REVELA
Periódico de Divulgação Científica da FALS
Ano V - Nº XII- DEZ / 2011 - ISSN 1982-646X

TÉCNICAS DE PERÍCIA FORENSE COMO FERRAMENTAS DE PREVENÇÃO DE INCIDENTES DE SEGURANÇA

Marcelo Teixeira de Azevedo (ITA-SP)
exemplo@fals.com.br

Ana Lucia Pegetti (ITA-SP/FALS)
exemplo@fals.com.br

Kleyton Maia dos Santos (ITA-SP)
exemplo@fgals.com.br

Resumo: O presente artigo estuda a perícia forense aplicada à informática e suas implicações para a prevenção de acidentes. Para tanto, são apresentados estudos de caso relevantes, que possibilitaram analisar eventos em que a perícia fez-se necessária e buscar as práticas que elevassem as camadas de segurança desse sistema, mitigando riscos e possibilitando ações proativas por parte de seus administradores. Por fim, são identificadas as medidas que possibilitam a prevenção de incidentes de segurança em uma rede computacional. Conclui-se que, com as práticas estudadas, o profissional de segurança da informação atuante em grupos de prevenção e resposta a incidentes poderá aumentar as camadas de segurança do ambiente computacional e, assim, prevenir incidentes, sabendo coletar e tratar uma evidência.

Palavras-chave: Perícia forense. Leis e crimes digitais. Ferramentas de segurança.

Abstract: The present paper studies the forensic skill applied to the computer science and its implications for the prevention of accidents. For so much, relevant case studies are presented, which made possible to analyze events in that the skill was made necessary, and to look for the practices to elevate the safety layers of that system, mitigating risks and making possible proactive actions on the part of its administrators. Finally, they are identified the measures that make possible the prevention of safety's incidents in a computer network. It is concluded that, with the studied practices, the professional of information safety active in groups of prevention and attack to incidents can increase the safety layers of the computer environment, and, like this, to prevent incidents, knowing how to collect and to treat an evidence.

Keywords: Forensic skill. Digital law and crimes. Safety tools.

1 INTRODUÇÃO

1.1 Preliminares

O tema escolhido aborda a perícia computacional, que é uma das áreas de atuação dos profissionais de segurança da informação, cujo panorama mundial está diretamente envolvido com o uso da tecnologia e de sistemas informatizados, os quais precisam ser seguros e garantir disponibilidade, integridade e confidencialidade das informações. Entretanto, quando algo de anormal acontece, como se deve proceder? Como saber, diante do emaranhado de bytes, o que constitui tal anormalidade? E o que ocorre quando, por trás dessa anormalidade, há uma ação ilícita?

Nesse sentido, os profissionais da área de segurança precisam, além de estar preparados para lidar com essas situações, ser capazes de ajudar a solucionar um evento, bem como de instruir as pessoas a seu redor a agir e proteger os dados e as evidências. Precisam, além disso, saber proteger os sistemas de informação, prever as ações de cibercriminosos e reagir diante de ameaças aos sistemas de informação, para que os criminosos não se prevaleçam de um sistema sob os cuidados de um profissional na segurança da informação.

1.2 Delimitação do Tema

O tema está delimitado no estudo de técnicas de perícia computacional e suas formas de atuação. Além disso, o estudo de caso realizado proporciona visualizar uma das formas de aplicação das técnicas e ferramentas da perícia, uma vez que tal aplicação dá-se conforme o objetivo do caso, seja ele judicial ou corporativo.

Por fim, apresenta-se uma proposta de métodos que um profissional na área de segurança da informação pode se valer para evitar a efetivação de um crime, no que se refere a práticas ilícitas nos sistemas computacionais.

1.3 Justificativa para a Pesquisa

O crescente uso e a grande aplicação de sistemas informatizados e da tecnologia trazem riscos, como o mau uso da tecnologia por parte de pessoas que criam sistemas motivadores de práticas ilícitas, como também por parte daquelas que, mesmo não os criando, utilizam-se deles. Cabe ressaltar, ainda, que cada nova tecnologia desenvolvida proporciona a possibilidade de seu uso para práticas ilícitas. Nesse sentido, neste estudo, será apresentado o panorama das práticas de perícia, com foco principalmente nas medidas para coibir e prevenir as ações ilícitas nos sistemas computacionais.

1.4 Referencial Teórico

Para o desenvolvimento deste trabalho, algumas obras são centrais para embasar e conduzir aos objetivos propostos. Dentre essas obras, tem-se a literatura relativa à área de Perícia Forense Aplicada à Informática, de autoria de Freitas (2006), que fornece, a partir de métodos e ferramentas estudados pelo autor, um embasamento sobre a evolução do assunto ao longo da história.

A obra de Tocchetto e Espindula (2005), por sua vez, é utilizada por apresentar informações a respeito de práticas e procedimentos metodológicos para um trabalho na área de perícia forense, sendo complementado pela literatura de Ng (2007), que informa os fatores motivadores para uma equipe de perícia corporativa, bem como desenvolve conceitos e práticas para compor um modelo de equipe forense computacional. Além disso, é responsável tanto pela intervenção nas corporações ligadas a casos de investigações forenses quanto pelas políticas e desenvolvimentos de práticas que objetivam aumentar a segurança de um sistema corporativo.

Nesse contexto, Farmer e Venema (2005) mostram não só a forma como informações forenses podem ser localizadas, como também algumas ferramentas específicas para práticas no sistema UNIX. Além disso, apresentam uma base de como as informações persistem e como podem sofrer com ações deliberadas, através de uma análise de procedimentos de perícia e solução de problemas.

Ainda, Alberto Filho (2010) expõe aspectos da prova pericial e da atividade do perito durante o processo judicial, além de apontar as principais modificações no ordenamento jurídico, sendo, assim, fonte de informação ao tema associado à perícia, à prova pericial e à atuação do perito.

1.5 Problema

O trabalho de perícia tem como dever buscar respostas sobre o que aconteceu em um incidente. Dessa forma, o perito, com requisitos técnicos para isso, precisa conhecer a fundo sua área de atuação, assim como as ferramentas que tem ao seu dispor. Além disso, é importante para o perito a organização de seu trabalho e orientação para alcançar os objetivos. Por esses motivos, este trabalho não se resume apenas a um estudo da aplicabilidade e dos métodos da perícia computacional, mas visa a contextualizar o panorama atual da perícia e indicar um método de atuação, a fim de melhorar a eficácia do trabalho do perito.

