SACI – Sistema de Auditoria Computacional Inteligente

Rosana de Lima BEZERRA (1); Leandro Cavalcanti de ALMEIDA (2); Diego Ernesto Rosa PESSOA (3); Elionildo da Silva MENEZES (4)
(1) Centro Federal de Educação Tecnológica da Paraíba (CEFET/PB), Avenida 1º de Maio, Nº 720, CEP 58015-430, João Pessoa – PB, (83) 32083062, e-mail: rosana.bezerra11@gmail.com
(2) CEFET/PB, e-mail: leandro@comunicacaodigital.org
(3) CEFET/PB, e-mail: diegopessoa12@gmail.com
(4) CEFET/PB, e-mail: elionildo@cefetpb.edu.br

RESUMO

A ciência forense vem sendo utilizada há muito tempo para resolver questões legais relacionadas a diversas áreas do conhecimento. Recentemente, ela passou a ser utilizada no campo da Tecnologia da Informação, recebendo a denominação de forense computacional. A forense computacional é um processo de investigação que envolve tarefas como identificação, coleta, preservação, análise de evidências e reconstrução de eventos, devendo ser executado por um especialista denominado perito computacional. Nesse cenário, este artigo pretende apresentar o projeto SACI (Sistema de Auditoria Computacional Inteligente), que consiste numa plataforma baseada em *software livre* para integrar ferramentas forenses existentes com as desenvolvidas pelo projeto, a fim de criar um kit que possa ser facilmente instalado, utilizado e mantido por peritos computacionais. O trabalho é uma pesquisa experimental baseada na seguinte metodologia: (i) Estudo dos conceitos relacionados e programas existentes na área; (ii) Especificação de requisitos, modelagem do SACI e implementação do seu primeiro módulo; (iii) Utilização do SACI para coleta de evidências e validação dos resultados obtidos. Como resultado e contribuição deste trabalho, destaca-se a criação de uma plataforma modular a ser usada por peritos computacionais com diversos níveis de conhecimento.

Palavras-chave: Forense computacional, Auditoria, SACI, *Software livre*

1. INTRODUÇÃO

Ao longo de sua história, o homem vem tentando criar mecanismos para a identificação da autoria de ações criminosas, o que levou ao surgimento da chamada ciência forense, a qual é utilizada nos dias atuais sobretudo para a identificação da autoria de crimes que vão desde a intrusão em ambientes de acesso restrito, passando pelo roubo de objetos e adulteração de documentos até casos de atentados contra a vida humana. Segundo Thorton (1997), a ciência forense é aquela exercida em favor da lei para uma justa resolução de um conflito.

As redes de computadores, notadamente a partir da década de 1990 com o advento da *World Wide Web* como aplicação na Internet, tornaram-se instrumento indispensável à comunicação entre pessoas espalhadas pelo mundo inteiro, influenciando assim a proliferação do uso de computadores pessoais. Por essas redes trafegam dados de simples mensagens eletrônicas trocadas entre amigos, projetos de pesquisa altamente confidenciais e transações comerciais envolvendo grandes organizações e verdadeiras fortunas. Por isso, as redes vêm sofrendo um número cada vez maior de incidentes de segurança, o que torna indivíduos e empresas vulneráveis a roubos de dados e a ataques que podem comprometer dados, corromper arquivos e provocar a queda de sistemas (DEITEL, 2005 e MANDIA, 2003). Apenas para ser ter uma idéia, estatísticas do CERT.br[1] (Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil) indicam um crescimento significativo do número de incidentes de segurança, que passou de 3107 em 1999 para 160080 em 2007, depois de atingir um pico de 197892 incidentes reportados no ano de 2006.

A área de segurança da informação tradicionalmente deu ênfase à definição de estratégias e implementação de mecanismos que ajudassem o analista de segurança a proteger dados, aplicações e componentes de hardware computacionais dos ataques realizados por indivíduos maliciosos. Com isso, uma vez que os ataques tivessem ocorrido, não havia uma metodologia que pudesse ser usada para descobrir provas que indicassem a origem e autoria de tais ataques. Para suprir essa lacuna, recentemente, surgiram estudos que integraram as áreas de ciência forense e segurança da informação, dando origem à ciência forense computacional ou simplesmente forense computacional e criando um novo perfil de analista de segurança, o perito computacional.

