UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

PAULA PORFÍRIO TEIXEIRA

ANÁLISE E COMPARAÇÃO DE SOFTWARES PARA PERÍCIA FORENSE EM AMBIENTES LINUX BASEADO NA MÉTRICA *FUNCTION POINT ANALYSIS*

CRICIÚMA, JUNHO DE 2010

PAULA PORFÍRIO TEIXEIRA

ANÁLISE E COMPARAÇÃO DE SOFTWARES PARA PERÍCIA FORENSE EM AMBIENTES LINUX BASEADO NA MÉTRICA *FUNCTION POINT ANALYSIS*

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, JUNHO DE 2010

PAULA PORFÍRIO TEIXEIRA

ANÁLISE E COMPARAÇÃO DE SOFTWARES PARA PERÍCIA FORENSE EM AMBIENTES LINUX BASEADO NA MÉTRICA *FUNCTION POINT ANALYSIS*

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. Esp. Fabrício Giordani (UNESC)

Prof. Esp. Sérgio Coral (UNESC)

Dedico este trabalho ao meu esposo Diego Pereira do Nascimento, que me ajudou de tantas formas que eu não conseguiria enumerá-las.

AGRADECIMENTOS

Agradeço, primeiramente ao meu orientador, Paulo Martins, pelas idéias e orientação recebidas, tão importantes do início ao fim. Agradeço também a professora Merisandra pelo auxílio na confecção deste trabalho, sem o qual talvez fosse impossível concluí-lo. Agradeço aos meus pais e irmã pelo apoio incondicional. Ao meu esposo pela dedicação e apoio, independente da situação. E aos meus amigos e colegas de classe, com quem partilhei, durante estes três semestres todos os momentos da minha vida acadêmica, e que, direta ou indiretamente, contribuíram para que eu pudesse seguir adiante.

“Algo só é impossível até que alguém duvide e prove o contrário.” (A. Einstein)

RESUMO

A perícia forense computacional é uma área que necessita evoluir na mesma velocidade que surgem novos recursos tecnológicos, pois é nesta mesma velocidade que surgem novos tipos de crimes digitais, uma vez que os criminosos digitais se utilizam destes mesmos recursos para a sua atuação ilícita. Portanto, visando contribuir com o seu crescimento, o presente trabalho apresentará uma análise comparativa entre duas ferramentas voltadas a esta área. A análise foi realizada em duas etapas: a primeira análise foi baseada na métrica de software Pontos de Função, que tem por objetivo medir a qualidade do software de uma aplicação, traduzindo em valores numéricos. A segunda análise foi realizada fazendo um levantamento dos recursos que cada ferramenta disponibiliza, atribuindo pontos a cada recurso identificado. Ao final destas duas análises foi realizada a comparação entre os resultados, sendo possível identificar qual das duas ferramentas possui uma maior quantidade de recursos e a maior qualidade destes recursos, baseando-se na métrica FPA.

Palavras chave: Segurança da informação, perícia forense, métricas de *software*.

ABSTRACT

The forensic computing is an area that needs to evolve at the same speed that new technological resources appear, therefore it is in this same speed that appears new types of digital crimes, because criminals are using these same digital resources for their illegal actions. Therefore, aiming to contribute to its growth, this paper presents a comparative analysis between two tools devoted to this area. The analysis was performed in two stages: the first analysis was based on software metrics Function Points Analysis, which aims to measure the quality of a software application, translating into numeric values. The second analysis was conducted by taking a survey of resources that each tool provides, assigning points to each resource identified. At the end of these two analyses a comparison between the results was conducted, making it possible to identify which of these two tools possess a superior amount of resources and the best quality of these resources, being based on the metrics FPA.

Keywords: Information security, forensic analysis, software metrics.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AIE	Arquivos de Interface Externa
ALI	Arquivos Lógicos Internos
AR	Arquivos Referenciados
ASCI	*American Standard Code for Information*
BSD	*Berkeley Software Distribution*
FFS	*Flash File System*
CD	*Compact Disk*
CDFS	*Compact Disk File System*
CE	Consulta Externa
CERT	*Computer Emergency Response Team*
CGS	Características Gerais do Sistema
CNPJ	Cadastro Nacional de Pessoa Jurídica
CPU	*Central Processing Unit*
CRC32	*Cyclic Redundancy Check (32 bit)*
DAT	*Digital Archive Tape*
DD	*Data Dumper*
DoS	*Denial Of Service*
DVD	*Digital Video Disc*
EDD	Elementos de Dados
EE	Entrada Externa
ER	Elementos de Registro
EXT	*Extended Filesystem*
FAT	*File Allocation Table*

FDTK	Forense Digital *ToolKit*
FIRE	*Forensic Incident Response Environment*
FPA	*Function Point Analysis*
FTK	*Forensic Toolkit*
GB	*Gigabytes*
GNU	*Gnu's Not Unix*
GZip	*GNU Zip*
HFS	*Hierarchical File System*
HTML	*HyperText Markup Language*
IBM	*International Business Machines Corporation*
IFPUG	*International Function Point Users Group*
IMG	*Image*
ISO	*International Organization for Standardization*
JPG	*Join Photographic expert Group*
LINUX	*Linux Is Not Unix*
MD5	*Message Digest 5 Algorithm*
MP3	*Moving Picture Experts Group Layer-3 Audio*
NI	Nível de Influência
NIT	Nível de Influência Total
NTFS	*New Technology File System*
PC	*Personal Computer*
PFNA	Pontos de Função Não Ajustados
RAID	*Redundant Array of Inexpensive Disks*
SMART	*Storage Media Analysis Recovery Toolkit*
SE	Saída Externa

SHA1	*Secure Hash Algorithm Version 1.0*
TCT	*The Coroner's Toolkit*
TSK	*The Sleuth Kit*
UFS	*Unix File System*
UNIX	*Uniplexed Information and Computing System*
USB	*Universal Serial Bus*
VFA	Valor do Fator de Ajuste

SUMÁRIO

INTRODUÇÃO

Vivemos em uma época em que a informação digital está cada vez mais acessível. As opções de conectividade são inúmeras. É possível acessar a Internet de qualquer computador, sendo este de posse do usuário ou não, de um telefone celular ou qualquer outro dispositivo móvel que possua conexão com a rede (OLIVEIRA, 2007).

No âmbito empresarial, o uso da informação digital se tornou imprescindível para o funcionamento de qualquer organização, seja para divulgação ou para o dia-a-dia. Setores precisam comunicar-se dentro da empresa e o meio digital é uma forma rápida e segura para troca de informações (BERALDI; ESCRIVÃO FILHO, 2000).

No aspecto pessoal, a Internet disponibilizou aos usuários a possibilidade de comunicação com pessoas de outras localidades e acesso à qualquer *site* de qualquer país, com apenas alguns cliques.

Porém, com o aumento de usuários e a facilidade de acesso ao ambiente virtual, houve também a criação de uma nova modalidade de crimes, os chamados “crimes digitais”. Estes crimes são dos mais variados: invasão de bancos de dados, de conteúdo de um *site*, de contas bancárias; e por motivos diversos: desvio de dinheiro, roubo de informações, destruição de dados ou apenas para “deixar a sua marca” (TREVENZOLI, 2006). Estes chamados “criminosos digitais” possuem à sua disposição inúmeras possibilidades para realizar seus crimes. E com as facilidades de acesso que existem hoje em dia, os crimes digitais estão se tornando cada vez mais frequentes.

Em alguns casos onde há a suspeita de invasão, o computador em questão ainda não foi investigado, nem mesmo desligado. Nestes casos é necessário que sejam tomados cuidados em relação ao manuseio do equipamento, pois informações valiosas podem ser

perdidas apenas desligando o computador (BERNARDO, 2006). E para tal, é necessário aplicar técnicas de perícia forense (FREITAS, 2006).

Perícia forense computacional consiste na aplicação de conhecimentos em informática e técnicas de investigação com a finalidade de obtenção de evidências de um crime digital (ANDRADE, 2005).

A análise forense computacional é uma área relativamente nova dentro da legislação brasileira. Existem conjuntos de ferramentas criados exclusivamente para essa prática, porém ainda existe uma carência em relação à normas e padrões relativos à investigação computacional (FREITAS, 2006). Em compensação, possuímos à disposição vários softwares de análise forense computacional, tanto livres quanto proprietários. Mas por justamente possuirmos várias opções de escolha em relação a qual *software* utilizar, acaba-se por ter certa dificuldade na escolha dentre tantas opções.

Sendo assim, este trabalho propõe um estudo comparativo entre dois *softwares* de análise forense disponíveis atualmente, que sejam voltados para o ambiente Linux.

1.1 OBJETIVO GERAL

Comparar softwares para perícia forense voltados ao ambiente Linux baseado em métricas de *software*.

1.2 OBJETIVOS ESPECÍFICOS

O presente trabalho foi desenvolvido buscando atingir os seguintes objetivos:

a) Descrição e definição das métricas de *software* para a comparação entre os *softwares* de análise forense;

b) Delimitação do ambiente de testes;

c) Realização de testes com os *softwares* de perícia forense escolhidos;

d) Análise dos resultados gerados após a aplicação das métricas de *software*.

1.3 JUSTIFICATIVA

Com a crescente popularização da Internet e dos meios de acessibilidade à mesma, criou-se uma nova modalidade de crimes, os crimes digitais. São crimes em que, na maioria das vezes, as provas estão apenas em formato digital.

Crime digital pode ser classificado como todo e qualquer ato que cause dano à terceiros, utilizando como principal meio de atuação o computador. Cabem nesta classificação invasão de contas bancárias e desvio de dinheiro, roubo de informações confidenciais, publicação de material de caráter ofensivo, como pedofilia, racismo entre outros (SILVA, 2003).

Baseado no Princípio da Troca de Locard[1], tudo que entra no local do crime leva consigo algo e deixa algo para trás. No mundo digital é possível aplicar esse conceito com algumas alterações: independente de onde o intruso esteve, ele deixa rastros (OLIVEIRA, 2007). Dentro da perícia forense, estes "rastros" são chamados de evidências digitais. Em alguns casos a evidência pode ser extremamente difícil de ser rastreada, mas ela existe de fato (STEPHENSON, 2000).

Para conseguir examinar estas evidências, é necessária uma investigação especial, utilizando softwares específicos, com o propósito de resgatar informações de determinados dispositivos e registrá-las para comprovar ter havido um crime ou não. Para tanto há a necessidade de prévios conhecimentos na área de análise forense computacional.

Existem disponíveis vários softwares voltados para a área de perícia forense, sendo que cada um possui suas particularidades, como por exemplo, possuir uma interface gráfica, realizar análise das propriedades dos arquivos, examinar os arquivos do disco rígido à procura de indícios de acesso não autorizado, procurar arquivos ocultos, listar arquivos e atributos de segurança, identificar o tipo de um arquivo por meio de seu conteúdo entre outras características (GEUS; REIS, 2002).

Em virtude aos fatos mencionados, este trabalho propõe um estudo comparativo de dois *softwares* de análise forense computacional voltados ao ambiente Linux. Serão realizados testes e comparações dos resultados, baseados em métricas que serão definidas ao longo do trabalho, a fim de analisar os recursos que cada ferramenta disponibiliza e avaliar a qualidade destes recursos, com o intuito de auxiliar na escolha pelo software que atenda melhor as necessidades do profissional de perícia forense computacional.

[1] A troca de Locard define que quando dois objetos entram em contato sempre haverá troca de material entre os eles. Dentro da análise forense, este conceito indica que a contaminação de provas pode ocorrer, não somente pelo criminoso, mas também pelo perito no momento da investigação (OLIVEIRA, 2007).

1.4 ESTRUTURA DO TRABALHO

A estrutura deste trabalho está organizado da seguinte forma: no Capítulo 2 serão discutidos conceitos de segurança da informação, desde os tempos da lendária Biblioteca de Alexandria até os dias de hoje. Também serão abordados conceitos de crimes digitais, evidências digitais e a posição atual da legislação brasileira em relação aos crimes envolvendo computadores.

No Capítulo 3 será abordado o conceito de perícia forense computacional. Serão apresentadas as etapas de uma investigação computacional, detalhando as particularidades de cada uma e a sua importância dentro do ciclo da perícia forense, que também será discutido neste capítulo. Também serão apresentadas algumas das ferramentas para perícia forense disponíveis no momento, tanto livres quanto proprietárias.

No Capítulo 4 serão apresentadas as métricas de *softwares* utilizadas para a confecção deste trabalho, bem como o conceito de métricas de *software*.

No Capítulo 5 será apresentado o ambiente de testes: o sistema operacional escolhido e as ferramentas a serem analisadas. Será feita uma breve apresentação da utilização das ferramentas.

No Capítulo 6 será realizado o desenvolvimento do trabalho em si: análise comparativa entre as ferramentas escolhidas utilizando as métricas de software citadas no Capítulo 4. Neste mesmo capítulo serão apresentados os resultados obtidos após a comparação entre os *softwares*.

2 SEGURANÇA DA INFORMAÇÃO

Desde tempos remotos observa-se a necessidade do homem em manter registros da sua vida (SILVA, 2003). Desenhos nas cavernas, entalhos em madeira e pedra, pergaminhos, livros, *sites*, *blogs*, a humanidade sempre buscou formas de guardar informações para gerações futuras.

O homem, até os dias de hoje, sempre buscou maneiras de aumentar cada vez mais seus bens. Isto se aplica também ao conhecimento. Filósofos, matemáticos, físicos são citados freqüentemente em toda a história, pois eram homens da ciência, estudiosos, que tinham em suas mãos o poder do conhecimento. E tentando ter controle sobre este poder, homens como Júlio César mantinham sempre um grupo de estudiosos aos seus serviços. Cobiçavam o conhecimento, mas o adquiriam por meio da força.

Com o passar do tempo é possível observar que a necessidade de obter e registrar as informações não eram a única preocupação. Não era preciso somente guardá-las, era necessário também mantê-las à salvo. Casos como o incêndio da lendária Biblioteca de Alexandria trouxeram à tona a questão da segurança da informação (SOUZA, 2005). Inúmeras técnicas foram desenvolvidas para este propósito, tais como guardas vigiando o local onde estão armazenados os registros, ou coisas mais atuais como cofres e *firewalls*, com o único intuito de proteger as informações, que hoje são considerados patrimônio.

Segurança da informação se aplica à todos os aspectos de proteção e armazenamento de informações e dados, em qualquer forma. É caracterizada pelo estudo de técnicas que visam fornecer um estado de proteção ao patrimônio computacional e intelectual de uma organização (OLIVEIRA, 2007).

Focando a segurança da informação dentro da computação, a situação atual é preocupante. A eliminação de fronteiras físicas oferecidas pela Internet gerou, por um lado,

desde a comodidade de realizar compras sem sair de casa, até conhecer lugares do outro lado do globo, porém, por outro lado, possibilitou o nascimento de uma nova modalidade de crimes, os crimes digitais. Pessoas de má índole viram no meio virtual uma forma cômoda de praticar ações criminosas, apoiados no anonimato quase absoluto que a Internet proporciona, colocando mais uma vez em risco a segurança da informação.

2.1 CRIMES DIGITAIS E EVIDÊNCIAS DIGITAIS

Dentro da abordagem do Direito Penal, para que um crime seja classificado como crime digital é necessário um meio, no caso qualquer dispositivo que consiga acessar a Internet. Em sua grande maioria podem ser enquadrados na categoria Estelionato / Extorsão, Falsidade Ideológica e Fraudes, encontrando assim embasamento dentro do Código Penal (ANDRADE, 2005).

2.1.1 Crimes Digitais

Podem ser classificados como crimes digitais, ou crimes informáticos, toda e qualquer conduta ilegal, lesiva, dolosa cometida contra o sistema informático ou por meio deste (SILVA, 2003). Cabem neste conceito ações que causam danos a terceiros, via rede, como por exemplo, acesso não autorizado a informações confidenciais, fraudes bancárias, publicação de material de caráter ofensivo, como pedofilia, racismo entre outros.

Um dos crimes virtuais mais comuns é o Estelionato, citado no artigo 171 do decreto-lei nº. 2848 do Código Penal Brasileiro: “Obter, para si ou para outrem, vantagem ilícita, em prejuízo alheio, induzindo ou mantendo alguém em erro, mediante artifício, ardil, ou qualquer outro meio fraudulento” (BRASIL, 1940).

Este tipo de crime pode ser exemplificado pelos clássicos *e-mails* onde o fraudador induz o usuário a visitar um endereço na Internet, sob diversos pretextos. Na maioria das vezes, o endereço visitado contém códigos maliciosos[2] que visam, basicamente, coletar informações confidenciais do usuário, como por exemplo, senhas de *Internet Banking*. No Anexo A são citados outros decretos-lei do Código Penal Brasil e da Constituição Federal que podem ser interpretados pelo poder judiciário para punir criminosos digitais.

Baseando-se na facilidade de obtenção de informações confidenciais dentro do meio digital, é possível para um criminoso atuar no anonimato, pois, dentro de uma residência qualquer, o mesmo pode realizar uma invasão à um banco de dados, sozinho ou com a ajuda de outros, pode também se apoderar de informações sigilosas e vendê-las, fazer transferências bancárias de contas de terceiros para sua própria e uma série de outras ações ilícitas (TREVENZOLI, 2006).

2.1.2 Tipos comuns de crimes digitais

Como descrito anteriormente, os crimes digitais são realizados por meio do uso do computador, e normalmente por meio de códigos maliciosos, os *malware's*, que o usuário, por várias razões, acaba permitindo a instalação em seu equipamento.

Em quase sua totalidade, essa permissão por parte do usuário acontece de forma despercebida: o código vem em forma de um *e-mail* despretensioso ou de um arquivo que algum site solicitou a instalação, acabando assim por abrir uma porta por onde o invasor consegue acessar a máquina infectada, conseguindo coletar informações confidenciais do usuário.

Abaixo seguem exemplos de *malware's* mais comuns:

[2] "Código malicioso ou *Malware* (***Mal**icious Soft**ware***) é um termo genérico que abrange todos os tipos de programa especificamente desenvolvidos para executar ações maliciosas em um computador." (CERT.br, 2010)

a) Vírus: código que pode tanto atuar independentemente ou alojar-se em outros programas executáveis, infectando-os. Para que o vírus seja executado é necessária a atuação do usuário: é preciso que seja executado por meio de um arquivo que foi infectado pelo vírus ou o arquivo hospedeiro. Mesmo que o usuário não esteja ciente desta permissão, ela pode acontecer se um cliente de *e-mail*, por exemplo, estiver executando. O programa solicita a execução do arquivo infectado e "confia" no mesmo, propagando assim o código malicioso (CERT.br, 2010);

b) *Trojan*: os *Trojans*, ou Cavalos de Tróia, atuam exatamente como o caso da famosa história do cavalo de madeira dado aos troianos: um código se instala no computador do usuário, disfarçando-se de algum aplicativo que o próprio usuário instalou, acabando por abrir uma brecha para que criminosos consigam acessar o sistema remotamente. Normalmente utilizado para que outros *malware's* se instalem no computador (CERT.br, 2010);

c) *Keyloggers*: tipo de *malware* com o funcionamento parecido com o *Trojan*, Ele se instala na máquina disfarçando-se de uma aplicação qualquer, normalmente com o consentimento do usuário, pois este pensa estar instalando um aplicativo específico, que ele mesmo tenha escolhido. Porém seu funcionamento se resume em coletar tudo o que é digitado na máquina, assim o invasor consegue captar informações pessoais, principalmente como senhas de banco e *e-mail* (MELO, 2009);

d) *Worm*: *malware* desenhado para se propagar automaticamente por meio de redes, enviando a si mesmo para outros computadores da mesma rede. Este tipo de *malware* consegue explorar vulnerabilidades de configuração da rede, tais

como *softwares* de gerenciamento de rede, e acaba por criar um cenário onde não necessita ser executado, explicitamente (CERT.br, 2010);

e) *Bot*: *malware* parecido com o *worm* em seu funcionamento, porém ele consegue se comunicar com o invasor, permitindo que seja controlado remotamente. Esta característica permite ao invasor infectar um grande número de computadores, dependendo do tamanho da rede, criando assim as *botnets*. As *botnets*, como são controladas remotamente pelo invasor, podem trabalhar em conjunto para vários fins escusos, como por exemplo, atacar um servidor *web* em conjunto, apagando assim o rastro do invasor, uma vez que foram as máquinas infectadas que, diretamente, realizaram o ataque (MELO, 2009);

f) *Backdoor*: código malicioso que possibilita ao atacante o retorno à máquina infectada. Em sua maioria o atacante apaga seus rastros para que não seja descoberto, pois com o *backdoor* a intenção é retornar ao sistema invadido (CERT.br, 2010);

g) *web*: um caso particular de ataque visando especificamente o comprometimento de servidores *web* ou desfigurações de páginas na Internet (CERT.br, 2010);

h) *scan*: notificações de varreduras em redes de computadores, com o intuito de identificar quais computadores estão ativos e quais serviços estão sendo disponibilizados por eles. É amplamente utilizado por atacantes para identificar potenciais alvos, pois permite associar possíveis vulnerabilidades aos serviços habilitados em um computador (CERT.br, 2010);

i) *dos* (DoS - *Denial of Service*): notificações de ataques de negação de serviço, onde o atacante utiliza um computador ou um conjunto de computadores para tirar de operação um serviço, computador ou rede (CERT.br, 2010).

2.2 EVIDÊNCIAS DIGITAIS

Apesar da atuação ilícita usando como instrumento o computador oferecer a facilidade e o conforto do anonimato, este não é completo. Baseando-se no *Princípio da Troca de Locard*, tudo que entra no local do crime leva consigo algo e deixa algo para trás. No mundo digital é possível aplicar esse conceito com algumas alterações: independente de onde o intruso esteve ele deixa rastros (OLIVEIRA, 2007).

Registros de acesso ao sistema, informações temporárias armazenadas em registradores[3] e outras informações que registram ações específicas dentro do sistema operacional, podem conter informações sobre o criminoso e, posteriormente, usadas como provas contra o mesmo, assumindo a denominação de evidências digitais. Tais evidências devem ser coletadas com prudência, afim de não danificá-las, pois, tal ação poderia invalidá-las em um processo judicial (TREVENZOLI, 2006).

Ainda não existem normas a serem seguidas durante a coleta de evidências, porém é necessário que o perito conheça dois artigos descritos no Código de Processo Penal, no Capítulo II, intitulado “Do Exame do Corpo de Delito, e das Perícias em Geral”, evitando que as evidências coletadas sejam consideradas ilegais:

a) O artigo 170 do decreto-lei nº. 3689 do Código de Processo Penal diz:

> Nas perícias de laboratório, os peritos guardarão material suficiente para a eventualidade de nova perícia. Sempre que conveniente, os laudos serão ilustrados com provas fotográficas, ou microfotográficas, desenhos ou esquemas. (BRASIL, 1941)

b) O artigo 171 do decreto-lei nº. 3689 do Código de Processo Penal diz:

> Nos crimes cometidos com destruição ou rompimento de obstáculo a subtração da coisa, ou por meio de escalada, os peritos, além de descrever os vestígios, indicarão com que instrumentos, por que meios e em que época presumem ter sido o fato praticado. (BRASIL, 1941)

[3] Registradores: estruturas físicas do computador que armazenam informações em forma de um conjunto pré-definido de *bits* (OLIVEIRA, 2006).

Evidências digitais têm um alto grau de volatilidade, podendo tornar a coleta de evidências particularmente difícil. Por este motivo, a fase de coleta de evidências deve ser realizada de uma maneira metódica e organizada, a fim de garantir que as informações obtidas existem nas evidências coletadas e que não foram alteradas ou contaminadas, uma vez que o simples fato de ligar ou desligar o computador pode alterar ou até destruir evidências (BERNARDO, 2006).

2.3 INCIDENTE DE SEGURANÇA

Um incidente de segurança pode ser classificado como qualquer evento adverso, confirmado ou sob suspeita, que possa ameaçar a segurança da informação dos sistemas ou das redes de computadores. Exemplos práticos de incidentes são: negação de serviço, violação da política de segurança, tentativas para ganhar acesso não autorizado, contaminação por vírus, exploração de dispositivos e serviços por meio de códigos maliciosos, entre outros (GUIMARÃES, 2005).

Para uma identificação e tratamento adequados a um incidente de segurança é necessário que seja estabelecida uma política de segurança. De uma forma mais simples, significa estabelecer as ações que podem ser consideradas incidentes de segurança, tais como os citados no parágrafo acima. Esta rotina também pode ser aplicada a usuários domésticos: o usuário pode definir quais atos podem ser classificados como um incidente e estar atento a qualquer comportamento estranho, a fim de quando ocorrer algum evento de natureza duvidosa o usuário possa tomar as devidas providências.

O foco do presente estudo é o processo de investigação em si, independente da origem do incidente de segurança. Ou seja, o processo de análise forense a ser estudado neste

trabalho será o mesmo, independente se o computador a ser analisado fizer parte de uma rede corporativa ou é utilizado apenas para o acesso doméstico.

3 PERÍCIA FORENSE COMPUTACIONAL

Muitos acreditam que o ramo de investigação criminal voltado à informática, neste trabalho tratado como Perícia Forense Computacional, é recente dentro da área da computação, porém é mais antigo do que se imagina. No dia 18 de outubro de 1966, o *Minneapolis Tribune*, jornal do estado de Minessota, EUA, publicou um artigo com o título "Perito de computador acusado de falsificar seu saldo bancário" (PARKER, 1990). Observa-se que desde esta data já existiam os crimes digitais, porém em pouca freqüência devido ao alto custo dos computadores.

A perícia forense computacional consiste, basicamente, em um conjunto de procedimentos que, baseando-se na legislação vigente, tem por objetivo coletar evidências de equipamentos de armazenamento de dados, de forma que possam ser apresentados em juízo como provas coerentes e significativas (GEUS, 2001). Compreende quatro etapas básicas: aquisição, identificação, avaliação e apresentação de evidências, formando o Ciclo de Perícia Forense, quaisquer que sejam as formas de armazenamento das mídias analisadas, com o objetivo de coletar provas que permitam a formulação de conclusões referentes ao caso investigado (GEUS; REIS, 2002; MELO, 2009).