Isso porque, mesmo atuando corretamente na perícia, o profissional de segurança da informação precisa saber como prevenir incidentes; evitando, minimizando e coibindo, assim, os riscos aos sistemas de informação.

Nesse sentido, espera-se que o presente trabalho mostre que em cada incidente existe uma rica fonte de informação para se analisar e identificar técnicas ou ferramentas, com o objetivo de aumentar a segurança de seus sistemas de informação.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Perícia

Do latim *peritia*, o vocábulo é definido por Ferreira (2009) como qualidade de perito, vistoria especializada, além de ser citado no glossário do Instituto Brasileiro de Avaliações e Perícias de Engenharia de São Paulo (IBAPE/SP, 2011) como atividade incumbida a profissional especializado, legalmente habilitado a esclarecer um fato, ou

seja, descobrir fatores ou estados motivadores, alegações de direito ou propriedade da coisa que é objeto de litígio ou processo. Além disso, no direito, a perícia é produtora de provas, sendo os peritos qualificados eleitos pelos juízes.

Definida no Código de Processo Civil, Lei nº 5.869, de 11 de janeiro de 1973, art. 420, a prova pericial, por sua vez, consiste em exame, vistoria ou avaliação, sendo considerada, no decorrer do processo civil, um dos momentos mais cautelosos na estrutura de composição do texto. Nesse contexto, considerando que o momento de avaliação é classificatório de valor ao fato ou objeto, a vistoria é o momento de análise da característica de bem imóvel e o exame, referente a bem móvel e pessoas. Para tanto, o profissional perito na área em questão utiliza técnicas e ferramentas desenvolvidas para tal, fazendo-se necessário o esclarecimento de detalhes técnicos.

Cumpre informar que, no presente trabalho, a perícia será abordada no escopo tecnológico, uma vez que a crescente e contínua complexidade dos sistemas de tecnologia exige, cada vez mais, a intervenção técnica para esclarecer a veracidade de fatos ou circunstâncias.

2.2 Forense

Do latim *forense*, o vocábulo é definido por Ferreira (2009) como aquilo que se refere ao foro judicial, sendo relativo aos tribunais. Dessa forma, a técnica forense diz respeito à aplicação de recursos científicos em um processo jurídico, sendo que essas práticas valem-se da perícia para alcançar os objetivos de prova, visto que muitas provas não são perceptíveis a olho nu nem estão disponíveis a pessoas desprovidas de conhecimentos técnicos necessários.

Ressalte-se que, com o passar do tempo, a criminologia desenvolve-se cada vez mais; isso porque, uma vez que os criminosos usam as mais criativas alternativas para driblar a lei, é necessário utilizar recursos científicos em busca de uma maior eficiência, visando a tomar as devidas precauções para o correto julgamento desses casos, mesmo que, aparentemente, no caso dos meios eletrônicos, não haja pistas nem rastros físicos.

2.3 A Perícia Forense Computacional

A perícia forense computacional é definida por Freitas (2006) como a aplicação de conhecimento de informática e técnicas de investigação com a finalidade de obtenção de evidências, além de, para o autor, ser uma área relativamente nova e em grande ascensão; justamente por isso tornou-se uma prática importante nas corporações e polícias, que utilizam resultados científicos e matemáticos estudados na ciência da computação.

Nesse sentido, a finalidade motivadora da perícia forense computacional é, com base nas suas práticas científicas, coletar e analisar as informações disponíveis no meio eletrônico, para que tais informações (antes intangíveis) passem a compor a certeza manifesta dos fatos que serão utilizados como provas. Essas informações serão de grande valia, também, para o ambiente corporativo e judicial, pois, mesmo que não exista um processo judicial formal, as práticas da perícia forense demonstrarão o que realmente aconteceu e as intenções de um fato que tenha se desenrolado num ambiente eletrônico.

Ainda, segundo Bustamante (2006), a perícia forense pode ser definida como coleção e análise de dados de um computador, sistema, rede ou dispositivos de armazenamento, de forma que sejam admitidos em juízo, sendo que as evidências que um criminalista ou *expert* (também chamado perito) encontra geralmente não podem ser vistas a olho nu, dependendo de ferramentas e meios para a sua obtenção. Nesse contexto, cabe ao profissional de informática coletar as evidências e produzir um laudo pericial com as evidências e técnicas abordadas na coleta.

A computação forense continua em crescimento, tendo em vista que, cada vez mais, a sociedade faz uso dos computadores, exigindo profissionais nas esferas corporativas e judiciais, nas quais estão presentes questões sobre procedimentos ilegais em computadores. Atualmente, a busca por evidências nos ambientes computacionais tem se tornado imprescindível para a força policial, pois pode esclarecer muitos crimes efetuados via computador.

A cada dia, os dispositivos de armazenamento e acesso à internet estão se tornando menores, mais baratos, mais rápidos, com maior portabilidade e com uso

amplamente difundido. Desse modo, na internet, não há como simplesmente isolar o local do crime, tampouco identificar de imediato a origem e autoria do crime, ou seja, é possível identificar a data e hora do evento, bem como alguns rastros de conexões em *logs*, mas não o autor do crime, como seria possível com um exame de DNA. Isso se justifica devido à natureza dinâmica da rede, de modo que um local usado na internet para cometer crimes pode ser diferente ou ausente no dia seguinte. Assim, não se pode tomar conclusões baseadas apenas em dados colhidos através de *softwares*.

3 ESTUDOS DE CASO

3.1 Invasão a Servidor – Empresa ACME

3.1.1 Descrição

Neste estudo de caso, será apresentada uma situação simulada, baseada nos casos referentes a *ssh* do Projeto Honeypots da Honeynet (HONEYNET, 2010), que é uma organização de pesquisa líder em segurança internacional, dedicada a investigar os recentes ataques e a desenvolver ferramentas *open source* para melhorar a segurança na internet.

Assim, considerar-se-á que foi detectado, pelos administradores de rede da empresa fictícia ACME, comportamento instável do servidor *Domain Name Server* (DNS) primário, alto tráfego de rede do servidor na porta do serviço *Secure Shell* (SSH) e um anormal consumo de processamento do servidor.

