Capítulo

2

Crimes Cibernéticos e Computação Forense

Wilson Leite da Silva Filho

Abstract

Cibercrimes have caused a deep impact in the society. They are among the three types of crimes that have caused the major financial loss in the world, staying only behind of traffic of drugs and falsification. From the investigation and production of proves needs against this type of crime, rises the computer forensics area. This area is responsible for collect, preserve, process and present its results to the legal authorities. Its a computer science area in continues development, that demands an ongoing research of theirs experts, once each new digital technology is also an opportunity to commit crimes.

Resumo

Crimes cibernéticos têm causado um impacto cada vez maior na sociedade. Está entre os três tipos de crimes que causam maior prejuízo financeiro no mundo, ficando atrás apenas do tráfico de drogas e a falsificação. Da necessidade de investigação e produção de provas para o combate a este ilícito, surge a área de computação forense. Esta área provê técnicas para a coleta, preservação, processamento e apresentação de evidências para autoridades legais. É uma área da Ciência da Computação em constante desenvolvimento e que demanda pesquisa e atualização contínua dos especialistas, na qual cada nova tecnologia emergente traz consigo o potencial de também ser explorada para fins ilícitos.

2.1. Introdução

O objetivo deste material didático é apresentar aos alunos a área de Computação Forense, com foco na área criminal, englobando alguns dos principais crimes cibernéticos e as técnicas e ferramentas usadas na área. O texto é embasado na literatura

e nos conhecimentos empíricos das perícias criminais oficiais de informática realizadas no Estado de Santa Catarina pelo Instituto de Criminalística do IGP/SC.

O texto está estruturado em oito seções. A seção 1 é esta introdução. Na seção 2, Crimes Cibernéticos, é apresentada uma definição de crimes que envolvem a área de informática e são mostrados exemplos de crimes cibernéticos de repercussão e cifras correspondentes ao prejuízo causado por este tipo de ato. O objetivo principal é sensibilizar o leitor acerca da importância e dimensão que esse tipo de delito possui nos dias atuais.

Na seção 3, Princípios da Computação Forense, são abordadas as principais etapas do processo de perícia em artefatos digitais, bem como, são enfatizadas as precauções necessárias para a correta aquisição e manipulação da evidência digital, mantendo-a válida durante todo o processo legal.

Aspectos Jurídicos em Computação Forense é o assunto da seção 4. São apresentadas, de forma sucinta, algumas das leis que estão diretamente relacionadas à área de computação forense, com foco na área criminal.

Na seção 5, Laboratório de Computação Forense: Preservação e Análise da Prova Digital, são apresentados os dispositivos de *hardware* e *software* usados em um laboratório de forense computacional. São mostrados *softwares* comerciais e *softwares* livres para diversas atividades dessa área.

Em Princípios da Recuperação de Evidências Digitais, tópico da seção 6, são abortadas descrições do funcionamento e estrutura de algumas tecnologias relacionadas à área. Tal conhecimento teórico é importante na atuação dos especialistas forenses. Assuntos como discos rígidos, discos de estado sólido (SSDs), sistemas de arquivos, e técnicas de *data carving* são detalhados. Também é feito o detalhamento técnico de tecnologias em que a evidência digital pode estar presente. São apresentados técnicas de perícias no Registro e em *logs* do Windows e perícias em dados voláteis.

A seção 7, Técnicas Antiforenses e Anti-Antiforenses discute os recursos que geralmente são usados para dificultar as perícias e como essas tecnologias podem ser contornadas. São apresentados tópicos como sanitização de discos, criptografia, quebra de senhas e esteganografia.

Finalmente, na seção 8, são abordadas as técnicas de perícias em dispositivos móveis, com ênfase aos que possuam sistema operacional Android.

2.2. Crimes Cibernéticos

Os equipamentos computacionais podem ser utilizados de duas formas para o cometimento de crimes: ferramenta de apoio à prática de delitos convencionais ou alvo/ peça imprescindível da ação criminosa.

Na primeira categoria, os crimes envolvidos são delitos que podem ser cometidos sem o uso de computadores, mas por estarmos cada vez mais envolvidos em um mundo digital, esses crimes tradicionais certamente deixarão vestígios digitais. Analisemos, como exemplo, o crime de corrupção passiva, tão em voga atualmente.

Para se corromper, o agente não precisa da ajuda de computadores. Mas, provavelmente suas atitudes ilícitas deixarão rastros digitais: e-mails com parceiros do crime, planilhas e outros documentos digitais que podem materializar o fato criminoso. Dessa forma, praticamente qualquer criminoso pode deixar rastros no mundo digital, tornando a computação forense uma área de muita importância na persecução penal.

A outra categoria são os crimes de informática propriamente ditos, nos quais os computadores são peças imprescindíveis para o cometimento do crime. Sem eles, tais crimes não existiriam. Ataques a *sites*, programas maliciosos para roubo de senhas, programas que sequestram os dados do usuário (*ransomware*), entre outros, são exemplos desse tipo de crime.

Muitos fatos criminosos de repercussão e correlacionados com a computação forense têm surgido na mídia. Abordaremos alguns, como o objetivo de, além de informar, sensibilizar o leitor em relação à dimensão que este tipo de delito tem tomado.

Preso em 2008 pela Polícia Federal, o traficante internacional de drogas Juan Carlos Ramírez Abadía teve seu computador periciado. Segundo matéria jornalística publicada pela Folha de São Paulo, o que causou estranheza à polícia foi ter encontrado centenas de fotos do desenho Hello Kitty, todas enviadas por *e-mail*. Em uma análise mais cuidadosa, descobriu-se que estas imagens carregavam mensagens escondidas pela técnica denominada esteganografia. Entre o conteúdo das mensagens havia ordens para movimentar cocaína entre países e para sumir com pessoas na Colômbia. Uma outra reportagem, do site de notícias G1, traz informações que o grupo extremista Al-Qaeda usou filmes pornográficos para esconder informações de ataques terroristas.

Outro caso de repercussão foi o embate entre o FBI e a Apple relacionado a um iPhone de um suposto terrorista. A polícia estadunidense requisitou que a Apple desenvolvesse uma versão especial do iOS que permitisse que o dispositivo fosse desbloqueado de forma segura. A empresa negou-se a desenvolver qualquer solução tecnológica que comprometesse a segurança de seus dispositivos. A saída encontrada pelo FBI foi pagar um grande quantia a um grupo especializado em segurança da informação que possuía em sua base privada de vulnerabilidades conhecidas uma falha de segurança que permitia desbloquear o iPhone.

Uma forte área de atuação dos cibercriminosos, principalmente no cenário brasileiro, são as fraudes bancárias pela internet. De acordo com estimativas da Febraban – Federação Brasileira de Bancos – 95% das perdas dos bancos do Brasil vem do cibercrime. Em números, o Brasil perde mais de US$ 8 bilhões por ano com fraudes digitais, um dos maiores crimes econômicos no País.

O crime cibernético já é o terceiro que mais causa prejuízo financeiro ao mundo depois do narcotráfico e da falsificação de marcas e de propriedade intelectual.

2.3. Princípios da Computação Forense

A computação forense consiste, basicamente, no uso de métodos científicos para preservação, coleta, validação, identificação, análise, interpretação, documentação e apresentação de evidência digital com validade probatória em juízo.

Segundo definição encontrada em Brooks (2014), computação forense é uma disciplina que combina elementos do direito e da computação com objetivo de coletar e analisar dados de sistemas computacionais, redes de computadores, comunicações sem fio e sistemas de armazenamento digitais de tal forma que esses dados sejam válidos na justiça.

Cuidados devem ser tomados para garantir a preservação e coleta dos dados digitais: isolar o local; evitar acessos remotos; utilizar funções de *hash* para garantir a integridade dos dados e a cadeia de custódia.

2.3.1. Preservação e coleta dos dados

Em alguns casos, o perito criminal é chamado para acompanhar uma operação policial em que haja a possibilidade de existir provas digitais. Para esses casos e para outros casos em que a perícia ou a coleta inicial dos dados aconteçam no local onde estão em funcionamento os sistemas computacionais, deve-se tomar precauções para que o local seja adequadamente isolado.

A prova digital pode ser bastante volátil. Se o local não for devidamente isolado, os dados de interesse podem ser corrompidos ou apagados. Para evitar o comprometimento das evidências, recomenda-se não permitir que os usuários dos locais acessem seus computadores, bem como, interromper as comunicações de rede externas para que comandos remotos para limpeza dos dados não possam ser executados.

Ao se deparar com máquinas que estejam desligadas, vide regra, não se deve ligá-las. O principal motivo dessa recomendação é a preservação dos dados. Ao se ligar as máquinas, o próprio processo de inicialização do sistema operacional fará alterações em alguns dados, e essas alterações podem ser detectadas examinando-se os metadados de carimbo de tempo dos arquivos. Além disso, acessar arquivos de interesse diretamente nas máquinas, também alterará, no mínimo, os dados de tempo dos arquivos. Desse modo, é preferível fazer imagem dos computadores e proceder as análises sobre as imagens.

Os equipamentos, se apreendidos, devem ser etiquetados, constando o nome da pessoa que usava aquele equipamento. Uma recomendação também é perguntar ao usuário a senha de acesso ao dispositivo. Equipamentos com senha podem demandar mais tempo para acesso aos dados ou mesmo inviabilizar a perícia. Dessa forma, não custa nada perguntar a senha. Se o usuário colaborar, anotar a senha para que possa ser usada, caso necessário.

Finalmente, para aqueles dispositivos que tenham conectividade com redes celulares, deve-se colocá-los em modo avião. Não sendo possível, deve-se retirar o chip SIM e desligá-lo. Esse processo é importante, pois os sistemas de dispositivos móveis permitem que os aparelhos sejam bloqueados e os dados apagados remotamente. Colocando-se o dispositivo em modo avião, elimina-se esse risco.

2.3.2. Integridade e cadeia de custódia

Garantir a integridade e prover meios de se assegurar a cadeia de custódia é uma das atividades do *expert* de computação forense.

Segundo Machado (2009), cadeia de custódia é procedimento preponderante e de suma importância para a garantia e transparência na apuração criminal quanto à prova material, sendo relato fiel de todas as ocorrências da evidência, vinculando os fatos e criando um lastro de autenticidade jurídica entre o tipo criminal, autor e vítima.

Na computação forense, o cálculo do *hash* das evidências digitais é um recurso fundamental para a garantia da integridade e da cadeia de custódia da prova. Pelo cálculo e documentação do *hash* da evidência original e da cópia forense, é possível garantir que a cópia é idêntica ao original e que em qualquer momento que se deseje analisar a cópia, basta recalcular o *hash* e verificar se aquela cópia está íntegra.

Toda essa garantia é possível devido as características matemáticas de uma função de *hash*, que é um algoritmo que mapeia dados de comprimento variável para dados de comprimento fixo, obedecendo determinadas propriedades. Essas propriedades asseguram que o resultado do *hash* será praticamente único para aquela coleção de dados original e que qualquer mudança nos dados originais gerará um código de *hash* totalmente diferente.

Do ponto de vista das características técnicas e propriedades necessárias, Stallings (2008) destaca que: uma função *hash* deverá poder ser aplicada sobre um bloco de dados de qualquer tamanho; sempre produzirá uma saída de tamanho fixo; deverá ser relativamente fácil de se calcular para qualquer bloco de dados, tornando as implementações em *hardware* e *software* práticas; deverá ser resistente à primeira inversão, ou seja, de posse da saída da função deverá ser computacionalmente inviável encontrar o bloco de dados de entrada; deverá possuir resistência fraca a colisões, ou seja, tendo-se o bloco de dados de entrada x, deve ser computacionalmente inviável encontrar um bloco de dados y que gere a mesma saída da função de *hash* e possuir resistência forte a colisões, ou seja, deverá ser computacionalmente inviável encontrar quaisquer pares de blocos x e y cujo resultado da função *hash* seja a mesma.

