UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA INTERNET

GUILHERME ZANINI DE SÁ

AVALIAÇÃO DE TÉCNICAS ANTIFORENSES COMPUTACIONAIS APLICADAS A REGISTROS DE SISTEMAS LINUX

TRABALHO DE CONCLUSÃO DE CURSO

CAMPO MOURÃO
2013

GUILHERME ZANINI DE SÁ

AVALIAÇÃO DE TÉCNICAS ANTIFORENSES COMPUTACIONAIS APLICADAS A REGISTROS DE SISTEMAS LINUX

Trabalho de Conclusão de Curso de graduação do Curso Superior de Tecnologia em Sistemas para Internet da Coordenação de Informática da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito para colação de grau.

Orientador: Prof. Me. Rodrigo Campiolo

CAMPO MOURÃO

2013

ATA DA DEFESA DO TABALHO DE CONCLUSÃO DE CURSO

Às **vinte e um horas** do dia **três de maio de dois mil e treze** foi realizada no Mini-auditório do EAD da UTFPR-CM a sessão pública da defesa do Trabalho de Conclusão do Curso Superior de Tecnologia em Sistemas para Internet do acadêmico **Guilherme Zanini de Sá** com o título **AVALIAÇÃO DE TÉCNICAS ANTIFORENSES COMPUTACIONAIS APLICADAS A REGISTROS DE SISTEMAS LINUX**. Estavam presentes, além do acadêmico, os membros da banca examinadora composta pelo professor **Me. Rodrigo Campiolo** (Orientador-Presidente), pelo professor **Me. Luiz Arthur Feitosa dos Santos** e pelo professor **Me. Frank Helbert Borsato**. Inicialmente, o aluno fez a apresentação do seu trabalho, sendo, em seguida, arguido pela banca examinadora. Após as arguições, sem a presença do acadêmico, a banca examinadora o considerou **APROVADO** na disciplina de Trabalho de Conclusão de Curso e atribuiu, em consenso, a nota ____ (_____). Este resultado foi comunicado ao acadêmico e aos presentes na sessão pública. A banca examinadora também comunicou ao acadêmico que este resultado fica condicionado à entrega da versão final dentro dos padrões e da documentação exigida pela UTFPR ao professor Responsável do TCC no prazo de **onze dias**. Em seguida foi encerrada a sessão e, para constar, foi lavrada a presente Ata que segue assinada pelos membros da banca examinadora, após lida e considerada conforme.

Observações:

_____

_____

_____

_____

_____

Campo Mourão, 03 de maio de 2013.

Me. Luiz Arthur Feitosa dos Santos	Me. Frank Helbert Borsato	Me. Rodrigo Campiolo
Membro	**Membro**	**Orientador**

À Deus, que me concedeu a Sua Graça, favor imerecido e benevolente para com os homens.

À memória de Élcio de Sá, cujos sábios conselhos me serviram a vida toda e por sua dedicação com a família que expressaram o que é amar.

À Djanira Zanini de Sá, não somente mãe, mas guerreira e refúgio nos momentos de dor, um presente de Deus para nós.

À Francisco Mineiro Junior e Deise Mineiro, que conduziram e incentivaram minha educação formal, que me amaram e me ampararam em momentos difíceis.

À Tais, Cintia e Ronaldo, meu irmãos os quais amo.

À minha companheira e amada Jennifer, uma batalhadora, mulher de Deus, minha esposa.

À todos meus amigos, que me ajudaram, me incentivaram e foram pacientes comigo.

AGRADECIMENTOS

Certamente estes parágrafos não irão atender a todas as pessoas que fizeram parte dessa importante fase de minha vida. Portanto, desde já peço desculpas àquelas que não estão presentes entre essas palavras, mas elas podem estar certas que fazem parte do meu pensamento e de minha gratidão.

Reverencio o Professor Me. Rodrigo Campiolo pela sua dedicação e pela orientação deste trabalho e, por meio dele, eu me reporto a toda a comunidade da Universidade Tecnológica Federal do Paraná (UTFPR) pelo apoio incondicional.

A todos os colegas da Universidade gostaria de externar minha satisfação de poder conviver com eles durante a realização deste curso, pelos momentos de estresse, pelas muitas conversas nas salas de aula, pois são como alavancas para o convívio sadio perante a sociedade e pela grande paciência comigo.

Agradeço aos pesquisadores e professores da banca examinadora pela atenção e contribuição dedicadas a este estudo.

Reconheço também o carinho da minha família, pois acredito que sem o apoio deles seria muito difícil vencer esse desafio. E por último, e nem por isso menos importante, agradeço a minha esposa Jennifer pelo carinho, amor e compreensão.

RESUMO

SÁ, Guilherme Zanini de. **Avaliação de Técnicas Antiforenses Computacional Aplicadas a Registros de Sistema Linux**. 59f. Trabalho de Conclusão de Curso (Tecnologia em Sistemas para a Internet) – Programa de Graduação em Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal de do Paraná. Campo Mourão, 2013.

Este trabalho aborda as técnicas antiforenses computacionais aplicadas a arquivos de registro, conhecido como logs. Os registros são uma das principais fontes de informações para peritos criminais e administradores de rede investigarem comportamentos anômalos em sistemas computacionais, como por exemplo, resultados de uma invasão. Objetiva-se neste trabalho investigar e analisar as técnicas antiforenses aplicadas a sistemas de registro e avaliar a efetividade e dificuldade de execução de tais técnicas. Foram selecionados a distribuição GNU/Debian Linux para a realização de dois casos de estudo: instalação padrão e medidas de segurança. Em ambos os casos foram aplicadas as técnicas antiforenses e avaliadas a efetividade e dificuldade baseadas na execução de cada técnica. Foi considerado neste trabalho que o sistema alvo já estava comprometido. Verificou-se que as técnicas usadas como um super usuário são eficientes em um ambiente sem proteção adequada, logo, torna-se simples um invasor ocultar seus rastros.

Palavras-chave: Antiforenses. Logs. Análise Forense. Técnicas. Segurança. Linux.

ABSTRACT

SA, Guilherme Zanini de. **Analysis of the Anti-forensic Techniques Applied to Linux Log Files**. 59p. End of Course Work (Systems Technology for the Internet) - Graduate Program in Systems Technology for the Internet, Federal Technological University of Paraná. Campo Mourão, 2013.

This work addresses anti-forensic techniques applied in log files. Log files are the main source of information for forensic experts and network administrators to investigate anomalies in computer systems, such as results from an invasion. The aim was to investigate anti-forensic techniques used in log files and also evaluate the efficacy and difficulty of implementing such techniques. We used GNU / Debian Linux and we carried out two case studies: standard installation and safety measures. In both case, antiforenses techniques were evaluated regarding efficacy and difficulty. We assumed that the target system was already compromised. We verified that the techniques used as a super user are efficient in an environment without proper protection hence it becomes easy an invader hide his actions.

Keywords: Anti-forensic. Security. Linux. Forensic analysis.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

SUMÁRIO

1. INTRODUÇÃO

Na mesma proporção que os sistemas de informações interagem entre si, as tentativas de fraude também ganham campo (CERT.BR, 2013). Devido a essa grande interação, os recursos computacionais são acessados repetidamente. Por essa razão, vestígios de atividades incomuns ao sistema, não apenas se destacam, mas também são notáveis durante um longo período de tempo, uma vez que são armazenados e preservados em registros (FARMER e VENEMA, 2007).

Em geral, os sistemas de registro computacionais armazenam arquivos como sequência de bytes e organizam esses arquivos dentro de uma hierarquia de diretórios. Possuem, no entanto, além dos nomes e conteúdo, também outros atributos como permissões de acesso, data/hora da última modificação, entre outros. Não é de se admirar que os registros sejam o principal alvo das tentativas de fraude pelos invasores (FARMER e VENEMA, 2007).

A análise forense computacional permite a coletar e analisar evidências digitais, reconstruindo ataques e recuperando dados, além de rastrear invasores. Logo, os registros constituem uma fonte valiosa de informações para a perícia forense (FARMER e VENEMA, 2007). Em contrapartida, para evitar ou ao menos dificultar a busca por evidências deixadas nesses sistemas, um invasor utiliza-se de técnicas antiforenses computacionais de ocultação ou corrupção de dados (HATCH, 2003).

Sites conhecidos como IDG Now e a CERT.BR apresentam constantemente notícias sobre ataques, invasões e furtos de informações sigilosas, que por sua vez, são evidências da importância deste trabalho, da conscientização sobre a necessidade de segurança e contextualização sobre o tema abordado (IDG NOW, 2013; CERT.BR, 2013).

Objetiva-se com este trabalho analisar as técnicas antiforenses computacionais aplicadas a registros dos sistemas operacionais Linux. Como objetivos específicos tem-se: simular um ambiente invadido e demonstrar a ocultação dos rastros nos registros, analisar a efetividade das técnicas em um ambiente padrão e em um ambiente com medidas de segurança previamente habilitadas.

A análise de como os invasores ocultam suas ações é importante para que profissionais e administradores de rede possam aplicar contramedidas para manter segura as informações de registros ou, no mínimo, identificar rapidamente a ocorrência de comportamentos suspeitos no sistema.

O estudo foi realizado utilizando uma distribuição Debian virtualizada e considerando que a máquina servidora já estava comprometida pelo invasor. Executou-se as técnicas em dois ambientes distintos, um com segurança padrão do sistema e o outro com o uso de *softwares* auxiliares que garantem a segurança do sistema. Para verificar as técnicas

antiforenses foram considerados os critérios de efetividade, facilidade de execução e ausência de rastros.

Este trabalho está dividido em 5 capítulos principais. O capítulo 2 apresenta o desenvolvimento teórico sobre a perícia forense computacional, sistemas de registros, registros no Linux, técnicas antiforenses aplicadas aos registros e possíveis contramedidas de segurança. O capítulo 3 contém os materiais e métodos utilizados na investigação das técnicas antiforenses e medidas de segurança. O capítulo 4 apresenta a aplicação e análise das técnicas de ocultação dos rastros gerados nos registros e também sua aplicação com algumas medidas de segurança previamente instaladas. O capítulo 5 apresenta os resultados e discussões, considerando cada técnica e sua eficácia. O capítulo 6 apresenta a conclusão do trabalho e possíveis trabalhos futuros.

2. DESENVOLVIMENTO TEÓRICO

Neste capítulo são abordadas as características, definições e processos da perícia forense computacional. Conseguinte, são abordados os sistemas de registros, conhecidos como *logs*. São descritas as principais características e como se comportam em ambiente Linux. A seguir, são definidas as práticas antiforenses computacionais e as etapas de um ataque computacional. Por último, são apresentadas algumas técnicas usadas por um invasor para manter seu sigilo e esconder seus delitos.

2.1. PERÍCIA FORENSE COMPUTACIONAL

Esta seção apresenta a definição e os processos da perícia forense computacional.

2.1.1. Definição

A perícia é considerada a ciência de coletar e analisar evidências digitais, reconstruindo ataques e recuperando dados para rastrear invasores (FARMER e VENEMA, 2000).

O responsável por realizar a perícia forense é denominado perito forense. O perito forense computacional consiste em um analista de sistemas, especialista em metodologias laboratoriais e padronização da investigação, aquisição, análise e manuseio de evidências em inquéritos, forense digital, inteligência e contra inteligência, como definem vários autores (ICCyber, 2010; FAERMER e VENEMA, 2007; TOMÁS, Eliane Maria Cordeiro, 2010).

Para que a perícia possa realizar seu trabalho, conceitos essenciais devem ser entendidos, sugerido por Farmer e Venema (2007), como a volatilidade dos dados, a divisão em camadas e confiança, a coleta de quantidade máxima de evidências confiáveis de um sistema em execução e a recuperação de informações parcialmente destruídas com o intuito de manter a credibilidade das informações analisadas.

Na tentativa de reconstruir a linha temporal do sistema analisado, a perícia deve seguir metodologias para o descobrimento de invasões e análise de informações, denominadas processos da perícia.

2.1.2. Processos da perícia

Ovie Carroll e outros (2008) distribuem a execução de perícia forense computacional em quatro fases: coleta de dados, exame de dados, análise de evidências e documentação técnica.

2.1.2.1. Coleta dos dados

A fase de coleta de dados conglomera uma sequência de passos que visam manter

a total integridade das evidências encontradas e coletadas.

Durante a pré-coleta, referente à chegada ao local da investigação, alguns cuidados são adotados. O isolamento da área ou local do crime, que visa impedir alguma alteração ou contaminação das evidências, ou se necessário, fotografar ou filmar o ambiente ou equipamento para amparar alguma análise futura, são fatores fundamentais para se manter a integridade da análise.

Investigam-se, na fase de coleta, possíveis fontes de dados. Computadores, dispositivos eletrônicos, portas de comunicação de periféricos, são identificados para procura de evidências criminais. Além dos dados internos ao ambiente analisado, dados significativos podem estar em provedores, servidores ou outros domínios externos à cena averiguada.

2.1.2.2. Exame dos dados

Na fase de exame dos dados, somente as informações mais importantes à investigação são avaliadas, extraídas, filtradas e documentadas, como por exemplo, informações que denunciem a atuação do invasor.

