fonte: O autor

Ao visualizar o arquivo .bash_history, descobriu-se pelos comandos utilizados que o usuário ronaldo-admin é o responsável por modificar a página da empresa, como também de excluir os *logs* numa tentativa de não ser identificado. A Figura 30 mostra os passos do invasor: primeiramente com o comando rm remove o arquivo index.html da pasta /var/www, depois com o comando wget realiza um *download* de um arquivo chamado index.zip, em seguida instala no servidor o programa gunzip para poder descompactá-lo. Logo após copia os arquivos index.hmtl e pirata.jpg para a pasta /var/www, por fim deleta os arquivos de *log*. Nesse momento é possível perceber a importância dos arquivos que foram obtidos durante a Forense In Vivo. O comando ilustrado na Figura 30 para apagar os *logs* é idêntico ao encontrado na busca feita na memória (ver Figura 27).

Depois deve-se buscar outras informações de *log* nos diretórios /tmp, /var/tmp e /etc. Tanto o /tmp como o /var/tmp contém arquivos temporários, a diferença entre os dois é que na hora do *boot* os dados contidos no /tmp são apagados, já o /var/tmp os dados são mantidos, podendo ser apagados pelo usuário ou um programa que tenha essa função. O diretório /etc contém as configurações da máquina.

Para descobrir se existe algum *rootkit* ou *backdoor* que o atacante pode ter instalado para ter uma volta fácil sem necessitar fazer uma nova invasão por métodos elaborados, pode-se usar a ferramenta clamav para varrer todo o disco, conforme abaixo:

```
root@caine:~# clamscan -r /mnt/
```

Nesta busca o clamav não encontrou nenhum arquivo suspeito, o que leva a crer que a intensão do invasor era apenas de denegrir a imagem da empresa.

Um cuidado que todo Perito deve ter antes de concluir qualquer investigação que seja, é analisar todas as evidências para não elaborar o Laudo Pericial com conclusões prematuras. O invasor, por exemplo, pode ter sido outra pessoa e utilizou a conta ronaldo-admin para incriminá-lo ou mesmo utilizado técnicas anti-forense para dificultar a investigação, mas pelas circunstâncias e evidências analisadas não restam dúvidas que foi mesmo este usuário quem praticou a ação. A seguir é descrito como elaborar o Laudo Pericial.

4.3 Elaborando o Laudo Pericial

Segundo Mota Filho (2011), não existe um modelo específico para Laudo ou relatório de forense, mas todos devem conter pelos menos informações como:

- Dados pessoais do perito (Qualificação);
- Período da realização da forense;
- Breve relato do ocorrido (notícias iniciais);
- Dados gerais da máquina e/ou sistema atacado (sistema Operacional, nome da máquina, portas abertas, partições, situações da memória);
- Detalhamento dos procedimentos realizados;
- Dados e fatos relevantes encontrados;
- Conclusão e recomendações;
- Apêndices e anexos.

O Laudo Pericial referente à parte prática deste trabalho pode ser visualizado no Apêndice A.

5 CONCLUSÃO

As etapas de coleta da Forense In Vivo, Forense de Rede e da Forense Post Mortem são distintas, mas em momento algum independentes, pois se complementam e concorrem para a criação do Laudo Pericial.

O sucesso de uma análise depende fundamentalmente da qualidade do material coletado e da validade dos procedimentos adotados para que se tornem válidas em um tribunal. No entanto, todos estes procedimentos devem ser baseados nas legislações locais.

As empresas estão cada vez mais vulneráveis a ações de criminosos virtuais já que a grande maioria delas está conectada a *Internet* e os tutoriais de invasão encontram-se aos montes neste meio atraindo muitos adeptos com objetivos variados que pode ser pela simples diversão de conseguir tal façanha ou até mesmo para obter algum benefício próprio. Por mais que o invasor tente apagar seus rastros, o Princípio da Troca de Locard, diz que isso é praticamente impossível, pois como na vida real qualquer um, ou qualquer coisa, que entra em um local de crime, leva consigo algo do local e deixa alguma coisa para trás quando parte.