3.1.2 Procedimento de análise

Feita a fase de levantamento e identificadas as tarefas do servidor no contexto da rede atual, concluiu-se que algo de errado estava ocorrendo, de modo que foram coletados os materiais necessários para a perícia através de um conjunto de discos com os *softwares* de coleta e análise e uma unidade externa devidamente sanitizada para a coleta da imagem de disco do servidor. Cabe ressaltar que o acesso físico ao servidor foi limitado apenas ao técnico que iria efetuar a perícia e que não foram necessários

registros adicionais, como fotos, pois a perícia foi realizada internamente; porém, em um caso oficial, esses procedimentos são impreteríveis.

A duplicação da unidade de disco do servidor foi feita com a ferramenta *dd*, utilizando o comando *dd if=/dev/sdb1 of=/media/3268509868505ca3/imagem.dd*, no qual 'if=' representa a unidade de origem a ser copiada e 'of=', o arquivo de imagem que irá conter as informações do disco bit a bit. Para melhor visualização, a saída do comando *dd* pode ser verificada na Figura 1.

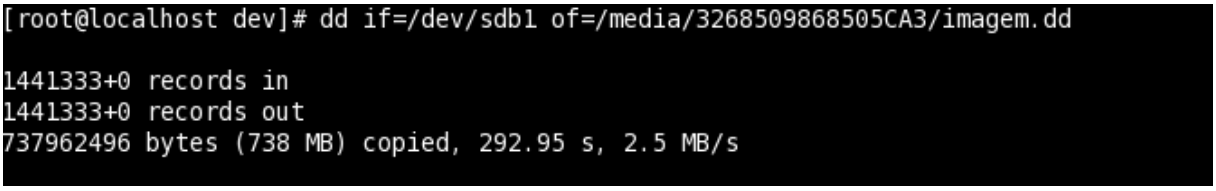

Figura 1 – Saída do comando *dd*.

Uma vez feita a cópia, foi necessário calcular o *hash* para garantir uma forma de validar a evidência, comprovando que ela não foi alterada. Para tanto, foi utilizada a ferramenta *sha512sum*, com a qual é possível calcular o *hash* em *sha* 512 bits. Na Figura 2, são apresentados o comando e a saída, obtendo como resultado o valor de *hash* calculado para o arquivo 'imagem.dd', localizado na unidade '/media/3268509868505CA3/'; tal valor deve ser armazenado para futuras comparações entre o disco original e a evidência para garantir a integridade.

Figura 2 – Saída do comando *sha512sum*.

Já na fase de coleta, iniciou-se a captura do tráfego com a ferramenta *Wireshark*, o que não auxilia muito no diagnóstico, pois, durante uma seção de conexão via *ssh*, todo o conteúdo trafega de forma criptografada. Contudo, como resultado do uso dessa ferramenta, na Figura 3, é possível identificar a seção estabelecida e dados como: origem, destino protocolo e porta de comunicação.

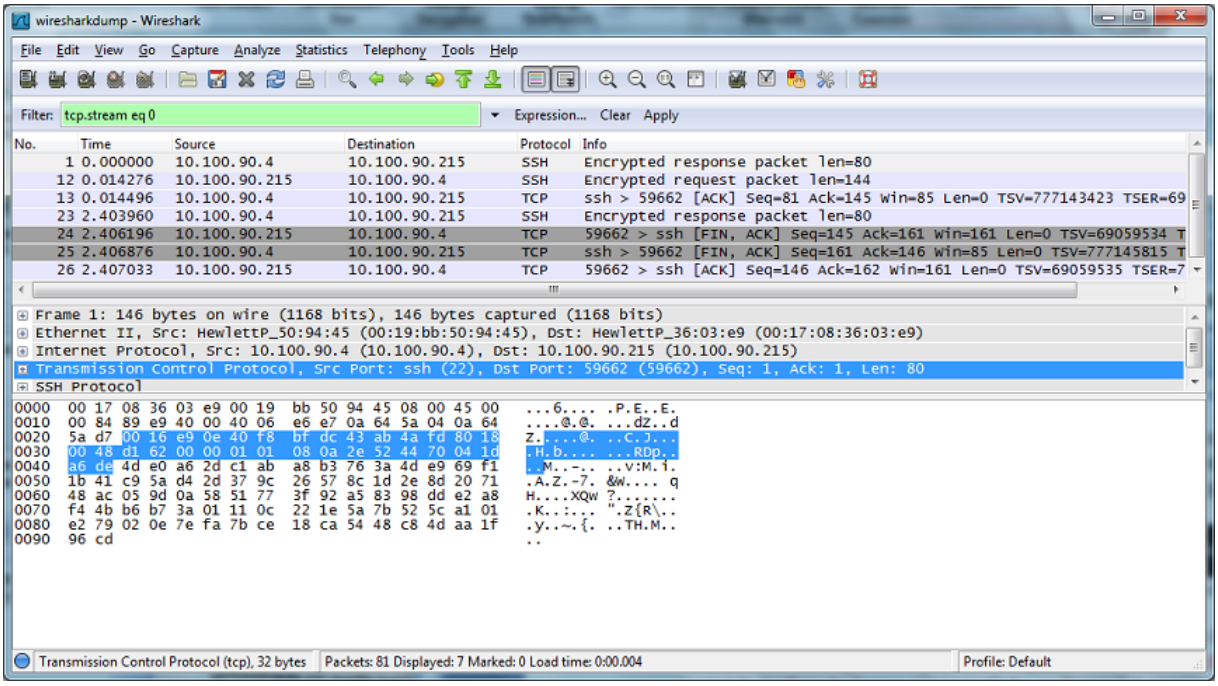

Figura 3 – Tráfego de rede da seção *ssh*.

Por fim, após a captura, foi interrompido o serviço *ssh*, com objetivo de cessar o alto tráfego de rede e a carga de processamento no servidor. Foram coletados, também, *logs* dos serviços de *ssh* e autenticação, bem como foi identificada uma conexão remota estabelecida no serviço de *ssh* que não era esperada.

3.1.3 Diagnóstico

Na fase de análise, detectou-se a exploração do serviço de *ssh* para acesso e troca de informações e comandos com o servidor vítima; considerando que o acesso foi consumado com uma conta administrativa de *root*.