2.3.3. Análise e apresentação dos resultados

Uma vez obtida as evidências digitais, de forma íntegra e com cuidados para garantir a cadeia de custódia, chega a hora de analisar os dados. É a fase do exame pericial em si. Entre as principais atividades dessa fase, estão, buscar evidências apagadas, buscar determinada evidência em um universo imenso de dados, decodificar e interpretar dados, compreender eventos dos sistemas computacionais envolvidos, entre outras atividades.

Para finalizar todo o trabalho forense, há a redação do laudo pericial, o qual apresentará os resultados da perícia. O principal desavio nesta etapa final é escrever um documento de maneira que seja tecnicamente preciso e compreensível aos operadores do direto.

2.4. Aspectos Jurídicos em Computação Forense

Por ser uma ciência que visa reportar suas análises e resultados a alguma instância da justiça, é estreita sua relação com as leis. Algumas delas afetam diretamente o trabalho dos peritos, pesquisadores e profissionais da área e devem ser de conhecimento desse grupo.

Primeiramente, talvez como a lei fundamental que garante o exame pericial, temos no Código de Processo Penal - Do exame do corpo de delito e das perícias em geral- Art. 158 e159 que dizem:

> Art. 158. Quando a infração deixar vestígios, será **indispensável** o exame de corpo de delito, direto ou indireto, não podendo supri-lo a confissão do acusado.
>
> Art. 159. O exame de corpo de delito e outras perícias serão realizados por perito oficial, portador de diploma de curso superior.
>
> § 1o Na falta de perito oficial, o exame será realizado por 2 (duas) pessoas idôneas, portadoras de diploma de curso superior preferencialmente na área específica, dentre as que tiverem habilitação técnica relacionada com a natureza do exame.
>
> § 2o Os peritos não oficiais prestarão o compromisso de bem e fielmente desempenhar o encargo.
>
> § 3o Serão facultadas ao Ministério Público, ao assistente de acusação, ao ofendido, ao querelante e ao acusado a **formulação de quesitos e indicação de assistente técnico**.
>
> § 4o O assistente técnico atuará a partir de sua admissão pelo juiz e após a conclusão dos exames e elaboração do laudo pelos peritos oficiais, sendo as partes intimadas desta decisão.

Dessa forma, no âmbito criminal, é obrigatório o exame pericial em todo crime que deixar vestígio. Outro ponto importante, é a possibilidade do perito da defesa, denominado assistente técnico. Esse profissional pode fazer a sua própria análise pericial e apresentar as suas conclusões em relatório próprio para a apreciação do judiciário.

Outra lei importante, com relação próxima a um tipo de perícia em informática, é a que trata do crime de pedofilia. Tipificado no ECA (Estatuto da Criança e do Adolescente), nos artigos 240 e 241.

> Art. 240. **Produzir, reproduzir, dirigir, fotografar, filmar ou registrar**, por qualquer meio, cena de sexo explícito ou pornográfica, envolvendo criança ou adolescente: Pena – reclusão, de 4 (quatro) a 8 (oito) anos, e multa.
>
> Art. 241. **Vender ou expor à venda** fotografia, vídeo ou outro registro que contenha cena de sexo explícito ou pornográfica envolvendo criança ou adolescente: Pena – reclusão, de 4 (quatro) a 8 (oito) anos, e multa.

> Art. 241-A. **Oferecer, trocar, disponibilizar, transmitir, distribuir, publicar ou divulgar por qualquer meio, inclusive por meio de sistema de informática ou telemático**, fotografia, vídeo ou outro registro que contenha cena de sexo explícito ou pornográfica envolvendo criança ou adolescente: Pena – reclusão, de 3 (três) a 6 (seis) anos, e multa.
>
> Art. 241-B. **Adquirir, possuir ou armazenar**, por qualquer meio, fotografia, vídeo ou outra forma de registro que contenha cena de sexo explícito ou pornográfica envolvendo criança ou adolescente: Pena – reclusão, de 1 (um) a 4 (quatro) anos, e multa. § 1o A pena é diminuída de 1 (um) a 2/3 (dois terços) se de pequena quantidade o material a que se refere o caput deste artigo.

É importante ressaltar algumas das condutas que são crimes em relação à pedofilia e qual o papel do perito em computação forense nesses casos. Primeiramente, destaca-se que o simples fato de armazenar, ou seja, possuir fotos de pedofilia no computador ou *smartphone* já é crime. O papel do perito em relação a esse fato é encontrar tais imagens, que podem estar escondidas, apagadas ou criptografadas. Caso essas imagens sejam encontradas, o próximo passo natural é determinar se o proprietário do dispositivo estava compartilhando essas imagens com outros usuários, o que constitui um crime mais grave. Esse compartilhamento pode ocorrer principalmente por aplicativos de redes ponto a ponto (P2P). Cabe ao perito, verificar essa situação e documentar todo o cenário encontrado.

A lei nº 12.965, de 23 de abril de 2014, conhecida como Marco Civil da Internet estipula algumas regras, das quais as que mais interessam à computação forense são as que regulam o armazenamento dos registros de acesso (*logs*) dos usuários, como mostrado a seguir.

> Art. 1o Esta Lei estabelece **princípios, garantias, direitos e deveres para o uso da internet no Brasil** e determina as diretrizes para atuação da União, dos Estados, do Distrito Federal e dos Municípios em relação à matéria.
>
> …
>
> Subseção I
>
> Da Guarda de Registros de Conexão
>
> Art. 13. Na provisão de conexão à internet, cabe ao administrador de sistema autônomo respectivo o dever de **manter os registros de conexão, sob sigilo, em ambiente controlado e de segurança, pelo prazo de 1 (um) ano**, nos termos do regulamento.
>
> § 2o **A autoridade policial ou administrativa ou o Ministério Público poderá requerer cautelarmente que os registros de conexão sejam guardados por prazo superior ao previsto no caput**.
>
> § 5o Em qualquer hipótese, a disponibilização ao requerente dos registros de que trata este artigo deverá ser precedida de **autorização judicial**, conforme disposto na Seção IV deste Capítulo.
>
> Subseção II
>
> Da Guarda de Registros de Acesso a Aplicações de Internet na Provisão de Conexão

> Art. 14. Na provisão de conexão, onerosa ou gratuita, é **vedado guardar os registros de acesso a aplicações de internet**.
>
> Subseção III
>
> Da Guarda de Registros de Acesso a Aplicações de Internet na Provisão de Aplicações
>
> Art. 15. O **provedor de aplicações de internet** constituído na forma de pessoa jurídica e que exerça essa atividade de forma organizada, profissionalmente e com fins econômicos **deverá manter os respectivos registros de acesso a aplicações de internet, sob sigilo, em ambiente controlado e de segurança, pelo prazo de 6 (seis) meses**, nos termos do regulamento.

A lei nº 12.737, de 30 de novembro de 2012, conhecida como lei Carolina Diekmann, tipifica, ou seja, torna crime, várias condutas relacionadas a atividades de invasão de sistemas de computador, conforme segue.

> Art. 1o Esta Lei dispõe sobre a **tipificação criminal de delitos informáticos** e dá outras providências.
>
> ...
>
> "Invasão de dispositivo informático"
>
> Art. 154-A. **Invadir dispositivo informático alheio, conectado ou não à rede de computadores, mediante violação indevida de mecanismo de segurança e com o fim de obter, adulterar ou destruir dados ou informações sem autorização expressa ou tácita do titular do dispositivo ou instalar vulnerabilidades para obter vantagem ilícita**:
>
> Pena - detenção, de 3 (três) meses a 1 (um) ano, e multa.
>
> § 1o Na mesma pena incorre quem **produz, oferece, distribui, vende ou difunde dispositivo ou programa de computador** com o intuito de permitir a prática da conduta definida no caput.
>
> Art. 154-B. Nos crimes definidos no art. 154-A, **somente se procede mediante representação, salvo se o crime é cometido contra a administração pública** direta ou indireta de qualquer dos Poderes da União, Estados, Distrito Federal ou Municípios ou contra empresas concessionárias de serviços públicos."

Resumidamente, essa lei trata das invasões de sistemas e confecção e uso de software maliciosos (*malware*). É papel do perito, analisar os computadores em que ocorreram as invasões, determinar como elas ocorreram e se possível apontar na direção do responsável por tais crimes.

2.5. Lab de Computação Forense: Preservação e Análise da Prova Digital

O laboratório de computação forense deve possuir *hardware* e *software* especializado que proporcione as condições técnicas, de forma eficiente, para se obter e processar os

dados digitais, transformando-os em evidências. Essas tecnologias estão disponíveis em produtos comerciais, *softwares* desenvolvidos por peritos e *softwares* livres.

2.5.1. Duplicação de dados de forma forense

Uma das primeiras atividades a ser realizada no laboratório é a cópia dos dados dos equipamentos originais. É nessas cópias que as análises serão realizadas.

Para realizar uma cópia de forma forense, todos os bits do equipamento original devem ser copiados, inclusive de áreas não alocadas do sistema de arquivo. Além dessa necessidade, a evidência digital deve ser acessada de forma que haja proteção contra escrita na interface em que ela for conectada. Essa precaução é necessária para que, ao se conectar a mídia original, nenhum dado seja alterado. Conectar a mídia original sem proteção de escrita pode alterar dados ou metadados de arquivos e essas alterações podem ser questionadas pelas partes envolvidas. Outra atividade essencial ao fazer a cópia é calcular o *hash* dos dados originais e o da cópia. Esses valores devem coincidir, garantindo-se, com isso, a integridade e a cadeia de custódia das evidências digitais.

Existem equipamentos especializados em duplicação pericial. Esses equipamentos permitem que as cópias sejam feitas de forma bastante simplificada e asseguram as recomendações citadas. Algumas opções de equipamentos que podem existir em um laboratório de computação forense são o Solo IV, da empresa ICS e o Tableaut TD3 da empresa Guidence Software. Esses equipamentos possuem entradas protegidas contra escrita para conexão das evidências originais, diversos tipos de adaptadores para as interfaces mais comuns de mídias de armazenamento, entre elas, adaptadores para conexões IDE, SATA, SAS, USB, cartões de memória SDCard, entre outros. Possuem também a vantagem de serem portáteis, podendo ser levados a campo. As figuras 1 e 2 ilustras os equipamentos.

Figura 1 - Solo IV Fonte da foto: https://portuguese.alibaba.com

Figura 2: Tableaut TD3 Fonte da foto: http://www.forensiccomputers.com/

Se usar um equipamento comercial especializado em duplicação de dados não for uma opção, existem soluções de baixo custo para esse processo. Uma forma de realizar essa cópia é usar uma distribuição Linux preparada para análises forenses. Estas distribuições permitem que se monte o disco original do suspeito no modo "somente leitura". Uma vez montado o disco das evidências, as cópias podem ser feitas por programas que acompanham essas distribuições, como, por exemplo, o dd, dc3dd, dcfldd, entre outros. Esses programas farão uma cópia de todos os bits do disco de origem, inclusive áreas não alocadas. Alguns deles já realizarão também o cálculo do *hash* dos dados originais e do arquivo de destino.

Duas distribuições que fornecem ferramental forense são a Deft Linux (http://www.deftlinux.net/) e Caine (http://www.caine-live.net/).