Existem hoje no mercado muitas ferramentas de software Livre, algumas já otimizadas para este fim (Live CDs), que podem ser obtidas sem custo algum para a análise destes incidentes nas mais diversas circunstâncias. Um exemplo delas são as nativas do próprio sistema operacional Debian e seus derivados, as quais foram aplicadas neste trabalho em um ambiente simulado.

Como a maioria do material que trata de forense computacional está em língua inglesa, este trabalho possibilita a realização de uma forense com ênfase nas Forenses In Vivo e Post Mortem de maneira fácil e intuitiva, até para aqueles com pouco conhecimento em sistemas GNU/Linux, não sendo uma literatura definitiva sobre o assunto.

O mais importante no Laudo Pericial é reproduzir todas as ações que foram feitas acompanhado por no mínimo duas testemunhas, para que não fiquem dúvidas do que foi feito e de que durante a perícia forense nenhum dado foi alterado.

A Forense Computacional é importante para as empresas, pois ela ajuda a reconstruir os crimes computacionais, conferindo-lhes validade probatória em juízo, mas para isso é necessário que seja feito por um profissional capacitado que tenha domínio das suas ferramentas e técnicas existentes. Caso este profissional não conheça o sistema operacional que irá investigar, poderá solicitar um auxiliar técnico.

REFERÊNCIAS

BORBA, Antônio. **Estudo diz que ataques internos são os novos vilões da área de tecnologia**. 2011. Disponível em: <http://www.tiinside.com.br/13/10/2011/estudo-diz-que-ataques-internos-sao-os-novos-viloes-da-area-de-tecnologia/sg/244782/news.aspx> Acessado em: 25 nov 2011.

CÂMARA, Paulo Ricardo Matos. **Segurança dos dados nas empresas:** problemas e soluções. 2010. Disponível em: <http://www.esab.edu.br/arquivos/monografias/Monografia%20Paulo%20Ricardo%20Matos%20C%C3%A2mara.pdf> Acessado em: 02 nov 2011.

CERT-BR. Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil. **Estatísticas de Incidentes**. 2011. Disponível em: <http://www.cert.br/stats/incidentes/> Acessado em: 18 set 2011.

FARMER, Dan; VENEMA, Wietse. **Perícia forense computacional:** teoria e prática aplicada. São Paulo: Editora Pearson Prentice Hall, 2007.

JOÃO IBAIXE JR. **Crimes eletrônicos:** algumas considerações. 2009. Disponível em: <http://ultimainstancia.uol.com.br/conteudo/colunas/2565/colunas+ultimainstancia.shtml> Acessado em 23 out 2011

LIMA, Arthur Diego de Lira. **FARO:** ferramenta para análise forense computacional em sistemas linux vivos. 2009. Disponível em: <http://www.engcomp.ufrn.br/publicacoes/TCC-2009-2-7.pdf> Acessado em: 16 ago 2011.

MACHADO, M. Pereira; ELEUTÉRIO, P. M. da Silva. **Computação forense e a perícia criminal**: a informática a serviço da justiça. 2011. Disponível em: <http://portal.sbc.org.br/horizontes/edicoes/v04n01/v04n01-19.pdf> Acessado em: 19 set 2011.

MELO, Sandro. **Computação forense com software livre**: conceitos, técnicas, ferramentas e estudos de casos. 1.ed. Rio de Janeiro: Alta Books, 2009.

MOTA FILHO, João Eriberto. **Forense computacional em linux for dummies:** uma rápida visão introdutória. 2011. Disponível em: <http://www.eriberto.pro.br/palestras/forense_linux.pdf> Acessado em: 27 nov 2011.

NEVES, Michel Weiler. **Crimes Eletrônicos:** a responsabilidade do usuário. 2009. Disponível em: <http://www.safernet.org.br/site/noticias/crimes-eletr%C3%B4nicos-responsabilidade-usu%C3%A1rio> Acessado em: 01 nov 2011.

