UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

SIDNEY ROBERTO DA SILVA WEBBA

PROCEDIMENTOS COMPUTACIONAIS NO AUXÍLIO À PERÍCIA FORENSE APLICADA EM WEB BROWSERS

CRICIÚMA, DEZEMBRO DE 2010

SIDNEY ROBERTO DA SILVA WEBBA

PROCEDIMENTOS COMPUTACIONAIS NO AUXÍLIO À PERÍCIA FORENSE APLICADA EM WEB BROWSERS

Trabalho de Conclusão de Curso apresentado para obtenção do Grau de Bacharel em Ciência da Computação da Universidade do Extremo Sul Catarinense.

Orientador: Prof. MSc. Paulo João Martins

CRICIÚMA, DEZEMBRO DE 2010

PROCEDIMENTOS COMPUTACIONAIS NO AUXÍLIO À PERÍCIA FORENSE APLICADA EM WEB BROWSERS

Submetido ao corpo docente do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Profa. MSc. Ana Claudia Garcia Barbosa
Coordenadora do Curso de Ciência da Computação

Banca Examinadora:

Prof. MSc. Paulo João Martins (UNESC)
Orientador

Prof. MSc. Gustavo Bisognin (UNESC)

Prof. MSc. Rogério Antônio Casagrande (UNESC)

À minha mãe, por ter sido a força motivadora para a realização deste sonho. Aos meus familiares e amigos que sempre torceram por mim.

AGRADECIMENTOS

Primeiramente a SONANGOL (Sociedade Nacional de Petróleos de Angola) pela oportunidade concedida, a UNESC (Universidade do Extremo Sul Catarinense) pelo seu apoio imprescindível, a todos os professores pelo carinho e suporte mostrados durante esta jornada, e ao meu orientador, professor Paulo João Martins pela ajuda, apoio e idéias.

Agradeço de maneira especial a minha mãe, Amélia Carneiro, pelo seu amor incondicional e esforço constante para me proporcionar o melhor. A Helder Carneiro pelo incentivo e motivação proporcionados, e a Catila Machado que sempre esteve a meu lado, torcendo por mim.

Não posso deixar de agradecer a todos os Angolanos residentes em Criciúma, pelo carinho fraterno, em especial aos amigos: Mauro Alves, Vera Ruth, Ferraz Manuel, Tiago Figueira, Aguinaldo Cristiano, Aniceto de Carvalho, Heidy Ramos, Nídia Monteiro e André Gonçalves.

Por último, e de maneira mais importante agradeço a Deus, por me ter proporcionado tantas bênçãos.

Obrigado, a todos!

"Ciência da Computação está tão relacionada aos computadores quanto a Astronomia aos telescópios, Biologia aos microscópios, ou Química aos tubos de ensaio. A Ciência não estuda ferramentas. Ela estuda como nós as utilizamos, e o que descobrimos com elas." Edsger H. Dijkstra.

RESUMO

Objetivo: analisar e aplicar os procedimentos de perícia forense computacional, com foco na coleta e análise de evidências em web browsers, bem como, contribuir socialmente aumentando o leque de pesquisas sobre o tema. **Métodos:** pesquisa bibliográfica; elaboração de um estudo de caso fictício simulando a condução de uma pericia forense computacional na presente universidade; utilização da metodologia SOP aplicando as suas 6 etapas: autorização e preparação, identificação, coleta e preservação, exame e análise, documentação e reconstrução da cena do crime. **Resultados:** conseguiu-se estudar e aplicar os conceitos de pericia forense computacional, analisando com sucesso muitos dos arquivos de cache, cookies, histórico de navegação e outros, dos browsers Internet Explorer e Firefox, utilizando-se das ferramentas Pasco, Galleta, Web Historian, Firefox3Extractor, Mozilla Cache View e PasswordFox. Ocasionalmente ocorreram falhas ao trabalhar com determinadas ferramentas, como por exemplo, a ferramenta Firefox3Extractor que falhou ao converter arquivos de cache do Firefox, bem como ao mostrar as senhas e nomes de usuário salvos pelo mesmo browser. Provas periciais foram encontradas em alguns dos computadores investigados, mas a análise de muitos outros, mostrou-se inconclusiva.

Palavras-chave: Segurança; Crimes Digitais; Perícia Forense; Web Browsers.

ABSTRACT

Objective: to analyse and apply the procedures of forensic computing analysis, focusing on collecting and analysing evidences in web browsers, and to contribute socially, by increasing the range of research on the topic. **Methods:** literature search; preparation of a fictional case study simulating the execution of a computer forensics analysis in the current university; use of the SOP methodology by applying its six steps: authorization and preparation, identification, collection and preservation, examination and analysis, documentation and reconstruction of the crime scene. **Results:** it was successfully achieved the study and apply of the concepts of computer forensics, by analysing many of the cache, cookies and browsing history files, and others as well, of the browsers Internet Explorer and Firefox, with the tools Pasco, Galleta, Web Historian, Firefox3Extractor, Mozilla Cache View and PasswordFox. Occasionally failures occurred when working with certain tools, such as the tool Firefox3Extractor, that failed to convert cache files of Firefox, as well as to show the user names and passwords saved by the same browser. Forensics evidences were found in some of the analysed computers, but the analysis of many others, was inconclusive.

Keywords: Security, Digital Crimes, Forensics Analysis, Web Browsers.

LISTA DE ILUSTRAÇÕES

LISTA DE SIGLAS E ABREVIATURAS

Advanced Research Projects Agency (ARPA)

Advanced Research Projects Agency Network (ARPANET)

American Standard Code for Information Interchange (ASCII)

Centro de Estudos Respostas e Tratamento de Incidentes de Segurança no Brasil (CERT.br)

Comitê Gestor da Internet no Brasil (CGI.br)

Digital Forensics Research WorkShop (DFRWS)

Domain Name Servers (DNS)

Eletronic Numeric Integrator and Calculator (ENIAC)

File Transfer Protocol (FTP)

Firefox 3 Extractor (F3E)

HyperText Markup Language (HTML)

HyperText Transfer Protocol Secure (HTTPS)

HyperText Transport Protocol (HTTP)

International Hi-Tech Crime and Forensics Conference (IHCFC)

International Organization on Computer Evidence (IOCE)

Internet Engineering Task Force (IETF)

Internet Explorer (IE)

Internet Protocol (IP)

Internet Service Providers (ISPs)

Microwave Communications, Inc (MCI)

National Center for Supercomputer Applications (NCSA)

National Science Foundation (NSF)

National Science Foundation Network (NSFNET)

Network News Transfer Protocol (NNTP)

Nordic Network on Software Architecture Research (NOSAR)

Projeto de Lei (PL)

Request For Comments (RFC)

Scientific Working Group on Digital Evidence (SWGDE)

Secure Sockets Layer (SSL)

Standard Generalized Markup Language (SGML)

Standard Operating Procedures (SOP)

Structured Query Language (SQL)

TCP/IP (Transmission Control Protocol/Internet Protocol)

Transmission Control Protocol (TCP)

Uniform Resource Locator (URL)

Universidade do Extremo Sul Catarinense (UNESC)

União das Repúblicas Socialistas Soviéticas (URSS)

Web Hypertext Application Technology Working Group (WHATWG)

World Wide Web (WWW)

World Wide Web Consortium (W3C)

eXtensible Hypertext Markup Language (XHTML)

eXtensible Markup Language (XML)

SUMÁRIO

1 INTRODUÇÃO

Nos últimos anos, as tecnologias em geral e principalmente os dispositivos de informática, ganharam um papel de destaque na sociedade tornando-se parte integrante da vida das pessoas. Com a popularização da Internet muitas atividades que antes requeriam um esforço físico considerável, podem hoje ser realizadas pelo computador, com o simples clique de um botão, de maneira cômoda mas ainda insegura.

A visualização de conteúdo na Internet é feita por meio da utilização de *browsers*, termo em inglês para navegadores, que são programas que habilitam os seus usuários a interagirem com páginas em HyperText Markup Language (HTML), hospedadas em um servidor Web. A maioria deles hoje, agrega a utilização de *plugins* e tecnologias como Java, Javascript e ActiveX, entre outras, permitindo-os receber código ativo pela rede, ou seja, o simples fato de se visitar um site já é o suficiente para receber um *script* malicioso que possa ter acesso a qualquer parte do sistema computacional (memória, discos, rede, entre outros) e realizar ações ilegais, que vão desde roubo de informações, à danificação do mesmo.

Segundo dados do Comitê Gestor da Internet no Brasil (CGI.br) o número de computadores e de uso da rede no Brasil continua aumentando. Pela primeira vez, desde que a pesquisa começou a ser realizada em 2005, atingiu-se um número de 54 milhões de usuários, e 60 milhões de pessoas que já haviam utilizado o serviço no período de três meses anteriores à realização da pesquisa.

Acompanhando essa evolução, os crimes comuns vêm se tornando cada vez mais tecnologicamente avançados. Fraudes eletrônicas, roubos de identidade, espionagem industrial, transmissão de pornografia infantil, pedofilia e incidentes de segurança convencionais como vírus, *worms*, *phishing*, *hacking* (casual ou direcionado), são alguns exemplos de crimes que ocorrem na Internet.

No Brasil, o órgão responsável por receber, analisar e responder a incidentes de segurança das redes conectadas a *web* é conhecido como Centro de Estudos Respostas e Tratamento de Incidentes de Segurança no Brasil (CERT.br), e segundo eles, o número de episódios vem aumentando consideravelmente a cada ano, sendo que desde Janeiro à Dezembro de 2009 houve um aumento de aproximadamente 38% quando comparado com o número total de ocorrências reportadas em 2008 (CERT.br, 2010b). Uma das razões para tal acontecimento é a impunidade que acontece no mundo cibernético.

Percebe-se portanto, a importância da definição de procedimentos que permitam a condução de uma análise em sistemas comprometidos, que tem por objetivo responder a questões como: quando aconteceu o incidente, como o mesmo foi realizado, quem foi o responsável, porque aconteceu e onde aconteceu.

Com essa nova era, as pistas deixadas pelos infratores ao cometer um crime, como anotações em um bloco de papel, objetos pessoais, entre outros, estão se transformando em informações guardadas em CDs e discos rígidos, ou seja, evidências digitais. E, surge dentro desse contexto a análise forense, como uma ciência capaz de assegurar que as manipulações de tais evidências sejam aceitas em juízo.

Embora os esforços das autoridades competentes tenham aumentado consideravelmente nos últimos anos, pouco se discute dentro da comunidade acadêmica sobre como proceder numa avaliação forense, de modo a que se consiga encontrar evidências eletrônicas suficientemente boas, para serem usadas em processos criminais.

Este trabalho propõe-se então, a analisar e estudar o processo forense em si, focando-se nas técnicas e softwares usados atualmente para coletar e analisar evidências oriundas de *web browsers*.

1.1 OBJETIVO GERAL

Analisar e aplicar os procedimentos de perícia forense computacional na busca por evidências em *web browsers*.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos a serem alcançados são:

a) abordar os aspectos de segurança em *web browsers*;
b) compreender e aplicar os conceitos sobre perícia forense computacional;
c) examinar e descrever quais informações um *web browser* armazena localmente durante uma seção de acesso a Internet;
d) analisar e delinear as ferramentas *open source* e de software livre usadas na busca e análise de evidências em *web browsers*;
e) observar os tipos de evidências obtidas em diferentes *web browsers*.

1.3 JUSTIFICATIVA

Não é incomum a resolução de crimes com o uso de provas como: impressões digitais, pegadas, documentos de papel e outros itens tangíveis extraídos da cena do crime. Hoje, entretanto, com a popularização da tecnologia, a evidência digital também pode ser utilizada para este fim. Podendo muitas vezes ser mais proveitosa que uma impressão digital (SCHWEITZER, 2003).

Como exemplo temos o caso de Neil Entwistle, que em 20 de Janeiro de 2006, foi condenado pelo tribunal da cidade de Massachusetts pelo assassinato de sua mulher e filha.

Segundo o jornalista West (2008) do serviço de notícias americano ValleyWag, uma das principais evidências apresentadas em tribunal foi um registro digital encontrado no computador pessoal do acusado, mostrando que apenas quatro dias antes de cometer o crime, ele havia pesquisado no Google a frase “como matar com uma faca”, o que foi suficiente para mostrar ao júri que o mesmo buscava por conhecimento em como matar de maneira instantânea. Crimes digitais associados a roubos, fraudes e manipulação indevida de dados são detectados diariamente.

Como viu-se com o exemplo acima, até mesmo os crimes violentos não estão livres da revolução tecnológica que está acontecendo. Com o aumento da relevância da evidência digital em processos jurídicos, cresce proporcionalmente a necessidade de que esta seja corretamente manipulada e examinada.

A perícia forense computacional suprindo essa necessidade crescente, mostra-se como uma arma eficaz na luta contra a impunidade que existe atualmente no mundo cibernético.

Existe, entretanto, uma carência bibliográfica dedicada à perícia forense computacional no país. Principalmente direcionada para objetivos específicos, neste caso abordando soluções práticas e mais aprofundadas na análise de evidências em *web browsers*. Observando que nas bibliografias pesquisadas o estudo fica em torno da busca e coleta de evidências e não na análise das mesmas.

Assim, tendo como base a falta de bibliografias e objetos de estudo que a área de perícia forense computacional tem, quando comparada a outras áreas da computação, o presente trabalho pretende através do estudo e análise das técnicas e softwares usados atualmente na prática forense, contribuir para o desenvolvimento da mesma, no que se refere à análise de evidências em *web browsers*.

Também, devido à urgente necessidade de se apresentar uma resposta adequada, consistente e persistente no Brasil para os graves problemas existentes relacionados ao uso indevido da Internet. Por exemplo, aliciamento, produção e difusão de imagens de abuso sexual de crianças e adolescentes, incitação do racismo, intolerância religiosa, homofobia, e estímulo a crimes contra a vida, entre outros crimes atentatórios aos Direitos Humanos (SaferNet Brasil, 2009). O atual trabalho terá uma contribuição social pertinente, uma vez que, a área de forense computacional aplicada à *web browsers* ganhará um estudo ainda pouco existente no país, mas de grande relevância no combate aos crimes digitais.

1.4 ESTRUTURA DO TRABALHO

Como já foi mencionado, a presente pesquisa tem como meta estudar o processo forense em si, focando-se nas técnicas e softwares usados atualmente para coletar e analisar evidências oriundas de *web browsers*. Para tanto, o trabalho foi divido em duas partes: a primeira, contemplando a fundamentação teórica dos objetos de estudo, e a segunda, constituindo a parte prática que simulará o uso de ferramentas forenses para a análise de evidências, num ambiente controlado.

O primeiro Capítulo apresenta o propósito do trabalho, sendo seguido por uma apresentação do que é a Internet, o seu histórico até os dias atuais e seu funcionamento, no segundo Capítulo. O terceiro, por sua vez, tem como objetivo fundamentar toda a teoria envolvida em um *web browser* e seus componentes. E o Capítulo a seguir, o quarto, discorre sobre os aspectos mais importantes de segurança em tais programas. Já o quinto, apresenta o que são crimes digitais, e qual a legislação vigente no Brasil para a punição dos mesmos. E no sexto é realizada uma breve fundamentação sobre a importância da segurança da informação.

No sétimo Capítulo, são demonstradas as principais etapas de uma perícia forense computacional, e algumas das metodologias usadas na condução da mesma; direcionando-se o foco para as técnicas de perícia específicas para *web browsers*. Ao final deste Capítulo, apresentaram-se ainda algumas das ferramentas forense livres que podem ser usadas na execução de uma perícia.

Já no oitavo, são explicitados alguns trabalhos estudados, cujo o tema era similar, mas o foco da pesquisa outro. E por fim, o nono Capítulo, apresenta o trabalho desenvolvido, constituído de um estudo de caso simulando a condução de uma perícia.

Ao final é possível encontrar ainda, uma breve conclusão, discorrendo sobre os resultados obtidos e recomendando trabalhos futuros.

2 HISTÓRICO DA INTERNET

Segundo Tanenbaum (2003) a Internet não é de modo algum uma rede, mas sim um extenso conjunto de redes diferentes, que utilizam alguns protocolos comuns e oferecem determinados serviços comuns. É um sistema pouco usual, uma vez que não foi planejado nem é controlado por alguém.

Foi durante a II Guerra Mundial que a ciência da computação deu seu salto significativo. No período entre 1937 e 1944, foi desenvolvido o primeiro computador o MARK I cujo nome oficial era "Calculador Automático Seqüencial Controlado", idealizado pelo Prof. Howard Aiken da Universidade de Harvard, nos EUA, e que foi parcialmente financiado pela International Business Machines (IBM). O MARK I media 2,5 m de altura e 18 m de comprimento, e comparado aos computadores atuais não passava de uma calculadora eletromecânica enorme (WILLRICH, 2004).

Em 1940, nos EUA, o matemático naturalizado norte-americano Von Neumann, propõe um modelo de computador que viria a se tornar a base das estações de informática utilizadas até hoje (WILLRICH, 2004). Foi o modelo de Von Neumann que deu origem a primeira máquina computacional digital, o Eletronic Numeric Integrator and Calculator (ENIAC), criado entre 1943 e 1946. Máquina esta que ocupava uma área de 170 m^2 e pesava 30 toneladas, e permaneceu operacional por mais de 10 anos (WILLRICH, 2004).

Anos mais tarde, em 4 de outubro de 1957, a extinta União das Repúblicas Socialistas Soviéticas (URSS) lança o primeiro satélite artificial da terra, o Sputnik (ZAKON, 2004). Este acontecimento impulsionou o governo norte-americano a criar a agência Advanced Research Projects Agency (ARPA) que tinha a missão de assegurar a liderança mundial norte-americana no uso da ciência e tecnologia em aplicações militares. Foi a ARPA que em 1969, criou uma rede de computadores denominada Advanced Research Projects

Agency Network (ARPANET) originando anos mais tarde o que conhecemos hoje como Internet (UFPA, 2004). Com o passar dos anos a ARPANET deixou cada vez mais de ser uma aplicação apenas militar, sendo no início da década de 70 integradas a rede, universidades e outras instituições que faziam trabalhos que envolvessem a defesa dos EUA. As primeiras conexões internacionais da ARPANET foram realizadas com a University College of London, na Inglaterra, e a Nordic Network on Software Architecture Research (NOSAR), na Noruega (ZAKON, 2004).

Em 1973, dois cientistas que haviam trabalhado no projeto ARPANET, Vint Cerf e Bob Kahn, colaboraram entre si e estabeleceram protocolos muito bem estruturados para promover a troca de dados numa rede. Este artigo tratava então, do protocolo de transmissão de dados, mais conhecido por Transmission Control Protocol (TCP) (FOROUZAN, 2006).

Em 1974, o cientista Vint Cerf que trabalhava na altura no Microwave Communications, Inc (MCI) usou o termo Internet pela primeira vez ao descrever o protocolo TCP (UFPA, 2004). Pouco tempo depois, autoridades da área da computação decidiram dividir o TCP em dois protocolos: o TCP e o Internet Protocol (IP), ficando o protocolo de Internet conhecido como TCP/IP (Transmission Control Protocol/Internet Protocol) (FOROUZAN, 2006).

O número de redes inter-conectadas a ARPANET cresceu rapidamente depois que o TCP/IP tornou-se o protocolo oficial de transmissão de dados em 1º de Janeiro de 1983 (TANENBAUM, 2003).

Em 1990, o Departamento de Defesa dos EUA desmantelou a ARPANET substituindo-a pela rede da National Science Foundation (NSF), batizada como National Science Foundation Network (NSFNET), e que se popularizou, em todo o mundo, com a denominação Internet (UFPA, 2004). Para a sua popularização foi também crucial o

surgimento de tecnologias como: a World Wide Web (WWW), e o HTML, e de aplicativos como os *browsers*.

2.1 A WORLD WIDE WEB

Até meados da década de 1990, a Internet era usada exclusivamente por pesquisadores ligados às universidades, ao governo e à indústria. Uma nova aplicação, a WWW, mudou essa realidade e atraiu para a rede milhares de usuários, sem qualquer vínculo acadêmico. Ela surgiu quando Tim Berners Lee, do European Organization for Nuclear Research (CERN), em Genebra – Suíça, criou o protocolo HyperText Transport Protocol (HTTP) (ZAKON, 2004).

Em 1993 surge o primeiro navegador gráfico para Internet, o Mosaic, desenvolvido por Marc Andreessen no National Center for Supercomputer Applications (NCSA), nos EUA (TANENBAUM, 2003). Ele integrou recursos de multimídia às páginas de Internet, através da utilização do protocolo HTTP, disponibilizando a informação de uma maneira mais simples e intuitiva (UNICAMP, 1998).

Grande parte do crescimento da WWW durante a década de 1990 foi também impulsionada por empresas provedoras de serviços da Internet ou Internet Service Providers (ISPs), que ofereceram a possibilidade de usuários individuais se conectarem a Internet. A característica da rede foi assim alterada, passando de um serviço militar e acadêmico à um serviço de utilidade pública, muito semelhante ao sistema telefônico.

De acordo com o CETIC.br (2009) o uso da mesma vem crescendo exponencialmente no Brasil, estando o computador presente em 25% dos domicílios brasileiros e a Internet em 18% como ilustra a Figura 1.

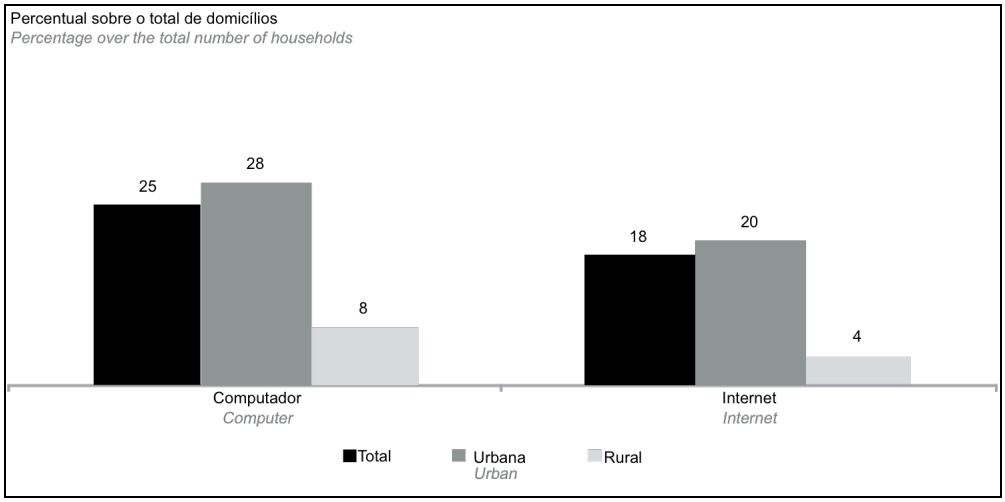


Figura 1. Proporção de domicílios com computador e Internet
Fonte: CETIC.br, 2009

2.2 PADRÕES DA INTERNET

De acordo com Forouzan (2006) padrões da Internet são especificações úteis, exaustivamente testadas, que regulamentam formalmente o que deve ser seguido para quem trabalha com a mesma.

Existe um procedimento rigoroso para que se possa definir uma especificação como padrão da Internet. Uma proposta de especificação primeiramente é definida como um Internet draft (minuta), ou seja, ela é definida como um trabalho em progresso, sem status oficial e com tempo de vigência de seis meses, para que as autoridades competentes julguem o documento (FOROUZAN, 2006).

Se as autoridades recomendarem o draft, este passará a ser um Request For Comments (RFC), isto é, ele será disponibilizado para a comunidade da Internet para análise, podendo o mesmo ser editado e atualizado quantas vezes se fizerem necessárias. O status de padrão da Internet é mais tarde atribuído num RFC intitulado, Internet Official Protocol Standards, que em tradução livre significa Protocolo Oficial de Normas da Internet (BRADNER, 1996).

2.3 COMPONENTES SEMÂNTICOS DA INTERNET

A Internet possui três componentes semânticos principais: Uniform Resource Locator (URL), Hypertext Markup Language (HTML) e Hypertext Transfer Protocol (HTTP). Sendo que uma URL é um mecanismo universal para identificar recursos na Internet, a HTML é uma linguagem padrão para a criação de documentos de hipertexto, e o HTTP é o protocolo de comunicação entre clientes e servidores na rede (KRISHNAMURTHY; REXFORD, 2001).

2.3.1 Uniform Resource Locator (URL)

O acesso e a manipulação de recursos distribuídos na *web* exige um modo para identificá-los. Um recurso na Internet é identificado por uma URL, ou seja, uma string formatada como *http://www.exemplo.com/index.html*.

Informalmente, uma URL normalmente consiste em três partes: o protocolo para a comunicação com o servidor (http), o nome do servidor (www.exemplo.com) e o nome do recurso nesse servidor (index.html) (KRISHNAMURTHY; REXFORD, 2001).

A sintaxe abstrata para uma URL é descrita na RFC 3986. O documento define a estrutura básica e hierárquica da mesma, uma lista de caracteres não reservados que podem aparecer nelas (0-9 a-z A-Z - . _ ~), outra lista de caracteres reservados que têm significados especiais e podem ser usados em alguns lugares apenas, para que executem a função pretendida (: / ? # [] @ ! $ & ' () * + , ; =), e estabelece uma codificação hexadecimal denotada por (%) para todos os caracteres que se encontrem fora das listas definidas acima (ZALEWSKI, 2009).

Alguns mecanismos adicionais são definidos na RFC 1738 como a sintaxe da URL no âmbito dos protocolos HTTP, File Transfer Protocol (FTP), Network News Transfer Protocol (NNTP), Gopher, bem como vários outros protocolos específicos. Juntas, essas RFCs definem a sintaxe para os recursos comuns da Internet (ZALEWSKI, 2009).

2.3.2 HyperText Markup Language (HTML)

A linguagem HTML oferece uma representação padrão para documentos de hypertexto no formato American Standard Code for Information Interchange (ASCII). Ela é derivada da linguagem Standard Generalized Markup Language (SGML), mais genérica (KRISHNAMURTHY; REXFORD, 2001).

O projeto inicial da linguagem, a definia com uma sintaxe muito limitada estritamente destinada a estruturar várias partes funcionais de um documento (ZALEWSKI, 2009). Com o rápido desenvolvimento dos *web browsers*, a tecnologia base da linguagem foi estendida rapidamente, mas com pouca supervisão, possibilitando a mesma fornecer funcionalidades adicionais relacionadas a apresentação visual como: inclusão de conteúdo multimídia diferente do padrão HTTP, encadeamento de diferentes documentos HTML dentro de quadros, submissão de complexas estruturas de dados e arquivos fornecidos pelo cliente, bem como a inclusão de *scripts* escritos em outras linguagens.

A falta de supervisão durante o seu crescimento, no entanto, levou a linguagem a tornar-se difícil de analisar, com peculiaridades únicas e limitações estranhas, e entrelaçando profundamente o estilo visual e a estrutura da informação do documento (KRISHNAMURTHY; REXFORD, 2001).

Desde 2004, a World Wide Web Consortium (W3C) e o Web Hypertext Application Technology Working Group (WHATWG), têm trabalhado focados em fazer da

HTML uma linguagem melhor estruturada, mais limpa, e rígida. Meta essa que só foi atingida com a versão 4 do HTML e com o surgimento da eXtensible Hypertext Markup Language (XHTML), que é uma variante do HTML em conformidade com as regras de sintaxe da linguagem eXtensible Markup Language (XML) (ZALEWSKI, 2009).

2.3.3 HyperText Transfer Protocol (HTTP)

Qualquer operação realizada na rede depende de se ter um modo padronizado e bem definido para os seus componentes se comunicarem. O HTTP é o modo mais comum de se transferir recursos na Internet (KRISHNAMURTHY; REXFORD, 2001).

Ele define o formato e o significado das mensagens trocadas entre os componentes da *web*, como clientes e servidores. E define também a sintaxe das mensagens e como os campos em cada linha da mensagem devem ser interpretados (ZALEWSKI, 2009).

Trata-se de um protocolo pedido-resposta, onde o cliente envia uma mensagem de pedido e o servidor responde. Sendo importante ressaltar que é um protocolo sem estado, onde clientes e servidores tratam cada troca de mensagens independentemente, e não precisam manter qualquer informação entre pedidos e respostas (KRISHNAMURTHY; REXFORD, 2001).

No seu início o HTTP era apenas um método de comunicação baseado em texto que surgiu como um projeto muito simples, elaborado por Tim Berners-Lee, e apelidado de HTTP/0.9. A versão 0.9 atualmente deixou de ser usada pelos navegadores *web*, mas ainda é reconhecida por alguns servidores. A versão a seguir foi a 1.1 descrita no RFC 2616, bastante complexa e mantendo alguma compatibilidade superficial com a idéia original (ZALEWSKI, 2009).

Segundo Bastos e Ladeiras (2001) o protocolo define oito métodos que indicam a ação a ser realizada pelo servidor com o recurso especificado na URL, fornecida no momento da requisição:

a) **GET –** solicita algum recurso, por exemplo um arquivo, por meio do protocolo HTTP. O cabeçalho Host reconhece diferentes Domain Name Servers (DNS) que tenham o mesmo IP;

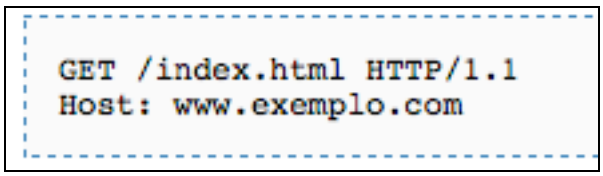


Figura 2. Requisição usando o método GET
Fonte: ZALEWSKI, M. (2009)

b) **HEAD –** implementa uma variação do método GET onde o recurso acessado não é retornado. É normalmente usado para se obter meta-dados por meio do cabeçalho de resposta sem ser necessária a recuperação de todo o conteúdo;

c) **POST –** envia dados ao servidor para serem processados geralmente por um programa ou *script* identificado na URL. Uma requisição por meio deste método sempre determina que as informações sejam submetidas inclusas no corpo da mensagem, e formatadas como uma *query string*. Ela deve também conter cabeçalhos especificando seu tamanho (*Content-Lenght*) e seu formato (*Content-Type*). Por esses motivos, o método POST é considerado mais seguro que o método GET, em relação aos dados transferidos, uma vez que este último anexa os dados a URL, ficando eles visíveis ao usuário (HERRMANN, 1997);

```
POST /index.html HTTP/1.0
Accept: text/html
If-modified-since: Sat, 29 Oct 1999 19:43:31 GMT
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Nome=NamePessoa&Idade=99&Curso=Computacao
```

Figura 3. Requisição usando o método POST
Fonte: ZALEWSKI, M. (2009)

d) **PUT –** envia determinado recurso ao servidor;

e) **DELETE –** exclui determinado recurso do servidor;

f) **TRACE –** permite ao cliente saber o que os servidores intermediários estão alterando em seu pedido;

g) **OPTIONS –** verifica os métodos HTTP que o servidor aceita;

h) **CONNECT –** usado quando se pretende utilizar um Proxy, que pode tornar-se num túnel Secure Sockets Layer (SSL). Túneis normalmente são usados para tornar a conexão mais segura.

No próximo capítulo é apresentado um breve histórico dos *web browsers*, e é explicitado o seu funcionamento básico.

3 HISTÓRICO DOS BROWSERS

Desde que Tim Berners-Lee apresentou a WWW como uma ferramenta de navegação na Internet aos seus colegas no CERN no início da década de 1990, o desenvolvimento dos *browsers* tem sido intrinsecamente ligado ao desenvolvimento da própria rede.

O Mosaic, que foi o primeiro *browser* gráfico a ser disponibilizado ao mercado, foi também um dos motivos da popularidade crescente da *web* na altura. Marc Andreessen seu criador, trabalhava então no NCSA, mas, frente à popularidade de seu *browser*, demitiu-se para formar a companhia que mais tarde seria conhecida como Netscape Communications Corporation (BRASIL ESCOLA, 1999).

Ao final do ano de 1994, Marc Andreensen criou um *browser* chamado Netscape Navigator a partir do Mosaic. O Netscape era capaz de operar corretamente e de manter a mesma aparência em sistemas operacionais diferentes, fato que o fez deter cerca de 90% do mercado (FONSECA, 2009).

A Microsoft, que até então não havia investido no mercado da Internet, disponibiliza em 1995, o *browser* IE como parte integrante do pacote Plus do Windows 95 (ABBATE, 2000). Tal advento marcou o começo da guerra dos *browsers*, ou seja, da luta pelo mercado dessas aplicações, entre a gigante companhia de software, e a companhia menor mas responsável pela popularização da Internet, a Netscape.

Em 1998, torna-se claro que o declínio do domínio de mercado do *browser* da Netscape era irreversível, e numa tentativa de voltar ao topo a mesma deu origem, ao projeto Mozilla (FONSECA, 2009). Tal projeto, consistia na disponibilização do seu *browser* Navigator, sob uma licença de código aberto, permitindo à qualquer usuário alterar o código fonte do mesmo.

A característica de possuir código aberto permitiu que fossem criadas várias alternativas de *browsers*, incluindo o Firebird, que deu origem em 2004 ao conhecido Firefox.

Rapidamente o Firefox ganhou o mercado, impulsionado pelas muitas inovações (navegação por abas, sistema de bloqueio de pop-ups, barra de navegação inteligente, entre outras.) que apresentou (FONSECA, 2009).

Desde então, o desenvolvimento dos *browsers* não mais parou, e muitos outros foram lançados no mercado. Além dos já citados, outros conhecidos e que lutam intensivamente por parcelas maiores de mercado são: o Safari, lançado pela Apple em 2003; o Ópera, lançado pela empresa norueguesa Telenor em 1996; e o Chrome lançado pela Google em 2008.

Segundo o último relatório da W3Counter (2009) ilustrado pela Figura 4, os dois *browsers* mais usados são respectivamente o Internet Explorer (IE) e o Firefox.

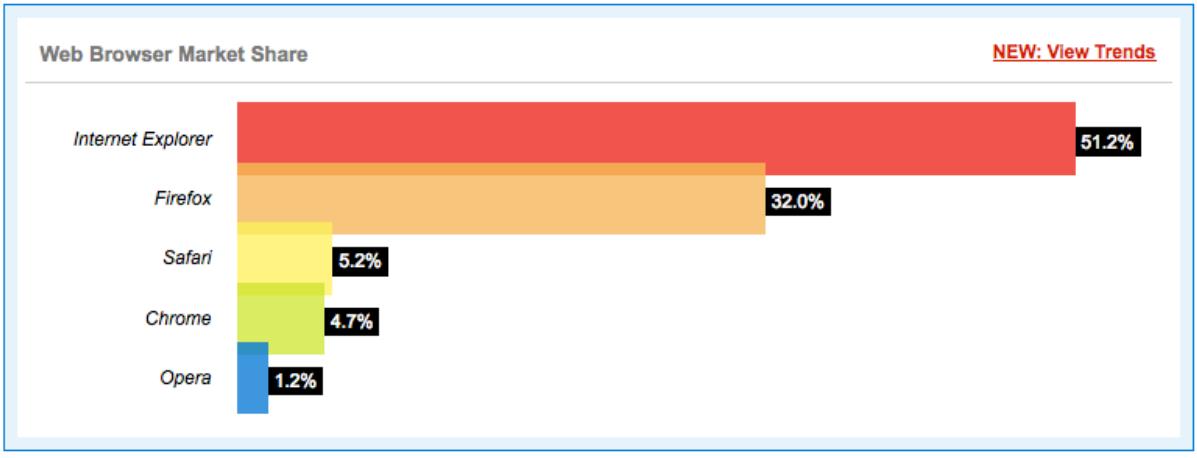


Figura 4. Percentagem de mercado dos Web Browser (Novembro 2009)
Fonte: W3Counter, 2009

3.1 FUNCIONAMENTO DOS BROWSERS

Segundo Krishnamurthy e Rexford (2001) o *browser* é percebido pelos usuários finais como a interface para a Internet, constituindo a forma mais conhecida de um cliente

para a mesma. A rede foi desenvolvida segundo uma arquitetura cliente-servidor tradicional, logo o *browser* é o cliente básico que se conecta a diversos servidores espalhados pelo mundo.

Num sistema cliente-servidor típico, os clientes usados são relativamente simples. Eles formatam um pedido, enviam-no ao servidor e lêem e apresentam a resposta. A ‘‘inteligência’’ normalmente é incorporada nos servidores.

Por sua vez no contexto de Internet, isso não acontece exatamente assim. Embora os servidores ainda possuam muita funcionalidade para oferecer, na prática, os clientes de Internet são também muito complexos.