A Figura 1 demonstra como as etapas do ciclo de Perícia Forense, descrito acima, se entremeiam.

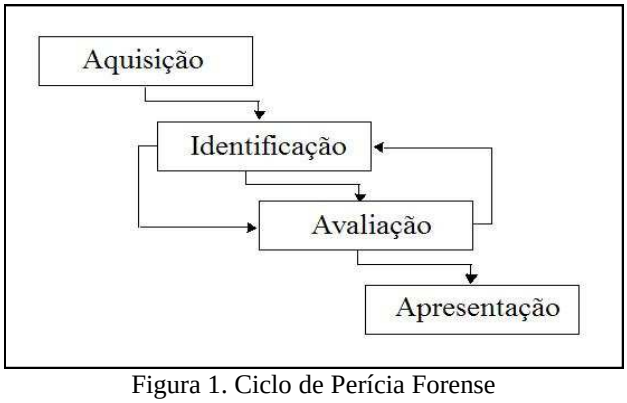

Figura 1. Ciclo de Perícia Forense
Fonte: MELO, S. (2009, p. 15)

Quando o perito é chamado para investigar uma ocorrência e o equipamento a ser investigado ainda não foi desligado, é possível fazer uma investigação forense *in vivo,* ou *live analysis*, que consiste em investigar o computador em busca de informações voláteis, que se perderiam com o desligamento do equipamento (MELO, 2009). Após a *live analysis* e a completa cópia do sistema (informações de registradores, *cache*[4], memória de periféricos, entre outras), o perito realiza o desligamento adequado dos dispositivos (COSTA, 2003).

3.1 INVESTIGAÇÃO FORENSE *IN VIVO* (*LIVE ANALYSIS*)

Para que o perito consiga conduzir uma investigação *in vivo* sem danificar possíveis evidências é aconselhável que o mesmo utilize um *live-CD* contendo ferramentas para a investigação, sem a necessidade de executar comandos no computador a ser investigado.

O método *in vivo*, quando há a possibilidade de ser executado, pode trazer informações valiosas sobre as ações do suspeito. Tráfego de rede, processos em execução e conexões abertas são os principais pontos a serem estudados, pois além de serem voláteis, podem indicar ações suspeitas, como envio e recebimento de informações restritas, usuários desconhecidos, entre outras informações (CASEY, 2004).

Durante a análise *in vivo*, o perito deve fazer uma cópia completa do sistema, tanto de informações voláteis quanto do disco rígido. É nesta etapa onde se encontra a primeira etapa do ciclo de perícia forense, a aquisição das evidências.

[4] *Cache*: dispositivo de armazenamento de informações temporárias com performance mais rápida que putros dispositivos de armazenamento. É acessada inúmeras vezes pelo sistema operacional para a consulta de informações (MARTINEZ, 2009).

3.1.1 Aquisição

Consiste basicamente em reunir todas as evidências possíveis para posterior análise. Como primeira fase do ciclo de perícia forense, o sucesso de uma investigação depende fundamentalmente da qualidade do material coletado.

A fase de aquisição das evidências não se restringe apenas ao computador. Toda mídia que possa ter entrado em contato com o sistema comprometido pode conter evidências. Portanto, devem ser investigadas também (COSTA, 2003).

Buscar evidências em um sistema operacional consiste em fazer uma varredura minuciosa nas informações, e este cenário nos remete ao conceito de ordem de volatilidade. Segundo Farmer e Venema (2007), a ordem de volatilidade das evidências deve ser respeitada como o fator de organização na coleta das evidências. Dependendo da volatilidade da informação, ela deve ser manuseada com mais cuidado, pois o simples ato da coleta pode danificá-las.

3.2 INVESTIGAÇÃO FORENSE DE REDE (*NETWORK FORENSIC*)

A etapa seguinte é a forense de rede, ou *network forensic*, que consiste em analisar informações sobre todos os ativos de rede que possam conter informações sobre o incidente. Nesta fase é possível reconstruir comunicações de rede interceptadas, com o intuito de recuperar informações transferidas, que serão devidamente analisadas na próxima etapa, a forense *post mortem*. Este tipo de investigação só é possível de ser realizada quando o equipamento a ser investigado faz parte de uma rede, independente se for corporativa ou doméstica.

Os principais objetivos da forense de rede é levantar informações a cerca do incidente. Porém, em alguns casos não é possível coletar evidências claras quando o incidente em si se deu por meio de um código malicioso com recursos criptografados. Ficam apenas as evidências de que ocorreu uma comunicação entre o computador comprometido e um outro computador (MELO, 2009).

Em suma, apesar das dificuldades que uma forense de rede pode apresentar, os resultados obtidos podem apresentar evidências ricas que possibilitem aos peritos reconstruírem muitas das ações cometidas (DIMER, 2007).

3.3 INVESTIGAÇÃO FORENSE *POST MORTEM* (*POST MORTEM FORENSIC*)

Nesta etapa são compreendidas as outras fases do ciclo de perícia forense: identificação, avaliação e apresentação. É neste momento que o perito analisa as informações colhidas durante a forense *in vivo* e forense de rede. A Figura 2 ilustra como a forense *in vivo* e forense de rede se entremeiam com o ciclo de perícia forense.

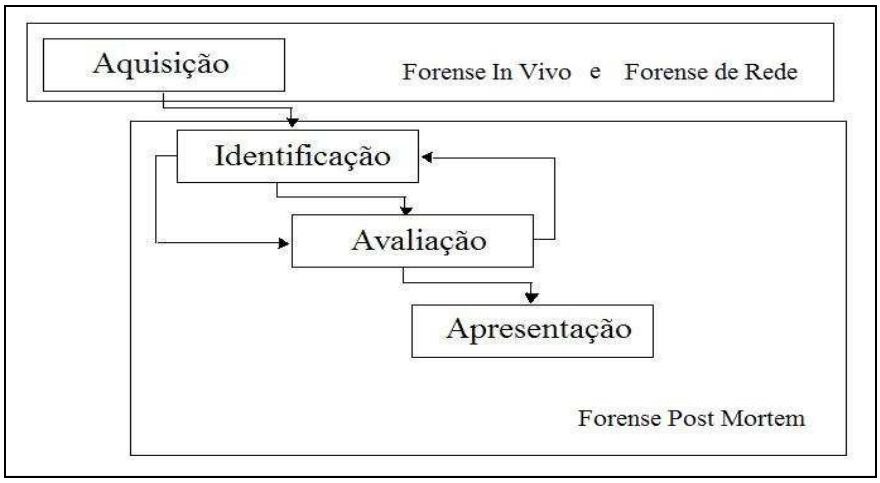

Figura 2. Ciclo de Perícia Forense e Etapas

3.3.1 Identificação

Consiste na etapa de organização dos artefatos encontrados na análise *in vivo* (*live analysis*). Nesta fase, o perito deve ser metódico, utilizando técnicas e procedimentos para a identificação e enumeração dos artefatos encontrados, pois, assim como a coleta de evidências, descrita no item aquisição da análise *in vivo*, é uma fase crucial para a elaboração do laudo pericial, resultado final da perícia forense (MELO, 2009).

3.3.2 Avaliação

É nesta etapa que o perito forense analisa os artefatos e enumera os eventos colhidos anteriormente em uma linha de tempo. É a etapa que representa a perícia forense computacional em si, ou seja, a fase onde são analisadas as evidências colhidas. O perito irá elaborar uma linha do tempo (*time line*) onde poderão ser compreendidos como e porque cada evento ocorreu, o que o ocasionou e o resultado do mesmo (COSTA, 2003).

Durante esta fase é impossível determinar um tempo médio para o término da avaliação de todo o material coletado. O perito pode se deparar com dados criptografados que para o qual ele não tenha uma ferramenta propícia para a análise. São inúmeras as situações possíveis com as quais o profissional de perícia forense pode se deparar, e manter um conjunto de ferramentas vasto pode facilitar seu trabalho (MELO, 2009).

Todo e qualquer procedimento que o perito realizar é necessário que seja documentado. O sucesso de uma análise forense computacional depende basicamente da qualidade e integridade das evidências coletadas e da documentação de todo o processo de perícia forense. O não cumprimento deste requisito pode invalidar toda a perícia, sendo para fins judiciais ou não (ARGOLO, 2005).

3.3.3 Apresentação

Última etapa da análise forense, a apresentação consiste na confecção e apresentação do laudo pericial. Ao perito cabe juntar e organizar todos os dados obtidos nas fases anteriores e relatar de forma clara e concisa. Pode-se dizer que o relatório final reconstrói o *modus operanti* do suposto atacante.

Apesar do laudo pericial não indicar diretamente a origem do ataque, as informações que nele contém podem ser utilizadas para este propósito. O relatório pode sugerir a origem e a categoria do incidente (COSTA, 2003).

3.4 ESTADO DA ARTE

3.4.1 Dos crimes digitais

Segundo dados do CERT.br, no período de janeiro de 1999 a março de 2010 houveram, aproximadamente, 1.211.994 incidentes de segurança reportados, sendo que 358.343 dos incidentes foram no ano de 2009, o maior índice atingido. Entre os casos reportados estavam casos de invasão, fraude, entre outros (CERT.br, 2010).

Abaixo segue um gráfico onde é possível verificar a quantidade de incidentes reportados em cada ano, onde também é possível verificar o pico de incidentes de 2009.

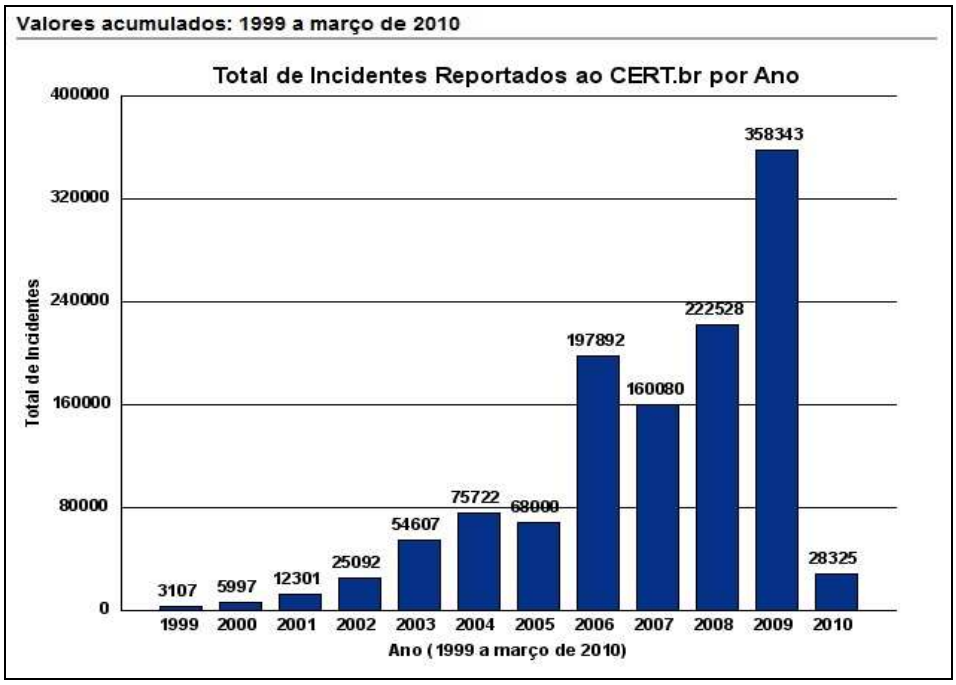

Figura 3**.** Incidentes de segurança reportados ao CERT.br – de 1999 à março de 2010
Fonte: CERT.br (2010)

Abaixo segue uma tabela com os tipos de ataques reportados ao CERT.br no período de janeiro a março de 2010 e o detalhamento da quantidade de ataques para cada mês.

Tabela 1. Incidentes reportados ao CERT.br -- Janeiro a Março de 2010

Mês	**Total**	**worm(%)**		**dos(%)**		**invasão(%)**		**web (%)**		**scan (%)**		**fraude (%)**		**outros(%)**	
jan	**8223**	1467	17	5	0	1	0	402	4	3042	36	2545	30	761	9
fev	**8266**	2170	26	5	0	5	0	470	5	2920	35	2313	27	383	4
mar	**11836**	2236	18	0	0	1	0	780	6	4573	38	3529	29	717	6
Total	**28325**	5873	20	10	0	7	0	1652	5	10535	37	8387	29	1861	6

Fonte: CERT.br (2010)

Analisando a Tabela 2 é possível notar que, no mês de março, último mês observado, houve um aumento significativo dos incidentes reportados. Com base nestes dados fornecidos pelo CERT.br é possível concluir que a quantidade de incidentes cresce vertiginosamente, juntamente com a evolução dos recursos computacionais.

3.4.2 Das ferramentas disponíveis

Atualmente existem várias opções de *softwares* disponíveis para a prática de perícia forense computacional, tanto livres quanto proprietários. Abaixo seguem algumas das ferramentas disponíveis:

a) *The Coroner's Toolkit* (TCT): o TCT é um conjunto de ferramentas de código aberto e gratuito, desenvolvido por Dan Farmer e Wietse Venema, que permite investigar um sistema Unix comprometido. Inicialmente, o TCT foi desenvolvido para ajudar a descobrir o que aconteceu em um sistema supostamente invadido, e não para apresentar evidências com fins judiciais (GEUS; REIS, 2002). Atualmente o projeto se encontra descontinuado, porém a ferramenta *The Sleuth Kit* foi desenvolvida baseada no TCT (FARMER; VENEMA, 1999);

b) *The Sleuth Kit* (TSK): como comentado no item anterior, este conjunto de ferramentas foi desenvolvido baseado no TCT. É um conjunto de ferramentas que pode ser utilizado tanto em sistema Linux como em Windows para a análise forense computacional. Além de ser gratuito, seu código é aberto e possui o recurso de *live-CD* que permite gerar um CD de *boot* capaz de carregar a ferramenta sem necessidade de ser instalada no sistema operacional da máquina a ser analisada (CARRIER, 2001);

c) *Autopsy Forensic Browser*: ferramenta utilizada para prover interface gráfica para o *Sleuth Kit* em HTML, sendo também gratuita e de código aberto. Com essa ferramenta, é possível visualizar de maneira mais amigável os dados apagados, detalhes e estrutura de arquivos, por meio de qualquer navegador *web* (CARRIER, 2001);

d) *Forensic and Incident Response Environment (Fire)*: é uma ferramenta que possui o recurso de *live-CD* e provê resposta imediata a incidentes, análise forense, recuperação de dados de partições perdidas, procura por vírus e vulnerabilidades. Também possui recursos para realizar análise forense *in vivo* em sistemas Windows 32 bits, bem como em sistemas Linux x86. Devido a esse recurso a ferramenta provê maior segurança na integridade dos dados, pois pode ser utilizada sem modificar qualquer informação na máquina analisada (SALUSKY, 2002);

e) *Trinux*: é uma distribuição baseada em sistemas Linux que pode ser inicializada tanto por disquetes como por um *live-CD* e traz as opções de segurança de rede para escaneamento de portas, escaneamento de vulnerabilidade, além de ferramentas para perícia forense (FRANZ, 1998);

f) *EnCase*: é uma ferramenta proprietária, desenvolvida pela empresa *Guidance Software*, que abrange muitos recursos gráficos, voltados para a análise forense. Desenvolvido baseado no ambiente Windows, esta ferramenta traz uma extensa gama de recursos para análise forense, tais como criação de imagens a partir do disco a ser evidenciado, suporte aos sistemas de arquivos FAT12 (*floppy*), FAT16, FAT32, NTFS, HFS, HFS+, *Solaris* UFS, EXT2/3, *Reiser*, BSD FFS, *Palm*, CDFS, *Joliet* e ISO 9660, suporte a configurações de disco como RAID 5, *Mirror, Striped*, entre outras opções (GUIDANCE SOFTWARE, 2010);

g) FTK (*Forensic Toolkit*): é uma ferramenta proprietária desenvolvida e comercializada pela empresa *Access Data*. Entre as suas funcionalidades estão recuperar arquivos excluídos, realizar recuperação de senhas, analisar arquivos compactados, entre outros recursos (ACCESS DATA, 2010);

h) *MD5summer*: é uma ferramenta gratuita, desenvolvida por Luke Pascoe, para o cálculo do *hash*, no caso da perícia forense é utilizado para verificar a integridade de imagens em qualquer formato. Desenvolvida para ambiente Windows, também é possível realizar a verificação de uma imagem feita de um sistema de arquivos Linux (PASCOE, 2010);

i*) ImageMASSter Solo-4*: esta ferramenta não contém somente software, e sim um conjunto com hardware e uma maleta, específicos para perícia forense. Desenvolvido e comercializado pela empresa *Intelligent Computer Solutions*, a ferramenta oferece a opção de realizar a duplicação pericial sem o auxílio de equipamentos extras. É possível capturar os dados suspeitos em uma velocidade muito maior que outras ferramentas, pois o disco a ser analisado fica em contato direto com mesmo, sem o auxílio de *drives* de CD, USB ou de disquetes. Ele também realiza o *hash* com MD5 ou CRC32, para verificar a integridade da cópia (INTELLIGENT COMPUTER SOLUTIONS, 2010);

j) FDTK: é uma distribuição GNU/Linux baseada na distribuição Ubuntu e contém mais de cem ferramentas para a realização da perícia forense. É possível utiliza-la via *live-CD* ou instalar em uma máquina e transforma-la em uma estação de análise. É um projeto brasileiro mantido pela comunidade de *software* livre e está na versão 2.01 (NEUKAMP; BOTELHO, 2010);

l) SMART Linux: é uma ferramenta, cuja sigla significa *Storage Media Analysis Recovery Toolkit*, desenvolvida e comercializada pela *ASR Data* e possui uma versão para ambiente Windows e outra para Linux. Possui a opção de criar um *live-CD* ou somente baixar a ferramenta e instala-la em uma estação (ASR DATA, 2010).

4 MÉTRICAS DE SOFTWARE

Métricas de *software* podem ser definidas como uma medição quantitativa simples, com o intuito de expressar em números a funcionalidade, a qualidade ou o tamanho de uma aplicação. Com esta informação em mãos é possível estimar custo, esforço e prazo total de um projeto de aplicação (PRESSMAN, 2002).

Utilizando a métrica de *software*, é possível que engenheiros de *software* consigam estimar os recursos necessários, bem como artefatos relevantes ao desenvolvimento do projeto. Com esta informação é possível quantificar propriedades de produtos de *software* em processo de planejamento ou já concluídos e prontos para o mercado (SOUSA, 2006).

4.1 MÉTRICAS DE DIMENSIONAMENTO FUNCIONAL

Existem dois tipos de métricas de dimensionamento funcional, que diferem-se quanto ao seu objetivo e momento de aplicação. Quanto ao objetivo realiza a contagem de linhas do *software*, buscando definir o seu “tamanho”, porém esta classificação é limitada à linguagem empregada para a construção da aplicação e ser dependente da conclusão do projeto, não sendo possível criar estimativas nesta categoria. Ela também realiza contagem de número de páginas, tempo de duração do projeto, número de erros, entre outros aspectos (SOUSA, 2006).

Quanto ao seu momento de aplicação, as métricas podem ser diferenciadas por criar estimativas (em um projeto em execução), analisar projetos híbridos ou aplicações prontas (SOUSA, 2006).

Esta última classificação de métricas é a que será utilizada no desenvolvimento deste trabalho.

4.2 MÉTRICA ANÁLISE DE PONTOS DE FUNÇÃO (*FUNCTION POINT ANALYSIS*)

Em 1979, o pesquisador da IBM Allan Albrecht propôs a métrica de análise de pontos de função, em inglês *function point analysis* (FPA), que consiste em atribuir pontos a fim de classificar uma aplicação. É largamente utilizada à nível mundial como principal indicador de formação de preços, bem como para estimar prazos e preços em licitações de projetos de *software* (SOUSA, 2006).

É necessário esclarecer que a análise e atribuição de pontos devem ser feitas pelo ponto de vista do usuário e não do desenvolvedor do *software*.

Primeiramente é necessário determinar o tipo de contagem. A Tabela 3 demonstra os tipos que podem ser definidos como tipo de contagem.

Tabela 2. Tipos de contagem previstos na métrica FPA

#	**Tipo**	**Detalhamento**
1	Desenvolvimento de projetos	Conta as funções fornecidas ou desejadas pelos usuários quando da primeira versão do software.
2	Melhoria de projetos	Conta as modificações necessárias (adição, melhoria, exclusão) em funcionalidades de uma aplicação existente.
3	Aplicação	Conta as funções de uma aplicação instalada. Inicialmente seu valor corresponde ao do tipo “Desenvolvimento de Projetos”. Porém, a cada melhoria do software, sua contagem precisa ser atualizada para refletir o valor real das funcionalidades presentes.

Fonte: IFPUG (2004)

Para a execução da análise comparativa que este trabalho se propõe será utilizado o tipo de contagem número três, pois serão comparados *softwares* que já estão prontos e disponíveis no mercado. O tipo de contagem número três se dispõe a pontuar as funções de uma aplicação pronta, com o intuito de realizar uma melhoria no mesmo, uma atualização, avaliando assim a qualidade do *software*.

Apesar do conceito de qualidade ser relativo, a métrica FPA, e as métricas em geral, analisam como qualidade a quantidade de recursos que uma aplicação apresenta e o detalhamento das informações apresentadas para o usuário. Exemplificando, a métrica faz a

análise de um *software*, listando todos os recursos oferecidos, todos os campos destas funcionalidades e o detalhamento dos resultados, analisando também a organização dos resultados. Então, atribuindo pontos á estas funcionalidades, o resultado obtido classifica a qualidade do *software*.

Em suma, os recursos oferecidos juntamente com as informações que a aplicação provê ao usuário e o detalhamento destas informações resultam na qualidade do *software*, e a métrica FPA atribui valores aos recursos da aplicação, então, consequentemente o valor resultante expressa a qualidade do *software*.

Porém, para aplicarmos este tipo de contagem ao presente estudo, será feita uma adaptação da métrica: será feita a contagem das funções para o estudo comparativo entre dois *softwares* de perícia forense, com a intenção de apenas conhecer a pontuação de cada um, e não o de atualizar os *softwares*.

4.2.1 Contagem de Funções de Dados

Funções de dados representam grupos lógicos que são referenciados pela aplicação por meio de requisitos funcionais do usuário. Pode-se citar como exemplo um arquivo de cadastro de clientes: cadastrar o cliente é uma função da aplicação, porém o arquivo gerado não faz necessariamente parte da aplicação (SOUSA, 2006).

A métrica FPA prevê dois tipos de funções de dados:

a) Arquivo Lógico Interno (ALI): são grupos lógicos de dados que a aplicação mantém e que são reconhecidos pelos usuários, como, por exemplo, inclusões, alterações ou exclusões de dados;

b) Arquivo de Interface Externa (AIE): são também grupos lógicos de dados referenciados pela aplicação, porém mantidos por outra aplicação, identificados pelo usuário.

Tanto os ALI como os AIE possuem campos, chamados de Elemento de Dados (EDD). Para o caso de ALI, em um exemplo de um cadastro de clientes, os EDD poderiam ser CNPJ, Razão Social, Nome Fantasia, entre outros.

Os EDD também podem possuir subgrupos, denominados de Elemento de Registro (ER). Um exemplo de ALI de cadastro de pedidos, além dos campos usuais de um cadastro deste tipo (código do pedido, valor total, data do pedido), um ER poderia ser a descrição do pedido e um outro ER seria relativo aos itens do pedido (SOUSA, 2006).

Os ALI e AIE identificados, EDD e ER associados, são utilizados para contagem dos Pontos de Função não Ajustados (PFNA). A Tabela 4 mostra como avaliar a complexidade funcional dos ALI e AIE.

Tabela 3. Critérios para a avaliação de complexidade funcional de ALI e AIE

	Entre 1 a 19 EDD	**Entre 20 a 50 EDD**	**51 ou mais EDD**
1 ER	Baixa	Baixa	Média
2 a 5 ER	Baixa	Média	Alta
6 ou mais ER	Média	Alta	Alta

Fonte: IFPUG (2004)

Para realizar a contagem de funções de dados é necessário que seja feita a avaliação da complexidade funcional e transformar em PFNA cada ALI e AIE identificados na aplicação. A Tabela 5 demonstra como fazer a transformação da complexidade funcional dos ALI identificados na aplicação em valores numéricos.

Tabela 4. Transformação da complexidade funcional de ALI em PFNA

Complexidade Funcional	**Pontos de Função não Ajustados (PFNA)**
Baixa	7
Média	10
Alta	15

Fonte: IFPUG (2004)

A Tabela 6 demonstra como fazer a transformação da complexidade funcional de dos AIE identificados na aplicação para valores numéricos.

Tabela 5. Transformação da complexidade funcional de AIE em PFNA

Complexidade Funcional	Pontos de Função não Ajustados (PFNA)
Baixa	5
Média	7
Alta	10

Fonte: IFPUG (2004)

Após a transformação de todos os ALI e AIE em valores de PFNA ser completada, os valores obtidos devem ser somados com os pontos que serão originários da contagem de funções de transação, que serão descritos no próximo item.

4.2.2 Contagem de Funções de Transação

Representam, em pontos, as funcionalidades de processamento de dados fornecidas pela aplicação aos usuários. Após a contagem deste tipo de função juntamente com os pontos de funções de dados, descritos no item anterior, irão totalizar os Pontos de Funções não Ajustados na FPA (SOUSA, 2006).

Os tipos de funções de transação previstos na FPA são:

a) Entrada Externa (EE): refere-se ao processamento de dados que são utilizados pela aplicação para atualizar ALI ou alterar o comportamento do sistema;
b) Saída Externa (SE): corresponde apresentar ao usuário o resultado do processamento pela aplicação. Para ser considerada a apresentação de informações ao usuário um resultado de processamento, deve-se, no mínimo, conter um cálculo matemático ou expressões para produção de dados derivados, além de atualizar um ou mais ALI ou alterar o comportamento do sistema, se necessário;

c) Consulta Externa (CE): superficialmente parecida com a SE, pois visa apresentar informações aos usuários. Porém a CE não pode envolver cálculos matemáticos ou expressões para produção de dados, além de não atualizar ALI nem alterar o comportamento do sistema.