Diversos processos, ferramentas e técnicas disponíveis, podem reduzir muito a quantidade de dados que necessitam de um exame minucioso, pois grande quantidade de informação pode não ser útil e muito trabalhosa de se analisar. Filtros de palavras-chaves ou de arquivos, por exemplo, pela extensão do mesmo, permissões de acesso, entre outros, são meios importantes para essa seleção.

2.1.2.3. Análise de evidências

Somente extrair dados relevantes, não é o suficiente. O perito forense deve analisar e interpretar essas informações de forma correta.

O passo de análise de informações visa identificar características que indiquem relações com a área do crime, como pessoas, *e-mails*, telefones, locais, vídeos.

A análise das evidências levanta artefatos que indicam ou não, paralela à etapa do exame, se os eventos investigados estão relacionados.

Freitas (2003) divide o método de análise pericial em duas categorias, a análise física e a análise lógica. O exame sequencial e a obtenção de dados de todo o conceito pericial, dos arquivos normais às partes inacessíveis da mídia, definem a análise física, em contra partida, a análise lógica refere-se à análise dos arquivos de partições.

2.1.2.4. Laudo técnico

O laudo técnico consiste no último processo da perícia forense computacional. Consiste no relatório gerado a partir da análise, destinado as pessoas leigas, por isso a

linguagem técnica deve ser simples, para que todos possam compreender. O perito deve utilizar análises gráficas, visuais, com o objetivo de facilitar a compreensão. O laudo tem o objetivo de materializar a cena do crime, demonstrar como foi feito e denunciar o autor do crime. (FREITAS, 2003)

2.2. SISTEMAS DE REGISTROS

Nesta seção são apresentados o conceito e uso dos registros, além de algumas de suas características como volatilidade e MACTimes. Também são abordadas as características desses registros no Linux, tais como a estrutura, a localização, a função e quais os principais registros.

2.2.1. Conceito e uso de registros

Os sistemas de registros são responsáveis por armazenar informações valiosas que são, praticamente, as mesmas em vários sistemas. As características comuns em um registro são: início de uma sessão, *login pid*, *tty device*, *tty*, usuário, endereço, *status* de saída, identificação da seção, tempo e *IP* do acesso e o *logout. Esses* registros podem ser armazenados em dois tipos de arquivos: os arquivos em formato de texto e os arquivos binários.

Os registros armazenados em arquivos texto são mais comuns e o conteúdo pode ser facilmente compreendido, pois pode ser visualizado em qualquer editor de texto. Possuem a facilidade de geração por não dependerem de aplicativos específicos.

Os registros armazenados em arquivos binários, por sua vez, somente podem ser entendidos por *softwares* específicos. Os binários são arquivos que contêm caracteres incompreensíveis para leitores de texto padrão. São gerados através de um processo de codificação de formato de determinado arquivo que só pode ser completamente visualizado por programas especializados.

Esses registros são usados tanto pelo invasor, quanto por profissionais e administradores para adquirem informações do sistema. Essas informações são essenciais para o controle do sistema.

2.2.2. OOV – “Ordem da Volatilidade”

Todos os dados são voláteis, e a Ordem da Volatilidade está intimamente relacionada com a coleta desses dados. A veracidade das informações, assim como a capacidade de recuperação ou validação dos dados, diminui com o passar do tempo. A Ordem da Volatilidade consiste em dois membros de uma hierarquia, as informações impossíveis de recuperar dentro de um espaço de tempo muito curto, definido por Farmer como parte superior, e a parte

inferior concernente às formas bastante persistentes e raramente mudam. (FARMER e VENEMA, 2007).

Qualquer alteração no sistema pode desencadear alterações em outras áreas, e dessa forma comprometer dados importantes em um registro. O Quadro 1 descreve a relação entre tipos de dados e seu tempo esperado de vida.

O ciclo de vida esperado dos dados.	
Tipos de Dados	**Tempo de Vida**
Registradores, memória periférica, caches, etc.	Nanossegundos
Memória Principal	Dez nanossegundos
Estado da rede	Milissegundos
Processos em execução	Segundos
Disco	Minutos
Disquetes, mídia de backup, etc.	Anos
CD-ROM, impressões, etc.	Dezenas de Anos

Quadro 1 - Ciclo de Vida dos Dados

Fonte: *Farmer e Venema (2007), p. 172.*

2.2.3. MACtimes

Farmer e Venema (2007) criaram o termo *mactime* para se referir aos três atributos de tempo – *mtime*, *atime* e *ctime* – que são anexados a qualquer arquivo ou diretório no *Windows*, *UNIX* e em outros sistemas de arquivos.

Essas abreviações possuem o seguinte significado:

- ***Mtime*:** muda quando um arquivo ou diretório for modificado.
- ***Atime*:** apresenta a última data/hora em que o arquivo ou diretório for acessado.
- ***Ctime*:** monitora quando o conteúdo ou as metainformações sobre o arquivo mudar, tais como o proprietário, grupo e as permissões de arquivo.

O quadro 2 mostra como se apresentam os *MACtimes* em um sistema computacional.

MACTimes de um Sistema							
Wed Mai 05 2013 15:17:02	3888	m.c	-/-rwxrwxrwx	48	0	1102686	/usr/bin/.y
Thu Mai 05 2013 21:01:34	2229	.ac	-/-rwxrwxrwx	48	0	1102688	/usr/bin/.y/back.tar.gz

Quadro 2 - MACTimes de um Sistema

O quadro 6 apresenta os campos respectivamente: data, tamanho, mactime, permissões, UID, GID, contador, localização do arquivo. Este quadro apresenta um nome de diretório incomum que pode ser forte indicação da ação de suspeitos. Os registros são de extrema importância para qualquer sistema, pois possuem as informações mais importantes

em relação a qualquer atividade realizada no mesmo.

2.2.4. Registros no Linux

Os registros no Linux geralmente estão contidos no diretório /var/log, e são arquivos de texto ou arquivos binários que armazenam as informações referentes ao sistema. O sistema de registro proporciona uma análise completa do funcionamento do *hardware*, acessos, *e-mails*, programas, entre outros.

Para o Linux o processo mais importante é conhecido como syslogd. A funcionalidade do syslogd é gerenciar os registros do sistema e capturar as mensagens do núcleo do Linux. O anexo A descreve sucintamente as funcionalidades do syslog.

Alguns registros, dos mais diversos existentes, proporcionam ao perito, informações mais importantes que outros. O Quadro 3 descreve alguns dos principais sistemas de registros do Linux e respectivas funcionalidades e finalidades.

REGISTRO	FUNCIONALIDADE
btmp	Informa quando ocorreu a última tentativa de login que não teve sucesso no sistema
messages	Armazena os acessos ocorridos nos sistema
smbd	Possui informações referentes ao servidor de arquivos do Linux que suportam uma rede Windows.
httpd	Informações do servidor Web Apache.
maillog	Informações referentes ao servidor de E-mails
syslog.conf	Informa onde estão armazenados os logs do sistema. (Deve-se cuidar com alterações feitas por um invasor).
auth.log	Informações de acesso como root, inclusive o sucesso.
daemon.log	Informação referente às tentativas de conexão, remotamente ou localmente.

Quadro 3 - Principais Registros e Sua Funcionalidade

Fonte: *Guia Foca GNU/Linux. Capítulo 17 - Arquivos e daemons de Log*

O *syslogd* consiste em um controlador de serviços de detalhamento padrão do sistema (*System Logging Daemon*). O arquivo */etc/syslog.conf* lhe permite especificar onde as mensagens *syslog* são armazenadas, baseadas no programa responsável e no nível de detalhamento da mensagem.

O Quadro 4 mostra o recurso para o qual são designados os grupos de registros.

Recurso	*Descrição*
auth	*Mensagens de segurança/autorização (desaprovado)*
authpriv	*Mensagens de segurança/autorização*
cron	*Tarefas agendadas e rotinas*
daemon	*Outros daemons do sistema (sshd, inetd, pppd)*
kern	*Mensagens do kernel.*
lpr	*Subsistema da impressora.*
mail	*Subsistema de correio (sendmail, postfix, qmail, etc.)*
news	*Mensagens de Notícias da Usenet.*
syslog	*Mensagens internas de syslog.*
user	*Mensagens genéricas no nível de usuário.*
uucp	*Subsistema UUCP.*
local0-local7	*Níveis reservados para uso local.*

Quadro 4 - Facility Syslog

Fonte: Segurança contra Hacker Linux (Hatch, Lee, Kurtz, 2003). Pg. 70.

Quando os programas capturam uma entrada do registro, dependendo de sua configuração, o processo do sistema syslog pode ignorá-lo ou informá-lo de acordo com o nível de detalhamento apresentado no quadro 5.

Nível de Logging	Descrição.
emerg	Sistema inutilizável.
alert	Ação precisa ser tomada por posthaste.
crit	Condições críticas.
err	Condição de erro.
warning	Condições de advertência.
notice	Condições normais, mas significativas.
info	Mensagens informativas.
debug	Mensagens de depuração.

Quadro 5 - Níveis de Logging (detalhamento) de Syslog

Fonte: Segurança contra Hacker Linux (Hatch, Lee, Kurtz, 2003). Pg. 70.

O *syslog-ng* consiste em um registro não padrão, porém mais eficiente que o *syslogd*. O arquivo de configuração *syslog-ng.conf* além das funcionalidades inerentes ao *syslogd*, tais como especificar vários destinos (servidores remotos, arquivos locais, etc.) o usuário pode definir as origens das mensagens e atuar de modo diferente para gerar eventos localmente, em contrapartida às mensagens *syslog* remotas (HATCH, LEE, KURTZ. 2003. Pg. 73).

O Linux possui uma ferramenta denominada *logrotate*. Segundo o manual do administrador do sistema, o *logrotate* é projetado para facilitar a administração de sistemas que geram um grande número de arquivos de registro. Ele permite a rotação automática,

compressão, remoção, e envio de arquivos de registro. Normalmente, o *logrotate* é executado como uma tarefa diária do agendador de tarefas do Linux: o *cron*. O que demonstra que o *logrotate* não executará várias vezes no mesmo dia, porém pode ocorrer dos registros adquirirem um tamanho muito grande, o que o fará rotacionar.

2.3. ANATOMIA DE UM ATAQUE

Nesta seção são apresentadas as etapas de um ataque a sistemas computacionais. É importante compreender as etapas para entender porque os invasores apagam as evidências da invasão.

Basicamente a realização de um ataque possui o padrão apresentado pela figura 1.

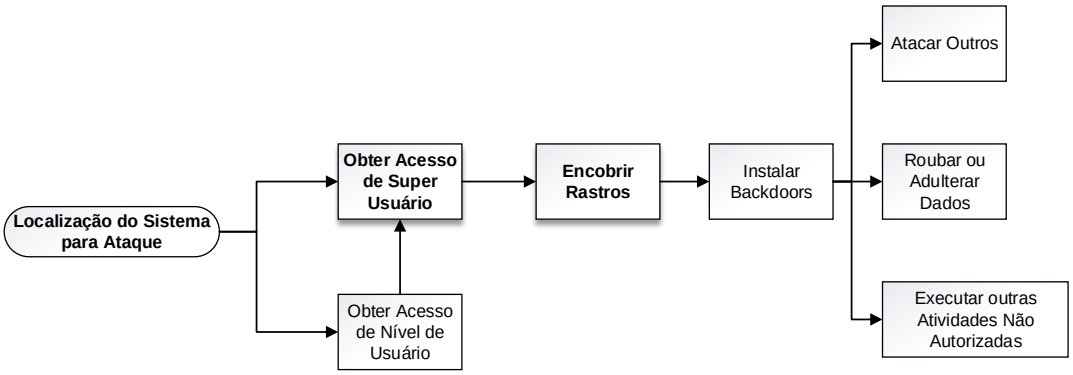

Figura 1 - Anatomia de um Ataque

Fonte: *Segurança da Informação – Disponível em: www.cin.ufpe.br*

A primeira etapa consiste na localização do sistema e conhecimento sobre o mesmo que é uma técnica de coleta de informações sobre os sistemas e as entidades a que pertencem. Isto é feito através do emprego de técnicas de segurança de vários computadores, como: consulta de *DNS*, enumeração de rede, consulta de rede, por exemplo, porta de acesso *ssh*, identificação do Sistema Operacional, consulta organizacional, ping, consulta de pontos de acesso e engenharia social (*WHOIS*). Algumas das ferramentas utilizadas para reconhecimento são: *nslookup*, *traceroute*, n*map* e *neotrace*.

O quadro 6 apresenta algumas características das ferramentas utilizadas.

Programa	*Características*
nslookup	É uma ferramenta utilizada para se obter informações sobre registros de *DNS* de um determinado domínio, *host* ou *IP*
traceroute	*Traceroute* é uma ferramenta que permite descobrir o caminho feito pelos pacotes da rede desde a sua origem até o seu destino
nmap	*Nmap* é um *software* livre que realiza escaneamento de portas desenvolvido pelo Gordon Lyon. É muito utilizado para avaliar a segurança dos computadores, e para descobrir serviços ou servidores em uma rede de computadores.
neotrace	*NeoTrace* é uma poderosa ferramenta para checar informações em locais da Internet.