A complexidade não vem das tarefas fundamentais que precisam realizar como: construir um pedido corretamente formatado, estabelecer uma conexão e comunicar-se com um servidor, mas sim de fatores ambientais ao redor da troca básica entre pedido e resposta (personalizar a criação do pedido de acordo com preferências do usuário, interpretar a resposta, e moldar a sua apresentação) (HERRMANN, 1997).

A Figura 5 ilustra as diferentes etapas envolvidas no processamento de um pedido pela Internet usando-se um *browser* típico.

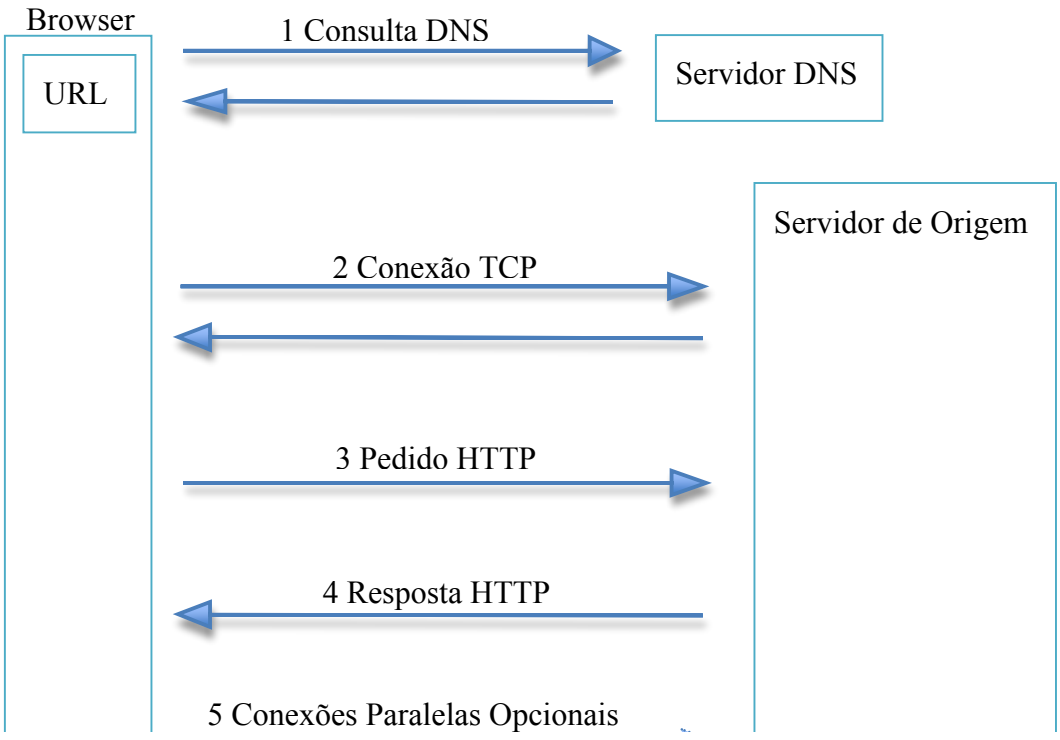


Figura 5. Etapas em um processo do browser
Fonte: adaptado KRISHNAMURTHY, B; REXFORD, J (2001)

Explicando de maneira bastante abstraída, pode-se dizer que a URL digitada, ou selecionada pelo usuário é analisada para se determinar o servidor *web* que será utilizado, uma conexão é estabelecida com o mesmo e um pedido HTTP é enviado com a URL para se obter uma resposta (KRISHNAMURTHY; REXFORD, 2001).

3.1.1 Emitindo um Pedido a partir de um Browser

A maioria dos usuários emite um pedido com um clique em um *link* de uma página de um site ou digitando a URL. No entanto, existem muitos casos em que a construção do pedido é um pouco mais complexa, como por exemplo no envio de formulários com informações sobre os usuários.

Nestes casos o usuário normalmente preenche um formulário e aciona um botão de envio, disparando uma ação, que resulta na construção de um pedido HTTP formatado corretamente.

O formulário completo pode ser enviado ao servidor de origem por pelo menos duas maneiras. Uma delas é tratando cada um dos valores preenchidos e os campos correspondentes como pares (nome, valor). A coleção de pares (nome, valor) é então codificada e enviada através do método GET do HTTP. A outra maneira seria incluir o formulário no corpo da mensagem HTTP a ser enviada usando o método POST (BASTOS; LADEIRAS, 2001).

3.1.2 Caching do Browser

Segundo Krishnamurthy e Rexford (2001) *cache* é um mecanismo dos *browsers* para reduzir a espera percebida pelos usuários na obtenção de uma resposta do servidor, usando o armazenamento local de mensagens. Existem dois tipos de *caching* que são comuns nos *browsers*: um tipo é o armazenamento na memória do processo em execução, e o segundo tipo é o armazenamento no disco rígido do computador utilizado.

A maioria dos usuários visita os mesmo sites frequentemente. Ao guardar em *cache* o conjunto mais recente de páginas visitadas, o *browser* reduz consideravelmente o tempo de resposta uma vez que ele não tem de fazer um novo pedido. Como um pedido HTTP novo, não é necessário, também não são necessárias a pesquisa do DNS para o servidor de origem, as conexões HTTP e TCP correspondentes e a transmissão dos bytes do recurso requisitado pelo usuário, culminando tudo em uma redução considerável no tempo de resposta percebida pelo usuário.

Pode ocorrer, entretanto, que o recurso em *cache* possa estar desatualizado quando comparado com a cópia original que o usuário tenta acessar. Nesse caso, o *browser* atualiza a copia que mantém salva, sendo esse processo conhecido como revalidação do *cache* (ZALEWSKI, 2009).

3.1.3 Tratando a Resposta do Servidor

Lidar com a resposta enviada pelo servidor pode ser considerada a última etapa das funções do *browser*. Ao aceitar a resposta ele analisa a mesma para verificar se existe algo a ser apresentado ao usuário ou não. Caso exista, o controle sobre o conteúdo e sua exibição é dividido da seguinte forma (KRISHNAMURTHY; REXFORD, 2001):

a) o usuário: Em termos do evento por ele gerado durante a navegação numa página (por exemplo, o evento gerado com o clique de um botão);

b) o servidor: Ao formatar a resposta para atender ao evento gerado pelo usuário;

c) as preferências do usuário: Com relação ao formato do conteúdo que ele pretende visualizar, expressas pelos cabeçalhos de pedido gerados, e enviados pelo *browser* ao servidor;

d) a resposta enviada pelo servidor: Com possíveis adaptações para atender as preferências do usuário com relação ao formato do conteúdo que irá visualizar;

e) o desejo do criador do conteúdo: De exibi-lo de determinada forma.

O resultado final a ser visualizado pelo usuário, depende assim, de diferentes componentes de software, da facilidade de configuração e do nível de controle em cada estágio citado acima.

4 SEGURANÇA EM WEB BROWSERS

Ao pensar sobre vulnerabilidades em um computador, podemos dizer que os *browsers* são uma das partes mais sensíveis de software pois eles são a porta pela qual os usuários interagem com a Internet. Eles são usados para interpretar conteúdo desenvolvido fora do controle do usuário, por profissionais e amadores que podem ter intenções benignas ou maliciosas.

Trata-se de um programa muito próximo do usuário manipulando informações sobre o mesmo sempre que necessário, tentando oferecer o máximo de flexibilidade a ele no controle de sua privacidade, ao mesmo tempo em que tenta garantir que programas remotos não terão acesso descontrolado ao seu computador (KRISHNAMURTHY; REXFORD, 2001).

Pesquisas sobre questões de segurança mostram que geralmente é mais difícil garantir a segurança quando uma parte complexa de software é usada. Visto que os *browsers* são primariamente usados para acessar um grande número de sites na Internet, com cada site rodando outro software complexo (o servidor), os riscos de existirem falhas de segurança é elevado (ZALEWSKI, 2009).

Os aspectos de segurança em um *browser* surgem principalmente no contexto do acesso ao disco e recursos de computação de uma máquina. Embora a maioria dos documentos que o usuário salva em seu computador usando o mesmo, sejam estáticos constituindo em texto e imagens, são muitas vezes salvos outros documentos executáveis que podem constituir um risco a segurança do usuário, principalmente quando este não possui uma idéia clara das reais capacidades do software.

As violações de segurança variam desde o acesso impróprio a arquivos, até problemas mais sérios, como o controle dos recursos de computação da máquina do usuário (KRISHNAMURTHY; REXFORD, 2001).

Para ilustrar que a qualidade da segurança em *web browsers* não é uma preocupação apenas dos usuários, mas da industria como um todo, temos a Figura 6 que mostra a quantidade de falhas de segurança corrigidas nos dois principais *browsers* usados no mercado, o IE e o Firefox, desde os seus lançamentos até novembro de 2007.

Podemos verificar, segundo a figura, que no período mencionado o Firefox teve 199 falhas corrigidas, onde 75 falhas foram graves, 100 falhas foram consideradas médias, e 24 falhas foram de baixa relevância. Já o IE teve 87 falhas de segurança corrigidas, onde 54 foram consideradas graves, 28 médias e 5 de baixa relevância (JONES, 2007).

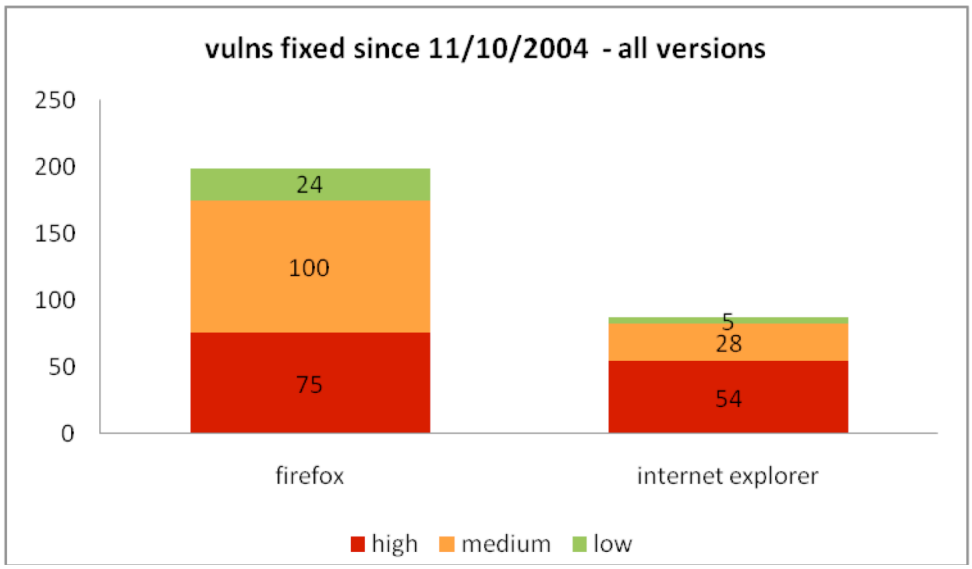


Figura 6. Quantidade de vulnerabilidades corrigidas em todas as versões suportadas dos browsers Firefox e Internet Explorer até Novembro de 2007
Fonte: JONES, R. (2007)

4.1 PROTOCOLO SSL

O protocolo SSL provê uma abstração de um canal de comunicação seguro, ou o que também é conhecido como um *stream* seguro, onde os princípios de integridade, privacidade, e autenticação de servidor, são oferecidos à interface HTML de maneira transparente (POUW; GEUS, 1999).

Para se criar um canal seguro com um servidor qualquer, deve-se usar o protocolo HyperText Transfer Protocol Secure (HTTPS) ao invés do HTTP normalmente usado. Por exemplo, *https://www.exemplo.net*.

A autenticação de clientes baseada em certificados SSL, no entanto, requer a geração dos mesmos de maneira específica para cada *browser* disponível no mercado. Sendo que, de maneira geral, esse tipo de mecanismo de segurança é muito pouco difundido, mesmo para aplicações críticas, como por exemplo as que envolvam a transferência de valores na Internet. A falta de padronização no uso e geração de certificados, além da complexidade de implementação em comparação a utilização de senhas convencionais, tem inibido o uso mais extensivo desta técnica de autenticação (JONES, 2007).

Além disso, a principal deficiência da arquitetura de segurança HTTPS está na fraca associação, por parte dos *browsers* comerciais, entre uma conexão segura e um servidor seguro propriamente dito. Tal vulnerabilidade é denominada *link spoofing*, e acontece quando o usuário tenta estabelecer um conexão segura (HTTPS) com um servidor, partindo de uma conexão não segura (HTTP) e por esse motivo vulnerável (POUW; GEUS, 1999). A referência ao *link* seguro pode ser, por exemplo, alterada quando em trânsito para o servidor, redirecionando o usuário para uma outra conexão (possivelmente também uma referência HTTPS), provavelmente em outro servidor e cuja página a ser visualizada teria o mesmo

aspecto da página original, mas com o único objetivo de enganar o usuário, forçando-o a disponibilizar informações suas importantes.

4.2 COOKIES

Como já foi descrito anteriormente, o HTTP é um protocolo sem estado, ou seja, ele não retém qualquer informação sobre pedidos passados ou futuros. No entanto, com o desenvolvimento de sites cada vez mais complexos surgiu a necessidade de se salvar informações sobre o usuário, como por exemplo, as suas preferências de navegação.

Com o intuito de suprir essa necessidade foram desenvolvidos os *cookies*, que são pequenos arquivos que salvam o estado da sessão do usuário e são enviados aos *browsers* e armazenados na máquina do usuário (KRISHNAMURTHY; REXFORD, 2001).

Os *cookies* foram introduzidos inicialmente pela Netscape em 1994, tendo as tentativas de padronização sido iniciadas pouco depois pelo Internet Engineering Task Force (IETF). O mecanismo de gerenciamento de estado formalizando o uso dos mesmos, é detalhado no RFC 2965, emitido em Outubro de 2000 como um padrão proposto pela IETF (ZALEWSKI, 2009).

Na Figura 7, podemos visualizar como ocorre a troca desse tipo de arquivos entre um browser e um servidor.

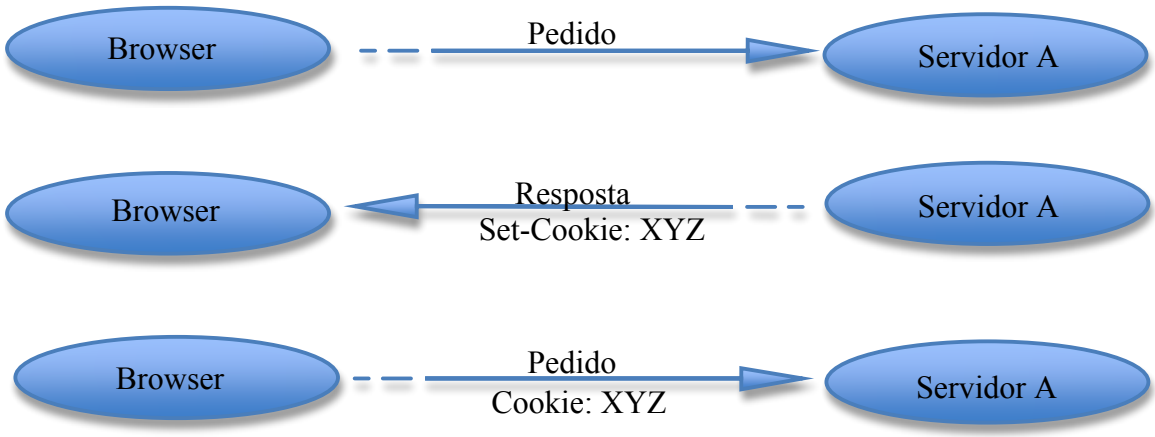


Figura 7. Troca de informações de cookie entre bowser e servidor.
Fonte: adaptado KRISHNAMURTHY, B; REXFORD J (2001)

A figura mostra um *browser* enviando um pedido a um servidor A qualquer. O servidor, em sua resposta, inclui um cabeçalho (*Set-Cookie*) com o valor (XYZ), onde XYZ representam as informações do usuário que o servidor pretende sempre obter, ou seja, o que será salvo pelo *browser*. Por fim, em todos os pedidos futuros do *browser* para o servidor A, será incluso o valor do *cookie* (XYZ) a ser enviado por meio do cabeçalho (*Cookie*).

Existem dois tipos de *cookies* (GALVÃO, 2008):

a) **Cookies de sessão –** são aqueles armazenados na memória;
b) **Cookies de persistência –** são armazenados em disco, em um pequeno arquivo texto contendo nomes e valores, a data em que o mesmo foi salvo, a data de expiração e informações de estado.

De acordo com as regras de implementação padronizadas, um *cookie* pode ter um tamanho de até 4KB, sendo que os *browsers* normalmente oferecem um armazenamento máximo de 200 por servidor e um total geral de 300 arquivos (KRISHNAMURTHY; REXFORD, 2001).

4.2.1 Controle dos Usuários sobre os Cookies

Os *cookies* normalmente são trocados sem o conhecimento do usuário, a menos que, este tenha solicitado que seja avisado toda vez que os mesmos forem enviados. Para os poucos usuários que conhecem essa possibilidade é disponibilizado um grau considerável de controle sobre esse tipo de arquivo.

Conforme Krishnamurthy e Rexford (2001) os usuários podem:

a) decidir se aceitam ou não, o uso de determinado *cookie*: Tal decisão pode impossibilitar o acesso a páginas de alguns sites;

b) definir um limite para o tamanho e o número de arquivos que aceitam: Isso controla a quantidade de espaço que eles têm para alocar em suas máquinas;

c) decidir se aceitam *cookies* de todos os sites ou apenas de sites específicos: Esse controle permite que os usuários aceitem arquivos apenas de sites confiados;

d) definir a aceitação desse tipo de arquivo, para a duração de uma sessão específica: Um grau de controle mais avançado em nível de sessão permite que os usuários ativem a aceitação de *cookies* para a realização de uma tarefa em particular. Ao final da sessão, o *browser* retorna ao modo padrão de não aceitá-los para as sessões futuras;

e) aceitar apenas aqueles que venham do mesmo servidor da página sendo atualmente vista pelo usuário: Esse controle garante que o usuário saiba de onde vêm os arquivos impedindo que outros sites (que podem ser acessados automaticamente pelo *browser*) salvem os seus *cookies* na máquina do usuário.

4.2.2 Problemas de Privacidade com os Cookies

Poucos assuntos causam tanta consternação quando se fala de tecnologias relacionadas à Internet quanto os *cookies*. Eles são frequentemente usados e/ou suspeitos de auxiliar na violação da privacidade dos usuários (KRISHNAMURTHY; REXFORD, 2001).

O problema começa com a falta de conhecimento dos usuários, que tais arquivos estão sendo enviados, com informações suas, para servidores remotos. A aceitação dos *cookies* é permitida como padrão na maioria dos *browsers*, sendo que raramente o usuário é notificado sobre o envio ou recebimento deles.

Eles são utilizados para basicamente três tipos de tarefa: gerenciar sessões de uso de um site, isto é, as visitas que o usuário faz ao site; personalizar a navegação do usuário; e

monitorar os caminhos e hábitos de navegação do usuário. É visível, que os de gerenciamento de sessão e os de monitoração podem constituir e/ou facilitar violações à segurança do usuário quando este navega em locais públicos, ou usa computadores compartilhados com outras pessoas, podendo assim, ter algumas informações privadas suas e armazenadas pelos *cookies*, acessadas.

Uma vez que esse tipo de arquivo é transmitido sem qualquer criptografia aplicada, fica fácil para um programa capaz de interceptar o tráfego da rede, descobrir o conteúdo dos mesmos, ou ainda, alterar o seu conteúdo enquanto eles trafegam pela rede, o que constituí uma ameaça direta a segurança do usuário (ZALEWSKI, 2009).

A alteração do valor de um *cookie* pode afetar seriamente o servidor que receberá o mesmo, pois dependendo da alteração, podem ocorrer mudanças graves no servidor. Por exemplo, se o servidor usava o arquivo como índice para acesso à um banco de dados, a modificação do mesmo pode resultar em alterações inadvertidas em partes do banco de dados (KRISHNAMURTHY; REXFORD, 2001).

Outro exemplo de como a privacidade do usuário pode ser colocada em risco, é o compartilhamento de suas informações conseguidas a partir desses arquivos, entre empresas, podendo estas gerar perfis de usuários e vendê-los a outros (ZALEWSKI, 2009).

O usuário não consegue controlar como as informações coletadas dos *cookies* serão utilizadas. Uma forma de ajudar o usuário a ter maior controle, é a criação mecanismos, para que o mesmo aceite que informações a seu respeito sejam recolhidas e usadas. Esse modelo de funcionamento é conhecido como *opt-in*. O esquema *opt-out*, alternativo, é aquele onde se exige que os provedores de conteúdo, ofereçam meios para que os usuários se excluam do compartilhamento de informações, caso assim o desejem.

No próximo capítulo, aborda-se um pouco sobre crimes digitais, e legislação vigente no Brasil sobre os mesmos.

5 CRIMES DIGITAIS

A palavra crime deriva do latim *crimen* que significa acusação. Segundo Aguiar (2009) para que um crime seja cometido é necessária a existência de uma conduta humana positiva (ação) ou negativa (omissão), que seja típica e descrita na lei como infração penal, sendo que somente o haverá se o fato for antijurídico. Por sua vez, de acordo com Jesus (2000), um delito tem como requisitos:

a) ser um fato típico, ou seja, estar incluso em uma norma penal incriminadora;
b) definir uma ação antijurídica, ou seja, uma ação contrária ao anseio da sociedade, ou as regras impostas por um grupo de pessoas;
c) ser uma ação culpável, que nada mais é que uma ação praticada por um indivíduo, seja ela com dolo ou não, em querer o resultado obtido.

Dentro desse contexto crimes digitais ou crimes de informática são definidos como condutas descritas em tipos penais, realizadas através de computadores ou voltadas contra computadores, sistemas de informática ou os dados e as informações neles encontrados (PAIVA, 2006).

Muitas vezes eles assemelham-se muito a crimes comuns, sendo a única diferença entre ambos, o fato de o digital utilizar um computador ou sistema informatizado para atingir o seu objetivo. Por este motivo, determinados autores como Pinheiro (2001), os classificam em três subgrupos:

a) **Crimes Digitais Puros –** visam lesar somente outros computadores ou sistemas de informática;
b) **Crimes Digitais Mistos –** não visam o sistema de informática em si, mas cujo o uso do mesmo é imprescindível para cometer o delito;

c) **Crimes Digitais Comuns –** utilizam um sistema de informática para cometer delitos comuns já homologados pela lei penal.

5.1 LEGISLAÇÕES PERTINENTES SOBRE CRIMES DIGITAIS

A idéia de crimes digitais ou crimes cometidos com o uso de computadores é relativamente nova, não existindo por esse motivo, no Brasil, leis específicas para esses atos. O que se tem hoje, capaz de condenar os praticantes de tais delitos, são alguns artigos do código civil, como o Art. 927, o Art. 186, e o Art. 187, (AGUIAR, 2009). Vide anexo A, para visualizá-los na íntegra.

Ainda segundo Aguiar (2009) a punição para esse tipo de delito, é atualmente determinada através de uma adequação do mesmo para leis vigentes, não específicas para crimes de informática, levando-se em consideração a consequência deles sobre as vitimas. Em alguns casos, tais adequações acabam gerando falhas nas tipificações de crimes cometidos usando-se o computador.

Por exemplo, de acordo com Pinheiro (2008) ao se usar a tipificação de crime de furto num ambiente digital, pode-se, em certos casos, invalidar o crime, visto que um agente criminoso que apenas invade um servidor e copia dados dele, não poderá ser condenado por furto, já que a tipificação do mesmo significa subtrair coisa alheia. É importante notar que neste caso, o fato de copiar o dado não o subtraiu, o que faz com que este tipo de delito passe a ser desqualificado para tal tipificação.

Por essa razão, estão em fase de elaboração e votação, projetos de lei que buscam punir casos de crimes digitais, com o objetivo de diminuir o elevado número de ocorrências destes.

Conforme Aguiar (2009) o Projeto de Lei (PL) nº 84/99 na câmara dos Deputados é um dos projetos mais importantes que estão em tramite no Congresso Nacional e cujo objetivo é a regulamentação dos delitos digitais.

O PL nº 84/99, vide anexo B para visualizá-lo na íntegra, prevê sete modalidades de crimes digitais, podendo a punição chegar até seis anos de reclusão e multa. O principal objetivo do projeto é o preenchimento das lacunas na legislação brasileira, isto é, retratar atos que não existem na legislação penal em vigor.

5.2 TIPOS MAIS COMUNS DE CRIMES DIGITAIS

A Figura 8 permite visualizar as percentagens de incidentes ocorridos durante o ano de 2009 e reportados ao CERT.br, separados por tipos de ataques.

Figura 8. Incidentes Reportados ao CERT.br -- Janeiro a Dezembro de 2009
Fonte: CERT.br (2010)

De acordo com o CERT.br (2010a), e como pode-se constatar na figura, os tipos mais comuns de crimes digitais são:

a) **Fraude –** definida como qualquer ação ardilosa, enganosa, de má-fé, com intuito de lesar ou ludibriar alguém, ou de não cumprir determinado dever (Houaiss, 2009). Nesta categoria encontram-se incidentes, onde alguém tentou obter uma vantagem sobre outrem, utilizando-se de meios digitais;

b) **Scan –** procuras realizadas em redes de computadores, com o intuito de identificar quais computadores estão em funcionamento e que serviços estes utilizam. Criminosos utilizam-se desta técnica para identificar potenciais alvos, uma vez que ela permite reconhecer vulnerabilidades nos sistemas informáticos tendo como base os serviços que estes utilizam;

c) **Worm –** definido como um programa malicioso, automatizado para se auto-propagar numa rede de computadores;

d) **Web –** ataques específicos cujo objetivo é o comprometimento de servidores web, ou a desfiguração de páginas na Internet;

e) **Denial of Service (DOS) –** modalidade de crime na Internet, onde o criminoso efetua uma grande quantidade de solicitações a servidores *web*, computadores, redes, ou outros sistemas informatizados com a finalidade de torná-los indisponíveis. O objetivo do autor deste ataque é o dano ao sistema atacado, sendo que os alvos normalmente são empresas de grande porte (AGUIAR, 2009);

f) **Invasão –** encontram-se nesta categoria ataques bem sucedidos que culminaram no acesso não autorizado a um computador ou a uma rede;

g) **Outros –** notificações recebidas pelo CERT.br que não se enquadraram em nenhuma das outras categorias.

Abaixo é apresentada uma breve fundamentação sobre a importância da segurança da informação, tanto para usuários comuns como para corporações empresariais.

6 IMPORTÂNCIA DA SEGURANÇA DA INFORMAÇÃO

Já foi verificado no decorrer do trabalho que, inúmeros são os crimes digitais que podem ser cometidos contra os usuários comuns de um computador, e que colocam as suas informações em perigo. Tais crimes não apenas violam a privacidade do usuário, como também acarretam consequências mais graves como roubos de identidade e fraudes.

Nas corporações e repartições de trabalho, a situação não é diferente. A segurança da informação neste contexto é de extrema importância, pois as informações são ativos importantes para os negócios. A confidencialidade, a integridade e a disponibilidade das informações atualmente são fatores importantes para se manter a competitividade, o fluxo de caixa, o atendimento à legislação e a imagem comercial das mesmas.

Assim como acontece com os usuários comuns, cada vez mais, as organizações, suas redes e sistemas de informação, enfrentam ameaças de segurança vindas das mais diversas fontes. Devido à dependência que as mesmas apresentam, de seus sistemas e serviços de informação, elas estão cada vez mais vulneráveis às ameaças contra a segurança.

A interligação de redes públicas e privadas bem como o compartilhamento de recursos de informação, são outros fatores que aumentam a dificuldade de se conseguir controlar o acesso as informações empresariais (ISO/IEC, 2004).

Independentemente de qual seja a forma que as informações assumam, ou os meios pelos quais sejam compartilhadas ou armazenadas, elas devem ser sempre protegidas adequadamente.

Segundo a ISO/EIC (2004) a segurança de informações é caracterizada como a preservação de:

a) **Confidencialidade –** garantindo que as informações estejam acessíveis somente aqueles autorizados;

b) **Integridade –** salvaguardando a exatidão e confiabilidade das informações e métodos de processamento;

c) **Disponibilidade –** assegurando que aqueles que tenham permissão para tal tenham acesso às informações e ativos associados quando necessário.

Ainda de acordo com a ISO/EIC (2004), a segurança da informação é adquirida através da implementação de um conjunto apropriado de controles, que podem ser políticas, práticas, procedimentos, estruturas organizacionais ou funções de software a serem adotadas pela organização ou empresa.

7 PERÍCIA FORENSE COMPUTACIONAL

A aplicação de estudos científicos a lei é chamada de ciência forense (CASEY, 2004). Podemos então, definir a mesma como a utilização da ciência ou da tecnologia em processos investigativos, estabelecendo fatos ou evidências num tribunal de justiça (CARRIER, 2005).

Por sua vez, a utilização dela em sistemas computacionais durante investigações oficiais por peritos judiciais, é comumente chamada de perícia forense computacional (BERNARDO, 2006).

Segundo Vacca (2002) a perícia forense computacional é definida como a coleta, preservação, análise e apresentação de evidências digitais, utilizando-se de ferramentas e técnicas computacionais no ambiente investigado, e auxiliando os juízes nas tomadas de decisões num processo judicial.

Para que as evidências sejam validadas judicialmente, é extremamente importante que os dados, as informações e as provas periciais sejam coletadas, analisadas e apresentadas somente por um perito no assunto, evitando que durante o processo não ocorra a perda ou contaminação das mesmas.

7.1 EVIDÊNCIAS DIGITAIS

Apesar dos muitos esforços empreendidos por criminosos para destruírem provas incriminadoras, normalmente sempre são deixadas para trás pistas, que podem ser usadas para traçar suas atividades e incriminá-los. Tais pistas, podem ser usadas como prova ou álibi em crimes, e são conhecidas como evidências (VACCA, 2002).

Conforme Casey (2004) a evidência digital é então definida como qualquer dado armazenado ou transmitido fazendo-se uso do computador que pode provar ou negar a teoria de como ocorreu um crime, intenção ou álibi, estabelecendo uma conexão entre um crime e a vítima ou um crime e um criminoso.

Como qualquer outro tipo de evidência, a digital deve ser: autêntica, exata, completa, convincente ao corpo de jurados, e estar em conformidade com as leis e legislação (VACCA, 2002). Para que tal aconteça, a manipulação da mesma deve ser feita por um perito forense computacional treinado, correndo-se o risco de ela ser rejeitada pela justiça caso contrário.

De acordo com Casey (2004) existem algumas particularidades quanto à manipulação deste tipo de provas, que beneficiam os peritos, como:

a) elas podem ser duplicadas, sendo que a cópia pode ser examinada como se fosse a original;
b) as cópias podem sempre ser comparadas com a original, verificando-se se ocorreram alterações durante a manipulação;
c) tais evidências são difíceis de serem realmente destruídas. Até mesmo quando o arquivo é excluído ou o disco rígido formatado, a evidência digital pode, em certas ocasiões, ser recuperada;
d) ao destruir-se uma evidência digital, cópias de outras associadas a ela podem residir em outros locais.

7.2 PROVAS PERICIAIS

Segundo Jesus (2000) uma prova é uma forma direta de se conhecer a verdade sobre um fato, através da aquisição e apresentação de documentos, declarações e outros elementos que foram analisados.

As provas periciais podem incriminar ou não um indivíduo, portanto, um parecer científico sobre um caso, deve ser emitido somente por uma pessoa de conhecimento técnico especializado, que fornecerá declarações científicas sobre fatos relevantes, relacionados ao crime a ser julgado.

7.3 PERITOS FORENSES E SUAS ESPECIFICAÇÕES

Conforme Casey (2004) a perícia forense computacional pode ser dividida em áreas chave, como: aquisição, processamento e análise de evidências. Exigindo cada área, especializações, habilidades e procedimentos diferentes no tratamento das evidências.

Como exemplo, podemos observar que, para o profissional que coleta os dados de um disco rígido, é necessário dominar a forma com que os mesmos são gravados nele e de que forma não contaminá-los durante a aquisição de evidências, mas para o profissional que irá analisar os dados coletados, tal conhecimento torna-se irrelevante, sendo importante para ele conhecer o funcionamento do sistema de arquivos do SO que o computador executa, o que lhe permitirá tentar recriar as atividades do criminoso.

Ainda segundo Casey (2004) os profissionais da área podem ser divididos de acordo com suas habilidades e especializações em:

a) **Técnico da cena do crime –** pessoa responsável pela aquisição dos dados;

b) **Examinador de evidências –** responsável pela examinação da qualidade e relevância das evidências digitais;

c) **Investigador digital –** responsável pela investigação completa. Também são responsáveis pela reconstrução das ações do criminoso, usando as informações adquiridas pelas primeiras respostas e examinadores.

7.4 METODOLOGIAS INVESTIGATIVAS

Os procedimentos técnicos a serem realizados pelo perito forense podem ser diferentes de acordo com os sistemas e aparatos tecnológicos envolvidos. A falta de métodos específicos que não se alterassem de acordo com a tecnologia usada, enfraquecia a credibilidade de provas periciais apresentadas em casos judiciais.

Na tentativa de dar-se maior credibilidade e solidez à perícia forense computacional em frente à jurados, criaram-se metodologias que são usadas como guias do processo investigativo, definindo etapas a serem seguidas pelos peritos, independentemente de qual o sistema computacional, ou de quais ferramentas foram usadas (BERNARDO, 2006).

7.4.1 Metodologia DFRWS

Elaborada por Gary Palmer no primeiro Digital Forensics Research WorkShop (DFRWS), propõe sete etapas (BERTOGLIO, 2008):

a) identificação: Esta etapa compreende o método através do qual o perito é notificado do incidente;

b) preservação: A integridade e estado das evidências devem ser asseguradas nesta etapa;

c) coleta: Nesta fase é realizada a extração ou coleta de itens individuais ou em grupo, usando-se de métodos específicos e ferramentas para aquisição de evidências;

d) exame: É realizada uma análise cuidadosa dos itens e suas características e atributos. Nesta etapa o foco está na extração de informações das evidências encontradas, sem o intuito de formar conclusões sobre o caso;

e) análise: Analisam-se todas as evidências encontradas desde o início da investigação, com a finalidade de desenvolver um conjunto de conclusões em relação as provas apresentadas;

f) apresentação: O perito deve nesta fase relatar os fatos de maneira organizada, clara, concisa e objetiva, e;

g) decisão: Contempla a etapa anterior a apresentação de laudos periciais para o tribunal, onde o perito determina as suas conclusões sobre o caso.

7.4.2 Metodologia de Reith, Carr and Gunsch

A metodologia proposta por Reith, Carr e Gunsch (2002), também conhecida como *Abstract Digital Forensics Model*, possui algumas similaridades com a metodologia DFRWS. Ambas apresentam as etapas de preservação, coleta, exame e apresentação, sendo a particularidade presente nesta metodologia, o fato de ela proporcionar suporte à preparação de ferramentas e à uma dinâmica formulação de abordagens investigativas.

A estrutura da metodologia é baseada em nove etapas, citadas abaixo (BARYAMUREEBA; TUSHABE, 2004):

a) identificação: Reconhecimento do incidente;
b) preparação: Preparação das ferramentas, técnicas, monitoração de autorização, mandatos de busca e suporte;
c) estratégia e abordagem: Desenvolvimento de uma estratégia de coleta de evidências que maximize a coleta de evidências não infectadas, e minimize o impacto para a vítima;
d) preservação: Proteção e conservação do estado físico e digital das evidências;
e) coleta: Gravação da cena do crime e reprodução das evidências digitais usando procedimentos aceitos e padronizados;
f) exame: Busca aprofundada e sistemática das provas relativas à suspeita do crime;
g) análise: Reconstrução dos fragmentos de dados e elaboração de conclusões baseadas nas provas encontradas;
h) apresentação: Explicação das conclusões;
i) devolução das evidências: Garante que a propriedade física e digital seja devolvida ao proprietário.

7.4.3 Metodologia SOP

Criada pelo Scientific Working Group on Digital Evidence (SWGDE), que é o representante norte-americano na International Organization on Computer Evidence (IOCE).