Este artigo apresenta a proposta de uma plataforma, integrando *softwares* existentes com outros desenvolvidos ao longo do projeto, que poderá ser utilizada como um *kit* de ferramentas de apoio ao trabalho de peritos computacionais com diferentes níveis de conhecimento técnico na área. A seção 2 apresenta os conceitos e trabalhos relacionados à forense computacional bem como *softwares* utilizados. A seção 3 apresenta a modelagem do SACI (Sistema de Auditoria Computacional Inteligente) e a implementação do seu primeiro módulo. A seção 4 descreve o estudo de caso realizado para validar a funcionalidade do primeiro módulo do SACI, bem como mostra a análise dos resultados obtidos. Finalmente, a seção 5 apresenta alguns comentários acerca do estágio atual do desenvolvimento do SACI e aponta para os passos futuros do projeto.

2. CONCEITOS E TRABALHOS RELACIONADOS

De acordo com Noblett (2000), a forense computacional pode ser definida como sendo a ciência de adquirir, preservar, recuperar e exibir dados que foram eletronicamente processados e armazenados digitalmente.

A forense computacional difere da perícia forense tradicional, uma vez que as ferramentas e técnicas utilizadas estão disponíveis para qualquer indivíduo que tente realizar uma investigação forense computacional, enquanto que a forense tradicional requer formação aprofundada em determinada área do conhecimento, como balística e medicina legal. Adicionalmente, os exames de um sistema – *hardware* ou *software* – comprometido podem ser realizados em qualquer ambiente físico e não apenas em ambientes sob condições controladas. Por fim, ao invés de produzir conclusões que precisam de interpretação por especialistas, a forense computacional produz informações e dados diretos que podem ter um papel importante na confirmação da prática de crimes digitais (SCHWEITZER, 2003).

Como ocorre com qualquer trabalho pericial, todas as etapas da forense computacional devem ser realizadas de forma metódica e obedecendo a procedimentos previamente definidos, validados e aceitos pela comunidade científica internacional, de forma que todos os resultados obtidos sejam passíveis de reprodução e tenham um alto nível de confiabilidade para possam ser usados como prova em um processo judicial. Por

[1] As estatísticas do CERT.br estão disponíveis em http://www.cert.br/stats/incidentes/.

isso, várias entidades nacionais e internacionais foram criadas com o objetivo de produzir padrões e procedimentos a serem utilizados na forense computacional. O Quadro 1 apresenta algumas dessas entidades (RODRIGUES, 2004):

Quadro 1 – Entidades de padronização na área de forense computacional

Entidades Internacionais	
IOCE (International Organization on Computer Evidence)	http://www.ioce.org/
SWGDE (Scientific Working Group on Digital Evidence)	http://www.swgde.org/
HTCIA (High Technology Crime Investigation Association)	http://www.htcia.org/
IACIS (International Association of Computer Investigatibe Specialists)	http://www.cops.org/
Entidades Nacionais	
SEPINF (Serviço de Perícias de Informática), da Polícia Federal	http://www.dpf.gov.br/
CERT.br (Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil - antigo NBSO -NIC BR Security Office)	http://www.cert.br/
CAIS (Centro de Atendimento a Incidentes de Segurança)	http://www.rnp.br/cais/

2.1.Etapas da forense computacional

A forense computacional engloba cinco etapas gerais, necessárias a qualquer investigação de crime digital, que são (PEREIRA, 2007; FREITAS, 2006; SOLOMON, 2005): Coleta das Informações, Reconhecimento das evidências, Análise das evidências, Reconstrução dos eventos e Apresentação dos Resultados. A Figura 1 apresenta um possível ciclo de vida do processo de forense computacional.