2.5.2. Processamento e Análise dos Dados

Uma fez feita a duplicação pericial dos dados e tendo garantido a sua integridade por meio do *hash*, o próximo passo é o processamento e análise de dados. Essa fase consiste na recuperação dos dados que estão nas mídias, muitos deles apagados, e a disponibilização desses dados aos peritos de modo que possam ser feitas pesquisas sobre eles. Dessa forma, as principais ferramentas dessa etapa do processo deverão entender os sistemas de arquivos envolvidos, executar técnicas de recuperação de dados apagados, indexar esses dados para futuras pesquisas e interpretar essas informações de modo que o grande volume de dados possa ser organizado em subgrupos e tipos para facilitar a análise dos peritos.

Para essa tarefa, os principais softwares comerciais são Encase (https://www.guidancesoftware.com/encase-forensic), FTK (http://accessdata.com/solutions/digital-forensics/forensic-toolkit-ftk), entre outros. Alternativamente, o IPED (Indexador e Processador de Evidências Digitais) é uma solução desenvolvida por peritos criminais da Polícia Federal que tem se mostrado bastante interessante e está disponível para o uso de peritos de outras instituições de segurança pública. Por fim, existem as opções livres, como The Sleuth Kit - TSK - e Autopsy (http://www.sleuthkit.org/). Analisaremos esses dois últimos com mais detalhes na próxima seção.

2.5.2.1 – The Sleuth Kit e Autopsy

The Sleuth Kit (TSK) é um conjunto de ferramentas de linha de comando e bibliotecas em C para análise de disco rígidos e recuperação de arquivos. O Autopsy é um ambiente gráfico que proporciona uma interface mais amigável sobre o TSK. Ambas as ferramentas são livres, de código aberto e estão em constante desenvolvimento pelos seus mantenedores.

A importância didática de ferramentas livres é ressaltado por Fagundes, Neukamp e Silva (2011), que apontam que o software de código aberto é uma modelo didático, pois fomenta o pensamento crítico, conta com uma capacidade de adaptação independente, conta com uma comunidade, na qual há compartilhamento de conhecimento e possibilita ao aluno, mesmo fora do ambiente acadêmico, acesso às ferramentas de forma legal.

Segundo Carrier (2006), o TSK é composto por mais de 20 programas, estilo linha de comando, organizados em grupos. Os grupos em que os programas são divididos são baseados nas entidades das estruturas dos sistemas de arquivos. São eles: categoria de sistemas de arquivos, categoria de conteúdo, categoria de metadados, categoria de aplicação e categorias múltiplas.

Pelos comandos do TSK, é possível examinar cada uma das entidades do sistema de arquivos. Para usá-los em sua plenitude, é necessário um entendimento de como os disco rígidos são estruturados e como as estruturas lógicas dos sistemas de arquivos funcionam.

Os programas do TSK são ótimas ferramentas para destrinchar os dados de um disco. Tem um papel didático importante e servem como os blocos de construção para ferramentas mais integradas, porém são pouco eficientes para lidar com diversos casos, nos quais o interesse é a recuperação do maior número de dados possível e a correta visualização deles, em tempo hábil.

Dessa forma surge a necessidade de se utilizar uma ferramenta que integre os diversos programas do TSK e forneça uma interface mais produtiva. Uma opção é o Autopsy.

O Autopsy utiliza as bibliotecas do TSK e apresenta uma interface gráfica intuitiva para o processamento dos dados a serem analisados. Após entrar com alguns dados sobre o caso, deve-se informar o arquivo de imagem, que é a cópia forense realizado conforme descrito anteriormente. Este arquivo pode estar no formato bruto, também conhecido com *raw* ou dd ou em algum outro formato usado por algum *software* ou equipamento de duplicação de dados. Um formato bastante popular é o formato E01, introduzido pela EnCase e usado por vários outros programas.

As figuras 3 e 4 ilustram duas telas do *Autopsy*.

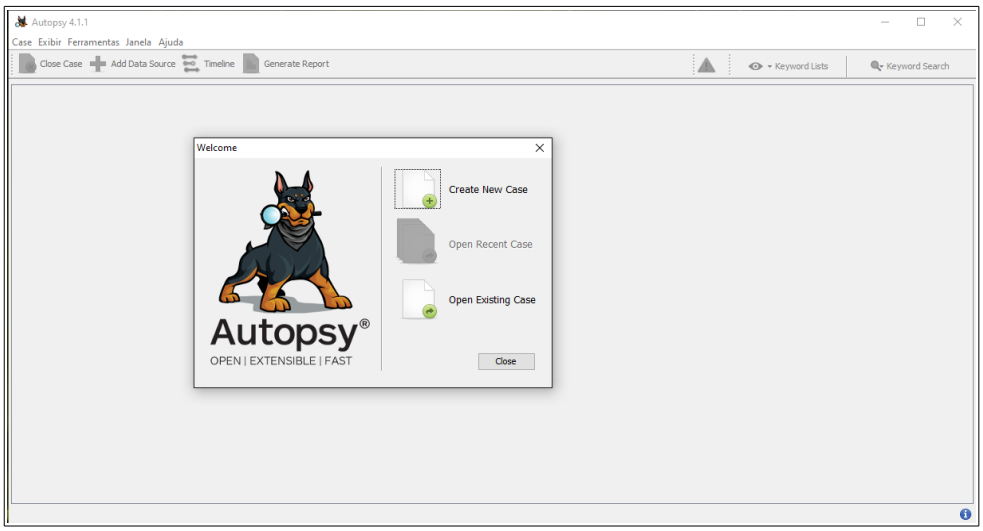

Figura 3: Autopsy

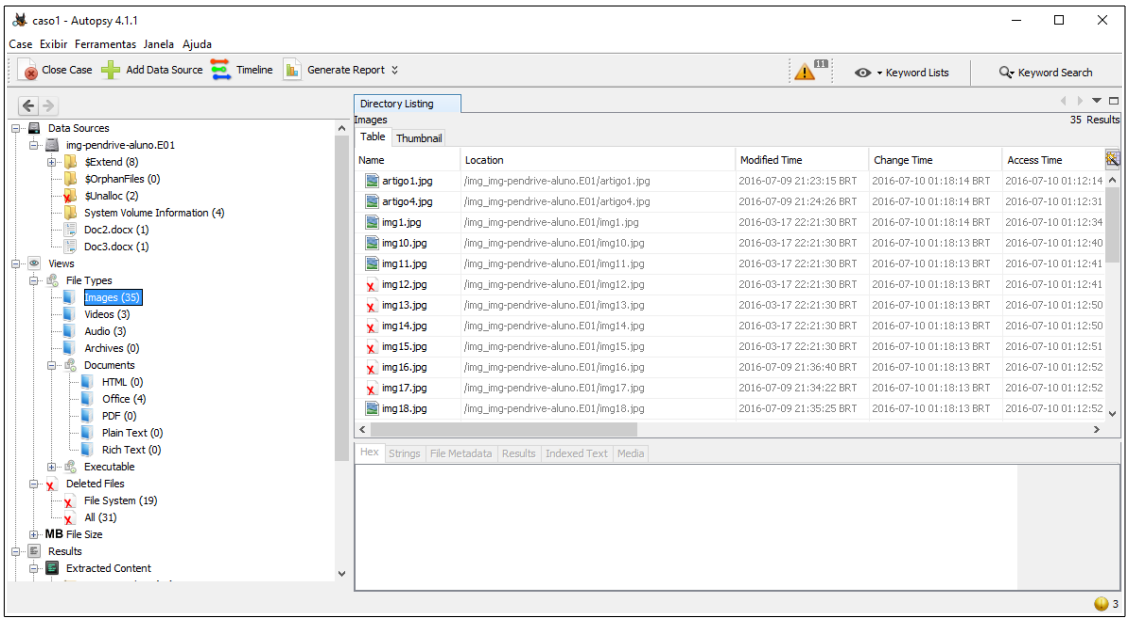

Figura 4: Autopsy

2.6. Princípios da Recuperação de Evidências Digitais

Essa seção tem como objetivo apresentar conceitos técnicos que permitam entender como as ferramentas usadas no laboratório de computação forense conseguem chegar aos resultados que se propõem.

Ter o conhecimento técnico do que está sendo feito e não apenas confiar nessas ferramentas e equipamentos como verdadeiras caixas-pretas, que apenas apresentam o resultado, permitirá ao perito uma melhor explanação técnica acerca do que se está periciando, além de subsidiá-lo com conhecimentos suficientes para responder a possíveis questionamentos das partes ou do juízo.

2.6.1. Mídias de Armazenamento

2.6.1.1. Discos rígidos

Os discos rígidos ainda são a principal mídia de armazenamento em massa. São a única parte do computador com componentes mecânicos. Por possuírem peças móveis, são vulneráveis a choques mecânicos. Internamente são compostos por discos magnéticos, nos quais as suas superfícies podem ser magnetizadas para representar bits 0 ou 1. Possuem um desempenho inferior aos componentes eletrônicos do computador, podendo ser um gargalo no desempenho.

Para ler e gravar dados no disco, são usadas cabeças de leitura eletromagnéticas que são presas a um braço móvel, o que permite seu acesso a todo o disco. Para que o disco rígido possa posicionar a cabeça de leitura sobre a área exata referente à trilha que vai ser lida, existem sinais de sincronismo gravados nas superfícies do disco que orientam o posicionamento da cabeça de leitura. Eles são sinais magnéticos especiais, gravados durante a fabricação dos discos, também conhecida como formatação física (Marimoto, 2010).

2.6.1.2. SSDs

Os discos de estado sólido (Solid State Disk – SSD) são memórias de armazenamento permanente. É um tipo de memória *flash*. São memórias eletrônicas que não precisam de alimentação para reter as informações. São constituídas de células compostas por transistores e uma fina camada de óxido de silício que funciona como uma espécie de armadilha para elétrons.

Marimoto (2010) aponta como vantagem dos SSDs o tempo de acesso baixo, com excelentes taxas de leitura e gravação, o que melhora o desempenho consideravelmente em uma grande gama de aplicativos e reduz bastante o tempo de *boot*, tornando o sistema muito mais respondível. Os SSDs também oferecem um consumo elétrico mais baixo, são silenciosos, resistentes a impactos e oferecem uma maior segurança contra perda de dados devido a defeitos de hardware, já que não possuem partes móveis.

2.6.2. Sistemas de Arquivos

Um sistema de arquivos é a estrutura lógica utilizada pelo computador para organizar os dados em um meio de armazenamento físico. Ele gerencia procedimentos relacionados a arquivos, tais como, criação, abertura, modificação, remoção etc. Entre os principais, pode-se listar: FAT 12 / 16 / 32, exFAT, NTFS, Ext2, Ext3 e Ext4.

2.6.2.1. FAT

Considerado um dos sistemas de arquivos mais simples. Foi introduzido com o Microsoft DOS e usado como sistema de arquivos padrão de algumas versões do Windows. Ainda é usado em mídias de armazenamento do tipo *flash*, como cartões de memória SDCard e *pendrives*.

Segundo Carrier (2006), um dos motivos de ser considerada simples é possuir um número pequeno de estrutura de dados. As duas principais estruturas são a FAT (*File Allocation Table)* e as entradas de diretório. O conceito de funcionamento básico desse sistema de arquivo é que cada diretório ou arquivo criado aloca uma estrutura de dados denominada entrada de diretório. Nessa estrutura ficam armazenados o nome do arquivo, o tamanho, o endereço do bloco inicial do arquivo, os carimbos de tempo (data de criação, modificação e último acesso) e outros metadados. Se o arquivo for maior do que um bloco, é usada a estrutura de dados FAT para armazenar a sequência de blocos que formam o arquivo.

A figura 5 ilustra a entrada de diretórios. A figura 6 ilustra a FAT.