A metodologia Standard Operating Procedures (SOP), foi apresentada durante a International Hi-Tech Crime and Forensics Conference (IHCFC), realizada em Londres, de 4 a 7 de outubro de 1999 (BERTOGLIO, 2008). Ela incorpora os conceitos e procedimentos da ciência forense, incluindo: comparação, classificação, individualização e avaliação da fonte

de evidências, sendo constituída por 6 etapas melhor explicitadas abaixo (BERNARDO, 2006):

a) autorização e preparação: Antes de qualquer investigação, o perito forense deve certificar-se de que não está infringindo nenhuma lei. Caso contrário sua perícia será afetada ou simplesmente invalidada. Torna-se então obrigatório, que o perito receba a autorização de um juiz, para efetuar a mesma, ficando o exame restrito somente ao que for determinado pelo órgão judicial;

b) identificação: Nesta etapa deve-se fazer o levantamento das informações relevantes ao crime e a identificação de todo o hardware e software do computador a ser examinado;

c) coleta e preservação: Após a identificação das fontes de evidências, estas devem ser coletadas e mais tarde autenticadas. É primordial que as evidências não sejam alteradas durante todo o processo investigativo. É aconselhável que o perito calcule o valor *hash* dos arquivos originais, antes de copiar as evidências. Segundo Place (2008) um *hash* é uma seqüencia de letras ou números gerados por um algoritmo de dispersão, buscando identificar um arquivo ou informação unicamente. Trata-se de um método para transformar dados de tal forma que o resultado seja o mais exclusivo possível. Assim sendo, uma função *hash* recebe determinado valor e retorna um código que funciona como um identificador único. Ao fazer-se a cópia de um arquivo, se a mesma for fiel ao arquivo original, ambas apresentarão o mesmo valor *hash*. O perito garante assim a integridade dos dados e a credibilidade da sua perícia;

d) exame e análise: Fase posterior à coleta das evidências, onde estas serão analisadas pelo perito na busca de provas;

e) documentação: A documentação é essencial em todas as etapas da perícia forense. Primeiro porque, caso seja necessário outro perito dar sequência ao processo, o seu trabalho estará facilitado, e também visto que, a perícia terá maior credibilidade se a documentação estiver completa, com dados como: quem coletou e tratou as evidências com data e hora, e os valores *hash* de todas as evidências copiadas para demonstrar que as cópias estão livres de alteração e são autênticas;

f) Reconstrução da Cena do Crime: Esta etapa tenta responder as seguintes perguntas: o que aconteceu? Quem executou? Quando aconteceu? Onde aconteceu? Como aconteceu? E Por quê?

Segundo o SWGDE (2008) SOPs são documentos únicos, específicos para determinado propósito, que descrevem os métodos e procedimentos a serem seguidos na realização de operações de rotina. Elas devem ser revistas anualmente, sendo as versões previamente aprovadas, guardadas para referência.

Os padrões desenvolvidos pelo SWGDE seguem o princípio, de que todas as organizações que trabalham com a investigação forense devem conservar um alto nível de qualidade, a fim de assegurar a confiança e a exatidão das evidências, e por se tratar de uma organização mundial, amplamente reconhecida, será esta a metodologia usada na aplicação prática do presente trabalho.

Para terminar, ainda que o perito siga rigidamente metodologias internacionais de perícia como as mostradas acima, ele deve sempre levar em consideração as leis e regras que regem o ambiente onde a perícia será executada. Por exemplo, se a perícia for executada numa empresa, ela deve estar de acordo com as regras internas da empresa, leis municipais, estaduais e federais para que a mesma não seja invalidada (BERNARDO, 2006).

7.5 PERÍCIA FORENSE EM WEB BROWSERS

Ao falar-se de perícia forense em *web browsers* devemos levar em consideração dois lados: o lado do cliente (o *browser* em si), e o lado do servidor (servidores *web*, servidores de aplicação, servidores de banco de dados). Deve-se ainda, levar em consideração outros fatores relevantes como o tráfego de rede e as características dos sistemas operacionais utilizados (CARUSO, 1999).

Tal perícia é normalmente usada em casos de: pornografia infantil, pedofilia, fraudes eletrônicas, roubo de identidade, espionagem industrial, e incidentes de segurança convencionais (vírus, *worms*, *phishing*, *hacking*, entre outros); objetivando sempre identificar se o usuário do equipamento periciado é vítima ou se está envolvido no incidente (JONES, 2003).

Como já foi abordado anteriormente, apesar dos dois *browsers* mais usados serem respectivamente o IE e o Firefox, existem muitas outras opções disponíveis no mercado. Tal fato pode dificultar a tarefa do perito forense, uma vez que, em alguns casos, ele pode não dispor de uma ferramenta adequada aquele a ser analisado.

Devido a imensidão de técnicas e ferramentas existentes, a presente pesquisa será restrita aquelas usadas para coletar e analisar evidências digitais no lado do cliente, ou seja, nos *browsers* em si, respectivamente no IE e no Firefox, nas suas versões 8.0.6 e 3.6.12 para o SO Windows XP. Um apêndice abordando brevemente algumas técnicas, que possibilitem a realização de uma perícia em um *browser* Firefox rodando em sistemas derivados do UNIX, pode ser encontrado ao final.

É importante mencionar que, assumiu-se durante o resto do trabalho que os diretórios relevantes a uma perícia forense em um SO Windows na sua versão XP (\Windows,

\Documents and Settings, \Arquivos de Programas, entre outros), estão atribuídos a uma unidade de disco rotulada C:\ , como é normal em instalações padrão deste sistema.

7.5.1 O que procurar primeiro

Segundo Hewitt e Peláez (2010) a quantidade de informações encontradas durante uma pesquisa forense é elevada, e deve-se estreitar o mais cedo possível o foco, determinando quais informações são relevantes ou não, para o caso em mãos.

Para tal, deve-se verificar primeiro como o usuário interagia com o seu *web browser*, procurando por:

a) **Pesquisas realizadas –** ao verificar as pesquisas realizadas pelo suspeito, o perito poderá não apenas conhecer os assuntos de interesse do mesmo, mas também determinar "palavras-chave" que poderão ser úteis mais tarde, ao se rever os arquivos mantidos pelo sistema;

b) **Histórico de navegação –** compreenderá a maior parte da pesquisa, sendo a informação mais importante a ser pesquisada. É importante lembrar que até sites aparentemente inofensivos podem dar pistas sobre as atividades do suspeito;

c) **Arquivos baixados pela Internet –** numa pesquisa forense em *web browsers* esta é a segunda peça mais importante de informação. Arquivos baixados da Internet são a principal causa de danos em sistemas computadorizados;

d) **Informações fornecidas (Formulários/Senhas) –** podem dar pistas sobre outros locais que o suspeito possa ter visitado, como clientes de email, contas bancárias, entre outros. É importante obter junto a empresa dona do site a ser

verificado, a permissão para tal, visto que o perito passará a se mover não apenas no *browser* do suspeito, mas também num servidor remoto;

e) **E-mails –** os emails do suspeito podem ser conseguidos muitas vezes não apenas no site do cliente de email, mas no próprio *browser* do suspeito, visto que muitos deles, atualmente permitem salvar o conteúdo dos emails no disco rígido, caso o usuário assim o queira;

f) **Cookies –** como já foi explicitado anteriormente, informações importantes podem ser salvas nos *cookies*. Eles são importantes para o perito forense, pois através deles é possível recuperar informações mesmo que o suspeito tenha sido cuidadoso o suficiente de apagar o seu histórico de navegação.

Outros elementos que podem dar informações importantes ao perito são: o *cache* do *browser* e os arquivos temporários de Internet, bem como a pasta de sites favoritos do suspeito, ou arquivos com a extensão *.url* (GALVÃO, 2008).

O perito deverá saber onde buscar tais informações, e para tal faz-se necessário que conheça, além da estrutura do sistema de arquivos do SO, como o *browser* usado salva as informações do utilizador e onde. Abaixo são mostrados os caminhos onde os *browsers* IE e Firefox, respectivamente, salvam essas informações, para que o perito possa coletá-las como evidências.

7.5.2 Internet Explorer

O Internet Explorer salva as informações de acesso de cada usuário em seu *profile* Windows, sendo possível encontrá-las nos seguintes caminhos (GALVÃO, 2008):

a) **Informações de cache –** C:\Documents and Settings\<usuário>\Local Settings\Temporary Internet Files\Content.IE5\;

b) **Histórico de navegação –** C:\Documents and Settings\<usuário>\Local Settings\History\History.IE5;

c) **Cookies –** C:\Documents and Settings\<usuário>\Cookies\;

d) **Arquivos temporários –** C:\Documents and Settings\<usuário>\Local Settings\Temp\.

Em cada caminho citado acima e ao longo do trabalho deve-se trocar <*usuário*> pelo nome de usuário do indivíduo a ser investigado.

As informações de *cache*, *cookies* e histórico de navegação são salvas em arquivos *index.dat* encontrados nos caminhos acima, e que podem ser visualizados com ferramentas específicas abordadas no próximo capítulo.

Existe um quarto arquivo *index.dat* localizado na pasta C:\Documents and Settings\<usuário>\UserData\, que é usado pelo SO para realizar tarefas periódicas como atualizações, mas que pode conter também, referências a sites visitados. Embora seja difícil que, evidências relacionadas ao uso que o usuário faz do seu *browser* sejam encontradas nele, é recomendado que o perito o verifique (JONES, 2005a).

Os *index.dat* encontram-se localizados nos caminhos especificados acima nas versões 2000, XP e 2003 do Windows, sendo que da versão Vista em diante é possível encontrá-los nos caminhos (HEWITT; PELÁEZ, 2010):

a) **Informações de cache –**

C:\Users\<usuário>\AppData\Local\Microsoft\Windows\Temporary Internet Files\Low\Content.IE5;

b) **Cookies –** C:\Users\<usuário>\Local\Microsoft\Windows\Cookies\Low;

c) **Histórico de navegação –**

C:\Users\<usuário>\AppData\Local\Microsoft\Windows\History\History.IE5

ou

C:\Users\<usuário>\AppData\Local\Microsoft\Windows\History\History.IE5\Low\.

O diretório *Low*, visualizado nos caminhos acima, é criado pelo SO quando o IE (versões 7 e 8) é utilizado no modo protegido (vem acionado por padrão). Somente nesses diretórios o SO aceita que o IE escreva arquivos, contendo assim arquivos maliciosos, e impedindo-os de danificar o resto do sistema.

O diretório *UserData*, citado anteriormente, pode ser encontrado nas versões Vista em diante, no caminho: C:\Users\<usuário>\AppData\Local\Microsoft\Internet Explorer\UserData\Low.

Ao abrir os arquivos *index.dat* o perito encontrará informações como (GALVÃO, 2008):

a) URLs visitadas;
b) nomes de arquivos armazenados localmente;
c) cabeçalhos (*headers*) HTTP;
d) *timestamps* de arquivos (último acesso, última modificação, e outras informações relacionadas).

É importante notar que alguns dos diretórios citados, estão marcados pelo SO como pastas do sistema. Tal fato pode dificultar o seu acesso através do programa Windows Explorer, ou ainda impedir a sua visualização. Nesses casos, o perito deve acessar os diretórios e arquivos mencionados por meio do terminal de comandos do SO, através do comando *CD* e tornando-os visíveis através do comando *DIR/AS* do Windows. Outra possibilidade, seria desmarcar a opção de <*Ocultar arquivos protegidos do sistema*> encontrada no menu *Ferramentas/Opções de Pastas/Modo de exibição*, existente em cada janela aberta pelo Explorer.

Mais um item importante a ser examinado, é a pasta de favoritos do usuário. O perito deve duplicar a pasta onde o IE salva as mesmas C:\Documents and Settings\<usuário>\Favorites\ para um computador pericial, pois ao acessar as URLs e identificar o seu conteúdo pode ocorrer alguma alteração do mesmo (GALVÃO, 2008). Uma ferramenta capaz de auxiliar na visualização dos sites favoritos, tal como eles foram vistos pelo usuário no seu último acesso, seria o site *WayBackMachine*, que pode ser acessado na URL *http://waybackmachine.org/,* e que permite visualizar o conteúdo da página em uma data no passado (GALVÃO, 2008).

Também é importante averiguar o registro do Windows, uma vez que o IE salva nele informações do usuário (nomes, endereços, senhas, entre outras) de forma encriptada. Caso o recurso *AutoComplete* do IE esteja ativado, permitindo ao usuário relembrar senhas e outras informações suas ao preencher, por exemplo, um formulário, os seguintes caminhos no registro do Windows devem ser examinados (HEWITT; PELÁEZ, 2010):

a) **Para versões 4 à 6 do IE –** O perito forense deve procurar no registro do Windows no caminho: HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\IntelliForms\SPW. Para recuperar URLs, senhas, endereços e outros campos de formulários ele deve procurar em: HKEY_LOCAL_MACHINE\Software\Microsoft\Protected Storage System Provider\<usuário>. Também é possível encontrar uma lista das URLs acessadas recentemente em: HKEY_USERS/Default/Software/Microsoft/Internet Explorer/TypedURLs/ ou HKEY_CURRENT_USER/Software/Microsoft/Internet Explorer/TypedURLs. Esta lista não está encriptada e pode ser visualizada diretamente no *browser* através do protocolo *file:///*caminho_do_arquivo. A criptografia dos dados

nestas versões é fraca e pode facilmente ser quebrada com ferramentas próprias para esse propósito;

b) **Para versões 7 e 8 do IE –** Informações encriptadas do usuário como senhas, endereços, entre outras, são agora encontradas no caminho: HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\IntelliForms\Storage2. Sendo a criptografia nestas versões melhorada, e muito mais difícil de ser quebrada.

Um último aspecto a ser ressaltado é que, apesar do IE salvar as informações de *cookies* em um arquivo *index.dat*, ele também os salva em arquivos de texto puro com a extensão *.txt*. Assim sendo, não é necessário nenhum programa específico além de um programa editor de texto padrão para ter acesso as informações. No entanto, para melhor compreensão e mais fácil visualização dos dados o perito pode usar ferramentas específicas para esse fim.

7.5.3 Firefox

No Firefox são seis os arquivos que o perito deve procurar, para acessar as informações do usuário do *browser*: *places.sqlite* (para informações do histórico de navegação), *CACHE_MAP* (que contém as informações de *cache*), *downloads.sqlite* (para os arquivos baixados), *formhistory.sqlite* (para campos preenchidos em formulários), *cookies.sqlite* (para informações de *cookies*), e finalmente *signons.sqlite* (que contém senhas e nomes de usuário encriptados) (HEWITT; PELÁEZ, 2010).

Cinco dos seis arquivos a serem examinados encontram-se no mesmo local (GALVÃO, 2008).:

a) **Windows Xp –** C:\Documents and Settings\<usuário>\Application Data\Mozilla\Firefox\Profiles\<caracteres-randômicos>.default\;

b) **Windows Vista em diante –** C:\Users\<usuário>\AppData\Local\Mozilla\Firefox\Profiles\<caracteres-randômicos>.default\.

Sendo que as URLs favoritas do usuário, também se encontram salvas nas localizações acima citadas.

As informações de *cache* são salvas em outra localização, e são mais difíceis de recuperar que no IE, uma vez que ficam em diferentes subpastas dependendo do quanto de espaço o *browser* necessita para o seu armazenamento (HEWITT; PELÁEZ, 2010). Apenas através de softwares específicos mencionados no próximo capítulo, a recuperação dos arquivos de *cache,* no Firefox, é facilitada.

Os caminhos para encontrar os arquivos de *cache* são (HEWITT; PELÁEZ, 2010):

a) **Windows Xp –** C:\Documents and Settings\<usuário>\Local Settings\Application Data\Mozilla\Firefox\Profiles\<caracteres-randômicos>.default\Cache\;

b) **Windows Vista em diante –** C:\Users\<usuário>\AppData\Local\Mozilla\Firefox\Profiles\<caracteres-randômicos>.default\Cache\.

É importante mencionar que o Firefox usa uma biblioteca de software conhecida como *SQLite*, que implementa em um único arquivo, e sem a necessidade de configuração, um banco de dados transacionais para armazenamento das informações do usuário. Tanto no Windows como nos sistemas derivados do UNIX (GALVÃO, 2008)..

Para ler os arquivos com extensão *.sqlite* do Firefox, o perito poderá usar algum programa específico que use a biblioteca *SQLite*, e permita a execução de consultas na linguagem Structured Query Language (SQL) no banco de dados.

7.5.4 Se o Usuário Deletar o Cache do Browser

Segundo Hewitt e Peláez (2010), na versão 6 do IE, se o usuário deletar o *cache* do *browser* os arquivos correspondentes são deletados do disco rígido, mas as entradas no arquivo *index.dat* são apenas marcadas como livres e não removidas. Tal fato permite ao perito forense, em alguns casos, identificar as URLs visitadas pelo usuário, mesmo depois de ele pensar que as deletou. Eventualmente, o IE subscreverá as entradas a medida que necessitar de espaço em disco, ou se fizer uma manutenção de sistema, onde ele compara os arquivos em *cache* com as entradas do arquivo *index.dat*.

Já nas versões 7 e 8 do IE, ao deletar o *cache* do *browser* são deletados os arquivos temporários de Internet, e subscritas as entradas do arquivo *index.dat*, dificultando o trabalho do perito forense. Nesses casos, o mesmo pode tentar recuperar os arquivos temporários deletados, usando ferramentas próprias para esse propósito, uma vez que o SO não os apaga realmente do disco, marcando-os apenas como livres (HEWITT; PELÁEZ, 2010).

No Firefox a situação para o perito forense complica-se ainda mais. Uma das razões que o faz ser visto como um *browser* mais seguro pelo usuário, é o fato de por padrão, ele vir com mais opções de segurança ativadas. Com relação ao *cache* do *browser* isso continua a ser verdadeiro, visto que o mesmo apresenta uma postura mais agressiva com relação ao apagar as informações salvas, desde o início. Ele apaga de uma só vez os arquivos

temporários salvos, as entradas salvas no arquivo *CACHE_MAP*, e outros ficheiros de *cache* espalhados pelo sistema (HEWITT; PELÁEZ, 2010).

O perito é então obrigado a recuperar os arquivos relacionados, conduzindo uma busca por arquivos deletados no disco rígido, para depois, caso encontre algo, conduzir uma análise.

7.5.5 Se o Usuário Empregar Navegação Privativa

Os *browsers* aqui estudados (IE e Firefox), bem como a maioria dos concorrentes, implementam atualmente um modo de navegação privativa, que previne que o histórico de navegação, arquivos temporários de Internet, dados de formulários, *cookies*, nomes de usuário e senhas sejam salvos neste modo (HEWITT; PELÁEZ, 2010). Segundo Burzstein *et al,* (2010) esse modo de navegação nos *browsers* tem dois motivos:

a) dar mais privacidade ao usuário em computadores compartilhados. A navegação em modo privativo não deve deixar qualquer rastro no computador, logo um familiar que pesquise o histórico de navegação, não deverá encontrar evidências de sites visitados nesse modo;
b) salvaguardar as informações do usuário, de ataques locais. Se o computador do usuário for invadido no horário X, as informações usadas pelo *browser* em modo privativo, antes desse horário não deverão estar comprometidas.

Porém, por ser uma técnica ainda recente, existe uma grande inconsistência no modo como diferentes *browsers* tratam a navegação privativa. Por exemplo, enquanto o Chrome desabilita o uso de *plugins* durante a navegação privativa, o Firefox não os desabilita favorecendo a usabilidade à segurança (BURZSTEIN *et al*, 2010).

O uso do navegador neste modo, dificulta a vida do perito forense, mas ele pode ainda assim, encontrar informações importantes salvas em arquivos *.dat* ou *.sqlite,* como os cabeçalhos HTTP das páginas. Outra maneira de se conseguir informações seria conduzindo a análise para o site ou servidor acessado pelo usuário (lado do servidor), uma vez que, o modo de navegação privativa não impede que sites visitados pelo mesmo salvem informações como, o IP da máquina que acessou, ou dados digitados (HEWITT; PELÁEZ, 2010).

7.6 FERRAMENTAS FORENSE

Algumas evidências coletadas pelo perito forense podem ser visualizadas com o uso de ferramentas comuns, como editores de texto, ou ainda no próprio browser, mas o uso de ferramentas específicas é recomendado devido as seguintes vantagens que elas proporcionam (GALVÃO, 2008):

a) identificação automática da localização de arquivos;
b) resolução de problemas, como nomes diferentes de arquivos (em função, por exemplo, do idioma ou versão do SO);
c) *parser* automático de arquivos codificados;
d) uso em vários tipos de browsers;
e) apresentação mais agradável das informações;
f) relatórios mais detalhados;
g) exportação para formatos manipuláveis.

Existem várias ferramentas que permitem realizar uma análise forense em *web browsers* auxiliando um perito, como: Web Historian, Pasco, NetAnalysis, Galleta, Cache View, IE HistoryView, IE CookiesView, Mozilla HistoryView, Mozilla CookiesView, Mozilla CacheView, Web Browser Forensics (WBF), Firefox 3 Extractor, Cache Monitor,

Internet Cache Explorer, IE Cache Auditor, Web Cache Illuminator, STG Cache Audit, Index.dat Analyzer, Forensic Tool Kit, IE History Manager, EnCase, Autopsy / Sleuthkit, entre outras.

Para o presente trabalho foram selecionadas ferramentas *open source*, ou seja, softwares que sejam disponibilizados sob uma licença de código aberto, e softwares livres, que embora não tenham o seu código fonte disponibilizado, são gratuitamente distribuídos. A vantagem de se trabalhar com software livre é clara, o baixo custo de aquisição. Já os benefícios de se trabalhar com softwares de código aberto são segundo Argolo (2005):

a) **Baixo Custo –** softwares sob a licença *open source* são normalmente gratuitos, ou o seu custo de aquisição é muito reduzido;
b) **Segurança –** o fato de o seu código fonte ser disponibilizado a comunidade, faz com que ele seja regularmente analisado e melhorado;
c) **Continuidade –** caso os desenvolvedores originais do software descontinuem as atualizações do mesmo, a comunidade pode fazê-lo, ou o código fonte pode ainda ser usado para iniciar outros projetos;
d) **Flexibilidade –** o código fonte do software, pode ser modificado para melhor satisfazer um usuário, atendendo a características específicas. Entre outras vantagens.

As ferramentas escolhidas e que serão melhor apresentadas a seguir são: Pasco, Galleta, Web Historian, e Firefox 3 Extractor. Elas são todas gratuitas, disponibilizando algumas o seu código fonte, e são amplamente utilizadas por peritos durante as suas investigações.

Outras ferramentas consideradas importantes, e que podem, caso surja a chance, ser usadas no presente trabalho são:

a) **MozillaCacheView –** é uma ferramenta gratuita que permite ao perito procurar

e visualizar os arquivos de *cache* do *browser* Firefox de uma maneira mais facilitada e amigável;

b) **IECacheView –** esta ferramenta realiza a mesma tarefa que a anterior, com a particularidade de ela só trabalhar com arquivos do IE;

c) **MozillaCookiesView –** é uma ferramenta gratuita, que realiza a mesma função do Galleta, posteriormente citado, que é a de exibir os *cookies* salvos no *browser*. Neste caso os arquivos reconhecidos são os do Firefox;

d) **PasswordFox** – ferramenta que permite recuperar nomes de usuário e senhas salvas pelo Firefox.

A seguir são melhor apresentadas as principais ferramentas a serem usadas no presente trabalho.

7.6.1 Pasco

A ferramenta Pasco é de autoria de Keith Jones e possui uma licença de código aberto. O seu nome vem do latim significando "busca", e o seu foco principal é a análise de arquivos de *cache* do IE. Ela possui versões para Windows (Cygwin), Linux, Mac OSX, e BSDs.

A sua interface é em linha de comando, e o seu funcionamento básico é o seguinte: o programa lê um arquivo *index.dat*, reconstrói os registros internos do arquivo, e retorna a informação em um arquivo no formato texto (GALVÃO, 2008).

A ferramenta apresenta-se como uma solução prática quando se quer, por exemplo, exportar os dados para uma planilha como o Excel.

O Pasco pode ser executado em dois modos diferentes: o modo padrão, já explicado acima, ou um modo conhecido como *undeletion*. Neste modo, a ferramenta ignora

as informações da tabela *hash* e reconstrói todos os registros de atividade válida até o limite de 0x80 *bytes*, podendo recuperar informações de atividades que não teriam sido encontradas por outras técnicas e ferramentas (JONES, 2005b).

Os comandos de uso do Pasco na linha de comandos são relativamente simples e são ilustrados na Figura 9.

```
[ kjones: pasco] kjones% ./pasco

Usage:  pasco [ options] <filename>
        -d Undelete Activity Records
        -t Field Delimiter (TAB by default)
```

Figura 9. Comandos de uso da ferramenta Pasco
Fonte: adaptado Jones, K. (2003, p. 28)

A opção *-d* ativa o modo *undeletion* do Pasco. Já a opção *-t* permite que o perito forense mude o delimitador de separação das informações, sendo por padrão o TAB, facilitando a abertura em planilhas (JONES, 2003). Um exemplo de uso da ferramenta é ilustrado pela Figura 10.

```
% ./pasco index.dat > index.txt
```

Figura 10. Exemplo do uso da ferramenta Pasco
Fonte: adaptado GALVÃO, R. (2008)

E podemos verificar na Figura 11, um exemplo de como é o arquivo excel exportado pela ferramenta.

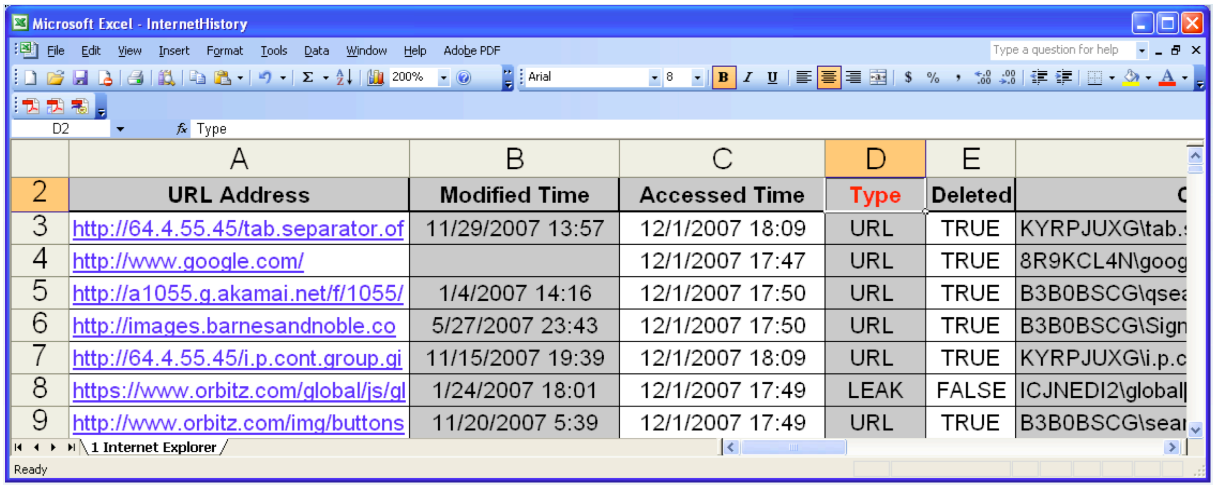


Figura 11. Exemplo de arquivo excel exportado pela ferramenta Pasco
Fonte: adaptado GALVÃO, R. (2008)

7.6.2 Galleta

Outra ferramenta de autoria de Keith Jones e que possui também uma licença de código aberto é o Galleta. O seu nome vem do espanhol que se traduz para inglês como *cookie*. A principal função desta ferramenta é a análise dos arquivos de *cookie* gerados pelo IE. Assim como o Pasco, o Galleta também possui versões para Windows (Cygwin,), Mac OSX, Linux, e BSDs.

A sua interface também é disponibilizada somente através da linha de comandos, sendo o seu funcionamento básico, similar ao da ferramenta anterior. O Galleta recebe um arquivo, reconstrói os seus registros de uma maneira legível e os exporta em um arquivo de texto (GALVÃO, 2008). Ele também apresenta a funcionalidade prática de exportar os registros num formato padronizado para abertura em planilhas como o Excel.

Os comandos de uso do Galleta são ilustrado pela Figura 12.

```
Uso: galleta [opções] <nome_do_arquivo>

-t Campo delimitador (TAB por default)
```

Figura 12. Comandos de uso da ferramenta Galleta
Fonte: adaptado GALVÃO, R. (2008)

A opção *-t* permite que o perito mude o delimitador de separação das informações, que tal como no Pasco, por padrão é o TAB (GALVÃO, 2008).

A Figura 13, apresenta um exemplo de como usar a ferramenta.

Exemplo de Uso:

```
% ./galleta arquivoexemplo.txt > cookies.txt
```

Figura 13. Exemplo do uso da ferramenta Galleta
Fonte: adaptado GALVÃO, R. (2008)

7.6.3 Web Historian

Este programa possui uma licença de software livre, e o seu código fonte não é disponibilizado com ele. A ferramenta foi desenvolvida por Red Cliff's, sendo o seu principal objetivo ler os arquivos responsáveis por salvar o histórico de navegação do usuário, e apresentá-los de uma maneira mais amigável (JONES, 2003).

Esta ferramenta só executa no SO Windows, sendo que o seu principal benefício para o perito é o fato de ela reconhecer arquivos importantes dos seguintes navegadores: IE, Firefox, Safari e Opera. Ou seja, apenas com uma ferramenta o perito consegue ter acesso ao histórico de navegação de quatro dos principais browsers do mercado.

O programa exporta as informações nos formatos: *xls*, *html*, e *txt*, o que também oferece certa flexibilidade ao perito na altura de analisar as evidências.

Uma grande limitação do programa é que, até o atual momento, ele não executa nas versões Vista em diante do SO Windows (GALVÃO, 2008).

O Web Historian apresenta-nos duas abordagens para a obtenção de um arquivo do *browser*, como pode ser visualizado na Figura 14. É importante ser cuidadoso ao escolher

uma das abordagens, pois nem todos os métodos funcionam para cada tipo de arquivo. O perito pode mesmo chegar a travar o programa durante a sua investigação (HEWITT; PELÁEZ, 2010).

O primeiro método, permite que o perito navegue até um arquivo *.dat* ou *.sqlite*, fazendo uso da interface do SO. Este método funciona para arquivos que não estão nos seus locais padrão, por exemplo, um arquivo copiado de outro computador. O segundo método, permite que o perito especifique uma pasta ou diretório para o programa procurar pelo arquivo *.dat* ou *.sqlite* correspondente. A pesquisa irá incluir subpastas da pasta especificada, portanto o perito deve ter cuidado e especificar uma pasta por vez, correndo o risco de travar o programa caso contrário (HEWITT; PELÁEZ, 2010).

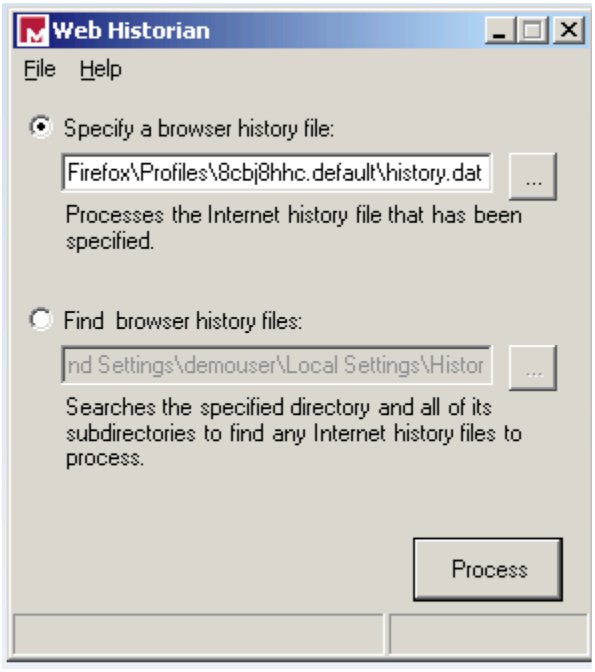
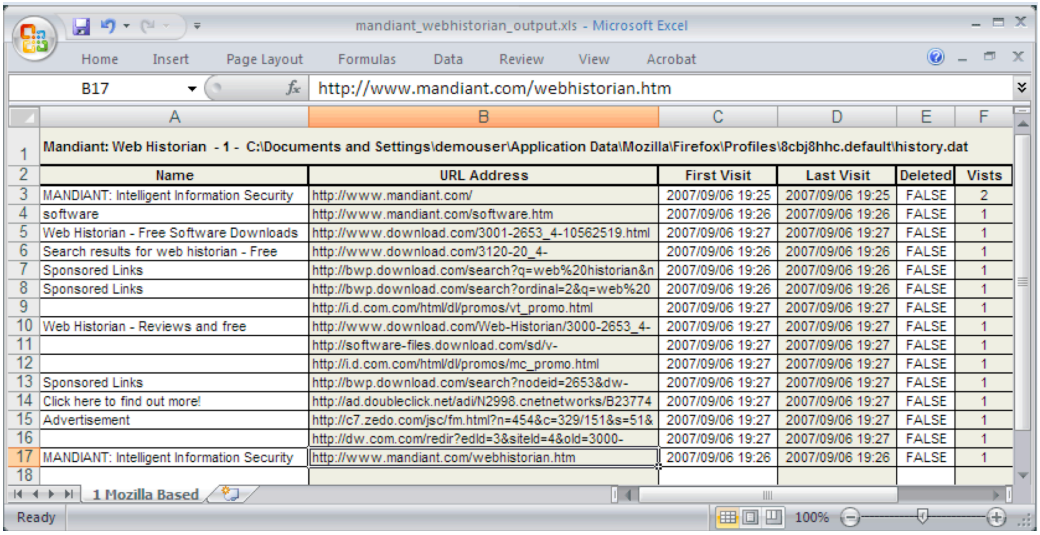


Figura 14. Tela de escolha do arquivo .dat no Web Historian
Fonte: adaptado GALVÃO, R. (2008)

As informações exportadas pelo Web Historian são semelhantes as da ferramenta Pasco citada no início, contendo algumas informações complementares como: estado de deleção (mostra se o usuário apagou a URL visitada ou não), e o número total de visitas a determinada URL, como ilustra a Figura 15.



mandiant_webhistorian_output.xls - Microsoft Excel

Home Insert Page Layout Formulas Data Review View Acrobat

B17 http://www.mandiant.com/webhistorian.htm

Mandiant: Web Historian - 1 - C:\Documents and Settings\demouser\Application Data\Mozilla\Firefox\Profiles\8cbj8hhc.default\history.dat

Name	URL Address	First Visit	Last Visit	Deleted	Vists
MANDIANT: Intelligent Information Security	http://www.mandiant.com/	2007/09/06 19:25	2007/09/06 19:25	FALSE	2
software	http://www.mandiant.com/software.htm	2007/09/06 19:26	2007/09/06 19:26	FALSE	1
Web Historian - Free Software Downloads	http://www.download.com/3001-2653_4-10562519.html	2007/09/06 19:27	2007/09/06 19:27	FALSE	1
Search results for web historian - Free	http://www.download.com/3120-20_4-	2007/09/06 19:26	2007/09/06 19:26	FALSE	1
Sponsored Links	http://bwp.download.com/search?q=web%20historian&n	2007/09/06 19:26	2007/09/06 19:26	FALSE	1
Sponsored Links	http://bwp.download.com/search?ordinal=2&q=web%20	2007/09/06 19:26	2007/09/06 19:26	FALSE	1
	http://i.d.com.com/html/dl/promos/vt_promo.html	2007/09/06 19:27	2007/09/06 19:27	FALSE	1
Web Historian - Reviews and free	http://www.download.com/Web-Historian/3000-2653_4-	2007/09/06 19:27	2007/09/06 19:27	FALSE	1
	http://software-files.download.com/sd/v-	2007/09/06 19:27	2007/09/06 19:27	FALSE	1
	http://i.d.com.com/html/dl/promos/mc_promo.html	2007/09/06 19:27	2007/09/06 19:27	FALSE	1
Sponsored Links	http://bwp.download.com/search?nodeid=2653&dw-	2007/09/06 19:27	2007/09/06 19:27	FALSE	1
Click here to find out more!	http://ad.doubleclick.net/adi/N2998.cnetnetworks/B23774	2007/09/06 19:27	2007/09/06 19:27	FALSE	1
Advertisement	http://c7.zedo.com/jsc/fm.html?n=454&c=329/151&s=51&	2007/09/06 19:27	2007/09/06 19:27	FALSE	1
	http://dw.com.com/redir?edId=3&siteId=4&old=3000-	2007/09/06 19:27	2007/09/06 19:27	FALSE	1
MANDIANT: Intelligent Information Security	http://www.mandiant.com/webhistorian.htm	2007/09/06 19:26	2007/09/06 19:26	FALSE	1

1 Mozilla Based

Ready 100%

Figura 15. Exemplo de arquivo excel exportado pela ferramenta Web Historian
Fonte: adaptado GALVÃO, R. (2008)

7.6.4 Firefox 3 Extractor

A ferramenta Firefox 3 Extractor (F3E) possui uma licença de software livre, e o seu principal objetivo é extrair dados dos arquivos *.sqlite* usados pelo Firefox e pelo Google Chrome para salvar o *cache*, o histórico de navegação, histórico de downloads, *cookies*, e outros dados do usuário. A sua interface é disponibilizada apenas pela linha de comandos, e a sua execução é restrita ao SO Windows.