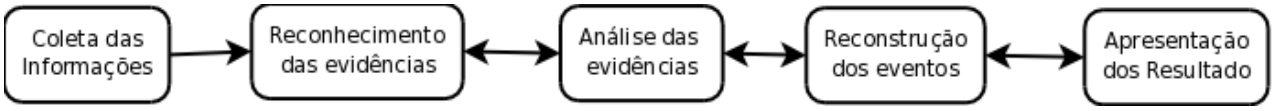

Figura 1 – Ciclo de vida do processo de forense computacional

- Coleta das informações – Esta etapa inclui duas "sub-etapas". A primeira é caracterizada como pré-coleta, que tem como principal objetivo a preparação do ambiente, ocorrendo a esterilização de todas as mídias que serão utilizadas durante o processo de coleta, além da definição dos responsáveis que irão trabalhar no caso. A segunda etapa caracteriza a coleta propriamente dita, em que ocorre a cópia segura dos dados, além da criação de arquivos de *hash*[2] para comprovar a integridade durante o processo de coleta. Um ponto interessante é coletar informações de acordo com a ordem de volatividade (FARMER, 2007 e BREZINSKI, 2002), isto é, inicialmente devem ser coletados dados mais voláteis como aqueles dos registradores, passando pela memória principal e disco rígido até chegar a mídias de armazenamento não regraváveis.
- Reconhecimento das evidências – Esta é uma das etapas mais trabalhosas para o perito, devido ao fato de se filtrar e identificar apenas as informações relevantes dente todas as possíveis, sendo indispensável a utilização de ferramentas específicas para o reconhecimento de evidências, como o *stegdetect* para identificar o uso da esteganografia, *The Coroners Tool Kit*, entre outros.
- Análise das evidências – Envolve ações que visam identificar informações sobre um crime digital, tais como a estratégia utilizada, o momento da ocorrência e a autoria. Esta fase compreende a análise de

[2] Um *hash* corresponde a um valor único obtido a partir da aplicação de uma função matemática sobre o dado – arquivo ou diretório e até mesmo um disco inteiro – alvo do *hash*. Com isso, a integridade do dado pode ser comprovada, haja vista que qualquer alteração neste dado produzirá um *hash* diferente do original.

sistemas operacionais, arquivos de dados, portas de comunicação abertas, arquivos de *logs*[3], usuários conectados e ações realizadas inclusive criação, modificação e remoção de arquivos; correlacionando fatos que tenham a ver com aquele evento. Durante essa etapa, o perito deve ter em mente que as evidências coletadas precisam ser autênticas, precisas e completas, além de terem sido coletadas de forma legal e poderem ser usadas em um tribunal (BREZINSKI, 2002). Adicionalmente, é importante que o perito forense documente todos os resultados obtidos durante esta etapa e trabalhe com cópias das informações coletadas, a fim de não alterar as provas originais obtidas e permitir que sua análise possa ser repetida tantas vezes quanto forem necessárias.

- Reconstrução dos eventos – Esta é a fase em que vão ser reconstruídos os eventos de acordo com as evidências analisadas, procurando-se encontrar quais foram os passos do infrator para conseguir realizar a conduta culposa.
- Apresentação dos resultados – Nessa etapa, deve ser gerado um relatório técnico sobre a investigação (laudo pericial), contendo o resultado da execução de todas as etapas anteriores, sobretudo da análise de evidências. Nesse relatório devem estar descritos todos os fatos levantados, procedimentos utilizados, análises realizadas e resultados obtidos. Com a conclusão do relatório, o perito forense encerra seu trabalho, cabendo à justiça o julgamento sobre o caso investigado.

Durante o ciclo de vida da forense computacional aplicada a um crime digital, é importante que o perito documente a cadeia de custódia de cada evidência (FREITAS, 2006), isto é, todo o trâmite da evidência durante o processo de investigação, incluindo informações sobre quando a evidência foi manipulada, por quem e onde, as mídias utilizadas para armazenar a evidência, dentre outros.