PAIVA, Jadilson Alves de. **Práticas anti-forense:** um estudo de seus impactos na forense computacional. 2009. Disponível em: <http://www.nogueira.eti.br/profmarcio/obras/Jadilson%20-%20Anti-Forense.pdf> Acessado em: 18 set 2011.

PEREIRA, E. D. V. et al. **Forense Computacional:** fundamentos, tecnologias e desafios atuais. 2007. Disponível em: <http://www.sbseg2007.nce.ufrj.br/documentos/Minicursos/minicurso_forense.pdf> Acessado em: 23 set 2011.

PONPOLIN, José Geraldo. **Análise de ferramentas para computação forense em sistemas ntfs**. 2011. Disponível em: <http://www.ginux.ufla.br/files/mono-JoseGeraldoPopolin_0.pdf> Acessado em: 29 set 2011.

SANTOS, Laudelino Azeredo dos. **Computação forense em sistemas gnu/linux**. 2008. Disponível em: <http://www.ginux.ufla.br/files/mono-LaudelinoSantos.pdf> Acessado em: 20 set 2011.

TANENBAUM, A. S. **Redes de computadores**. 4.ed. Rio de Janeiro: Campus, 2003.

APÊNDICE A: LAUDO PERICIAL

Perito/examinador: Luciano Araújo de Medeiros

Data: 30/11/2011

Hora de início: 19:26

Hora de término: 07:05 – 01/12/2011

Descrição da perícia: Investigação de um ataque do tipo *Defacement* – A página web da empresa "A" foi modificada.

Observações: A máquina estava ligada, o cabo de rede não foi desconectado e após o incidente ninguém a utilizou.

IDENTIFICAÇÃO DO SISTEMA E PROCEDIMENTOS REALIZADOS

Nome do computador: srv-web

Sistema Operacional: Debian versão 2.6.32

Mac Address: 08:00:27:1a:f9:db

Em virtude de a máquina invadida ter permanecido ligada e o cabo de rede não ter sido desconectado, a primeira coisa a ser feita foi usar a estação forense para capturar o tráfego de entrada e saída da placa de rede desta. Em seguida foram utilizadas várias ferramentas contidas em um dispositivo externo conectado à máquina para capturar e salvar as seguintes informações: *kernel*, memória (*dump*), tabela ARP e de rotas, conexões, portas, interface de rede, processos, usuários, data e hora, pacotes e disco. Depois de gerado o *hash* de todos eles, a máquina foi desligada puxando-se o cabo força da energia elétrica para evitar que a mesma grave dados no disco.

O próximo passo foi gerar as imagens do disco. Para isso a máquina foi configurada para dar *boot* por um *Pen-Drive* com o sistema Ubuntu instalado. Após o Ubuntu ser iniciado foi conectado e montado na máquina um dispositivo externo com capacidade maior que o HD da máquina para salvar as imagens criadas. Também foram criados o *hash* das imagens.

Em seguida foi utilizada uma estação forense com o sistema Ubuntu instalado para montar uma das imagens que corresponde a partição raiz do sistema como *loop* - por ser arquivo e *read only* - para não permitir que os dados sejam alterados.

Por se tratar de um ataque do tipo *Defacement*, resolveu-se verificar primeiro os *logs* do sistema, começando pelo diretório /var. Com o comando ls constatou-se que o diretório *log*

havia sido apagado. A atual situação do diretório /var está presente nos anexos.

Ao varrer a imagem e o arquivo de memória com a ferramenta strings, configurada para pesquisar pela data do incidente, foi possível visualizar os *logs* apagados e algumas evidências referentes ao ataque conforme os anexos. Ao examinar o arquivo de memória (*dump*) com a mesma ferramenta, só que configurada para pesquisar pelo comando de exclusão rm, obteve-se as informações presentes nos anexos.