Nesse sentido, a Figura 4 demonstra o resultado do comando *tail* para listar o final do arquivo de *log*, sendo possível analisar as várias tentativas de autenticação no serviço *ssh* com usuário administrativo oriundas do mesmo endereço atacante.

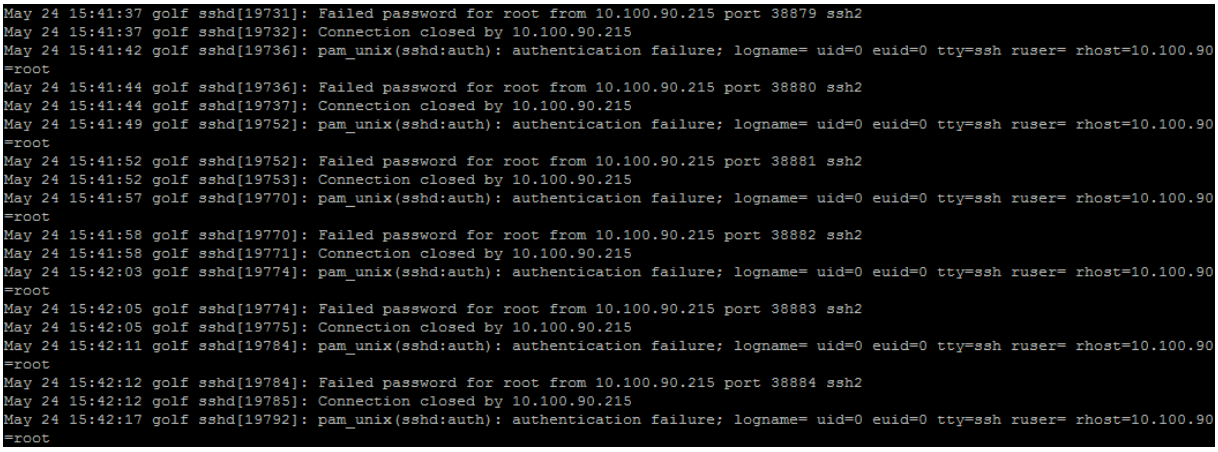

Figura 4 – Trecho de conteúdo arquivo de *log /var/log/secure*.

Já a Figura 5 mostra intervalo do *log*, através do comando *tail*, no qual a conexão com o serviço foi bem-sucedida, ou seja, momento em que o ataque obteve sucesso.

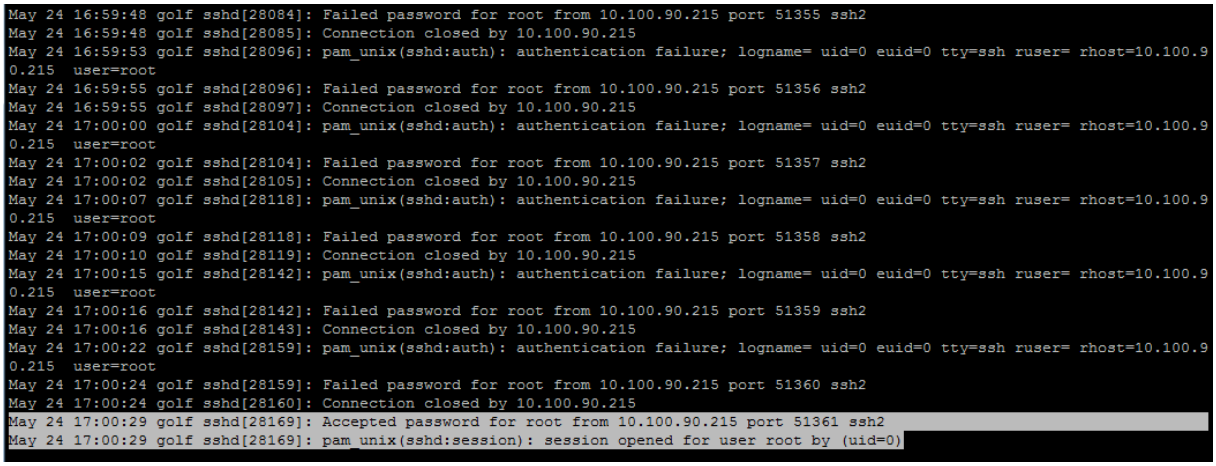

Figura 5 – *Log* em destaque com conexão em */var/log/secure*.

Identificada a intrusão no sistema, foram analisados os arquivos criados ou alterados pelo atacante. Para tanto, foi feita a análise dos *mactimes* com as ferramentas *mac-robber* e *mactime*, sendo necessário gerar a base de dados para análise com a ferramenta *mac-robber*, através do comando *mac-robber /root/exploit > exploit.mactimes*, no qual '/root/exploit' representa o diretório a ser analisado e 'exploit.mactime', o arquivo de destino dos dados de *mactimes*, formando uma base de

dados para ser processada pela ferramenta, a qual irá compor a linha de tempo dos arquivos, conforme demonstrado na Figura 6.

```
[root@localhost exploit]# mactime -b exploit.mactime -d
Date,Size,Type,Mode,UID,GID,Meta,File Name
Wed May 25 2011 09:42:24,0,ma..,-rw-r--r--,0,0,0,/root/exploit/exploitlnx.sh
Wed May 25 2011 09:42:33,0,ma..,-rw-r--r--,0,0,0,/root/exploit/exploitweb.sh
Wed May 25 2011 09:53:51,0,..c.,-rw-r--r--,0,0,0,/root/exploit/exploitlnx.sh
Wed May 25 2011 09:53:51,0,..c.,-rw-r--r--,0,0,0,/root/exploit/exploitweb.sh
Wed May 25 2011 09:55:43,27,mac.,-rw-r--r--,0,0,0,/root/exploit/exploitws.sh
Wed May 25 2011 10:18:52,0,mac.,-rw-r--r--,0,0,0,/root/exploit/exploit.mactime
```

Figura 6 – Saída da linha de tempo do comando *mactime*.

Analisando todos os arquivos de imagem com a ferramenta *autopsy* (Figuras 7 e 8), verificou-se que os arquivos da pasta '/root/exploit' foram os únicos criados e alterados após o ataque; essa pasta continha alguns arquivos de *scripts* maliciosos, que seriam posteriormente utilizados pelo atacante, porém a invasão foi frustrada antes que isso ocorresse, o que foi comprovado pelo fato de os atributos de execução desses *scripts* ainda não terem recebido marcação.