É importante salientar que, com exceção da fase de Coleta das informações, todas as outras etapas, conforme pode ser visto na Figura 1, são iterativas, ou seja, se o perito verificar que surgiu um novo indício, ele pode voltar à(s) etapa(s) anterior(es) para coletar as evidências que porventura tenham passado despercebidas.

A seguir serão apresentadas algumas das ferramentas comumente utilizadas em forense computacional.

2.2.Programas utilizados em forense computacional

Atualmente existem várias programas (ferramentas de *software*) que podem ser usados em forense computacional. Alguns deles são descritos a seguir (SOLOMON, 2005 e REIS, 2003).

- ChkRootKit – É um programa utilizado para a detecção de *rootkits*[4], cuja implementação foi feita utilizando-se as linguagens Shell Script e C. Ao ser executado, o ChkRootKit verifica se o sistema operacional possui os comandos utilizados para a detecção de *rootkits*, se há arquivos ou diretórios com permissões não usuais, se há nomes de arquivos e diretórios associados a *rootkits*, se a placa de rede encontra-se em modo promíscuo, se ocorreu alguma remoção no arquivo *lastlog*, etc. O poder do ChkRootKit reside na base de dados que ele utiliza, a qual deve estar sempre atualizada.
- EasyRecovery – Consiste em um programa utilizado para recuperar dados de arquivos removidos, partições corrompidas, discos que tenham sido formatados. Com ele, os dados recuperados podem ser enviados para mídias removíveis ou unidades de rede.
- Encase[5] – É um programa desenvolvido para uso no sistema operacional Windows, cuja concepção está baseada no conceito de caso, ou seja, o perito forense deverá criar inicialmente um arquivo para cada caso investigado. Desse momento em diante, toda e qualquer ação realizada pelo perito usando o programa deve ser associada ao caso correspondente.Dentre as funcionalidades do Encase destacam-se: monitoramento e investigação em tempo real e sem interrupção de serviços; captura de informações voláteis, incluindo conteúdo da memória RAM, de programas em execução, arquivos e portas abertos; gerenciamento da cadeia de custódia, dentre outras.

---

[3] Os arquivos de *log* registram informações sobre eventos que ocorrem com o sistema operacional ou serviço. Tais eventos incluem data de acesso, usuário que realizou o acesso, dentre outras informações.

[4] O termo *rootkit* comumente refere-se a um conjunto de programas utilizados pelo atacante de um sistema operacional para ocultar sua presença.

[5] Site oficial do *software:* http://www.encase.com.

- The Coroner´s Toolkit[6] – Compreende um conjunto de programas de código aberto utilizados para realizar análise de ambientes baseados em sistema operacional Linux/UNIX. Ele possui quatro principais funcionalidades: coleta de informações, utilizando o programa *grave-robber*; análise de arquivos, utilizando os programas *ils* e *mactimes*; recuperação de arquivos removidos, utilizando os programas *unrm* e *lazarus*; e recuperação de chaves de criptografia, através do uso do programa *findkey*.
- The Sleuth Kit[7] – Compreende um conjunto de programas de código aberto e com interfaces baseadas em linha de comando, que pode ser utilizado para realizar análise de evidências em sistemas de arquivos dos sistemas operacionais Linux, UNIX e Windows.

3. O SACI

Não existem padrões ou normas em relação ao modo de trabalho na perícia forense computacional, o que existem são algumas fases ou processos que devem são realizados pela grande maioria de profissionais que trabalham na área, a fim de que os resultados tenham confiabilidade.

Como já foi dito, vários *softwares* foram criados para auxiliar o trabalho de um perito forense computacional, porém nenhum daqueles considerados *softwares* livres implementa todos os processos necessários a uma análise forense computacional adequada. Esta é então a principal motivação para o estudo e desenvolvimento do SACI, sigla para Sistema de Auditoria Computacional Inteligente.

O SACI tem como objetivo contemplar todas as etapas necessárias em um processo de análise forense computacional. Sendo assim, o processo de desenvolvimento do SACI segue estas etapas, que são denominadas de “módulos”, ou seja, o SACI é uma plataforma modular a ser usada por peritos computacionais com níveis de conhecimentos diversos.