Ela funciona de maneira bastante simples, necessitando o perito apenas, de transferir para o mesmo diretório do programa as evidências encontradas, e executá-lo. Como pode ser visualizado na figura 16, o programa apresenta na linha de comandos um menu de opções, que permite:

a) extrair todos os dados do Firefox e exportá-los para um arquivo *.csv*;

b) extrair todos os dados do Firefox e exportá-los para um arquivo *.csv* decodificando também as datas e horários;

c) criar um relatório do histórico de navegação do usuário com extensão *.csv*;

d) criar um relatório do histórico de navegação do usuário com extensão *.html*;

e) decodificar PRTime;

f) extrair todos os dados do Chrome e exportá-los para um arquivo *.csv*;

g) extrair todos os dados do Chrome e exportá-los para um arquivo *.csv* decodificando também as datas e horários;

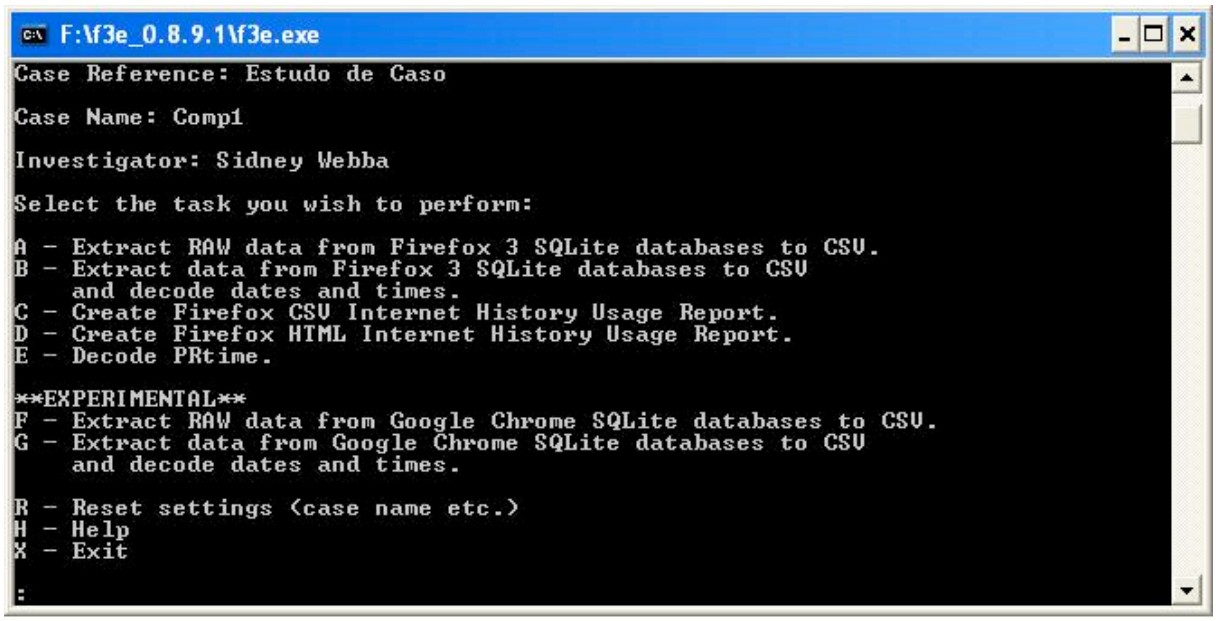


Figura 16. Menu de Navegação da Ferramenta F3E

Uma das grandes vantagens que a ferramenta apresenta, é o fato de com um simples comando o perito ter todas as evidências por ele coletadas, concernentes ao Firefox ou ao Chrome, convertidas para formatos mais legíveis que facilitam a análise, otimizando assim o tempo de realização da perícia.

Também é importante mencionar que a ferramenta converte as datas e horários para formatos legíveis. O Firefox, por exemplo, usa um formato de data e hora conhecido como PRTime, que é um inteiro de 64 bits representando o número de microssegundos desde a meia-noite de 1 de Janeiro de 1970. Como tais informações são cruciais para a investigação, é importante que o perito trabalhe com ferramentas que façam tal conversão.

7.6.5 Tabela Comparativa de Ferramentas Forense

Para melhor se verificar as características em comum, bem como as peculiaridades de cada ferramenta forense escolhida para a presente pesquisa, foi elaborada uma tabela comparativa entre as mesmas, que pode ser observada abaixo.

Nome	SO	Função	Interface	Browsers Suportados	Formatos Exportáveis	Dados Coletados
Pasco	Windows (Cygwin), Mac OSX, Linux, BSD	Análise de arquivos de *cache.*	Linha de Comandos.	IE.	.xls, .csv, .txt, .html	Mostra se a URL foi acessada pelo usuário ou redirecionada, mostra a URL visitada, a data da última mudança sofrida pelo site, a data de acesso do usuário, o nome do arquivo com uma cópia da URL listada, o diretório onde o arquivo se encontra, e os cabeçalhos HTTP recebidos.
Galleta	Windows (Cygwin), Mac OSX, Linux, BSD	Análise de arquivos de *cookie.*	Linha de Comandos.	IE.	.xls, .csv, .txt, .html	Mostra a URL do site que salvou o cookie, a variável que o cookie salvou e o seu valor, a data e o horário de criação, a data e o horário de expiração, e Flags.
Web Historian	Windows	Análise do Histórico de Navegação.	Interface Gráfica.	IE, Firefox, Safári, Opera.	.xls, .csv, .txt, .html	Mostra a URL visitada, a data da última alteração sofrida pelo site, a data de acesso do usuário, se a URL foi acessada pelo usuário ou redirecionada, se o usuário deletou a URL do histórico, os ficheiros em cache da URL, e os cabeçalhos HTTP recebidos.
F3E	Windows	Análise de arquivos.*sqlite* do Firefox e do Chrome.	Linha de Comandos.	Firefox, Google Chrome.	.html, .csv	Mostra as 20 URLS mais visitadas, a data de acesso, a URL visitada, o Titulo da página, e se a URL foi acessada pelo usuário ou redirecionada.

Figura 17. Tabela Comparativa de Ferramentas Forense

No próximo capítulo são apresentados alguns trabalhos estudados com propósitos semelhantes a presente pesquisa.

8 TRABALHOS CORRELATOS

No decorrer dos estudos objetivando esta pesquisa, seja na proposta ou no seu desenvolvimento, foram analisados alguns trabalhos com propósitos semelhantes , mas cujo foco era outro. Abaixo é feita a descrição de alguns trabalhos escolhidos, envolvendo a perícia forense computacional, e a perícia forense em web browsers.

8.1 PERÍCIA FORENSE EM WEB BROWSERS

O primeiro trabalho estudado foi a apresentação mostrada em palestra pelo professor do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, Ricardo Galvão. A palestra teve lugar em porto alegre no Fórum Internacional de Software Livre (FISL), na sua nona edição em 2008.

A apresentação aborda de maneira rápida e resumida o que é uma perícia forense computacional, como se deve proceder para realizar a mesma em web browsers, e quais as principais ferramentas disponíveis que auxiliam o perito na sua tarefa.

Uma compilação dos slides usados em pdf, pode ser encontrada no seguinte endereço: ftp://ftp.registro.br/pub/gts/gts11/01-forenseweb.pdf.

8.2 PERÍCIA FORENSE: PROPOSTA DE UMA METODOLOGIA DE COLETA DE INDÍCIOS PARA AMBIENTE WINDOWS

Este trabalho apresentado no Centro Universitário Feevale como trabalho de conclusão de curso, foi elaborado por Daniel Bertoglio, e foi o segundo a ser estudado para a presente pesquisa.

O trabalho tem como meta abordar e explicar o tema de perícia forense computacional, propondo uma metodologia de coleta de indícios para um ambiente Windows.

Ele pode ser visualizado no seguinte endereço: http://tconline.feevale.br/tc/files/0001_1690.pdf.

8.3 ANÁLISE FORENSE EM SISTEMAS GNU/LINUX

O terceiro trabalho estudado para a presente pesquisa, foi elaborado por Frederico Argolo na Universidade Federal do Rio de Janeiro (UFRJ) como um projeto final de graduação.

O trabalho tem como objetivo explicar os principais procedimentos adotados por especialistas no campo de análise computacional forense, e associá-los ao contexto nacional. Bem como aplicar o uso de algumas das principais ferramentas existentes, em ambiente de laboratório. Ele pode ser encontrado no seguinte endereço: http://www.ravel.ufrj.br/arquivosPublicacoes/projetofinal_fred.pdf.

No próximo capítulo são apresentados: o estudo desenvolvido, e a metodologia usada para desenvolver o mesmo, visando ratificar a pesquisa.

9 TRABALHO DESENVOLVIDO

O presente estudo de caso foi realizado na Universidade do Extremo Sul Catarinense (UNESC), localizada em Criciúma/SC. A instituição possui uma estrutura com 27 laboratórios de informática de grande porte (até 24 computadores), e 6 laboratórios de pequeno porte (até 12 computadores), sendo os laboratórios 13 e 14 do Bloco XXI-C reservados para uso livre da comunidade interna da mesma. A comunidade externa da universidade tem disponível, Internet gratuita nos computadores da biblioteca.

Nos computadores disponíveis para a comunidade interna, é necessário possuir um código de matrícula que é requisitado juntamente com uma senha na hora da efetuar o login[1] nas máquinas.

Para o controle e segurança dos laboratórios, já que muitas pessoas acessam as máquinas, a instituição possui uma política de segurança adequada ao ambiente pedagógico, principalmente no quesito de uso da Internet. Assim sendo, os sites são categorizados e o bloqueio efetivo de alguns deles é realizado. Além da política de segurança, é possível encontrar as Normas de Utilização dos Laboratórios no Departamento de Tecnologia da Informação da UNESC, onde funciona a coordenação dos Laboratórios de Informática (LabInfo) que é responsável por essa estrutura. Estas normas também estão fixadas em cada um dos laboratórios.

Para a realização da perícia forense e aplicação da metodologia SOP foram escolhidos aleatoriamente os laboratórios 16 do Bloco XXI-A e 8 do Bloco XXI-B, e da mesma forma, delimitaram-se 10 computadores a serem analisados em cada sala. Faz-se necessário mencionar, entretanto, que o correto na realização de uma pericia forense computacional desse tipo, seria a análise dos computadores de todos os laboratórios da

[1] Login é o procedimento de logar-se na rede, ou em qualquer outro serviço informando seu nome de usuário e senha (MORIMOTO, 2003).

instituição. Por se tratar de um caso fictício, optou-se por escolher uma amostra do total de máquinas, realizando-se a pericia em 20 estações.

Agora, para melhor contextualizar o estudo de caso fictício, e permitir uma compreensão facilitada das etapas realizadas, foi suposto que o seguinte crime digital foi cometido:

- Alguém, usando um dos computadores dos laboratórios da UNESC e um *browser,* acessou de maneira indevida o Sistema de Vendas Online do Supermercado HLT (nome fictício) com fins de prejudicar o mesmo.

A seguir, são expostos os conceitos relevantes a todo processo de metodologia científica aplicado neste trabalho.

9.1 METODOLOGIA

A pesquisa tem como embasamento um estudo de caso fictício, que simula a ocorrência de uma perícia forense nos computadores dos laboratórios da Universidade do Extremo Sul Catarinense (UNESC), objetivando buscar conhecimento detalhado sobre os procedimentos de interesse para este estudo, considerando a ocorrência de um crime digital.

Martins e Theóphilo (2009) afirmam que um estudo de caso:

> "trata-se de uma investigação empírica que pesquisa fenômenos dentro de seu contexto real [...] onde o pesquisador não tem controle sobre eventos e variáveis, buscando apreender a totalidade de uma situação e criativamente descrever, compreender e interpretar a complexidade de um caso concreto. [...] possibilita a penetração na realidade social, não conseguida pela avaliação quantitativa".

Por se considerar que cada caso tem as suas características próprias, não existe um modelo traçado de forma específica para elaboração de um estudo de caso, apenas uma sequência de práticas metodológicas, para orientação, que são: coleta das evidências, composição, análise e validação dos resultados, conclusões, verificação de possíveis interferências e relatório final. (MARTINS; THEÓPHILO, 2009).

Logo, para que se cumpram os objetivos desta pesquisa, definiu-se que a metodologia forense Standard Operating Procedures (SOP) será usada durante a realização do estudo, pois ela incorpora algumas das práticas metodológicas recomendadas acima, bem como os princípios e técnicas da ciência forense.

As etapas do modelo proposto podem ser visualizadas na Figura 18, disposta na página seguinte. O fluxograma ilustra cada uma das etapas com algumas das ações a serem tomadas no decorrer da perícia. Tais ações atuam como processos em um projeto, pelo fato de que se é respeitada uma ordem, mesmo que, se necessário, a investigação retorne a determinada etapa.

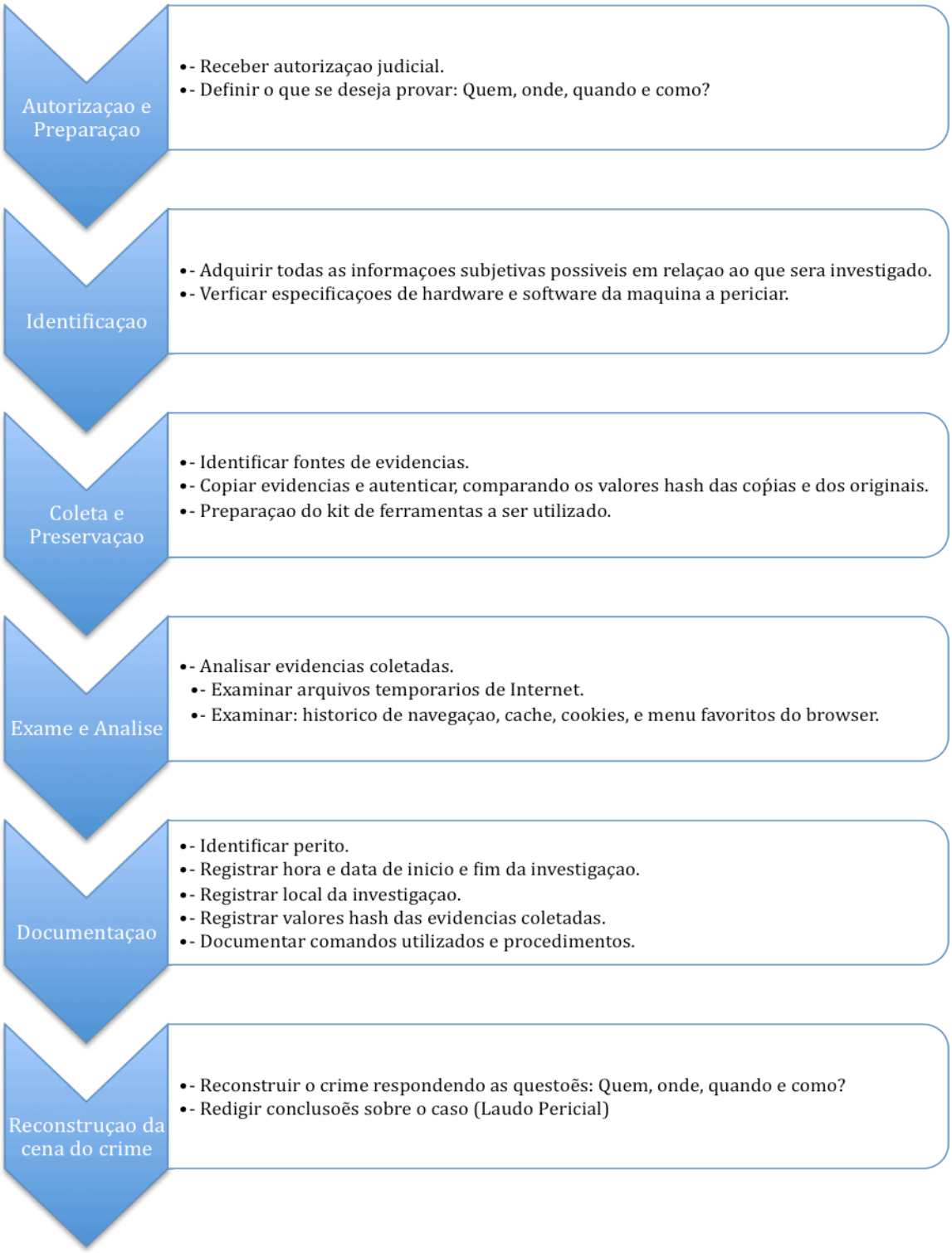


Figura 18. Fluxograma de Etapas da Metodologia SOP

Algo importante de se mencionar, é o fato de que no processo tradicional de análise forense, o perito copia as evidências encontradas para um disco ou computador "limpo", podendo assim examiná-las em um ambiente controlado. Tal abordagem foi aplicada neste trabalho.

A explicação detalhada de como cada etapa da metodologia foi aplicada, é apresentada abaixo.

9.2 ETAPA 1 – AUTORIZAÇÃO E PREPARAÇÃO

Pelo fato de o presente estudo de caso ser fictício, não se fez necessária a busca por uma autorização judicial, para que a perícia forense fosse realizada. No entanto, para que os propósitos da pesquisa se cumpram e a metodologia SOP seja seguida da maneira mais fiel possível, recebeu-se a autorização do Prof. MSc. Rogério Casagrande, professor da disciplina de Redes de Computadores do curso de Ciência da Computação e gerente do Departamento de Tecnologia da Informação da UNESC, para a realização de uma perícia nos computadores da instituição.

Em seguida, definiu-se o escopo da pesquisa, determinando o que se deseja provar. Para tal, estabeleceu-se que pretende-se descobrir ao final da mesma: quem realizou o crime (aluno, professor, funcionário da UNESC, visitante), onde realizou (de que computador o crime foi cometido), quando (em que horário e dia) e como (que procedimentos o infrator usou).

9.3 ETAPA 2 – IDENTIFICAÇÃO

Novamente por se tratar de um estudo de caso hipotético, não foi necessário o levantamento junto as pessoas envolvidas no crime (pessoas que identificaram a ocorrência do crime; pessoas que sofreram as conseqüências do ato criminoso; responsáveis da Empresa lesada e dos laboratórios da UNESC) de informações relevantes, por meio de questionamentos, que permitiriam ao perito contextualizar melhor os fatos que surgissem

durante o decorrer da investigação.

Passou-se então para a aquisição das especificações relevantes de hardware e software das máquinas a serem periciadas. Para tal, acessou-se o programa “Informações do Sistema” encontrado sob o menu *Iniciar/Todos os Programas/Acessórios/Ferramentas do Sistema/*, disponibilizado no Windows XP.

Todos os computadores apresentam as seguintes configurações de hardware:

a) Memória RAM: 2 GB;

b) Disco Rígido: 150 GB;

c) Processador: Intel Core 2 Duo;

d) Velocidade do Processador: 2,26 GHz;

e) Número de Processadores: 1;

f) Número de Núcleos: 2;

g) Placas de áudio, vídeo e rede on-board;

h) Fabricante: HP;

i) Modelo: HP Compaq dc5850 Microtower.

Bem como, as seguintes especificações de software:

a) SO: Windows XP, Service Pack 3;

b) Browsers: IE versão 8.0.6, Firefox versão 3.6.12.

Com o detalhamento de hardware e software é possível de maneira mais facilitada identificar as fontes de evidências digitais, e selecionar que ferramentas serão usadas, preparando-se assim o kit de investigação.

9.4 ETAPA 3 – COLETA E PRESERVAÇÃO

Como já foi mencionado no decorrer do trabalho, o kit de ferramentas a ser

utilizado é composto pelos softwares:

a) Pasco;

b) Galleta;

c) Web Historian;

d) Firefox 3 Extractor;

e) Mozilla Cache View, e;

f) PasswordFox.

Além destes, utilitários nativos do sistema operacional como a linha de comandos do Windows, e outros programas como: *Cygwin* (Emulador de um ambiente semelhante ao Linux, no Windows) e *md5sum* (Gerador de *hashes* MD5 de arquivos) também compõem o kit.

Deu-se então inicio ao processo de recolha de evidências como: arquivos de *cache*, histórico de navegação, *cookies* e arquivos temporários, tanto do IE quanto do Firefox, pela cópia direta dos mesmos nos caminhos especificados em capítulos anteriores do trabalho.

Com relação as evidências coletadas do IE, ao final obteve-se um total de:

g) 10 arquivos de *cache* coletados do laboratório 8;

h) 10 arquivos de *cache* coletados do laboratório 16;

i) 1908 *cookies* coletados do laboratório 8;

j) 757 *cookies* coletados do laboratório 16;

k) 63 arquivos de histórico de navegação coletados do laboratório 8, e;

l) 44 arquivos de histórico de navegação coletados do laboratório 16.

Por sua vez, com relação as evidências coletadas do Firefox, ao final obteve-se um total de:

a) 105 arquivos *.sqlite* coletados do laboratório 8, e;

b) 114 arquivos *.sqlite* coletados do laboratório 16.

Pelo fato de ser um estudo de caso fictício, não se fez necessária a análise de todas as evidências coletadas, sendo que foram analisados:

a) todos os arquivos de *cache* do IE coletados, 20 no total (10 de cada laboratório);

b) 60 arquivos de *cookie* do IE coletados, 30 de cada laboratório;

c) todos os arquivos de histórico de navegação do IE coletados, 107 no total;

d) 110 arquivos *.sqlite* do Firefox coletados, 55 de cada laboratório.

Para a garantia da integridade, foi criado o *hash* de cada evidência encontrada usando o programa *md5sum*, bem como de cada cópia realizada, permitindo verificar que os arquivos analisados são cópias fiéis da evidências localizadas.

A Figura 19 abaixo, exemplifica o uso da ferramenta *md5sum*.

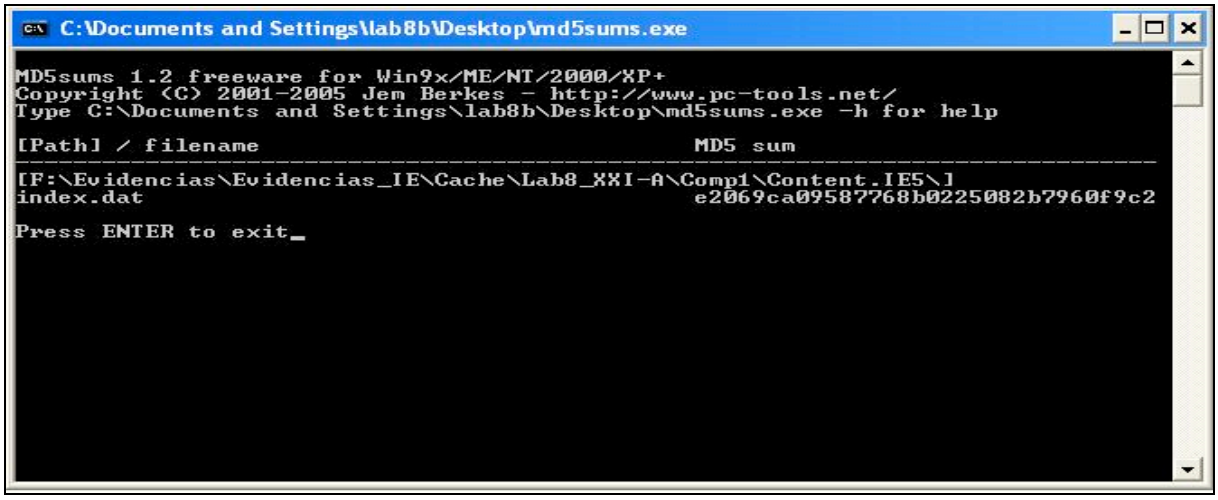


Figura 19. Ferramenta md5sum criando o *hash* de uma evidência

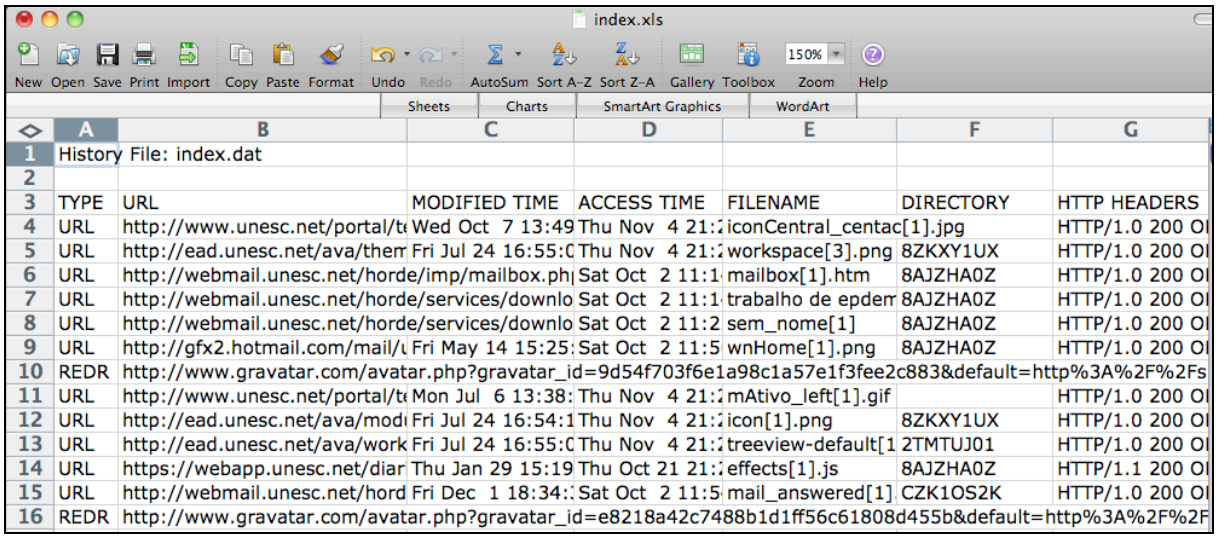
9.5 ETAPA 4 – EXAME E ANÁLISE

A etapa 4 contemplou o processo de conversão das evidências encontradas para formatos de fácil leitura, nomeadamente: *xls* (Ferramentas: Pasco, Galleta e Web Historian), *csv* e *html* (Ferramentas: F3E, Mozilla Cache View e PasswordFox); bem como a análise do conteúdo das evidências para obtenção de informações relevantes ao crime.

Para que os propósitos do atual trabalho se cumpram, acessou-se deliberadamente a URL *http://www.hltlda.com/*, tanto no IE quanto no Firefox, simulando o ataque ao sistema de vendas online fictício que já foi anteriormente mencionado na metodologia. Foram realizadas buscas no site, bem como um cadastramento, objetivando demonstrar que usando as técnicas estudadas e as ferramentas propostas, algumas informações digitadas e visualizadas no browser podem ser recuperadas.

9.5.1 Análise de Arquivos de Cache do IE

A primeira ferramenta a ser usada foi o Pasco, para conversão dos arquivos *index.dat* de *cache* do IE. Um dos arquivos *.xls* obtidos pode ser visualizado abaixo, na Figura 20.


index.xls

New Open Save Print Import Copy Paste Format Undo Redo AutoSum Sort A-Z Sort Z-A Gallery Toolbox 150% Zoom Help

Sheets Charts SmartArt Graphics WordArt

	A	B	C	D	E	F	G
1	History File: index.dat						
2							
3	TYPE	URL	MODIFIED TIME	ACCESS TIME	FILENAME	DIRECTORY	HTTP HEADERS
4	URL	http://www.unesc.net/portal/te	Wed Oct 7 13:49	Thu Nov 4 21:2	iconCentral_centac[1].jpg		HTTP/1.0 200 OI
5	URL	http://ead.unesc.net/ava/them	Fri Jul 24 16:55:0	Thu Nov 4 21:2	workspace[3].png	8ZKXY1UX	HTTP/1.0 200 OI
6	URL	http://webmail.unesc.net/horde/imp/mailbox.ph	Sat Oct 2 11:1	Sat Oct 2 11:1	mailbox[1].htm	8AJZHA0Z	HTTP/1.0 200 OI
7	URL	http://webmail.unesc.net/horde/services/downlo	Sat Oct 2 11:1	Sat Oct 2 11:1	trabalho de epdem	8AJZHA0Z	HTTP/1.0 200 OI
8	URL	http://webmail.unesc.net/horde/services/downlo	Sat Oct 2 11:2	Sat Oct 2 11:2	sem_nome[1]	8AJZHA0Z	HTTP/1.0 200 OI
9	URL	http://gfx2.hotmail.com/mail/t	Fri May 14 15:25:	Sat Oct 2 11:5	wnHome[1].png	8AJZHA0Z	HTTP/1.0 200 OI
10	REDR	http://www.gravatar.com/avatar.php?gravatar_id=9d54f703f6e1a98c1a57e1f3fee2c883&default=http%3A%2F%2Fs					
11	URL	http://www.unesc.net/portal/te	Mon Jul 6 13:38:	Thu Nov 4 21:2	mAtivo_left[1].gif		HTTP/1.0 200 OI
12	URL	http://ead.unesc.net/ava/mod	Fri Jul 24 16:54:1	Thu Nov 4 21:2	icon[1].png	8ZKXY1UX	HTTP/1.0 200 OI
13	URL	http://ead.unesc.net/ava/work	Fri Jul 24 16:55:0	Thu Nov 4 21:2	treeview-default[1	2TMTUJ01	HTTP/1.0 200 OI
14	URL	https://webapp.unesc.net/diar	Thu Jan 29 15:19	Thu Oct 21 21:2	effects[1].js	8AJZHA0Z	HTTP/1.1 200 OI
15	URL	http://webmail.unesc.net/hord	Fri Dec 1 18:34:3	Sat Oct 2 11:5	mail_answered[1]	CZK1OS2K	HTTP/1.0 200 OI
16	REDR	http://www.gravatar.com/avatar.php?gravatar_id=e8218a42c7488b1d1ff56c61808d455b&default=http%3A%2F%2F					

Figura 20. Exemplo de arquivo de cache convertido para o formato .xls

Como se pode observar, o arquivo exportado possui 7 campos para investigação, que são respectivamente:

a) type: Define se a atividade realizada pelo usuário foi o acesso a uma URL diretamente, ou se uma URL foi procurada e redirecionada para outro site (REDR);

b) URL: Exibe a URL do site visitado pelo usuário;

c) modified Time: Mostra a data da última alteração sofrida pelo site;

d) access Time: Aponta a data em que o usuário acessou o mesmo;

e) filename: Exibe o nome do arquivo que contém uma cópia da URL listada;

f) directory: Permite saber o nome do diretório local onde se pode achar o arquivo acima;

g) HTTP headers: Apresenta os cabeçalhos HTTP que o usuário recebeu quando visitou a URL.

Ao analisar as informações acima, procurou-se identificar acessos ao site do sistema em questão, dando atenção especial as datas, tanto de modificação do mesmo no servidor (Modified Time), quanto de visitação pelo usuário (Access Time), pois elas permitem ao perito colocar em ordem cronológica os acontecimentos que culminaram no ato criminoso.

Também prestou-se atenção ao campo de cabeçalhos HTTP, pois nestes podem encontrar-se informações valiosas como:

a) nome do usuário logado no SO no momento de acesso ao site;

b) nome do servidor que enviou o cabeçalho HTTP: No caso de um redirecionamento automático, esta informação é importante;

c) tipo de conteúdo acessado: Texto, imagem, áudio, entre outros;

d) tamanho do arquivo acessado;

e) Entity Tags (ETags)2: São importantes porque, com as atualizações constantes de recursos na web, permitem demonstrar que um recurso existiu em determinada data, num servidor web específico.

Por fim, foram encontrados 6 resultados referenciando o site, como ilustra a

[2] ETags são valores únicos atribuídos por servidores web a recursos disponíveis na Internet. Sempre que o recurso for atualizado, um novo valor será atribuído (W3C, 2009).

Figura 21.

Cache File: index.dat			
TYPE	URL	MODIFIED TIM	ACCESS TIME
URL	http://www.hltlda.com/templates/template/imag	Wed Oct 7 13	Thu Nov 11 2
URL	http://www.hltlda.com/themes/xp/images/menu	Fri Jul 24 16:5	Thu Nov 11 2
URL	http://webmail.unesc.net/horde/imp/mailbox.php?nocache=24		Sat Oct 2 11:
URL	http://webmail.unesc.net/horde/services/download/?module=i		Sat Oct 2 11:
URL	http://webmail.unesc.net/horde/services/download/?module=i		Sat Oct 2 11:
URL	http://www.hltlda.com/uxp/w4/m3/pr16/commc	Fri May 14 15:	Thu Nov 11 21
REDR	http://www.hltlda.com/avatar.php?gravatar_id=9d54f703f6e1a98c1a57e1f3fe		
URL	http://www.hltlda.com/images/mAtivo_left.gif	Mon Jul 6 13:	Thu Nov 11 21
URL	http://www.hltlda.com/modules/members/image	Fri Jul 24 16:5	Thu Nov 11 2
URL	http://ead.unesc.net/ava/workspace/treeview/in	Fri Jul 24 16:5	Thu Nov 4 21

Figura 21. Arquivo de *cache* com provas periciais

Como se pode averiguar, as datas de acesso do usuário (Access Time) permanecem iguais para todas as referências, ou seja, ao visitar-se a URL uma única vez, o browser salvou seis dados distintos relacionados a ela.

No entanto, apesar de se ter identificado o acesso, não foi possível saber quem efetuou o mesmo. Para responder a tal questão, os cabeçalhos HTTP foram examinados na busca por mais informações, sendo possível verificar, como ilustra a Figura 22, que o usuário estava logado no SO sob o nome *lab8b*.

h: 8378 Content-Type: image/jpeg X-Cache: MISS from hltlda.com ~U :lab8b
ontent-Type: image/png X-Cache: MISS fro m hltlda.com ~U:lab8b
ype: text/html; charset=ISO-8859-1 X-Cache: MISS from unesc.net Content-Length:
ame="trabalho de epdemiop.pptx" Content-Length: 77892 Content-Type: application/
ame="sem_nome" Content-Length: 9522 Content-Type: application/x-msdownload X
HMServer: col0-g7 Content-Length: 7112 X-Cache: HI from hltlda.co ~U:lab8b
ating=PG
195 Content-Type: image/gif X-Cache: MIS S form hltlda. .com ~U:lab8b
Content-Type: image/png X-Cache: MISS f rom hltlda.co m ~U:lab8b

Figura 22. Cabeçalhos HTTP salvos no *cache*

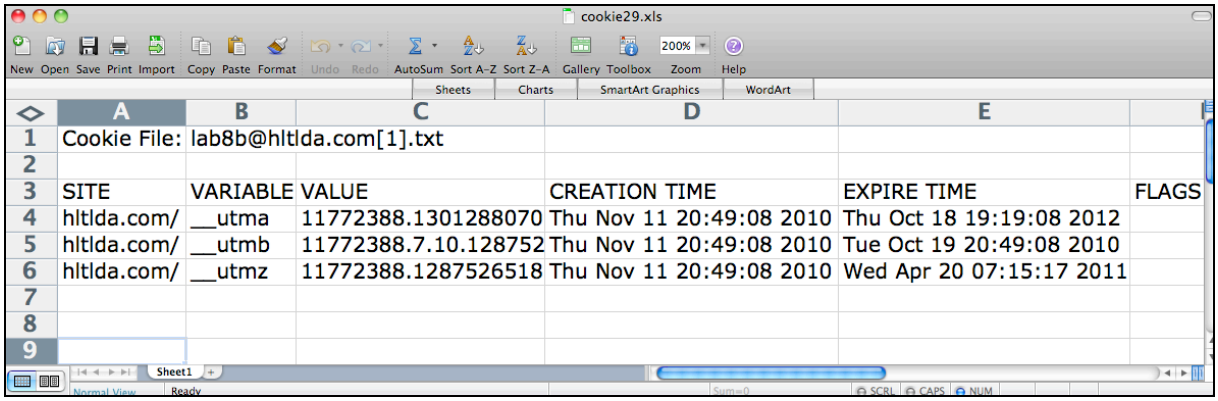
Infelizmente, pelo fato de os computadores periciados no presente trabalho, serem usados por muitos alunos, e todos logarem com contas padrão definidas pelo LabInfo da UNESC (por exemplo: lab8b), a informação angariada dos cabeçalhos HTTP, não ajudou a reduzir o número de pessoas que poderiam ter entrado no site. Foi possível, entretanto, verificar que esse tipo de informação é importante, reduzindo em alguns casos de maneira

abrupta o número de suspeitos.