A quantidade de Arquivos Referenciados (AR) e os Elementos de Dados (EDD) associados aos arquivos, são utilizados para obtenção da complexidade funcional das EE, SE e CE identificadas na aplicação.

A Tabela 7 mostra os critérios para avaliar a complexidade funcional de EE.

Tabela 6. Critérios para a avaliação de complexidade funcional de EE

	Entre 1 a 4 EDD	**Entre 5 a 15 EDD**	**16 ou mais EDD**
0 ou 1 AR	Baixa	Baixa	Média
2 AR	Baixa	Média	Alta
3 ou mais AR	Média	Alta	Alta

Fonte: IFPUG (2004)

A Tabela 8 mostra os critérios para avaliar a complexidade funcional de SE e CE.

Tabela 7. Critérios para a avaliação de complexidade funcional de SE e CE

	Entre 1 a 5 EDD	**Entre 6 a 19 EDD**	**20 ou mais EDD**
0 ou 1 AR	Baixa	Baixa	Média
2 a 3 AR	Baixa	Média	Alta
4 ou mais AR	Média	Alta	Alta

Fonte: IFPUG (2004)

A transformação da complexidade funcional em PFNA deve ser realizada para SE, EE e CE individualmente. A Tabela 9 representa a transformação em PFNA os SE identificados na aplicação.

Tabela 8. Transformação da complexidade funcional de SE em PFNA

Complexidade Funcional	**Pontos de Função não Ajustados (PFNA)**
Baixa	4
Média	5
Alta	7

Fonte: IFPUG (2004)

A Tabela 10 representa a transformação em PFNA os EE e CE identificados na aplicação.

Tabela 9. Transformação da complexidade funcional de EE e CE em PFNA

Complexidade Funcional	Pontos de Função não Ajustados (PFNA)
Baixa	3
Média	4
Alta	6

Fonte: IFPUG (2004)

Após serem transformados em PFNA a complexidade funcional de todos os SE, EE e CE identificados, os valores parciais acumulados devem ser somados com os valores obtidos nas transformações em PFNA das funções de dados. Tendo o valor total de PFNA, é necessário definir o Valor do Fator de Ajuste (VFA) para que seja obtido o valor total de avaliação da aplicação. Como obter o VFA será detalhado no próximo item deste capítulo.

4.2.3 Determinação do valor do fator de ajuste

A métrica de avaliação FPA prevê quatorze Características Gerais do Sistema (CGS) para que seja definido o VFA, independente da tecnologia empregada na confecção da aplicação.

As CGS devem ser classificadas de acordo com uma escalada de influência ordinal, representada na Tabela 11.

Tabela 10. Escala de influência para a avaliação das CGS na FPA

Nível de Influência (NI)	Semântica
0	Característica não presente, ou não influente
1	Influência inicial
2	Influência moderada
3	Influência média
4	Influência significativa
5	Forte influência

Fonte: IFPUG (2004)

A seguir são descritas as quatorze CGS da FPA. É necessário que sejam definidos os Níveis de Influência (NI) mais apropriados em cada CGS, e não a dificuldade apresentada pela aplicação exatamente (IFPUG, 2004).

4.2.3.1 Descrição das CGS previstas na FPA

a) Comunicação de dados: grau de comunicação direta da aplicação com o processador. Toda e qualquer informação e/ou dados recebidos e/ou enviados pela aplicação irão utilizar algum recurso de comunicação de dados, tais como terminais conectados localmente na unidade de controle (IFPUG, 2004). Abaixo a descrição de como pontuar o graus de comunicação de dados de acordo com a sua influência;

Tabela 11. Como pontuar de acordo com o grau de influência de comunicação de dados

Descrição dos graus de influência	**Pontuar Como**
O processamento da aplicação será puramente *batch* [5]ou será executado em um PC isolado	0
A aplicação será *batch* mas tem entrada de dados remota ou impressão remota	1
A aplicação será *batch* mas tem entrada de dados remota e impressão remota	2
Captura de dados *on-line* via terminal de vídeo ou via um processador *front-end*[6], para alimentar processos *batch* ou sistemas de consultas	3
Mais que um *front-end*, mas a aplicação suportará apenas um tipo de protocolo de comunicação	4
Mais que um *front-end* e a aplicação suportará mais de um tipo de protocolo de comunicação	5

Fonte: SOUSA (2006)

b) Processamento de dados distribuído: grau de transferência de dados entre os componentes da aplicação avaliada (IFPUG, 2004). Abaixo a descrição de como pontuar de acordo com o grau de processamento de dados;

[5] "Processamento tradicional, normalmente executado em equipamento de grande porte. A entrada e processamento dos dados são realizados por lotes periódicos e contrasta com o modo *on-line*." (SAWAYA, 1999)

[6] Pode ser considerado a interface entre o usuário e o *back-end* (o responsável por executar os processos), onde o *front-end* coleta as informações providas pelo usuário e processa-as para a utilização pelo *back-end.*

Tabela 12. Como pontuar de acordo com o grau de influência de processamento de dados

Descrição dos graus de influência	Pontuar Como
A aplicação não auxiliará na transferência de dados ou processamento entre as CPU da instalação	0
A aplicação preparará dados para o usuário final processar em outra CPU da instalação. Por exemplo, planilhas eletrônicas ou gerenciadores de banco de dados de PC	1
Os dados serão preparados para transferência, transferidos e processados em uma outra CPU da instalação (mas não para processamento pelo usuário final)	2
Processamento distribuído e transferência de dados *on-line* apenas em uma direção	3
Processamento distribuído e transferência de dados *on-line* em ambas as direções	4
As funções de processamento serão executadas dinamicamente na CPU mais apropriada	5

Fonte: SOUSA (2006)

c) Desempenho: grau de influência sobre tempo de resposta e performance que influenciam no desenvolvimento da aplicação. Identifica objetivos de desempenho aprovados pelo usuário que influenciarão o projeto, desenvolvimento, instalação e suporte da aplicação (IFPUG, 2004). Abaixo a Tabela 14 descreve como pontuar de acordo com o nível de influência de desempenho;

Tabela 13. Como pontuar de acordo com o grau de influência de desempenho

Item	Descrição dos graus de influência	Pontuar Como
1	Nenhuma exigência especial de desempenho foi fixada pelo usuário	0
2	Requisitos de projeto e desempenho foram estabelecidos e revisados, mas nenhuma ação especial será necessária	1
3	O tempo de resposta será crítico durante as horas de pico. Nenhuma consideração especial para utilização de CPU foi requerida. O intervalo de tempo limite do processamento é sempre para o próximo dia útil	2
4	O tempo de resposta será crítico durante todo o horário de utilização. Não será necessário nenhum procedimento especial para utilização de CPU. Os requisitos de prazo de processamento com outros sistemas são limitados	3
5	Além do descrito no item 3, os requisitos de desempenho estabelecidos pelo usuário são rigorosos o bastante para requerer tarefas de análise de desempenho na fase de análise e projeto da aplicação	4
6	Além do descrito no item 4, ferramentas de análise de desempenho deverão ser usadas nas fases de projeto, desenvolvimento e/ou implementação a fim de proporcionar o desempenho estabelecido pelo usuário	5

Fonte: SOUSA (2006)

d) Utilização de equipamento: grau de restrições de recursos computacionais que influenciam no desenvolvimento da aplicação, tais como hardware ou uma política de acesso à rede (IFPUG, 2004). Abaixo segue tabela descrevendo como pontuar de acordo com o nível de influência de utilização de equipamento;

Tabela 14. Como pontuar de acordo com o grau de influência de utilização de equipamento

Descrição dos graus de influência	**Pontuar Como**
Não há restrições operacionais explícitas ou implícitas	0
Existem restrições operacionais, mas são menos restritivas do que aplicações típicas. Nenhum esforço extra será necessário para suplantar as restrições	1
Algumas considerações sobre tempo e segurança são necessárias	2
Necessidades especiais de processador para uma parte específica da aplicação	3
Restrições operacionais estabelecidas requerem atenção especial a nível de processador central ou processador dedicado para executar a aplicação	4
Além do descrito acima, existirão sobrecargas nas unidades de processamento (CPU) distribuídas da instalação	5

Fonte: SOUSA (2006)

e) Taxa de transações: grau de transações de negócio que influenciam no desenvolvimento da aplicação. De acordo com o que o usuário estipulou, será necessário verificar o grau de ocorrências de transações, pois se no ambiente onde será implantado o sistema existe um grau alto de transações comerciais, é necessário implantar tarefas de análise de desempenho (IFPUG, 2004). Segue tabela descrevendo como pontuar de acordo com o grau de influência de transações de negócio;

Tabela 15. Como pontuar de acordo com o grau de influência de transações de negócio

Descrição dos graus de influência	**Pontuar Como**
Nenhum período de pico de transações é esperado	0
Picos de transações mensais, quadrimestrais, sazonais e anuais são esperados	1
Picos semanais de transações são esperados	2
Picos diários de transações são esperados	3
Altos volumes de transações foram fixados pelo usuário para a aplicação, o que força a execução de tarefas de análise de performance na fase de projeto da aplicação	4
Requer o uso de ferramentas de análise de performance nas fases de projeto, desenvolvimento e/ou implantação, além das considerações acima	5

Fonte: SOUSA (2006)

f) Entrada de dados *on-line*: grau de interação com o meio *on-line*, tais como atualizações de dados, entrada de dados na aplicação, disponibilidade de funções de controle via *web*, entre outras (IFPUG, 2004). Na Tabela 17 é exemplificado como pontuar de acordo com o grau de influência de interação *on-line*;

Tabela 16. Como pontuar de acordo com o grau de influência de entrada de dados *on-line*

Descrição dos graus de influência	**Pontuar Como**
Todas as transações serão processadas em modo *batch*	0
1% a 7% das transações serão entradas de dados interativas	1
8% a 15% das transações serão entradas de dados interativas	2
16% a 23% das transações serão entradas de dados interativas	3
24% a 30% das transações serão entradas de dados interativas	4
Mais de 30% das transações serão entradas de dados interativas	5

Fonte: SOUSA (2006)

g) Eficiência do usuário final: grau de facilidade de uso da aplicação. Considerações em relação ao fator humano na utilização da aplicação devem ser avaliadas (IFPUG, 2004). Abaixo segue uma lista de itens que devem ser analisados para verificar se existem na aplicação para realizar a pontuação dos graus de influência:

- Facilidades para navegação (teclas de função, geração dinâmica de menus);
- Existência de menus;
- Ajuda e documentação *on-line*;
- Movimento automático do cursor (pular de um campo a outro após o seu preenchimento);
- Movimento de *scroll* (vertical e horizontal);
- Impressão remota via transações *on-line*;
- Teclas de função pré-definidas;
- Execução de trabalhos *batch* a partir de transações *on-line*;
- Seleção de dados da tela por meio de movimentação do cursor;

- Uso intensivo de vídeo reverso, brilho, sublinhado, cores e outros recursos de vídeo;

- Documentação de transações *on-line*;

- Possibilidade do uso de *mouse*;

- Janelas *pop-ups*;

- Número mínimo de telas para execução de funções de negócio;

- Suporte a dois idiomas (contar como quatro itens);

- Suporte a múltiplos idiomas, ou seja, suportar mais de dois idiomas (contar como seis itens) (SOUSA, 2006).

Na Tabela 18 é exemplificado como pontuar de acordo com o grau de facilidade de utilização da aplicação levando em consideração os itens citados acima;

Tabela 17. Como pontuar de acordo com o grau de influência da eficiência do usuário final

Descrição dos graus de influência	**Pontuar Como**
Nenhum dos itens acima	0
Apresenta de 1 a 3 dos itens acima	1
Apresenta de 4 a 5 dos itens acima	2
Apresenta 6 ou mais dos itens acima, mas não há requisito do usuário relacionado à eficiência	3
Apresenta 6 ou mais dos itens acima, e os requisitos estabelecidos para eficiência do usuário são rigorosos o suficiente para que a fase de projeto da aplicação inclua fatores, tais como: minimizar a digitação, maximizar os valores padrões, utilizar padrões de visualização, etc.	4
Apresenta 6 ou mais dos itens acima, e os requisitos estabelecidos para eficiência do usuário são rigorosos o suficiente para que seja necessário o uso de ferramentas e processos especiais para demonstrar que os objetivos de eficiência foram alcançados	5

Fonte: SOUSA (2006)

h) Atualização *on-line*: grau de atualizações *on-line* dos arquivos lógicos internos (ALI) (IFPUG, 2004). A Tabela 19 descreve como pontuar de acordo com o grau de influência de atualizações *on-line*;

Tabela 18. Como pontuar de acordo com o grau de influência de atualizações *on-line*

Descrição dos graus de influência	Pontuar Como
Nenhuma atualização	0
Atualização *on-line* de 1 a 3 arquivos de controle. O volume de atualizações é baixo e a recuperação de dados é simples	1
Atualização *on-line* de 4 ou mais arquivos de controle. O volume de atualizações é baixo e a recuperação de dados é simples	2
Atualização *on-line* da maioria dos Arquivos Lógicos Internos	3
Além dos itens anteriores, a proteção contra perda de dados é essencial e foi especificamente projetada e codificada no sistema	4
Além dos itens anteriores, altos volumes de dados trazem considerações sobre custo para o processo de recuperação. Exigem procedimentos de recuperação totalmente automatizados com a mínima intervenção do operador	5

Fonte: SOUSA (2006)

i) Complexidade de processamento: grau de processamento lógico que influencia o desenvolvimento da aplicação. É necessário analisar os itens abaixo para verificar se existem na aplicação avaliada para poder realizar a pontuação do grau de influência:

- Controle sensível (processamento especial de auditoria) e/ou processamento específico de segurança da aplicação;

- Processamento lógico extensivo;

- Processamento matemático extensivo;

- Grande quantidade de processamento de exceções, resultante de transações incompletas que necessitam de novo processamento (transações incompletas de caixas automáticos causadas por interrupções de comunicação, valores de dados ausentes ou validações de erros);

- Processamento complexo para manipular múltiplas possibilidades de entrada/saída (múltiplos meios e independência de equipamentos) (IFPUG, 2004).

A Tabela abaixo demonstra como pontuar de acordo com o grau de influência de complexidade de processamento, após analisar os itens citados acima;

Tabela 19. Como pontuar de acordo com o grau de influência da complexidade de processamento

Descrição dos graus de influência	Pontuar Como
Nenhum dos itens acima	0
Apresenta um dos itens acima	1
Apresenta dois dos itens acima	2
Apresenta três dos itens acima	3
Apresenta quatro dos itens acima	4
Apresenta todos os itens acima	5

Fonte: SOUSA (2006)

j) Reusabilidade: possibilidade de reutilizar o código da aplicação em outras aplicações. Grau de reusabilidade refere-se a projetar e desenvolver a aplicação objetivando o seu reuso (IFPUG, 2004). A Tabela 21 exemplifica como pontuar de acordo com o grau de reusabilidade de código;

Tabela 20. Como pontuar de acordo com o grau de influência de reusabilidade de código

Descrição dos graus de influência	Pontuar Como
Não apresenta código reutilizável	0
O código reutilizável será usado somente dentro da própria aplicação	1
Menos de 10% do código da aplicação considera sua utilização por outras aplicações	2
10% ou mais da aplicação considera sua utilização por outras aplicações	3
A aplicação será projetada e documentada para facilitar a reutilização de código e a aplicação será customizada pelo usuário a nível do código fonte	4
A aplicação será projetada e documentada para facilitar a reutilização de código e a aplicação será customizada para uso por meio de parâmetros que podem ser atualizados pelo usuário	5

Fonte: SOUSA (2006)

l) Facilidade de instalação: grau de facilidade de instalação da aplicação de acordo com especificações do usuário. Importante também analisar a conversão de dados de aplicações que precederam a aplicação analisada (IFPUG, 2004). A Tabela 22 descreve como pontuar de acordo com o grau de influência da facilidade de instalação da aplicação;

Tabela 21. Como pontuar de acordo com o grau de influência da facilidade de instalação

Item	Descrição dos graus de influência	Pontuar Como
1	Nenhuma consideração especial foi feita pelo usuário e nenhum procedimento especial foi requerido para a implantação	0
2	Nenhuma consideração especial foi feita pelo usuário, mas um procedimento especial é requerido para a implantação	1
3	Requisitos de implantação e conversão de dados foram fixados pelo usuário, e roteiros de implantação e conversão de dados devem ser preparados e testados. O impacto da conversão de dados no projeto não é considerado importante	2
4	Requisitos de implantação e conversão de dados foram fixados pelo usuário e roteiros de implantação e conversão de dados devem ser preparados e testados. O impacto da conversão de dados no projeto é considerado importante	3
5	Além do descrito no item 2, ferramentas automatizadas de implantação e conversão de dados devem ser preparadas e testadas	4
6	Além do descrito no item 3, ferramentas automatizadas de implantação e conversão de dados devem ser preparadas e testadas	5

Fonte: SOUSA (2006)

m) Facilidade operacional: grau de facilidade de utilização da aplicação analisada. Funções manuais tais como manipulação de formulários e intervenção direta do operador devem ser minimizadas pela aplicação (IFPUG, 2004). Abaixo a Tabela 23 demonstra como pontuar de acordo com o grau de facilidade operacional;

Tabela 22. Como pontuar de acordo com o grau de influência da facilidade operacional

Descrição dos graus de influência	Pontuar Como
Nenhuma consideração especial sobre facilidade operacional, além dos procedimentos normais de *backup*, foi feita pelo usuário	0
Um, alguns, ou todos os itens seguintes aplicam a aplicação. Avalie todos que se aplicam. Cada item tem o valor de um ponto, exceto quando apontar em contrário - Procedimentos eficientes de inicialização, *backup* e recuperação devem ser preparados, mas a intervenção do operador é necessária; - Procedimentos eficientes de inicialização, *backup* e recuperação devem ser preparados, mas nenhuma intervenção do operador é necessária (contar como dois itens); - A aplicação minimizará a operação de montagem de fitas magnéticas; - A aplicação minimizará a necessidade de manuseio de formulários.	1 a 4
A aplicação será projetada para não precisar de intervenção do operador no seu funcionamento normal. Apenas a inicialização e parada do sistema ficam a cargo do operador. A recuperação automática de erros será uma característica da aplicação	5

Fonte: SOUSA (2006)

n) Múltiplos locais: grau de utilização da aplicação em várias máquinas ou organizações (IFPUG, 2004). A Tabela 24 descreve como pontuar de acordo com o grau de influência da utilização da aplicação em múltiplos locais;

Tabela 23. Como pontuar de acordo com o grau de influência da utilização em múltiplos locais

Item	Descrição dos graus de influência	Pontuar Como
1	Nenhuma solicitação do usuário para considerar a necessidade de instalar a aplicação em mais de um local	0
2	Necessidade de instalação em múltiplos locais deve ser levada em consideração no projeto do sistema e a aplicação será projetada para operar somente em ambientes idênticos de hardware e software	1
3	Necessidade de instalação em múltiplos locais deve ser levada em consideração no projeto do sistema e a aplicação será projetada para operar somente em ambientes similares de hardware e software	2
4	Necessidade de instalação em múltiplos locais deve ser levada em consideração no projeto do sistema e a aplicação será projetada para operar inclusive em ambientes diferentes de hardware e/ou software	3
5	Um plano de documentação e manutenção deve ser elaborado e testado para suportar a aplicação em múltiplos locais e a aplicação atende aos itens 1 e 2	4
6	Um plano de documentação e manutenção deve ser elaborado e testado para suportar a aplicação em múltiplos locais e a aplicação atende ao item 3	5

Fonte: SOUSA (2006)

o) Facilidades de mudanças: grau de facilidades de mudanças na aplicação em relação a lógica de processamento e estrutura de dados. Abaixo alguns itens que devem ser analisados para a posterior avaliação da pontuação:

- Será fornecido recurso de consulta e relatórios flexíveis, capaz de manipular solicitações simples de consulta, aplicada a somente um Arquivo Lógico Interno (contar como um item);

- Será fornecido recurso de consulta e relatórios flexíveis, capaz de manipular solicitações de consulta de média complexidade, aplicada a mais de um Arquivo Lógico Interno (contar como dois itens);

- Será fornecido recurso de consulta e relatórios flexíveis capaz de manipular solicitações complexas de consulta, com combinações de um ou mais Arquivos Lógicos Internos (contar como três itens);

- Dados de controle de negócio serão mantidos em Tabelas que são atualizadas pelo usuário por meio de processos *on-line* e interativos, mas as alterações só serão efetivadas no próximo dia útil;

- Dados de controle de negócio serão mantidos em Tabelas que podem ser atualizadas pelo usuário por meio de processos *on-line* e interativos e as alterações serão efetivadas imediatamente (contar como dois itens) (IFPUG, 2004).

Após analisar os itens acima e verificar quais que condizem com o projeto, é necessário realizar a pontuação de acordo com o grau de influência da facilidade de mudanças, exemplificada pela Tabela 25.

Tabela 24. Como pontuar de acordo com o grau de influência das facilidades de mudança

Descrição dos graus de influência	**Pontuar Como**
Nenhum dos itens acima	0
Apresenta um dos itens acima	1
Apresenta dois dos itens acima	2
Apresenta três dos itens acima	3
Apresenta quatro dos itens acima	4
Apresenta cinco dos itens acima	5

Fonte: SOUSA (2006)

4.2.4 Cálculo do valor final da aplicação de acordo com a métrica FPA

Após realizar a definição dos NI de todas as CGS, é necessário calcular o valor de Nível de Influência Total (NIT), que é a soma de todos os valores de NI de todas as CGS. Após a obtenção deste valor, é necessário aplicar a fórmula abaixo para obter o Valor do Fator de Ajuste (VFA):

$$VFA = (NIT * 0,01) + 0,65$$

Os valores 0,01 e 0,65 correspondem à um índice que pode variar entre 0,65 e 1,35. Para a elaboração desta fórmula foi feita uma análise de estatísticas, buscando um

equilíbrio entre a contagem dos pontos de função e o real esforço de implementação necessário (SOUSA, 2006).

Tendo o VFA em mãos, o próximo passo é realizar o cálculo dos FPA, que é o valor final correspondente a avaliação da aplicação de acordo com o métrica de Pontos de Função. Para o cálculo final é utilizada a fórmula a seguir.

$$PFA = PFNA * VFA$$

O valor obtido em PFA possibilita uma variação de ±35% entre os PFNA e os PFA.

5 AMBIENTE DE TESTES

Para a confecção deste estudo foi escolhida a distribuição GNU/Linux Ubuntu versão 9.04, utilizando o sistema de arquivos ext4. O sistema foi instalado em uma partição de 4 GB criada em um disco rígido de 80 GB. Não foi utilizado o disco em sua totalidade para que a criação da imagem para a análise não demandasse muito tempo, pois o intuito deste trabalho é apenas simular um ambiente de utilização doméstica, onde o usuário utiliza o disco inteiro.

Foram realizadas algumas simulações de utilização na partição de 4 GB, criando arquivos, alterando documentos e excluindo alguns arquivos, gerando assim dados para serem analisados pelas ferramentas de perícia forense.

O objetivo do trabalho é analisar os recursos oferecidos pelas ferramentas, bem como a organização dos resultados apresentados pelos *softwares*, e não os resultados em si, pois, como o ambiente analisado é controlado, os resultados das ferramentas já são esperados, então o que será analisado pela métrica são os recursos e a organização dos resultados e formas de visualização destes, uma vez que o ambiente contém informações simuladas.

5.1 DEFINIÇÃO DOS *SOFTWARES* A SER ESTUDADOS

Foram selecionados, dentre os *softwares* já citados no capítulo 3, os *softwares The Sleuth Kit* utilizando a sua interface gráfica, o *Autopsy* e a ferramenta SMART Linux. Foram escolhidas estas ferramentas, pois ambas são voltadas ao ambiente Linux, possuem uma interface gráfica e tem a opção de somente instalar a ferramenta, ou seja, não é preciso criar um CD de *boot*[7] para utilizá-las, apesar das duas possuírem esta opção.

[7] Processo de inicialização do computador, onde é carregado o sistema operacional.

A seguir será feita uma breve apresentação dos principais recursos de cada um e como utilizá-los, demonstrando por meio de imagens que foram capturadas dos *softwares* em utilização.

5.1.1 *The Sleuth Kit* e *Autopsy*

Este conjunto de ferramentas, como dito anteriormente no Capítulo 3, foi desenvolvido baseando-se na ferramenta *The Coroner's Toolkit* e utiliza como interface gráfica o *Autopsy*. É voltado ao ambiente Linux, porém é possível instalá-lo em um ambiente Windows por meio de um emulador, o *Cygwin*, e executar a ferramenta por este utilitário, e não por meio dos instaladores nativos do Windows.

Após a instalação do *Sleuth Kit* e do *Autopsy*, é necessário executar o arquivo *autopsy* via terminal, não sendo necessárias permissões de administrador. Este arquivo é um *script*[8] e se encontra dentro da própria pasta do *software*, que foi criado quando o mesmo foi instalado. A Figura 4 mostra como é a sua execução.

[8] São mini programas, baseados numa determinada linguagem, que executam determinadas operações.