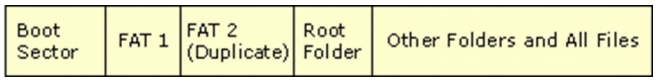

Figura 5

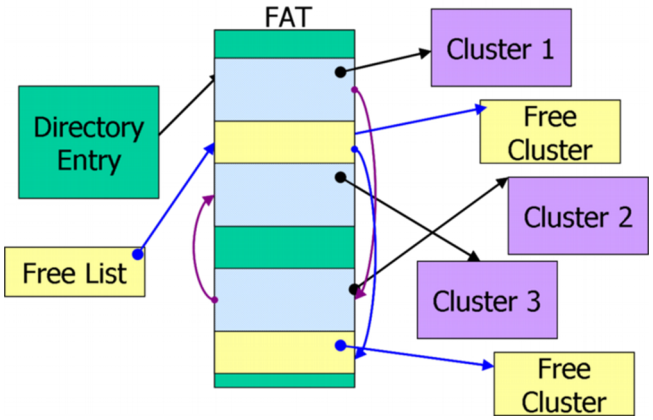

Figura 6

No disco, o sistema de arquivos FAT ocupa três regiões lógicas distintas. A área reservada, que contém informações sobre o sistema de arquivos, a área FAT que contém a estrutura de dados FAT primária e uma cópia de segurança dessa estrutura e a área de dados, onde estarão armazenados os arquivos.

Existem quatro versões do sistema de arquivos FAT.

A FAT 12, usada nos antigos disquetes e discos rígidos “pré-históricos”. Permite a utilização de blocos de 512 bytes a 4KB, podendo endereçar de 2MB a 16MB. A FAT 16, usada em disco rígidos muito antigos. Permite blocos de 2KB a 32KB, podendo endereçar 128MB a 2GB. A FAT 32, usada em discos rígidos antigos e nas atuais

memórias *flash* (pendrives, cartões de memória etc), permite utilizar blocos de 4KB a 32KB, podendo endereçar 1TB a 2TB. O campo de tamanho de arquivo na FAT é de 32 bits, limitando cada arquivo a no máximo 4GB (2^{32}). Finalmente, a exFAT ou FAT64. Possui limite máximo do tamanho de cada arquivo de 16 Exabytes. Sua capacidade teórica de armazenamento é de 64 ZB (zettabyte), mas a Microsoft não recomenda capacidades acima de 512 TB. Possui suporte para um número maior de arquivos no mesmo diretório (1000). Implementa uma melhor alocação e gerência do espaço livre em disco devido à introdução de uma nova organização da memória (*bitmap*) e possui suporte a lista de controle de acesso.

2.6.2.2. NTFS

O NTFS (*New Technologies File System*) é o atual sistema de arquivos padrão do Windows.

NTFS foi desenvolvido para ser confiável, seguro, com suporte para dispositivos de grande capacidade de armazenamento. Um conceito importante do NTFS é que todos os dados e metadados são armazenados em arquivos. Não existe um layout predefinido para diferentes áreas do sistema de arquivos, exceto para o setor de *boot*. Um outro recurso adicionado ao NTFS é o *journal*, que permite a recuperação do sistema de arquivos após determinadas falhas. Com o recurso de *journal*, as alterações realizadas no sistema de arquivos são gravadas no arquivo $LogFile. Permite a implementação do conceito de transações, como *redo*, *undo* e *commit*. (Carrier, 2006).

Os metadados são armazenados em arquivos ocultos, no diretório raiz, denominados *metafiles*. Todos eles começam com o caractere “$”. A figura 7 lista os principais *metafiles*.

A principal estrutura do NTFS é a MFT (*Master File Table*). Ela contém informações sobre todos os arquivos e diretórios. Cada arquivo ou diretório tem, pelo menos, uma entrada na tabela MFT. Uma entrada nessa tabela é composta por um cabeçalho e alguns atributos. Cada tipo de atributo tem uma função própria. Podem ser residentes (atributos pequenos armazenados na própria entrada de diretório da MFT) ou não residentes (o cabeçalho do atributo fica na MFT, mas seu conteúdo fica em blocos do disco rígido). A figura 8 apresenta uma lista de alguns atributos.

Entrada MFT	Nome Arquivo	Descrição
0	$MFT	Entrada para a própria MFT.
1	$MFTMirr	Contém backup das primeiras entradas da MFT.
2	$LogFile	Arquivo que armazena o *journal* do NTFS.
3	$Volume	Contém informações do volume, como por exemplo, no nome, identificação e versão do volume.
4	$AttrDef	Contém dados de atributos.
5	.	Contém o diretório raiz do sistema de arquivos.
6	$Bitmap	Contém o estado de alocação de cada bloco do sistema de arquivos.
7	$Boot	Contém o setor de *boot* e o código de *boot*. É o único arquivo do NTFS quem tem a localização estática no disco. Seu conteúdo sempre está no setor 0 do sistema de arquivos.
8	$BadClus	Contém a lista dos blocos que possuem setores defeituosos.
9	$Secure	Contém informações sobre segurança e a lista de controle de acessos do sistema de arquivos.
10	$Upcase	Contém a versão em caixa alta de cada caractere Unicode.
11	$Extend	Diretório que contém arquivos para extensões opcionais.

Figura 7: Metafiles

Tipo	Nome	Descrição
16	$STANDARD_ INFORMATION	Informações gerais, como por exemplo, datas de criação, modificação e acesso; proprietário do arquivo; ID de segurança etc.
32	$ATTRIBUTE_LIST	Lista de atributos de arquivos.
48	$FILE_NAME	Nome do arquivo e as últimas datas de acesso, criação e modificação do arquivo.
64	$VOLUME_VERSION	Informações sobre o volume (apenas na versão 1.2 do NTFS).
64	$OBJECT_ID	Identificador único de 16 *bits* de arquivos e diretórios (versão 3.0+ do NTFS).
80	$SECURITY_ DESCRIPTOR	Contém controle de acesso e propriedades de segurança de um arquivo (obsoleto, usado em versões da NTFS anteriores a 3.0).
96	$VOLUME_NAME	Nome do volume.
112	$VOLUME_ INFORMATION	Versão do sistema de arquivos e outros *flags*.
128	$DATA	Conteúdo de arquivo.
144	$INDEX_ROOT	Nó raiz de uma árvore de índices.
160	$INDEX_ALLOCATION	Nós de uma árvore de índices com raiz em $INDEX_ROOT.
176	$BITMAP	Mapa de *bits* para o arquivo $MFT e para os índices.
192	$SYMBOLIC_LINK	Informações de ligações flexíveis (apenas NTFS versão 1.2).
192	$REPARSE_POINT	Informações sobre ponto de *reparse*, que é usado para ligações flexíveis.
208	$EA_INFORMATION	Usado para compatibilidade com OS/2.
224	$EA	Usado para compatibilidade com OS/2.
256	$LOGGED_UTILITY_ STREAM	Contém chaves e informações sobre atributos criptografados (versão 3.0+ do NTFS – *Windows* 2000+).

Figura 8: Atributos MFT

Os carimbos de tempo (*timestamps*) do NTFS ficam armazenados nos atributos $STANDARD_INFORMATION e $FILE_NAME. São quatro tipos: data de criação, data de modificação (alterações no $DATA ou $INDEX), data de acesso e data

modificação MFT (não visível para usuários do Windows). Os carimbos de tempo são campos de 64 bits, com precisão de nanosegundos.

2.6.2.3. Ext2, Ext3 e Ext4

São os sistemas de arquivos padrão do Linux. Foram projetados para serem rápidos e confiáveis. A cada versão foram adicionadas novas funcionalidades. A principal diferença entre Ext2 e Ext3 é que nesse último foi adicionado suporte a *journal*, com funcionalidade semelhante ao recurso de *journal* discutido no NTFS. A Ext4, versão mais atual dessa família de sistemas de arquivos, adicionou novos recursos, tais como, alocação tardia (*delayed allocation*), carimbos de tempo com maior resolução (nanossegundos), verificação de integridade do *journal* (*journal checksums*), suporte para tamanhos maiores de volumes e arquivos, pré alocação de arquivos e sistemas de verificação mais rápidos. O Ext4 também é utilizado por algumas versões do sistema operacional Android.

As informações sobre o layout básico do sistema de arquivos são armazenados numa estrutura de dados denominada superbloco, que fica armazenada no começo do sistema de arquivos. O conteúdo dos arquivos fica armazenado em estruturas denominadas blocos, que são agrupamentos de setores consecutivos da mídia de armazenamento. Os metadados de cada arquivo e diretório são armazenados em uma estrutura de dados denominada *i-node*. Os *i-nodes* ficam armazenados na tabela de *i-nodes*. Existem várias tabelas de *i-nodes* distribuídas pelo sistema de arquivo, uma para cada agrupamento de blocos. Os nomes dos arquivos são armazenados em uma estrutura de dados denominada entrada de diretório. Além do nome do arquivo, essa estrutura armazena um ponteiro para o *i-node* relacionado ao arquivo (Carrier, 2006).

A figura 9 ilustra as estruturas de dados do sistema de arquivos Ext.

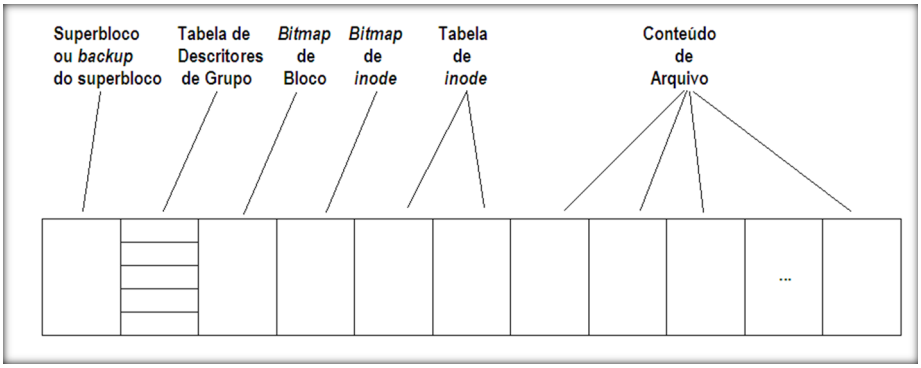

Figura 9

O *i-node* armazena diversos metadados dos arquivos, entre eles, permissões, tamanho do arquivo, os carimbos de tempo (*timestamp*) e a lista de blocos que armazenam o conteúdo do arquivo (figura 10).

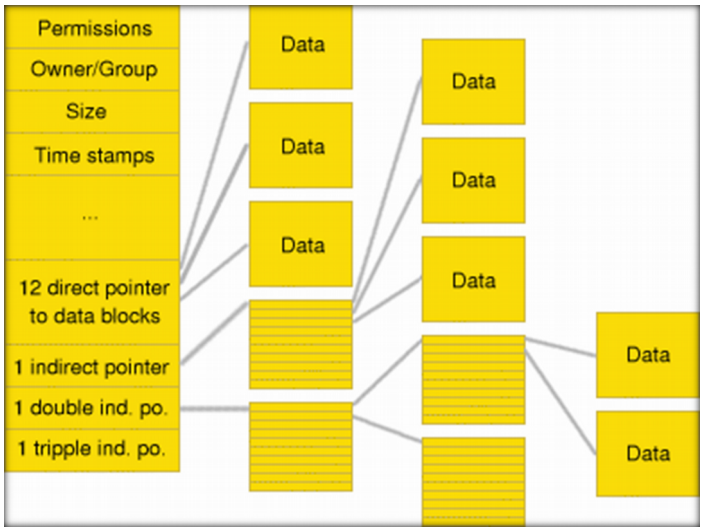

Figura 10

2.6.4. Técnicas de Recuperação de Arquivos Apagados

Apesar das peculiaridades de cada sistema de arquivos e das técnicas para recuperá-los, basicamente a recuperação de dados apagados é possível porque ao se apagar um arquivo, ele é apagado apenas logicamente do sistema de arquivos, ou seja, o espaço ocupado por aquele arquivo é liberado para reutilização, mas por questões de desempenho o seu conteúdo permanece intacto até que aquele espaço seja necessário para alocar outro arquivo.