Como os arquivos de *cache* contendo provas pericias, de acesso ao site por um usuário *lab8b*, no dia 11/11/10 às 20:25:17, foram encontradas no computador 5 do laboratório 8 do Bloco XXI-B, decidiu-se dar continuar a análise das evidências do mesmo, passando-se para a análise dos arquivos de *cookie*.

9.5.2 Análise de Arquivos de Cookie do IE

Nesta fase, converteram-se com o uso da ferramenta Galleta, os *cookies* do IE, como se pode observar na Figura 23.



	A	B	C	D	E	
1	Cookie File:	lab8b@hltlda.com[1].txt				
2						
3	SITE	VARIABLE	VALUE	CREATION TIME	EXPIRE TIME	FLAGS
4	hltlda.com/	__utma	11772388.1301288070	Thu Nov 11 20:49:08 2010	Thu Oct 18 19:19:08 2012	
5	hltlda.com/	__utmb	11772388.7.10.128752	Thu Nov 11 20:49:08 2010	Tue Oct 19 20:49:08 2010	
6	hltlda.com/	__utmz	11772388.1287526518	Thu Nov 11 20:49:08 2010	Wed Apr 20 07:15:17 2011	
7						
8						
9						

Figura 23. Exemplo de cookie convertido para o formato .xls

Foram obtidos então, 6 campos para análise que são respectivamente:

a) Site: diz respeito a URL do servidor que salvou o cookie;

b) Variable: nome da variável salva no mesmo;

c) Value: valor da variável, que será enviada de volta ao servidor sempre que o mesmo for acessado;

d) Creation time: data de criação;

e) Expire time: data de expiração;

f) Flags: configurações salvas pelo servidor. Por exemplo: um servidor pode configurar um cookie para ser enviado apenas por um canal seguro HTTPS,

ou impedir que ele seja acessado por scripts no lado do cliente.

Ao fim da analise dos arquivos, e como pode ser observado na figura acima, verificou-se que, ao visitar a URL *http://www.hltlda.com/* que no presente estudo de caso identifica o sistema afetado, o suspeito salvou no computador um *cookie* com as variáveis: *_utma*, *_utmb* e *_utmz*, respectivamente. Tais variáveis são usadas pelo sistema de monitoramento de sites, Google Analytics, para acompanhar os sites que o usam. A variável *_utma* verifica o numero de visitas do usuário, a *_utmb* verifica o tempo que o usuário fica no site, e por fim a *_utmz* verifica de onde o usuário partiu para acessá-lo (mecanismo de busca, link, acesso direto a URL, entre outros meios).

Atualmente esse tipo de *cookie* é muito recorrente, sendo que muitos dos analisados durante a pesquisa apresentaram essa mesma estrutura de variáveis. Infelizmente para os propósitos do presente trabalho, os arquivos coletados não ajudaram muito na redução do número de suspeitos, pois falharam em fornecer mais informações, como: nomes de usuário, endereços, entre outras, que são interessantes quando se conduz uma perícia forense.

Não obstante, todos devem sempre ser verificados, pois sabe-se que alguns sites salvam informações importantes e que podem ajudar na solução de um caso. Por exemplo, um dos arquivos analisados e mostrado abaixo na Figura 24, apresenta a variável *USER_ID*. Como o nome sugere, supõe-se que a mesma guarde como seu valor, um identificador único usado pelo servidor para reconhecer o usuário que está acessando. Se esse mesmo usuário tivesse cometido algum ato contra o servidor em questão, seria possível usando o *cookie*, identificá-lo, sendo necessário apenas descriptografar o valor salvo.

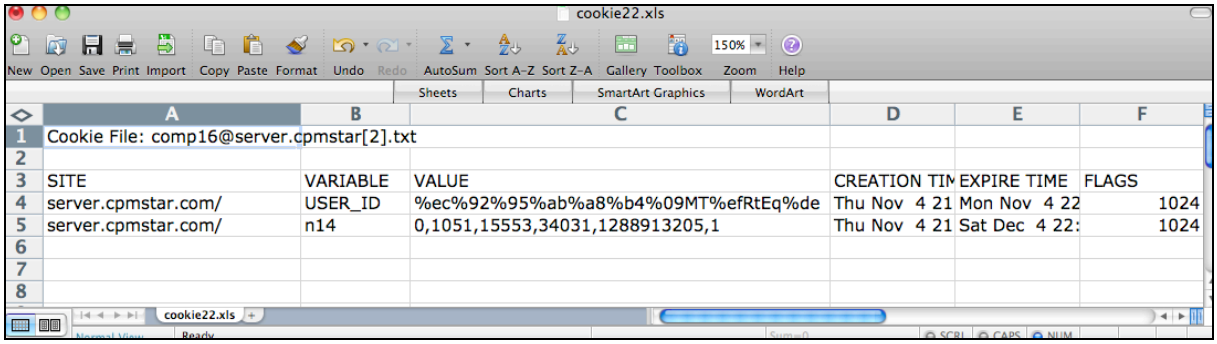


Figura 24. Exemplo de cookie com variável a ser analisada

Por fim, é importante mencionar que o nome de um *cookie* do IE, é uma combinação do nome de usuário de quem está logado no SO, no momento em que o mesmo é salvo, e do nome do servidor que o salvou no computador. Logo, ainda que o perito não encontre muitas informações relevantes em determinado arquivo, ele terá pelo menos, o nome de usuário de quem usava o sistema operacional quando o mesmo foi salvo no computador.

Por se ter mostrado pouco proveitosa a análise dos *cookies* do IE, confirmando apenas que realmente ocorreu um acesso ao servidor comprometido, passou-se para a análise dos históricos de navegação ainda no computador 5.

9.5.3 Análise do Histórico de Navegação do IE

Com a ferramenta Web Historian deu-se continuidade a pesquisa, exportando os arquivos de histórico de navegação para o formato *.xls*, mais fácil de ser analisado, como ilustra a Figura 25.

Mandiant: Web Historian - 1 - C:\Documents and Settings\comp8\Desktop\Evidencias\Evidencias_IE\Historico Navegacao\Lab8_XXI-A\Comp1\History.IE5\MSHist012010110120101108\index.dat

URL Address	Modified Time	Accessed Time	Type	Deleted	Cached Files
:2010110120101108: comp8@:Host: r1rk9np7bpcsfoeekl0khkd2juj27q3o-a-fc-opensocial.googleusercontent.com	04.11.10 17:21	09.11.10 18:24	URL	FALSE	
:2010110120101108: comp8@http://blog.globalcode.com.br/2010/04/o-que-e-logica-de-programacao.html	04.11.10 17:21	09.11.10 18:24	URL	FALSE	
:2010110120101108: comp8@:Host: blog.globalcode.com.br	04.11.10 17:21	09.11.10 18:24	URL	FALSE	
:2010110120101108: comp8@http://ead.unesc.net/ava/modules/material/actions.php?identifier=0l33l0l0l19016l0	04.11.10 19:22	09.11.10 18:24	URL	FALSE	
:2010110120101108: comp8@http://www.jogosonline.com.br/Luta/Jogo-274/Practice-Chapped	04.11.10 19:39	09.11.10 18:24	URL	FALSE	
:2010110120101108: comp8@http://ead.unesc.net/ava/modules/material/upload.php?identifier=0l33l0l0l19016l0	04.11.10 19:21	09.11.10 18:24	URL	FALSE	
:2010110120101108: comp8@:Host: www.ebah.com.br	04.11.10 17:46	09.11.10 18:24	URL	FALSE	
:2010110120101108: comp8@http://ead.unesc.net/ava/modules/material/upload.php?identifier=0l33l0l0l19016l0&pagina_retorno=	04.11.10 19:21	09.11.10 18:24	URL	FALSE	
:2010110120101108: comp8@file:///C:/Documents%20and%20Settings/comp8/Desktop/asds.docx	04.11.10 19:00	09.11.10 18:24	URL	FALSE	
:2010110120101108: comp8@:Host: ead.unesc.net	04.11.10 19:20	09.11.10 18:24	URL	FALSE	
:2010110120101108: comp8@:Host: www.din.uem.br	04.11.10 18:56	09.11.10 18:24	URL	FALSE	
:2010110120101108: comp8@http://www.hltlda.com/	04.11.10 19:35	09.11.10 18:24	URL	FALSE	setasAcontece[1].png
:2010110120101108: comp8@http://www.hltlda.com/index.php?option=com_content&view=article&id=19&Itemid=53	04.11.10 19:40	09.11.10 18:24	URL	FALSE	arrow_right[1].gif

Figura 25. Exemplo de arquivo de histórico de navegação

O arquivo obtido é formado por 7 campos, nomeadamente:

a) URL Address: endereço visitado pelo usuário;

b) Modified Time: data da última modificação que o site sofreu;

c) Accessed Time: data em que o usuário acessou;

d) Type: identifica se o usuário visitou deliberadamente a URL ou se houve um redirecionamento;

e) Deleted: mostra se a URL foi apagada pelo usuário;

f) Cached Files: indica se a URL acessada salvou algum arquivo em cache;

g) HTTP Headers: cabeçalhos HTTP, salvos ao se acessar a URL.

A análise do histórico normalmente é a mais extensiva, visto que é elevado o número de evidências coletadas. Por exemplo, cada arquivo isolado pode conter mais de 500 URLs acessadas para verificação. Contudo, por outro lado, neste tipo de análise existem mais chances de se encontrar provas ou pistas que ajudem a resolver o caso em mãos.

No presente estudo de caso, tanto a URL usada para a simulação do sistema comprometido, como as URLs correspondentes às buscas realizadas no site, foram encontradas com sucesso ao analisar-se os arquivos, conforme ilustrado na Figura 26. Pode-se observar também que mais uma vez o nome do usuário logado no SO (*lab8b*) é disponibilizado.

Mandiant: Web Historian - 1 - C:\Documents and Settings\lab8b\Desktop\Evidencias\Evidencias_IE\Historico Navegacao\Lab8_XXI-A\Comp1\History.IE		
URL Address	**Modified Time**	**Accessed Time**
Visited: lab8b@http://www.hltlda.com/	04.11.10 21:40	11.11.10 20:46
Visited: lab8b@http://www.vtunnel.com/index.php/1010110A/5b85fbc8507b3cb463b66c85ff6eba6ec433907ab1a6b6f8293d301053	21.10.10 20:01	11.11.10 20:47
Visited: lab8b@http://www.hltlda.com/index.php?option=com_content&view=article&id=19&Itemid=53	04.11.10 19:21	11.11.10 20:48
Visited: lab8b@file:///E:/VIDEO_TS/VIDEO_TS.VOB	26.10.10 21:32	11.11.10 20:48
Visited: lab8b@file:///E:/VIDEO_TS/VTS_02_1.VOB	26.10.10 21:34	11.11.10 20:49
Visited: lab8b@file:///J:/ARQ/UNIDADE%20CENTRAL%20DE%20PROCESSAMENTO.ppt	28.10.10 19:33	11.11.10 20:50
Visited: lab8b@http://www.hltlda.com/index.php?option=com_content&view=article&id=82&Itemid=58	04.11.10 21:35	11.11.10 20:50
Visited: lab8b@http://www.hltlda.com/index.php?option=com_content&view=article&id=20&Itemid=55	04.11.10 21:22	11.11.10 20:56
Visited: [illegible] opensocial.googleusercontent.com/gadgets/ifr?url=http%3A%2F%2Fgadgetsforblogger.googlecode.com%2Ffiles%2Fpicas	04.11.10 19:21	11.11.10 20:58
Visited: lab8b@http://www.hltlda.com/index.php?option=com_content&view=article&id=26&Itemid=64	21.10.10 20:01	11.11.10 21:05
Visited: lab8b@http://www.hltlda.com/index.php?option=com_contact&view=contact&id=1&Itemid=57	21.10.10 21:29	11.11.10 21:13
Visited: lab8b@http://ead.unesc.net/ava/modules/modules/material_list/index.php?identifier=15519l35l597l24645l19016l0	04.11.10 21:22	11.11.10 21:22

Figura 26. URLs relevantes do histórico de navegação

Importa salientar que a análise do histórico, permite que se contextualize as ações do usuário do browser de maneira mais rápida que as outras formas mencionadas no decorrer do trabalho. Ela mostra os acessos realizados pelo mesmo numa sequência linear formando uma linha do tempo, onde é possível verificar passo à passo cada ação tomada pelo mesmo. Essa mesma percepção ocorre de maneira mais lenta ao analisar-se arquivos de *cookie*, que não são salvos de maneira cronológica, tendo o perito o trabalho de encaixá-los na ordem correta, e de *cache* que mostram em sua maioria as URLs das imagens e outros arquivos que foram salvos, não mostrando URLs mais relevantes como buscas, por exemplo.

Por fim, verificou-se que ao usar a ferramenta Web Historian no presente estudo, conseguiram-se recuperar poucos arquivos de *cache* e cabeçalhos HTTP que ajudariam o perito a melhor identificar quem realizou os acessos. Tais campos nos arquivos *.xls* exportados ficaram quase sempre vazios. No entanto, por conter URLs mais reveladoras como buscas realizadas, e páginas de login, entre outras, conseguiu-se verificar, pela análise do histórico, que o usuário do computador 5 do laboratório 8, efetuou um login no servidor afetado no dia 11/11/10 às 20:46:12.

Deu-se prosseguimento ao estudo, analisando-se os arquivos *.sqlite* do Firefox, localizados nesse mesmo computador.

9.5.4 Análise do Histórico de Navegação do Firefox

Iniciou-se então, a conversão das evidências coletadas do Firefox, usando a ferramenta F3E.

Como já foi mencionado ao longo do trabalho o Firefox usa seis arquivos principais para salvar as informações do usuário:

a) places.sqlite: para informações do historio de navegação;

b) CACHE_MAP: que contém as informações de cache;

c) downloads.sqlite: para os arquivos baixados;

d) formhistory.sqlite: para campos preenchidos em formulários;

e) cookies.sqlite: para informações de cookies, e finalmente;

f) signons.sqlite: que contém senhas e nomes de usuário encriptados.

Cada um destes arquivos implementa um banco de dados transacional, e a ferramenta F3E exporta cada tabela encontrada, com os seus respectivos campos e valores, para um arquivo *.csv* facilmente legível.

O primeiro arquivo, *places.sqlite*, é formado por 10 tabelas inter-relacionadas como pode ser observado na Figura 27. Elas são respectivamente:

a) moz_places: Sites visitados;

b) moz_anno_attributes: Nomes e códigos de todas as anotações feitas pelo usuário;

c) moz_annos: Anotações que o usuário possa ter feito sobre sites acessados;

d) moz_bookmarks: Sites salvos como favoritos;

e) moz_favicons: Ícones favoritos, incluindo a URL do ícone;

f) moz_historyvisits: Histórico do número de vezes que um site foi visitado;

g) moz_inputhistory: Histórico das URLs digitadas pelo usuário;

h) moz_items_annos: Anotações de cada site salvo como favorito;

i) mos_keywords: Palavras chaves usadas para identificar os sites favoritos;

j) moz_bookmarks_roots: Pastas onde estão salvos os sites favoritos.

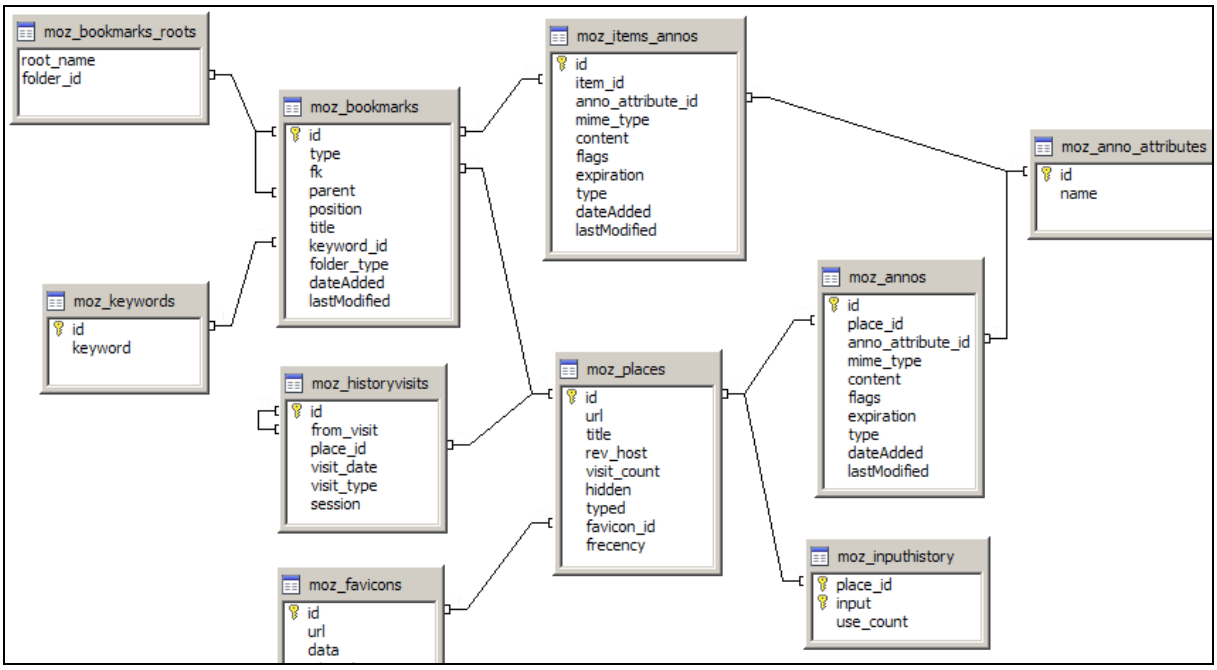


Figura 27. Diagrama mostrando as tabelas do arquivo *places.sqlite*

É importante citar que a tabela moz_places é uma das mais importantes a ser analisada, visto que é nela que são salvas as URLs acessadas. Como se pode verificar novamente na Figura 27, ela é constituída por 9 campos, informando respectivamente:

a) id: Código da URL salva;

b) URL: Mostra o domínio do site acessado;

c) title: Título da página acessada;

d) rev_host: O nome do servidor da URL acessada, salvo de maneira invertida. Ex: www.google.com/search ficaria moc.elgoog.www;

e) visit_Count: Número de visitas a determinada URL;

f) hidden: Identifica URLs que não foram especificamente digitadas pelo usuário como: i-frames, links RSS, chamadas de funções javascript, entre outras (0 significa que a URL não é hidden e 1 que sim);

g) typed: Marca as URLs digitadas diretamente na barra de endereços do Firefox (0 significa que não foi digitada na barra, e 1 que sim);

h) favicon_Id: Código do ícone salvo pelo Firefox correspondente a URL acessada;

i) frecency: Uma combinação de frequência (número de vezes que determinada URL foi acessada em determinado intervalo de tempo) e recência (a última data em que a URL foi acessada dentro de determinado periodo).

A figura 28, ilustra como os acessos ao site do sistema fictício (*http://hltlda.com/*) desta vez no Firefox, foram encontrados com sucesso ao analisar-se a tabela *moz_places*.

263	http://www.revistaplatina.com/estilo-de-vida/custom-modules
264	http://hltlda.com/index.php
265	http://hltlda.com/index.php?searchword=sobre+nFigura+27.+URLs+relevantes+do+hist%C3%B3rico+de+navega%C3%A7%C3%A3os&ordering=&
266	http://www.revistaplatina.com/component/content/article/39-rokfeature/6736-entrevista-exclusiva-yannuza
267	http://hltlda.com/index.php?searchword=como+invadir+um+servidor+web&ordering=&searchphrase=all&Itemid=89&option=com_search
268	http://hltlda.com/index.php?searchword=sobre+ns&ordering=&searchphrase=all&Itemid=89&option=com_search
269	http://hltlda.com/index.php?option=com_content&view=article&id=19:joomla-overview&catid=40:dados-hlt&Itemid=53
270	http://hltlda.com/index.php?option=com_content&view=article&id=21&Itemid=59
271	http://hltlda.com/index.php?option=com_contact&view=contact&id=1&Itemid=57
272	http://hltlda.com/index.php?option=com_user&view=register
273	http://hltlda.com/index.php?option=com_user#content
274	http://hltlda.com/index.php?option=com_contact&view=contact&id=1%3Acontact-us&catid=12%3Acontacts&Itemid=57

Figura 28. Arquivo *csv* da tabela *moz_places*

Outras tabelas importantes e que foram analisadas foram: *moz_bookmarks*, *moz_favicons* e *moz_inputhistory*, pois poderiam conter informações importantes que auxiliassem na investigação. Das três, apenas a *moz_favicons* apresentou evidências mostrando que a URL do sistema havia sido acessada.

Identificou-se com esta análise a ocorrência de buscas objetivando ganhar conhecimento sobre como invadir um servidor, bem como um novo login no servidor ainda no mesmo dia 11/11/10, mas desta vez às 21:33:10. Não se conseguiu apurar, contudo, se as buscas e o efetivo acesso ao servidor em questão foram realizadas pela mesma pessoa. Assim, decidiu-se analisar mais a fundo os arquivos do Firefox, passando-se para a análise dos arquivos de *cache*.

9.5.5 Análise dos Arquivos de Cache do Firefox

Verificou-se que a ferramenta F3E falha ao converter de maneira apropriada este tipo de arquivo. Como se pode observar na Figura 29, as informações conseguidas não são legíveis, e por esse motivo os arquivos de *cache* coletados, foram examinados usando-se a ferramenta Mozilla Cache View.

id	domain	partial_data	complete_data	chunk_id	table_id
12	TTdû			25041	3
18	¸Ò			25041	3
19	œ≈É			25041	3
21	´8'q			25041	3
23	Í·v			25041	3
26	´‡^ª			25041	3
28	º»¬˜			25041	3
33	;X≥È			25041	3
36	Σ•äõ	-9T0		25041	3
37	º‰ú			25041	3
40	W»S$			25041	3
41	`ËÚ˜			25041	3
42	vÆ≤ó			25041	3
45	<:ô,",",25041,3, 49,A				

Figura 29. Arquivo *csv* de *cache* do Firefox gerado pela ferramenta F3E

Obtiveram-se assim, 8 campos para análise, que são respectivamente:

a) URL: Identifica a URL do arquivo salvo;

b) content type: Tipo de arquivo (imagem, áudio, vídeo, entre outros);

c) file size: Tamanho do arquivo;

d) last modified: Data da última alteração que o arquivo sofreu;

e) last fetched Time: Data da última vez que o arquivo foi carregado para visualização;

f) expiration time: Data de expiração;

g) fetch count: Número de visualizações;

h) server name: Nome do servidor de onde o arquivo foi baixado.

A Figura 30, ilustra o arquivo em análise com a nova ferramenta.

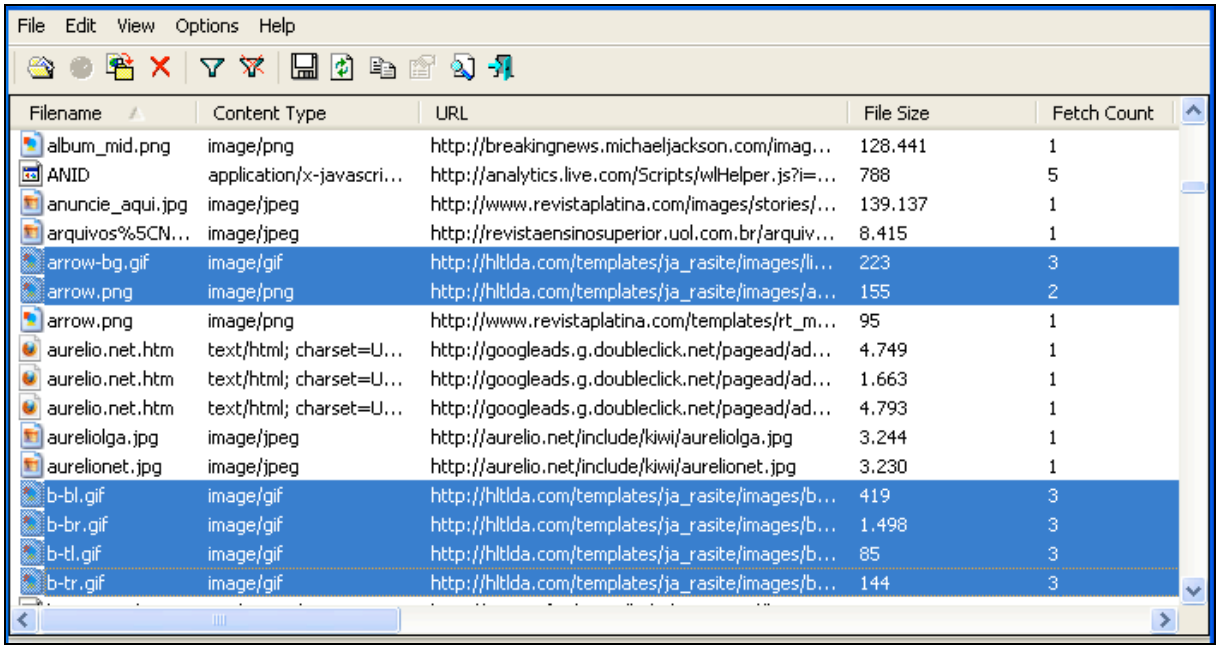


Figura 30. Arquivo *csv* de *cache* do Firefox gerado pela ferramenta Mozilla Cache View

Com ela foi possível comprovar os acessos ao servidor afetado, mas não se obtiveram novas informações ou pistas relevantes. Optou-se por analisar a seguir os downloads realizados pelo usuário da máquina.

9.5.6 Análise do Histórico de Downloads do Firefox

A verificação dos arquivos *downloads.sqlite* é importante, uma vez que, apurar que tipos de documentos o usuário baixou para o computador pode revelar novas pistas ou informações pertinentes ao caso. O arquivo é formado por apenas uma tabela, *moz_downloads* com quinze campos para análise:

a) id: Código do arquivo salvo no computador;

b) name: Nome do arquivo;

c) source: URL do arquivo na Internet;

d) target: Caminho onde foi salvo o arquivo no computador;

e) tempPath: Caminho do arquivo temporário sendo usado para realizar o download;

f) startTime: Data e horário de começo do download;

g) endTime: Data e horário de fim do download;

h) state: Estado atual do download (-1/Não iniciado, 0/Em execução, 1/Completado, 2/Falhou devido a erro, 3/Cancelado pelo usuário, 4/Pausado, 5/Em lista de espera, 6/Bloqueado, 7/Sendo escaneado para detecção de vírus, 8/Vírus detectado, 9/Bloqueado pelo SO,);

i) referrer: URL do site referenciando o Download;

j) entityID: Código do gerenciador de conexão usado para abrir o canal de download. Tal gerenciador é usado para resumir downloads que foram pausados;

k) currBytes: número atual de bytes salvos no computador;

l) maxBytes: número total de bytes a serem salvos;

m) mimeType: tipo de arquivo sendo salvo;

n) preferredAplication: aplicação predefinida para abrir o arquivo depois de salvo;

o) preferredAction: ação a realizar depois de terminado o download;

p) autoResume: Define se o download deve-se auto-resumir depois de uma pausa (0 para não , 1 para sim).

Na figura 31, pode-se observar a estrutura de um dos arquivos de downloads analisado.

id	name	source
1	pesquisa.docx	http://65.55.155.121/att/GetAttachment.aspx?file=e367114b-a950-4b71-a881-002526f28297.docx&ct=YXBwbGljYXRpb24vdm5kLm9wZW54bWxmb3JtYXRzLW9mZmljZWRvY3VtZW50Lr
2	mostrando acara fichas certa.doc	http://65.55.130.121/att/GetAttachment.aspx?file=1436a24a-9e29-4676-b532-66644d1394c9.doc&ct=YX
3	Aula2_Exerccio.doc	http://ead.unesc.net/ava/modules/file2/viewfile.php?id=4658l34l862l5718l13175l0l9183l5580l5
4	Aula2_Matriz.doc	http://ead.unesc.net/ava/modules/file2/viewfile.php?id=4659l34l862l5719l13175l0l9182l5581l5
5	Dica02-Img06.jpg	http://www.dicas-l.com.br/imagens/Dica02-Img06.jpg
6	euu.....laura.jpg	http://www.dicas-l.com.br/imagens/Dica02-Img06.jpg
7	11111111.jpg	http://www.dicas-l.com.br/imagens/Dica02-Img01.jpg
8	22222222222222222222222.jpg	http://www.dicas-l.com.br/imagens/Dica02-Img02.jpg
9	LegislaoAmbiental.ppt	http://ead.unesc.net/ava/modules/file2/viewfile.php?id=4373l34l809l11401l5191l0l1327l2583l17
10	ecologiaedessustentavel.ppt	http://ead.unesc.net/ava/modules/file2/viewfile.php?id=4369l34l809l11397l5191l0l1321l2580l17
11	ecologiaedessustentavel.ppt	http://ead.unesc.net/ava/modules/file2/viewfile.php?id=4369l34l809l11397l5191l0l1321l2580l17

Figura 31. Arquivo *csv* de *downloads* do Firefox gerado pela ferramenta F3E

Neste tipo de análise, os principais campos a serem observados são: *referrer*, pois indica o site no qual o usuário estava navegando quando iniciou o download, *target*, porque no caso do comprometimento de um sistema por vírus, deve-se fazer uma cópia do arquivo que causou o mesmo, *mimeType*, uma vez que, apesar de um arquivo ter determinada extensão, pode tratar-se de um tipo de arquivo completamente diferente (Ex: um arquivo executável *.exe*, com extensão de imagem *.jpg*, muito comum em virus), e por último as datas de início e fim. O campo *source*, também deve receber especial atenção, pela fato de o usuário muitas vezes ser redirecionado para outros sites e salvar arquivos no computador sem ter conhecimento.

No presente caso fictício, por não terem sido usados arquivos maliciosos (como vírus, trojans, backdoors, entre outros), a análise procurou identificar arquivos que provessem pistas sobre quem poderia ter cometido o crime. Por exemplo, arquivos de tutoriais sobre como invadir um servidor, arquivos contendo informações sobre a empresa prejudicada, entre outros, poderiam indicar quem teria interesse em cometer tal crime. Também deve-se prestar atenção aos nomes das pastas onde os arquivos foram salvos, pois muitos usuários nomeiam as pastas, ou dispositivos de armazenamento como Pen-Drives com o seu nome. Se um

arquivo *.pdf* recuperado pelo perito e contento informações da empresa foi salvo no caminho: *file:///D:/Nome_do_Individuo/projetos/invasão.pdf*, ao analisar o histórico de downloads conseguir-se-ia o nome de um indivíduo a acrescentar a lista de suspeitos.

No entanto, não se conseguiu nenhum tipo de informação que ajudasse a solucionar o caso em mãos, dando-se assim continuidade a pesquisa.

9.5.7 Análise do Histórico de Formulários Preenchidos do Firefox

O arquivo *formhistory.sqlite* foi o próximo a ser analisado, sendo constituído por apenas uma tabela *moz_formhistory*, e oferecendo 6 campos para análise:

a) id: Código da palavra ou frase digitada;

b) fieldname: Nome do campo que recebeu a palavra ou frase;

c) value: Valor, ou seja, a frase ou palavra digitada propriamente dita;

d) timesUsed: Número de vezes que a palavra ou frase foi usada;

e) firstUsed: Data em que foi usada pela primeira vez;

f) lastUsed: Data em que foi usada pela última vez.

Esta análise é crucial pois permite obter algumas das informações digitadas pelo usuário no browser, seja preenchendo formulários, enviando emails, ou realizando buscas, e contextualizar as mesmas para adquirir pistas sobre o crime cometido.

Por exemplo, como pode-se conferir examinando a figura 32, um dos arquivos *formhistory.sqlite* analisados no decorrer do estudo, salvou com sucesso as buscas realizadas no site do sistema comprometido, bem como algumas das informações disponibilizadas durante o cadastramento no mesmo. Para preservar a identidade dos acadêmicos da universidade, verifica-se na imagem abaixo, e noutras do corrente estudo, que alguns nomes, emails e outros dados que pudessem comprometer a integridade dos mesmos, foram

ocultados.

id	fieldname	value	timesUsed	firstUsed	lastUsed
1	q	reparar erros windows	1	1285617917859000	1285617917859000
2	q	download framedyn.dll	1	1285620417924000	1285620417924000
3	q	cygwin	1	1286308673741000	1286308673741000
4	q	download internet explorer	1	1287326643238000	1287326643238000
5	q	internet explorer nao aparece	1	1288535395565000	1288535395565000
6	q	cmd dir	1	1288883535143000	1288883535143000
7	q	wbf santander	1	1289389618996000	1289389618996000
8	searchword	sobre nFigura 27. URLs relevantes do hist√≥rico de navega√ß√£o	2	1289615427448000	1289615429101000
9	searchword	como invadir um servidor web	2	1289615452494000	1289615542103000
10	searchword	sobre ns	1	1289615552658000	1289615552658000
11	name	[redacted]	3	1289615682705000	1289615978781000
12	username	[redacted]	2	1289615682705000	1289615737745000
13	email	[redacted]@gmail.com	3	1289615682705000	1289615978791000
14	subject	Estudo de Caso	1	1289615978791000	1289615978791000
15	username	[redacted]@gmail.com	2	1289615994103000	1289616012400000

Figura 32. Arquivo *csv* de histórico de formulário do Firefox

É importante ressaltar que, analisando os nomes dos campos nos quais o usuário digitou, é possível deduzir que tipo de ação o mesmo realizava (busca, envio de email, entre outras), adquirindo, em alguns casos, pistas sobre as suas intenções ao realizar tal ato. Exemplificando, no caso ilustrado acima, deduziu-se que os campos cujo nome é “q” foram de buscas realizadas no Google, pois sabe-se que o campo de busca neste sistema apresenta tal nome, já os itens com nomes como “searchword” poderiam ter sido de campos de busca em sites acessados pelo usuário, e variáveis como “name”, “username”, “email” e “subject” são auto-explicativas dando informações importantes sobre quem acessava o computador em dado momento e que ação o mesmo realizava.

Como as informações salvas são organizadas segundo a data de criação, formou-se novamente uma linha do tempo, sendo possível observar passo à passo cada ação realizada pelo usuário. Torna-se deste modo crucial, que o perito decodifique o PRTime (formato de data usado pelo Firefox), pois só assim ele saberá os intervalos de tempo entre uma ação e outra, permitindo muitas vezes saber se a mesma pessoa realizou todas as ações identificadas ou não.

Ao decodificar o PRTime no presente estudo de caso, como ilustra a Figura 33, apurou-se que a busca sobre “como invadir um servidor web” foi realizada no dia 11/11/10 às 21:31:40 e *fulano_de_tal@gmail.com* foi digitado no campo *email* de algum formulário no mesmo dia às 21:35:10, ou seja, pelo curto espaço de tempo existente entre uma ação e outra, é muito provável que a pessoa que realizou a busca, e a que enviou o email sejam uma só, ou que ambas estivessem juntas usando o computador. Tem-se aqui, pela primeira vez, a chance de estreitar a investigação, focando na pessoa cuja as informações (nome e email) foram encontradas.