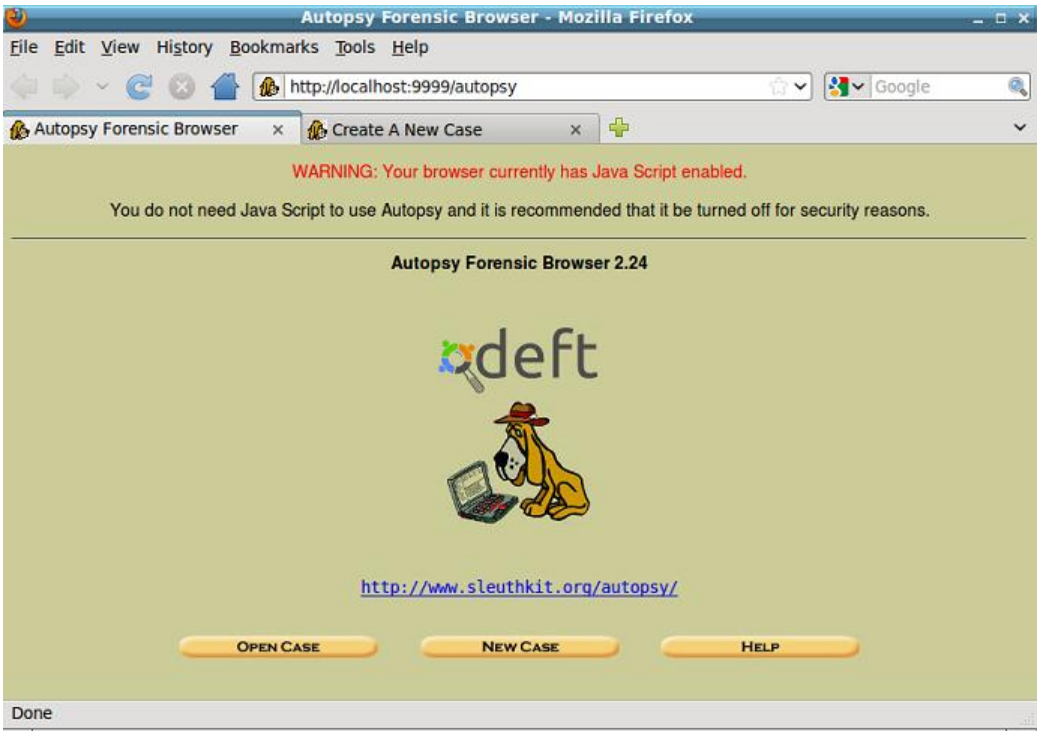

Figura 7 – Tela inicial da ferramenta *autopsy*.

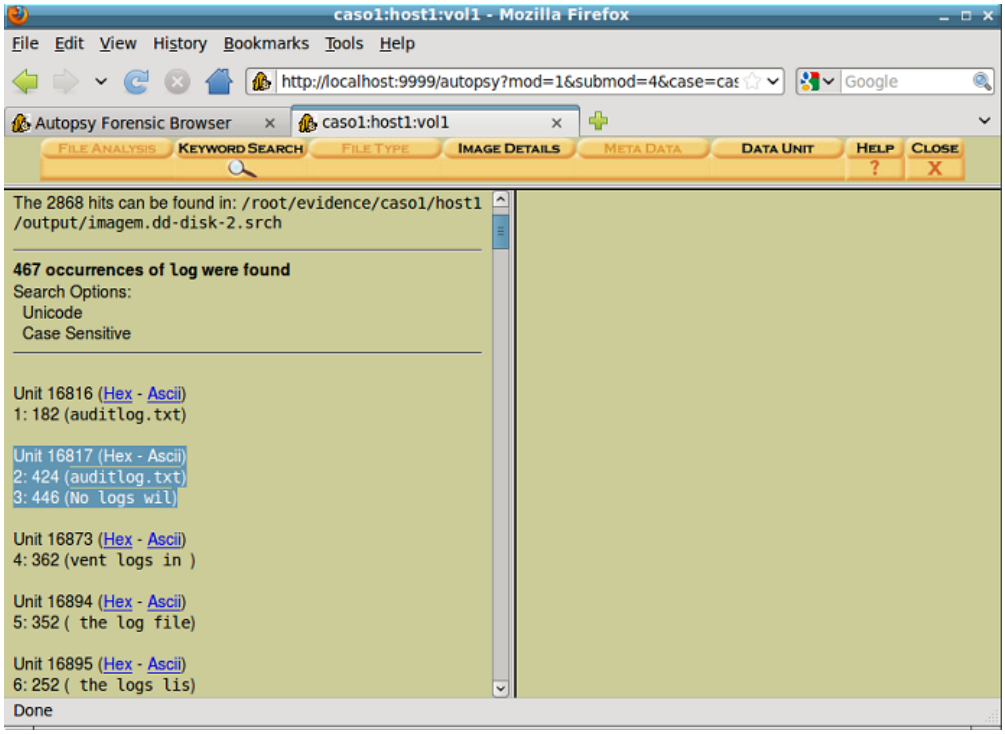

Figura 8 – Pesquisa por arquivos de *log* em *Keyword Search* da ferramenta *autospy*.

3.1.4 Resultados

Com a análise, foi possível diagnosticar uma tentativa bem-sucedida de acesso ao servidor através do serviço *ssh*, por uma técnica chamada força bruta, definida pelo Centro de Atendimento a Incidentes de Segurança (CAIS) da Rede Nacional de Ensino e Pesquisa (RNP, 2010) como sucessões de tentativa e erro utilizando um conjunto de duplas usuário/senha para tentar obter acesso ao serviço.