Atualmente o SACI encontra-se no processo de desenvolvimento do seu primeiro módulo, utilizando inicialmente duas linguagens de programação: Shell Script e PHP (*Hypertext Preprocessor*). Todo o núcleo será feito em Shell Script, e a interface com o usuário será desenvolvida em PHP. O núcleo abrange toda a parte de manipulação de arquivos, formatação (física e lógica) e clonagem de discos, geração de *hashes*, ou seja, seria a camada que iria trabalhar diretamente com o sistema operacional . A interface e o *layout* (ver Figura 2) abrange toda a parte de cadastro de peritos e casos, *uploads* de fotos/videos, preenchimento da cadeia de custódia, acompanhamento do caso, ou seja, seria a camada de interação com o usuário.

Figura 2 - Interface gráfica do sistema - Tela de cadastro de Peritos

[6] Site do projeto: http://www.porcupine.org/forensics/tct.html.
[7] Site do projeto: http://www.sleuthkit.org/sleuthkit/index.php.

Para manipular o sistema, o usuário precisará inicialmente de um *login* e senha, que serão configurados no ato de instalação. Após se identificar, ele poderá selecionar um caso já existente, iniciando a coleta de dados ou poderá cadastrar novos casos e peritos, conforme ilustra a Figura 3.

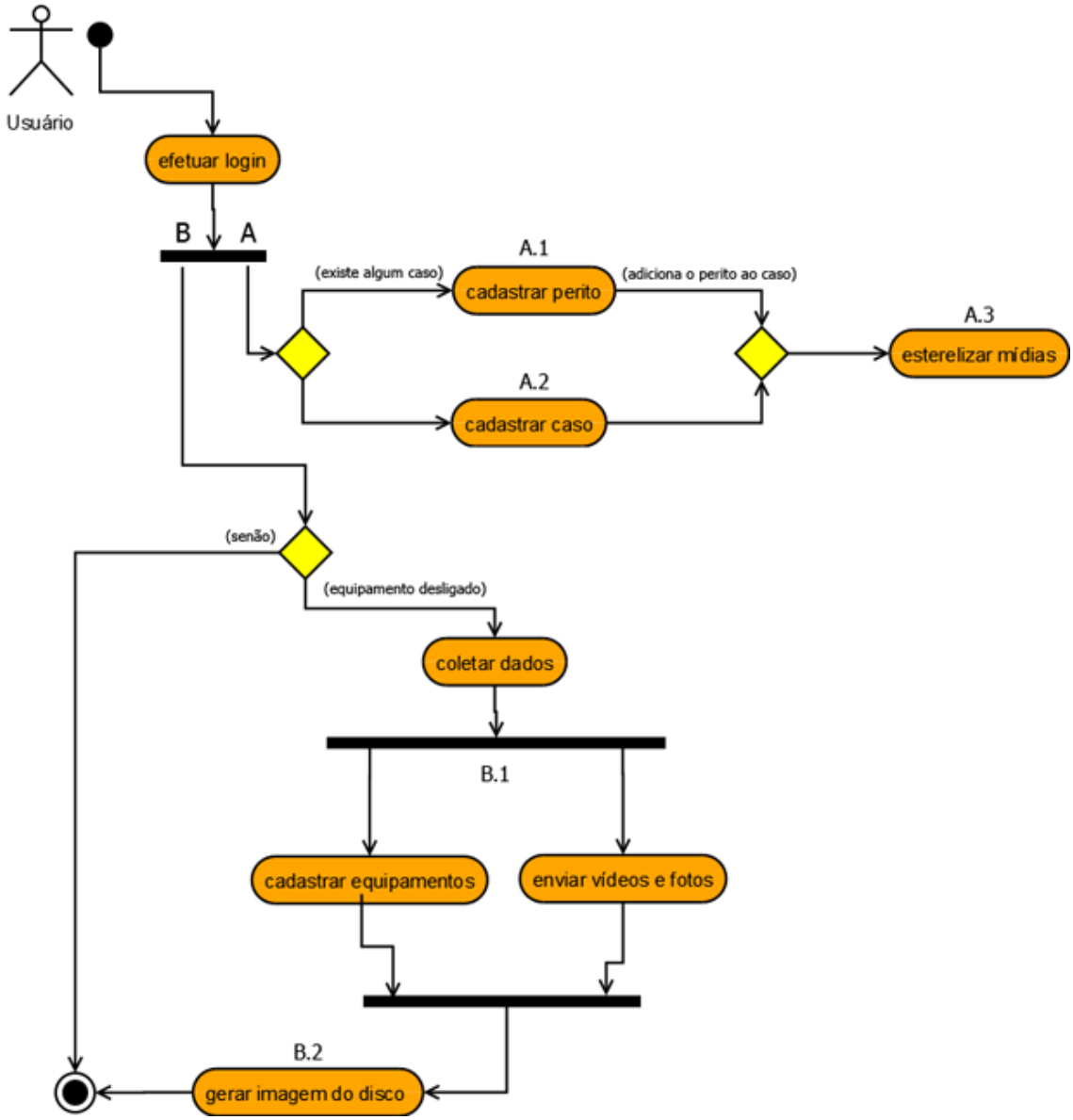

Figura 3 - Diagrama de atividades do módulo de coleta