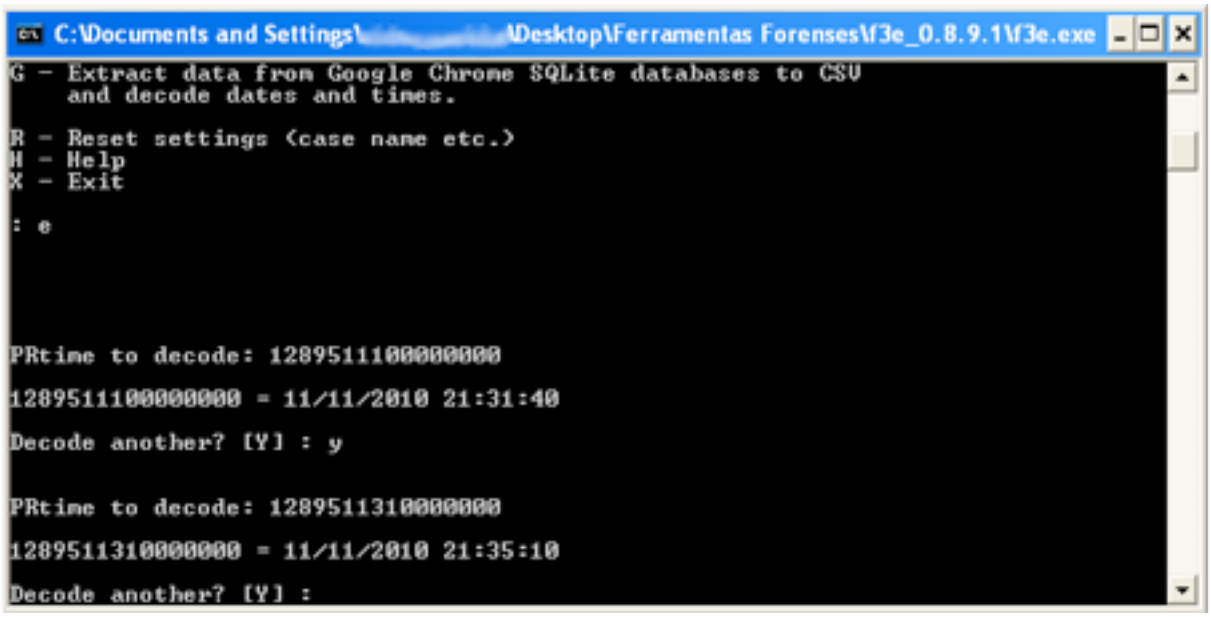


Figura 33. Decodificação do PRTime

Ao adquirir uma informação como a exemplificada acima, deve-se então procurar por mais pistas que comprovem a culpabilidade ou inocentem o suspeito. Informações que comprovem onde o mesmo estava e com estava no momento identificado pelo PRTime podem ajudar a solucionar o caso.

Também é importante verificar que ao decodificar o PRTime foi possível depois analisar no histórico de navegação do *browser*, que páginas estavam sendo acessadas nos horários apurados. Ao comparar os PRTimes do exemplo acima, com as entradas do histórico de navegação verificou-se que o usuário *Fulano_de_tal* realizou buscas no Google,

posteriormente efetuou buscas no site do sistema fictício do atual estudo de caso, enviou um email usando o formulário de contato do site, e por fim efetivou um login no mesmo.

9.5.8 Análise dos Arquivos de Cookie do Firefox

Continuando-se a investigação, passou-se para a verificação dos arquivos de *cookie*. Eles são formados por uma tabela *moz_cookies*, e originaram 8 campos a serem analisados que são, respectivamente:

a) id: Código do cookie;

b) name: Nome da variável salva pelo mesmo;

c) value: Valor da variável;

d) host: URL do servidor que o salvou;

e) path: Caminho no servidor onde o mesmo está armazenado;

f) expiry: Data de expiração;

g) lastAccessed: Data do último acesso;

h) isSecure: Flag controlando se o cookie é enviado por um canal seguro ou não (0 significa que não, e 1 que sim);

i) isHttpOnly: Flag controlando se o cookie pode ser acessado por scripts no lado do cliente (0 significa que não, e 1 que sim).

Ao realizar-se a análise dos mesmos, foram encontrados apenas dois com relação ao site em estudo. Um deles apresentando uma estrutura similar ao *cookie* encontrado anteriormente ao analisar-se os arquivos do IE, contendo as variáveis: *_utma*, *_utmb*, *_utmz*, e outro com uma variável de nome *ja_rasite_tp* e cujo valor era *ja_rasite*. Este último, foi salvo no dia 11/11/10 às 21:36:46 e uma averiguação demonstrou que a variável em questão é usada para enviar ao servidor as configurações do usuário com relação ao layout do site do

sistema. Logo, neste caso, o *cookie* serviu mais uma vez para confirmar que o site foi acessado, e permitiu adicionar na ordem cronológica dos eventos que até as 21:36:46 o suspeito ainda estava usando o site.

A Figura 34, ilustrando o *cookie* encontrado, pode ser visualizada abaixo.

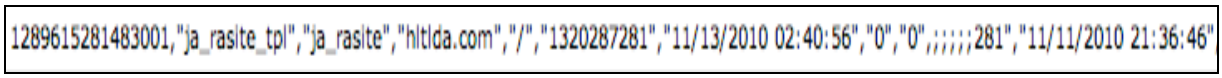


Figura 34. Arquivo de *cookie* do Firefox referenciando o sistema comprometido

9.5.9 Análise das Senhas e Nomes de Usuário do Firefox

Por último, passou-se para a análise dos arquivos *signons.sqlite* onde estão armazenados os nomes de usuário e as senhas salvas pelo Firefox. Tal análise é relevante, pois possibilita verificar que usuários têm privilégios de logar no servidor comprometido, obtendo os nomes de usuário e senhas dos mesmos.

Para o presente estudo de caso, como já foi mencionado antes, foi efetuado um cadastro no site que supostamente foi afetado. Objetivando simular a ação de um criminoso, que teve de se logar no servidor, para puder então danificá-lo.

No entanto, a ferramenta F3E não foi capaz de exportar as informações dos arquivos *signons.sqlite*, ainda que encriptadas, para o formato *.csv*. Assim sendo, para visualização de tais informações usou-se a ferramenta PasswordFox, cuja tela é possível visualizar na Figura 35, abaixo.

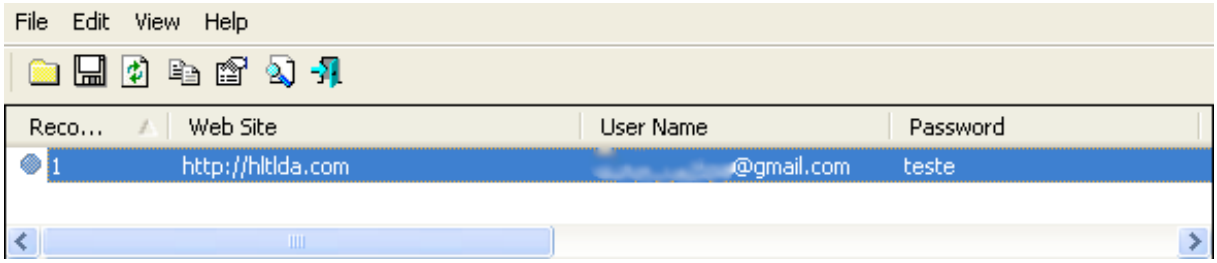
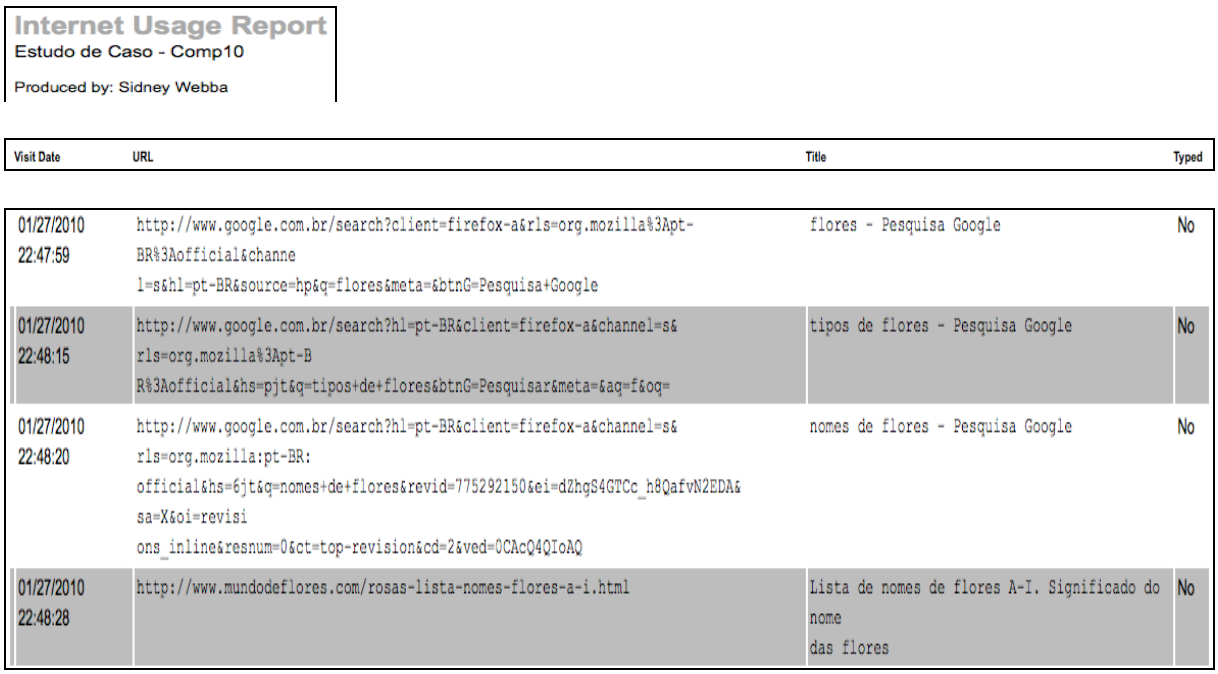


Figura 35. Tela da ferramenta PasswordFox recuperando senhas do Firefox

Observa-se então, que caso o usuário não tenha o cuidado de deletar os nomes de usuário e senhas salvas, as mesmas podem facilmente ser visualizadas. No presente trabalho, conseguiu-se com sucesso recuperar o nome de usuário e senha usada para logar no site do sistema prejudicado. Mas, infelizmente, o programa não forneceu informações como: data em que os nomes de usuário e senhas foram salvas, ou datas do último login efetuado usando tal senha e tal nome de usuário, impossibilitando assim verificar de maneira irrefutável que esses dados foram salvos pelo *browser* na mesma data e horário da tentativa de invasão do servidor.

Outra possibilidade para aquisição das informações dos arquivos *signon.sqlite*, seria o uso de um gerenciador de banco de dados Sqlite, e buscar por elas usando a linguagem STRUCTURED QUERY LANGUAGE (SQL). Ainda assim, as informações coletadas teriam de ser descriptografadas, para que pudessem ser corretamente visualizadas.

Terminando a etapa 4, é importante mencionar que a ferramenta F3E possibilita que se gerem relatórios dos históricos de navegação dos computadores periciados, nos formatos *.csv* e *.html*. Um exemplo de como seria um desses relatórios é mostrado abaixo na Figura 36.



Internet Usage Report
Estudo de Caso - Comp10
Produced by: Sidney Webba

Visit Date	URL	Title	Typed
01/27/2010 22:47:59	http://www.google.com.br/search?client=firefox-a&rls=org.mozilla%3Apt-BR%3Aofficial&channe l=s&hl=pt-BR&source=hp&q=flores&meta=&btnG=Pesquisa+Google	flores - Pesquisa Google	No
01/27/2010 22:48:15	http://www.google.com.br/search?hl=pt-BR&client=firefox-a&channel=s& rls=org.mozilla%3Apt-B R%3Aofficial&hs=pjt&q=tipos+de+flores&btnG=Pesquisar&meta=&aq=f&oq=	tipos de flores - Pesquisa Google	No
01/27/2010 22:48:20	http://www.google.com.br/search?hl=pt-BR&client=firefox-a&channel=s& rls=org.mozilla:pt-BR: official&hs=6jt&q=nomes+de+flores&revid=775292150&ei=dZhgS4GTCc_h8QafvN2EDA& sa=X&oi=revisi ons_inline&resnum=0&ct=top-revision&cd=2&ved=0CAcQ4QIoAQ	nomes de flores - Pesquisa Google	No
01/27/2010 22:48:28	http://www.mundodeflores.com/rosas-lista-nomes-flores-a-i.html	Lista de nomes de flores A-I. Significado do nome das flores	No

Figura 36. Exemplo de relatório do histórico de navegação gerado pela ferramenta F3E

Pode-se observar que as informações exportadas para o relatório são básicas, sendo constituídas de:

a) data de acesso ao site;

b) URL do site;

c) título da página acessada, e;

d) informa se a URL foi apagada pelo usuário ou não.

No entanto, para que se verifique o histórico de navegação de maneira rápida e facilitada, recomenda-se a geração de relatórios no formato .html. No relatório também é possível verificar os 20 sites mais acessados pelo usuário.

9.6 ETAPA 5 – DOCUMENTAÇÃO

Esta etapa foi recorrente durante topo o processo pericial, pois ela é crucial para que se possa depois redigir um laudo pericial fiel aos fatos. A documentação completa pode ser visualizada no apêndice B do presente trabalho.

9.7 ETAPA 6 – RECONSTRUÇÃO DA CENA DO CRIME

Na atual etapa, deve-se fazer a reconstrução dos eventos, juntando-se todas as evidências para que se possa determinar o que realmente ocorreu.

Como já foi mencionado, o presente estudo de caso simula que o sistema de vendas online de uma empresa de varejo fictícia foi acessado inadequadamente, e identificou-se que tal acesso partiu de um dos computadores dos laboratórios da UNESC. Realizou-se então a condução de uma perícia forense computacional objetivando descobrir: quem?, onde?, quando? e como?, realizou tal ato.

Para que os propósitos da pesquisa se cumpram, supôs-se que o crime foi identificado pelo responsável técnico do sistema da empresa, no dia 11/11/10 às 20:46:12, e que nesse mesmo dia às 21:33:10 uma nova tentativa foi identificada.

Iniciando então a reconstrução da cena do crime, descobriu-se analisando as evidências do IE que:

a) ocorreram acessos ao site do sistema comprometido, partindo do computador 5 do laboratório 8, do Bloco XXI-B da UNESC, pois foram encontradas evidências nesse computador;
b) os acessos foram efetuados no dia 11/11/10, onde o primeiro acesso ocorreu às 20:46:12 e o último ocorreu às 21:22:45;
c) no horário em que o técnico identificou a primeira tentativa de ataque ao sistema, às 20:46:12, o usuário desse computador estava acessando o site em questão, usando o IE;
d) o usuário logado no SO estava identificado como lab8b.

Por sua vez, analisando-se as evidências coletadas do Firefox, foi possível verificar que:

a) ocorreram acessos ao site do sistema comprometido, novamente partindo do computador 5 do laboratório 8, do Bloco XXI-B da UNESC, pois foram encontradas evidências nesse computador;
b) os acessos foram efetuados também no dia 11/11/10, onde o primeiro acesso ocorreu às 21:31:15 e o último às 21:36:46;
c) uma busca sobre “URLs relevantes do histórico de navegação” foi realizada às 21:31:15;
d) outra busca sobre “como invadir um servidor web” foi realizada às 21:31:40;
e) e mais uma busca contendo os termos “sobre ns”, foi realizada às 21:33:57;

f) às 21:34:07, foi digitado o nome Fulano_de_Tal no campo name de um formulário de contato, numa das páginas do site comprometido;

g) às 21:34:17, foi digitado fulano_de_tal, no campo username desse mesmo formulário;

h) às 21:35:10, foi digitado fulano_de_tal@gmail.com, no campo email do formulário;

i) às 21:35:29, foi digitado "Estudo de caso", no campo subject do formulário;

j) às 21:36:46, foi digitado fulano_de_tal@gmail.com, no campo username de um formulário de login do site afetado;

k) no horário em que foi identificada a segunda tentativa de comprometimento do sistema, às 21:33:10, o usuário desse computador ainda estava acessando o site em questão, usando o Firefox.

Com as provas encontradas em mãos, procurou-se apurar mais pistas sobre o caso, e confirmou-se junto a instituição que no dia 11/11/10 o acadêmico esteve presente no laboratório 8 do Bloco XXI-B atendendo a uma aula, das 19:00 às 22:00. Confirmou-se também, junto aos seus colegas que no dia em questão o mesmo usou exclusivamente o computador 5 do laboratório, o mesmo onde foram encontradas as provas.

Concluiu-se ao final da pericia que:

a) Quem perpetrou o crime foi o acadêmico Fulano_de_Tal, estudante da universidade em causa;

b) Ele foi cometido a partir do computador 5, do laboratório 8 do Bloco XXI-B da universidade, e;

c) Foi efetuado no dia 11/11/10 às 20:46:12 e às 21:33:10;

d) Para executar o mesmo, o acadêmico registrou-se no site alvo, obteve acesso a página contento um formulário de contato da empresa, e tentou usando scripts

burlar os mecanismos de segurança do formulário e enviar ao servidor código malicioso capaz de danificar o mesmo.

Para documentação dos procedimentos e registro das informações sobre a perícia, foi gerado um laudo pericial contendo tais informações. O mesmo pode ser visualizado abaixo.

LAUDO PERICIAL

Perito/Examinador: Sidney Roberto da Silva Webba

Data: 12 de Novembro de 2010 **Horário:** 21:30

Descrição da Perícia: Perícia realizada nos laboratórios da Unesc

Observações: A presente pericia foi realizada como um estudo de caso para o Trabalho de Conclusão de Curso requisitado pela Universidade do Extremo Sul Catarinense, para obtenção do grau de Bacharel em Ciência da Computação.

Identificação dos Sistemas Analisados

	Lab 16, Bloco XXI-A	Lab 8, Bloco XXI-B
Nome do Computador	comp16_1	lab8b_1
Nome do Computador	comp16_2	lab8b _2
Nome do Computador	comp16_3	lab8b _3
Nome do Computador	comp16_4	lab8b _4
Nome do Computador	comp16_5	lab8b _5
Nome do Computador	comp16_6	lab8b _6
Nome do Computador	comp16_7	lab8b _7
Nome do Computador	comp16_8	lab8b _8
Nome do Computador	comp16_9	lab8b _9
Nome do Computador	comp16_10	lab8b _10

Sistema Operacional: Windows XP, Service Pack 3

Browsers: Internet Explorer versão 8.0.6; Firefox versão 3.6.12

Análise de Arquivos de Cache do Internet Explorer:

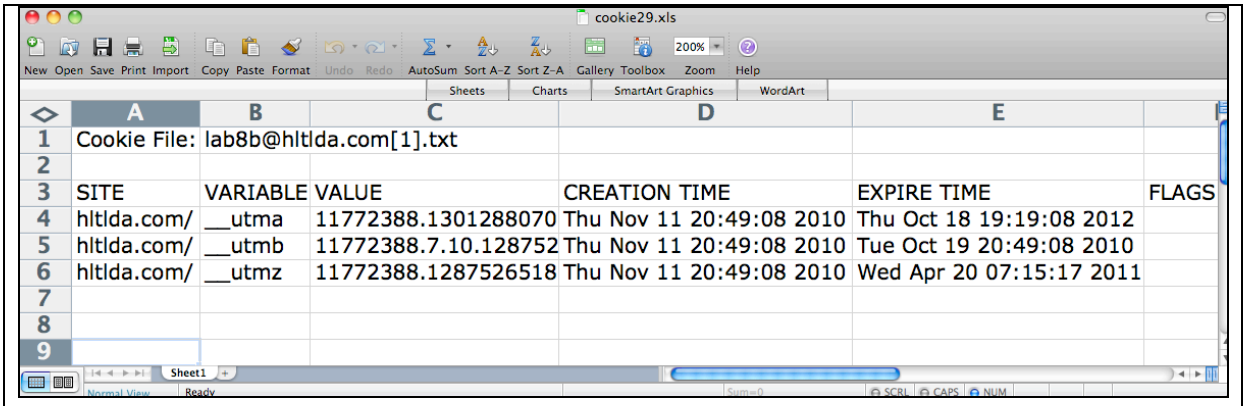
Usando-se a ferramenta Pasco, realizou-se a conversão dos arquivos para o formato xls, e usando o programa Excel efetuou-se a análise dos próprios. Provas periciais indicando o acesso ao servidor comprometido no dia 11/11/10 às 20:25:17 pelo usuário *lab8b*, foram encontradas no arquivo coletado do computador 5, de nome lab8b_5 do laboratório 8 do Bloco XXI-B, como pode ser observado abaixo.

Cache File: index.dat			
TYPE	URL	MODIFIED TIM	ACCESS TIME
URL	http://www.hltlda.com/templates/template/imag	Wed Oct 7 13	Thu Nov 11 2
URL	http://www.hltlda.com/themes/xp/images/menu	Fri Jul 24 16:5	Thu Nov 11 2
URL	http://webmail.unesc.net/horde/imp/mailbox.php?nocache=24		Sat Oct 2 11:
URL	http://webmail.unesc.net/horde/services/download/?module=i		Sat Oct 2 11:
URL	http://webmail.unesc.net/horde/services/download/?module=i		Sat Oct 2 11:
URL	http://www.hltlda.com/uxp/w4/m3/pr16/commc	Fri May 14 15:	Thu Nov 11 21
REDR	http://www.hltlda.com/avatar.php?gravatar_id=9d54f703f6e1a98c1a57e1f3fe		
URL	http://www.hltlda.com/images/mAtivo_left.gif	Mon Jul 6 13:	Thu Nov 11 21
URL	http://www.hltlda.com/modules/members/image	Fri Jul 24 16:5	Thu Nov 11 2
URL	http://ead.unesc.net/ava/workspace/treeview/in	Fri Jul 24 16:5	Thu Nov 4 21

h: 8378 Content-Type: image/jpeg X-Cache: MISS from		hltlda.com ~U	:lab8b
ontent-Type: image/png X-Cache: MISS fro	m hltlda.com	~U:lab8b	
ype: text/html; charset=ISO-8859-1 X-Cache: MISS from unesc.net Content-Length:			
ame="trabalho de epdemiop.pptx" Content-Length: 77892 Content-Type: application/			
ame="sem_nome" Content-Length: 9522 Content-Type: application/x-msdownload X			
HMServer: col0-g7 Content-Length: 7112 X-Cache: HI		from hltlda.co	~U:lab8b
ating=PG			
195 Content-Type: image/gif X-Cache: MIS	S form hltlda.	.com ~U:lab8b	
Content-Type: image/png X-Cache: MISS f	rom hltlda.co	m ~U:lab8b	

Análise de Arquivos de Cookie do Internet Explorer:

Por sua vez, com a ferramenta Galleta foram originados arquivos xls para análise, e a mesma foi efetuada usando-se o programa Excel. Provas periciais indicando o acesso ao servidor comprometido no dia 11/11/10 às 20:49:08, foram encontradas no arquivo coletado do computador 5, de nome lab8b_5 do laboratório 8 do Bloco XXI-B, como pode ser visualizado abaixo.



Análise de Arquivos de Histórico de Navegação do Internet Explorer:

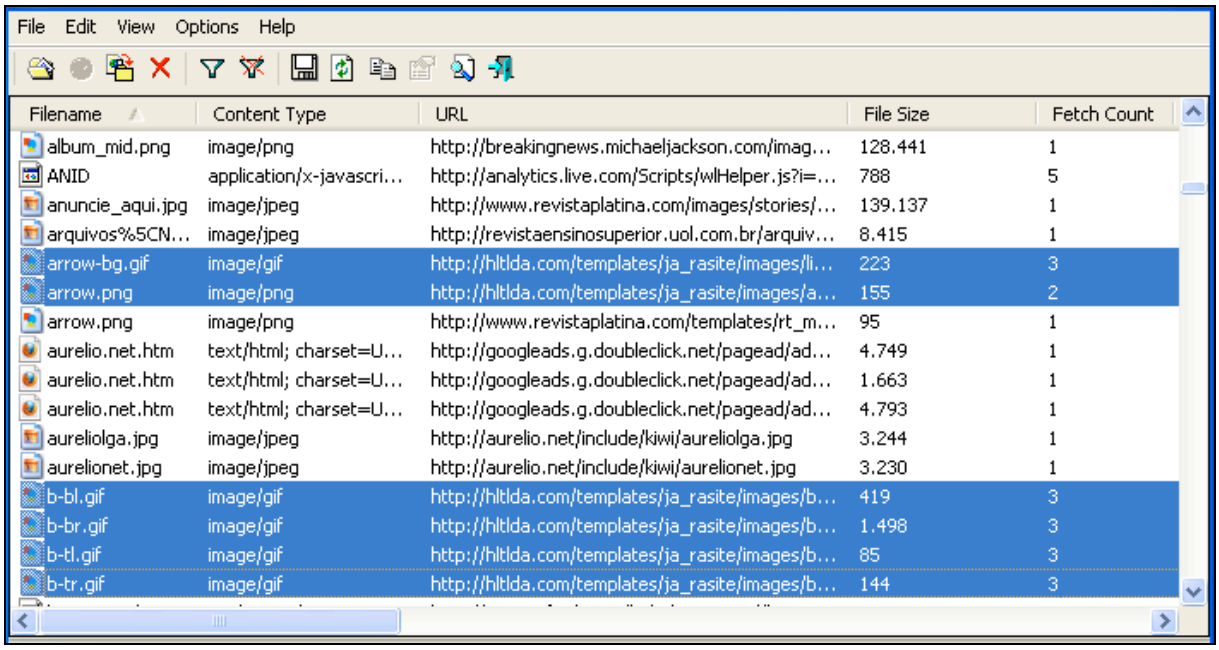
Os arquivos foram convertidos para o formato xls, e com o uso do programa Excel efetuou-se a análise dos próprios. Provas periciais indicando o acesso ao servidor comprometido no dia 11/11/10 das 20:46:30 às 21:22:45, foram encontradas no arquivo coletado do computador 5, de nome lab8b_5 do laboratório 8 do Bloco XXI-B, como pode ser apurado abaixo.

Mandiant: Web Historian - 1 - C:\Documents and Settings\lab8b\Desktop\Evidencias\Evidencias_IE\Historico Navegacao\Lab8_XXI-A\Comp1\History.IE		
URL Address	**Modified Time**	**Accessed Time**
Visited: lab8b@http://www.hltlda.com/	04.11.10 21:40	11.11.10 20:46
Visited: lab8b@http://www.vtunnel.com/index.php/1010110A/5b85fbc8507b3cb463b66c85ff6eba6ec433907ab1a6b6f8293d301053	21.10.10 20:01	11.11.10 20:47
Visited: lab8b@http://www.hltlda.com/index.php?option=com_content&view=article&id=19&Itemid=53	04.11.10 19:21	11.11.10 20:48
Visited: lab8b@file:///E:/VIDEO_TS/VIDEO_TS.VOB	26.10.10 21:32	11.11.10 20:48
Visited: lab8b@file:///E:/VIDEO_TS/VTS_02_1.VOB	26.10.10 21:34	11.11.10 20:49
Visited: lab8b@file:///J:/ARQ/UNIDADE%20CENTRAL%20DE%20PROCESSAMENTO.ppt	28.10.10 19:33	11.11.10 20:50
Visited: lab8b@http://www.hltlda.com/index.php?option=com_content&view=article&id=82&Itemid=58	04.11.10 21:35	11.11.10 20:50
Visited: lab8b@http://www.hltlda.com/index.php?option=com_content&view=article&id=20&Itemid=55	04.11.10 21:22	11.11.10 20:56
Visited: lab8b@http://[illegible]opensocial.googleusercontent.com/gadgets/ifr?url=http%3A%2F%2Fgadgetsforblogger.googlecode.com%2Ffiles%2Fpicas	04.11.10 19:21	11.11.10 20:58
Visited: lab8b@http://www.hltlda.com/index.php?option=com_content&view=article&id=26&Itemid=64	21.10.10 20:01	11.11.10 21:05
Visited: lab8b@http://www.hltlda.com/index.php?option=com_contact&view=contact&id=1&Itemid=57	21.10.10 21:29	11.11.10 21:13
Visited: lab8b@http://ead.unesc.net/ava/modules/modules/material_list/index.php?identifier=15519l35l597l24645l19016l0	04.11.10 21:22	11.11.10 21:22

Análise de Arquivos de Cache do Firefox:

A ferramenta Mozilla Cache View foi capaz de exportar os dados para o formato csv, e usando a mesma, efetuou-se a análise do arquivo conseguido. Provas periciais indicando o acesso ao servidor comprometido no dia 11/11/10 às 21:22:57, foram encontradas no arquivo coletado do computador 5, de nome lab8b_5 do laboratório 8 do Bloco XXI-B. A figura

comprovando, é mostrada abaixo.



Análise de Arquivos de Cookie do Firefox:

Usando-se a ferramenta F3E, realizou-se a conversão dos arquivos para o formato csv, e usando o programa Excel efetuou-se a análise dos mesmos. Provas periciais indicando o acesso ao servidor comprometido no dia 11/11/10 às 21:36:46, foram encontradas no arquivo coletado do computador 5, de nome lab8b_5 do laboratório 8 do Bloco XXI-B. Tal arquivo pode ser visto abaixo.

1289615281483001,"ja_rasite_tpl","ja_rasite","hltlda.com","/","1320287281","11/13/2010 02:40:56","0","0",;;;;;281","11/11/2010 21:36:46"

Análise de Arquivos de Histórico de Downloads do Firefox:

Novamente com a ferramenta F3E, exportaram-se os arquivos para o formato csv, e com o programa Excel analisou-se os mesmos. Nenhuma prova pericial foi encontrada examinando tais arquivos.

Análise de Arquivos de Histórico de Formulários Preenchidos do Firefox:

Ainda com a ferramenta F3E, geraram-se arquivos csv, e procedeu-se a análise dos mesmos com o programa Excel. Provas periciais indicando que foram realizadas buscas procurando por conhecimento sobre como invadir um servidor, e acessos ao servidor comprometido no dia 11/11/10, foram encontradas no arquivo coletado do computador 5, de nome lab8b_5 do laboratório 8 do Bloco XXI-B. O arquivo pode ser examinado abaixo.

id	fieldname	value	timesUsed	firstUsed	lastUsed
1	q	reparar erros windows	1	1285617917859000	1285617917859000
2	q	download framedyn.dll	1	1285620417924000	1285620417924000
3	q	cygwin	1	1286308673741000	1286308673741000
4	q	download internet explorer	1	1287326643238000	1287326643238000
5	q	internet explorer nao aparece	1	1288535395565000	1288535395565000
6	q	cmd dir	1	1288883535143000	1288883535143000
7	q	wbf santander	1	1289389618996000	1289389618996000
8	searchword	sobre nFigura 27. URLs relevantes do hist√≥rico de navega√ß√£o	2	1289615427448000	1289615429101000
9	searchword	como invadir um servidor web	2	1289615452494000	1289615542103000
10	searchword	sobre ns	1	1289615552658000	1289615552658000
11	name	[redacted]	3	1289615682705000	1289615978781000
12	username	[redacted]	2	1289615682705000	1289615737745000
13	email	[redacted]@gmail.com	3	1289615682705000	1289615978791000
14	subject	Estudo de Caso	1	1289615978791000	1289615978791000
15	username	[redacted]@gmail.com	2	1289615994103000	1289616012400000

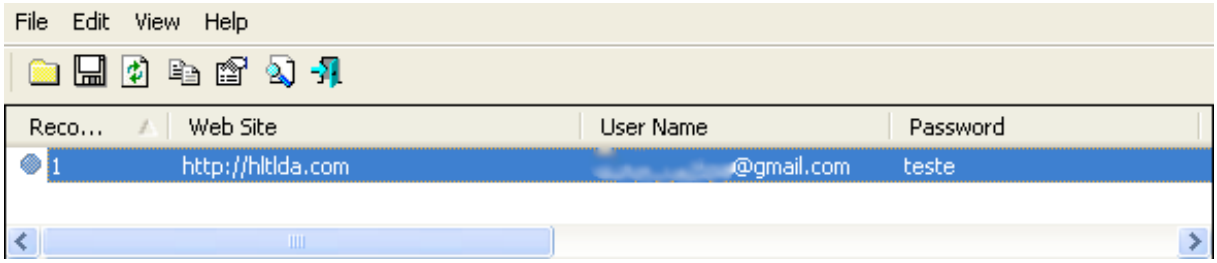
Análise de Arquivos de Histórico de Navegação do Firefox:

Os arquivos de histórico de navegação do Firefox foram convertidos com a ferramenta F3E, realizou-se então a análise dos mesmos com o Excel para visualização dos arquivos csv e o browser Firefox para visualização dos arquivos html. Provas periciais indicando o acesso ao servidor comprometido no dia 11/11/10 das 21:31:15 às 21:36:46, foram encontradas no arquivo coletado do computador 5, de nome lab8b_5 do laboratório 8 do Bloco XXI-B. O arquivo é encontrado abaixo para análise.

263	http://www.revistaplatina.com/estilo-de-vida/custom-modules
264	http://hltlda.com/index.php
265	http://hltlda.com/index.php?searchword=sobre+nFigura+27.+URLs+relevantes+do+hist%C3%B3rico+de+navega%C3%A7%C3%A3os&ordering=t
266	http://www.revistaplatina.com/component/content/article/39-rokfeature/6736-entrevista-exclusiva-yannuza
267	http://hltlda.com/index.php?searchword=como+invadir+um+servidor+web&ordering=&searchphrase=all&Itemid=89&option=com_search
268	http://hltlda.com/index.php?searchword=sobre+ns&ordering=&searchphrase=all&Itemid=89&option=com_search
269	http://hltlda.com/index.php?option=com_content&view=article&id=19:joomla-overview&catid=40:dados-hlt&Itemid=53
270	http://hltlda.com/index.php?option=com_content&view=article&id=21&Itemid=59
271	http://hltlda.com/index.php?option=com_contact&view=contact&id=1&Itemid=57
272	http://hltlda.com/index.php?option=com_user&view=register
273	http://hltlda.com/index.php?option=com_user#content
274	http://hltlda.com/index.php?option=com_contact&view=contact&id=1%3Acontact-us&catid=12%3Acontacts&Itemid=57

Análise de Nomes de Usuário e Senhas do Firefox:

Com a ferramenra PasswordFox foi possível recuperar e decodificar com sucesso nomes de usuários e senhas salvas no browser, Efetuou-se a análise dos mesmos, e provas periciais indicando que o usuário *fulano_de_tal@gmail.com* logou no servidor comprometido. As evidências foram encontradas no arquivo coletado do computador 5, de nome lab8b_5 do laboratório 8 do Bloco XXI-B. Abaixo é ilustrado o arquivo coletado.



Resultado:

Analisando-se todas as provas encontradas, especialmente combinando a análise do histórico de formulários preenchidos do Firefox e do histórico de navegação do mesmo browser, bem como a análise dos nomes de usuário e senhas salvas pelo Firefox, foi possível concluir que o acadêmico Fulano_de_Tal, realizou os acessos ao site do sistema de vendas online da empresa HLT (http://www.hltlda.com), no dia 11/11/10 nos horários 20:46:12, e 21:33:10, com objetivo de prejudicar a mesma.

CONCLUSÃO

Com o aumento crescente do número de usuários de computadores e utilizadores de Internet, aumenta proporcionalmente o número de crimes digitais cometidos. A pericia forense computacional, surge dentro desse contexto como uma arma eficaz, na luta contra a impunidade atualmente existente, contribuindo para se alcançar uma sociedade mais segura e justa. Por utilizar processos científicos na identificação e combate aos crimes digitais, cada vez mais os resultados das investigações são utilizados em processos judiciais, aumentando assim a necessidade de se ter processos otimizados de coleta, preservação e análise de evidências.

É importante mencionar que qualquer erro do perito durante a realização de uma investigação, pode invalidar a mesma, ou pior ainda, inocentar culpados ou culpar inocentes. Torna-se primordial que o perito conduza a pericia usando metodologias aprovadas e padronizadas, que certificarão aos órgãos judiciais a credibilidade da mesma.

Por existirem poucos estudos no País sobre pericia forense computacional, sobretudo com foco na coleta e análise de evidências em *web browsers*, o presente trabalho objetivou analisar e aplicar os procedimentos de perícia forense computacional na busca por evidências em tal ambiente. Para tal foram estabelecidos cinco objetivos específicos que foram alcançados no decorrer do estudo.

O primeiro objetivo específico foi abordado no Capítulo 4, onde se delineou quais os aspectos relevantes de segurança em *web browsers*. Abordou-se sobre o protocolo SSL e suas falhas, bem como, quais os problemas de privacidade gerados por *cookies*. Já o segundo objetivo específico foi atendido no Capítulo 7, onde foram abordados os principais conceitos de pericia forense computacional, e no Capítulo 11 onde foram aplicadas as técnicas e metodologias da mesma, na execução de um estudo de caso.

O terceiro objetivo específico foi alcançado no Capítulo 8, ao se descrever que informações os *browsers* (Internet Explorer e Firefox) salvam localmente e onde tais informações podem ser encontradas nas versões XP em diante do Windows. E o quarto objetivo específico foi atingido no Capítulo 9, onde foram apresentadas as ferramentas forenses open source e/ou livres que seriam usadas no estudo, bem como outras disponíveis no mercado.

Por fim, o quinto objetivo específico foi conquistado ao se abordar no Capítulo 11, que tipos de evidências foram coletadas dos *browsers* IE e Firefox no decorrer do estudo.