Algumas técnicas de recuperação de dados levam em conta a estrutura de dados providas pelos sistemas de arquivos. Para entender como esse tipo de recuperação de dados é possível, precisamos entender, primeiramente, o que acontece em cada sistema de arquivos quando um arquivo é apagado.

Como mostrado por Carrier (2006), nos sistemas de arquivos FAT ao se apagar um arquivo, o primeiro caractere na tabela de entrada de diretório é substituído por 0xe5 e os endereços na tabela FAT são zerados. Para recuperá-lo, deve-se encontrar o nome dele na tabela de diretório, o endereço do primeiro bloco e os metadados que informam o tamanho do arquivo. De posse da informação de qual é o primeiro bloco e o tamanho do arquivo, a recuperação é trivial. Porém, arquivos fragmentados podem inviabilizar a recuperação pelo uso apenas dessa técnica.

No NTFS, quando um arquivo é apagado, a entrada de diretório na MFT desse arquivo é marcada como não alocada e os blocos desse arquivo são adicionados na tabela de blocos livres. Com isso, a estrutura de alocação de arquivos permanece praticamente intacta, permitindo a recuperação do arquivo até que a entrada de diretório seja reutilizada.

No Ext2, o *i-node* da entrada de diretório é apagado. Para recuperar arquivos apagados, deve-se pesquisar por *i-nodes* não alocados. Encontrando-se um *i-node* não alocado, ele conterá a lista de blocos daquele arquivo apagado. No Ext3 e Ext4, o *i-node* da entrada de diretório não é apagado, porém, os campos com os endereços dos

blocos no *i-node* são apagados. Dessa forma, tem-se o *i-node* de determinada entrada de diretório (nome do arquivo), porém não se consegue obter a lista de blocos que compunham esses arquivos. A recuperação de arquivos no Ext3 e Ext4 é mais difícil que no Ext2.

Uma outra técnica promissora de recuperação de arquivos apagados é o *data carving.* No processo clássico de *data carving*, as estruturas do sistema de arquivos não são levadas em consideração.

Merola (2008) cita o exemplo de arquivos PDF e JPEG. Os arquivos PDF possuem uma assinatura inicial, ou seja, começam sempre da mesma forma, o que permite distingui-los de outros tipos de arquivos examinado apenas seu conteúdo. Dessa forma, todos os arquivos PDF iniciarão com os caracteres "%PDF". Essa assinatura também é conhecida como cabeçalho do arquivo. Alguns arquivos, além do cabeçalho, possuem também um rodapé, ou seja, sempre terminarão com o mesmo caractere. No caso dos PDFs será "%EOF". Para arquivos JPEG, teremos os padrões "0xFFD8" para o cabeçalho e "0xFFD9" para o rodapé.

É com base nas assinaturas dos arquivos que as técnicas básicas de *data carving* funcionam. Uma ferramenta empregando essa técnica terá uma ampla base de assinaturas dos mais variados tipos de arquivos. Uma vez identificado o início de um arquivo, a ferramenta irá considerando que tudo o que virá depois dessa assinatura é o corpo do arquivo. Ao encontrar o rodapé, a ferramenta conclui a recuperação daquele arquivo e o processo de repete a partir do próximo byte, até que todos os bytes não alocados da mídia de armazenamento sejam processados.

Porém, dificuldades podem ser encontradas nesse processo. Arquivos podem possuir cabeçalho, mas não rodapé. Arquivos podem estar também fragmentados, compactados ou incompletos. Para lidar com essas questões, as técnicas mais avançadas de *data carving* baseiam-se não apenas nas assinaturas dos arquivos, mas também possuem conhecimento das estruturas internas de cada tipo de arquivo, o que permite às ferramentas tentar encaixar todas as peças, num verdadeiro quebra-cabeça de bytes e fragmentos de estruturas de arquivos.

Em relação à recuperação de arquivos nos discos de estado sólido (SSDs), deve-se notar que a dinâmica de leitura e escrita de dados difere dos discos rígidos magnéticos tradicionais, impactando nas técnicas de recuperação de evidências digitais.

Conforme explicado por Gomes (2012), diferentemente dos disco rígidos, nos quais os dados podem ser apagados e sobrescritos de maneira independente, nos SSDs as páginas na memória *flash* não podem ser simplesmente regravadas. Sempre que se precisa gravar dados em uma página já ocupada, a controladora do SSD precisa primeiro apagar os dados anteriores, levando a célula ao seu estado original, para só então, realizar a nova operação de escrita. Além disso, não é possível apagar apenas uma página, deve-se apagar um bloco de páginas. Se houver informações válidas nessas páginas, elas precisam ser copiadas e depois reescritas. Todas essas operações podem comprometer o desempenho do SSD.

Para lidar com essas características, os SSDs utilizam técnicas de coleta de lixo (*garbage collection*). O coletor de lixo será executado em segundo plano, pelo próprio

hardware do SSD e será responsável por garantir que sempre haja blocos livres, em estado original, prontos para escrita. Para garantir isso, uma de suas tarefas é mover dados, realizando uma espécie de desfragmentação do disco. Essa característica tem uma impacto negativo sobre a recuperação de arquivos apagados, tendo em vista que a chance de sobreposição de dados não alocados é bem maior por conta do coletor de lixo.

2.6.5. Perícias em Dados Voláteis

Informações preciosas podem estar armazenadas apenas na memória RAM. Se o conteúdo do disco rígido estiver criptografado, fazer a extração e análise dos dados voláteis pode possibilitar a obtenção da chave usada para proteger os dados do disco. Outras informações como processos em execução e bibliotecas de software carregadas também podem ser obtidas por meio desse tipo de análise.

Silva e Lorens (2009) discorrem sobre a necessidade de um exame pericial em memória RAM, também conhecido como *live forensics*, tendo em vista que circunstâncias específicas justificam a realização de procedimentos de coleta de vestígios digitais no local em que se encontram instalados os equipamentos computacionais, enquanto ligados e em funcionamento normal. Instalações de equipamentos de grande porte, não convencionais, ou que suscitem o risco de perda de informações significativas ou ainda, a inviabilização da perícia são exemplos dessas circunstâncias. Destaca-se, também, a situação cada vez mais frequente do uso de criptografia nas mídias de armazenamento.

A primeira tarefa a ser realizada em uma perícia de dados voláteis é obter uma cópia da memória RAM. O termo *dump* de memória também é usado para se referir a este tipo de cópia. Existem várias ferramentas que podem ser usadas para essa tarefa. É interessante que essa ferramenta possa ser executada na máquina alvo sem a necessidade de instalação, para não escrever no disco e correr o risco de sobrescrever algum dado não alocado. Um exemplo de ferramenta livre para Windows que faz a cópia de memória é o *FTK Imager Lite* (figura 11).

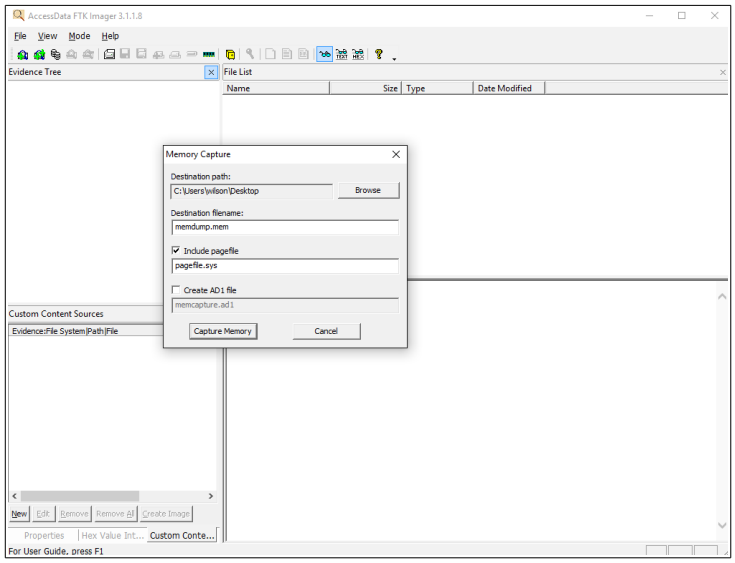

Figura 11

Obtida a cópia da memória RAM, é preciso saber interpretá-la. Para essa tarefa existem softwares que podem auxiliar o perito. Um deles é o *framework* livre *Volatility (*http://www.volatilityfoundation.org/*)* (figura 12).

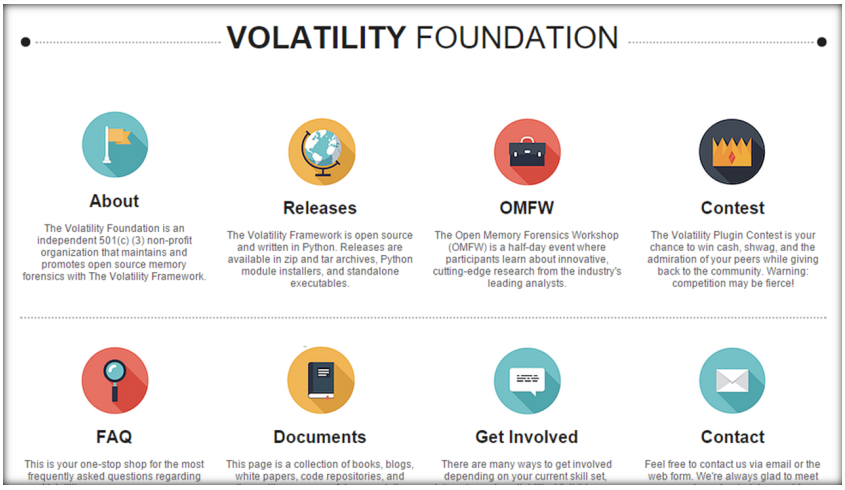

Figura 12

O *Volatility* é um conjunto de ferramentas abertas, escritas em *Python*, destinado à extração de conteúdos digitais armazenados em memória volátil de sistemas operacionais Windows. Realiza interpretação (*parser*) de *dump* de memória, *crash dump*, arquivo de hibernação, *snapshot* de máquinas virtuais etc.

Com o uso desse *framework,* podem ser obtidos dados referentes a processos em execução, soquetes de rede abertos, DLLs carregadas para cada processo, arquivos abertos para cada processo, chaves de registro para cada processo, memória endereçável de um processo, módulos do *kernel* do sistema operacional, chaves criptográficas, entre outros.

2.6.6. Busca de evidências no Registro do Windows

O registro do Windows é um banco de dados hierárquico que armazena uma grande quantidade de informações sobre a configuração do Windows, aplicações instaladas e informações sobre atividades dos usuários que interagiram com o sistema operacional. Boa parte dessa informação pode ser de interesse da perícia computacional forense.

Em Carvey (2009), é discutida a importância da análise do Registro em caso de softwares maliciosos. Segundo o autor, softwares maliciosos podem deixar rastros digitais no Registro e interpretar tais rastros dará, ao analista, pistas sobre o comportamento do programa malicioso.

As informações do Registro são organizadas de forma hierárquica. No nível mais alto, existem cinco chaves principais ou raízes. A figura 13 descreve essas chaves.