Para melhor entendimento e análise dos resultados, simulou-se o ataque utilizando o seguinte *script* através da distribuição Linux Fedora Security:

```
#!/bin/bash
ARQSENHAS=senhas.txt
BINSSH=/usr/bin/ssh
BINPASS=/usr/local/bin/sshpass
j=0
#
for i in `cat $ARQSENHAS`;
do
```

```
        $BINPASS -p$i $BINSSH root@0.0.0.0 "echo 'Conexao bem sucedida'"
        j=$((j+1));
        echo $j;
done
#
```

Nesse *script*, foram utilizadas duas ferramentas para conexão: o *sshpass* e o cliente de conexão remota *ssh*. A primeira é destinada a prover senha para uma autenticação não interativa em uma ferramenta que necessite de uma autenticação interativa, como no exemplo; ou seja, quando se necessita de uma entrada via teclado para a autenticação, o *sshpass* simula essa entrada, possibilitando a automação da tarefa de tentativa de *login*. Assim, através do uso desse *script*, foi possível testar várias senhas para o mesmo usuário e servidor, sem intervenção humana, agilizando a tarefa de tentativa e erro no *login*.

Na Figura 9, é possível verificar as várias tentativas de *login*, sendo que, na de número 10, a conexão foi bem-sucedida, possibilitando utilizar a senha presente no arquivo ‘senhas.txt’, na linha 10, para o acesso remoto via *ssh*. É interessante ressaltar que esse arquivo deve conter as possíveis senhas e ser gerado através de uma ferramenta de dicionário de senhas, para melhorar as tentativas do acesso.

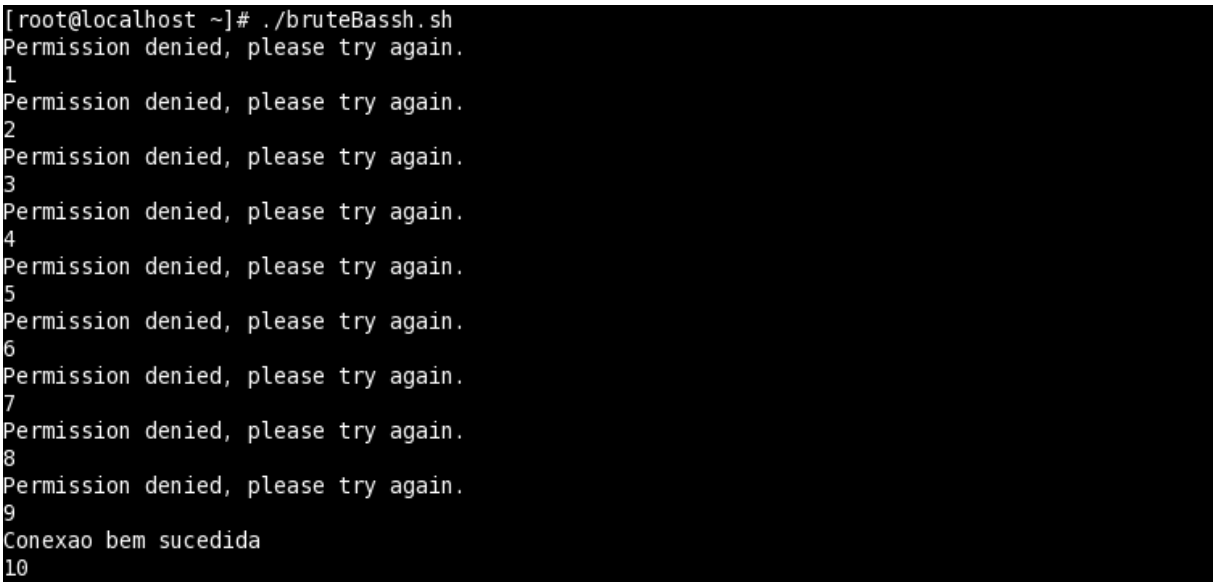

Figura 9 – Saída do *script* de força bruta no *ssh*.

Já na Figura 10, são apresentados os *logs* do servidor vítima, nos quais as tentativas são registradas pelo serviço de conexão remota *sshd*. Na última linha, tem-se o momento da conexão bem-sucedida no serviço durante o teste.

```
82452 May 24 15:41:22 golf sshd[19706]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.100.90.2
      5  user=root
82453 May 24 15:41:23 golf sshd[19706]: Failed password for root from 10.100.90.215 port 51710 ssh2
82454 May 24 15:41:23 golf sshd[19707]: Connection closed by 10.100.90.215
82455 May 24 15:41:28 golf sshd[19723]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.100.90.2
      5  user=root
82456 May 24 15:41:30 golf sshd[19723]: Failed password for root from 10.100.90.215 port 51711 ssh2
82457 May 24 15:41:30 golf sshd[19724]: Connection closed by 10.100.90.215
82458 May 24 15:41:35 golf sshd[19731]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.100.90.2
      5  user=root
82459 May 24 15:41:37 golf sshd[19731]: Failed password for root from 10.100.90.215 port 38879 ssh2
82460 May 24 15:41:37 golf sshd[19732]: Connection closed by 10.100.90.215
82461 May 24 15:41:42 golf sshd[19736]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.100.90.2
      5  user=root
82462 May 24 15:41:44 golf sshd[19736]: Failed password for root from 10.100.90.215 port 38880 ssh2
82463 May 24 15:41:44 golf sshd[19737]: Connection closed by 10.100.90.215
82464 May 24 15:41:49 golf sshd[19752]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.100.90.2
      5  user=root
82465 May 24 15:41:52 golf sshd[19752]: Failed password for root from 10.100.90.215 port 38881 ssh2
82466 May 24 15:41:52 golf sshd[19753]: Connection closed by 10.100.90.215
82467 May 24 15:41:57 golf sshd[19770]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.100.90.2
      5  user=root
82468 May 24 15:41:58 golf sshd[19770]: Failed password for root from 10.100.90.215 port 38882 ssh2
82469 May 24 15:41:58 golf sshd[19771]: Connection closed by 10.100.90.215
82470 May 24 15:42:03 golf sshd[19774]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.100.90.2
      5  user=root
82471 May 24 15:42:05 golf sshd[19774]: Failed password for root from 10.100.90.215 port 38883 ssh2
82472 May 24 15:42:05 golf sshd[19775]: Connection closed by 10.100.90.215
82473 May 24 15:42:11 golf sshd[19784]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.100.90.2
      5  user=root
82474 May 24 15:42:12 golf sshd[19784]: Failed password for root from 10.100.90.215 port 38884 ssh2
82475 May 24 15:42:12 golf sshd[19785]: Connection closed by 10.100.90.215
82476 May 24 15:42:17 golf sshd[19792]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.100.90.2
      5  user=root
82477 May 24 15:42:19 golf sshd[19792]: Failed password for root from 10.100.90.215 port 38885 ssh2
82478 May 24 15:42:19 golf sshd[19793]: Connection closed by 10.100.90.215
82479 May 24 15:42:24 golf sshd[19803]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.100.90.2
      5  user=root
82480 May 24 15:42:27 golf sshd[19803]: Failed password for root from 10.100.90.215 port 38886 ssh2
82481 May 24 15:42:27 golf sshd[19804]: Connection closed by 10.100.90.215
82482 May 24 15:42:32 golf sshd[19828]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.100.90.2
      5  user=root
82483 May 24 15:42:34 golf sshd[19828]: Failed password for root from 10.100.90.215 port 38887 ssh2
82484 May 24 15:42:34 golf sshd[19829]: Connection closed by 10.100.90.215
82485 May 24 15:42:39 golf sshd[19830]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.100.90.2
      5  user=root
82486 May 24 15:42:41 golf sshd[19830]: Failed password for root from 10.100.90.215 port 38888 ssh2
82487 May 24 15:42:41 golf sshd[19831]: Connection closed by 10.100.90.215
82488 May 24 15:42:46 golf sshd[19832]: Accepted password for root from 10.100.90.215 port 38889 ssh2
```