Quadro 6 - Ferramentas utilizadas para reconhecimento

Já durante a fase de varredura, conhecida como *scanning*. De posse dos dados coletados, o invasor pode determinar quais sistemas estão ativos e alcançáveis ou, por exemplo, quais portas estão ativas em cada sistema. Para isso utilizam-se de muitas ferramentas que estão disponíveis, tais como: *nmap*, *system banners*, informações via *SNMP* (do inglês *Simple Network Management Protocol* - Protocolo Simples de Gerência de Rede).

Para a descoberta da topologia da rede, utiliza-se de ferramentas de descoberta automatizadas, tais como c*heops*, *ntop*, entre outras. Para tal fim o ping, traceroute, nslookup, tornam-se comandos de maior utilização. O quadro 7 apresenta algumas ferramentas e suas respectivas características.

Programa	**Característica**
cheops	Desenvolvimento *Open Source* que faz varreduras na rede identificando *Hosts*, *IPS*, sistemas operacionais, mostrando conexões entre outras ferramentas administrativas.
ntop	O *ntop* é uma ferramenta para monitorar e gerenciar redes de computadores, além de ter muitos recursos gráficos e informações detalhadas.

Quadro 7 - Programas usados para descoberta de topologia de rede

Outra etapa importante é a enumeração, onde o invasor faz consultas intrusivas como consultas diretas ao sistema, no qual pode ser notado, faz a identificação de acessos válidos, identificação versões de *HTTP* e *FTP*. Para a identificação da rede verifica-se: compartilhamentos (*Windows*) com os comandos *"net view"* e *"nbstat"*, os arquivos de sistema exportados (*Unix*) com o comando *showmount*. Para então, verificar as vulnerabilidades mais comuns do sistema através do *"nessus", "saint", "satan" ou "tara".* O quadro 8 demonstra as principais ferramentas e suas características.

Programa	Característica
Nessus	*Nessus* é um programa de verificação de falhas/vulnerabilidades de segurança. Ele é composto por um cliente e servidor, sendo que a varredura propriamente dita é feita pelo servidor.
Saint	*SAINT* (Ferramenta de Integração de Rede) é um software de computador usado para a varredura de redes de computadores em busca de vulnerabilidades de segurança e exploração de vulnerabilidades encontradas.
Satan	A ferramenta de administração de segurança para análise de redes consiste em um software de testes e relatórios que reúne uma variedade de informações sobre os hosts da rede. Possui uma interface web com formulários completos.
Tara	Assistente de pesquisas analíticas *Tiger*. É uma atualização do programa *TAMU* '*tiger'*, e apresenta relatórios legíveis após uma varredura.

Quadro 8 - Programas usados para varredura de rede

Na etapa da invasão, ou no ganho de acesso, com as informações coletadas, as estratégias de invasão são definidas. O invasor por possuir uma base de informações referentes a vulnerabilidades procura bugs no sistema operacional, *kernel*, serviço, aplicativo, por respectivas versões. Assim pode adquirir ao menos os privilégios de usuário comum. Para conseguir senhas e outros privilégios, utiliza-se de técnicas como: *"password sniffing"*, *"password crackers"*, *"password guessing"*, *"session hijacking"* (sequestro de sessão) e ferramentas para bugs conhecidos *"buffer overflow"*.

O quadro 9 apresenta as técnicas de aquisição de senhas e suas respectivas características.

Técnica	Característica
Password sniffing	É uma técnica que visa a obtenção de senhas, envolvendo o monitoramento do tráfego em uma rede para retirar informações.
Password crackers	É o processo de recuperação de senhas a partir de dados que tenham sido armazenados ou transmitidos por um sistema de computador.
Password guessing	É o processo de adivinhação de senha por dois modos possíveis, ataques de força bruta ou ataques de dicionário.
Session hijacking	Refere-se à exploração de uma sessão válida de um computador – as vezes também chamada de Session Key ou ID. Possui o intuito de conseguir acesso não-autorizado a informações ou serviços em um sistema de computador, utilizando de possíveis interceptações de cookies usados para manter seções.
Buffer overflow	Consiste na aplicação de variáveis maiores que as definidas

Quadro 9 - Técnicas de obtenção de senhas e características

Com o acesso comum, procura o acesso completo do sistema (*root* ou administrador), o que define a escalada de privilégios. Algumas ferramentas são conhecidas para a procura de bugs, tais quais os *"exploits"*. A escalada de privilégios utiliza-se das mesmas técnicas anteriores, com acréscimos de *"replay attacks"* e *"trojans horse"*.

O invasor procura evitar a detecção de sua presença, utilizando-se de ferramentas que desabilitem a auditoria, para isso é necessário, inclusive tomar o cuidado com o tempo excessivo de inatividade, que denunciam um ataque, pois deixam espaços evidentes nos registros. São utilizadas algumas ferramentas para remoção seletiva do *"event log"*, os quais podem esconder arquivos instalados no sistema ("*backdoors*").

O objetivo é a manutenção do acesso, utilizando de ferramentas como *"rootkits"*, *"trojan horse" e "backdoors"*. Através dos *"rootkits"*, que consistem em ferramentas ativas, porém ocultas, se confundem com o sistema, escondendo comandos modificados para não revelar o invasor. Através dos *"trojan horses"*, o invasor obtém informações como captura do teclado ou um e-mail com senhas do sistema. E por último com os backdoors, o invasor possui acesso remoto sem autenticação, inclusive não aparecem na lista de processos.

Os ataques são realizados pelo invasor com os seguintes objetivos: consumos de banda da rede, colapso de recursos, falhas de programação (Exemplo: *Ping* da Morte), dano de roteamento ou sabotagem de *DNS*.

2.4. TÉCNICAS ANTIFORENSES

Nesta seção são apresentadas as técnicas antiforenses aplicadas a registros. Assume-se que durante a aplicação dessas técnicas, o invasor obteve acesso administrativo ao servidor e provavelmente a uma parte significativa da árvore de uma rede ou então ao

computador local.

2.4.1. Desabilitação de registros

A desabilitação de registros consiste em uma prática de desativar o sistema de auditoria, modificar as datas de acesso e arquivos de registro. Para realizar esta técnica o invasor pode-se utilizar das ferramentas do sistema operacional ou algum *software* auxiliar. (HATCH, LEE, KURTZ, 2003).

2.4.2. Adulteração dos registros

A adulteração dos registros consiste no ato de modificar dados ou propriedades dos arquivos, como por exemplo a mudança do *ctime* de um arquivo. O método menos sofisticado de remover informações de *logging* é editar, apagar ou excluir arquivos de registro.

Existem muitos registros importantes do sistema, tais como *messages*, *xfelog*, *secure*, *wtmp*, *mail.log* e *bash_history*. O invasor poderá utilizar programas auxiliares do tipo *logcleanner* para apagar as informações que ficam contidas no *wtmp*, *utmp* e *lastlog*, o qual é visualizado pelo super administrador para encontrar informações dos usuários, tais como: início de uma sessão, *login pid*, *tty device*, *tty*, usuário, endereço, *status* de saída, identificação da seção, tempo e *IP* do acesso e o *logout* (HATCH, LEE, KURTZ, 2003).

Os registros do servidor Web Apache são muito visados para adulteração. Esses registros, como *access_log*, *ssl_request_log* e *ssl_engine_log* podem fornecer informações vitais para um administrador (Oracle, 2013).

2.4.3. Exclusão segura de registros

A exclusão segura de registros é uma técnicas antiforense, ou uma ferramenta, que consiste em excluir uma pasta ou arquivo de forma que não haja maneiras de recuperação de dados, mesmo com programas especializados em recuperação. Para isso o invasor pode utilizar ferramentas que realizam essa tarefa de excluir permanentemente um arquivo, como por exemplo, o *wipe* ou o *secure-delete* que não são padrões do sistema operacional Linux.

2.4.4. Inserção de entradas falsas de registro

O intuito do invasor na questão de inserir entradas falsas no sistema consiste em desviar a atenção do perito para outro usuário. Como por exemplo utilizando-se do programa *logger* para confundir e culpar outro usuário, assim quando já obteve o acesso *root*, outro pode ser culpado em seu lugar (HATCH, LEE, KURTZ, 2003).

Mesmo que o administrador do sistema seja atencioso e veja que o usuário invasor está tentando enganar, para evitar esse tipo de descoberta, facilmente o invasor esconderia

seu nome de usuário fornecendo uma opção –t para *logger*:

2.4.5. Ataque a logrotate para eliminar registros antigos

Como os arquivos de registros podem ficar grandes, o Linux faz um rodízio dos mesmos, com o programa chamado logrotate, normalmente configurado em */etc/logrotate.conf,* podendo, um invasor, utilizar-se disso para obter vantagens, uma vez que o logrotate gira os registros de acordo com o tamanho do arquivo ou o tempo que está no sistema, definido pelo administrador, consiste em utilizar do logger para encher o registro, tornando-o muito grande. Assim, quando atingir determinado número, o arquivo logrotate apagará seus rastros (HATCH, LEE, KURTZ, 2003).

2.5. MEDIDAS DE SEGURANÇA

Muitas devem ser as medidas tomadas por um administrador de rede para que evite invasões, dificulte o prejuízo causado, restaure o que foi perdido, se recupere de um dano, encontre rastros e procure anomalias no sistema. Este trabalho é centrado nos registros, logo não aborda todas as contramedidas para incidentes de segurança, não obstante apresentar-se-á as mais importantes, priorizando as medidas dos sistemas de registro.

2.5.1. Princípios básicos de segurança

O administrador deve-se certificar de que seus arquivos só possam ser lidos e gravados pelo super usuário, ou um grupo com uma finalidade especial, como *logs* ou *adminlogs*.

O administrador também deve ter uma conta especial sem privilégios de super usuário para acessar aos registros, assim verá que qualquer acesso de super usuário não veio dele.

Para dificultar ainda mais, o administrador também pode armazenar remotamente qualquer registro gerado pelo sistema, dessa forma obrigando o invasor além de esconder seus rastros também procurar comprometer a máquina remota.

2.5.2. Protegendo arquivos de relatório de acesso ao sistema

A maioria das ferramentas que alteram os arquivos *utmp* e *wtmp* substituem as entradas de alvo por bytes NULL, o que pode ser facilmente detectado por ferramentas como logchk ou chrootkit (HATCH, LEE, KURTZ, 2003).

2.5.3. Cópia de segurança online

A melhor maneira de se recuperar de arquivos adulterados consiste no *backup*, além

tanto físico, como em CDs ou DVDs, como os efetuados *online*. Pode-se analisar, dessa forma, quais registros são incompatíveis, e possivelmente se há alguma adulteração nos processos do sistema, com programas que verificam a integridade dos arquivos. Um administrador sempre mantém uma cópia de ferramentas intactas, como *cat, more, prep, netstat, md5sum, iptables, ps, rpm, lsof* e outras de relatório (HATCH, 2003. Pág. 609).

2.5.4. Registros externos ao registro

Segundo as práticas de segurança do CERT.BR (2013) os registros são tradicionalmente armazenados em disco, no próprio sistema onde são gerados. Entretanto, essa prática apresenta riscos inerentes. O primeiro problema ocorre com a destruição dos registros durante uma invasão do sistema. Para contornar a situação aconselha-se a instalação de um *loghost* centralizado.

Os *loghosts* apresentam duas vantagens importantes que consistem em um repositório redundante de registros e a indisponibilidade de acesso remoto, nem mesmo para os administradores. Além de permitirem uma análise mais facilitada e a correlação entre os eventos ocorridos nos sistemas. Para evitar um ataque de negação de serviço recomenda-se o armazenamento dos registros em partição separada, e para evitar a perda dos dados durante a rotação automática de registros, deve-se garantir que os registros sejam movidos para o armazenamento *off-line*.

2.5.5. Nomes de arquivos manipulados com espaço

Quando um administrador utiliza-se da ferramenta *ls* para verificar as pastas, o invasor pode ter utilizado de espaços para enganar o administrador, forçando o mesmo a não compreender o nome real do arquivo e não pensar que há um espaço vazio depois do último caractere de um nome, conhecido como o caráter não imprimível mais vulnerável do Linux. A melhor forma de se utilizar o *ls* consiste em adicionar a propriedade *–aF* depois do comando. Dessa forma o Linux acrescenta um caractere / depois de cada diretório e um caractere * após cada executável (HATCH, LEE, KURTZ, 2003).

2.5.6. Servidor de registros

Em investigações de incidentes eletrônicos a análise de registros é de suma importância, por isso um servidor de registro facilita grandemente a investigação dos mesmos, já que os eventos ficam armazenados de forma centralizada. A Cert.br publicou um material de Cristine Hoepers e Klaus Steding-Jessen (2013) sobre a importância da análise e interpretação dos registros, apresentando várias ferramentas de interpretação e auxílio para manipulação de registros, o que é de extrema importância para um administrador de redes,

inclusive para um servidor de registros. O alvo *@loghost* do *logging* de mensagens é um modo simples de fazer com que seus registros vão para mais de uma máquina. Segundo Hatch (2003) se quaisquer rastros forem apagados de uma máquina invadida, eles ainda poderão estar disponíveis no servidor ou máquina de registro secundária.