Chave raiz	Descrição
HKEY_CURRENT_USER*	Armazena informações, *profiles* e preferências do usuário que está localmente conectado ao sistema no momento. O *profile* do usuário fica localizado em \Documents and Settings\Nome Usuário\Ntuser.dat, e a partir do *Windows Vista*, em \Users\Nome Usuário\Ntuser.dat.
HKEY_USERS	Armazena informações sobre todas as contas de usuários do sistema.
HKEY_CLASS_ROOT*	Armazena as associações de arquivos e informações de registro dos objetos COM (*Component Object Model*).
HKEY_LOCAL_MACHINE	Armazena a maior parte das configurações do sistema operacional. Exemplos de sub-chaves: BCD (*Boot Configuration Database* – apenas a partir o *Windows Vista*), COMPONENTS, HARDWARE, SAM, SECURITY, SOFTWARE e SYSTEM.
HKEY_CURRENT_CONFIG*	Armazena informações sobre a configuração atual de hardware.
HKEY_PERFORMANCE_ DATA	Armazena informações sobre o desempenho do sistema. (Só é possível acessar essa chave por meio das APIs de programação do *Windows*).

Figura 13

* Essas chaves são, na verdade, links para outras chaves que ficam armazenadas embaixo das chaves raízes que não são links. Devido à importância delas, a Microsoft adicionou os links ao nível raiz.

Cada chave pode ter associado um tipo de valor. Os tipos de valores possíveis estão descritos na figura 14.

Tipo	Descrição
REG_NONE	Nenhum valor.
REG_SZ	String Unicode de tamanho fixo.
REG_EXPAND_SZ	String Unicode de tamanho variável que pode ter variáveis de ambiente.
REG_BINARY	Dados binários.
REG_DWORD	Número de 32 *bits*.
REG_DWORD_ LITTLE_ENDIAN	Número de 32 *bits* com bytes menos significativos primeiro.
REG_DWORD_BIG_ ENDIAN	Número de 32 *bits* com bytes mais significativos primeiro.
REG_LINK	Ligação simbólica Unicode.
REG_MULTI_SZ	Arranjo de strings Unicode terminadas em zero.
REG_RESOURCE_LIST	Descrição de recursos de hardware.
REG_FULL_RESOURCE_ DESCRIPTOR	Descrição de recursos de hardware.
REG_RESOURCE_ REQUIREMENTS_LIST	Lista de recursos.
REG_QWORD	Número de 64 *bits*.
REG_QWORD_LITTLE_ ENDIAN	Número de 64 *bits* com bytes menos significativos primeiro.

Figura 14

Fisicamente, os dados do Registro ficam armazenados em arquivos denominados *HIVE*. A descrição desses arquivos é apresentada na figura 15.

Chave do Registro	Arquivo *Hive*
HKEY_LOCAL_MACHINE\System	\Windows\System32\config\System
HKEY_LOCAL_MACHINE\SAM	\Windows\System32\config\SAM
HKEY_LOCAL_MACHINE\Security	\Windows\System32\config\Security
HKEY_LOCAL_MACHINE\Software	\Windows\System32\config\Software
HKEY_LOCAL_MACHINE\Hardware	Chave volátil, armazenada apenas na RAM.
HKEY_LOCAL_MACHINE\System\ Clone	Chave volátil, armazenada apenas na RAM.
HKEY_USERS\[SID Usuário]	\Documents and Settings\[usuário]\NTUSER.DAT (até Windows XP) \Users\[usuário]\NTUSER.DAT (a partir do Windows Vista)
HKEY_USERS\Default	\Windows\System32\config\Default

Figura 15

O Registro é uma grande fonte de informação, mas para ser útil ao perito, esses dados devem ser extraídos e interpretados. Fazer essa atividade sem ajuda de alguma ferramenta é contra producente. Uma opção de automatizar esse processo de extração, interpretação e apresentação desses dados é a ferramenta RegRipper.

O RegRipper é um *framework* composto por uma coleção de *scripts* escritos na linguagem Perl. Os *scripts* funcionam como *plug-ins* do *framework*. Novos *scripts* podem ser adicionados ou escritos por terceiros.

O RegRipper lê as informações do Registro diretamente dos arquivos (*hives*) que as armazenam, interpreta esses dados e os disponibiliza para o usuário.

A seguir são apresentadas algumas das informações que podem ser obtidas do Registros, e os comandos do RegRipper para obtê-las.

Nome do computador:

Informação encontrada na chave SYSTEM\CurrentControlSet\Control\ ComputerName\ActiveComputerName.

```
C:\RegRipper>rip -p compname -r f:\T\system
Launching compname v.20080324
ComputerName = COMPUTADORNTFS
```

Informações relacionadas à versão do sistema operacional:

Informações encontradas nas chaves: SYSTEM\ControlSet00x\Control\Windows e SOFTWARE\Microsoft\Windows NT\CurrentVersion

```
C:\RegRipper>rip -p winnt_cv -r f:\T\software
Launching winnt_cv v.20080609
WinNT_CV
Microsoft\Windows NT\CurrentVersion
LastWrite Time Wed Nov 18 09:25:43 2009 (UTC)
```

```
SubVersionNumber :
  RegDone :
  RegisteredOrganization : .
  RegisteredOwner : .
  CurrentVersion : 5.1
  CurrentBuildNumber : 2600
SoftwareType : SYSTEM
SourcePath : F:\I386
SystemRoot : C:\WINDOWS
PathName : C:\WINDOWS
CSDVersion : Service Pack 2
CurrentType : Multiprocessor Free
ProductName : Microsoft Windows XP
ProductId : 55274-640-8816093-23950
BuildLab : 2600.xpsp_sp2_rtm.040803-2158
InstallDate : Mon Oct 26 15:26:01 2009 (UTC)
CurrentBuild : 1.511.1 () (Obsolete data - do not use)
```

Ressalta-se que o campo "LastWrite Time Wed Nov 18 09:25:43 2009 (UTC)" é a data e horário em que o sistema foi desligado (shutdown) pela última vez.

Interfaces de rede e endereço IP

Informação encontrada na chave ControlSet00x\Services\Tcpip\Parameters\Interfaces.

```
C:\RegRipper>rip -p networkcards -r f:\T\software
Launching networkcards v.20080325
NetworkCards
Microsoft\Windows NT\CurrentVersion\NetworkCards
AMD PCNET Family PCI Ethernet Adapter

C:\RegRipper>rip -p nic_mst2 -r f:\T\system
Launching nic_mst2 v.20080324
Network key
ControlSet001\Control\Network\{4D36E972-E325-11CE-BFC1-08002BE10318}

ControlSet001\Services\Tcpip\Parameters\Interfaces
LastWrite time Mon Oct 26 15:20:28 2009 (UTC)

Interface {A69C78C2-98A2-4F6A-9FEC-534A380240B1}
Name: Conex ├úo local
Control\Network key LastWrite time Mon Oct 26 15:20:31 2009 (UTC)
Services\Tcpip key LastWrite time Fri Nov 27 14:20:03 2009 (UTC)
    DhcpDomain     = localdomain
    DhcpIPAddress  = 192.168.145.131
    DhcpSubnetMask = 255.255.255.0
    DhcpNameServer = 192.168.145.1
    DhcpServer     = 192.168.145.254
```

Outras informações que podem ser obtidas do Registro são *Wireless* SSIDs, lista de dispositivos móveis que foram conectados a USB, contas do usuário no sistema, atividades do usuário etc.

2.6.7. Busca de Evidências nos LOGs do Windows

Assim como o Registro do Windows, os *logs* do sistema são uma fonte de dados a ser analisada em determinados tipos de perícia.

Vários tipos de eventos do sistema operacional e atividades de programas e usuários são registrados nos *logs*. Existem quatro categorias padrão, que os dividem

conforme seu tipo: aplicação, segurança, instalação e sistema. A figura 16 ilustra o programa Visualizador de Eventos do Windows, no qual é possível ter acesso ao conteúdo dos *logs* e fazer pesquisas simples.

Caso haja necessidade de se fazer pesquisas mais elaboradas, pode-se usar a ferramenta de linha de comando LogParser. Com essa ferramenta é possível fazer pesquisas nos *logs* usando-se uma sintaxe similar as pesquisas realizadas em linguagem SQL. O LogParser é ilustrado na figura 17.

Fisicamente os *logs* ficam armazenados em arquivos específicos. No Windows XP, ficam armazenados na pasta "%SystemRoot%\System32\Config", nos arquivos sysevt.evt, secevent.evt, appevent.evt, entre outros. Do Windows 7 em diante ficam armazenados nas pastas "%SystemRoot%\System32\winevt\Logs", nos arquivos Application.evtx, Security.evtx, System.evtx, entre outros.

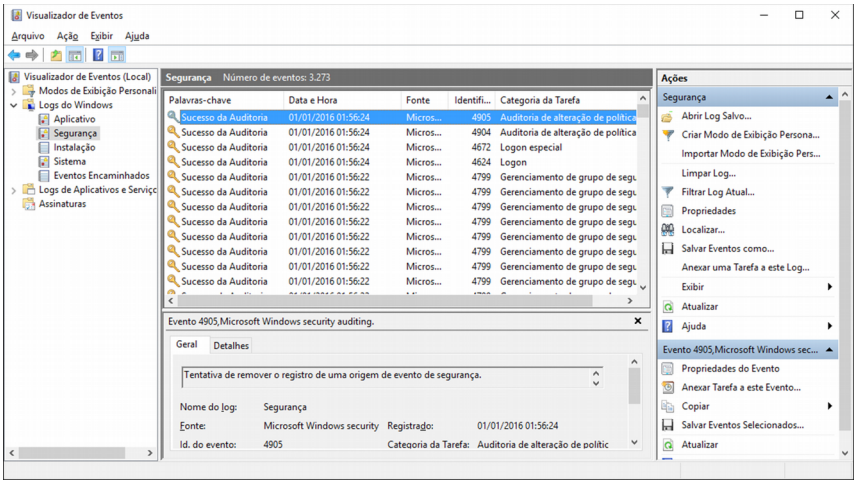

Figura 16

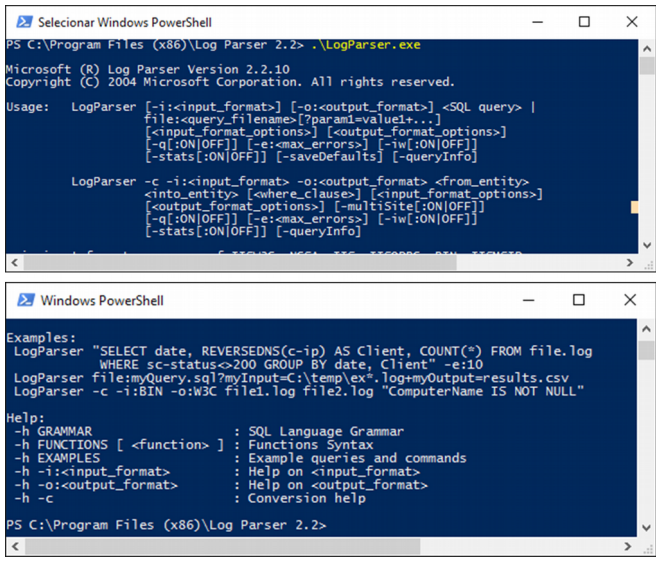

Figura 17

2.7. Técnicas Antiforenses e Anti-Antiforenses

Em uma definição de técnicas antiforenses encontrada em Velho et al (2016), os autores classificam-na como um conjunto de técnicas que objetivam inviabilizar, dificultar, iludir ou impossibilitar que a análise forense ocorra de forma satisfatória.

Basicamente, trata-se de formas de manipular os dados digitais de tal forma que esses dados sejam destruídos de maneira irrecuperável, ou, de alguma forma, não possam ser acessados pelo perito, ou ainda, que iludam o perito em suas conclusões.