Figura 10 – *Log* do servidor *sshd*.

Uma vez que a tentativa foi bem-sucedida, conclui-se que o servidor vítima estava de fato vulnerável a esse tipo de ataque, com uma senha fraca, descoberta no processo de tentativa e erro.

Cumpre informar que, nessa fase, o perito deve compor um laudo expondo o parecer técnico e apresentá-lo a quem for necessário, seja para a autoridade judicial ou os responsáveis pela empresa ou setor.

4.2 Prevenindo Incidentes com Base no Estudo de Caso

4.2.1 Prevenções no caso de invasão de servidor

A prevenção de incidentes em sistemas que provem acesso remoto requer uma análise detalhada da infraestrutura de rede, com o intuito de configurá-la corretamente para atender aos requisitos mínimos de segurança e funcionalidades do serviço. Para tanto, os principais itens a serem considerados na análise são: as questões de proteção e configuração correta do servidor que hospeda o serviço de acesso remoto; a proteção dos dados que trafegam na rede, com uso de sistemas de *firewalls* e IDS/IPS/NIPS; e os métodos de autenticação e identificação da identidade dos usuários que logam no acesso remoto. A seguir, serão apresentadas algumas questões para a prevenção desses incidentes.

4.2.1.1 DMZ

O modelo DMZ, sigla para de *DeMilitarized Zone* ou zona desmilitarizada, implementa na rede uma zona separada da rede externa (internet – *Wan*) e da rede interna (rede local – *Lan*). Esse termo, de origem militar, classifica uma região sem atividades militares de defesa ou ataque, por vezes entre dois territórios em conflito; aplicado a redes, considera que ela contém serviços com exposição à internet, como de acesso remoto, correio, *sites* e protocolo de transferência de arquivos FTP, os quais devem ficar separados da rede local. Assim, quando um ataque for bem-sucedido nesse ambiente, está confinado à rede DMZ e não irá impactar em toda a rede da organização.

Esse conceito também é aplicado para separar setores e usuários-chave em diferentes níveis de proteção, além de ditar que a rede possui um nível de segurança intermediário, o qual, entretanto, não a colocaria como única para armazenar dados críticos para a empresa, devendo, dessa forma, ser estudada e bem projetada antes de sua implementação.

Normalmente, o conceito de DMZ é aplicado utilizando *firewalls* e suas políticas para separar os tráfegos entre as redes, sendo as seguintes regras e políticas de segurança aplicadas:

- comunicação da rede externa para DMZ é aceita;
- comunicação da rede externa para a rede interna é negada;
- comunicação da rede interna para DMZ é aceita;
- comunicação da rede interna para a rede externa é aceita;
- comunicação da DMZ para a rede interna é negada;
- comunicação originada na DMZ para a rede externa é negada.

Cabe ressaltar, ainda, que uma DMZ previne um incidente mais abrangente, pois, se um serviço for comprometido nela, existem outras barreiras a transpassar antes de se alcançar os demais serviços da rede. No contexto do estudo de caso, o serviço *ssh*, uma vez violado, ficaria com o acesso isolado somente àquela rede DMZ, dando uma melhor possibilidade de combate e proteção.

4.2.1.2 Servidor de log

Em redes que crescem em sistemas e equipamentos, o gerenciamento de *logs* de forma descentralizada irá demandar cada vez mais tempo e recursos, uma vez que cada *log*, com suas particularidades e seu local de alocação, complica ainda mais o processo de análise e monitoramento. Para resolver esse problema, utiliza-se a centralização dos *logs*, que são arquivos com registros sobre serviços ou máquinas e que auxiliam na administração dos recursos, com informações sobre acessos, problemas, avisos, alertas e demais informações para a administração do ambiente.

O conceito consiste em definir um servidor responsável por armazenar todos os *logs*, configurando as demais máquinas da rede para que seus *logs* sejam enviados para esse servidor. Essa função de centralização pode ser aplicada para vários fins, como, por

exemplo, auditoria ou proteção de *logs* de alguma invasão; ainda, no servidor de *logs*, é possível obter uma melhor análise da informação, para estudar o desempenho e o dimensionamento das máquinas, analisando, por exemplo, o serviço que recebe mais acessos e precisa ser equilibrado. Para tanto, é utilizada a ferramenta Linux *Rsyslog*, que coleta e centraliza os *logs* de máquinas Linux e Windows.

Para uma ação de prevenção de incidentes, é muito importante que as informações de *logs* e os registros estejam preservados e acessíveis para análise e perícia, pois é necessário saber o que está ou estava acontecendo durante uma ação maliciosa; a tentativa de apagar ou encobrir rastros irá afetar os *logs* e registros, porém se esses registros forem também armazenadores em outro local, como no servidor de *logs*, isso preservará as evidências, contribuindo para uma coleta de informações sobre os eventos de forma eficaz.