2.5.7. Scanners de segurança do sistema

Os programas de scanner são ótimos auxiliares para encontrar e identificar problemas no sistema, entre os mais importantes são os de sistema e os de rede. Dessa forma o administrador pode verificar possíveis brechas ou inseguranças em potencial. Vários são os pacotes de verificação de registros, com diferentes funcionalidades entre eles: LogSentry, Logsurfer, Sec e o Lire.

2.5.8. Permissões dos registros

Para uma máxima proteção dos registros, o administrador deve ter o cuidado de tornar os registros reconhecidos e graváveis apenas por um super usuário e legíveis por um grupo chamado *log* sem permissões para outros.

Utilizando-se de permissões específicas de filesystem o administrador pode impedir até mesmo que o usuário administrador altere os registros, como exemplo o comando *chattr*, dos *filesystems ext2* e *ext3* (HATCH, LEE, KURTZ, 2003).

2.5.9. Criptografia do tráfego de syslog

Com o propósito de evitar que as mensagens syslog sejam facilmente identificadas por um invasor no ato de leitura dos pacotes da rede, o administrador deve criptografar o tráfego da rede. Uma maneira fácil de se executar essa medida de segurança consiste em criar um túnel *SSL* entre o emissor e receptor usando o *Stunnel*, particularmente utilizando-se do *syslog-ng*. Para isso a máquina que envia os registros deve acrescentar alguns comandos em */etc/syslog-ng/syslog-ng*.conf (HATCH, LEE, KURTZ, 2003).

Após adicionados os comandos necessários para a criação do destino, o *host* que deseja enviar a mensagem, executará o comando Stunnel para a criação do túnel com o destino.

No host receptor o *Stunnel* é chamado para receber a conexão do registro criptografada mudando-se apenas a posição das portas. Em seguida configura-se o *syslog-ng* do receptor para ler da porta *TCP* local (HATCH, LEE, KURTZ, 2003).

2.5.10. OSSEC – Sistema de detecção de intrusão de host

O *software* OSSEC consiste em um sistema *Open Source* para detecção de intrusão

em máquinas locais que realiza análise dos registros, verificação de integridade de arquivos, monitoramento de políticas, detecção de *rootkits*, alertas em tempo real e resposta ativa. O OSSEC é um excelente programa pois possui muitas características que auxiliam um administrador de sistema. É multiplataforma com alerta em tempo real. É possível configurar as alterações do nível de prioridade e integra-lo com *smtp*, *sms* e *syslog*, enviando *e-mails* e mensagens para dispositivos móveis (OSSEC, 2013).

2.5.11. SELinux – Security Enhanced Linux

A estrutura SELinux (Linux Seguro) fornece segurança adicional aos sistemas, limitando as ações dos usuários e programas pela imposição de políticas de segurança para o sistema operacional, que fornecem segurança contra acesso não autorizado. O núcleo do Debian Linux tem suporte ao SELinux, mas é desabilitado por padrão (DEBIAN, Wiki. 2013).

3. MATERIAIS E MÉTODOS

Este capítulo apresenta os materiais e os métodos utilizados para a realização do estudo de caso. Nos materiais são apresentadas as ferramentas e o cenário do estudo de caso. Nos métodos são apresentados e explicados os passos e critérios utilizados na avaliação das técnicas antiforenses.

3.1. MATERIAIS

Com o intuito de analisar técnicas antiforenses em registros Linux e verificar a efetividade destas e também auxiliar administradores de rede a se prevenirem contra elas selecionou-se os seguintes materiais:

- Oracle VM Virtual Box 4.2.12, com o intuito de simular as máquinas que estarão no teste, dispostos em uma rede com topologia estrela, utilizando de conexão em modo *bridge*.
- Duas máquinas virtuais instaladas com a versão do Sistema Operacional Debian 6.2 para arquitetura i386.
- Utilizou-se do *rootkit LRK* versão 6 com o intuito de analisar a efetividade sobre o sistema e o impacto que pode causar apenas com a substituição de binários ou utilização de outras ferramentas inclusas no mesmo.
- Utilizou-se o *OSSEC* versão 2.7 como detector de intrusão baseado em *host*, com o intuito de realizar as mesmas técnicas antiforenses em um sistema com uma segurança mais elevada.
- Reforçou-se o *kernel* do Linux com o *SELinux* pois consiste numa camada de implementação de segurança para o núcleo do Linux. Na prática, o *kernel* consulta o *SELinux* antes de cada chamada do sistema para saber se o processo está autorizado a fazer a operação dada.
- Utilizou-se do syslog-ng 3.4, uma vez que ele fornece algumas das mesmas funções do *syslog* e *metalog*, com uma pequena diferença: pode filtrar mensagens com base no nível e conteúdo (como o metalog), fornece registro remoto como o *syslog*, lida com registros do *syslodg*, pode escrever em um *TTY*, executar programas, e pode agir como um servidor de registros.
- Utilizou-se do *secure-shell* (ssh) 5.5p1 para conectar as máquinas, uma vez que fornece conexão criptografada e acesso remoto.

A figura 2 apresenta a topologia da rede utilizada para a realização do estudo.

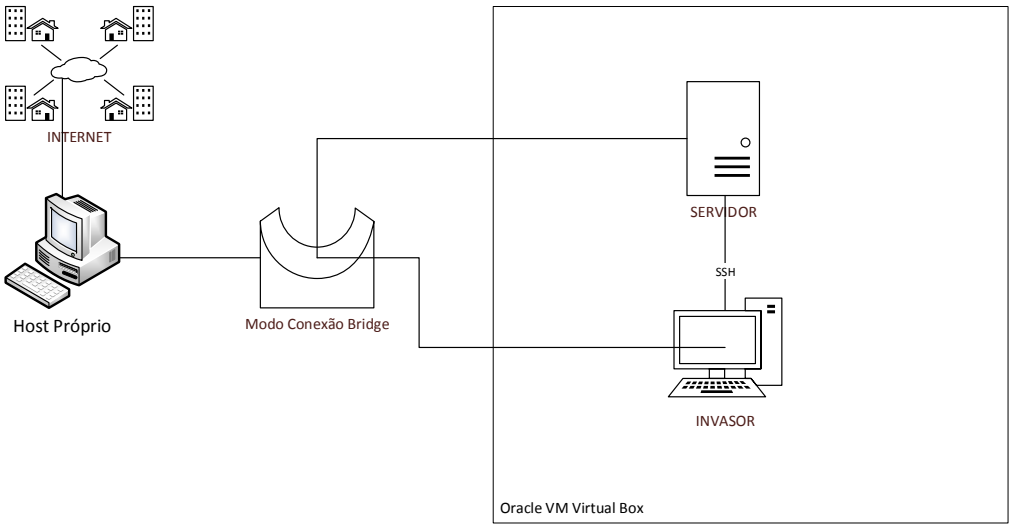

Figura 2- Topologia da Rede Utilizada

Verifica-se na figura 2 que o invasor utiliza de uma máquina para acessar remotamente via ssh, com uma conexão bridge, o servidor com a distribuição Linux Debian.

3.2. MÉTODOS

A metodologia utilizada para a execução das técnicas antiforenses considera hipoteticamente que o servidor já fora invadido, onde o invasor já possui os privilégios de super usuário ou administrador do sistema.

A amostra selecionada para abranger a totalidade do problema investigado são as duas máquinas virtuais, uma simulando o ambiente, inseguro e seguro, do servidor, e a outra representando o invasor do sistema.

Os métodos de coleta dos dados se baseiam na observação dos registros gerados pelo invasor, possível técnica para ocultação dos rastros produzidos e qual a possível contramedida para se evitar a aplicação dessas mesmas técnicas.

Foi separado o ambiente em duas fases distintas. O primeiro ambiente foi constituído de um ambiente com instalação padrão, para isso, foram seguidos os passos descritos a seguir.

Configuração do ambiente da arquitetura de rede: nesta etapa, inicialmente, foi instalado o Oracle VM Virtual Box. Foi criado uma nova máquina virtual com o nome de Servidor, com o sistema Operacional Linux versão Debian. Durante a configuração, foi selecionado um disco rígido VDI (VirtualBox Disk Image). Foi escolhido o armazenamento em disco rígido dinamicamente alocado. Em seguida, foi escolhida imagem do disco de instalação do Debian 6. Durante a instalação foram escolhidas as configurações padrões para o Brasil, inclusive o espelho de atualizações. O método de particionamento escolhido foi “Assistido –

usar disco inteiro". E foi selecionado a instalação separadas das partições */home, /usr, /var* e */tmp*. Foi nomeada a máquina de servidor. O servidor foi clonado e a máquina clonada foi nomeada para "invasor". Através do ssh o invasor foi conectado em modo bridge com a máquina virtual que representa o servidor.

Analisou-se os principais registros, binários e textuais, do Debian. Verificou-se quais são gerados no acesso ao sistema. Forma aplicadas técnicas antiforenses de manipulação de registros, adição de registros, exclusão de registros e instalação de programas que possam auxiliar o invasor a esconder seus rastros.

Em segunda instância foi analisado a efetividade das mesmas técnicas em um sistema seguro, configurado com os passos descritos a seguir.

Foi selecionado o servidor, intacto, sem qualquer atividade maliciosa do invasor, para a instalação das ferramentas de segurança. O *SELinux,* que constitui uma dessas ferramentas, vem desativado nessa versão do Debian. Foi utilizado do comando "aptitude install selinux-basics selinux-policy-refpolicy-targeted", para que o Linux resolvesse as dependências e se encarregasse de instalar mais alguns pacotes que auxiliam na administração do *SELinux*. Depois ativou-se o *SELinux*.

Para a instalação do Ossec-Hids 7 utilizou-se o "aptitude install build-essential", depois o "wget -cv http://www.ossec.net/files/ossec-hids-2.6.tar.gz" e descompactou-se o arquivo com "tar –xvzf" e executou-se o instalador.

Instalou-se o *syslog-ng* com o comando "aptitude install syslog-ng".

Instalou-se o *ssh* com "aptitude install ssh".

Conectou-se o servidor com o *host* do invasor através do protocolo de rede *ssh*, uma vez que possui conexão criptografada e permite-se executar comandos de uma unidade remota. As técnicas foram organizadas de forma cronológica e sequencial, procurando apresentar primeiramente as que evitam a geração de mais registros, as que podem ser adulteradas, as que podem ser removidas, as que devem ser removidas de forma segura, as que podem prejudicar o funcionamento dos processos responsáveis por esses registros, utilizando-se principalmente de *rootkit*, por último, ataques de negação de serviço em registros.

Para a conclusão do trabalho analisou-se as técnicas utilizando-se de critérios como facilidade em se encontrar as técnicas aplicadas, a dificuldade de execução das técnicas e a efetividade das mesmas, nível de conhecimento necessário, a aplicação de contramedidas e sua efetividade em relação às técnicas. Analisou-se a efetividade das técnicas em nos dois sistemas distintos: em um sistema seguro e no mesmo sistema com uma segurança padrão.

4. APLICAÇÃO E ANÁLISE DAS TÉCNICAS ANTIFORENSES

Este capítulo apresenta dois estudos de caso, onde foram aplicadas e analisadas as técnicas antiforenses.

No estudo de caso 1 foram aplicadas a um sistema operacional GNU/Linux Debian com instalação padrão as técnicas de remoção segura dos registros, conhecida como wipe shared, alteração dos registros, acréscimo de informações falsas, modificação do comportamentos dos processos dos registros, sobrecarga dos registros (ataque de negação de serviço - DOS), ataque a logrotate, diminuir ou desabilitar o nível de detalhamento.

No estudo de caso 2 foram aplicadas medidas de segurança ao sistema operacional GNU/Linux Debian. Foram instalados o SELinux, o syslog-ng, o OSSEC HIDS.