Portanto, cabe ao perito conhecer sobre essas técnicas, saber identificá-las e contorná-las sempre que possível. No restante dessa seção são apresentadas as principais técnicas antiforenses e possíveis formas de lidar com elas.

2.7.1. *Wipe* ou Sanitarização de Dados

Como foi apresentado na seção 6, ao se apagar um arquivo, os dados desse arquivo não são realmente apagados, são feitas algumas alterações nas estruturas de controle de alocação de arquivos e aquele espaço ocupado pelo arquivo fica disponível para ser reutilizado, mas principalmente por motivo de desempenho, os dados desse arquivo apagado permanecem na mídia de armazenamento, até o momento em que forem reutilizados por outro arquivo. A partir desse momento, quando os dados são sobrescritos por um arquivo novo, a recuperação torna-se inviável.

É por isso que existe uma forma de apagar um arquivo de forma irrecuperável. Para isso, além de ser marcado nas estruturas do sistema operacional como apagado, o seu conteúdo em todo o disco deve ser sobrescrito. Existem inclusive protocolos para realizar essa técnica, conhecida como *wipe* ou sanitização de dados. Dependendo da sensibilidade e importância dos arquivos a serem apagados, esses protocolos recomendam que a área da mídia de armazenamento em que os dados estavam armazenados seja sobrescrita diversas vezes. A figura 18 ilustra o programa Disk Wipe, que entrega a técnica de sanitização de dados em uma mídia completa. Nota-se que se pode escolher qual protocolo de *wipe* utilizar. Na figura, são apresentados de cima para baixo os protocolos do menos para o mais seguro.

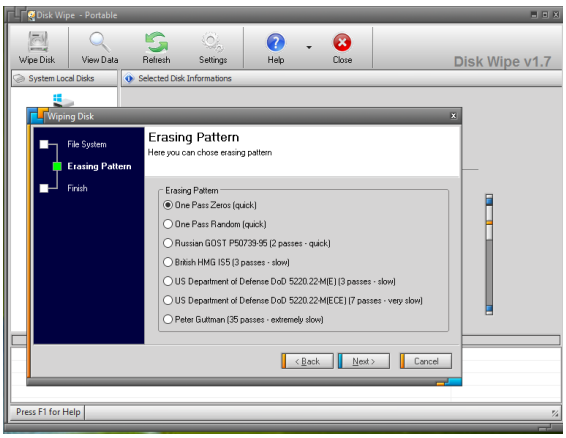

Figura 18

2.7.2. Criptografia e Quebra de Senhas

Criptografia passa a ser cada vez mais popularizada, sendo que vários sistemas já estão sendo configurados por padrão com seus dados criptografados. Há também diversas softwares que podem ser usados para criptografar arquivos, partes de uma mídia de armazenamento ou a mídia inteira.

Esses recursos, importantes para a segurança do usuário, representam um desafio técnico para os peritos, que precisam tentar ao máximo encontrar evidências digitais, mesmo que estas estejam protegidas por criptografia.

Desse cenário pericial, surge a necessidade de técnicas de quebra de criptografia e senhas, tornando essa área mais uma área da computação que deve ser dominada pelos peritos.

Mas como quebrar a criptografia, tendo em vista que a maioria dos sistemas utilizam criptografia forte, com algoritmos robustos e chaves grandes? A resposta encontra-se em atacar o elo mais fraco da segurança da informação, o usuário. Segundo Velho et al (2016), pesquisas mostram que a maioria dos usuários usa senhas fracas. A capacidade humana de memorização não facilita que usuários guardem como senhas sequências muitos grandes e aleatórias. Geralmente serão usadas expressões que são familiares aos usuários e as senhas tendem a se repetir em diversos sistemas.

Com isso, ao se deparar com conteúdo criptografado, o perito deve ao menos tentar as técnicas básicas de recuperação de senhas, limitando-se aos recursos computacionais e a um prazo de tempo de tentativa estipulado.

2.7.2.1 Ataques a Dados Criptografados

Podemos definir os ataques a sistemas com senha em dois tipos. Os ataques *on-line*, que visam sistemas que estão em funcionamento no momento dos ataques. Esse tipo de ataque é menos promissor, já que o tempo de resposta em que várias senhas podem ser testadas é alto.

Os ataques *offline* ou *post-mortem*, tentam decifrar os dados já obtidos das mídias de armazenamento, mas que ainda não estão acessíveis por estarem criptografados. É um ataque mais promissor que o anterior, pois a taxa de senhas que podem ser testadas é muito superior. Para a computação forense, esse é o tipo de ataque que mais interessa e é esse tipo que será discutido no restante da seção.

Para se quebrar a criptografia no ataque *offline* deve-se descobrir qual foi a senha usada pelo usuário para criptografar os dados. Para tanto, as possíveis senhas são testadas uma a uma. Porém, essa tarefa computacional é altamente paralelizável. Dessa forma, quanto mais processadores o perito tiver a disposição para a tarefa, mais rápido ela poderá ser cumprida. Atualmente, as duas formas mais utilizadas para paralelizar essa tarefa é por meio de *cluster* de computadores ou por meio de placas de processamento gráfico (GPUs).

Na opção de *cluster* de computadores, usa-se várias máquinas trabalhando em paralelo e em cooperação para o processamento dos ataques ao conteúdo criptografado.

Geralmente usa-se um esquema em que uma das máquinas é um ponto central que gerencia todas as demais, distribuindo a carga de processamento.

Outra forma de conseguir alto nível de paralelização é por meio do uso de GPUs. As GPUs são projetadas com diversos núcleos de processamento para atividades específicas. Essa arquitetura pode ser usada para paralelizar as computações necessárias para quebra de senhas. Uma GPU voltada para jogos, pode ter até aproximadamente 3.000 núcleos (*cores*). A figura 19 ilustra uma comparação da arquitetura *multicore* de uma CPU e de uma GPU.

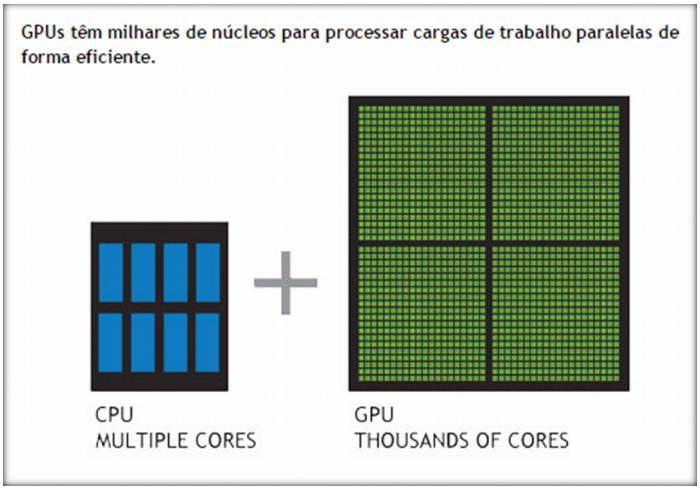

Figura 19

A figura 20 exibe um teste de desempenho de quatro configurações de máquinas usando GPUs processando quebra de criptografia em vários algoritmos.

- PC1: Windows 7, 32 bit · Catalyst 14.9 · 1x AMD hd7970 · 1000mhz core clock · oclHashcat v1.35
- PC2: Windows 7, 64 bit · ForceWare 347.52 · 1x NVidia gtx580 · stock core clock · oclHashcat v1.35
- PC3: Ubuntu 14.04, 64 bit · ForceWare 346.29 · 8x NVidia Titan X · stock core clock · oclHashcat v1.36
- PC4: Ubuntu 14.04, 64 bit · Catalyst 14.9 · 8x AMD R9 290X · stock core clock · oclHashcat v1.35

Hash Type	PC1	PC2	PC3	PC4
MD5	8581 Mh/s	2753 Mh/s	115840 Mh/s	92672 Mh/s
SHA1	3037 Mh/s	655 Mh/s	37336 Mh/s	31552 Mh/s
SHA256	1122 Mh/s	355 Mh/s	14416 Mh/s	12288 Mh/s
SHA512	414 Mh/s	104 Mh/s	4976 Mh/s	4552 Mh/s
SHA-3(Keccak)	179 Mh/s	92 Mh/s	3400 Mh/s	2032 Mh/s
RipeMD160	1810 Mh/s	623 Mh/s	23936 Mh/s	20016 Mh/s
Whirlpool	65845 kh/s	85383 kh/s	1480000 kh/s	1122304 kh/s
LM	1388 Mh/s	450 Mh/s	15616 Mh/s	16392 Mh/s
NTLM	16916 Mh/s	4185 Mh/s	250360 Mh/s	175808 Mh/s
NetNTLMv1	9108 Mh/s	2330 Mh/s	56448 Mh/s	97800 Mh/s
NetNTLMv2	589 Mh/s	200 Mh/s	7944 Mh/s	6496 Mh/s
WPA/WPA2	142 kh/s	48 kh/s	2096 kh/s	1536 kh/s

Figura 20 Fonte: www.hashcat.com, acesso em 22/09/2016

Mesmo com todo poder de processamento dos *clusters* e das GPUs, testar todas as combinações, num ataque denominado força bruta, não é viável para senhas com um tamanho e complexidade razoáveis (mais de 8 caracteres começa a inviabilizar a força bruta).

Por conta disso, deve-se tentar antes ataques mais inteligentes, nos quais a chance de se descobrir a senha seja melhor do que o acaso. O ataque de força bruta deve ser o último a ser utilizado, apenas quando todos os outros tiverem falhado.

Um dos ataques que pode obter sucesso é o ataque de dicionário. Nesse ataque, tenta-se todas as senhas de um determinado dicionário (lista de palavras), como por exemplo, um dicionário com todas as palavras da língua portuguesa. É um ataque rápido de ser realizado. Nesse tipo de ataque, a criatividade é o limite. Pode-se tentar dicionários temáticos, palavras que sejam da lista de interesses do alvo e até mesmo buscar dicionários de dados recuperando todas as palavras de outros dispositivos de informática do alvo que não estejam criptografados. Não obtendo-se sucesso, pode-se tentar a abordagem híbrida. Nesse ataque, cada palavra do dicionário sofrerá determinada variação, como por exemplo, colocar o primeiro caractere em maiúsculo, adicionar números ao final de cada palavra etc. Novamente, a criatividade é o limite. Vale ressaltar que, quanto mais variedade for adicionada, maior será o tempo necessário para completar o ataque.

2.7.3. Esteganografia e Esteganálise

Esteganografia, ou escrita oculta, pode ser usada como uma técnica antiforense. Segundo Velho et al (2016), esteganografia é o estudo de técnicas para ocultar a existência de uma mensagem dentro de outra, uma forma de segurança por obscurantismo. Ainda segundo esses autores, a informação a ser escondida é inserida em um arquivo hospedeiro, que precisará ser capaz de sofrer pequenas alterações em seus bytes e, ainda assim, reter suas principais características.

A esteganálise tem como objetivo descobrir e revelar mensagens ocultas por técnicas esteganográficas. Sua base é a análise estatística, buscando padrões de ocorrência do uso de técnicas de esteganografia nas mídias suspeitas.

2.8. Perícias Em Dispositivos Móveis

A popularização dos dispositivos móveis tem os tornados fonte de dados imprescindível na busca de evidências digitais. Diferente da perícia em computadores, na qual é possível retirar o disco rígido e fazer uma cópia bit a bit de seu conteúdo em ambiente controlado, as extrações de dados em dispositivos móveis devem ser feitas usando as próprias interfaces desses dispositivos. Essa característica impõe alguns desafios técnicos. Muitas vezes, mecanismos de segurança desses dispositivos têm que ser ultrapassados para se chegar à informação que se quer extrair, tornando esse tipo de perícia mais invasiva e os riscos de comprometimento das informações e do próprio dispositivo, maiores.