Ao analisar o arquivo lastlog capturado quando a máquina ainda estava ligada, descobriu-se que o usuário ronaldo-admin havia feito *login* na máquina minutos antes do incidente acontecer. A informação que obtive da empresa é que esta conta de usuário pertence a um ex-funcionário e administrador de redes que foi demitido há alguns dias, devido ter se envolvido em uma briga com outro funcionário e esta não havia sido desativada. Por este motivo resolveu-se verificar os arquivos e diretórios deste usuário. Ao verificar o arquivo referente aos comandos utilizados e compará-los com as evidências obtidas anteriormente, chegou-se a conclusão que o responsável pelo incidente havia sido este. Os demais diretórios incluindo o tmp, /var/tmp e /etc foram verificados, mas nenhuma evidência foi encontrada. Para tentar descobrir se foi instalado algum *rootkit* ou *backdoor* na máquina, utilizou-se a ferramenta clamav para varrer toda a imagem, mas nenhum arquivo suspeito foi encontrado.

CONCLUSÃO E RECOMENDAÇÕES

Conclui-se que o ataque se tratou de um *Defacement*, onde o objetivo do autor era de denegrir a imagem da empresa. Foi descoberto que a conta de usuário usada para modificar a página pertencia a um ex-funcionário e administrador de redes, mas não havia sido desativada. Sugere-se que a conta seja desativada e que o antigo dono seja investigado como também qualquer computador de uso pessoal deste, e sempre que algum funcionário for demitido sua conta seja desativada imediatamente para evitar que outras invasões como esta se repitam.

APÊNDICES E ANEXOS

1. Atual situação do diretório /var:

```
root@caine:~# ls /mnt/var/
backups  cache  lib  local  lock  mail  opt  run  spool  tmp  www
```

2. Utilizando a ferramenta strings na imagem da máquina invadida:

```
root@caine:~# strings /media/sdc1/hd_raiz.dd |grep 30/Nov/2011
192.168.2.11 - - [30/Nov/2011:18:34:43 -0300] "GET / HTTP/1.1" 200 484 "-" "Mozilla/5.0 (Windows NT 6.1
) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:34:43 -0300] "GET /favicon.ico HTTP/1.1" 404 500 "-" "Mozilla/5.0 (Win
dows NT 6.1) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:43:55 -0300] "GET /favicon.ico HTTP/1.1" 404 501 "-" "Mozilla/5.0 (Win
dows NT 6.1) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:43:57 -0300] "GET / HTTP/1.1" 304 209 "-" "Mozilla/5.0 (Windows NT 6.1
) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:43:57 -0300] "GET /favicon.ico HTTP/1.1" 404 500 "-" "Mozilla/5.0 (Win
dows NT 6.1) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:53:13 -0300] "GET /favicon.ico HTTP/1.1" 404 501 "-" "Mozilla/5.0 (Win
dows NT 6.1) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:53:15 -0300] "GET / HTTP/1.1" 200 641 "-" "Mozilla/5.0 (Windows NT 6.1
) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:53:15 -0300] "GET /pirata.jpg HTTP/1.1" 200 25973 "http://192.168.2.1/
" "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:53:15 -0300] "GET /favicon.ico HTTP/1.1" 404 500 "-" "Mozilla/5.0 (Win
dows NT 6.1) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
10.0.0.1 - - [30/Nov/2011:19:27:08 -0300] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (Windows NT 5.2; rv
:7.0.1) Gecko/20100101 Firefox/7.0.1"
```