Dessa forma, ao se alcançar os objetivos específicos delineados, acredita-se que o objetivo geral também foi atingido, pois foi possível analisar os procedimentos necessários a execução de uma perícia forense computacional e aplicá-los com sucesso em um estudo de caso fictício. No entanto, é importante mencionar que a aplicação prática comprovou que as ferramentas estudadas não são infalíveis, ocorrendo ocasionalmente erros na conversão das evidências coletadas, e que existem muitas outras ferramentas, técnicas e procedimentos, que dependendo do ambiente podem ser utilizadas.

Encerrando, o presente estudo acaba abrindo portas para trabalhos futuros, como a análise de ferramentas que permitam, por exemplo, executar uma pericia em várias máquinas conectadas em rede, ou que permitam recuperar os arquivos deletados dos browsers. Outra linha de pesquisa seriam as técnicas e ferramentas a usar na realização de uma perícia quando o suspeito utiliza a navegação privativa, opção onipresente nos browsers hoje em dia.

REFERÊNCIAS

ABBATE, Janet. **Inventing the Internet**. Cambridge, MA, MIT Press, 2000.

AGUIAR, Daniel Pedrosa. **Estudo sobre crimes praticados na Internet com o uso do computador.** São Paulo: Faculdade de Tecnologia da Zona Leste, 2009.

ARGOLO, Frederico Henrique Böhm. **Análise Forense em sistemas GNU/Linux.** Universidade Federal do Rio de Janeiro, 2005.

BARYAMUREEBA, Venansius; TUSHABE, Florence. The Enhanced Digital Investigation Process Model. DIGITAL FORENSIC RESEARCH WORKSHOP, 2004, Maryland, USA. **Proceedings of the 2004 Digital Forensic Research Workshop.** Maryland, EUA: DFRWS, 2004. p. 1 - 9.

BASTOS, Leonara de Oliveira; LADEIRA, Adriane Cristina. **Protocolo HTTP**. UFSCAR-Universidade Federal de São Carlos, 2002. Disponível em: <http://www.comp.ufscar.br/~drica/http.html> Acesso em: 11 jun. 2010, 10:12:35.

BERNARDO, Adauto de Sousa. **Técnicas Computacionais no Auxílio à Perícia Forense na Análise de Evidências Coletadas em Servidores Gnu/Linux.** Universidade do Extremo Sul Catarinense – Unesc, 2006.

BERTOGLIO, Daniel Dalalana. **Perícia Forense: Proposta de Uma Metodologia de Coleta de Indícios Para Ambiente Windows.** Trabalho de Conclusão de Curso - Curso de Ciência da Computação, Departamento de Instituto de Ciências Exatas e Tecnológicas, Centro Universitário Feevale, Novo Hamburgo, 2008.

BRADNER, S. **Request for Comments-2026.** Harvard University, 1996. Disponível em: <http://tools.ietf.org/html/rfc2026> Acesso em: 04 maio. 2010, 14:44:00.

BRASIL ESCOLA. **História do Navegador.** 1999. Disponível em: <http://www.brasilescola.com/informatica/navegador.htm> Acesso em: 04 maio. 2010, 15:30:00.

BURZSTEIN, Elie; GAURAV, Aggarwal; JACKSON, Collin; BONEH, Dan. **An Analysis of Private Browsing Modes in Modern Browsers.** 2010. Disponível em: <http://www.collinjackson.com/research/private-browsing.pdf> Acesso em: 17 jun. 2010, 17:33:05.

CARRIER, Brian. **File System Forensic Analysis**. Indiana: Addison Wesley Professional, 2005.

CARUSO, Carlos A. A.; STEFFEN, Flávio Deny. **Segurança em Informática e de Informações.** 2a ed. São Paulo: Editora SENAC, 1999.

CASEY, Eoghan. **Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet.** Londres: Academic Press, 2004.

CERT.br. CENTRO DE ESTUDOS, RESPOSTAS E TRATAMENTO DE INCIDENTES DE SEGURANÇA NO BRASIL. **Estatísticas dos Incidentes Reportados ao CERT.br.** 2010a. Disponível em: <http://www.cert.br/stats/incidentes/> Acesso em: 13 jun. 2010, 18:03:25.

CERT.br. CENTRO DE ESTUDOS, RESPOSTAS E TRATAMENTO DE INCIDENTES DE SEGURANÇA NO BRASIL. **Incidentes Reportados ao CERT.br – Janeiro a Dezembro de 2009.** 2010b. Disponível em: < http://www.cert.br/stats/incidentes/2009-jan-dec/tipos-ataque.html> Acesso em: 17 jun. 2010, 17:13:05.

CETIC.br. CENTRO DE ESTUDOS SOBRE AS TECNOLOGIAS DA INFORMAÇÃO E DA COMUNICAÇÃO. **Pesquisa sobre o uso das Tecnologias da Informação e da Comunicação no Brasil 2008.** São Paulo, 2009.

FONSECA, Allan Valin Ribeiro da. **A História e a Evolução dos Navegadores.** 2009. Disponível em: < http://www.baixaki.com.br/info/2063-a-historia-e-a-evolucao-dos-navegadores.htm> Acesso em: 04 maio, 15:29:00.

FOROUZAN, Behrouz A. **TCP/IP Protocol Suíte.** McGraw-Hill, 2006.

GALVÃO, Ricardo Kleber Martins. **Perícia Forense em Web Browsers.** Centro Federal de Educação Tecnológica do Rio Grande do Norte, 2008.

HERRMANN, Eric. **Aprenda em 1 semana programação CGI em Perl 5.** Rio de Janeiro: Campus, 1997.

HEWITT, Peter C.; PELÁEZ, Manuel H. Santander. **Browser Forensics.** 2010. Disponível em: < http://www.browserforensics.org/wp-content/uploads/2010/03/BrowserForensics-v1-03-03-2010.pdf > Acesso em: 13 jun. 2010, 11:23:31.

HOUAISS, Antonio. **Dicionário Houaiss da Língua Portuguesa: Com a nova Ortografia da Língua Portuguesa.** Editora: Objetiva, 2009.

ISO/IEC. International Organization for Standardization/ International Electrotechnical Commission. **Tecnologia da Informação - Código de Prática para Gestão da Segurança de Informações.** 2004.

JESUS, Fernando de. **Perícia e Investigação de Fraude.** 2. ed. Goiania, GO: AB, 2000.

JONES, Keith J.. **Forensic Analysis of Internet Explorer Activity Files.** 2003. Disponível em: < http://www.foundstone.com/us/pdf/wp_index_dat.pdf > Acesso em: 16 jun. 2010, 11:23:31.

JONES, Keith J.; BELANI, Rohyt. **Web Browser Forensics: Part 1.** 2005a. Disponível em: <http://www.securityfocus.com/infocus/1827> Acesso em: 26 out. 2009, 11:23:31.

JONES, Keith J.; BELANI, Rohyt. **Web Browser Forensics: Part 2.** 2005b. Disponível em: <http://www.securityfocus.com/infocus/1832> Acesso em: 26 out. 2 009, 11:40:22.

JONES, R. Jeffrey. **Browser Vulnerability Analysis:** Of Internet Explorer and Firefox. 2007.

KRISHNAMURTHY, Balachander; REXFORD, Jennifer. **Redes para a Web: HTTP/1.1, Protocolos de Rede, Caching e Medição de Tráfego.** Editora Campus, 2001.

MARTINS, G. A; THEÓPHILO, C. R. **Metodologia da investigação científica para ciências aplicadas** – São Paulo: Atlas, 2009

PAIVA, Luciano Carneiro de. **A prova nos crimes de Informática: Aspectos Técnicos e Jurídicos.** Dissertação, 2006.

PINHEIRO, Patrícia Peck. **Direito Digital.** Editora Saraiva, 2ª edição, 2008.

PINHEIRO, Reginaldo César. **Os crimes virtuais na esfera jurídica brasileira.** São Paulo, 2001.

PLACE, Ricardo Leocádio. **Criptografia, assinatura digital e alguns outros conceitos.** Disponível em: <http://eltiger.wordpress.com/2008/10/12/ciptografia-assinatura-digital-e-alguns-outros-conceitos/> Acesso em: 26 out. 2009, 09:28:57.

POUW, Keesje; GEUS Paulo. **Desenvolvendo Aplicações Seguras Em Ambiente Html/Https.** Universidade Estadual de Campinas – UNICAMP, 1999.

REITH, Marc; CARR, Clint; GUNSCH, Gregg. **An Examination of Digital Forensic Model**. International Journal of Digital Evidence, Nova York, EUA, v. 1, n. 4, p. 1 – 12, 2002.

SAFERNET BRASIL. **Quem Somos.** 2009. Disponível em: <http://www.safernet.org.br/site/institucional > Acesso em: 16 nov. 2009, 15:25:5 1.

SCHWEITZER, Douglas. **Incident Response: Computer Forensics Toolkit.** Indiana: Wiley Publishing, 2003.

SWGDE. SCIENTIFIC WORKING GROUP ON DIGITAL EVIDENCE. **Digital Evidence: Standards and Principles.** 2008. Disponível em: <http://www.fbi.gov/hq/lab/fsc/backissu/april2000/swgde.htm> Acesso em: 12 nov. 2009, 15:10:45.

TANENBAUM, Andrew S. **Redes de Computadores.** Editora Campus, 3a edição,2003.

UNICAMP. Universidade Estadual de Campinas. **A evolução da Internet.** Universidade Estadual de Campinas, 1998. Disponível em: <http://www.revista.unicamp.br/infotec/internet/internet1-1.html> Acesso em: 14 jun 2010.

UFPA. Universidade Federal da Paraíba. **Internet - História.** 2004. Disponível em: <http://www.cultura.ufpa.br/dicas/int-his.htm> Acesso em: 15 jun 2010.

VACCA, John R. **Computer Forensics: Computer Crime Scene Investigation.** Massachusetts: Charles River Media, 2002.

W3Counter. **Global Web Stats October 2009.** 2009. Disponível em: <http://www.w3counter.com/globalstats.php> Acesso em: 10 nov. 2009, 16:30:30.

WEST, Jackson. **Acused murderer Googled "how to kill with a knife".** 2008. Disponível em: <http://valleywag.gawker.com/5017716/accused -murderer-googled-how-to-kill-with-a-knife> Acesso em: 16 nov. 2009, 15:1 3:31.

WILLRICH, Roberto. **Conceitos básicos de informática.** Universidade Federal da Paraíba. 2004. Disponível em: <http://www.comp.ufla.br/~monserrat/icc/Historia2.pdf> Acesso em: 15 jun 2010.

ZAKON, Robert. **Hobbes' Internet Timeline v6.1.** 2004. Disponível em: <http://www.simonevb.com/hobbestimeline> Acesso em: 23 maio 2010.

ZALEWSKI, Michal. **Browser Security Handbook.** 2009. Disponível em: <http://code.google.com/p/browsersec/wiki/Main> Acesso em: 1 0 out. 2009, 13:30:40.

APÊNDICE A – REALIZANDO A PERÍCIA FORENSE NO FIREFOX EM SISTEMAS DERIVADOS DO UNIX

Para realizar uma perícia forense em um *browser* Firefox nos sistemas derivados do UNIX é necessário ter em atenção os caminhos onde o mesmo salva as informações do usuário. Logo, o perito deve ter em atenção os seguintes caminhos:

a) para arquivos de *cookies*, senhas, campos de formulário, histórico de navegação, entre outras informações: /home/<usuário>/.mozilla/firefox/<caracteres-randômicos>.default;
b) para os arquivos de *cache*: /home/<usuário>/.mozilla/firefox/< caracteres-randômicos>.default/Cache.

Como já foi mencionado no decorrer deste trabalho, embora seja recomendado que o perito use ferramentas específicas para conduzir a sua investigação os browsers permitem a visualização de certos conteúdos por padrão. Assim, informações como o histórico de navegação, *cookies* e *cache*, podem ser visualizados diretamente no Firefox.

Para visualizar o histórico de navegação diretamente no browser, o perito deve usar a opção “Histórico” do menu do programa, ou usar a combinação das teclas de acesso <Ctrl-h>. Os *cookies* podem ser visualizados, selecionando-se no menu: Edit / Preferences / Privacy / Cookies / View Cookies (versão em inglês do Firefox). Já os arquivos de *cache* podem ser visualizados, digitando-se na barra de endereços do *browser* “about:cache”.

Outra opção, e a mais viável, para a condução de uma investigação mais confiável, seria usar as ferramentas existentes para versões de sistemas derivados do UNIX como o Linux. Um exemplo de uma ferramenta seria a WBF, melhor abordada abaixo.

SOBRE O WEB BROWSER FORENSICS

O programa Web Browser Forensics, mais conhecido como WBF foi desenvolvido por Manuel Santander e o seu foco principal é a análise de arquivos de histórico de navegação (GALVÃO, 2008). A ferramenta apresenta versões para os SOs Windows (Cygwin) e Linux, e possui uma licença de software livre.

Ela utiliza a interface do browser do computador para interagir com o usuário, mostrando na tela uma tabela com o histórico de navegação do usuário. Foi desenvolvida na linguagem C, e permite visualizar os históricos de navegação dos seguintes browsers: Firefox, Ópera e Epiphany.

As informações mostradas pelo WBF são básicas, constituindo-se de:

a) data da última visita a um dado URL;

b) URLs visitados;

c) numero total de visitas a determinado URL.

A Figura 17, apresenta-nos a tela do programa WBF, no browser Firefox.

Figura 17. Tela do programa WBF no browser Firefox
Fonte: adaptado GALVÃO, R. (2008)

APÊNDICE B - DOCUMENTAÇÃO DA PERICIA FORENSE REALIZADA EM 12 DE NOVEMBRO DE 2010

Identificação do Perito

Nome: Sidney Roberto da Silva Webba

Qualificações: Finalista do Curso de Ciência da Computação da UNESC

Endereço: Av. Universitária, 1105 - Bairro Universitário C.P. 3167 | CEP: 88806-000 Criciúma / Santa Catarina

Telefone para contato: +55 48 3431-2500

Data e Horário de Início da Pericia Forense: 12 de Novembro de 2010. 18h38m

Data e Horário de Fim da Pericia Forense: 12 de Novembro de 2010. 21h19m

Local da Pericia Forense: Laboratório 8 do Bloco XXI-A e Laboratórios 16 do Bloco XXI-B da UNESC. Av. Universitária, 1105 - Bairro Universitário C.P. 3167 | CEP: 88806-000 Criciúma / Santa Catarina

Alguns dados acima não correspondem a realidade para manter a confidencialidade de informações consideradas pessoais.

Computadores Periciados

	Lab 16, Bloco XXI-A	Lab 8, Bloco XXI-B
Nome do Computador	comp16_1	lab8b_1
Nome do Computador	comp16_2	lab8b _2
Nome do Computador	comp16_3	lab8b _3
Nome do Computador	comp16_4	lab8b _4
Nome do Computador	comp16_5	lab8b _5
Nome do Computador	comp16_6	lab8b _6
Nome do Computador	comp16_7	lab8b _7
Nome do Computador	comp16_8	lab8b _8
Nome do Computador	comp16_9	lab8b _9
Nome do Computador	comp16_10	lab8b _10

É importante ressaltar que todos os computadores apresentam as mesmas especificações de hardware e software, e que os mesmos procedimentos computacionais foram realizados em cada uma das máquinas.

Especificações de Hardware:

e) Memória RAM: 2 GB;

f) Disco Rígido: 150 GB;

g) Processador: Intel Core 2 Duo;

h) Velocidade do Processador: 2,26 GHz;

i) Número de Processadores: 1;

j) Número de Núcleos: 2;

k) Placas de áudio, vídeo e rede on-board;

l) Fabricante: HP;

a) Modelo: HP Compaq dc5850 Microtower.

Especificações de Software:

a) SO: Windows XP, Service Pack 3;

b) Browsers: IE versão 8.0.6, Firefox versão 3.6.12.

Procedimentos Computacionais:

a) Execução dos comandos DATE e TIME para verificação do horário de início da perícia;

b) Verificação usando a aplicação "Informações do Sistema" do Windows as configurações de hardware e software das estações, bem como os nomes de cada máquina periciada;

c) Usando-se o comando *md5sum* gerou-se o *hash* da evidência encontrada no caminho: C:\Documents and Settings\<usuário>\Local Settings\Temporary Internet Files\Content.IE5\;

d) Copiou-se a evidência para o caminho: D:\Evidencias_IE\Cache\ e gerou-se o

hash do arquivo copiado;

e) Usando-se o comando *md5sum* gerou-se o *hash* da evidência encontrada no caminho: C:\Documents and Settings\<usuário>\Local Settings\History\History.IE5;

f) Copiou-se a evidência para o caminho: D:\Evidencias_IE\Historico\ e gerou-se o *hash* do arquivo copiado;

g) Usando-se o comando *md5sum* gerou-se o *hash* da evidência encontrada no caminho: C:\Documents and Settings\<usuário>\Cookies\;

h) Copiou-se a evidência para o caminho: D:\Evidencias_IE\Cookies\ e gerou-se o *hash* do arquivo copiado;

i) Usando-se o comando *md5sum* gerou-se o *hash* da evidência encontrada no caminho: C:\Documents and Settings\<usuário>\ApplicationData\Mozilla\Firefox\Profiles\<caracteres-randômicos>.default\;

j) Copiou-se a evidência para o caminho: D:\Evidencias_FF\ e gerou-se o *hash* do arquivo copiado;

k) Usando-se o comando *md5sum* gerou-se o *hash* da evidência encontrada no caminho: C:\Documents and Settings\<usuário>\Local Settings\Application Data\Mozilla\Firefox\Profiles\<caracteres-randômicos>.default\Cache\;

l) Copiou-se a evidência para o caminho: D:\Evidencias_FF\ e gerou-se o *hash* do arquivo copiado;

m) Usando-se a ferramenta Pasco, gerou-se um arquivo *.xls* para cada arquivo de *cache* do IE coletado;

n) O mesmo procedimento foi realizado com a ferramenta Galleta, gerando-se neste caso um arquivo *.xls* para cada arquivo de *cookie* do IE coletado;

o) Voltou-se a repetir o procedimento, desta vez com os arquivos de histórico de navegação do IE, e usando a ferramenta Web Historian;

p) Por último a ferramenta F3E foi usada para converter os arquivos *.sqlite* do Firefox em arquivos *.csv* e *.html;*

q) Os arquivos *.xls* e *.csv* foram analisados com o uso da ferramenta Excel do pacote Office;

r) Já os arquivos *.html* foram analisados usando o browser Firefox.

s) Execução dos comandos DATE e TIME para verificação do horário de final da perícia.

Hashes Criados:

a) Pasta com os arquivos de cache do IE: 7957272e69f90e695af79fe2d5f157f5;

b) Pasta com os arquivos de Histórico de Navegação do IE: a65c6d1e5f38d9d8f085bc9abf03f20c;

c) Pasta com os arquivos de cookies do IE: b2442d3c5e6fa8919e069be186d8ff37;

d) Pasta com os arquivos .sqlite do Firefox: be0f3e6a22bb033da6c6054f17489e85;

e) Pasta com os arquivos de cache do Firefox: 150583d00d56eda859320f81079fb0c9.

Resultados:

Foram encontradas provas periciais no computador de nome lab8b_5, do laboratório 8 do bloco XXI-B.

APÊNDICE C – ARTIGO: PROCEDIMENTOS COMPUTACIONAIS NO AUXÍLIO À PERÍCIA FORENSE EM WEB BROWSERS

Procedimentos Computacionais no Auxílio à Perícia Forense aplicada em Web Browsers

Sidney Roberto da Silva Webba[1], Paulo João Martins[2]

[1]Acadêmico do Curso de Ciência da Computação - Departamento de Ciência da Computação - Universidade do Extremo Sul Catarinense (UNESC) – Criciúma / SC – Brazil

[2]Professor do Curso de Ciência da Computação - Departamento de Ciência da Computação - Universidade do Extremo Sul Catarinense (UNESC) – Criciúma / SC – Brazil

sidney_webba@hotmail.com, pjm@unesc.net

Abstract. *This paper describes the conclusion work submitted for obtaining the Degree of Bachelor of Computer Science at the UNESC University, whose goal was to analyze and apply the procedures of forensic computing, focusing on collecting and analyzing evidence in web browsers, contributing to increasing the range of research on the subject. To achieve it we performed a literature search and a fictional case study simulating the execution of a computer forensics analysis at the university in question, using the SOP methodology.*

Resumo. *O presente artigo descreve o trabalho de conclusão de curso apresentado para obtenção do Grau de Bacharel em Ciência da Computação da Universidade do extremo Sul Catarinense, cujo objetivo foi analisar e aplicar os procedimentos de perícia forense computacional, com foco na coleta e análise de evidências em web browsers, contribuindo para o aumentando do leque de pesquisas sobre o tema. Para a realização do mesmo efetuou-se uma pesquisa bibliográfica, bem como um estudo de caso fictício simulando a condução de uma pericia forense computacional na universidade em questão, utilizando-se da metodologia SOP.*

1. Introdução

Com a popularização da Internet muitas atividades que antes requeriam um esforço físico considerável, podem hoje ser realizadas pelo computador, com o simples clique de um botão, de maneira cômoda mas ainda insegura.

Acompanhando essa evolução, os crimes comuns vêm se tornando cada vez mais tecnologicamente avançados. Fraudes eletrônicas, roubos de identidade, espionagem industrial, transmissão de pornografia infantil, pedofilia e incidentes de segurança convencionais como vírus, worms, phishing, hacking (casual ou direcionado), são exemplos de crimes que ocorrem na rede. Percebe-se portanto, a importância da definição de procedimentos que permitam a condução de uma análise em sistemas comprometidos, que

tem por objetivo responder a questões como: quando aconteceu o incidente, como o mesmo foi realizado, quem foi o responsável, porque aconteceu e onde aconteceu.

Surge dentro desse contexto, a análise forense, como uma ciência capaz de assegurar que as manipulações das evidências eletrônicas deixadas pelos infratores sejam aceitas em juízo. Mas, embora os esforços das autoridades competentes tenham aumentado consideravelmente nos últimos anos, pouco se discute dentro da comunidade acadêmica sobre como proceder numa avaliação forense, coletando e manipulando evidências eletrônicas de modo a que a integridade das mesmas seja preservada, assegurando assim a confiabilidade dos dados, e permitindo que as mesmas sejam usadas em processos criminais.

Este trabalho propõe-se então, a analisar e estudar o processo forense em si, focando-se nas técnicas e softwares usados atualmente para coletar e analisar evidências oriundas de web browsers.

2. Perícia Forense Computacional

A aplicação de estudos científicos a lei é chamada de ciência forense (CASEY, 2004). Podemos então, definir a mesma como a utilização da ciência ou da tecnologia em processos investigativos, estabelecendo fatos ou evidências num tribunal de justiça (CARRIER, 2005).

Por sua vez, a utilização dela em sistemas computacionais durante investigações oficiais por peritos judiciais, é comumente chamada de perícia forense computacional (BERNARDO, 2006).

Segundo Vacca (2002) a perícia forense computacional é definida como a coleta, preservação, análise e apresentação de evidências digitais, utilizando-se de ferramentas e técnicas computacionais no ambiente investigado, e auxiliando os juízes nas tomadas de decisões num processo judicial.

Para que as evidências sejam validadas judicialmente, é extremamente importante que os dados, as informações e as provas periciais sejam coletadas, analisadas e apresentadas somente por um perito no assunto, evitando que durante o processo não ocorra a perda ou contaminação das mesmas.

3. Metodologias Investigativas

Os procedimentos técnicos a serem realizados pelo perito forense podem ser diferentes de acordo com os sistemas e aparatos tecnológicos envolvidos. A falta de métodos específicos que não se alterassem de acordo com a tecnologia usada, enfraquecia a credibilidade de provas periciais apresentadas em casos judiciais.

Na tentativa de dar-se maior credibilidade e solidez à perícia forense computacional em frente à jurados, criaram-se metodologias que são usadas como guias do processo investigativo, definindo etapas a serem seguidas pelos peritos, independentemente de qual o sistema computacional, ou de quais ferramentas foram usadas (BERNARDO, 2006). Uma dessas metodologias é melhor explicitada abaixo.

3.1 Metodologia SOP

Criada pelo Scientific Working Group on Digital Evidence (SWGDE), que é o representante norte-americano na International Organization on Computer Evidence (IOCE).

A metodologia Standard Operating Procedures (SOP), foi apresentada durante a International Hi-Tech Crime and Forensics Conference (IHCFC), realizada em Londres, de 4 a 7 de outubro de 1999 (BERTOGLIO, 2008). Ela incorpora os conceitos e procedimentos da ciência forense, incluindo: comparação, classificação, individualização e avaliação da fonte de evidências, sendo constituída por 6 etapas (BERNARDO, 2006):

a) autorização e preparação: Antes de qualquer investigação, o perito forense deve certificar-se de que não está infringindo nenhuma lei. Caso contrário sua perícia será afetada ou simplesmente invalidada. Torna-se então obrigatório, que o perito receba a autorização de um juiz, para efetuar a mesma, ficando o exame restrito somente ao que for determinado pelo órgão judicial;

b) identificação: Nesta etapa deve-se fazer o levantamento das informações relevantes ao crime e a identificação de todo o hardware e software do computador a ser examinado;

c) coleta e preservação: Após a identificação das fontes de evidências, estas devem ser coletadas e mais tarde autenticadas. É primordial que as evidências não sejam alteradas durante todo o processo investigativo. É aconselhável que o perito calcule o valor hash dos arquivos originais, antes de copiar as evidências. Segundo Place (2008) um hash é uma seqüencia de letras ou números gerados por um algoritmo de dispersão, buscando identificar um arquivo ou informação unicamente. Trata-se de um método para transformar dados de tal forma que o resultado seja o mais exclusivo possível. Assim sendo, uma função hash recebe determinado valor e retorna um código que funciona como um identificador único. Ao fazer-se a cópia de um arquivo, se a mesma for fiel ao arquivo original, ambas apresentarão o mesmo valor hash. O perito garante assim a integridade dos dados e a credibilidade da sua perícia;

d) exame e análise: Fase posterior à coleta das evidências, onde estas serão analisadas pelo perito na busca de provas;

e) documentação: A documentação é essencial em todas as etapas da perícia forense. Primeiro porque, caso seja necessário outro perito dar sequência ao processo, o seu trabalho estará facilitado, e também visto que, a perícia terá maior credibilidade se a documentação estiver completa, com dados como: quem coletou e tratou as evidências com data e hora, e os valores hash de todas as evidências copiadas para demonstrar que as cópias estão livres de alteração e são autênticas;

f) Reconstrução da Cena do Crime: Esta etapa tenta responder as seguintes perguntas: o que aconteceu? Quem executou? Quando aconteceu? Onde aconteceu? Como aconteceu? E Por quê?

Segundo o SWGDE (2008) SOPs são documentos únicos, específicos para determinado propósito, que descrevem os métodos e procedimentos a serem seguidos na realização de operações de rotina. Elas devem ser revistas anualmente, sendo as versões previamente aprovadas, guardadas para referência.

Os padrões desenvolvidos pelo SWGDE seguem o princípio, de que todas as organizações que trabalham com a investigação forense devem conservar um alto nível de qualidade, a fim de assegurar a confiança e a exatidão das evidências, e por se tratar de uma organização mundial, amplamente reconhecida, será esta a metodologia usada na aplicação prática do presente trabalho.

Ainda que o perito siga rigidamente metodologias internacionais de perícia como as mostradas acima, ele deve sempre levar em consideração as leis e regras que regem o ambiente onde a perícia será executada. Por exemplo, se a perícia for executada numa empresa, ela deve estar de acordo com as regras internas da empresa, leis municipais, estaduais e federais para que a mesma não seja invalidada (BERTOGLIO, 2008).

4 Perícia Forense em Web Browsers

Ao falar-se de perícia forense em web browsers devemos levar em consideração dois lados: o lado do cliente (o browser em si), e o lado do servidor (servidores web, servidores de aplicação, servidores de banco de dados). Deve-se ainda, levar em consideração outros fatores

relevantes como o tráfego de rede e as características dos sistemas operacionais utilizados (CARUSO, 1999).

Tal perícia é normalmente usada em casos de: pornografia infantil, pedofilia, fraudes eletrônicas, roubo de identidade, espionagem industrial, e incidentes de segurança convencionais (vírus, worms, phishing, hacking, entre outros); objetivando sempre identificar se o usuário do equipamento periciado é vítima ou se está envolvido no incidente (JONES, 2003).

O fato de existirem muitas opções de browsers disponíveis no mercado, dificulta o trabalho do perito forense, uma vez que, em alguns casos, ele pode não dispor de uma ferramenta adequada ao browser a ser analisado.

Devido a imensidão de técnicas e ferramentas existentes, a presente pesquisa restringiu-se aquelas usadas para coletar e analisar evidências digitais no lado do cliente, ou seja, nos browsers em si, respectivamente no Internet Explorer e no Firefox, nas suas versões 8.0.6 e 3.6.12 para o SO Windows XP.

4.1 O que Procurar Primeiro

Segundo Hewitt e Peláez (2010) a quantidade de informações encontradas durante uma pesquisa forense é elevada, e deve-se estreitar o mais cedo possível o foco, determinando quais informações são relevantes ou não, para o caso em mãos.

Para tal, deve-se verificar primeiro como o usuário interagia com o seu web browser, procurando por:

a) Pesquisas realizadas – ao verificar as pesquisas realizadas pelo suspeito, o perito poderá não apenas conhecer os assuntos de interesse do mesmo, mas também determinar "palavras-chave" que poderão ser úteis mais tarde, ao se rever os arquivos mantidos pelo sistema;

b) Histórico de navegação – compreenderá a maior parte da pesquisa, sendo a informação mais importante a ser pesquisada. É importante lembrar que até sites aparentemente inofensivos podem dar pistas sobre as atividades do suspeito;

c) Arquivos baixados pela Internet – numa pesquisa forense em web browsers esta é a segunda peça mais importante de informação. Arquivos baixados da Internet são a principal causa de danos em sistemas computadorizados;

d) Informações fornecidas (Formulários/Senhas) – podem dar pistas sobre outros locais que o suspeito possa ter visitado, como clientes de email, contas bancárias, entre outros. É importante obter junto a empresa dona do site a ser verificado, a permissão para tal, visto que o perito passará a se mover não apenas no browser do suspeito, mas também num servidor remoto;

e) E-mails – os emails do suspeito podem ser conseguidos muitas vezes não apenas no site do cliente de email, mas no próprio browser do suspeito, visto que muitos deles, atualmente permitem salvar o conteúdo dos emails no disco rígido, caso o usuário assim o queira;

f) Cookies – dados importantes podem ser salvas nos cookies. Eles são importantes para o perito forense, pois através deles é possível recuperar informações mesmo que o suspeito tenha sido cuidadoso o suficiente de apagar o seu histórico de navegação.

Outros elementos que podem dar informações importantes ao perito são: o cache do browser e os arquivos temporários de Internet, bem como a pasta de sites favoritos do suspeito, ou arquivos com a extensão .url (GALVÃO, 2008).

O perito deverá saber onde buscar tais informações, e para tal faz-se necessário que conheça, além da estrutura do sistema de arquivos do SO, como o *browser* usado salva as informações do utilizador e onde.

5 Ferramentas Forense

Algumas evidências coletadas pelo perito forense podem ser visualizadas com o uso de ferramentas comuns, como editores de texto, ou ainda no próprio browser, mas o uso de ferramentas específicas é recomendado devido as seguintes vantagens que elas proporcionam (GALVÃO, 2008):

a) identificação automática da localização de arquivos;

b) resolução de problemas, como nomes diferentes de arquivos (em função, por exemplo, do idioma ou versão do SO);

c) parser automático de arquivos codificados;

d) uso em vários tipos de browsers;

e) apresentação mais agradável das informações;

f) relatórios mais detalhados;

g) exportação para formatos manipuláveis.

Existem várias ferramentas que permitem realizar uma análise forense em web browsers auxiliando um perito, como: Web Historian, Pasco, NetAnalysis, Galleta, Cache View, IE HistoryView, IE CookiesView, Mozilla HistoryView, Mozilla CookiesView, Mozilla CacheView, Web Browser Forensics (WBF), Firefox 3 Extractor, Cache Monitor, Internet Cache Explorer, IE Cache Auditor, Web Cache Illuminator, STG Cache Audit, Index.dat Analyzer, Forensic Tool Kit, IE History Manager, EnCase, Autopsy / Sleuthkit, entre outras.

Para o presente trabalho foram selecionadas ferramentas open source, ou seja, softwares que sejam disponibilizados sob uma licença de código aberto, e softwares livres, que embora não tenham o seu código fonte disponibilizado, são gratuitamente distribuídos. A vantagem de se trabalhar com software livre é clara, o baixo custo de aquisição. Já os benefícios de se trabalhar com softwares de código aberto são segundo Argolo (2005):

a) Baixo Custo – softwares sob a licença open source são normalmente gratuitos, ou o seu custo de aquisição é muito reduzido;

b) Segurança – o fato de o seu código fonte ser disponibilizado a comunidade, faz com que ele seja regularmente analisado e melhorado;

c) Continuidade – caso os desenvolvedores originais do software descontinuem as atualizações do mesmo, a comunidade pode fazê-lo, ou o código fonte pode ainda ser usado para iniciar outros projetos;

d) Flexibilidade – o código fonte do software, pode ser modificado para melhor satisfazer um usuário, atendendo a características específicas. Entre outras vantagens.

As ferramentas escolhidas e que foram usadas no decorrer do estudo de caso são:

a) Pasco - A ferramenta Pasco é de autoria de Keith Jones e possui uma licença de código aberto. O seu nome vem do latim significando “busca”, e o seu foco principal é a análise de arquivos de *cache* do IE. Ela possui versões para Windows (Cygwin), Linux, Mac OSX, e BSDs;

b) Galleta - Outra ferramenta de autoria de Keith Jones e que possui também uma licença de código aberto é o Galleta. O seu nome vem do espanhol que se traduz para inglês como *cookie*. A principal função desta ferramenta é a análise dos arquivos de *cookie* gerados pelo IE. Assim como o Pasco, o Galleta também possui versões para Windows (Cygwin,), Mac OSX, Linux, e BSDs;

c) Web Historian - Este programa possui uma licença de software livre, e o seu código fonte não é disponibilizado com ele. A ferramenta foi desenvolvida por Red Cliff's, sendo o seu principal objetivo ler os arquivos responsáveis por salvar o histórico de navegação do usuário, e apresentá-los de uma maneira mais amigável;

d) Firefox 3 Extractor - A ferramenta Firefox 3 Extractor (F3E) possui uma licença de software livre, e o seu principal objetivo é extrair dados dos arquivos *.sqlite* usados pelo Firefox e pelo Google Chrome para salvar o *cache*, o histórico de navegação, histórico de downloads, *cookies*, e outros dados do usuário. A sua interface é disponibilizada apenas pela linha de comandos, e a sua execução é restrita ao SO Windows.

Outras ferramentas consideradas importantes, e que também foram usadas são:

a) MozillaCacheView – é uma ferramenta gratuita que permite ao perito procurar e visualizar os arquivos de cache do browser Firefox de uma maneira mais facilitada e amigável;

b) IECacheView – esta ferramenta realiza a mesma tarefa que a anterior, com a particularidade de ela só trabalhar com arquivos do IE;

c) MozillaCookiesView – é uma ferramenta gratuita, que realiza a mesma função do Galleta, anteriormente citado, que é a de exibir os cookies salvos no browser. Neste caso os arquivos reconhecidos são os do Firefox;

d) PasswordFox – ferramenta que permite recuperar nomes de usuário e senhas salvas pelo Firefox.

Na próxima secção são apresentados: o estudo desenvolvido, e a metodologia usada para desenvolver o mesmo, visando ratificar a pesquisa.

6. Estudo de Caso

O presente estudo de caso foi realizado na Universidade do Extremo Sul Catarinense (UNESC), localizada em Criciúma/SC. A instituição possui uma estrutura com 27 laboratórios de informática de grande porte (até 24 computadores), e 6 laboratórios de pequeno porte (até 12 computadores), sendo os laboratórios 13 e 14 do Bloco XXI-C reservados para uso livre da comunidade interna da mesma. A comunidade externa da universidade tem disponível, Internet gratuita nos computadores da biblioteca.

Nos computadores disponíveis para a comunidade interna, é necessário possuir um código de matrícula que é requisitado juntamente com uma senha na hora da efetuar o login nas máquinas.

Para o controle e segurança dos laboratórios, já que muitas pessoas acessam as máquinas, a instituição possui uma política de segurança adequada ao ambiente pedagógico, principalmente no quesito de uso da Internet. Assim sendo, os sites são categorizados e o bloqueio efetivo de alguns deles é realizado. Além da política de segurança, é possível encontrar as Normas de Utilização dos Laboratórios no Departamento de Tecnologia da Informação da UNESC, onde funciona a coordenação dos Laboratórios de Informática (LabInfo) que é responsável por essa estrutura. Estas normas também estão fixadas em cada um dos laboratórios.