4.2.1.3 Restrição e controle de acesso remoto

Neste item, analisa-se tanto as medidas para uma correta configuração que, aplicadas ao servidor *sshd*, oferecem maior segurança e robustez ao serviço, quanto outras alternativas. É importante destacar que os conceitos podem ser aplicados a outros modelos de serviço para acesso remoto, como Remote Desktop, Vpn, Logmein, Teamviewr e afins, bastando saber a forma de configuração destes.

No caso do serviço *sshd*, as configurações devem ser feitas no arquivo '/etc/ssh/sshd_config', no qual se encontram os parâmetros de funcionamento do serviço, lembrando que, após qualquer alteração, deve-se recarregar o serviço, para que as alterações tenham efeito. Também, é recomendada a troca da porta-padrão de acesso do serviço de acesso remoto, pois ataques automatizados vão explorar justamente essa porta; no *ssh*, a porta-padrão é a 22, que pode ser alterada através do parâmetro 'Port XX', onde 'xx' representa o número da porta em que o serviço irá aguardar a conexão. Caso se utilize um *firewall*, nele também haverá uma regra permitindo a essa porta comunicar-se.

Além disso, devem ser verificadas as seguintes recomendações:

- não permitir autenticação com usuário sem senha, ativando o seguinte parâmetro na configuração do *sshd*: *PermitEmptyPasswords no*;
- usar senhas fortes, complexas e com comprimento mínimo de 15 caracteres, ativando a opção de utilizar o pacote *Pluggable Authentication Module* (PAM), que auxilia nas regras de autenticação, exigindo requisitos de senhas e tentativas limitadas, através do parâmetro *UsePAM Yes*;
- limitar a quantidade de tentativas de senha errada, derrubando a conexão após a tentativa, ativando a opção: *MaxAuthTries 3.* Nesse exemplo, limitar a três as tentativas de senhas; caso, na terceira, a senha esteja errada, a conexão é finalizada e terá de ser reiniciada;
- não permitir que o usuário *root* ou outro que faça parte do grupo *root* efetue o *login* diretamente. É conveniente, nesse caso, que se logue como usuário restrito, utilizando o comando *su* para elevar o privilégio como *root*, caso seja necessário, pois tarefas simples e testes de funcionalidades podem ser executados como usuário comum, não afetando e alterando nada. Utilizar os seguintes parâmetros: *PermitRootLogin no*, *AllowUsers usuraiocomum1 usuariocomun2*, *AllowGroups gruposshdousuariocomun*;
- utilizar apenas a última versão do protocolo *ssh*, que, atualmente, é a 2, ativando o parâmetro: *Protocol 2*;
- utilizar regras de *firewall* para permitir conexão na porta do serviço de acesso remoto apenas por IPs conhecidos e para restringir ainda mais o serviço de acesso remoto.

5 CONCLUSÃO

Com este trabalho, foi possível analisar a aplicação da perícia forense à informática, em que cada caso representa um desafio ímpar, devido à crescente evolução dos ambientes computacionais. As dificuldades para os peritos e suas perícias também se elevam com o panorama atual da legislação brasileira, que abrange, em parte, as tecnologias atuais. Sobre isso, Marco Aurélio Greco (2000) conclui que o

direito vem sofrendo igualmente os reflexos das modificações profundas no mundo por conta dos avanços tecnológicos e da globalização.

As novas tecnologias, cada qual com seus padrões e dignas de um estudo individual, aparentam representar um volume de informação muitas vezes maior do que os recursos e ferramentas da perícia forense computacional, porém é possível identificar que o método proposto, seguindo uma sequência lógica e organizada, possibilita ao perito atuar de forma mais eficiente e eficaz.

Além disso, é importante ressaltar que os meios eletrônicos ainda despertam uma falsa sensação de anonimato e impunidade, sentimento confirmado pelo crescimento dos crimes eletrônicos nas pesquisas, mesmo à margem de grandes dificuldades em torno de legislações e tecnologias.

Nesse contexto, este estudo demonstrou que os poderes competentes e as organizações têm alternativas para buscar a segurança para seus ambientes computacionais. Com as práticas aqui estudadas, o profissional de segurança da informação atuante em grupos de prevenção e resposta a incidentes poderá aumentar as camadas de segurança do ambiente computacional e, assim, prevenir incidentes, sabendo coletar e tratar uma evidência.

Por fim, o caso apresentado constitue rica fonte de informação para uma análise objetivando a prevenção de incidentes, uma vez que trazem à tona aspectos relevantes, tais como: os ambientes de rede precisam ser organizados e estruturados, ter redundância e dispor de acompanhamento, monitoramento, manutenção e segurança adequada. Também, revelam que os sistemas precisam ser testados, atualizados e corrigidos, sendo, para tanto, utilizada uma gama de técnicas, metodologias, procedimentos e ferramentas, a fim de prevenir incidentes e riscos à segurança dos ambientes computacionais.

REFERÊNCIAS

FREITAS, Andrey Rodrigues. **Perícia forense aplicada à informática**. Rio de Janeiro: Brasport, 2006.

ALBERTO FILHO, Reinaldo Pinto. **Da perícia ao perito**. Niterói: Ímpetus, 2010.

BUSTAMANTE, Leonardo. Computação forense: preparando o ambiente de trabalho. **Uol**, julho, 2006. Disponível em: <http://imasters.uol.com.br/artigo/4335/forense/computacao_forense_-_preparando_o_ambiente_de_trabalho/>. Acesso em: 04 maio 2009.

FARMER, Dan; VENEMA, Wietse. **Perícia forense computacional**: teoria e prática aplicada. São Paulo: Prentice Hall, 2005.

GRECO, Marco Aurélio. **Internet e direito**. 2. ed. São Paulo: Dialética, 2000.

HONEYNET. **Projeto Honeypots**. Disponível em: <http://www.honeynet.org>. Acesso em: 20 mar. 2010.

NG, Reginaldo. **Forense computacional corporativa**. Rio de Janeiro: Brasport, 2007.

REDE NACIONAL DE ENSINO E PESQUISA (RNP). **Centro de Atendimento a Incidentes de Segurança (CAIS)**. Disponível em: <http://www.rnp.br/cais/>. Acesso em: 29 mar. 2010.

TOCCHETTO, Domingos; ESPINDULA, Alberi. **Criminalística procedimentos e metodologias**. Porto Alegre: Cleusa dos Santos Novak, 2005.