4.1. ESTUDO DE CASO 1: SEGURANÇA PADRÃO

Ao fazer um acesso ao host que deseja invadir, o invasor acabará gerando um registro de acesso remoto em */var/log/auth.log* como mostra a figura 3.

```
Mar 22 17:37:07 servidor su[1170]: Sucessful su for root by administrador
Mar 22 17:37:07 servidor su[1170]: + /dev/tty1 administrador:root
Mar 22 17:37:07 servidor su[1170]: pam_unix(su:session): session opened for user
  root by administrador(uid=1000)
Mar 22 19:11:30 servidor sshd[1042]: Server listening on 0.0.0.0 port 22.
Mar 22 19:11:30 servidor sshd[1042]: Server listening on :: port 22.
```

Figura 3 - Log gerado em auth.log por acesso SSH

Devido a isso, o invasor deve se preocupar em apagar seu acesso registrado no arquivo *auth.log*, entretanto não se deve esquecer que seus comandos ficam registrados em *history*, como demonstra a figura 4.

```
124  exit
125  ifconfig
126  cat auth.log
127  last
128  history
```

Figura 4 – Uso do comando history

Então, para evitar que o perito ou administrador do host invadido possa analisar suas estratégias de invasão, o invasor terá o cuidado de impedir que seus comandos sejam gravados com o comando mostrado na figura 5. Caso o invasor já tenha digitado alguns comandos e tenha esquecido de executar o comando mencionado, ele pode apagar os comandos com *history –c*.

```
root@servidor:~# unset HISTFILE
```

Figura 5 - Primeiro comando executado por um invasor

Depois disso o invasor terá a liberdade de apagar o registro gerado em *auth.log*. Como demonstra a figura 6.

```
root@servidor:~# cd /var/log/
root@servidor:/var/log# pico auth.log
```

Figura 6 - Localização e Edição do registro auth.log

Após aberto o arquivo, o invasor pode apagar o registro de acesso *ssh* gerado. Para não levantar suspeitas, ele procurará apagar somente as linhas que indicam o acesso. Como na figura 7.

```
Mar 22 15:02:38 servidor sshd[991]: Server listening on 0.0.0.0 port 22.
Mar 22 15:02:38 servidor sshd[991]: Server listening on :: port 22.
```

Figura 7 - Apagando dados do auth.log

Majoritariamente *softwares* de *login* como *OpenSSH*, *daemons telnet* ou *daemons Rlogin*, registram os acessos ao sistema bem-sucedidos em um arquivo chamado */var/run/utmp* ou */var/log/wtmp*, os quais armazenam informações importantes como horário de *login* e *logout* de cada usuário. Com o comando *last* pode-se verificar os dados contidos nesses arquivos que possuem um formato legível apenas para máquina. O comando *last* é demonstrado na figura 8.

```
root@servidor:/var/log# last
root          pts/0         192.168.1.4          Fri Apr 12 15:19 still logged in
root          pts/0         192.168.1.4          Fri Apr 12 15:14 – 15:19 (00:05)
administ      tty1                               Fri Apr 12 15:05 still logged in
administ      tty1                               Fri Apr 12 15:05 – 15:22 (00:28)

wtmp begins Mon Mar 10:46:35 2013
root@servidor:/var/logs# _
```

Figura 8 - Comando last verifica últimos acessos bem sucedidos

O invasor deve-se preocupar em editar o arquivo para remover seus rastros de acesso, para isso existem muitas ferramentas como: *wzap, zap, zap2, unix2, cloak* e *clear*. Na falta de ferramentas o invasor pode usar o próprio *pico* ou *vi*, assim pode apagar as partes que são legíveis e que crê ser útil para apagar as informações.

O comando *last* apresenta o registro gerado por *wtmp*, que também gera um arquivo binário. Devido a isso, o invasor deve alterar o arquivo *wtmp* encontrado em */var/log/wtmp*

para que o mesmo não lhe apresente as informações de *login*. Como demonstra a figura 9.

```
root@servidor:/var/log# last
wtmp begins fri Apr 12 15:37:44 2013
root@servidor:/var/log# _
```

Figura 9 - Comando last adulterado

O invasor pode deletar as informações geradas pelo *lastlog*, como demonstra a figura 10, na qual apresenta o arquivo binário, legível somente pela máquina.

```
GNU nano 2.2.4                    Arquivo: lastlog
=PhQpts/0^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^192.168.1.4^@^
```

Figura 10 - Editando arquivo lastlog com nano

A figura 11 demonstra como se apresentaria o comando *lastlog*, mostrando informações importantes para o administrador, como o endereço de *IP* do invasor.

```
Nome de Usuário        Porta          De            Último
root           pts/0          192.168.1.4        Sex Abr 12 15:19:41 -0300 2013
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**
nobody                                           **Nunca logou**
administrador                                    Sex Abr 12 15:05:33 -0300 2013
```

Figura 11 - Execução do comando lastlog e informações de IP

Depois de apagado os dados que são perceptíveis por serem diferentes de *"^@"*, a figura 12 mostra como se apresentaria o comando *lastlog* com as alterações feitas.

```
Nome de usuário              Porta         De             Ultimo
root                                                  **Nunca logou**
daemon                                                **Nunca logou**
bin                                                   **Nunca logou**
sys                                                   **Nunca logou**
```

Figura 12 - Resultado do comando lastlog adulterado

HATCH (2003) afirma que um cracker pode adulterar o próprio processo *syslog*, uma vez que a maioria dos programas do sistema submetem seus registros ao mesmo processo. Dessa forma o invasor pode recompilar parte do arquivo *syslogd.c* para ocultar quaisquer entradas que contenha seu endereço *IP*. A maneira mais fácil que um invasor encontra e que possui grande impacto no sistema, consiste em se utilizar de ferramentas que substituem os arquivos binários e que são encontrados facilmente na Internet, são os chamados rootkits. Como apresentado no quadro 10. O *rootkit* mais famoso se encontra no endereço http://packetstormsecurity.com/ e é conhecido como *LRK6*, ou Linux *Rootkit* versão 6.

Ferramentas	***Descrição***
syslogd	Esconde registros.
killall	Não encerra processos escondidos.
Backdoors	
wted	Editor de wtmp/utmp
z2	Apagador de utmp/wtmp/lastlog

Quadro 10 - Rootkit LRK6 e alguns binários inclusos

A figura 13 demonstra uma simples transferência de um arquivo, no caso o arquivo *z2* foi transferido para o servidor, com o intuito descrito no quadro acima. A transferência é muito fácil e o impacto pode ser grande.

```
root@servidor:/home/administrador/rootkit# ls
bin                    fileutils-3.13         lrk-4.1.tar.gz         rshd
bindshell.c            findutils              Makefile               shadow-961025
chfn                   fix.c                  MCONFIG                sysklog-1-3
                       Inetd                  net-tools-1.32-a       tcpd_7.4
chsh                   linsniffer.c           pssw                   wted.c
cron3.Op11             Login                  procps-1.01            z2.c
root@servidor:/home/administrador/rootkit# scp z2.c 192.168.1.6:/home/
z2.c                                       100% 2107     2.1KB/s       00:00
```

Figura 13 - Arquivos contidos no rootkit e sua transferência

O arquivo foi movido para a pasta de compilação como demonstra a figura 14. São muitas as ferramentas para um invasor, cada qual com sua função e a escolha de cada uma delas vai depender da necessidade do invasor e do seu objetivo.

```
root@servidor:/home# which gcc
/usr/bin/gcc
root@servidor:/home# mv z2.c /usr/bin/
root@servidor:/home# _
```

Figura 14 - Movendo o arquivo para o compilador C

Quando o invasor remove algum arquivo ainda há algumas formas de recuperá-los, no entanto há maneira de se remover arquivos ou diretórios de forma segura, não recuperável. A figura 15 demonstra a instalação do *wipe* e do *secure-delete* para a remoção segura de arquivos e diretórios. Vale ressaltar que um registro no *cache* do *apt-get* foi armazenado.

```
root@servidor:/# apt-get install wipe secure-delete -y
```

Figura 15 - Instalação do Wipe e o Secure-Delete

Primeiramente criou-se algumas pastas para o teste do *wipe* e *secure-delete* como a figura 16 demonstra.

```
root@servidor:/# mkdir pasta_a_ser_removida
root@servidor:/# touch pasta_a_ser_removida/arquivo_a_remover
root@servidor:/# touch pasta_a_ser_removida/arquivo_a_remover2
root@servidor:/# touch pasta_a_ser_removida/arquivo_a_remover3
```

Figura 16 - Criação de pastas para Teste

Pode-se utilizar o wipe ou o *secure-delete* para remoção dos arquivos e diretórios. A figura 17 demonstra a exclusão segura utilizando o *wipe*.

```
root@servidor:/pasta_a_ser_removida# wipe –fi –P 1 –r arquivo_a_remover
File arquivo_a_remover (0 bytes) wiped
Operation fineshed.
1 file wiped and 0 special files ignored in 0 directories, 0 symlinks removed
but not followed, 0 errors occured.
root@servidor:/pasta_a_ser_removida# _
```

Figura 17 - Remoção de Arquivo com Wipe e Teste

O *secure-delete* é um *suíte* que além de remover arquivos de forma segura pode limpar o espaço livre em disco, memória e *swap*. A figura 18 apresenta a remoção dos arquivos de três formas diferentes: sem chances de recuperação, aplicação balanceada entre segurança e rapidez e remoção insegura, respectivamente.

```
root@servidor:/pasta_a_removida# srm –d –r- -v –z arquivo_a_remover2
Using /dev/urandom for random input.
Wipe mode is secure (38 special passes)
Wiping arquivo_a_remover2 ********************************** Removed file ar
quivo_a_remover2 ... Done
root@servidor:/pasta_a_ser_removida# sem –d –l –r –v –z arquivo_a_remover3
Wipe mod is insecure (two passes [0xff/zero])
Wiping arquivo_a_remover3 ** Removed file arquivo_a_remover3 ... Done
root@servidor:/pasta_a_remover# cd ..
root@servidor:/# sem –d –l –l –r –v –z pasta_a_ser_removida/
Using /dev/urandom for random input.
Wipe mode is insecure (one pass [zero])
Wiping pasta_a_ser_removida/ DIRECTORY (going recursive now)
Warning: Couldn’t find a free filename for pasta_para_ser_removida/!
Removed directory pasta_a_ser_removida/ ... Done
```

Figura 18 - Remoção com Secure-Delete em 3 Etapas

Removido de forma segura, não há mais como se recuperar um arquivo, no entanto, a instalação de programas gera registros do *apt-get install* como demonstra a figura 19. Os registros dos programas instalados podem ser encontrados em */var/cache/apt/archives*. Os pacotes instalados vão sendo armazenados na pasta mencionada, o que pode denunciar a invasão.

```
root@servidor:/var/cache/apt/arqchives# ls
apache2_2.2.16-6+squeeze10_i386.deb
apache2.2-bin_2.2.16-6+squeeze10_i386.deb
apache2.2-commom_2.2.16-6+squeeze10_i386.deb
apache2-mpm-worker_2.2.16-6+squeeze10_i386.deb
apache2-utils _2.2.16-6+squeeze10_i386.deb
chkconfig_11.0-79.1-2_all.deb
lock
partial
secure-delete_3.1-5_i386.deb
shared-mime-info_0.71-4_i386.deb
ssh_1%3ª5.5p1-6+squeeze3_all.deb
syslog-ng_3.1.3-3_i386.deb
wipe_0.21-9_i386.deb
root@servidor:/var/cache/apt/archives# _
```

Figura 19 - Lista de Arquivos Instalados pelo apt-get

Com o simples comando *apt-get clean* as dependências não utilizadas podem ser apagadas e assim esconder os programas que foram instalados. Para removê-los utiliza-se o *apt-get remove*. Como demonstra a figura 20.

```
root@servidor:/var/cache/apt/archives# apt-get clean
root@servidor:/var/cache/apt/archives# ls
lock partial
root@servidor: /var/cache/apt/archives# apt-get remove wipe secure-delete
Lendo listas de pacotes... Pronto
Construindo árvore de dependências
Lendo informação de estado... Pronto
Os pacotes a seguir serão REMOVIDOS:
   secure-delete wipe
```

Figura 20 – Limpeza apt-get e remoção do wipe e secure-delete

Outra técnica utilizada por um invasor não consiste somente na exclusão ou adulteração dos registros, uma muito utilizada consiste adulterar os registros com o intuito de culpar outro usuário do sistema como suspeito de invasão, como demonstra a figura 21.

```
root@servidor:~# logger –p kern.alert “authentication failure; logname=administr
ador uid=510 euid=0 tty= ruser= rhost= user=root”
root@servidor:~# _
```

Figura 21 - Falso registro inserido para culpar outro usuário

Fornecendo a opção –t para *logger* o invasor pode facilmente esconder seu nome de usuário para evitar que um administrador atencioso veja que o invasor está tentando enganar, como demonstra a figura 22.

```
root@servidor:~# logger –p kern.alert –t ’su(root)’ ”authentication failure..”
```

Figura 22 - logger -t para esconder o nome de usuário do invasor

Um invasor pode fazer alterações no *logrotate* para fazê-lo girar, diminuindo o tamanho e o tempo dos registros para que o *logrotate* gire ou pode usar o *logger* para enche-los. Como demonstra a figura 23 a ordem para rotacionar diariamente o registro e executar o comando */sbin/killall – HuP syslogd* para reiniciar o processo indicado, foi adicionado no final do arquivo */etc/logrotate.conf*.

```
/var/log/messages{
          rotate 1
          size=10k
          posrotate
                 /sbin/killall –HUP syslogd
                 endscript
}
```

Figura 23 - Mensagem adicionada ao /etc/logrotate.conf

O invasor pode ainda manipular as datas forjando pacotes *NTP* (*Network Time Protocol*) para aparentar vir de um servidor confiável com o intuito de não precisar esperar o tempo de rotacionamento dos registros. Dessa forma, o invasor mudará a data da máquina e fará o *logrotate* executar, acreditando que a data está, por exemplo, a uma semana na frente, fazendo assim, o rotacionamento.