Desse modo, as atividades realizadas pelo usuário no módulo de coleta da ferramenta são as seguintes:

A) Cadastro de casos ou peritos

A.1) Cadastro de Peritos
Pequeno cadastro dos peritos que poderão utilizar o sistema. Não é obrigatória a participação inicial do perito em algum caso.

A.2) Cadastro de casos e peritos responsáveis
Formulário onde o usuário informará os dados básicos daquele caso e poderá administrar os peritos responsáveis.

A.3) Esterilização de mídias
Antes de iniciar a coleta é necessário que se faça uma limpeza nos discos que serão utilizados para receber os dados, a fim de garantir a total integridade da cópia obtida. Nesta ação, a camada da interface em PHP irá acionar os comandos da camada em Shell Script.

B) Escolha de um caso e início do processo de coleta

B.1) Coleta de dados
Nesta parte serão enviados os dados referentes à coleta, como *upload* de vídeos, fotos e a descrição, gerando como resultado o relatório da coleta, também serão cadastrados os equipamentos que fizeram parte da investigação.

B.2) Geração de imagem do disco
Feito todo o processo de coleta, novamente o sistema irá se integrar com o Shell e executar o processo de geração de imagem do disco nas mídias que foram esterelizadas anteriormente.

A Figura 4 descreve o diagrama de classes do módulo de coleta do SACI:

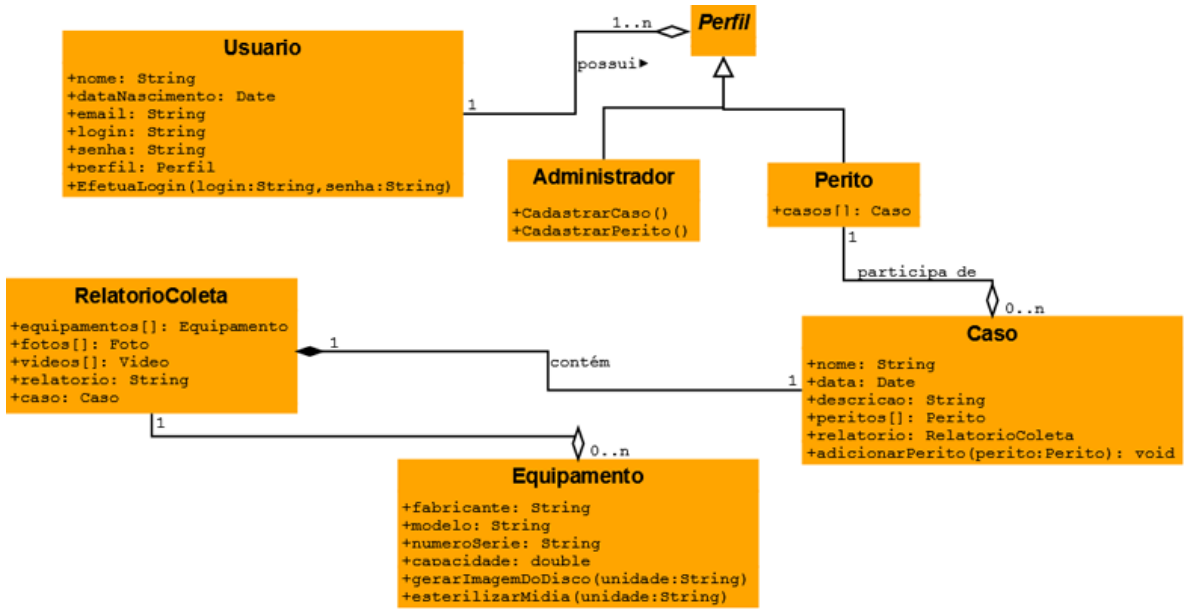

Figura 4 - Diagrama de classes do módulo de coleta

4. ESTUDO DE CASO

Nesta parte do artigo, iremos demonstrar um estudo de caso utilizando o primeiro módulo (coleta) do SACI. O principal objetivo do estudo foi realizar a coleta das informações de um sistema *Red Hat* invadido, disponibilizado no site http://www.honeynet.org (*Scan of the Month 29*). O cenário utilizado está exemplificado de acordo com a Figura 5:

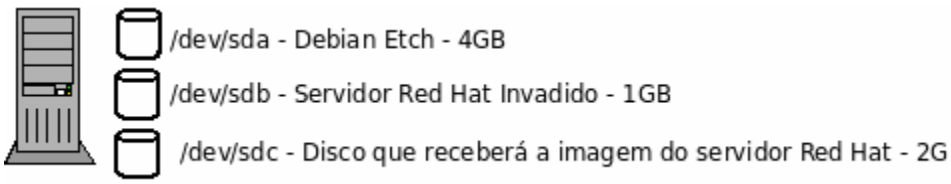

Figura 5 – Descrição dos discos no cenário estudado

Basicamente, temos um servidor com o Sistema Operacional *Debian Etch* instalado em um disco (/dev/sda) de 4GB, um disco (/dev/sdb) de 1GB que contém um servidor *Red Hat* invadido e um outro disco (/dev/sdc) de 2GB que foi utilizado para armazenar a imagem do servidor *Red Hat*.

Neste caso, o processo de coleta seria simplesmente gerar uma imagem do disco que contém o servidor *Red Hat* invadido para uma análise posterior, além da criação de *hashes* criptográficos para comprovar a integridade durante esta coleta.

Existem diversas maneiras para que se obtenha sucesso durante o processo de coleta, porém existe uma carência de softwares livres que façam esta tarefa de forma transparente e completa. É neste contexto que utilizou-se o primeiro módulo (coleta) da plataforma SACI - Sistema de Auditoria Computacional Inteligente.

Este primeiro módulo consiste de alguns *scripts* feitos em Shell Script e PHP, que foram instalados no Servidor Debian para que o processo de coleta fosse realizado com sucesso. Antes de começar o processo de coleta propriamente dito, existe algumas medidas que devem ser tomadas para preparar o ambiente e permitir que o processo ocorra de forma correta. Estas medidas também são conhecidas como uma fase de pré-coleta.

A primeira ação do SACI foi cadastrar o usuário (perito que iria trabalhar no caso) e o caso em si (descrição do caso do Servidor *Red Hat* invadido). Após o período de cadastro, o usuário (logado) efetuou o processo de esterilização da mídia (/dev/sdb) que iria receber a imagem do servidor invadido, de forma que outras informações existentes na mesma não interferissem no processo de análise, sendo assim, a partir deste ponto, começa efetivamente o processo de coleta.