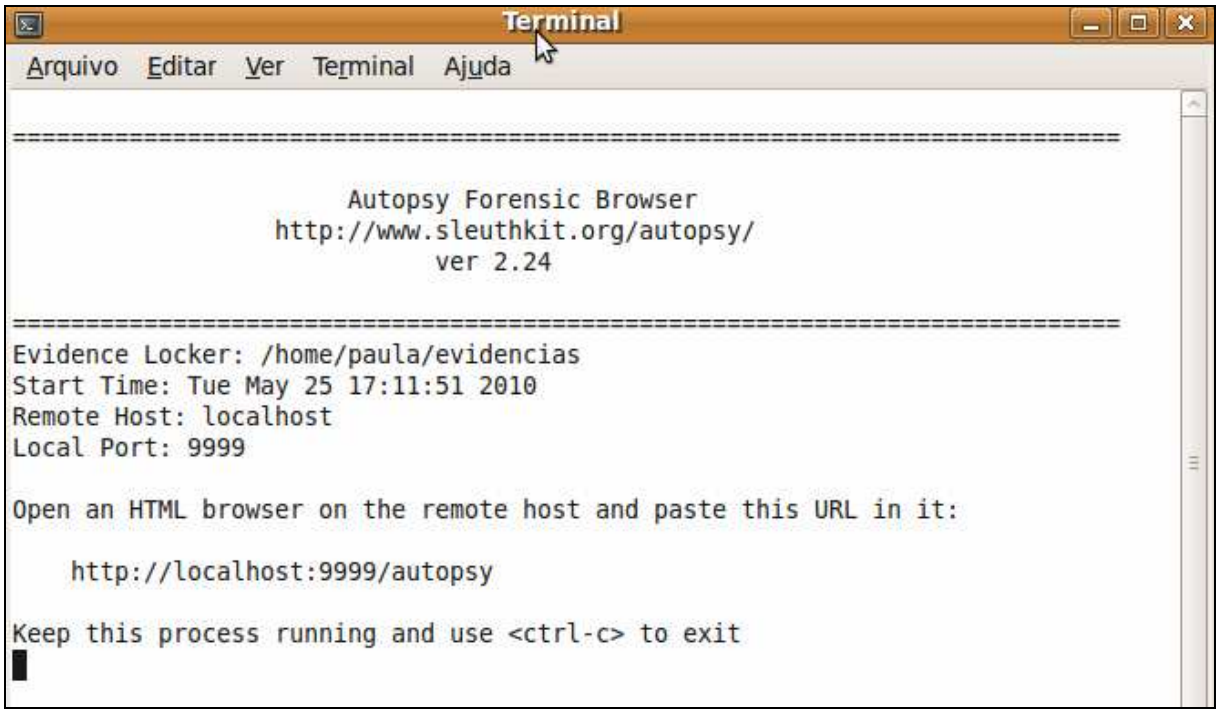

Figura 4. Executando o *scritp Autopsy* via terminal

Para a execução do *Autopsy* é necessário que este *script* esteja executando durante a utilização via *browser*. Sem a execução deste não é possível utilizar a ferramenta.

As instruções para a execução se encontra na própria tela do terminal onde foi executado o script *autopsy*, na penúltima linha. É necessário somente abrir um navegador *web* e digitar na barra de endereços http://localhost:9999/autopsy. Este endereço acessa, pela porta 9999 o aplicativo *Autopsy*, A Figura 5 apresenta a tela inicial do *software*.

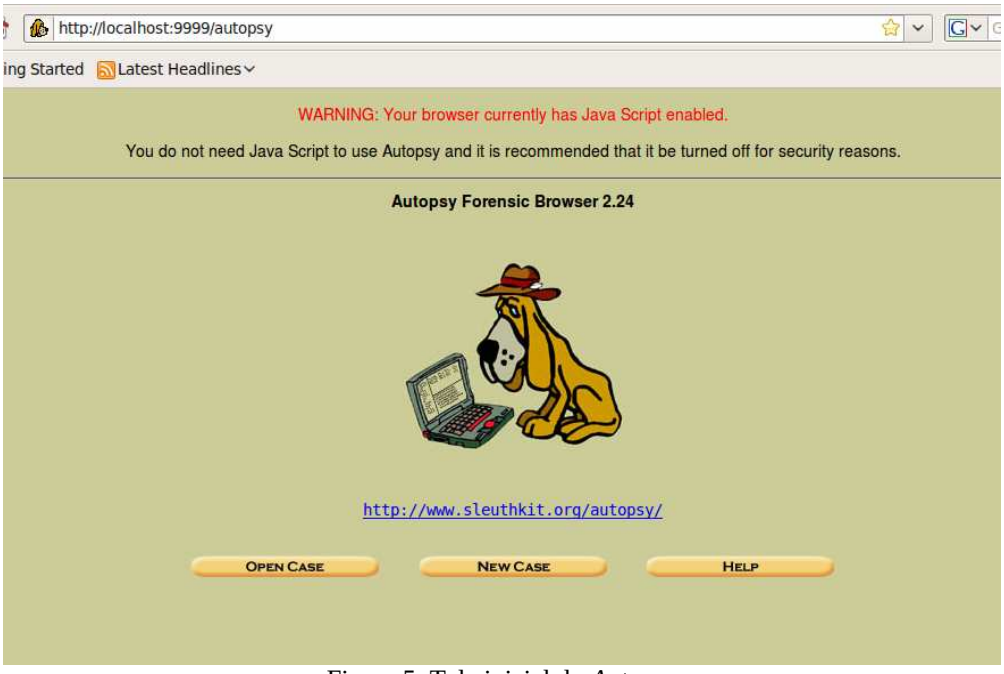

Figura 5. Tela inicial do *Autopsy*

A partir desta tela é possível criar um novo caso, abrir um já existente ou obter ajuda para a utilização do *software*. Para uma demonstração simples da sua utilização será criado um novo caso. A Figura 6 mostra a tela de criação de um novo caso.

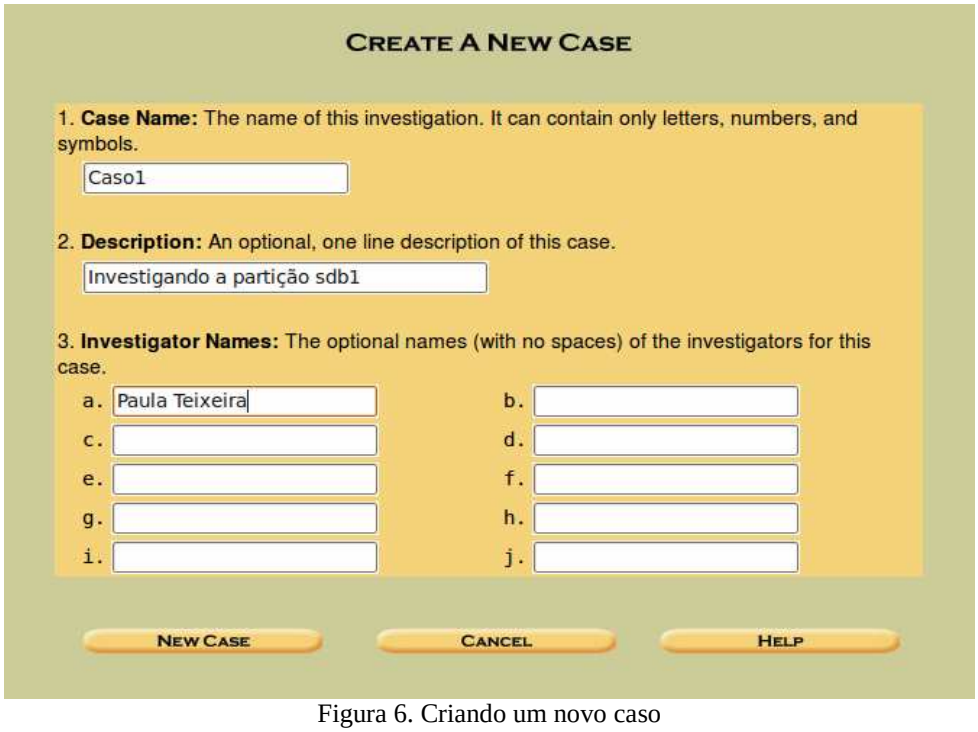

Figura 6. Criando um novo caso

Nesta tela é definido o nome do caso, neste exemplo chamado de Caso 1. É possível adicionar uma descrição para o caso, para que fique mais fácil a sua identificação posteriormente. É definido também o nome dos investigadores envolvidos.

Após preencher estas informações o *software* informa onde foi criado o diretório onde serão armazenados todos os arquivos referentes ao caso, como demonstrado na Figura 7.

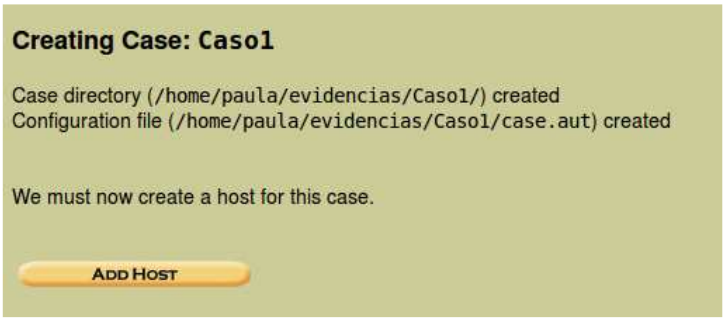

Figura 7. Criação do diretório para armazenamento dos arquivos pertinentes ao caso

Informados os dados para a criação do caso e definido o diretório de armazenamento, o software solicita a identificação do *host* a ser analisado. O *host* refere-se ao computador a ser investigado, sendo possível vários computadores serem investigados em um mesmo caso. A Figura 8 demonstra a tela de identificação do *host*.

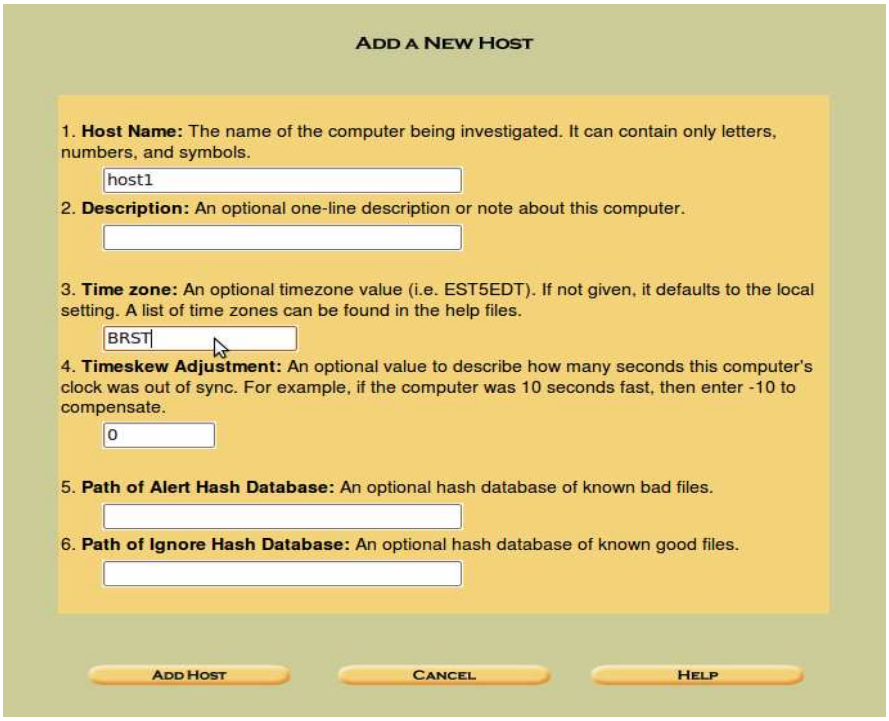

Figura 8. Criando um *host*

Nesta etapa é informado o nome do *host*, ou seja o nome do computador a ser investigado. É informado também o fuso horário (*time zone)* do local onde será feita a investigação. É possível nesta etapa informar quantos segundos o *clock* computador a ser analisado está atrasado. Também é possível informar o diretório de um arquivo *hash* que contenha uma lista de arquivos corrompidos. Esta opção também existe para os arquivos bons, que sejam de conhecimento do investigador.

Após a criação do *host*, o *Autopsy* solicita a inserção de uma imagem para a análise. A Figura 9 mostra a tela onde o *software* apresenta a confirmação da criação do *host* bem como o seu diretório e a solicitação da imagem a ser analisada.

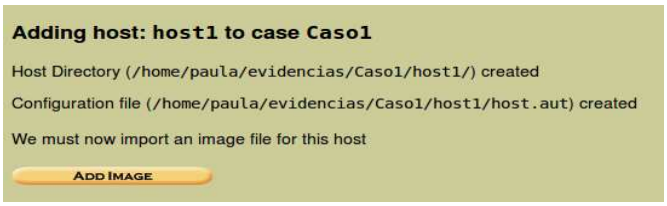

Figura 9. Confirmação de criação do *host* e solicitação da imagem

A imagem faz referência ao computador a ser analisado, ou uma partição de um disco, que deve ser copiada *bit* a *bit*. Para este exemplo foi utilizada a ferramenta *data dumper* (dd), que é nativa do Linux. Esta ferramenta é utilizada via terminal por meio do comando:

dd if=/dev/sdb1 of=/home/paula/imagem.img

O primeiro parâmetro do comando faz referência ao que será copiado, neste caso a partição sdb1, onde *if* significa *input file*. A segunda parte faz referência ao arquivo de destino, para onde será copiada a partição, gerando um arquivo com a extensão que o investigador desejar. Neste exemplo foi escolhida a extensão *img*, apenas para fins de padronização. O parâmetro *of* significa *output file,* fazendo referência à imagem de destino.

Após a execução deste comando, que deve ser executado com permissões de *root*, ou administrador do sistema, é inicializada a cópia *bit* a *bit* da partição ou disco informado. Quando o processo de cópia terminar, é mostrado no próprio terminal o tamanho da imagem

em *bytes*, entre parênteses o tamanho em *gigabytes*, o tempo total decorrido entre o inicio e o término da cópia e a velocidade em que foi copiada. A Figura 10 mostra a tela do terminal após a cópia da imagem ser completada.

Figura 10. Criando uma imagem com o dd

Tendo a imagem pronta, é necessário informar ao *Autopsy* o diretório onde ela se encontra. Na próxima tela, demonstrada na Figura 11, o *software* apresenta opções de adicionar uma imagem, bem como verificar a sua integridade, caso já tenha sido inserida uma. Também é possível verificar informações pertinentes ao caso, tais como o arquivo de atividades em ordem cronológica, seqüência de eventos (que são cadastrados pelo próprio investigador), bancos de dados *hash*, anotações feitas pelos envolvidos na investigação e a opção de ajuda.

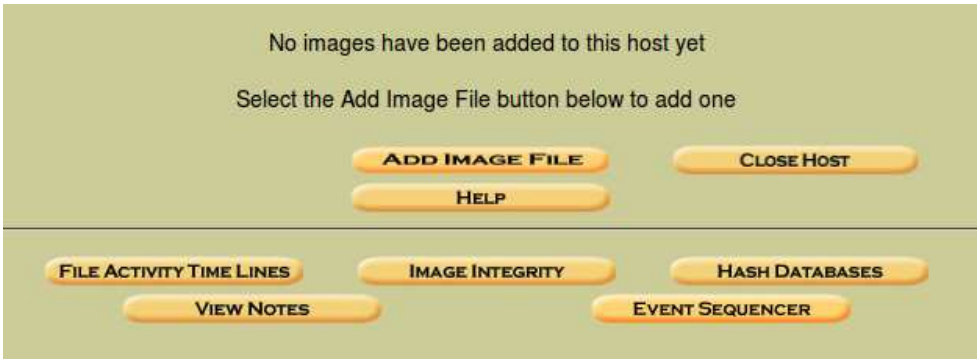

Figura 11. Tela para inserção de uma imagem

A Figura 12 mostra a tela do *Autopsy* onde é inserido o diretório da imagem. Nesta tela é preciso informar ao *software* se a imagem adicionada faz referência à uma partição de um disco ou um disco inteiro.

A próxima opção é selecionar o método que o *Autopsy* utilizará para adicionar a imagem ao caso: *symlink*, onde o *software* se conectará à imagem apenas por um *link* simbólico, sem modificar a sua localização; *copy*, onde o *software* irá copiar a imagem para o diretório do caso; e *move*, onde o *software* irá mover a imagem do seu diretório original para o diretório do caso. Esta última opção não é indicada, pois podem ocorrer erros durante o processo de remoção e danificar a imagem.

Para este exemplo foi selecionada a opção de copiar, como poder ser visualizada na Figura 12.

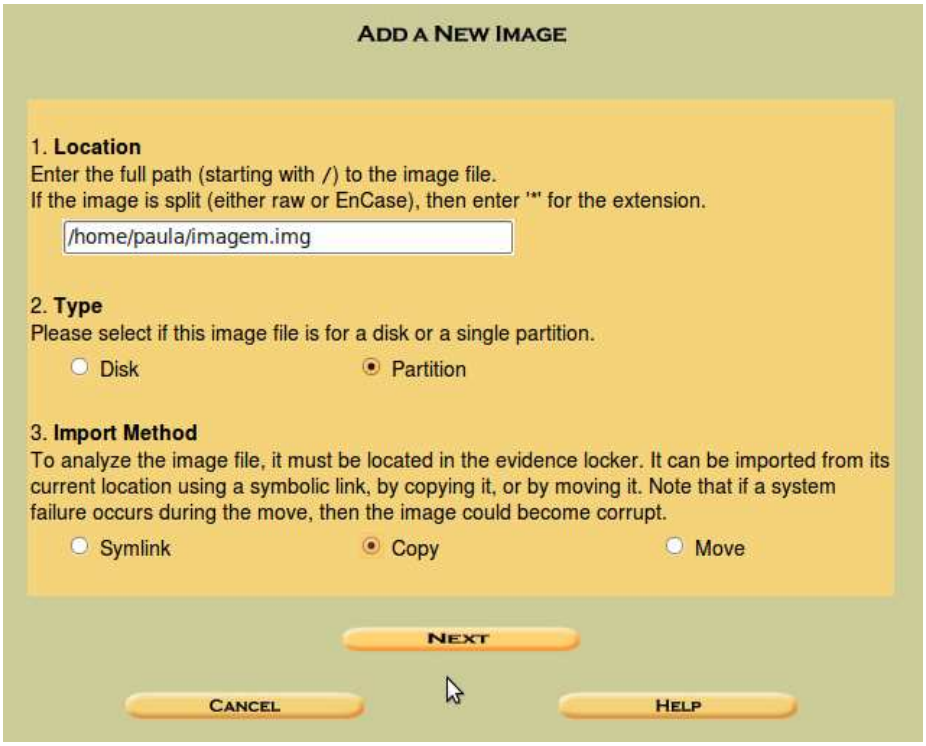

Figura 12. Informando ao *Autopsy* o diretório da imagem

Após informar o diretório da imagem, o *Autopsy* ainda oferece a opção de verificar a integridade da imagem com a ferramenta MD5, que faz parte do conjunto de ferramentas contidas no *Sleuth Kit*. Também é possível calcular o *hash* da imagem depois de

adicionar a imagem ao caso. A Figura 13 apresenta a tela do *software* onde são inseridas estas informações.

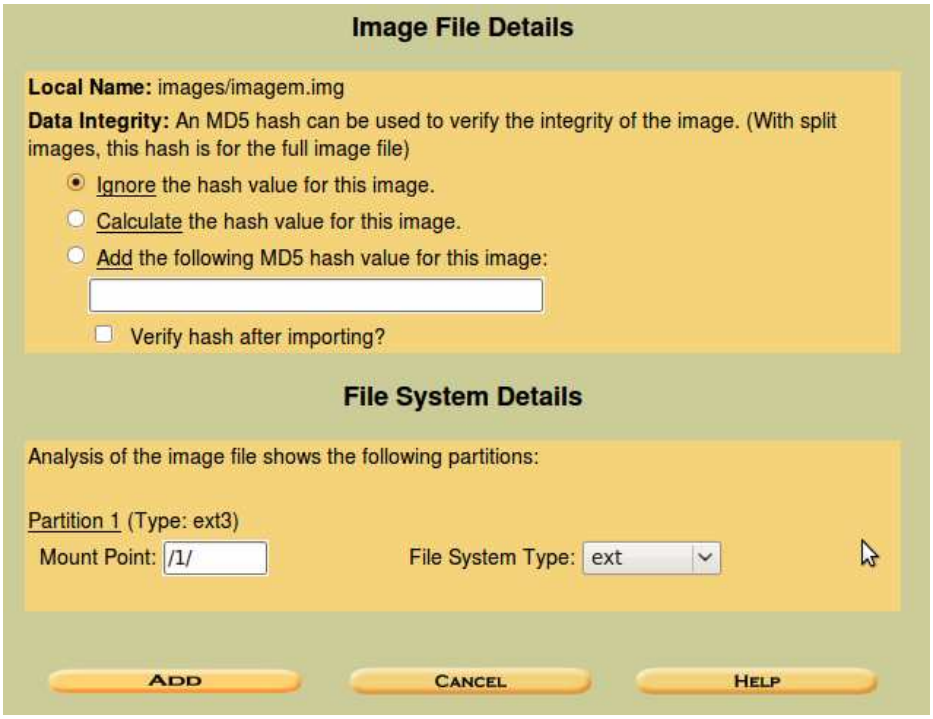

Figura 13. Tela com a opção de calcular o *hash* da imagem

Depois de selecionar a opção de calcular o *hash* da imagem ou não, e clicar no botão *ADD*, o *Autopsy* apresentará uma tela de progresso. O tempo de duração desta etapa depende principalmente do tamanho da imagem e de onde ela se encontrava originalmente.

No caso deste exemplo, a imagem tinha o tamanho de 4 GB e se encontrava na mesma partição que estava sendo utilizada pelo sistema operacional atual, portanto esta etapa foi rápida, durando em torno de cinco minutos.

Quando o *Autopsy* terminar de adicionar a imagem e testá-la, ele apresentará uma tela confirmando a inserção da imagem ao caso e a opção de adicionar mais imagens, como pode ser visualizada na Figura 14.

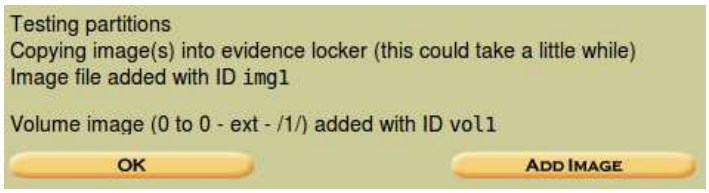

Figura 14. Confirmação de inserção da imagem ao caso

Após esta etapa, o *software* retorna para a tela de inserção de imagens já vista na Figura 11, porém, como já foi inserida uma imagem, estão disponíveis as opções de analisar a imagem adicionada, bem como visualizar a galeria de casos, galeria de *hosts* e manipular os *hosts* do caso. A Figura 15 apresenta a tela do *Autopsy* com estas opções.

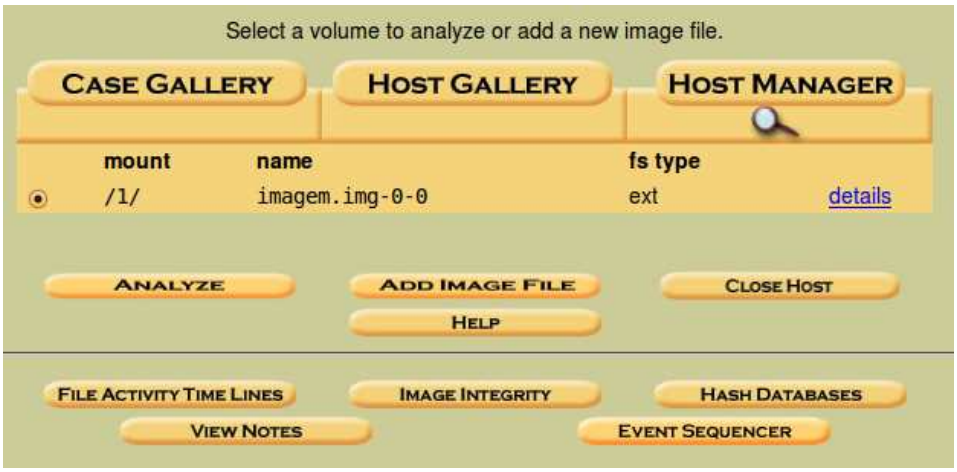

Figura 15. Lista de imagens adicionadas ao caso

Tendo uma imagem adicionada ao caso, é possível realizar a sua análise. Clicando no botão *ANALYZE*, o *Autopsy* apresentará uma tela, que pode ser visualizada na Figura 16, que contém as opções de análise.