Outro comando muito utilizado para retirar as restrições de exclusão dos arquivos mesmo por super usuários é o *chattr*. A figura 24 mostra a utilização do comando *chattr*, cujo intuito do administrador consiste em alterar a permissão dos registros para evitar que sejam alterados.

```
root@servidor:/var/log# chattr +a /var/log/auth.log
root@servidor:/var/log# cat /dev/null  >  /var/log/auth.log
bash:  /var/log/auth.log: Operação não permitida
root@servidor:/var/log# _
```

Figura 24 - Utilização do comando chattr +a

Entretanto um usuário com acesso de super usuário pode executar o comando *chattr –a* para retirar o comando que impede a exclusão ou remoção do arquivo, não obstante esse comando é muito útil, principalmente se o invasor não conhecer o comando, nem souber o que está acontecendo.

4.2. ESTUDO DE CASO 2: MEDIDAS DE SEGURANÇA

Nesta seção utilizou-se medidas de segurança adicionais com o intuito de verificar a efetividade das mesmas técnicas nesse sistema e se há mais registros gerados e se há a possibilidade de excluí-los ou adulterá-los.

O administrador pode-se utilizar dessas ferramentas de verificação de forma muito simples como demonstra a figura 25.

```
apt-get install build-essential
wget http://www.ossec.net/files/ossec-hids-latest.tar.gz
wget http://www.ossec.net/files/ossec-hids-latest_sum.txt
```

Figura 25 - Download do OSSEC HIDS

O OSSEC HIDS pode ser configurado em */ossec/etc/ossec.conf* que é um arquivo no formato XML. A configuração usada para o estudo é demonstrado na figura 26. Criou-se as regras de diretórios “ossecAnalise” para facilitar a análise dos registros.

```
<syscheck>
   <!-- Frequência que o Syscheck é executado (dado em segundos)-->
   <frequency>200</frequency>

   <!-- Diretórios analisados -->
   <directories check_all="yes">/var/log/ossecAnalise</directories>
   <directories check_all="yes">/etc,/usr/bin,/usr/sbin</directories>
      <directories check_all="yes">/bin,/sbin</directories>
      <!-- Arquivos e diretórios ignorados -->
      <ignore>/var/log/ossecAnalise/arquivo.txt</ignore>
      <ignore>/etc/mtab</ignore>
      <ignore>/etc/mnttab</ignore>
      <ignore>/etc/hosts.deny</ignore>
      <ignore>/etc/mail/statistics</ignore>
      <ignore>/etc/random-seed</ignore>
      <ignore>/etc/adjtime</ignore>
      <ignore>/etc/httpd/logs</ignore>
      <ignore>/etc/utmpx</ignore>
      <ignore>/etc/wtmpx</ignore>
      <ignore>/etc/cups/certs</ignore>
      <ignore>/etc/dumpdates</ignore>
      <ignore>/etc/svc/volatile</ignore>
   <!-- Alerta caso sejam criados novos arquivos -->
   <alert_new_files>yes</alert_new_files>

   <!-- Arquivos e diretórios ignorados -->
   <ignore>/var/log/ossecAnalise/arquivo.txt</ignore>
</syscheck>
```

Figura 26 - Configuração do arquivo XML do OSSEC

A maioria dos registros do ossec se encontram na pasta */ossec/logs/ossec.log e /ossec/logs/alerts/alerts.log*. Primeiramente verificou-se os registros gerados quando um usuário acessa remotamente o sistema, no caso o invasor, no entanto o *ossec* não avisou sobre qual o IP do invasor, apenas que houve alguma mudança. Se um o invasor exclui algo do registro /var/log/auth.log o ossec verifica a integridade do arquivo e emite o alerta da figura 27.

```
** Alert 136655497.3640: mail  -  ossec.attacks,
2013 Apr 21 11:26:37  servidor ->ossec-logcollector
Rule: 592 (level 8)  ->  'Log file size reduced.'
ossec: File size reduced (inod remained): '/var/log/auth.log'.
```

Figura 27 - Alerta do OSSEC HIDS

O invasor pode tentar excluir também a mensagem de alerta do ossec, caso o mesmo ainda não tenha enviado um *e-mail* para o administrador, no caso de configuração *default*, essa informação do registro foi excluída e não gerou nenhum outro registro. O invasor com a conta de super usuário pode parar o funcionamento do *ossec hids*. Com o comando */ossec/bin/ossec-control stop*. Um registro será gerado em */ossec/logs/ossec.log*, como demonstrado na figura 28.

```
2013/04/21 18:26:54 ossec-monitord(1225): INFO: SIGNAL Received. Exit Cleaning…
2013/04/21 18:26:54 ossec-logcollector(1225): INFO: SIGNAL Received. Exit Cleaning…
2013/04/21 18:26:54 ossec-syscheckd(1225): INFO: SIGNAL Received. Exit Cleaning…
2013/04/21 18:26:54 ossec-analysisd(1225: INFO: SIGNAL Received. Exit Cleaning…
2013/04/21 18:26:54 ossec-execd(1314): INFO: Shutdown received. Deleting responses.
2013/04/21 18:26:54 ossec-execd(1225): INFO: SIGNAL Received. Exit Cleaning…
```

Figura 28 - Registro do ossec.log

O invasor pode excluir seus rastros, caso consiga impedir que o *ossec* envie o e-mail para o administrador, ou ainda se o mesmo não estiver funcionando corretamente, pois com o privilégio de super usuário ele pode simplesmente excluir as informações que julgar pertinente.

O SELinux foi instalado e configurado para reforçar a segurança do núcleo do Linux e o controle de acesso obrigatório. Instalou-se, primeiramente, os pacotes que contém a política padrão para o funcionamento do SELinux com o comando "*apt-get install selinux-basics auditd selinux-policy-default*". Para ativa-lo foi utilizado o comando "*selinux-activate*". Ele apresentou um falso positivo, alertando sobre */etc/pam.d/login*. É necessário reiniciar a máquina para que o SELinux entre em funcionamento como demonstra a figura 29.

```
Mounting local filesystems...done.
Activating swapfile swap...done.
[. . . .] Checking SELinux contexts: selinux-basics
.  ok
```

Figura 29 - Selinux em funcionamento

Para verificar se o SELinux está com a instalação de forma correta é necessário utilizar do comando "*sestatus"*, como demonstra a figura 30.

```
servidorSeguro:~# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             default
Current mode:                   permissive
Mode from config file:          permissive
Policy MLS status:              enabled
Policy deny_unknown status:     denied
Max kernel policy version:      26
servidorSeguro:~# _
```

Figura 30 - Status do SELinux

O SELinux pode ser desativado temporariamente utilizando-se do *"/usr/sbin/setenforce 0",* e depois o invasor pode mudar a configuração em /etc/selinux/conf para desativar o SELinux como demonstra a figura 31, modificando o SELINUX=permissive para SELINUX=disabled.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
SELINUX=disabled_
# SELINUXTYPE= can take one of these two values:
# default - equivalent to the old strict and targeted policies
# mls     - Multi-Level Security (for military and educational use)
# src     - Custom policy built from source
SELINUXTYPE=default

# SETLOCALDEFS= Check local definition changes
SETLOCALDEFS=0
```

Figura 31- Alteração da Configuração do SELinux

Vale ressaltar que alguns núcleos exigem que seja adicionado ao boot (grub) o parâmetro "selinux=0", como por exemplo, o Kernel UEK (Oracle Unbreakable Enterprise Kernel). Verifica-se o resultado na figura 32.