Cada tipo de dispositivo móvel tem suas particularidades, sendo que o mercado atual é dominado por dispositivos com sistema operacional Android e dispositivos da Apple. Além dos aspectos gerais, válidos para qualquer fabricante, o restante da seção focará em dispositivos com o sistema Android.

Em relação à tecnologia disponível de extração de dados de celulares e afins, temos um cenário parecido com o já apresentado anteriormente, ou seja, existem soluções comerciais e soluções livre e/ou de código aberto. As soluções comerciais oferecem uma interface intuitiva para a extração dos dados e constantes atualizações conforme a evolução dos sistemas operacionais e aplicativos dos dispositivos móveis. Entre os principais produtos, podemos citar Microsystemation XRY e Cellebrite UFED, ilustrados nas figuras 21 e 22, respectivamente.

Figura 21: XRY *Fonte da foto: http://aresources.pt/*

Figura 22 Cellebrite UFED *Fonte da foto: http://www.tecmundo.com.br/*

Quando falamos em extração de dados de dispositivos móveis temos, basicamente, seis tipos prenominantes: extração manual, extração lógica, extração física, JTAG, chip-off e micro read. As três últimas necessitam de conhecimento e manipulação do hardware e fogem do escopo deste trabalho.

Segundo apresentado por Velho et al (2016) e Tamma e Tindall (2015), cada tipo de extração apresenta a sua particularidade. A extração manual é a mais simples de todas e consiste em acessar o dispositivo manualmente e transcrever ou fotografar o conteúdo visível pelo próprio dispositivo. É um tipo de extração demorada, não recupera informações apagadas, devendo ser feita apenas em último caso, quando nenhum outro tipo de extração funcionou. Na extração lógica, o dispositivo móvel é conectado ao computador por meio de cabo USB ou rede sem fio de curto alcance (*bluetooth*, por exemplo) e o software de extração usa as APIs do sistema operacional para extrair os arquivos. Esse tipo extração não recuperará arquivos apagados. Uma forma especial de extração lógica é a extração sistema de arquivos, que usará um conjunto de APIs do sistema de arquivos. Essa extração poderá recuperar dados apagados que estejam em arquivos tipo banco de dados, como por exemplo, arquivos do banco de dados SQLite. Na extração física, será feita uma cópia bit a bit da memória interna do dispositivo. É o tipo que permite a maior recuperação de dados. Pode ser feita por meio de um *bootloader* específico, copiado para o dispositivo móvel ou por meio da instalação de um programa no dispositivo que copiará toda a memória. Essa segunda opção é mais invasiva e necessitará de privilégios de super-usuário (*root*).

2.8.1. Perícias em Dispositivos Android

Nessa seção serão apresentadas algumas características do sistema operacional Android, os principais locais em que os dados ficam armazenados e formas de se extrair dados dos dispositivos usando os próprios recursos do Android ou usando software livre.

O Android foi desenvolvido baseado no *kernel* do Linux, com modificações para adequar o sistema ao ambiente de dispositivos móveis. Uma vez obtido acesso ao dispositivo, muitos comandos do Linux podem ser executados no Android, inclusive aqueles para navegação pelos diretórios e para realizar cópias de dados.

Um dos aspectos importantes que o especialista deve ter conhecimento em relação a esse tipo de perícia é a segurança implementada pelo Android. A segurança em dispositivos móveis é um recurso importante aos usuários, mas pode dificultar o trabalho do perito. Segundo Tamma e Tindall (2015), recursos importantes de segurança são assegurados utilizando-se os recursos de segurança do *kernel* do Linux. Por essa implementação, é garantido um modelo de acesso baseado em permissões, isolamento de processos e um mecanismo de segurança para chamadas interprocessos (IPC).

No modelo de segurança do Android, cada aplicação é executada com seu próprio UID, não permitindo com isso que, vide regra, uma aplicação acesse os dados armazenados no sistema de arquivos de outra. As aplicações também são executadas em uma *sandbox*, rodando em uma máquina virtual Java modificada, chamada Dalvik.

Além disso, cada aplicação deve ser assinada por uma chave criptográfica do fornecedor da aplicação e deve declarar em um arquivo de manifesto quais são os recursos do sistema que ela necessita.

2.8.1.1. Processo de *Boot* do Android

Entender o processo de *boot* do Android ajudará na compreensão de como algumas técnicas de extração de dados funcionam.

O processo de *boot* é dividido em seis fases: código de *boot* ROM, o *boot loader*, o *kernel*, processo init, o zygote e Dalvik e o serviço de sistema. A primeira fase é o *boot* ROM, que inicializa o hardware e procura por uma mídia de *boot*. A próxima fase é o *boot loader*, responsável por carregar o sistema operacional. É o *boot loader* que também permite inicializar o dispositivo em outros modos, como o modo *fastboot*, *recovery* etc. Numa inicialização padrão, o *boot loader* carrega o *kernel* na memória e passa o controle para ele. Uma vez carregado, o kernel monta o sistema de arquivos de *root* (*rootfs*) e executa o primeiro processo do sistema, o init. O processo init executará os comandos do script init.rc. O processo zygote, responsável por criar as máquinas virtuais Java Dalvik será iniciado. Por fim, o serviço do sistema inicializá diversos serviços e o dispositivo Android estará operacional para o usuário.

2.8.1.2. Acessando Dispositivos Android por meio do ADB

O ADB (*Android Debug Bridge*) é um mecanismo de depuração de aplicações Android que permite que o dispositivo seja acessado de um computador. Pode ser uma das formas de se obter os dados do dispositivo.

Para usar esse recurso, o ADB deve estar instalado no computador do perito, por meio da instalação do pacote de desenvolvimento Android SDK (https://developer.android.com/studio/index.html). No dispositivo móvel, o software necessário para conexão já existe, mas por padrão está desabilitado. Para habilitá-lo, deve-se ir em “configurações”, selecionar “opções do desenvolvimento” e “Depuração de USB”.

Realizadas essas configurações, o dispositivo pode ser conectado por meio do ADB. A estrutura dos diretórios do sistema de arquivos do Android assemelha-se ao do Linux. A maioria dos dados de interesse forense estarão armazenadas no diretório /data/data.

Pelo método do ADB existem basicamente duas formas de se extrair os dados do Android. Usando o comando ADB push, que copiará os arquivos ativos (não apagados) para o computador do especialista ou uma forma mais interessante, que é fazer uma cópia física dos dados.

Para realizar a cópia física, deve-se usar algum programa que possibilite esse tipo de cópia e enviar o resultado para o computador do perito. Uma opção é usar os programas dd e netcat, o primeiro para copiar os dados, o segundo para transmiti-los via uma conexão entre o dispositivo móvel e o computador. Esses dois programas não estão

presentes na configuração padrão do Android, mas podem ser encontrados na Internet e baixados para o dispositivo por meio dos comandos ADB.

Os pontos fracos dessa abordagem de extração de dados são a necessidade de habilitar o modo de depuração de USB no próprio dispositivo e a necessidade de se ter acesso de super-usuário (*root*). Caso o dispositivo esteja protegido por senha, não será possível habilitar a depuração USB. Dessa forma, um outro meio de extração de dados é necessário.

2.8.1.3. Acessando dispositivo Android por meio de substituição de partição de recuperação (*recovery partition*)

Os dispositivos Android podem ser inicializados em outros modos além do modo padrão. Um modo de inicialização importante é o modo *recovery*. A inicialização em um modo diverso do padrão pode ser realizada pressionando uma combinação de teclas, que varia de fabricante para fabricante, durante o processo de *boot*.

O software de fábrica que está instalado na partição *recovery*, também conhecido como *stock recovery*, permite que o dispositivo seja restaurado para a configuração de fábrica, no qual todos os dados do usuário serão apagados, fazer atualizações do sistema, entre outros. Não há nenhuma opção na *stock recovery* que seja de interesse para a área forense.

Porém, a *stock recovery* pode ser substituída por um programa alternativo, geralmente chamado de *custom recovery*. Por meio da *custom recovery* é possível fazer o *backup* dos dados, ganhar acesso de *root* e outras atividades de interesse pericial. As principais *custom recoveries* encontradas na Internet atualmente são a CWM (ClockWorkMod) e TWRP (TeamWin, https://twrp.me/). Cada modelo de dispositivo móvel necessitará de uma *custom recovery* apropriada. Uma fonte de informação sobre as *custom recoveries* é http://forum.xda-developers.com/.

Um comparativo das funcionalidades disponíveis nesse dois tipos de *recoveries* é apresentado nas figuras 23 e 24.

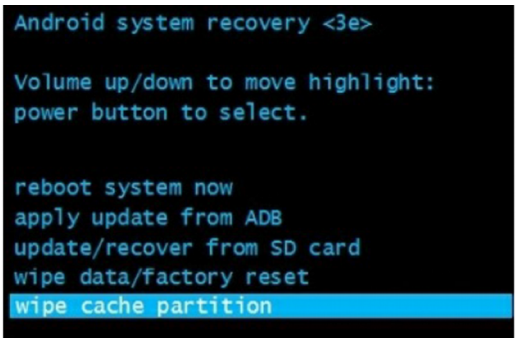

Figura 24: Stock recovery

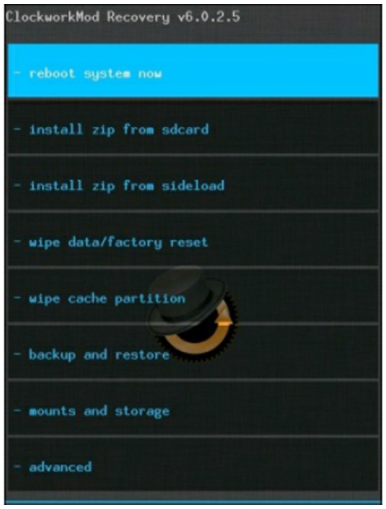

Figura 23: Custom recovery

Referências

Brooks, Charles L. - CHFI Computer Hacking Forensic Investigator Certification All-in-One Exam Guide; 2014

Carrier, Brian – File System Analysis – USA, 2006

Carvey, Harlan – Windows Forensic Analysis – USA: Syngress, 2009

Epifani, Mattia; Stirparo, Pasquale - Learning iOS Forensics – Packet Publishing: 2015

Gomes, Jeremias Moreira – A forense computacional e os discos de estado sólido – ICoFCS, 2012

Fagundes, Leonardo L.; Neukamp, Paulo A.; da Silva, Pamela C. – Ensino da Forense Digital Baseado em Ferramentas Open Source – ICoFCS, 2011

Machado, Margarida Helena Serejo. A Regulamentação da Cadeia de Custódia na Ação Penal: Uma necessidade Premente. Corpo Delito, n.1, p. 18-23, Brasília, 2009.

Marimoto, Carlos Eduardo - Hardware II – O Guia Definitivo – Sul Editores - Porto Alegre, 2010

Merola, Antonio - Data Carving Concepts - SANS Institute – November, 2008

Silva, Gilson Marques; Lorens, Evandro Mário – Extração e Análise de Dados em Memória na Perícia Forense Computacional – ICoFCS, 2009

Stallings, Willian – Criptografia e segurança de redes – 4ª Edição – São Paulo, 2008

Tamma , Rohit; Tindall, Donnie – Learning Android Forensics – Packet Publishing: 2015

Velho, Jesus Antonio; et al – Tratado de Computação Forense – Millenium Editora; São Paulo, 2016

Yiannis, Chrysanthou - Modern Password Cracking: A hands-on approach to creating an optimised and versatile attack. Technical Report RHUL–MA–2013– 7; 2013