Figura 16. Opções de análise que o *Autopsy* possui

A partir desta tela que inicia-se a análise forense. As opções que o *Autopsy* apresenta são:

a) *file analysis*: nesta opção é possível verificar arquivos excluídos, arquivos que foram apenas modificados, qual o seu diretório, visualizar somente arquivos excluídos, visualizar somente um diretório e a movimentação de arquivos dentro dele e até buscar por um arquivo em específico por meio de uma expressão regular. O investigador pode também adicionar uma anotação

referente aos arquivos e gerar o *hash* da lista de arquivos para futuras verificações. A Figura 17 mostra a tela do *Autopsy* onde contém estas opções;

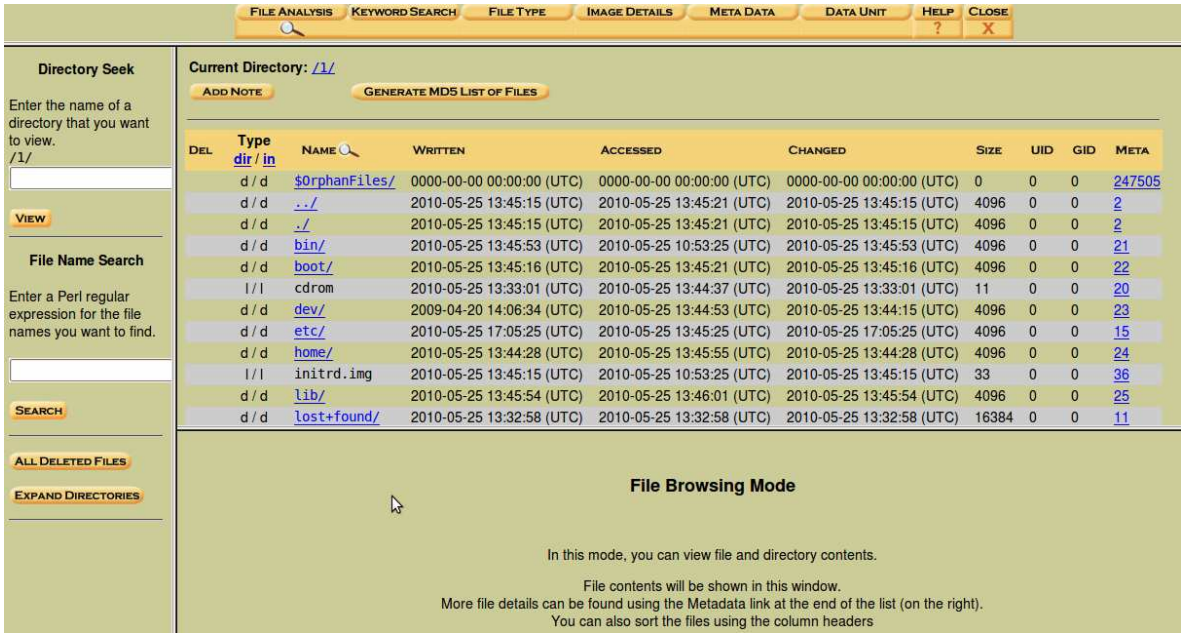

Figura 17. Analisando arquivos modificados ou excluídos

b) *keyword search*: esta opção busca, dentro dos arquivos, qualquer palavra informada pelo investigador. Possui a opção de diferenciar maiúsculas e minúsculas, buscar palavras que utilizando ASCII ou *Unicode*, ou os dois e realizar buscas pré-definidas, como é possível visualizar na Figura 18;

Figura 18. Tela com as opções de busca por palavra-chave

c) *file type*: este item realiza uma análise em arquivos alocados e não alocados e organiza-os por categorias e por extensão. Com esta opção é possível descobrir arquivos ocultos, pois o *software* faz uma varredura pela imagem toda. A Figura 19 mostra o resultado de uma busca organizado por categorias;

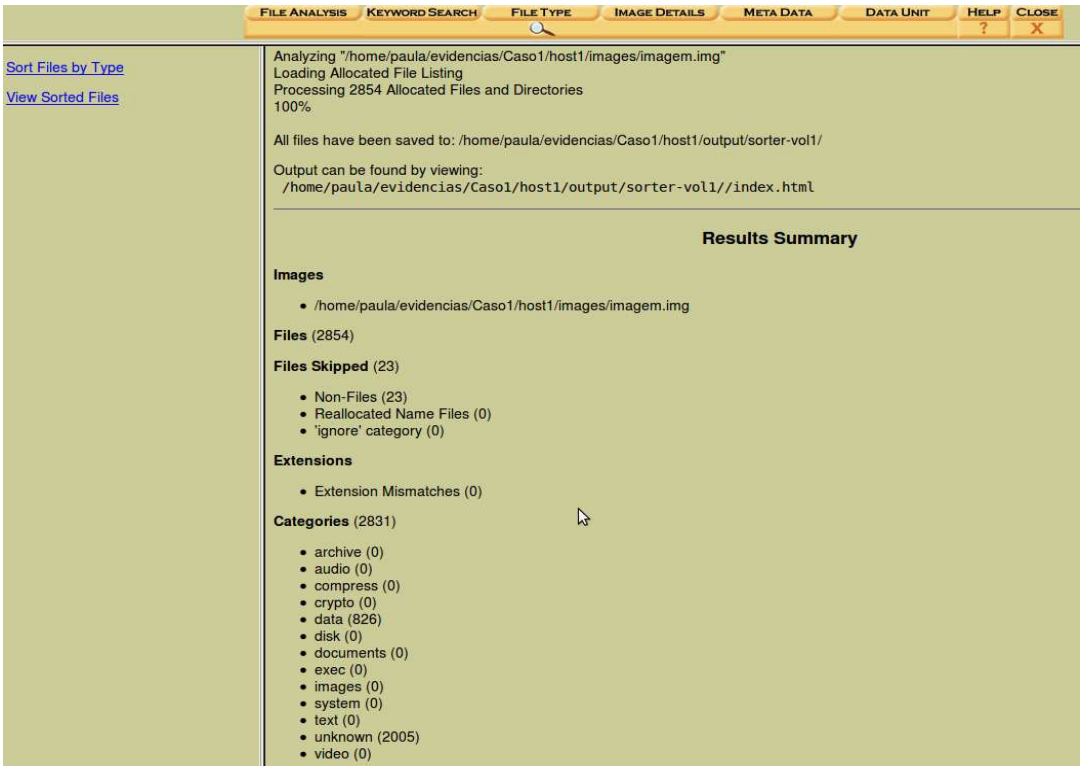

Figura 19. Resultado da busca organizado por categorias

d) *image details*: nesta opção o *software* realiza uma análise aprofundada da imagem, mostrando informações sobre o sistema de arquivos, entre outras informações que podem ser visualizadas na Figura 20;

Figura 20. Detalhes da imagem

e) *meta data*: esta opção permite ao investigador visualizar qualquer informação sobre qualquer *inode*, ou nó-i, ou *index node*, que nada mais é do que uma estrutura de dados que armazena informações sobre um arquivo, tais como o seu dono, permissões e a sua localização. É possível visualizar informações sobre um nó em especifico ou visualizar a lista de nós. O resultado de uma consulta por um determinado nó pode ser conferido na Figura 21;

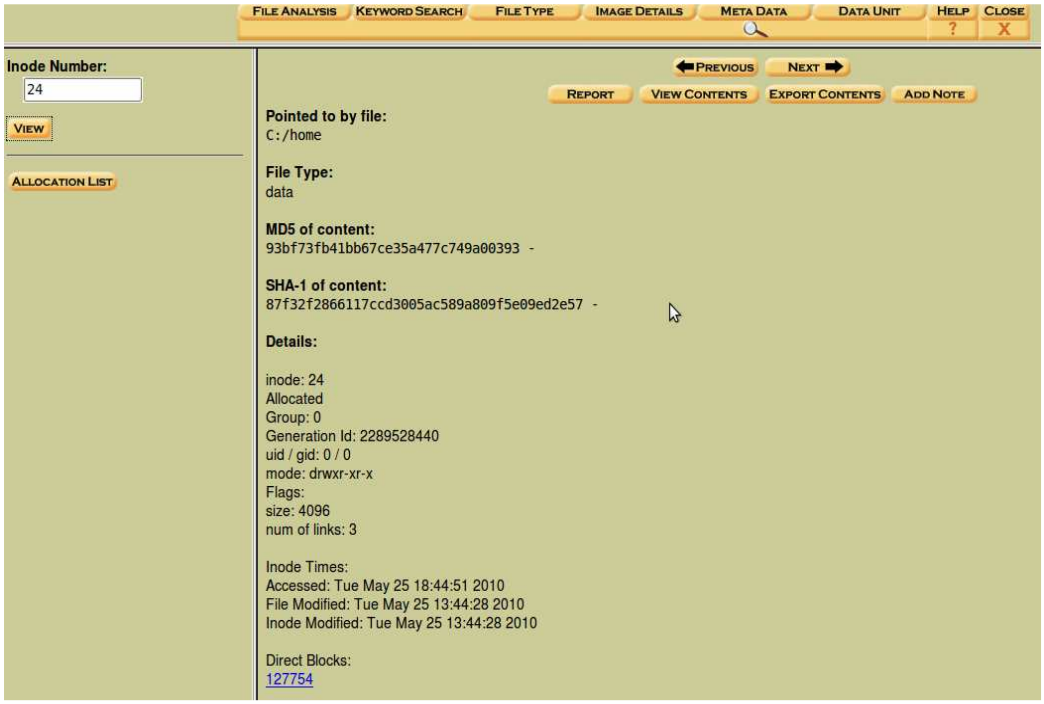

Figura 21. Detalhes de um nó específico

f) *data unit*: nesta opção é possível visualizar o conteúdo de qualquer fragmento no sistema de arquivos, basta indicar o número do fragmento a ser visualizado ou abrir a lista de arquivos alocados e o *software* trará o conteúdo do fragmento, como pode ser observado na Figura 22.

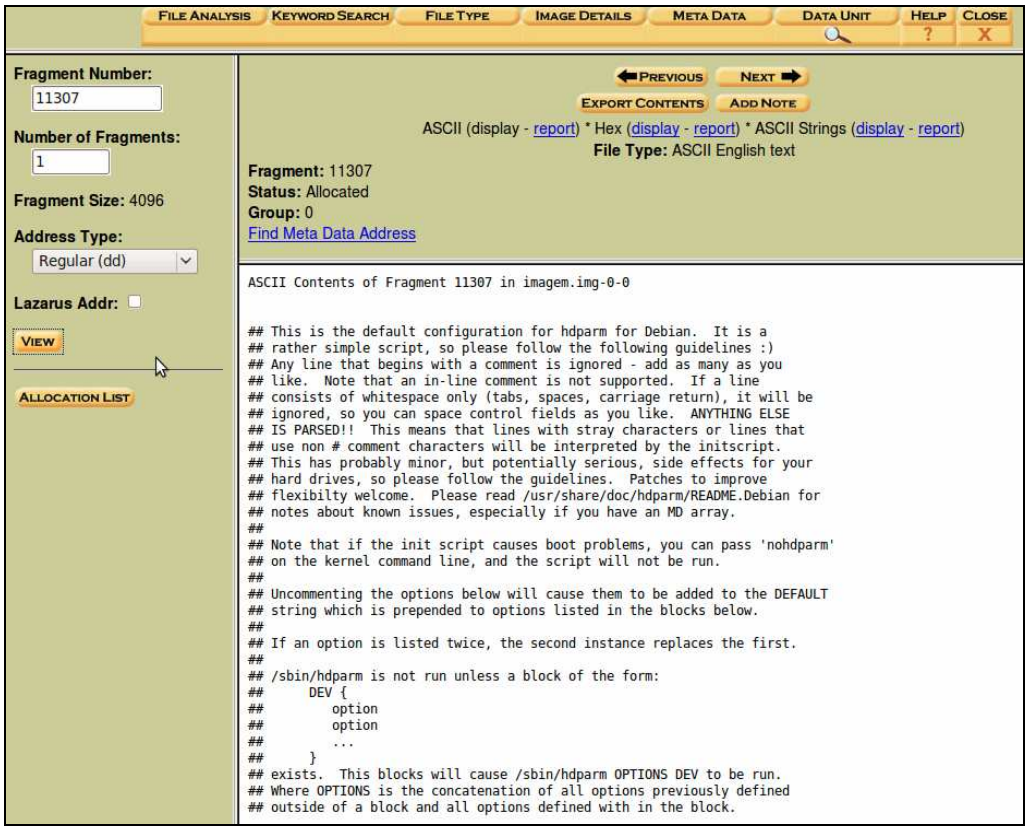

Figura 22. Visualização do conteúdo de um fragmento do sistema de arquivos

Estas são as opções que o *Sleuth Kit* juntamente com o *Autopsy* disponibiliza. As telas apresentadas nas figuras anteriores é uma demonstração de utilização básica da ferramenta, deixando claro que não é a única forma de utilização. O exemplo apresentado é somente para fins de apresentação da ferramenta.

5.1.2 SMART Linux

Esta ferramenta, desenvolvida e comercializada pela empresa *ASR Data*, possui uma versão para ambiente Windows e Linux. Pode ser instalada em uma máquina que já possua um sistema operacional ou utiliza-la em um *live-CD*, pois também possui esta opção.

Após a sua instalação, é necessário executar, com privilégios de *root*, ou administrador do sistema, o executável criado após a instalação, que se encontra no diretório */usr/local/bin*. Para isso é necessário executar via terminal, como é demonstrado na Figura 23.

Figura 23. Executando o SMART via terminal

Informando a senha de *root*, o aplicativo será inicializado. Primeiramente será feito um levantamento dos sistemas de arquivos que estão sendo utilizados no computador e depois irá para a tela de identificação de usuários, que pode ser visualizada na Figura 24.

Figura 24. Tela inicial do SMART

Nesta tela é possível criar um novo usuário ou entrar com um já criado. A próxima tela, que pode ser conferida na Figura 25, mostra todos os tipos de sistemas de arquivos utilizados na máquina, bem como a utilização do disco por cada um. Também mostra os espaços não alocados de cada disco.

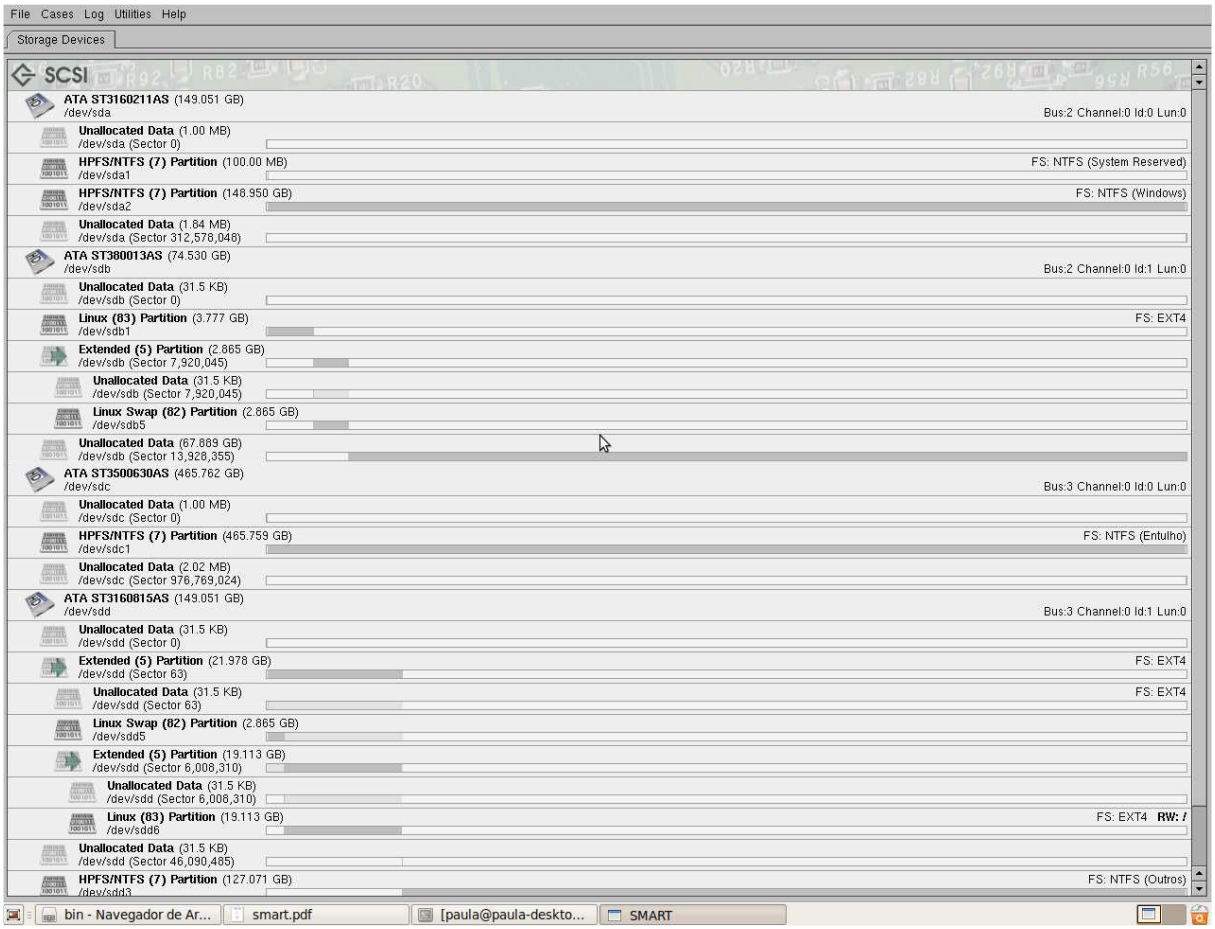

Figura 25. Levantamento dos sistemas de arquivos separados por disco

O primeiro passo para o início de uma investigação é a criação de um caso. No menu *Cases*, executando a primeira opção, que é *New Case*, uma janela é aberta solicitando o nome do caso, como poder ser observado na Figura 26.

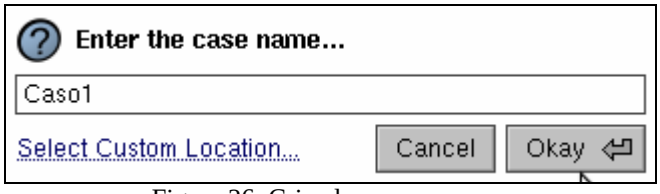

Figura 26. Criando um novo caso

Informando o nome do caso, uma nova aba irá se abrir, correspondendo ao caso recém criado. Nesta tela é possível inserir imagens ao caso, como é demonstrado na Figura 27. Para inserir uma imagem é preciso clicar com o botão direito do *mouse* e selecionar a opção *Import Image*. Selecionando esta opção uma janela se abrirá solicitando o nome da

imagem, caso esteja no mesmo diretório que o SMART, ou o diretório de onde se encontra a imagem, clicando com o botão direito do *mouse* novamente e selecionando a opção *Add File*. Então irá aparecer uma outra janela com todos as pastas dentro do diretório raiz. A partir desta tela é necessário selecionar o diretório onde está a imagem a ser analisada e clicar em *Add*.

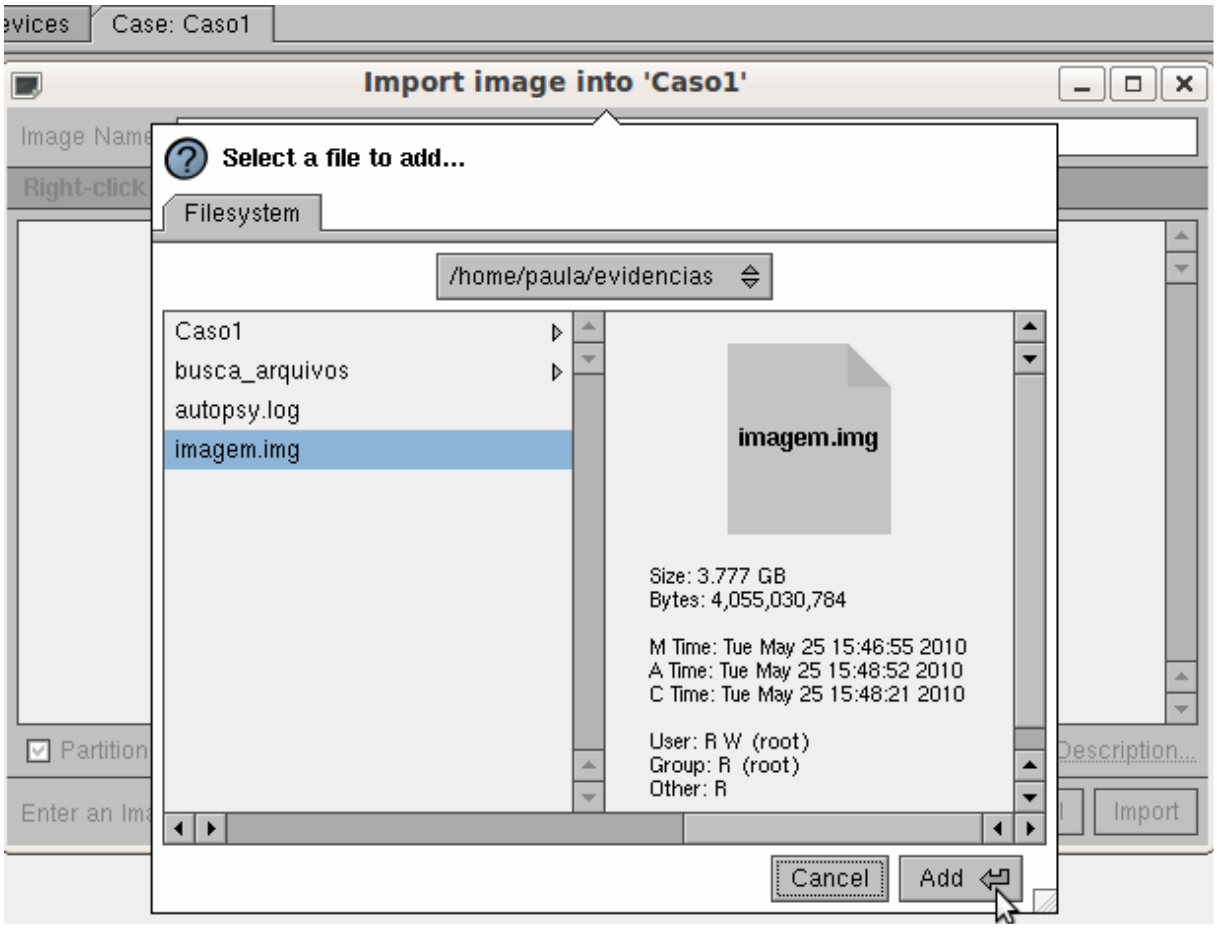

Figura 27. Inserindo uma imagem ao caso

Com o diretório de uma imagem já informado ao *software*, ele irá retornar à tela de inserção de imagem. A partir deste ponto é possível inserir mais imagens ou somente importar a que foi informada, apenas clicando no botão *Import*.

Tendo uma imagem adicionada ao caso é possível realizar a sua análise, apenas clicando com o botão direito do *mouse* sobre a imagem. As opções de análise podem ser observadas na Figura 28.

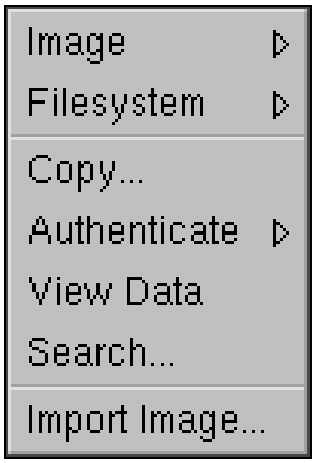

Figura 28. Opções de análise

A partir deste menu é possível realizar as seguintes opções:

a) *Image → Get Info*: traz as principais informações acerca da imagem. O menu para selecionar esta opção é demonstrado na Figura 29.

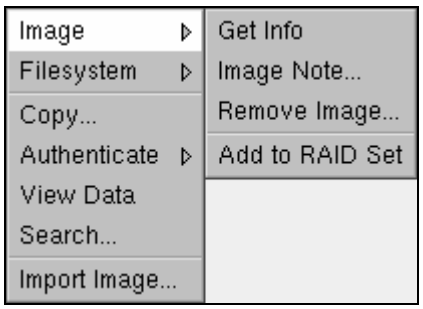

Figura 29. Menu *Get Info*

E as informações que esta opção traz são demonstradas na Figura 30;

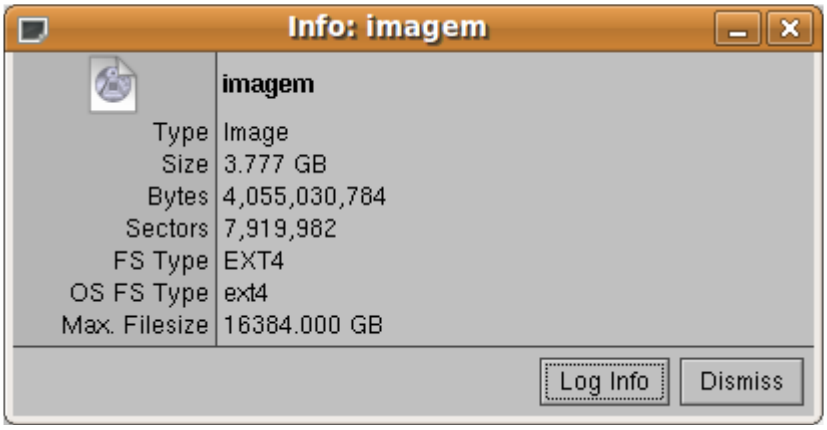

Figura 30. Detalhes da imagem

b) *Image → Image Note*: adiciona uma anotação ao caso, que podem ser visualizadas posteriormente no menu *Log*;

c) *Image → Remove Image*: opção para remover a imagem do caso atual;

d) *Image → Add to RAID Set*: selecionando esta opção o SMART irá determinar a configuração RAID para os dados da imagem. É possível adicionar várias imagens;

e) *Filesystem → Mount*: com esta opção é possível montar a imagem para que ela fique disponível para a visualização dos arquivos. O menu que apresenta esta opção está disponível na Figura 31.

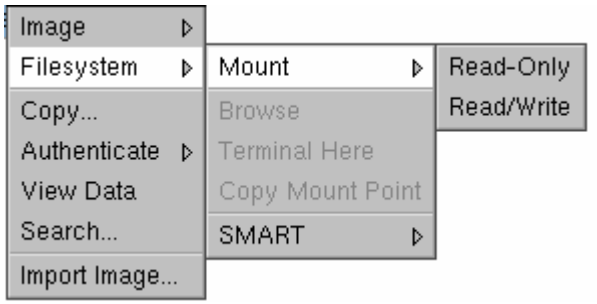

Figura 31. Montando a partição

Depois de montada, tanto no modo de apenas leitura *(Read-Only)* como o de leitura e escrita *(Read/Write)*, o menu de *Filesystem* muda, apresentado novas opções para a imagem. O novo menu pode ser visualizado na Figura 32.