Utilizando o SACI, o usuário efetuou o processo de coleta das informações do disco (/dev/sdb) que continha o Servidor *Red Hat* invadido para o disco (/dev/sdc), observando-se ainda que foram gerados quatro arquivos de *hash*, para comprovar a integridade no processo de coleta. Dois destes arquivos de *hash* (MD5SUM e SHA1SUM) foram gerados antes do procedimento de coleta, e os outros dois arquivos de *hash* (MD5SUM e SHA1SUM) foram gerados após o processo de coleta e foram comparados, comprovando assim a total transparência e integridade durante o processo de coleta.

5. CONSIDERAÇÕES FINAIS

A Perícia Forense Computacional vêm se mostrando imprescindível para a Tecnologia da Informação, na medida em que permite que os administradores possam descobrir as vulnerabilidades dos seus sistemas e encontrar culpados, sejam esses tanto em nível judicial como em nível empresarial.

Várias ferramentas estão sendo utilizadas para conseguir realizar uma perícia forense computacional, porém, nenhuma delas é completa o suficiente para que possibilite ao usuário executar todas as etapas de uma perícia forense de modo confiável. Por isso, a proposta do SACI é ser uma plataforma modular, que contemple todas as fases da forense computacional, tendo uma interface amigável e permitindo que usuários, com diversos níveis de conhecimento, possam utilizá-la.

Até o momento, foi desenvolvido o primeiro módulo do SACI, o que corresponde à implemantação das fases de pré-coleta e coleta, que foram validadas com o estudo de caso apresentado.

Como perspectivas de continuidade do trabalho, pretende-se implementar os demais módulos da ferramenta e disponibilizá-la para a comunidade do *software* livre, a fim de permitir que seus membros possam utilizar, avaliar e contribuir para o desenvolvimento e manutenção da mesma.

REFERÊNCIAS

BREZINSKI, D., KILLALEA, T. **Guidelines for Evidence Collection and Archiving**. RFC (Request For Comments) 3227. Feb. 2002.

DEITEL, H. M., DEITEL, P. J., CHOFFNES, D. R. **Sistemas Operacionais**. 3. ed.: Pearson Prentice Hall, 2005.

FARMER, D., VENEMA, W. **Perícia Forense Computacional: Teoria e Prática Aplicada**. Pearson Prentice Hall, 2007.

FREITAS, A. R. **Perícia Forense Aplicada à Informática: Ambiente Microsoft**. Brasport, 2006.

MANDIA, K., PROSISE, C., PEPE, M. **Incident Response & Computer Forensics**. 2nd. Edition: McGraw-Hill/Osborne, 2003.

NOBLETT, M. G., POLLITT, M. M.; PRESLEY, L. A. **Recovering and Examining Computer Forensic Evidence**. Vol. 2 N. 4, Forense Science Communications; Federal Bureau of Investigation, Oct. 2000.

PEREIRA, E. D. V.; FAGUNDES, L. L.; NEUKAMP, P.; LUDWIG, G.; KONRATH, M. **Forense Computacional: fundamentos, tecnologias e desafios atuais**. *In:* SIMPÓSIO BRASILEIRO EM SEGURANÇA DA INFORMAÇÃO E DE SISTEMAS COMPUTACIONAIS, VII, 2007, Rio de Janeiro. **Minicurso ...** Rio de Janeiro: UFRJ.

REIS, M. A. **Forense computacional e sua aplicação em segurança imunológica**. Dissertação de Mestrado. Instituto de Computação da UNICAMP – São Paulo, 2003.

RODRIGUES, W. P. **Análise Pericial em Sistema Operacional MS-Windows**. Monografia de Especialização. Universidade Estadual de Londrina – Paraná, 2004.

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

SOLOMON, M., BARRET, D., BROOM, N. **Computer Forensics JumpStart**. Sybex, 2005.

THORTON, J. **The general assumptions and rationale of forensic identification; Modern Scientific Evidence: The Law and Science of Expert Testimony**. Vol. 2. West Publishing Co., 1997.