3. Utilizando a ferramenta strings na memória da máquina invadida:

```
root@caine:~# strings /media/sdc1/forense/memoria.dd |grep 30/Nov/2011
N[30/Nov/2011:18:53:15 -0300]
192.168.2.11 - - [30/Nov/2011:18:53:15 -0300] "GET /favicon.ico HTTP/1.1" 404 500 "-" "Mozilla/5.0 (Windows NT 6.1)
 AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
N[30/Nov/2011:18:53:13 -0300]
192.168.2.11 - - [30/Nov/2011:18:53:13 -0300] "GET /favicon.ico HTTP/1.1" 404 501 "-" "Mozilla/5.0 (Windows NT 6.1)
 AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:53:
N[30/Nov/2011:18:53:15 -0300]XF
N[30/Nov/2011:19:27:08 -0300]
10.0.0.1 - - [30/Nov/2011:19:27:08 -0300] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (Windows NT 5.2; rv:7.0.1) Geck
o/20100101 Firefox/7.0.1"
N[30/Nov/2011:19:27:08 -0300]
10.0.0.1 - - [30/Nov/2011:19:27:08 -0300] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (Windows NT 5.2; rv:7.0.1) Geck
o/20100101 Firefox/7.0.1"
192.168.2.11 - - [30/Nov/2011:18:34:43 -0300] "GET / HTTP/1.1" 200 484 "-" "Mozilla/5.0 (Windows NT 6.1) AppleWebKi
t/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:34:43 -0300] "GET /favicon.ico HTTP/1.1" 404 500 "-" "Mozilla/5.0 (Windows NT 6.1)
 AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:43:55 -0300] "GET /favicon.ico HTTP/1.1" 404 501 "-" "Mozilla/5.0 (Windows NT 6.1)
 AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:43:57 -0300] "GET / HTTP/1.1" 304 209 "-" "Mozilla/5.0 (Windows NT 6.1) AppleWebKi
t/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:43:57 -0300] "GET /favicon.ico HTTP/1.1" 404 500 "-" "Mozilla/5.0 (Windows NT 6.1)
 AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:53:13 -0300] "GET /favicon.ico HTTP/1.1" 404 501 "-" "Mozilla/5.0 (Windows NT 6.1)
 AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:53:15 -0300] "GET / HTTP/1.1" 200 641 "-" "Mozilla/5.0 (Windows NT 6.1) AppleWebKi
t/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:53:15 -0300] "GET /pirata.jpg HTTP/1.1" 200 25973 "http://192.168.2.1/" "Mozilla/5
.0 (Windows NT 6.1) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
192.168.2.11 - - [30/Nov/2011:18:53:15 -0300] "GET /favicon.ico HTTP/1.1" 404 500 "-" "Mozilla/5.0 (Windows NT 6.1)
 AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.121 Safari/535.2"
10.0.0.1 - - [30/Nov/2011:19:27:08 -0300] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (Windows NT 5.2; rv:7.0.1) Geck
```

4. Código fonte da página web modificada:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http:/
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title></title>
</head>
<body>
<h1>Pagina pichada</h1>
<p>Informacoes negredindo a imagem da empresa</p>
<img src="pirata.jpg"/>
<h1>ELFOH4CK3R</h>
</body>
```

5. Realizando uma busca por comandos com a ferramenta strings na memória:

```
root@caine:~# strings /media/sdc1/forense/memoria.dd |grep rm |grep /var/log
rm -r /var/log
```

6. Lendo o arquivo lastlog com a ferramenta cat

```
root@caine:~# cat /media/sdc1/forense/lastlog
Nome de Usuário         Porta     De               Último
root             pts/0    192.168.2.11     Qua Nov 30 18:47:47 -0300 2011
daemon                                      **Nunca logou**
bin                                         **Nunca logou**
sys                                         **Nunca logou**
sync                                        **Nunca logou**
games                                       **Nunca logou**
man                                         **Nunca logou**
lp                                          **Nunca logou**
mail                                        **Nunca logou**
news                                        **Nunca logou**
uucp                                        **Nunca logou**
proxy                                       **Nunca logou**
www-data                                    **Nunca logou**
backup                                      **Nunca logou**
list                                        **Nunca logou**
irc                                         **Nunca logou**
gnats                                       **Nunca logou**
nobody                                      **Nunca logou**
libuuid                                     **Nunca logou**
Debian-exim                                 **Nunca logou**
statd                                       **Nunca logou**
sshd                                        **Nunca logou**
luciano          pts/0    192.168.2.11     Seg Nov 14 08:26:43 -0300 2011
ronaldo                                     **Nunca logou**
ronaldo-admin    pts/0    192.168.2.11     Qua Nov 30 18:47:47 -0300 2011
```