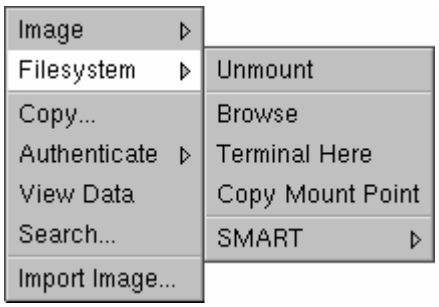

Figura 32. Menu com opções para a imagem montada

A partir deste menu é possível abrir a partição como um diretório do próprio sistema (*Browse*), abrir um terminal para executar comandos específicos dentro

da imagem montada *(Terminal Here)*, copiar o ponto de montagem *(Copy Mount Point)* e desmontar a imagem;

f) *Filesystem → SMART → Study*: esta opção possibilita ao investigador extrair dados de uma imagem de um disco ou partição, tanto arquivos excluídos como ativos. Também realiza um estudo detalhado do sistema de arquivos da imagem, partição ou disco selecionado;

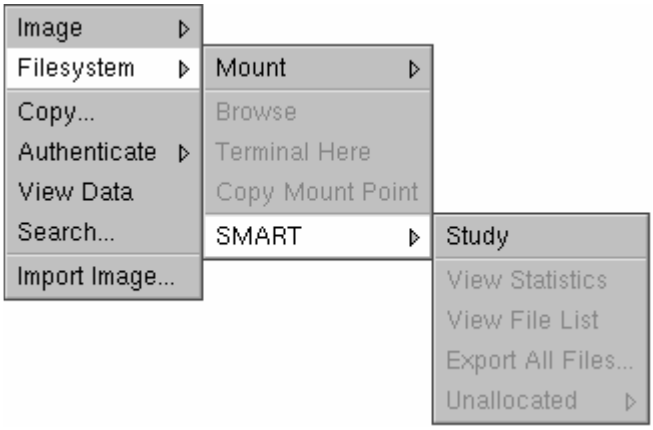

Figura 33. Estudando a imagem

g) *Copy*: este item possibilita a cópia fiel da imagem para um outro diretório, com as opções de calcular o *hash* utilizando as ferramentas SHA1, MD5 ou CRC32, ou todos os três juntos. Também possibilita realizar uma compressão durante a cópia, podendo escolher entre três ferramentas de compressão: *EXCompress*, *BZip2* ou *GZip*. Outra opção disponível é a possibilidade de restauração completa da imagem para qualquer partição disponível. A Figura 34 mostra como é a tela de seleção das preferências da cópia;

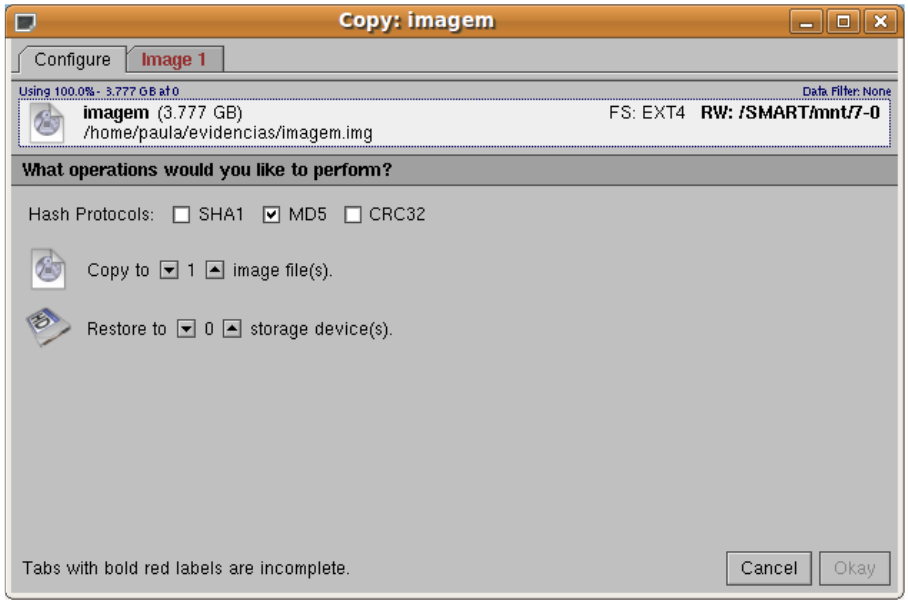

Figura 34. Tela de seleção das preferências da cópia

h) *Authenticate → Produce Hash*: calcula o valor *hash* da imagem, podendo escolher entre as ferramentas SHA1, MD5 ou CRC32, ou utiliza-las juntas. O menu para a escolha desta opção pode ser observado na Figura 35;

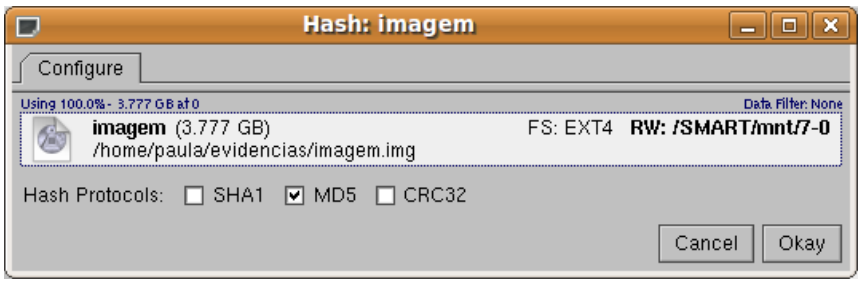

Figura 35. Tela com as opções de ferramentas para gerar o *hash* da imagem

i) *Authenticate → Against a Device*: realiza a comparação entre os valores *hash* da imagem e de uma partição selecionada;

j) *View Data*: traz todos os dados presentes na imagem em hexadecimal e em *Unicode*. Selecionando um *bit* qualquer, todas as informações sobre ele são mostradas, tais como data e hora da sua criação, último acesso, última modificação, atributos e tamanho, além de trazer também a opção de gerar o *hash* de qualquer dado selecionado. A tela com estas informações pode ser visualizada na Figura 36;

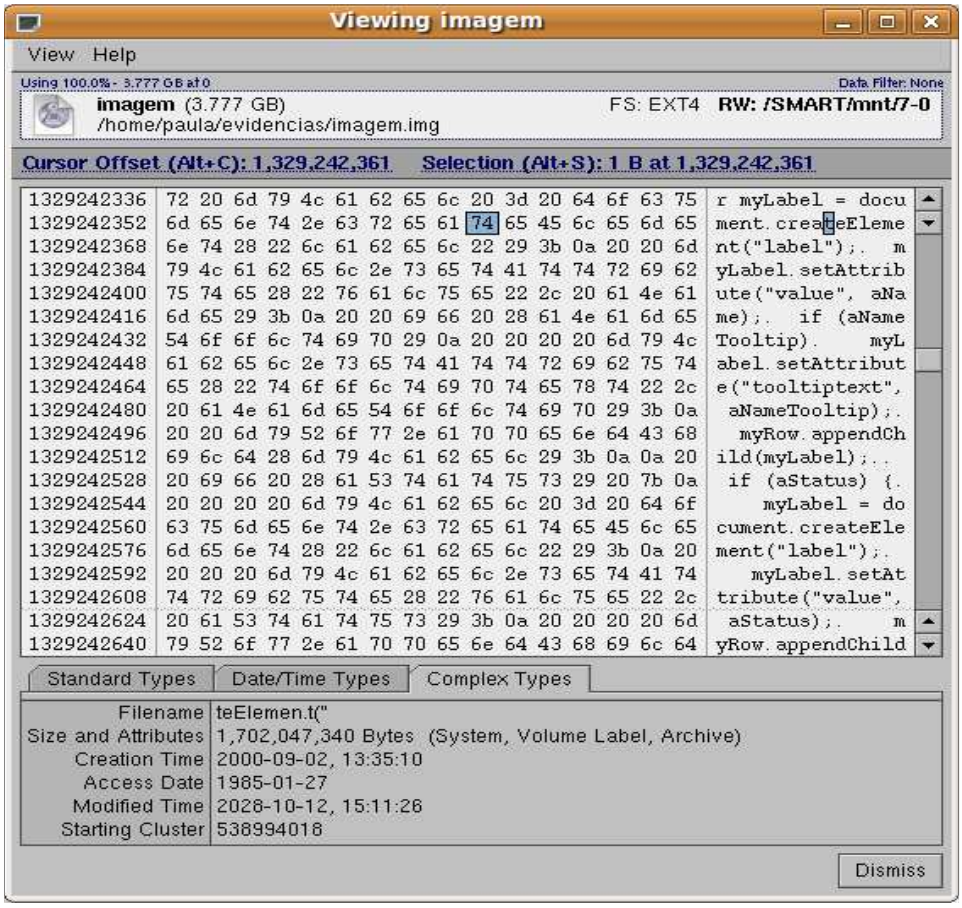

Figura 36. Tela com informações sobre todos os dados contidos na imagem

l) *Search*: selecionando este item é possível realizar uma busca dentro da imagem. A busca pode ser por um termo simples, como uma palavra, ou realizar buscas pré-definidas, como arquivos JPG, vídeos ou documentos do *Open Office*. Também busca informações de cartão de crédito, sendo possível selecionar a bandeira do cartão, busca por IP no histórico de acessos à internet, entre outras várias opções. A Figura 37 apresenta as opções de busca pré-definidas, selecionando a opção de busca por informações de cartão de crédito, por exemplo.

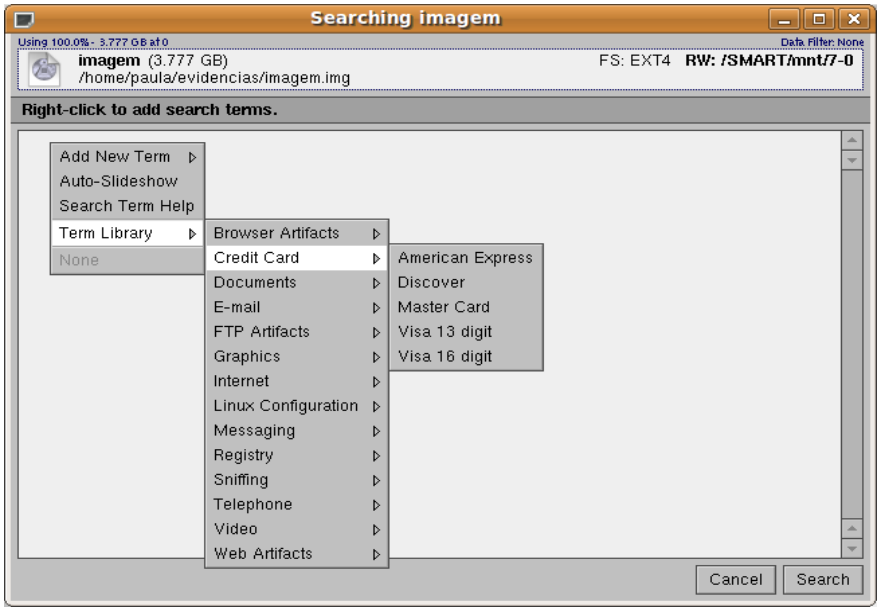

Figura 37. Opções de busca pré-definidas

Além destas opções de análise, o SMART traz um menu com opções para visualizar anotações feitas durante a investigação.

Estes são os recursos que o SMART possui na sua forma básica, porém existe ainda a opção de instalação de extensões, aumentando assim o leque de opções que esta ferramenta pode apresentar.

6 TRABALHOS CORRELATOS

Com o aumento e popularização dos recursos computacionais atuais, as estatísticas de incidentes de segurança só tendem a crescer. Para poder responder à um incidente de segurança com prudência, a é necessário que sejam utilizadas técnicas de perícia forense para que as provas sejam coletadas corretamente, sem danificá-las.

A área de perícia forense vem se tornando cada vez mais requisitada, fazendo com que pesquisas e desenvolvimento de ferramentas sejam cada vez mais necessárias.

Neste capítulo serão citados alguns trabalhos envolvendo a perícia forense computacional.

6.1 PERÍCIA FORENSE EM SISTEMAS GNU/LINUX

Monografia apresentada ao Curso de Pós-graduação em Segurança de Redes de Computadores, como requisito parcial para obtenção do título de Especialista em Segurança de Redes de Computadores, realizada em 2007, na Faculdade Salesiana de Vitória.

Este trabalho tem como objetivo apresentar e detalhar o conceito de perícia forense. Detalha, ao longo do trabalho, conceitos como evidências digitais e perícia forense, os tipos mais comuns de ataques à computadores e o processo de investigação em si. E também sugere softwares voltados a perícia forense e métodos para que a coleta e investigação das evidências seja realizada sem correr o risco de contaminação das provas (OLIVEIRA, 2007).

6.2 TÉCNICAS COMPUTACIONAIS NO AUXÍLIO À PERÍCIA FORENSE NA ANÁLISE DE EVIDÊNCIAS COLETADAS EM SERVIDORES GNU / LINUX

Trabalho de Conclusão de Curso de Ciência da Computação, como requisito para obtenção do titulo de Bacharel em Ciência da Computação, realizado em 2006, na Universidade do Extremo Sul Catarinense, Santa Catarina.

O trabalho apresenta algumas técnicas e procedimentos para a análise e coleta de evidências em ambientes GNU / Linux. Detalha as etapas necessárias para a realização da perícia forense em servidores GNU / Linux, aplicando técnicas computacionais forenses em um estudo de caso, realizando uma análise forense na memória principal, memória secundária (*Hard Disk*), processos e módulos do *kernel* do GNU / Linux (BERNARDO, 2006).

6.3 PERÍCIA FORENSE COMPUTACIONAL APLICADA A AMBIENTES NEW TECHNOLOGIES FILE SYSTEM (NTFS)

Trabalho de Conclusão de Curso de Ciência da Computação, como requisito para obtenção do titulo de Bacharel em Ciência da Computação, realizado em 2007, na Universidade do Extremo Sul Catarinense, Santa Catarina.

O trabalho apresenta métodos para a preservação, busca e análise das evidências em ambientes NTFS, com o intuito de sugerir um kit de ferramentas para que as mesmas sejam consideradas válidas, seja o caso com finalidade jurídica ou não.

7 ANÁLISE E COMPARAÇÃO DE SOFTWARES PARA PERÍCIA FORENSE EM AMBIENTES LINUX BASEADO NA MÉTRICA *FUNCTION POINT ANALYSIS*

No decorrer deste capítulo é apresentada a análise dos dois *softwares* citados no capítulo 5, utilizando a métrica de *software* Pontos de Função, que foi detalhada no capítulo 4. Também é apresentada uma análise de cada uma das ferramentas escolhidas utilizando como parâmetros recursos exclusivos de *softwares* voltados à perícia forense. A cada recurso existente nas ferramentas é atribuído o valor de um ponto, com o objetivo de realizar uma somatória com os valores resultantes da aplicação da métrica de *software* citada anteriormente.

Cada análise é separada em subtítulos diferentes: um específico para a aplicação da métrica de *software* e outro para a análise dos recursos exclusivos de *softwares* de perícia forense. Ao final de cada subtítulo será apresentado o valor resultante de cada ferramenta.

O terceiro subtítulo será a apresentação dos resultados. Nesta etapa que será realizado o estudo comparativo, que é o objetivo geral deste trabalho.

7.1 ANÁLISE DAS FERRAMENTAS UTILIZANDO A MÉTRICA FPA

A primeira ferramenta analisada foi *The Sleuth Kit* utilizando a interface gráfica *Autopsy*. Seguidamente foi analisada a ferramenta SMART Linux.

A análise utilizando a métrica FPA foi realizada em três etapas: primeiramente foi realizada a contagem de funções de dados, em seguida a contagem de funções de transação e por último a determinação do valor do fator de ajuste. Para esta última foi obtido o valor analisando quatorze características gerais do sistema, as CGS, que também estão previstas na métrica FPA.

7.1.1 Análise da ferramenta *The Sleuth Kit* e *Autopsy*

7.1.1.1 Contagem de Funções de Dados

A primeira etapa da aplicação da métrica FPA consiste em contabilizar pontos de acordo com a contagem de funções de dados. De acordo com a métrica, é necessário identificar no *software* analisado os Arquivos Lógicos Internos, ou ALI, e os Arquivos de Interface Externa, ou AIE. Como esta ferramenta não trabalha em conjunto com nenhuma outra e as informações por ela salvas, apesar de ser possível serem visualizadas por outras aplicações, são mantidas somente pela ferramenta, o *Sleuth Kit* não possui AIE, somente ALI.

Após identificar todos os ALI e AIE na ferramenta, a métrica FPA prevê identificar todos os Elementos de Dados (EDD) e Elementos de Registro (ER) relacionados aos ALI e AIE identificados.

A Tabela 26 apresenta todos os ALI, EDD e ER identificados na ferramenta analisada.

Tabela 25. Descrição de todos os ALI, EDD e ER identificados na aplicação *Sleuth Kit*

ALI identificados	EDD relacionados	ER relacionados
Cadastro de um caso	Nome do caso Nome dos investigadores	Descrição do caso
Cadastro de um *host*	Nome do *host*	Descrição do *host* Fuso horário Ajuste de atraso do *clock* Diretório de um banco de dados *hash* de arquivos ruins Diretório de um banco de dados *hash* de arquivos bons
Cadastro de uma imagem	Localização Tipo da imagem (disco ou partição) Método de importação	
Criação de um arquivo de dados	Selecionar a imagem Selecionar arquivos alocados ou não alocados Nome do arquivo de saída	Opção de gerar valor *hash*
Criação de um arquivo de *time line*	Selecionar o arquivo de *body* Nome do arquivo de *time line* Escolher o formato do arquivo de saída	Selecionar data de início para análise Selecionar data de término para análise Selecionar uma imagem UNIX Opção de gerar valor *hash*
Cadastro de eventos	Descrição do evento Data do evento Origem do evento	
Cadastro de anotações	Descrição da anotação Escolha da data	
Total	**17**	**11**

Para realizar a atribuição de valores é necessário realizar a conversão da quantidade de EDD e ER identificados de acordo com os critérios de complexidade funcional descritos no item 4.3.1 do capítulo sobre métricas. De acordo com a Tabela 4, que se encontra no mesmo item citado, o grau de complexidade funcional é MÉDIO.

Para realizar a contagem de funções de dados é necessário que seja transformado o grau de complexidade funcional obtido em Pontos de Função Não Ajustados (PFNA). De

acordo com a Tabela 5, o valor numérico correspondente ao grau de complexidade MÉDIO é dez (10).

7.1.1.2 Contagem de Funções de Transação

A próxima etapa é realizar a contagem de funções de transação. Para o tal é preciso identificar na aplicação todas as Entradas Externas (EE), Saídas Externas (SE) e Consultas Externas (CE) e os EDD e os Arquivos Relacionados (AR) à eles relacionados. A Tabela 27 mostra todas as EE encontrados e os EDD e AR à eles relacionados.

Tabela 26. Descrição de todos os EE, EDD e AR identificados na aplicação *Sleuth Kit*

EE identificados	**EDD relacionados**	**AR relacionados**
Adição de mais uma imagem ao caso	Localização Tipo da imagem (disco ou partição) Método de importação	Imagem *Log* de utilização do software
Inserção de mais um *host* ao caso	Nome do *host*	*Log* de utilização
Cadastro de um evento	Descrição do evento Data do evento Origem do evento	Lista de eventos Arquivo que contém o evento
Cadastro de uma anotação	Descrição da anotação Escolha da data	Lista de anotações Arquivo que contém a anotação
Busca por arquivos alocados e não alocados	Opção de organizar os arquivos em categorias pelo tipo Opção de salvar a cópia das imagens Opção de não salvar informação sobre arquivos desconhecidos Opção de salvar somente imagens	Arquivo de saída contendo o resultado
Total	**13**	**8**

De acordo com a Tabela 7, que descreve os critérios para a avaliação da complexidade funcional de EE, o grau atingido é ALTO. Transformando em PFNA, de acordo com a Tabela 10, o valor obtido é seis (6).

A Tabela 28 mostra todas as SE identificadas na aplicação, bem como os EDD e AR a eles relacionados.

Tabela 27. Descrição de todos os SE, EDD e AR identificados na aplicação *Sleuth Kit*

SE identificadas	EDD relacionados	AR relacionados
Cálculo do valor *hash* da imagem	Seleção da imagem	Imagem
Validação do *hash* do arquivo de *body*	Seleção do arquivo de *body*	Arquivo de *body*
Validação do *hash* do arquivo de *time line*	Seleção do arquivo de *time line*	Arquivo de *time line*
Geração do valor *hash* para a lista de arquivos exibidos em *file analysis*	Seleção de arquivos para a geração do valor de *hash*	Arquivo que contém o valor *hash* Arquivos selecionados
Total	**4**	**5**

De acordo com a Tabela 8, que descreve os critérios para a avaliação da complexidade funcional de SE, o grau atingido é MÉDIO. Realizando a conversão em valores de PFNA, de acordo com a Tabela 9, o resultado obtido é cinco (5).

A Tabela 29 mostra todos as CE identificadas e os EDD e AR à elas relacionadas.

Tabela 28. Descrição de todos os CE, EDD e AR identificados na aplicação *Sleuth Kit*

CE identificadas	EDD relacionados	AR relacionados
Análise de todos os arquivos contidos na imagem	Nome de um diretório a ser consultado Expressão regular para busca de nomes de arquivos Visualização de todos os arquivos excluídos	Imagem
Busca por palavras chave	Palavra para a busca Escolha entre ASCII ou *Unicode* ou os dois Escolha se a busca é *case sensitive* Escolha se a busca é uma expressão regular Buscas pré-definidas	Imagem
Detalhes da imagem	Informações detalhadas sobre a imagem	Imagem
Visualização de meta dados	Opção de visualizar um nó específico Opção de visualizar a lista de alocação com os valores de identificação dos nós Visualização do conteúdo de um nó	Imagem
Visualização do conteúdo de um fragmento	Opção de especificar o fragmento Quantidade de fragmentos a busca retornará Tipo do endereço Opção de utilizar a ferramenta Lazarus Opção de visualizar a lista de alocação	Imagem
Visualização do arquivo de *time line*	Seleção da data de visualização	Arquivo de *time line*
Visualização das anotações		Anotações
Total	**18**	**7**

De acordo com a Tabela 8, que descreve os critérios para a avaliação da complexidade funcional de CE, o grau atingido é ALTO. Realizando a transformação em valores de PFNA, de acordo com a Tabela 10, o valor obtido é seis (6).

Após obter os valores resultantes da identificação dos ALI e AIE e os EDD e ER associados, EE, SE e CE e os EDD e AR relacionados, é necessário somá-los entre si para obter o valor total dos PFNA. A próxima etapa é determinar o valor do fator de ajuste, para que no final estes valores sejam aplicados à uma fórmula, prevista na FPA, e obter o valor total da aplicação analisada.

7.1.1.3 Determinação do Valor do Fator de Ajuste (VFA)

Para obter o valor do fator de ajuste é necessário analisar quatorze itens, previstos na FPA, que são as Características Gerais do Sistema (CGS) e atribuir pontos de acordo com o seu nível de influência na aplicação, como descrito no capítulo 4. Após o resultado dos pontos da análise das CGS, para obter o VFA é necessário aplicar a fórmula

$$VFA = (NIT * 0,01) + 0,65$$

onde NIT, que significa Nível de Influência Total, é a soma dos pontos obtidos na análise do Nível de Influência (NI) de cada uma das CGS e os valores 0,01 e 0,65 são índices, definidos na FPA, que podem variar entre 0,65 e 1,35.

A Tabela 30 apresenta os valores de NI obtidos em cada item das CGS e a descrição do valor atribuído.

Tabela 29. Atribuição de valores de acordo com as CGS previstas na FPA

Item das CGS	**Valor atribuído**	**Descrição**
A	4	Mais que um *front-end*, mas a aplicação suportará apenas um tipo de protocolo de comunicação
B	1	A aplicação preparará dados para o usuário final processar em outra CPU da instalação. Por exemplo, planilhas eletrônicas ou gerenciadores de banco de dados de PC
C	3	O tempo de resposta será crítico durante todo o horário de utilização. Não será necessário nenhum procedimento especial para utilização de CPU. Os requisitos de prazo de processamento com outros sistemas são limitados
D	2	Algumas considerações sobre tempo e segurança são necessárias
E	0	Nenhum período de pico de transações é esperado
F	0	Todas as transações serão processadas em modo *batch*
G	4	Apresenta os itens: - Existência de menus; - Ajuda e documentação *on-line*; - Movimento de *scroll* (vertical e horizontal); - Seleção de dados da tela por meio de movimentação do cursor; - Uso intensivo de vídeo reverso, brilho, sublinhado, cores e outros recursos de vídeo; - Possibilidade do uso de *mouse*. - Os requisitos estabelecidos para eficiência do usuário são rigorosos o suficiente para que a fase de projeto da aplicação inclua fatores, tais como: minimizar a digitação, maximizar os valores padrões, utilizar *templates* etc.
H	0	Nenhuma atualização
I	3	Apresenta os itens: - Controle sensível (processamento especial de auditoria) e/ou processamento específico de segurança da aplicação; - Processamento lógico extensivo; - Processamento complexo para manipular múltiplas possibilidades de entrada/saída (múltiplos meios e independência de equipamentos).
J	4	A aplicação será projetada e documentada para facilitar a reutilização de código e a aplicação será customizada pelo usuário a nível do código fonte
L	1	Nenhuma consideração especial foi feita pelo usuário, mas um procedimento especial é requerido para a implantação
M	5	A aplicação será projetada para não precisar de intervenção do operador no seu funcionamento normal. Apenas a inicialização e parada do sistema ficam a cargo do operador. A recuperação automática de erros será uma característica da aplicação
N	0	Nenhuma solicitação do usuário para considerar a necessidade de instalar a aplicação em mais de um local
O	3	Apresenta o item: - Será fornecido recurso de consulta e relatórios flexíveis capaz de manipular solicitações complexas de consulta, com combinações de um ou mais Arquivos Lógicos Internos (contar como três pontos)
Total	**30**	

Com os resultados obtidos na contagem de funções de dados, de transação e na determinação do VFA é possível calcular o valor da aplicação, que será realizado mais adiante.

7.1.2 Análise da ferramenta SMART Linux

7.1.2.1 Contagem de Funções de Dados

Primeiramente, como previsto na métrica FPA, foi realizada a contagem de funções de dados, utilizando as Tabelas apresentadas no capítulo 4 como base. Nesta etapa foram identificados os ALI presentes na aplicação, uma vez que a ferramenta não possui AIE, pois não se comunica com nenhuma outra durante o seu funcionamento.

A Tabela 31 mostra os resultados da análise da aplicação, com todos os ALI identificados e os EDD e ER relacionados.

Tabela 30. Descrição de todos os ALI, EDD e ER identificados na aplicação SMART

ALI identificados	EDD relacionados	ER relacionados
Cadastro de um caso	Cadastro de um usuário Nome do caso	Opção de selecionar um diretório para o armazenamento de arquivos referentes ao caso
Cadastro de uma imagem	Localização da imagem	Selecionar o tipo da imagem (partição ou disco) Descrição da imagem
Cadastro de anotações	Seleção do caso Anotação de sessão	
Cadastro de *log* de usuário	Localização do arquivo Nome do arquivo	
Cadastro de *log* do caso	Seleção do caso Nome do arquivo de *log*	Opção de selecionar um diretório para o armazenamento do arquivo de *log*
Total	**9**	**4**

De acordo com a Tabela 4, após realizar a conversão da quantidade de EDD e ER relacionados à todos os ALI identificados na função, o grau de complexidade funcional atingido é BAIXO. E realizando a transformação em valores de PFNA, de acordo com a Tabela 5, o valor obtido é sete (7).