```
servidorSeguro:~# sestatus
SELinux status:                 disabled
servidorSeguro:~# _
```

Figura 32 - Status desativado do SELinux

O invasor deve se preocupar em neutralizar a ação do ossec, para depois excluir ou adulterar os registros. Os experimentos demonstram que com privilégios de super usuário, um invasor possui autonomia para adulterar os registros e esconder seus rastros, e mesmo com uma sistema de verificação de integridade, as técnicas se mostram efetivas. Porém, um sistema seguro e bem configurado pode amenizar os danos de uma invasão, ou ainda gerar registros que não podem ser excluídos pelo invasor, como registros remotos ou e-mails.

5. RESULTADOS E DISCUSSÕES

A técnica de desabilitar o history com unset apresenta muita eficiência, uma vez que impede do administrador entender quais foram os passos utilizados pelo invasor e quais métodos utilizou para prejudicar o sistema. Uma possível contramedida em relação aos comandos para desabilitar o histórico de comandos, pode ser o desenvolvimento de tarefas agendadas através do cron para manter o tamanho do HISTSIZE, ou ainda desenvolver scripts que adquiram as informações digitadas por super usuários, não somente através do history e enviem e-mails para o administrador, ou ainda que desconecte temporariamente o usuário que utilize o comando unset HISTFILE.

Adulteração dos registros *auth.log*, *wtmp*, *lastlog* são de suma importância para a ocultação dos rastros de um invasor e a técnica usada para a exclusão de seus rastros é muito eficiente e fácil de executar. Uma possível contramedida consiste em utilizar-se de sistema de detecção de intrusão com o intuito de analisar possíveis inconsistências nos registros, além de servidores remotos com o intuito de manter um *backup* dos registros. Pode-se também, acrescentar medidas de proteção de arquivos, como o *chattr* e permissões apenas para grupos específicos. Deve-se também instalar os registros em partições diferentes, com o intuito de evitar que o disco seja completamente cheio caso haja um ataque com *logger*.

A utilização de *rootkits* são ferramentas de muita eficiência e dificilmente um administrador descobriria, sem qualquer ferramenta, onde estão. Se houve qualquer desconfiança sobre o estranho funcionamento de algum processo ou registro, provavelmente haverá a necessidade de reinstalação do sistema. Para se evitar que problemas assim aconteçam é aconselhável ao administrador manter uma imagem do sistema em sua forma mais pura, com o intuito de recuperar o sistema rapidamente, além de instalar um detector de intrusão, que compara o sistema integro com qualquer atividade suspeita verificando a integridade dos arquivos e alertando sobre possível *rootkit* no mesmo.

A técnica de remoção segura é totalmente eficaz, uma vez excluído um arquivo não há mais como recuperá-lo, por isso há extrema urgência em se manter a proteção dos arquivos mais importantes e uma verificação constante se não houve qualquer modificação no sistema.

Ataque a *logrotate* não é uma técnica muito fácil de se aplicar, porém pode atrapalhar o administrador na verificação dos registros. Como contramedida o administrador deve se preocupar em manter uma cópia segura dos registros e manter uma rotina de verificação do tamanho dos registros que estão sendo rotacionados.

A inserção de registros falsificados pode prejudicar outro usuário, livrando a culpa do

invasor, porém a mesma técnica não é tão eficiente. Uma contra medida consiste em uma maior atenção do administrador ao verificar os registros e utilizar-se de ferramentas de interpretação de registro

No ambiente com segurança reforçada foi verificado que o Ossec Hids é um programa essencial para auxiliar profissionais e administradores. Foi verificado que um invasor possui autonomia para adulterar as configurações do ossec hids, quando esse possui o privilégio de super usuário. Entretanto, comprometer o servidor se torna muito mais difícil, uma vez que, o ossec bem configurado, alerta o administrador por e-mail. Com isso, o administrador consegue muito mais controle sobre o sistema, e pode, por sua vez, analisar com mais cuidado os alertas gerados pelo ossec.

O SELinux é essencial para o administrador que deseja reforçar a segurança do sistema operacional Linux. A lógica da política de tomada de decisões provê segurança sobre todos os processos e objetos do sistema, uma vez que de até mesmo o super usuário é tratado pelo SELinux como apenas um usuário comum. Evitando, assim, que os objetos do Linux (arquivos, *devices*, *sockets*, portas, processos) sejam expostos, por exemplo, a brechas de segurança utilizando-se incorretamente o comando de direitos de acesso.

O syslog-ng se apresenta muito eficiente em infraestruturas de segurança baseado na verificação dos registros, pois permite centralizar os registros em um servidor, permitindo implementá-los até mesmo em tabelas de banco de dados, como MySQL, e ser mais confiável que o syslog padrão.

6. CONCLUSÃO

Realizando os experimentos com as técnicas antiforenses computacionais em sistemas de registro Linux com o intuito de ocultação dos rastros deixados por uma invasão, verifica-se que as técnicas antiforenses são efetivas para eliminar rastros em uma invasão.

Verificou-se também que os registros são de extrema importância não somente para a perícia forense, mas para qualquer sistema, pois possuem todas as informações do funcionamento das máquinas. Devido a isso, quando um sistema está comprometido, ou está em processo de comprometimento, um invasor procurará com toda certeza ocultar seus rastros, e seu alvo principal são os registros.

Obtido a conta de super usuário de um sistema, o invasor pode realizar qualquer coisa e não há como saber se ele limpou todos os seus rastros, a não ser que o perito ou administrador conheça muito bem o seu sistema e possua um controle rigoroso do seu funcionamento.

Com poucos comandos e pouco esforço, um invasor pode esconder seus rastros, pois existem muitas ferramentas prontas e bem desenvolvidas. Devido a isso, pode causar grandes prejuízos a um sistema. Todo o impacto gerado nas invasões, a ocultação dos crimes e a importância da informação para a nossa sociedade, confirma a importância das precauções que qualquer administrador deve ter ao gerenciar sistemas, principalmente seus registros, reforçando a importância desse trabalho.

Um ambiente com uma segurança reforçada é essencial para os administradores e profissionais de redes, uma vez que o número de invasores interessados em dados pessoais ou empresariais é cada vez maior. Cuidados com a conta de super usuário devem ser as maiores possíveis, pois um invasor com esses privilégios pode ocultar seus rastros e adulterar os sistemas de segurança instalados no sistema, como demonstrado no estudo de caso com os sistema seguro.

Sintetizando as observações: quando um invasor adquire a conta de super usuário não há dificuldade alguma em esconder suas atividades em um sistema, a não ser que se depare com um sistema fortemente seguro. Mesmo que não exista um sistema totalmente seguro, há medidas que podem facilitar a recuperação de um sistema invadido e amenizar os danos que podem ser causados ao mesmo. Para isso, um administrador deve executar medidas de segurança antes e durante o funcionamento do sistema.

Este trabalho poderia ter continuidade em muitos pontos, tais como uma análise mais completa das segurança nos registros, como o uso de servidores remotos para armazenamento dos mesmos, o uso mais profundo do AIDE, Ossec-hids, SELinux, LIDS e syslog-ng. Poder-se-ia estudar sobre técnicas e vulnerabilidades dos servidores DNS

Recursivos e DNS Spoofing com o intuito de roubo de informações. Analises futuras poderiam verificar mais medidas de segurança proativa e ativa, mais técnicas antiforenses verificando falhas nos sistemas. Verificar medidas de recuperação de danos. Este trabalho poderia se estender para a parte dos crimes de informática aplicadas aos registros ou ocultação de informações. Aprofundar a análise sobre ataques de negação de serviço dos sistemas responsáveis pelos registros ou ainda no próprio núcleo do Linux. Poderia ser feito pesquisas com a população em geral para verificar o uso e conhecimento dos registros. Poderia ser estudado a particularidade dos registros nos principais Sistemas Operacionais Linux, aplicando técnicas em outros além do Debian.

As técnicas antiforense são dinâmicas e estão inovando com o passar do tempo, assim também as medidas de segurança, com isso, o leque para este trabalho também tem a tendência de se ampliar. O fator fundamental que rege esse leque, consiste na necessidade da sociedade e os prejuízos que ela poderia sofrer sem a devida busca por conhecimento e atualização.

7. ANEXOS

ANEXO A - Syslog e Syslogd

1. Syslog

O syslog tem duas funções:

- Liberar os programadores de gerar seu arquivos de log.
- Deixar o administrador do sistema no controle dos logs.

O syslog é constituído de 3 partes:

- O daemon syslogd.
- Openlog, que são rotinas e bibliotecas chamadas para ter acesso ao syslog.
- Logger, que é um comando shell para o usuário enviar entradas para o syslog.

2. Daemon

O syslogd, escrito inicialmente por Eric Allman para *UNIX de kernel com poucos Kbytes, trabalhava com poucas mensagens de log. Os *UNIX modernos com kernel que podem variar de 400K a 16MB (Linux) passaram a produzir uma quantidade muito grande de logs, sem contar os programas (ssh, Postfix, SGBDS, etc), que produzem ainda mais log. Então o daemon do syslog foi divido em dois:

- syslogd - Linux system logging utilities;
- klogd - Kernel Log Daemon.

3. Klogd

O klogd é o daemon responsável por capturar as mensagens de lançadas pelo kernel e guarda suas mensagens em dois locais, no /proc ou usando a "sys_syslog interface". O klogd escolhe qual dos dois vai usar da seguinte maneira: verifica se o /proc está montado, se estiver, utilizará o /proc/kmsg para gravar as mensagens de log, caso contrário utilizará o "system call interface" para mandar as mensagens de erro.

Uma vez que a mensagem é enviada para o syslogd, o klog pode priorizar as mensagens enviadas pelo kernel.

As mensagens enviadas pelo kernel têm a seguinte sintaxe:

<[0-7]> mensagem do kernel

Os valores <[0-7]> são definidos no kernel.h.

O klogd também envia as mensagens para o console toda a mensagem com valor menor que 7. Já as mensagens com valor 7 são tratadas como mensagem de debug.

O problema é que a configuração padrão gera mensagens em demasia na saída de tela, então normalmente se utiliza o klog da seguinte maneira:

klogd -c 0

O comando acima mostra na tela apenas os logs de PANIC.

Kernel Address Resolution: Uma vez que o kernel detecta um erro interno, uma trigger é disparada mostrando todo o conteúdo que estava no processador no momento do erro. O klogd fornece essa facilidade de especificar que função foi chamada e quais variáveis estavam envolvidas no erro.

4. Syslogd

Captura tanto as mensagens do kernel quanto as mensagens do sistema. O syslogd dá suporte para logs remotos, por exemplo, você pode fazer todas as máquinas mandarem um log para uma máquina específica que analisa os logs de todas as máquinas de uma empresa.

5. Os arquivos

Um dos problemas do syslog é a falta de padronização dos arquivos de log, então dependendo da distribuição do seu Linux, pode haver alguma diferença.

Geralmente os arquivos de logs estão no diretório /var/log. Vejamos alguns exemplos dos arquivos de log que podem ser abertos com um editor de texto.

O Quadro 11 apresenta os principais arquivos de log e suas respectivas funcionalidade:

Nome	Descrição
messages	Um dos principais arquivos de log do sistema (kernel/sistema)
syslog	Um dos principais arquivos de log (kernel)
secure	Uso do su, sudo, mudança de senhas pelo root, etc
maillog	Arquivo de log do servidor de e-mail
cron	Log do cron

Quadro 11 - Arquivos de Log e Funcionalidades

O syslog também gera arquivos de contabilidade, que não podem ser abertos por editores de textos e sim por programas especiais (exemplo: "last" para o wtmp).

O Quadro 12 apresenta os principais arquivos de contabilidade e qual sua respectiva funcionalidade:

Nome	Descrição
wtmp	Contabilidade do tempo de conexão
acct/pacct	Contabilidade de processo BSD/Sysv

Quadro 12 - Arquivos de Contabilidade e Funcionalidade

6. Entendo o syslog.conf.

Toda a configuração do syslogd está no arquivo /etc/syslog.conf. A sintaxe básica do syslog.conf é a seguinte:

```
recurso.nível      ação
```

Onde "recurso" é o recurso do sistema que envia a mensagem. São 18 os recursos definidos na maioria das versões, mas o syslog já define 21 (para uso futuro). O Quadro 13 apresenta os principais recursos do syslog.conf e quais os respectivos programas que os utilizam:

Recurso	Programas que utilizam
kernel	O kernel
user	Processos do usuário
mail	mail server
daemon	Daemons do sistema
auth	Autenticação/segurança
cron	Cron
syslog	Mensagens internas do syslog
*	Todos exceto o mark

Quadro 13 - Recursos do Syslog e Programas que os utilizam

O Quadro 14 mostra qual o “nível" e qual sua determinação de grau de severidade do log.

Nível	Significado
emerg	Mensagens Críticas
alert	Situações de emergências
crit	Condições críticas
err	Erros
warning	Mensagem de advertência
notice	Algo que merece uma investigação
info	Informativas
debug	Depuração

Quadro 14 - Nível do log e sua relação com o significado

Para mostras as ações do syslog utilizou-se do Quadro 15.

Ação	Significado
nomedoarquivo	Grava a mensagem no arquivo (path completo)
@nomedohost/ipdohost	Encaminha a mensagem para um syslog em outra máquina
usuário1,usuário2,	Imprime as mensagens na tela do usuário se ele estiver logado

Quadro 15 - Ação do syslog e seu significado

O syslog permite entradas com operador lógico (OR) e wildcards (*,!) da seguinte maneira:

```
recurso.nível;
recurso2.nível4          ação
recurso1,recurso2.nível  ação
*.nível                  ação
recurso.!                Ação
```

Alguns *UNIX aprimoraram seu syslog, os *UNIX derivados do BSD (O Slackware em especial) implementou os qualificadores demonstrados no Quadro 16:

Seletor	Significado
kernel.info	**Seleciona as mensagens com nível info ou mais altos**
kernel.>=info	**O mesmo do kernel.info**
kernel.= info	**Só as mensagens de info**
kernel.!= info	**As mensagens diferentes de info**
kernel.<=info	**As mensagens com o nível < ou = a info**
kernel.<info	**Seleciona as mensagens com prioridade menor que info**
kernel.>info	**Seleciona as mensagens com prioridade maior que info**

Quadro 16 - Qualificadores do syslog do Unix e seus significados

Alguns syslogs também implementam o m4, que consiste em um pré-processador de macro. Vejamos um exemplo:

```
auth.notice  ifdef('LOGSERVER','/var/log/xpto','@LOGSERVER')
```

Essa linha direciona a mensagem para o /var/log/xpto se o LOGSERVER não estiver definido.

7. Login remoto

Para montar um servidor de log temos que verificar se temos que fazer duas coisas:

- Iniciar o syslogd com a opção -r, para que o syslog receba mensagens enviadas pela rede;
- Fazer o daemon escutar a porta 514 (/etc/services), tanto no servidor quanto no cliente.
- Pronto, basta configurar o syslog.conf do servidor e dos clientes.

8. Questão de segurança

O uso de um servidor de log pode deixar seu servidor susceptível à ataques D.O.S. Para evitar possíveis D.O.S:

- Definir no firewall quem poderá enviar mensagens;
- Fazer com que os logs não fiquem na partição raiz do sistema (evitando que o sistema encha apenas com os logs);
- Definir uma "quota" para os arquivos de log.

8. GLOSSÁRIO

PALAVRA	DEFINIÇÃO
Adware	Do inglês Asymmetric Digital Subscriber Line. Software especificamente projetado para apresentar propagandas. Constitui uma forma de retorno financeiro para aqueles que desenvolvem software livre ou prestam serviços gratuitos. Pode ser considerado um tipo de spyware, caso monitore os hábitos do usuário, por exemplo, durante a navegação na Internet para direcionar as propagandas que serão apresentadas.
Artefato	De forma geral, artefato consistem em qualquer informação deixada por um invasor em um sistema comprometido. Pode ser um programa ou script utilizado pelo invasor em atividades maliciosas, um conjunto de ferramentas usadas pelo invasor, log ou arquivos deixados em um sistema comprometido, a saída gerada pelas ferramentas do invasor etc.
Assinatura Digital	Código utilizado para verificar a integridade de um texto ou mensagem. Também pode ser utilizado para verificar se o remetente de uma mensagem é mesmo quem diz ser.
Atacante	Pessoa responsável pela realização de um ataque. Veja também Ataque.
Ataque	Tentativa, bem ou mal sucedida, de acesso ou uso não autorizado a um programa ou computador. Também são considerados ataques as tentativas de negação de serviço.
Backdoor	Programa que permite a um invasor retornar a um computador comprometido. Normalmente este programa é colocado de forma a não ser notado.
Boato	E-mail que possui conteúdo alarmante ou falso e que, geralmente, tem como remetente ou aponta como autora da mensagem alguma instituição, empresa importante ou órgão governamental. Por meio de uma leitura minuciosa desse tipo de e-mail, normalmente, é possível identificar em seu conteúdo mensagens absurdas e muitas vezes sem sentido.

Bot	Programa que, além de incluir funcionalidades de worms, sendo capaz de se propagar automaticamente através da exploração de vulnerabilidades existentes ou falhas na configuração de softwares instalados em um computador, dispõe de mecanismos de comunicação com o invasor, permitindo que o programa seja controlado remotamente. O invasor, ao se comunicar com o bot, pode orientá-lo a desferir ataques contra outros computadores, furtar dados, enviar spam, etc
Botnets	Redes formadas por diversos computadores infectados com bots. Podem ser usadas em atividades de negação de serviço, esquemas de fraude, envio de spam etc.
Cavalo de Troia	Programa, normalmente recebido com um "presente" (por exemplo, cartão virtual, álbum de fotos, protetor de tela, jogo, etc), que além de executar funções para as quais foi aparentemente projetado, também executa outras funções normalmente maliciosas e sem o conhecimento do usuário.
Código Malicioso	Termo genérico que se refere a todos os tipos de programas que executam ações maliciosas em um computador. Exemplos de códigos maliciosos são os vírus, worms, bots, cavalos de troia, rootkits, etc.
Comprometimento	Veja Invasão
Conexão segura	Conexão que utiliza um protocolo de criptográfica para a transmissão de dados, como, por exemplo, HTTPS ou SSH
Criptografia	Ciência ou arte de escrever mensagens em forma cifrada ou em código. É parte de um campo de estudos que trata das comunicações secretas
DDoS	Do inglês Distributed Denial of Service. Ataque de negação de serviço distribuído, ou seja, um conjunto de computadores é utilizado para tirar de operação um ou mais serviços ou computadores conectados à Internet. Veja Negação de Serviço.
DoS	Do inglês Denial of Service. Veja Negação de Serviço
Endereço IP	Este endereço é um número único para cada computador conectado à Internet, composto por uma sequência de 4 números que variam de 0 até 255, separados por ".". Por exemplo, 192.168.2.3.
Engenharia Social	Método de ataque onde uma pessoa faz uso de persuasão, muitas vezes abusando da ingenuidade ou confiança do usuário, para obter informações que podem ser utilizadas para ter acesso não autorizado a computadores ou informações.
Exploit	Programa ou parte de um programa malicioso projetado para explorar uma vulnerabilidade existente em um software de computador
Falsa Identidade	Ato onde o falsificador atribui-se identidade ilegítima, podendo se fazer passar por outra pessoa, com objetivo de obter vantagens indevidas, como, por exemplo, obter crédito, furtar dinheiro de contas bancárias das vítimas, utilizar cartões de crédito de terceiros, entre outras.

Firewall	Dispositivo constituído pela combinação de software e hardware. Utilizado para dividir e controlar o acesso entre redes de computadores.
GnuPG	Conjunto de programas gratuito e de código aberto, que implementa criptografia de chave única, de chaves pública e privada e assinatura digital.
Harvesting	Técnica utilizada por spammers, que consiste em varrer páginas Web, arquivos de listas de discussão, entre outros, em busca de endereços de e-mail.
Hoax	Veja Boato.
HTML	Do inglês HyperText Markup Language. Linguagem universal utilizada para estruturação de páginas da Internet
HTTP	Do inglês HyperText Transfer Protocol. Protocolo usado para transferir página Web entre um servidor e um cliente.
Malware	Do inglês Malicious software (software malicioso). Veja Código Malicioso.
HTTPS	Quando utilizado como parte de uma URL, especifica a utilização de HTTP com algum mecanismo de segurança, normalmente o SSL.
IDS	Do inglês Intrusion Detection System. Programa, ou um conjunto de programas, cuja função é detectar atividades maliciosas ou anômalas
IEEE	Acrônimo para Insitute of Electrical and Electronics Engineers, uma organização composta por engenheiros, cientistas e estudantes, que desenvolvem padrões para a indústria de computadores e eletroeletrônicos.
Invasão	Ataque bem sucedido que resulte no acesso, manipulação ou destruição de informações em um computador
Invasor	Pessoa responsável pela realização de uma invasão(comprometimento). Veja também Invasão
Keylogger	Programa capaz de capturar e armazenar as teclas digitadas pelo usuário no teclado de um computador. Normalmente, a ativação do keylogger é condicionada a uma ação prévia do usuário, como por exemplo, após o acesso a um site de comércio eletrônico ou Internet Banking, para captura de senhas bancárias ou número de cartões de crédito
Log	Registro de atividades gerado por programas de computador. No caso de logs relativos a incidentes de segurança, eles normalmente são gerados por firewalls ou por IDSs.
Malware	Do inglês Malicious software (software malicioso). Veja Código Malicioso.
Negação de Serviço	Atividade maliciosa onde o atacante utiliza um computador para tirar de operação um serviço ou computador conectado à Internet

PGP	Do inglês Pretty Good Privacy. Programa que implementa criptografia de chave única e assinatura digital.
Phishing	Também conhecido como phishing scam. Mensagem não solicitada que se passa por comunicação de uma instituição conhecida, como um banco, empresa ou site popular, e que procura induzir usuários ao fornecimento de dados pessoais e financeiros. Inicialmente, este tipo de mensagem induzia o usuário ao cesso a páginas fraudulentas na Internet. Atualmente, o termo também se refere à mensagem que induz o usuário à instalação de códigos maliciosos
Rootkit	Conjunto de programas que tem como finalidade esconder e assegurar a presença de um invasor em um computador comprometido. É importante ressaltar que o nome rootkit não indica que as ferramentas que o compõem são usadas para obter acesso privilegiado (root ou Administrador) em um computador, mas sim para manter o acesso privilegiado em um computador previamente comprometido
Scam	Esquemas ou ações enganosas e/ou fraudulentas. Normalmente, tem como finalidade obter vantagens financeiras
Scan	Técnica normalmente implementada por um tipo de programa, projetado para efetuar varreduras em redes de computadores
SSH	Do inglês Secure Shell. Protocolo que utiliza criptografia para cesso a um computador remoto, permitindo a execução de comandos, transferência de arquivos, entre outros.
Trojan Horse	Veja cavalo de Tróia.
Vírus	Programa ou parte de um programa de computador, normalmente malicioso, que se propaga infectando, isto é, inserindo cópias de si mesmo e se tornando parte de outros programas e arquivos de um computador. O vírus depende da execução do programa ou arquivo hospedeiro para que possa se tornar ativo e dar continuidade ao processo de infecção.

REFERÊNCIAS

AIDE. **Advanced Intrusion Detection Environment.** Endereço: http://aide.sourceforge.net/. Acesso em 10 de abril de 2013.

BARRETO, G.L. **Utilização de técnicas antiforenses para garantir a confidencialidade**. 2009, pp. 10

CARPANEZ, Juliana. **Conheça os crimes virtuais mais comuns.** Folha de 07/01/2006. Disponível em: www1.folha.uol.com.br/folha/informatica/ult124u19455.shtml. Acesso em 25 de março de 2013.

CARROLL, Ovie L., BRANNON, Stephen K., SONG, Thomas. **Computer Forensics: Digital Forensic Analysis Methodology.** United States Attorneys ' Bulletin. Volume 56, Número 1, Janeiro de 2008.

CENTRO DE ESTUDO, RESPOSTA E TRATAMENTO DE INCIDENTES DE SEGURANÇA NO BRASIL (CERT.BR). **Estatística dos Incidentes Reportados ao CERT.br.** Endereço: http://www.cert.br/stats/incidentes/2010-jul-sep/tipos-ataque-acumulado.html. Acesso em 11 de abril de 2013.

CORRÊA, Gustavo Testa. **Aspectos jurídicos da internet**. São Paulo: Saraiva. 2000.

DEBIAN, Wiki. **SELinux.** Endereço http://wiki.debian.org/SELinux. Acesso em 11 de maio de 2013.

GULIK, Dirk-Willem van. **Ataque de negação de serviço devido a vulnerabilidade em faixa do cabeçalho Apache HTTPD 1.3/2.x.** Disponível em: http://article.gmane.org/gmane.comp.apache.announce/58. Acesso em 20 de abril de 2013.

HOEPERS, Cristine. STEDING-JESSEN, Klaus. **Análise e Interpretação de Logs.** Endereço: http://www.cert.br/docs/palestras/nbso-gter15-tutorial2003.pdf. Acesso em 22 de março de 2013.

FARMER, Dan, VENEMA, Wietse. **Pericia Forense Computacional: Teoria e Prática Aplicada**. Pearson Prentice Hall. 2007.

FARMER, Dan; VENEMA, Wietse; **Forensic Computer Analysis: An Introduction**; Dr. Dobb's Journal. Setembro de 2000.

FREITAS, Andrey. R. de. **Pericia Forense Aplicada à Informática**. Trabalho de Curso de Pós-Graduação "Lato Sensu" em Internet Security. IBPI. 2003.

HATCH, Brian. Brian Hatch, James B. Lee, George Kurtz: Tradução Daniel Vieira. – **Segurança Contra Hackers Linux.** São Paulo: Futura. 2003.

ICCyber 2010. **Combate aos Crimes Cibernéticos**. Endereço: http://www.crimesciberneticos.com/2010/09/combate-aos-crimes-ciberneticos-e.html. Acesso em 02 de abril de 13.

IDG NOW! **Petrobras perde dados sigilosos em furto de computadores**, 14 de fevereiro de 2008. Endereço: http://idgnow.uol.com.br/. Acesso em 11 de abril de 2013.

LUEST, Severin. **Netlog.** Endereço: http://netlog.sourceforge.net/. Aceso em 05 de maio de 2013.

MAIN, P. C. Van Oorschot. **Software Protection and Application Security: Understanding the Battleground**. Procedings of the International Course on State of the Art and Evolution of Computer Security and Industrial Cryptography. 2003.

MCCLURE, Stuart, SCAMBRAY, Joel, KURTZ, George. **Hackers exposto: segredos e soluções para a segurança de redes.** Tradução de Daniel Vieira – Rio de Janeiro: Elsevier, 2003. 2ª Reimpressão.

MILAGRE, José Antônio. **Perícia digital e computação forense.** Revista Visão Jurídica. Disponível no endereço: http://revistavisaojuridica.uol.com.br/advogados-leis-jurisprudencia/47/imprime170142.asp. Acesso em 02 de abril de 2013.

NOBLETT, Michael G.; **Report of the Federal Bureau of Investigation on**

development of forensic tools and examinations for data recovery from computer evidence; Proceedings of the 11th INTERPOL Forensic ScienseS ymposium. 1995.

NOBLETT, Michael G., POLLITT, Mark M., PRESLEY, Lawrence A**. Recovering and Examining Computer Forensic Evidence**. Forensic science communications. Outubro de 2000 volume 2 Número 4.

ORACLE. **Oracle® HTTP Server Administrator's Guide**. Disponível em: http://www.di.unipi.it/~ghelli/didattica/bdldoc/B19306_01/server.102/b14190/servlog.htm. Acesso em 12 de abril de 2013.

OSSEC, Host-Based Intrusion Detection Systems. **Open Source Host-based Intrusion Detection System.** Disponível em: http://www.ossec.net/doc/. Acesso em 15 de abril de 2013.

PEREIRA, Evandro, FAGUNDES, Leonardo, NEUKAMP, Paulo, LUDWIG, Glauco, KONRATH, Marlom. **Forense Computacional: fundamentos, tecnologias e desafios atuais.** VII Simpósio Brasileiro em Segurança da Informação e de Sistemas computacionais. Rio de janeiro. Agosto de 2007.

RENAESP. **Revista da Rede Nacional de Altos Estudos em Segurança Pública, Edição 2008.**

RODRIGUES, R. **Brasil é líder em vírus que roubam dados bancários, diz pesquisa**. 2010. Disponível em: http://idgnow.uol.com.br/seguranca/2010/08/24/brasil-e-lider-em-virus-que-roubam-dados-bancarios-diz-pesquisa/. Acesso em 15 de abril de 2013.

RODRIGUES, Thalita Scharr, FOLTRAN, Dierone César. **Análise de Ferramentas Forenses na Investigação Digital.** 5º Encontro de Engenharia e Tecnologia dos Campo Gerais. Outubro de 2010.

SILVA, Bruno Siqueira da. **Análise de estudos sobre investigação forense computacional em sistema Unix alterado por rootkits.** URI Santo Ângelo – Universidade Regional Integrada do Alto Uruguai e das Missões Departamento de Engenharias e Ciência da Computação. 2011.

SOUZA, Carlos André Evangelista de. **Técnicas Antiforense em Desktops**. Pró-Reitoria de Pós-Graduação e Pesquisa Lato Sensu em Perícia Digital. Trabalho de Conclusão de Curso. Brasília. Distrito Federal. 2012.

STRAUHS, Faimara do Rocio. **Gestão do Conhecimento em Laboratório Acadêmico: Proposição de Metodologia**. 2003. 480 f. Tese (Doutorado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2003.

THOMPSON, Kerry. **Logsurfer.** Endereço: http://www.crypt.gen.nz/logsurfer/. Acesso em 05 de maio de 2013.

TOLEDO, Alex Sander de Oliveira, SOUZA, Robert de. **Perícia Forense Computacional: Análise de Malware em Forense computacional utilizando de sistema operacional GNU/Linux.** PÓS EM REVISTA. Pg. 172.

TOMÁS, Eliane Maria Cordeiro. **Crimes Informáticos: Legislação brasileira e técnicas de forense computacional aplicadas à essa modalidade de crime**. Artigo acessível no endereço: http://www.artigos.etc.br/crimes-informaticos-legislacao-brasileira-e-tecnicas-de-forense-computacional-aplicadas-a-essa-modalidade-de-crime.html. Acesso em 02 de abril de 2013.

VARGAS, R. **Perícia Forense Computacional e metodologias para obtenção de evidências**. 2007

VARGAS, R. **Processos e Padrões em Perícia Forense Aplicado a Informática**. Trabalho de Conclusão de Curso, Bacharelado em Sistemas de Informação, Faculdade Metodista Granbery, Juiz de Fora, Minas Gerais, 2006.

VIANNA, Túlio Lima. **Fundamentos de Direito Penal Informático.** Rio de Janeiro: Forense, 2003.

WIKIPEDIA. **Crime.** Disponível em: http://pt.wikipedia.org/wiki/Crime. Acesso em: 25 de março de 2013.

WIKIPEDIA. **Auditoria**. Disponível em: http://pt.wikipedia.org/wiki/Auditoria. Acesso em: 25 de março de 2013.

ZANELLATO, Marco Antônio. **Condutas ilícitas na sociedade digital**. In Caderno Jurídico da Escola Superior do Ministério Público de São Paulo, ano II, nº. IV, – julho de 2002.