Para a realização da perícia forense e aplicação da metodologia SOP foram escolhidos aleatoriamente os laboratórios 16 do Bloco XXI-A e 8 do Bloco XXI-B, e da mesma forma, delimitaram-se 10 computadores a serem analisados em cada sala. Faz-se necessário mencionar, entretanto, que o correto na realização de uma pericia forense computacional desse tipo, seria a análise dos computadores de todos os laboratórios da instituição. Por se tratar de um caso fictício, optou-se por escolher uma amostra do total de máquinas, realizando-se a pericia em 20 estações.

Agora, para melhor contextualizar o estudo de caso fictício, e permitir uma compreensão facilitada das etapas realizadas, foi suposto que o seguinte crime digital foi cometido:

- Alguém, usando um dos computadores dos laboratórios da UNESC e um browser, acessou de maneira indevida o Sistema de Vendas Online do Supermercado HLT (nome fictício) com fins de prejudicar o mesmo.

A seguir, são expostos os conceitos relevantes a todo processo de metodologia científica aplicado neste trabalho.

6.1 Metodologia

A pesquisa tem como embasamento um estudo de caso fictício, que simula a ocorrência de uma perícia forense nos computadores dos laboratórios da Universidade do Extremo Sul Catarinense (UNESC), objetivando buscar conhecimento detalhado sobre os procedimentos de interesse para este estudo, considerando a ocorrência de um crime digital.

Martins e Theóphilo (2009) afirmam que um estudo de caso:

> "trata-se de uma investigação empírica que pesquisa fenômenos dentro de seu contexto real [...] onde o pesquisador não tem controle sobre eventos e variáveis, buscando apreender a totalidade de uma situação e criativamente descrever, compreender e interpretar a complexidade de um caso concreto. [...] possibilita a penetração na realidade social, não conseguida pela avaliação quantitativa".

Por se considerar que cada caso tem as suas características próprias, não existe um modelo traçado de forma específica para elaboração de um estudo de caso, apenas uma sequência de práticas metodológicas, para orientação, que são: coleta das evidências, composição, análise e validação dos resultados, conclusões, verificação de possíveis interferências e relatório final. (MARTINS; THEÓPHILO, 2009).

Logo, para que se cumpram os objetivos desta pesquisa, definiu-se que a metodologia forense Standard Operating Procedures (SOP) será usada durante a realização do estudo, pois ela incorpora algumas das práticas metodológicas recomendadas acima, bem como os princípios e técnicas da ciência forense.

As etapas do modelo proposto podem ser visualizadas na Figura 1, disposta abaixo. O fluxograma ilustra cada uma das etapas com algumas das ações a serem tomadas no decorrer da perícia. Tais ações atuam como processos em um projeto, pelo fato de que se é respeitada uma ordem, mesmo que, se necessário, a investigação retorne a determinada etapa.

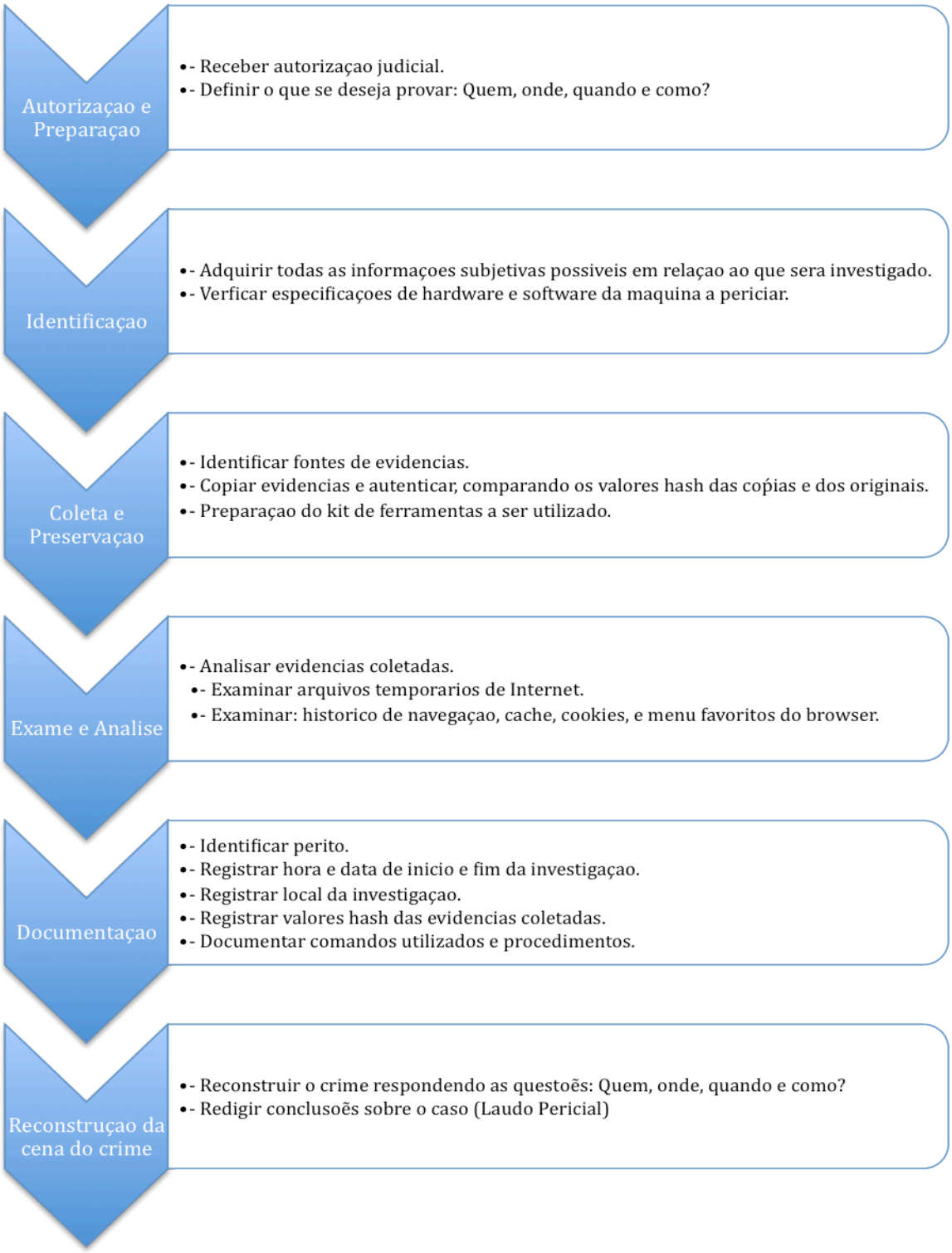


Figura 1. Fluxograma de Etapas da Metodologia SOP

Algo importante de se mencionar, é o fato de que no processo tradicional de análise forense, o perito copia as evidências encontradas para um disco ou computador “limpo”, podendo assim examiná-las em um ambiente controlado. Tal abordagem foi aplicada no trabalho em mãos.

6.2 Etapa 1 – Autorização e Preparação

Pelo fato de o presente estudo de caso ser fictício, não se fez necessária a busca por uma autorização judicial, para que a perícia forense fosse realizada. No entanto, para que os propósitos da pesquisa se cumpram e a metodologia SOP seja seguida da maneira mais fiel possível, recebeu-se a autorização do Prof. MSc. Rogério Casagrande, professor da disciplina de Redes de Computadores do curso de Ciência da Computação e gerente do Departamento de Tecnologia da Informação da UNESC, para a realização de uma perícia nos computadores da instituição.

6.3 Etapa 2 – Identificação

Novamente por se tratar de um estudo de caso hipotético, não foi necessário o levantamento junto as pessoas envolvidas no crime (pessoas que identificaram a ocorrência do crime; pessoas que sofreram as conseqüências do ato criminoso; responsáveis da Empresa lesada e dos laboratórios da UNESC) de informações relevantes, por meio de questionamentos, que permitiriam ao perito contextualizar melhor os fatos que surgissem durante o decorrer da investigação.

Passou-se então para a aquisição das especificações relevantes de hardware e software das máquinas a serem periciadas. Para tal, acessou-se o programa “Informações do Sistema” encontrado sob o menu *Iniciar/Todos os Programas/Acessórios/Ferramentas do Sistema/*, disponibilizado no Windows XP.

Todos os computadores apresentaram as seguintes configurações de hardware:

a) Memória RAM: 2 GB;

b) Disco Rígido: 150 GB;

c) Processador: Intel Core 2 Duo;

d) Velocidade do Processador: 2,26 GHz;

e) Número de Processadores: 1;

f) Número de Núcleos: 2;

g) Placas de áudio, vídeo e rede on-board;

h) Fabricante: HP;

i) Modelo: HP Compaq dc5850 Microtower.

Bem como, as seguintes especificações de software:

a) SO: Windows XP, Service Pack 3;

b) Browsers: IE versão 8.0.6, Firefox versão 3.6.12.

Com o detalhamento de hardware e software foi possível de maneira mais facilitada identificar as fontes de evidências digitais, e selecionar que ferramentas seriam usadas, preparando-se assim o kit de investigação.

6.4 Etapa 3 – Coleta e Preservação

Como já foi mencionado no decorrer do trabalho, o kit de ferramentas a ser utilizado foi composto pelos softwares:

a) Pasco;

b) Galleta;

c) Web Historian;

d) Firefox 3 Extractor;

e) Mozilla Cache View, e;

f) PasswordFox.

Além destes, utilitários nativos do sistema operacional como a linha de comandos do Windows, e outros programas como: Cygwin (Emulador de um ambiente semelhante ao Linux, no Windows) e md5sum (Gerador de hashes MD5 de arquivos) também compuseram o kit.

Deu-se então inicio ao processo de recolha de evidências como: arquivos de cache, histórico de navegação, cookies e arquivos temporários, tanto do IE quanto do Firefox, pela cópia direta dos mesmos nos caminhos onde os browsers os salvam.

Com relação as evidências coletadas do IE, ao final obteve-se um total de:

a) 10 arquivos de cache coletados do laboratório 8;

b) 10 arquivos de cache coletados do laboratório 16;

c) 1908 cookies coletados do laboratório 8;

d) 757 cookies coletados do laboratório 16;

e) 63 arquivos de histórico de navegação coletados do laboratório 8, e;

f) 44 arquivos de histórico de navegação coletados do laboratório 16.

Por sua vez, com relação as evidências coletadas do Firefox, ao final obteve-se um total de:

a) 105 arquivos .sqlite coletados do laboratório 8, e;

b) 114 arquivos .sqlite coletados do laboratório 16.

Pelo fato de ser um estudo de caso fictício, não se fez necessária a análise de todas as evidências coletadas, sendo que foram analisados:

a) todos os arquivos de cache do IE coletados, 20 no total (10 de cada laboratório);

b) 60 arquivos de cookie do IE coletados, 30 de cada laboratório;

c) todos os arquivos de histórico de navegação do IE coletados, 107 no total;

d) 110 arquivos .sqlite do Firefox coletados, 55 de cada laboratório.

Para a garantia da integridade, foi criado o hash de cada evidência encontrada usando o programa md5sum, bem como de cada cópia realizada, permitindo verificar que os arquivos analisados são cópias fiéis da evidências localizadas.

A Figura 2 abaixo, exemplifica o uso da ferramenta md5sum.

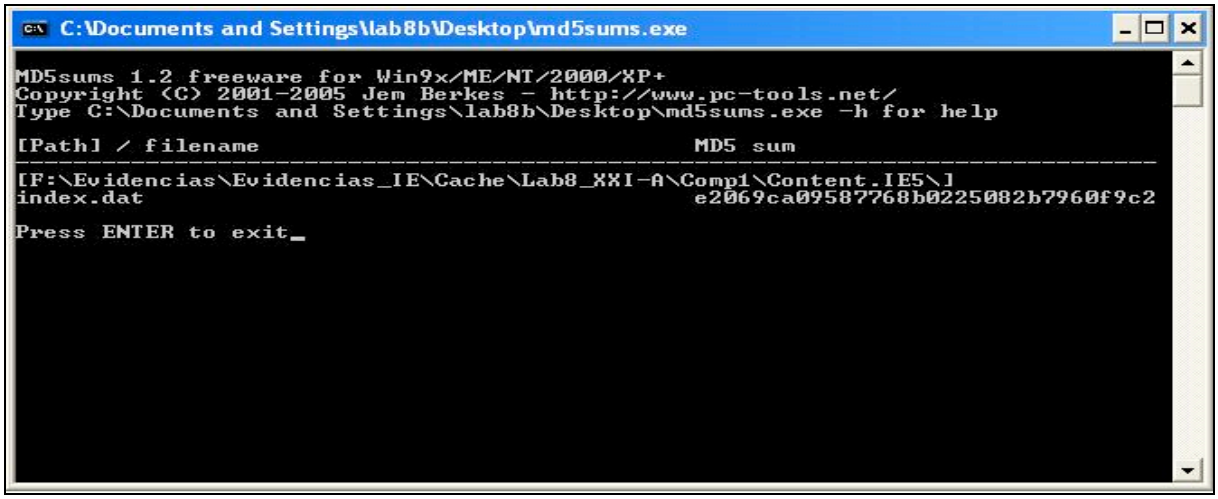
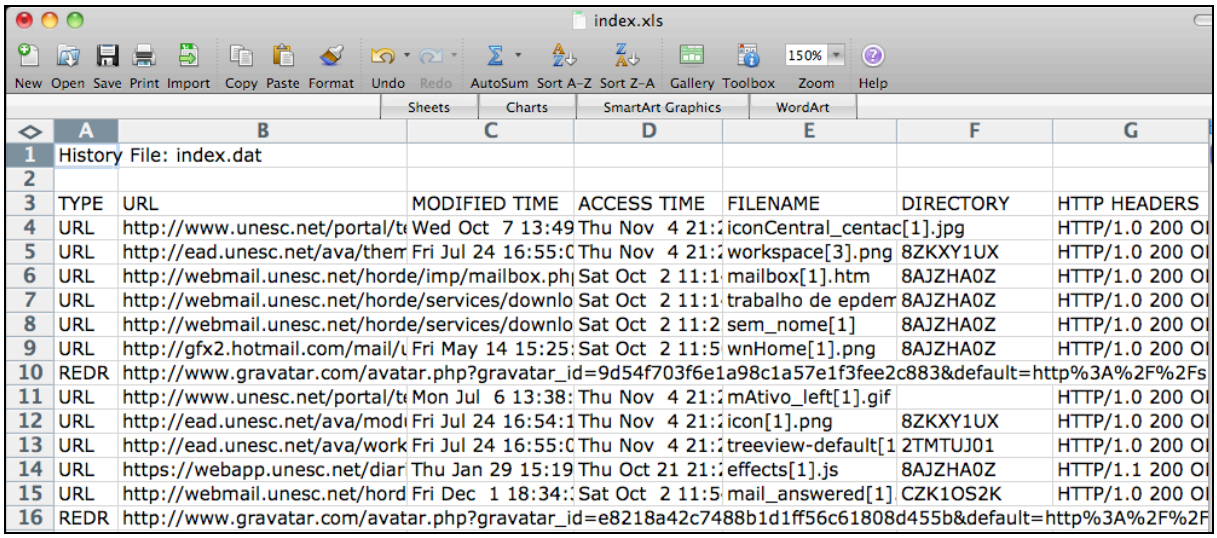


Figura 2. Ferramenta md5sum criando o *hash* de uma evidência

6.5 Etapa 4 – Exame e Análise

A etapa 4 contemplou o processo de conversão das evidências encontradas para formatos de fácil leitura, nomeadamente: xls (Ferramentas: Pasco, Galleta e Web Historian), csv e html (Ferramentas: F3E, Mozilla Cache View e PasswordFox); bem como a análise do conteúdo das evidências para obtenção de informações relevantes ao crime.

A Figura 3, abaixo, exemplifica um arquivo xls gerado para análise do perito.



index.xls

	A	B	C	D	E	F	G
1	History	File: index.dat					
2							
3	TYPE	URL	MODIFIED TIME	ACCESS TIME	FILENAME	DIRECTORY	HTTP HEADERS
4	URL	http://www.unesc.net/portal/t	Wed Oct 7 13:49	Thu Nov 4 21:2	iconCentral_centac[1].jpg		HTTP/1.0 200 OI
5	URL	http://ead.unesc.net/ava/them	Fri Jul 24 16:55:0	Thu Nov 4 21:2	workspace[3].png	8ZKXY1UX	HTTP/1.0 200 OI
6	URL	http://webmail.unesc.net/horde/imp/mailbox.ph	Sat Oct 2 11:1		mailbox[1].htm	8AJZHA0Z	HTTP/1.0 200 OI
7	URL	http://webmail.unesc.net/horde/services/downlo	Sat Oct 2 11:1		trabalho de epdem	8AJZHA0Z	HTTP/1.0 200 OI
8	URL	http://webmail.unesc.net/horde/services/downlo	Sat Oct 2 11:2		sem_nome[1]	8AJZHA0Z	HTTP/1.0 200 OI
9	URL	http://gfx2.hotmail.com/mail/u	Fri May 14 15:25:	Sat Oct 2 11:5	wnHome[1].png	8AJZHA0Z	HTTP/1.0 200 OI
10	REDR	http://www.gravatar.com/avatar.php?gravatar_id=9d54f703f6e1a98c1a57e1f3fee2c883&default=http%3A%2F%2Fs					
11	URL	http://www.unesc.net/portal/t	Mon Jul 6 13:38:	Thu Nov 4 21:2	mAtivo_left[1].gif		HTTP/1.0 200 OI
12	URL	http://ead.unesc.net/ava/mod	Fri Jul 24 16:54:1	Thu Nov 4 21:2	icon[1].png	8ZKXY1UX	HTTP/1.0 200 OI
13	URL	http://ead.unesc.net/ava/work	Fri Jul 24 16:55:0	Thu Nov 4 21:2	treeview-default[1	2TMTUJ01	HTTP/1.0 200 OI
14	URL	https://webapp.unesc.net/diar	Thu Jan 29 15:19	Thu Oct 21 21:2	effects[1].js	8AJZHA0Z	HTTP/1.1 200 OI
15	URL	http://webmail.unesc.net/hord	Fri Dec 1 18:34:3	Sat Oct 2 11:5	mail_answered[1]	CZK1OS2K	HTTP/1.0 200 OI
16	REDR	http://www.gravatar.com/avatar.php?gravatar_id=e8218a42c7488b1d1ff56c61808d455b&default=http%3A%2F%2F					

Figura 3. Exemplo de arquivo de cache convertido para o formato .xls

6.6 Etapa 5 – Documentação

Esta etapa foi recorrente durante topo o processo pericial, pois ela é crucial para que se possa depois redigir um laudo pericial fiel aos fatos.

6.7 Etapa 6 – Reconstrução da Cena do Crime

Na atual etapa, deve-se fazer a reconstrução dos eventos, juntando-se todas as evidências para que se possa determinar o que realmente ocorreu.

Como já foi mencionado, o presente estudo de caso simula que o sistema de vendas online de uma empresa de varejo fictícia foi acessado inadequadamente, e identificou-se que tal acesso partiu de um dos computadores dos laboratórios da UNESC. Realizou-se então a

condução de uma perícia forense computacional objetivando descobrir: quem?, onde?, quando? e como?, realizou tal ato.

Para que os propósitos da pesquisa se cumpram, supôs-se que o crime foi identificado pelo responsável técnico do sistema da empresa, no dia 11/11/10 às 20:46:12, e que nesse mesmo dia às 21:33:10 uma nova tentativa foi identificada.

Iniciando então a reconstrução da cena do crime, descobriu-se analisando as evidências do IE que:

a) ocorreram acessos ao site do sistema comprometido, partindo do computador 5 do laboratório 8, do Bloco XXI-B da UNESC, pois foram encontradas evidências nesse computador;

b) os acessos foram efetuados no dia 11/11/10, onde o primeiro acesso ocorreu às 20:46:12 e o último ocorreu às 21:22:45;

c) no horário em que o técnico identificou a primeira tentativa de ataque ao sistema, às 20:46:12, o usuário desse computador estava acessando o site em questão, usando o IE;

d) usuário logado no SO estava identificado como lab8b.

Por sua vez, analisando-se as evidências coletadas do Firefox, foi possível verificar que:

a) ocorreram acessos ao site do sistema comprometido, novamente partindo do computador 5 do laboratório 8, do Bloco XXI-B da UNESC, pois foram encontradas evidências nesse computador;

b) os acessos foram efetuados também no dia 11/11/10, onde o primeiro acesso ocorreu às 21:31:15 e o último às 21:36:46;

c) uma busca sobre “URLs relevantes do histórico de navegação” foi realizada às 21:31:15;

d) outra busca sobre “como invadir um servidor web” foi realizada às 21:31:40;

e) e mais uma busca contendo os termos “sobre ns”, foi realizada às 21:33:57;

f) às 21:34:07, foi digitado o nome Fulano_de_Tal no campo name de um formulário de contato, numa das páginas do site comprometido;

g) às 21:34:17, foi digitado fulano_de_tal, no campo username desse mesmo formulário;

h) às 21:35:10, foi digitado fulano_de_tal@gmail.com, no campo email do formulário;

i) às 21:35:29, foi digitado “Estudo de caso”, no campo subject do formulário;

j) às 21:36:46, foi digitado fulano_de_tal@gmail.com, no campo username de um formulário de login do site afetado;

k) no horário em que foi identificada a segunda tentativa de comprometimento do sistema, às 21:33:10, o usuário desse computador ainda estava acessando o site em questão, usando o Firefox.

Com as provas encontradas em mãos, procurou-se apurar mais pistas sobre o caso, e confirmou-se junto a instituição que no dia 11/11/10 o acadêmico esteve presente no laboratório 8 do Bloco XXI-B atendendo a uma aula, das 19:00 às 22:00. Confirmou-se também, junto aos seus colegas que no dia em questão o mesmo usou exclusivamente o computador 5 do laboratório, o mesmo onde foram encontradas as provas.

Concluiu-se ao final da pericia que:

a) Quem perpetrou o crime foi o acadêmico Fulano_de_Tal, estudante da universidade em causa;

b) Ele foi cometido a partir do computador 5, do laboratório 8 do Bloco XXI-B da universidade, e;

c) Foi efetuado no dia 11/11/10 às 20:46:12 e às 21:33:10;

d) Para executar o mesmo, o acadêmico registrou-se no site alvo, obteve acesso a página contento um formulário de contato da empresa, e tentou usando scripts burlar os mecanismos de segurança do formulário e enviar ao servidor código malicioso capaz de danificar o mesmo.

Para documentação dos procedimentos e registro das informações sobre a perícia, foi gerado um laudo pericial contendo tais informações.

3. Conclusão

Com o aumento crescente do número de usuários de computadores e utilizadores de Internet, aumenta proporcionalmente o número de crimes digitais cometidos. A pericia forense computacional, surge dentro desse contexto como uma arma eficaz, na luta contra a impunidade atualmente existente, contribuindo para se alcançar uma sociedade mais segura e justa. Por utilizar processos científicos na identificação e combate aos crimes digitais, cada vez mais os resultados das investigações são utilizados em processos judiciais, aumentando assim a necessidade de se ter processos otimizados de coleta, preservação e análise de evidências.

É importante mencionar que qualquer erro do perito durante a realização de uma investigação, pode invalidar a mesma, ou pior ainda, inocentar culpados ou culpar inocentes. Torna-se primordial que o perito conduza a pericia usando metodologias aprovadas e padronizadas, que certificarão aos órgãos judiciais a credibilidade da mesma.

Por existirem poucos estudos no País sobre pericia forense computacional, sobretudo com foco na coleta e análise de evidências em web browsers, o presente trabalho objetivou analisar e aplicar os procedimentos de perícia forense computacional na busca por evidências em tal ambiente.

Acredita-se que o objetivo geral foi atingido, pois foi possível analisar os procedimentos necessários a execução de uma perícia forense computacional e aplicá-los com sucesso em um estudo de caso fictício. No entanto, é importante mencionar que a aplicação prática comprovou que as ferramentas estudadas não são infalíveis, ocorrendo ocasionalmente erros na conversão das evidências coletadas, e que existem muitas outras ferramentas, técnicas e procedimentos, que dependendo do ambiente podem ser utilizadas.

Encerrando, o presente estudo acaba abrindo portas para trabalhos futuros, como a análise de ferramentas que permitam, por exemplo, executar uma pericia em várias máquinas conectadas em rede, ou que permitam recuperar os arquivos deletados dos browsers. Outra linha de pesquisa seriam as técnicas e ferramentas a usar na realização de uma perícia quando o suspeito utiliza a navegação privativa, opção onipresente nos browsers hoje em dia.

Referências

ARGOLO, Frederico Henrique Böhm. Análise Forense em sistemas GNU/Linux. Universidade Federal do Rio de Janeiro, 2005.

BERNARDO, Adauto de Sousa. Técnicas Computacionais no Auxílio à Perícia Forense na Análise de Evidências Coletadas em Servidores Gnu/Linux. Universidade do Extremo Sul Catarinense – Unesc, 2006.

BERTOGLIO, Daniel Dalalana. Perícia Forense: Proposta de Uma Metodologia de Coleta de Indícios Para Ambiente Windows. Trabalho de Conclusão de Curso - Curso de Ciência da Computação, Departamento de Instituto de Ciências Exatas e Tecnológicas, Centro Universitário Feevale, Novo Hamburgo, 2008.

CARRIER, Brian. File System Forensic Analysis. Indiana: Addison Wesley Professional, 2005.

CARUSO, Carlos A. A.; STEFFEN, Flávio Deny. Segurança em Informática e de Informações. 2a ed. São Paulo: Editora SENAC, 1999.

CASEY, Eoghan. Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet. Londres: Academic Press, 2004.

GALVÃO, Ricardo Kleber Martins. Perícia Forense em Web Browsers. Centro Federal de Educação Tecnológica do Rio Grande do Norte, 2008.

HEWITT, Peter C.; PELÁEZ, Manuel H. Santander. Browser Forensics. 2010. Disponível em: < http://www.browserforensics.org/wp-content/uploads/2010/03/BrowserForensics-v1-03-03-2010.pdf > Acesso em: 13 jun. 2010, 11:23:31.

JONES, Keith J.. Forensic Analysis of Internet Explorer Activity Files. 2003. Disponível em: < http://www.foundstone.com/us/pdf/wp_index_dat.pdf > Acesso em: 16 jun. 2010, 11:23:31.

PLACE, Ricardo Leocádio. Criptografia, assinatura digital e alguns outros conceitos. Disponível em: <http://eltiger.wordpress.com/2008/10/12/ciptografia-assinatura-digital-e-alguns-outros-conceitos/> Acesso em: 26 out. 2009, 09:28:57.

SWGDE. SCIENTIFIC WORKING GROUP ON DIGITAL EVIDENCE. Digital Evidence: Standards and Principles. 2008. Disponível em: <http://www.fbi.gov/hq/lab/fsc/backissu/april2000/swgde.htm> Acesso em: 12 nov. 2009, 15:10:45.

VACCA, John R. Computer Forensics: Computer Crime Scene Investigation.

ANEXO A – ART. 186, ART. 187 E ART. 927 DO CÓDIGO CIVIL BRASILEIRO

Art. 186. Aquele que, por ação ou omissão voluntária, negligência ou imprudência, violar direito e causar dano a outrem, ainda que exclusivamente moral, comete ato ilícito.

Art. 187. Também comete ato ilícito o titular de um direito que, ao exercê-lo, excede manifestamente os limites impostos pelo seu fim econômico ou social, pela boa-fé ou pelos bons costumes.

Art. 927. Aquele que, por ato ilícito (arts. 186 e 187), causar dano a outrem, fica obrigado a repará-lo.

Parágrafo único. Haverá obrigação de reparar o dano, independentemente de culpa, nos casos especificados em lei, ou quando a atividade normalmente desenvolvida pelo autor do dano implicar, por sua natureza, risco para os direitos de outrem.

ANEXO B – PROJETO DE LEI 84/99 DO CONGRESSO NACIONAL

Dispõe sobre os crimes cometidos na área de informática, suas penalidades e dá outras providências.

De autoria do Deputado Luiz Piauhylino.

O Congresso Nacional decreta:

CAPÍTULO I
DOS PRINCÍPIOS QUE REGULAM A PRESTAÇAO DE SERVIÇO POR REDES DE COMPUTADORES

Art. 1º - O acesso, o processamento e a disseminação de informações através das redes de computadores devem estar a serviço do cidadão e da sociedade, respeitados os critérios de garantia dos direitos individuais e coletivos e de privacidade e segurança de pessoas físicas e jurídicas e da garantia de acesso às informações disseminadas pelos serviços da rede.

Art. 2º - É livre a estruturação e o funcionamento das redes de computadores e seus serviços, ressalvadas as disposições específicas reguladas em lei.

CAPÍTULO II
DO USO DE INFORMAÇÕES DISPONÍVEIS EM COMPUTADORES OU REDES DE COMPUTADORES.

Art. 3º - Para fins desta lei, entende-se por informações privadas aquelas relativas a pessoa física ou jurídica identificada ou identificável.

Parágrafo único. É identificável a pessoa cuja individuação não envolva custos ou prazos desproporcionados.

Art. 4º - Ninguém será obrigado a fornecer informações sobre sua pessoa ou de terceiros, salvo nos casos previstos em lei.

Art. 5º - A coleta, o processamento e a distribuição, com finalidades comerciais, de informações privadas ficam sujeitas à prévia aquiescência da pessoa a que se referem, que poderá ser tomada sem efeito a qualquer momento, ressalvando-se o pagamento de indenizações a terceiros, quando couberem.

§ 1º. A toda pessoa cadastrada dar-se-á conhecimento das informações privadas armazenadas e das respectivas fontes.

§ 2º. Fica assegurado o direito à retificação de qualquer informação privada incorreta.

§ 3º. Salvo por disposição legal ou determinação judicial em contrário, nenhuma informação privada será mantida à revelia da pessoa a que se refere ou além do tempo previsto para a sua validade.

§ 4º. Qualquer pessoa, física ou jurídica, tem o direito de interpelar o proprietário de rede de computadores ou provedor de serviço para saber se mantém informações a seu respeito, e o respectivo teor.

Art. 6º - Os serviços de informações ou de acesso a bancos de dados não distribuirão

informações privadas referentes, direta ou indiretamente, a origem racial, opinião política, filosófica, religiosa ou de orientação sexual, e de filiação a qualquer entidade, pública ou privada, salvo autorização expressa do interessado.

Art. 7º - O acesso de terceiros, não autorizados pelos respectivos interessados, a informações privadas mantidas em redes de computadores dependerá de prévia autorização judicial.

CAPÍTULO III
DOS CRIMES DE INFORMÁTICA

Seção I
Dano a dado ou programa de computador

Art. 8º - Apagar, destruir, modificar ou de qualquer forma inutilizar, total ou parcialmente, dado ou programa de computador, de forma indevida ou não autorizada.

Pena: detenção, de um a três anos e multa.

Parágrafo único. Se o crime é cometido:

I - contra o interesse da União, Estado, Distrito Federal, Município, órgão ou entidade da administração direta ou indireta ou de empresa concessionária de serviços públicos;
II- com considerável prejuízo para a vítima;
III - com intuito de lucro ou vantagem de qualquer espécie, própria ou de terceiro;
IV - com abuso de confiança;
V - por motivo fútil;
VI - com o uso indevido de senha ou processo de identificação de terceiro , ou
VII - com a utilização de qualquer outro meio fraudulento.
Pena: detenção, de dois a quatro anos e multa.

Seção II
Acesso indevido ou não autorizado

Art. 9º Obter acesso, indevido ou não autorizado, a computador ou rede de computadores.

Pena: detenção, de seis meses a um ano e multa.

Parágrafo primeiro. Na mesma pena incorre quem. sem autorização ou indevidamente, obtém, mantém ou fornece a terceiro qualquer meio de identificação ou acesso a computador ou rede de computadores.

Parágrafo segundo. Se o crime é cometido:

I - com acesso a computador ou rede de computadores da União, Estado, Distrito Federal, Município, órgão ou entidade da administração direta ou indireta ou de empresa concessionária de serviços públicos;
II - com considerável prejuízo para a vítima;
III - com intuito de lucro ou vantagem de qualquer espécie, própria ou de terceiro;
IV - com abuso de confiança;
V - por motivo fútil;

VI - com o uso indevido de senha ou processo de identificação de terceiro; ou
VII - com a utilização de qualquer outro meio fraudulento.

Pena: detenção, de um a dois anos e multa.

Seção III
Alteração de senha ou mecanismo de acesso a programa de computador ou dados

Art. 10. Apagar, destruir, alterar, ou de qualquer fama inutilizar, senha ou qualquer outro mecanismo de acesso a computador, programa de computador ou dados, de forma indevida ou não autorizada.

Pena: detenção, de um a dois anos e multa.

Seção IV
Obtenção indevida ou não autorizada de dado ou instrução de computador

Art. 11. Obter, manter ou fornecer, sem autorização ou indevidamente, dado ou instrução de computador.
Pena: detenção, de três meses a um ano e multa.

Parágrafo Único. Se o crime é cometido:

I - com acesso a computador ou rede de computadores da União, Estado, Distrito Federal, Município, órgão ou entidade da administração direta ou indireta ou de empresa concessionária de serviços públicos;
II - com considerável prejuízo para a vítima;
Parágrafo Único. Se o crime é cometido:
I - com acesso a computador ou rede de computadores da União, Estado, Distrito Federal, Município, órgão ou entidade da administração direta ou indireta ou de empresa concessionária de serviços públicos;
II - com considerável prejuízo para a vítima;
III - com intuito de lucro ou vantagem de qualquer espécie, própria ou de terceiro;
IV - com abuso de confiança;
V - por motivo fútil;
VI - com o uso indevido de senha ou processo de identificação de terceiro; ou
VII - com a utilização de qualquer outro meio fraudulento.

Pena: detenção, de um a dois anos e multa.

Seção V
Violação de segredo armazenado em computador, meio magnético, de natureza magnética, óptica ou similar

Art. 12. Obter segredos, de indústria ou comércio, ou informações pessoais armazenadas em computador, rede de computadores, meio eletrônico de natureza magnética, óptica ou similar, de forma indevida ou não autorizada.

Pena: detenção, de um a três anos e multa.

Seção VI
Criação, desenvolvimento ou inserção em computador de dados ou programa de computador c nocivos

Art. 13. Criar, desenvolver ou inserir, dado ou programa em computador ou rede de computadores, de forma indevida ou não autorizada com a finalidade de apagar, destruir, inutilizar ou modificar dado ou programa de computador ou de qualquer forma dificultar ou impossibilitar, total ou parcialmente, a utilização de computador ou rede de computadores.

Pena: reclusão, de um a quatro anos e multa.

Parágrafo único. Se o crime é cometido:

I - contra a interesse da União, Estado, Distrito Federal. Município, órgão ou entidade da administração direta ou indireta ou de empresa concessionária de serviços públicos;
II - com considerável prejuízo para a vítima;
III - com intuito de lucro ou vantagem de qualquer espécie, própria ou de terceiro;
IV - com abuso de confiança;
V - por motivo fútil;
VI - com o uso indevid6 de senha ou processo de Identificação de terceiro; ou
VII - com a utilização de qualquer outro meto fraudulento.

Pena: reclusão, de dois a seis anos e multa.

Seção VII
Veiculação de pornografia através de rede de computadores

Art. 14. Oferecer serviço ou informação de caráter pornográfico, em rede de computadores, sem exibir, previamente, de forma facilmente visível e destacada, aviso sobre sua natureza, indicando o seu conteúdo e a inadequação para criança ou adolescentes.

Pena: detenção, de um a três anos e multa.

CAPITULO IV
DAS DISPOSIÇÕES FINAIS

Art. 15. Se qualquer dos crimes previstos nesta lei é praticado no exercício de atividade profissional ou funcional, a pena é aumentada de um sexto até a metade.

Art. 16. Nos crimes definidos nesta lei somente se procede mediante representação do ofendido, salvo se cometidos contra o interesse da União, Estado, Distrito Federal Município, órgão ou entidade da administração direta ou indireta, empresa concessionária de serviços públicos, fundações instituídas ou mantidas pelo poder público, serviços sociais autônomos, instituições financeiras ou empresas que explorem ramo de atividade controlada pelo poder público, casos em que a ação é pública incondicionada.

Art. 17. Esta lei regula os crimes relativos à informática sem prejuízo das demais comunicações previstas em outros diplomas legais.

Art 18. Esta lei entra em vigor 30 (trinta) dias a contar da data de sua publicação.