7.1.2.2 Contagem de Funções de Transação

Nesta etapa foram identificadas todas as EE, SE e CE presentes na aplicação analisada, bem como os EDD e AR associados. A Tabela 32 mostra o resultado da busca por todas as EE presentes na ferramenta SMART e os EDD e AR relacionados.

Tabela 31. Descrição de todos os EE, EDD e AR identificados na aplicação SMART

EE identificados	**EDD relacionados**	**AR relacionados**
Inserir outra imagem	Localização da imagem	Imagem Lista de imagens
Cadastro de uma nota	Descrição da anotação	Anotação Lista de anotações
Cadastro de *log* de usuário	Localização Nome do arquivo	Arquivo de *log* Lista de *logs*
Cadastro de *log* do caso	Seleção do caso Nome do arquivo de *log*	Arquivo de *log* Lista de *logs*
Cadastro de anotação da imagem	Descrição	Arquivo de anotação Lista de anotações
Cadastro de *log* de seleção de dados	Descrição	Arquivos de *log* Lista de *logs*
Total	**8**	**12**

A conversão de todos os EDD e AR relacionados à todas as EE identificadas na aplicação resultou no grau de complexidade funcional ALTO, segundo a Tabela 7. O valor de PFNA obtido transformando o grau de complexidade, de acordo com a Tabela 10, é seis (6).

A Tabela 33 mostra o total de SE identificadas na ferramenta juntamente com todos os EDD e AR à elas associados.

Tabela 32. Descrição de todos os SE, EDD e AR identificados na aplicação SMART

SE identificadas	**EDD associados**	**AR associados**
Geração de *hash* de um disco	Selecionar o disco Selecionar a ferramenta	Disco Arquivo com o valor *hash*
Geração de *hash* de uma imagem	Selecionar a imagem Selecionar a ferramenta	Imagem Arquivo com o valor *hash*
Comparar *hash* de uma imagem com o hash de um disco	Selecionar a imagem Selecionar a ferramenta Selecionar o disco	Imagem Disco
Geração de *hash* de dados selecionados	Selecionar os dados Selecionar a ferramenta	Dados Arquivo com o valor *hash*
Total	**8**	**8**

O grau de complexidade funcional obtido convertendo todos os EDD e AR é ALTO, segundo a Tabela 8. O valor resultante da transformação do grau de complexidade é sete (7), segundo a Tabela 9.

A Tabela 34 apresenta todas as CE reconhecidas na aplicação e os EDD e AR relacionados.

Tabela 33. Descrição de todos os CE, EDD e AR identificados na aplicação SMART

CE identificadas	EDD relacionados	AR relacionados
Visualização de informações sobre a imagem	Selecionar a imagem	Imagem
Montagem da imagem como somente leitura	Selecionar a imagem	Imagem
Visualização em formato *raw*	Selecionar a imagem	Imagem
Visualização de detalhes de um dado	Selecionar a imagem	Imagem Dado
Busca por expressão regular	Selecionar a imagem Expressão	Imagem
Busca por termo em *unicode*	Selecionar a imagem Termo	Imagem
Visualização de imagens em *slides*	Selecionar a imagem	Imagem Arquivos gráficos dentro da imagem
Busca pré-definida	Selecionar a imagem Navegadores Cartão de crédito Documentos E-mail Artefatos FTP Arquivos gráficos Internet Arquivos de configuração Linux Mensageiros Registro *Sniffing* Telefone Vídeo Artefatos *Web*	Imagem
Total	**24**	**10**

De acordo com a Tabela 8, o grau de complexidade funcional atingido é ALTO, resultando no valor numérico seis (6), segundo a Tabela 10.

Com os graus de complexidade funcional de todos os ALI, EE, SE, CE e todos os EDD, ER e AR à eles relacionados transformados em valores de PFNA, é possível passar para a última fase da análise do *software* SMART Linux, a determinação do VFA.

7.1.2.3 Determinação do Valor do Fator de Ajuste (VFA)

Segundo a métrica FPA, nesta última etapa foram analisadas as quatorze CGS previstas, sendo que para cada item analisado foram atribuído pontos, de acordo com a Tabela de atribuição de graus de influência que cada item possui.

A Tabela 35 apresenta os valores obtidos em cada item das CGS, a descrição do valor atribuído e o valor total de NIT.

Tabela 34. Atribuição de valores de acordo com as CGS previstas na FPA

Item das CGS	**Valor obtido**	**Descrição**
A	0	O processamento da aplicação será puramente *batch* ou será executado em um PC isolado
B	1	A aplicação preparará dados para o usuário final processar em outra CPU da instalação. Por exemplo, planilhas eletrônicas ou gerenciadores de banco de dados de PC
C	3	O tempo de resposta será crítico durante todo o horário de utilização. Não será necessário nenhum procedimento especial para utilização de CPU. Os requisitos de prazo de processamento com outros sistemas são limitados
D	2	Algumas considerações sobre tempo e segurança são necessárias
E	0	Nenhum período de pico de transações é esperado
F	0	Todas as transações serão processadas em modo *batch*
G	4	Apresenta os itens: - Facilidades para navegação (teclas de função, geração dinâmica de menus); - Existência de menus; - Ajuda e documentação *on-line*; - Movimento de *scroll* (vertical e horizontal); - Seleção de dados da tela por meio de movimentação do cursor; - Uso intensivo de vídeo reverso, brilho, sublinhado, cores e outros recursos de vídeo; - Possibilidade do uso de *mouse*; - Janelas *pop-ups*;
H	0	Nenhuma atualização
I	3	Apresenta os itens: - Controle sensível (processamento especial de auditoria) e/ou processamento específico de segurança da aplicação; - Processamento lógico extensivo; - Processamento complexo para manipular múltiplas possibilidades de entrada/saída (múltiplos meios e independência de equipamentos)
J	0	Não apresenta código reutilizável
L	1	Nenhuma consideração especial foi feita pelo usuário, mas um procedimento especial é requerido para a implantação
M	5	A aplicação será projetada para não precisar de intervenção do operador no seu funcionamento normal. Apenas a inicialização e parada do sistema ficam a cargo do operador. A recuperação automática de erros será uma característica da aplicação.
N	0	Nenhuma solicitação do usuário para considerar a necessidade de instalar a aplicação em mais de um local
O	3	Apresenta o item: - Será fornecido recurso de consulta e relatórios flexíveis capaz de manipular solicitações complexas de consulta, com combinações de um ou mais Arquivos Lógicos Internos (contar como três pontos)
Total	**22**	

Com o valor obtido na avaliação das CGS é possível determinar o VFA. Aplicando a fórmula de determinação da FPA juntamente com os valores convertidos dos graus de complexidade funcional de todos os ALI, EE, SE e CE é possível calcular o valor total de uma aplicação, de acordo com a métrica FPA, o qual será realizado mais adiante.

7.2 ANÁLISE DOS RECURSOS OFERECIDOS PELAS FERRAMENTAS

Neste item do capítulo atual serão mostrados os resultados obtidos em uma análise de recursos específicos das ferramentas de perícia forense. O intuito de realizar mais uma analise, além da prevista na métrica FPA, é fazer uma análise mais aprofundada e com foco exclusivo nas ferramentas de perícia forense, uma vez que a métrica FPA foi projetada para analisar todos os tipos de aplicações, independente da sua área de atuação.

Foi realizado um levantamento dos recursos que as ferramentas de perícia forense analisadas disponibilizam e depois comparados os resultados entre si. A cada item foi atribuído um ponto e o total de pontos obtidos por cada uma das ferramentas será somado com o total de pontos resultantes da análise pela métrica FPA.

A Tabela abaixo descreve os recursos levantados e os resultados obtidos.

Tabela 35. Pontuação de acordo com os recursos dos *softwares*

Recursos analisados	***The Sleuth Kit* e *Autopsy***	**SMART Linux**
Execução independente de permissões de *root*	✔	
Software livre	✔	
Open source	✔	
Edição de preferências de visualização		✔
Ajuda em todas as etapas	✔	✔
Informações sobre todos os sistemas de arquivos		✔
Informações gerais sobre o caso	✔	
Execução independente de outra ferramenta		✔
Opção de instalação de novas extensões		✔
Possibilidade de adicionar mais de uma imagem ao caso	✔	✔
Rapidez na inserção da imagem no caso		✔
Busca por palavras chave	✔	✔
Buscas pré-definidas (documentos, imagens)	✔	✔
Possibilidade de montar a imagem para navegação entre os arquivos		✔
Busca por arquivos excluídos	✔	
Visualização de arquivos por diretório	✔	
Busca utilizando expressão regular	✔	✔
Visualização de informações sobre último acesso, última modificação e exclusão de arquivos	✔	✔
Geração de valor *hash* da qualquer arquivo	✔	✔
Detalhamento de informações da imagem	✔	
Visualização de dados de um nó específico	✔	
Visualização de dados de um fragmento específico	✔	
Opção de visualização do conteúdo da imagem em hexadecimal		✔
Opção de visualização do conteúdo em *Unicode*		✔
Opção de criar buscas pré-definidas		✔
Criação de arquivo de *time line*	✔	
Criação de anotações em qualquer etapa da análise	✔	✔
Opção de extração de dados em sua forma original para outro diretório		✔
Rapidez na análise dos arquivos	✔	
Total	**19**	**18**

7.3 RESULTADOS OBTIDOS

A métrica FPA prevê a análise de uma ferramenta em três etapas: a identificação de todos os ALI e AIE e os EDD e ER relacionados, a identificação de todas as EE, SE e CE e os EDD e AR relacionados e por último a avaliação de quatorze CGS de acordo com o seu

grau de influência na aplicação. Desta análise resultam um total de pontos de PFNA e o valor do fator de ajuste (VFA). Para finalizar a contagem de pontos, a métrica prevê duas fórmulas matemáticas que devem ser aplicadas aos resultados obtidos anteriormente.

A primeira fórmula é relacionada à determinação do valor do fator de ajuste (VFA) e a segunda é referente ao cálculo final de cada aplicação de acordo com a métrica de Pontos de Função (FPA).

7.3.1 *The Sleuth Kit* e *Autopsy*

Abaixo segue uma Tabela com a contagem dos pontos obtidos pela aplicação *The Sleuth Kit* e *Autopsy*.

Tabela 36. PFNA obtidos pela ferramenta *Sleuth Kit*

Itens Identificados	**PFNA obtidos**
ALI	10
EE	6
SE	5
CE	6
Total	**27**

Aplicando a fórmula de determinação do VFA ao NIT resultante da análise das CGS.

$$VFA = (NIT * 0,01) + 0,65$$

$$VFA = (30 * 0,01) + 0,65$$

$$VFA = 0,95$$

Com o valor total do VFA é possível aplicar a fórmula final prevista na métrica FPA.

$$PFA = PFNA * VFA$$

$$PFA = 27 * 0,95$$

$$\mathbf{PFA = 25,65}$$

Portanto, o valor final da aplicação *The Sleuth Kit* e *Autopsy* de acordo com a métrica FPA é 30,4.

E somando este valor com o resultado da análise dos recursos o valor final é 49,4.

7.3.2 SMART Linux

A Tabela seguinte traz os valores de PFNA obtidos pela aplicação SMART Linux.

Tabela 37. PFNA obtidos pela ferramenta SMART

Itens Identificados	PFNA obtidos
ALI	7
EE	6
SE	7
CE	6
Total	**26**

Aplicando a fórmula de determinação do VFA ao NIT resultante da análise das CGS.

$$VFA = (NIT * 0,01) + 0,65$$

$$VFA = (22 * 0,01) + 0,65$$

$$VFA = 0,87$$

Aplicando a fórmula final prevista na métrica FPA com os valores de PFNA e VFA.

$$PFA = PFNA * VFA$$

$$PFA = 26 * 0,91$$

$$\mathbf{PFA = 22,62}$$

Portanto, o valor final da aplicação SMART Linux de acordo com a métrica FPA é 22,62.

Somando este valor com o resultado da análise dos recursos o valor final é 40,62.

7.3.3 Realizando a comparação entre os resultados

Após realizar a análise das ferramentas *The Sleuth Kit* e SMART Linux baseando-se na métrica Pontos de Função (FPA), foram obtidos os valores 25,65 e 22,62, respectivamente. Com estes valores em mãos é possível observar que a ferramenta *The Sleuth Kit* atingiu uma maior pontuação segundo a métrica. Portanto, é a ferramenta que mais se aproxima do esperado de uma ferramenta de perícia forense, levando em consideração a rotina de profissionais da área.

A análise dos recursos exclusivos de ferramentas de perícia forense resultou em 19 pontos para a ferramenta *The Sleuth Kit* e 18 pontos para a ferramenta SMART Linux. É possível perceber que a diferença entre as pontuações é pequena, deixando claro que ambas as ferramentas possuem vários recursos, porém o *Sleuth Kit* se sobressaiu também nesta segunda análise.

A pontuação final, obtida por meio da soma do resultado da análise segundo a métrica FPA e o resultado da análise dos recursos, foi de 44,65 para o *Sleuth Kit* e 40,62 para o SMART Linux.

O objetivo deste estudo comparativo é analisar qual das duas ferramentas estudadas possui a maior quantidade de recursos e a qualidade destes recursos, medidos pela métrica FPA. Portanto, baseando-se nos resultados obtidos, o *Sleuth Kit* seria a melhor opção na escolha de um *software* de perícia forense.

7.4 TRABALHOS FUTUROS

O presente trabalho apresentou uma análise comparativa entre dois *softwares* de perícia forense voltados ao ambiente Linux, utilizando a métrica de software FPA. Porém a área de perícia forense é muito abrangente, assim como a área de métricas de *software*.

Portando, como sugestão de trabalhos futuros seria a análise de outras ferramentas de perícia forense voltados ao ambiente Windows, análise comparativa utilizando outras métricas de *software*.

Uma outra opção seria o desenvolvimento de métricas de comparação de *software* específicas para a área de perícia forense, pois atualmente as métricas existentes não levam com consideração a área da aplicação analisada, tanto que para a confecção deste estudo foi necessário realizar uma segunda análise, onde são levantados os recursos específicos de ferramentas de perícia forense, para que análise final levasse em consideração a área das aplicações analisadas.

CONCLUSÃO

O presente trabalho apresentou conceitos de perícia forense computacional, aplicando-os na análise das ferramentas *The Sleuth Kit,* versão 3.1.2 (atual), utilizando a sua interface gráfica *Autopsy*, e o SMART Linux, versão liberada no dia cinco de junho de 2010 (atual), ambas voltadas à área de perícia forense.

A análise foi realizada baseando-se na métrica de *software* Pontos de Função (FPA), com o objetivo de medir a qualidade do *software,* a fim de apresentar uma avaliação detalhada destas ferramentas. Foi realizada uma segunda análise levando em consideração a quantidade de recursos que as ferramentas disponibilizam. Por fim foram comparados os resultados de ambas as ferramentas, comparando a quantidade de recursos de cada uma e a qualidade dos mesmos, segundo a métrica FPA. Baseando-se no resultado desta comparação é possível observar que a ferramenta *The Sleuth Kit* é a que possui uma maior quantidade de recursos e detém uma maior qualidade dos mesmos, de acordo com a métrica citada.

A perícia forense é uma área que necessita evoluir à mesma velocidade que crescem os recursos computacionais. A cada dia surgem novas tecnologias, novos recursos, e também surgem novos tipos de crimes digitais, que utilizam estas mesmas tecnologias para a sua execução. Portanto é necessário que haja preparação dos profissionais para lidar com qualquer tipo de situação.

O profissional da área de perícia forense, durante a atuação em uma investigação, pode se deparar com inúmeras situações. É por este motivo que, além de estar sempre se atualizando, o profissional deve manter a sua disposição várias ferramentas. E é preciso que ele esteja familiarizado com os recursos disponíveis por elas, bem como o desempenho de cada uma e o seu funcionamento, para que seja possível atender qualquer incidente de segurança a que for chamado.

O presente estudo, além de realizar a análise comparativa entre os *softwares* de perícia forense, tem por objetivo principal trazer ao alcance dos profissionais da área, e à qualquer pessoa que se interesse, uma análise dos recursos disponíveis de cada uma das ferramentas citadas, bem como a qualidade dos mesmos e também uma demonstração do funcionamento de cada uma.

Para encerrar, este trabalho tem por um de seus objetivos contribuir para o crescimento da área de perícia forense, sem almejar esgotar o assunto. E, principalmente, contribuir para que sejam confeccionados novos trabalhos futuramente, com outros focos e objetivos, para que esta área evolua cada dia mais e para que seja possível o surgimento de profissionais capacitados o suficiente para realizar uma investigação transparente e objetiva, com o objetivo de descobrir a verdade.

REFERÊNCIAS

ACCESS DATA. **Access Data: A pioneer in digital investigations since 1987.** 2010. Disponível em <http://www.accessdata.com/> Acessado em 2 jun. 2010.

ASR DATA. **SMART Linux ASR Data**. 2010. Disponível em < http://www.asrdata2.com/ > Acessado em 2 jun. 2010.

ANDRADE, Thiago Felipe de. **Perícia Forense Computacional Baseada em Sistema Operacional Windows.** 2005. 71 f. Monografia (Graduação) - Curso de Sistemas de Informação, Centro Universitário de Jaraguá do Sul, Jaraguá do Sul, 2005.

ARGOLO , Frederico Henrique Böhm. **Análise Forense em sistemas GNU/Linux**. 2005. 114 f. Monografia (Graduação), Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005.

BERALDI, Lairce Castanhera; ESCRIVÃO FILHO, Edmundo. **Impacto da tecnologia de informação na gestão de pequenas empresas.** Ciência da Informação: Instituto Brasileiro de Informação em Ciência e Tecnologia - IBICT, Brasília, v. 29, n. 1, p.46-50, 17 jan. 2000. Semestralmente.

BERNARDO, Adauto de Souza. **Técnicas Computacionais no Auxílio à Perícia Forense na Análise de Evidências Coletadas em Servidores Gnu/Linux.** 2006. 109 f. Monografia (Bacharel) - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense - UNESC, Criciúma, 2006.

BRASIL. Decreto-lei n.º 2.848, de 7 de dezembro de 1940, **Código Penal Brasileiro.** Disponível em <http://www.planalto.gov.br/ccivil/decreto-lei/Del2848compilado.htm>. Acessado em 1 jun. 2010

BRASIL. Decreto-lei n.º 9296, de 24 de julho de 1996, **Constituição Federal.** Disponível em < http://www.planalto.gov.br/ccivil_03/Leis/L9296.htm>. Acessado em 1 jun. 2010

BRASIL. Decreto-lei n.º 3689, de 3 de outubro de 1941, **Código de Processo Penal.** Disponível em < http://www.planalto.gov.br/CCIVIL/Decreto-Lei/Del3689.htm>. Acessado em 1 jun. 2010

CARRIER, Brian. **The Sleuth Kit and Autopsy Browser.** 2001. Disponível em < http://www.sleuthkit.org/ > Acessado em 1 jun. 2010.

CASEY, Eoghan. **Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet**. 2d. Londres: Academic Press, 2004.

COSTA, Marcelo A. S. Lemos. **Computação Forense.** Campinas: Millenium, 2003.

DIMER, Ramiro Webber. **Perícia Forense Computacional Aplicada a Ambientes New Technologies File System(NTFS).** 2007. 94 f. Monografia (Bacharel) - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense - UNESC, Criciúma, 2007.

FARMER, Dan; VENEMA, Wietse. **Perícia Forense Computacional – Teoria e Prática Aplicada**. 2007. 190 f. Ed. Pearson Prentice Hall. São Paulo, 2007.

FARMER, Dan; VENEMA, Wietse. **The Coroner's Toolkit (TCT).** 1999. Disponível em < http://www.porcupine.org/forensics/tct.html >. Acessado em 1 jun. 2010.

FONSECA, Glauco Alves; SILVA, Crístian Alves. **Sistema de Detecção de Intrusos para Redes Locais**. 2007. 91f. Curso de Pós graduação em Segurança de Redes de Computadores, Faculdade Salesiana de Vitória, Vitória, 2007.

FRANZ, Matthew. **Trinux Linux Security Toolkit**. 1998. Disponível em < http://trinux.sourceforge.net/legacy/ > Acessado em 1 jun. 2010.

FREITAS, Andrey Rodrigues de. **Perícia Forense Aplicada à Informática – Ambiente Microsoft**. 2006. 216 f. Ed. Brasport. São Paulo, 2006.

GEUS, Paulo Lício de; REIS, Marcelo Abdalla dos. **Análise Forense de Intrusões em Sistemas Computacionais: Técnicas, Procedimentos e Ferramentas.** 2002. 80 f. Instituto de Computação Universidade Estadual de Campinas, Campinas, 2002.

GEUS, Paulo Lício de et al. **Forense Computacional: Aspectos Legais e Padronização.** Artigo científico do Instituto de Computação da UNICAMP. Campinas, 2001. Disponível em < http://www.las.ic.unicamp.br/paulo/papers/2001-WSeg-flavio.oliveira-marcelo.reis-forense.pdf > Acessado em 16 jun. 2010.

GUIDANCE SOFTWARE. **Guidance Software: The world leader in digital investigations.** 2010. Disponível em < http://www.guidancesoftware.com/ > Acessado em 2 jun. 2010.

GUIMARÃES, Bruno Salgado. **O Grupo de Respostas a Incidentes de Segurança do DCC-UFRJ (GRIS-UFRJ) e a Importância dos CSIRTs.** GRIS-UFRJ. 2005. Artigo científico - Conceito - Informativo Técnico do Núcleo de Informação Eletrônica da UFRJ. Rio de Janeiro. Segunda edição, agosto 2005. Disponível em <http://www.nce.ufrj.br/conceito/artigos/2005/02-1.htm>. Acessado em: 11 mar. 2010.

HOUAISS, Antônio. **Dicionário Houaiss da Língua Portuguesa.** Rio de Janeiro: Objetiva, 2009.

IFPUG. International Function Point Users Group. **Function Point Practices Manual: Release 4.2.** International Function Point Users Group, Ohio, 2004.

INTELLIGENT COMPUTER SOLUTIONS**. Intelligent Computer Solutions Forensics**. 2010. Disponível em < http://www.icsforensic.com/ > Acessado em 2 jun. 2010.

MARTINEZ, Vinícius Aparecido. **Memórias *Cache*: Uma Breve Contextualização**. Artigo Científico – Universidade Federal de Mato Grosso do Sul (UFMS) – Faculdade de Computação, Campo Grande, 2009.

MELO, Sandro. **Computação Forense com Software Livre**. Rio de Janeiro: Alta Books, 2009.

NEUKAMP, Paulo Alberto; BOTELHO, Aderbal. **FDTK-UbuntuBr -- Forense Digital ToolKit.** 2010. Disponível em < http://www.fdtk.com.br > Acessado em 2 jun. 2010.

OLIVEIRA, David Sena. **Computação Quântica, Algoritmos e Simulações.** 2006. 68 f. Monografia – Universidade Estadual do Ceará – Graduação em Ciências da Computação, Fortaleza, 2006.

OLIVEIRA, Sabrina Vitório. **Perícia Forense Em Sistemas Gnu/Linux.** 2007. 79 f. Monografia - Faculdade Salesiana de Vitória - Pós-Graduação em Segurança de Redes de Computadores, Vitória, 2007.

PARKER, Donn B.. **Crime by Computer**. New York : Prentice Hall & IBD, 1990.

PASCOE, Luke. **MD5summer: Windows MD5 sum generator**. 2010. Disponível em < http://www.md5summer.org/ > Acessado em 2 jun.2010.

PRESSMAN, R. **Engenharia de Software: um enfoque prático.** São Paulo: McGraw_Hill, 2002.

SALUSKY, William. **F.I.R.E**..2002. Disponível em < http://fire.dmzs.com/> Acessado em 1 jun. 2010.

SAWAYA, Márcia Regina. **Dicionário de informática e internet - Inglês/Português**. 3. ed. São Paulo: Nobel, 1999.

SILVA, Rita de Cássia Lopes da. **Direito Penal e Sistema Informático.** São Paulo: Revista dos Tribunais, 2003. 4 v.

SLOUDIS, Ed; ZELTER, Lenny. **Malware: Fighting Malicious Code**. Nova Jersey: Prentice Hall PTR, 2003.

SOUZA, Clarice Muhlethaler de. **Biblioteca – uma Trajetória**. In: III Congresso Internacional de Biblioteconomia, 2005, Rio de Janeiro. Disponível em: <http://geocities.yahoo.com.br/csouza952/IIICIB.pdf.>. Acesso em: 05 out. 2009.

SOUSA, Augusto Gonçalves de. **Análise de pontos de função estendida: Métrica de software baseada na Abordagem das dimensões tecnológica e Ambiental/contextual.** 2006. 168 f. Dissertação (Mestrado) - Mestrado Interdisciplinar em Modelagem Computacional, Centro de Pós-graduação e Pesquisa Visconde de Cairu - Fundação Visconde de Cairu, Salvador, 2006. Disponível em <http://www.fattocs.com.br/artigos/Dissertacao _XFPA_MIMC_AugustoSousa.pdf>. Acessado em 25 mai. 2010.

STEPHENSON, P. **Investigating Computer-related Crime**. CRC Press. Boca Raton, 2000.

TREVENZOLI, Ana Cristina. **Perícia forense computacional – ataques, identificação da autoria, leis e medidas preventivas.** 2006. 89 f. Monografia (Especialização) - Curso de Segurança de Redes e Sistemas, Faculdade Senac de Sorocaba, Sorocaba, 2006.

APÊNDICE A – ARTIGO CIENTÍFICO

Análise e comparação de *softwares* para perícia forense em ambientes Linux baseado na métrica *Function Point Analysis*

Paula Porfírio Teixeira1

1Curso de Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma, SC – Brazil

xpaula@gmail.com

Resumo. *A perícia forense computacional é uma área que necessita evoluir na mesma velocidade que surgem novos recursos tecnológicos, pois é nesta mesma velocidade que surgem novos tipos de crimes digitais, uma vez que os criminosos digitais se utilizam destes mesmos recursos para a sua atuação ilícita. Portanto, visando contribuir com o seu crescimento, o presente trabalho apresentará uma análise comparativa entre duas ferramentas voltadas a esta área. A análise foi realizada em duas etapas: a primeira análise foi baseada na métrica de software Pontos de Função, que tem por objetivo medir a qualidade do software de uma aplicação, traduzindo em valores numéricos. A segunda análise foi realizada fazendo um levantamento dos recursos que cada ferramenta disponibiliza, atribuindo pontos a cada recurso identificado. Ao final destas duas análises foi realizada a comparação entre os resultados, sendo possível identificar qual das duas ferramentas possui uma maior quantidade de recursos e a maior qualidade destes recursos, baseando-se na métrica FPA.*

Palavras chave: *Segurança da informação, perícia forense, métricas de software.*

Abstract: *The forensic computing is an area that needs to evolve at the same speed that new technological resources appear, therefore it is in this same speed that appears new types of digital crimes, because criminals are using these same digital resources for their illegal actions. Therefore, aiming to contribute to its growth, this paper presents a comparative analysis between two tools devoted to this area. The analysis was performed in two stages: the first analysis was based on software metrics Function Points Analysis, which aims to measure the quality of a software application, translating into numeric values. The second analysis was conducted by taking a survey of resources that each tool provides, assigning points to each resource identified. At the end of these two analyses a comparison between the results was conducted, making it possible to identify which of these two tools possess a superior amount of resources and the best quality of these resources, being based on the metrics FPA.*

Keywords: *Information security, forensic analysis, software metrics.*

1. Introdução

Com a crescente popularização da Internet e dos meios de acessibilidade à mesma, criou-se uma nova modalidade de crimes, os crimes digitais. São crimes em que, na maioria das vezes, as provas estão apenas em formato digital.

Crime digital pode ser classificado como todo e qualquer ato que cause dano à terceiros, utilizando como principal meio de atuação o computador. Cabem nesta classificação invasão de contas bancárias e desvio de dinheiro, roubo de informações confidenciais, publicação de material de caráter ofensivo, como pedofilia, racismo entre outros (SILVA, 2003).

Baseado no Princípio da Troca de Locard , tudo que entra no local do crime leva consigo algo e deixa algo para trás. No mundo digital é possível aplicar esse conceito com algumas alterações: independente de onde o intruso esteve, ele deixa rastros (OLIVEIRA, 2007). Dentro da perícia forense, estes “rastros” são chamados de evidências digitais. Em alguns casos a evidência pode ser extremamente difícil de ser rastreada, mas ela existe de fato (STEPHENSON, 2000).

Para conseguir examinar estas evidências, é necessária uma investigação especial, utilizando softwares específicos, com o propósito de resgatar informações de determinados dispositivos e registrá-las para comprovar ter havido um crime ou não. Para tanto há a necessidade de prévios conhecimentos na área de análise forense computacional.

Existem disponíveis vários *softwares* voltados para a área de perícia forense, sendo que cada um possui suas particularidades, como por exemplo, possuir uma interface gráfica, realizar análise das propriedades dos arquivos, examinar os arquivos do disco rígido à procura de indícios de acesso não autorizado, procurar arquivos ocultos, listar arquivos e atributos de segurança, identificar o tipo de um arquivo por meio de seu conteúdo entre outras características (GEUS; REIS, 2002).

Em virtude aos fatos mencionados, este trabalho propõe um estudo comparativo de dois softwares de análise forense computacional voltados ao ambiente Linux. Serão realizados testes e comparações dos resultados, baseados na métrica Pontos de Função, a fim de analisar os recursos que cada ferramenta disponibiliza e avaliar a qualidade destes recursos, com o intuito de auxiliar na escolha pelo *software* que atenda melhor as necessidades do profissional de perícia forense computacional.

2. Conceitos de Perícia Forense

Para o completo entendimento do presente artigo, se faz necessária a compreensão de alguns conceitos básicos sobre perícia forense computacional. Tais conceitos serão apresentados nos próximos tópicos.

2.1 Perícia Forense Computacional

A perícia forense computacional consiste, basicamente, em um conjunto de procedimentos que, baseando-se na legislação vigente, tem por objetivo coletar evidências de equipamentos de armazenamento de dados, de forma que possam ser apresentados em juízo como provas coerentes e significativas (GEUS, 2001).

Compreende quatro etapas básicas: aquisição, identificação, avaliação e apresentação de evidências, formando o Ciclo de Perícia Forense, quaisquer que sejam as formas de armazenamento das mídias analisadas, com o objetivo de coletar provas que permitam a formulação de conclusões referentes ao caso investigado (GEUS; REIS, 2002; MELO, 2009).

A Figura 1 demonstra como as etapas do ciclo de Perícia Forense, descrito acima, se entremeiam.

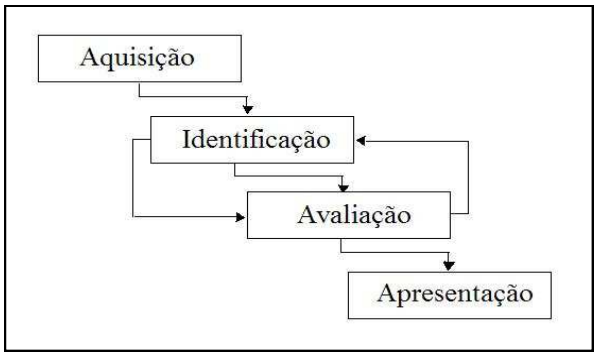

Figura 1. Ciclo de Perícia Forense
Fonte: MELO, S. (2009, p. 15)

2.1.1 Aquisição

Consiste basicamente em reunir todas as evidências possíveis para posterior análise. Como primeira fase do ciclo de perícia forense, o sucesso de uma investigação depende fundamentalmente da qualidade do material coletado.

A fase de aquisição das evidências não se restringe apenas ao computador. Toda mídia que possa ter entrado em contato com o sistema comprometido pode conter evidências. Portanto, devem ser investigadas também (COSTA, 2003).

2.1.2 Identificação

Consiste na etapa de organização dos artefatos encontrados na fase de aquisição das evidências. Nesta fase, o perito deve ser metódico, utilizando técnicas e procedimentos para a identificação e enumeração dos artefatos encontrados, pois, assim como a coleta de evidências, é uma fase crucial para a elaboração do laudo pericial, resultado final da perícia forense (MELO, 2009).

2.1.3 Avaliação

É nesta etapa que o perito forense analisa os artefatos e enumera os eventos colhidos anteriormente em uma linha de tempo. É a etapa que representa a perícia forense computacional em si, ou seja, a fase onde são analisadas as evidências colhidas. O perito irá elaborar uma linha do tempo (*time line*) onde poderão ser compreendidos como e porque cada evento ocorreu, o que o ocasionou e o resultado do mesmo (COSTA, 2003).

Todo e qualquer procedimento que o perito realizar é necessário que seja documentada. O sucesso de uma análise forense computacional depende basicamente da qualidade e integridade das evidências coletadas e da documentação de todo o processo de perícia forense. O não cumprimento deste requisito pode invalidar toda a perícia, sendo para fins judiciais ou não (ARGOLO, 2005).

2.1.4 Apresentação

Última etapa da análise forense e consiste na confecção e apresentação do laudo pericial. Ao perito cabe juntar e organizar todos os dados obtidos nas fases anteriores e relatar de forma clara e concisa. Pode-se dizer que o relatório final reconstrói o *modus operanti* do suposto atacante.

Apesar do laudo pericial não indicar diretamente a origem do ataque, as informações que nele contém podem ser utilizadas para este propósito. O relatório pode sugerir a origem e a categoria do incidente (COSTA, 2003).

3. Métricas de *Software*

Métricas de *software* podem ser definidas como uma medição quantitativa simples, com o intuito de expressar em números a funcionalidade, a qualidade ou o tamanho de uma aplicação. Com esta informação em mãos é possível estimar custo, esforço e prazo total de um projeto de aplicação (PRESSMAN, 2002).

3.1 Métrica *Function Point Analysis* (FPA)

Em 1979, o pesquisador da IBM Allan Albrecht propôs a métrica de análise de pontos de função, em inglês *function point analysis* (FPA), que consiste em atribuir pontos a fim de classificar uma aplicação. É largamente utilizada à nível mundial como principal indicador de formação de preços, bem como para estimar prazos e preços em licitações de projetos de *software* (SOUSA, 2006).

A aplicação desta métrica é feita em três etapas: contagem de funções de dados, contagem de funções de transação e determinação do valor do fator de ajuste. O resultado das duas primeiras etapas são valores de pontos de função não ajustados (PFNA). A terceira etapa é onde é determinado o valor do fator de ajuste (VFA). Tendo estes valores em mãos é possível aplicar a fórmula que define o valor final da aplicação.

3.1.1 Contagem de Funções de Dados

Funções de dados representam grupos lógicos que são referenciados pela aplicação por meio de requisitos funcionais do usuário. A métrica FPA prevê dois tipos de funções de dados:

a) Arquivo Lógico Interno (ALI): são grupos lógicos de dados que a aplicação mantém e que são reconhecidos pelos usuários;

b) Arquivo de Interface Externa (AIE): são também grupos lógicos de dados referenciados pela aplicação, porém mantidos por outra aplicação, identificados pelo usuário (IFPUG, 2004).

Tanto os ALI como os AIE possuem campos, chamados de Elemento de Dados (EDD). Os EDD também podem possuir subgrupos, denominados de Elemento de Registro (ER) (SOUSA, 2006).

Com os ALI e AIE identificados, EDD e ER associados, é possível determinar o grau de complexidade funcional da aplicação. O grau pode ser Alto, Médio ou Baixo, dependendo da quantidade de EDD e ER associados.

Obtendo o grau de complexidade funcional, é necessário realizar a conversão em valores de PFNA. Dependendo do grau obtido, o valor de PFNA será equivalente. No caso de grau Baixo, o valor é de cinco, Médio é sete e Alto é dez.

3.1.2 Contagem de Funções de Transação

Representam, em pontos, as funcionalidades de processamento de dados fornecidas pela aplicação aos usuários (SOUSA, 2006). Os tipos de funções de transação previstos na FPA são:

a) Entrada Externa (EE): refere-se ao processamento de dados que são utilizados pela aplicação para atualizar ALI ou alterar o comportamento do sistema;

b) Saída Externa (SE): apresenta o resultado do processamento de um cálculo matemático ou expressões para produção de dados derivados;

c) Consulta Externa (CE): visa apresentar informações aos usuários, sem envolver cálculos matemáticos ou expressões para produção de dados.

A quantidade de Arquivos Referenciados (AR) e os Elementos de Dados (EDD) associados aos arquivos, são utilizados para obtenção da complexidade funcional das EE, SE e CE identificadas na aplicação. E após obter o grau de complexidade é possível obter o valor de PFNA de cada um deles. Os resultados devem ser somados entre si juntamente com os resultados obtidos na contagem de funções de dados, totalizando os PFNA.

3.1.3 Determinação do Valor do Fator de Ajuste (VFA)

O VFA é determinado avaliando quatorze Características Gerais do Sistema (CGS) previstas na FPA. A cada item das CGS é obtido um valor, que corresponde ao Nível de Influência daquele item em relação a aplicação. Somando estes valores entre si será obtido o Nível de Influência Total (NIT) (IFPUG, 2004). O passo seguinte para a determinação do VFA é aplicar o valor de NIT obtido na fórmula abaixo.

$$VFA = (NIT * 0,01) + 0,65$$

Tendo o VFA em mãos é possível definir o valor final da aplicação.

$$PFA = PFNA * VFA$$

4. Análise comparativa das ferramentas

Para a confecção deste trabalho foram analisadas duas ferramentas de perícia forense. A primeira ferramenta analisada foi *The Sleuth Kit*, juntamente com a sua interface gráfica, o *Autopsy*. A segunda ferramenta foi SMART Linux. Ambas as ferramentas foram eleitas dentre as ferramentas disponíveis por serem voltadas ao ambiente Linux, possuírem interface gráfica e terem a opção de ser instalada em uma máquina como uma aplicação qualquer.

Os resultados obtidos pelas ferramentas após a aplicação da contagem de funções de dados e funções de transação podem ser conferidos nas Tabelas 1 e 2.

Tabela 1. *The Sleuth Kit* e *Autopsy*

Itens	**Quantidade**	**Complexidade**	**PFNA**
ALI EDD ER	7 17 11	Médio	10
EE EDD ER	5 13 8	Alto	6
SE EDD ER	4 4 5	Médio	5
CE EDD ER	7 18 7	Alto	6
PFNA			**27**

Tabela 2. SMART Linux

Itens	**Quantidade**	**Complexidade**	**PFNA**
ALI	5		
EDD	9	Baixo	7
ER	4		
EE	6		
EDD	8	Alto	6
ER	12		
SE	4		
EDD	8	Alto	7
ER	8		
CE	8		
EDD	24	Alto	6
ER	10		
PFNA			**26**

O nível de influência total obtido na avaliação dos quatorze itens das CGS e a aplicação da fórmula para a determinação do VFA pode ser conferido no cálculo abaixo.

The Sleuth Kit e *Autopsy*
NIT = 30
VFA = (30 * 0,01) + 0,65
VFA = 0,95

SMART Linux
NIT = 22
VFA = (22 * 0,01) + 0,65
VFA = 0,87

A aplicação da fórmula para a definição do valor final de cada aplicação pode ser conferida no cálculo a seguir.

The Sleuth Kit
PFA = PFNA * VFA
PFA = 27 * 0,95
PFA = 25,65

SMART Linux
PFA = PFNA * VFA
PFA = 26 * 0,87
PFA = 22,62

Além da análise realizada utilizando a métrica FPA, foi realizado um levantamento dos recursos que as ferramentas de perícia forense analisadas disponibilizam e depois comparados os resultados entre si. A cada item foi atribuído um ponto e o total de pontos obtidos por cada uma das ferramentas será somado com o total de pontos resultantes da análise pela métrica FPA.

O intuito de realizar mais uma analise, além da prevista na métrica FPA, é fazer uma análise mais aprofundada e com foco exclusivo nas ferramentas de perícia forense, uma vez que a métrica FPA foi projetada para analisar todos os tipos de aplicações, independente da sua área de atuação.

Os recursos analisados bem como o resultado obtido por cada ferramenta podem ser visualizados na tabela abaixo.

Tabela 3. Recursos analisados nas ferramentas

Recursos analisados	***The Sleuth Kit* e *Autopsy***	**SMART Linux**
Execução independente de permissões de *root*	✔	
Software livre	✔	
Open source	✔	
Edição de preferências de visualização		✔
Ajuda em todas as etapas	✔	✔
Informações sobre todos os sistemas de arquivos		✔
Informações gerais sobre o caso	✔	
Execução independente de outra ferramenta		✔
Opção de instalação de novas extensões		✔
Possibilidade de adicionar mais de uma imagem ao caso	✔	✔
Rapidez na inserção da imagem no caso		✔
Busca por palavras chave	✔	✔
Buscas pré-definidas (documentos, imagens)	✔	✔
Possibilidade de montar a imagem para navegação entre os arquivos		✔
Busca por arquivos excluídos	✔	
Visualização de arquivos por diretório	✔	
Busca utilizando expressão regular	✔	✔
Visualização de informações sobre último acesso, última modificação e exclusão de arquivos	✔	✔
Geração de valor *hash* da qualquer arquivo	✔	✔
Detalhamento de informações da imagem	✔	
Visualização de dados de um nó específico	✔	
Visualização de dados de um fragmento específico	✔	
Opção de visualização do conteúdo da imagem em hexadecimal		✔
Opção de visualização do conteúdo em *Unicode*		✔
Opção de criar de buscas pré-definidas		✔
Criação de arquivo de *time line*	✔	
Criação de anotações em qualquer etapa da análise	✔	✔
Opção de extração de dados em sua forma original para outro diretório		✔
Rapidez na análise dos arquivos	✔	
Total	**19**	**18**

4.1 Comparando os resultados obtidos

Após realizar a análise das ferramentas *The Sleuth Kit* e SMART Linux baseando-se na métrica Pontos de Função (FPA), foram obtidos os valores 25,65 e 22,62, respectivamente. Com estes valores em mãos é possível observar que a ferramenta *The Sleuth Kit* atingiu uma maior pontuação segundo a métrica. Portanto, é a ferramenta que mais se aproxima do esperado de uma ferramenta de perícia forense, levando em consideração a rotina de profissionais da área.

A análise dos recursos exclusivos de ferramentas de perícia forense resultou em 19 pontos para a ferramenta *The Sleuth Kit* e 18 pontos para a ferramenta SMART Linux. É possível perceber que a diferença entre as pontuações é pequena, deixando claro que ambas as ferramentas possuem vários recursos, porém o *Sleuth Kit* se sobressaiu também nesta segunda análise.

A pontuação final, obtida por meio da soma do resultado da análise segundo a métrica FPA e o resultado da análise dos recursos, foi de 44,65 para o *Sleuth Kit* e 40,62 para o SMART Linux.

O objetivo deste estudo comparativo é analisar qual das duas ferramentas estudadas possui a maior quantidade de recursos e a qualidade destes recursos, medidos pela métrica FPA. Portanto, baseando-se nos resultados obtidos, o Sleuth Kit seria a melhor opção na escolha de um software de perícia forense.

5. Conclusão

O presente trabalho apresentou conceitos de perícia forense computacional, aplicando-os na análise das ferramentas *The Sleuth Kit*, versão 3.1.2 (atual), utilizando a sua interface gráfica *Autopsy*, e o SMART Linux, versão liberada no dia cinco de junho de 2010 (atual), ambas voltadas à área de perícia forense.

A análise foi realizada baseando-se na métrica de *software* Pontos de Função (FPA), com o objetivo de medir a qualidade do *software*, a fim de apresentar uma avaliação detalhada destas ferramentas. Foi realizada uma segunda análise levando em consideração a quantidade de recursos que as ferramentas disponibilizam. Por fim foram comparados os resultados de ambas as ferramentas, comparando a quantidade de recursos de cada uma e a qualidade dos mesmos, segundo a métrica FPA. Baseando-se no resultado desta comparação é possível observar que a ferramenta *The Sleuth Kit* é a que possui uma maior quantidade de recursos e detém uma maior qualidade dos mesmos, de acordo com a métrica citada.

A perícia forense é uma área que necessita evoluir à mesma velocidade que crescem os recursos computacionais. A cada dia surgem novas tecnologias, novos recursos, e também surgem novos tipos de crimes digitais, que utilizam estas mesmas tecnologias para a sua execução. Portanto é necessário que haja preparação dos profissionais para lidar com qualquer tipo de situação.

O profissional da área de perícia forense, durante a atuação em uma investigação, pode se deparar com inúmeras situações. É por este motivo que, além de estar sempre se atualizando, o profissional deve manter a sua disposição várias ferramentas. E é preciso que ele esteja familiarizado com os recursos disponíveis por elas, bem como o desempenho de cada uma e o seu funcionamento, para que seja possível atender qualquer incidente de segurança a que for chamado.

O presente estudo, além de realizar a análise comparativa entre os *softwares* de perícia forense, tem por objetivo principal trazer ao alcance dos profissionais da área, e

à qualquer pessoa que se interesse, uma análise dos recursos disponíveis de cada uma das ferramentas citadas, bem como a qualidade dos mesmos e também uma demonstração do funcionamento de cada uma.

Para encerrar, este trabalho tem por um de seus objetivos contribuir para o crescimento da área de perícia forense, sem almejar esgotar o assunto. E, principalmente, contribuir para que sejam confeccionados novos trabalhos futuramente, com outros focos e objetivos, para que esta área evolua cada dia mais e para que seja possível o surgimento de profissionais capacitados o suficiente para realizar uma investigação transparente e objetiva, com o objetivo de descobrir a verdade.

6. Referências

ARGOLO , Frederico Henrique Böhm. Análise Forense em sistemas GNU/Linux. 2005. 114 f. Monografia (Graduação), Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005.

Costa, Marcelo A. S. Lemos. Computação Forense. Campinas: Millenium, 2003.

IFPUG. International Function Point Users Group. Function Point Practices Manual: Release 4.2. International Function Point Users Group, Ohio, 2004.

Geus, Paulo Lício de; Reis, Marcelo Abdalla dos. Análise Forense de Intrusões em Sistemas Computacionais: Técnicas, Procedimentos e Ferramentas. 2002. 80 f. Instituto de Computação Universidade Estadual de Campinas, Campinas, 2002.

Geus, Paulo Lício de et al. Forense Computacional: Aspectos Legais e Padronização. Artigo científico do Instituto de Computação da UNICAMP. Campinas, 2001. Disponível em < http://www.las.ic.unicamp.br/paulo/papers/2001-WSeg-flavio.oliveira-marcelo.reis-forense.pdf > Acessado em 16 jun. 2010.

Melo, Sandro. Computação Forense com Software Livre. Rio de Janeiro: Alta Books, 2009.

Oliveira, Sabrina Vitório. Perícia Forense Em Sistemas Gnu/Linux. 2007. 79 f. Monografia - Faculdade Salesiana de Vitória - Pós-Graduação em Segurança de Redes de Computadores, Vitória, 2007.

Pressman, R. Engenharia de Software: um enfoque prático. São Paulo: McGraw_Hill, 2002.

Silva, Rita de Cássia Lopes da. Direito Penal e Sistema Informático. São Paulo: Revista dos Tribunais, 2003. 4 v.

Sousa, Augusto Gonçalves de. Análise de pontos de função estendida: Métrica de software baseada na Abordagem das dimensões tecnológica e Ambiental/contextual. 2006. 168 f. Dissertação (Mestrado) - Mestrado Interdisciplinar em Modelagem Computacional, Centro de Pós-graduação e Pesquisa Visconde de Cairu - Fundação Visconde de Cairu, Salvador, 2006. Disponível em <http://www.fattocs.com.br/artigos/Dissertacao_XFPA_MIMC_AugustoSousa.pdf>. Acessado em 25 mai. 2010.

Stephenson, P. Investigating Computer-related Crime. CRC Press. Boca Raton, 2000.

ANEXO A – A LEGISLAÇÃO E OS CRIMES DIGITAIS

A seguir são apresentadas algumas leis do Código Penal Brasileiro e da Constituição Federal que, interpretadas pelo poder judiciário, podem punir os criminosos digitais adequadamente:

a) O artigo 153 do decreto-lei nº. 2848 do Código Penal Brasileiro diz:

> Divulgar alguém, sem justa causa, conteúdo de documento particular, ou de correspondência confidencial, de que é destinatário ou detentor, e cuja divulgação possa produzir dano a outrem (BRASIL, 1940).

Códigos maliciosos, na grande maioria das vezes, têm por objetivo coletar informações confidenciais do usuário. Alguns desses códigos, além de roubar informações as enviam para uma lista de *e-mails* ou para um remetente específico, normalmente do próprio invasor, caracterizando um crime digital e que poderia ser punido pelo artigo citado;

b) O artigo 155 do decreto-lei nº. 2848 do Código Penal Brasileiro diz: “Subtrair, para si ou para outrem, coisa alheia móvel" (BRASIL, 1940).

Enquadra-se no crime citado acima o roubo de informações, basicamente. Dados computacionais também podem ser classificados como bem móvel, pois podem ser transferidos ou enviados à outras pessoas. Apoderar-se de informações de autenticação de algum sistema bancário, por exemplo, pode ser julgado sob o artigo citado acima, pois o infrator estará subtraindo para si as informações do usuário;

c) O artigo 163 do decreto-lei nº. 2848 do Código Penal Brasileiro diz: “Destruir, inutilizar ou deteriorar coisa alheia" (BRASIL, 1940).

Neste artigo se enquadram os códigos maliciosos que visam unicamente causar dano ao sistema invadido. Ataques à servidores *web*, por exemplo, que fazem com que o serviço provido pelo mesmo fique inacessível por tempo indeterminado, podem ser julgados sob o artigo 163 do Código Penal;

d) O artigo 171 do decreto-lei nº. 2848 do Código Penal Brasileiro diz: “Obter, para si ou para outrem, vantagem ilícita, em prejuízo alheio, induzindo ou mantendo alguém em erro, mediante artifício, ardil, ou qualquer outro meio fraudulento" (BRASIL, 1940).

Os crimes que visam o beneficiamento financeiro do criminoso podem ser enquadrados nesse artigo. Roubo de informações de autenticação em sistemas bancários, por exemplo, podem sofrer a punição prevista nesse artigo. Geralmente essas informações são adquiridas por meio da criação de uma página de Internet visualmente idêntica à original, mas as informações inseridas pelo usuário, não tem o destino previsto por este. Elas são enviadas para o criminoso que a criou, com o intuito de serem ilicitamente utilizadas;

e) O artigo 307 do decreto-lei nº. 2848 do Código Penal Brasileiro diz: “Atribuir-se ou atribuir a terceiro falsa identidade para obter vantagem, em proveito próprio ou alheio, ou para causar dano a outrem" (BRASIL, 1940).

Após apropriar-se das informações alheias, o criminoso as utiliza para fins ilícitos, como transferência bancária, envio de e-mails em nome da vitima, acesso às informações sigilosas mediante autenticação do usuário, entre outras situações. Como o criminoso está utilizando de artifícios para se passar por uma outra pessoa, pode ser enquadrado neste artigo da lei;

f) O artigo 10 do decreto-lei nº. 9296 da Constituição Federal diz: "Constitui crime realizar interceptação de comunicações telefônicas, de informática ou telemática, ou quebrar segredo da Justiça, sem autorização judicial ou com objetivos não autorizados em lei” (BRASIL, 1996).

Crimes digitais como coleta de informações do usuário por meio de ferramentas que coletam dados que trafegam na rede do usuário, e não dentro do computador invadido, especificamente, podem ser enquadrados neste artigo e sofrer as devidas punições previstas.