FACULDADE DE TECNOLOGIA IBRATEC DE JOÃO PESSOA
CURSO DE PÓS-GRADUAÇÃO EM SEGURANÇA DA INFORMAÇÃO

JADILSON ALVES DE PAIVA

PRÁTICAS ANTI-FORENSE: UM ESTUDO DE SEUS IMPACTOS NA FORENSE COMPUTACIONAL

João Pessoa – PB
2009

JADILSON ALVES DE PAIVA

PRÁTICAS ANTI-FORENSE: UM ESTUDO DE SEUS IMPACTOS NA FORENSE COMPUTACIONAL

Monografia apresentada à Coordenação de Pós-Graduação da Faculdade de Tecnologia IBRATEC de João Pessoa, sob a orientação do Prof. Msc. Marcio Luiz Machado Nogueira, como exigência à obtenção do título de especialista em Segurança da Informação.

João Pessoa – PB

Fevereiro de 2009

FICHA CATALOGRÁFICA – Verso da Folha de Rosto
Deve ser elaborada por profissionais da área da Biblioteconomia

JADILSON ALVES DE PAIVA

TÉCNICAS ANTI-FORENSE: UM ESTUDO DE SEUS IMPACTOS NA ANÁLISE FORENSE COMPUTACIONAL

Aprovado em de de 2009

BANCA EXAMINADORA

_____

Prof. MsC. Márcio Luiz Machado Nogueira
Orientador

DEDICATÓRIA

Dedico este trabalho a Adriana, por me incentivar desde o início ao fim desta especialização e a faculdade IBRATEC João Pessoa, pelas oportunidades oferecidas.

AGRADECIMENTOS

Primeiramente aos meus pais.

Ao Professor Márcio que sugeriu o tema e apostou no meu trabalho.

Aos amigos do curso pelos grupos de estudos nas madrugadas e fins de semanas.

Aos meus alunos e ex-alunos que me motivaram para a realização desta especialização.

A todos que, direta ou indiretamente, contribuíram para a realização deste trabalho, consigno a mais elevada gratidão.

PAIVA, Jadilson Alves. **PRÁTICAS ANTI-FORENSE: UM ESTUDO DE SEUS IMPACTOS NA FORENSE COMPUTACIONAL**: 2009 - 102p. Monografia (Especialização em Segurança da Informação) – Faculdade de Tecnologia IBRATEC de João Pessoa.

RESUMO

O crescimento desordenado do uso de computadores sob procedimentos de segurança pouco disseminados e com a ausência de regulamentação das leis do uso do computador, promove a criatividade para a prática de crimes cibernéticos. Diante do aspecto da crescente criminalidade cibernética, surgem os peritos forenses computacionais que tem como função, levantar todo o histórico de uso dos computadores a procura de evidencias que levam ao possível culpado pelos crimes. Em contra partida, os criminosos estudam a forma de como os peritos forenses trabalham e desenvolvem as técnicas anti-forense para dificultar ou inviabilizar o trabalho do perito, além de causar danos sem precedentes em empresas e/ou pessoas. Este trabalho apresenta de um modo geral as aplicações práticas das técnicas anti-forense mostrando os impactos de cada técnica em um cenário de análise forense, como forma de alerta emergencial para estudos futuros de como se precaver destes tipos de ataques.

Palavras-chave: Anti-forense, slack spaces, NTFS, ADS.

PAIVA, Jadilson Alves. **PRÁTICAS ANTI-FORENSE: UM ESTUDO DE SEUS IMPACTOS NA FORENSE COMPUTACIONAL**: 2009 - 102p. Monografia (Especialização em Segurança da Informação) – Faculdade de Tecnologia IBRATEC de João Pessoa.

ABSTRACT

The disorderly growth of the use of computers in safety procedures with little spread and the lack of laws regulating the use of computer, promotes creativity to the practice of cyber crimes. Considering the aspect of increasing cyber crime, computer forensic experts arise which has as its objective to raise the entire history of use of computers to search for evidence leading to the possible guilt for the crimes. On the departure, the criminals learn how to work as forensic experts and developing anti-forensic techniques to hinder or impede the work of the expert, and unprecedented damage to businesses and / or persons. This paper presents a general practical applications of anti-forensic techniques showing the impacts of each technique in a scenario of forensic analysis, as an emergency alert for future studies of how to prevent these types of attacks.

Keywords: Anti-forense, slack spaces, NTFS, ADS.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

SUMÁRIO

INTRODUÇÃO

Com o advento das redes de computadores e em seguida a internet, o computador passou a fazer parte da vida de milhões de pessoas espalhadas pelo mundo, tornando-as dependente na troca de informações, realização de compras, transações bancárias, redes de relacionamento e dentre várias outras inúmeras atividades que alavancam a tecnologia e a produtividade (PIMENTA, 2007). Por outro lado, os crimes cibernéticos cometidos através das redes de computadores totalizam um faturamento de mais de 105 bilhões de dólares, que são motivados pela falta de regulamentação das leis que punem os crimes cometidos no meio computacional (WIRELESSBR, 2009).

E segundo Lima (2008), “os crimes continuam os mesmos, as armas que mudaram com a tecnologia.” Logo, o que antes era uma ferramenta para solução de problemas, hoje, o computador em mãos erradas se torna uma arma mutável de acordo com o avanço tecnológico. Levando a um novo mundo de pessoas, tecnologias e crimes cibernéticos que originaram a ciência forense computacional (NG, 2007).

A ciência forense computacional foi criada com o objetivo de suprir as necessidades das instituições legais no que se refere à manipulação das novas formas de evidências eletrônicas, sendo a ciência que estuda a aquisição, preservação, recuperação e análise de dados que estão em formato eletrônico e armazenados em algum tipo de mídia computacional (NOBLETT; POLLIT; PRESLEY, 2000).

Diante desta ciência que aos poucos padroniza suas ações de coleta e análise de dados, os autores dos crimes cibernéticos melhoram as suas técnicas dentro do perfil normalizado pelos peritos forense. De forma a esconder dados, informações e evidências que despistam ou inviabilizam o trabalho do perito (SILVA, 2003). Esta nova modalidade de crimes cibernéticos é denominada como técnicas anti-forense (SILVA, 2003).

Conforme Harris (2006), a anti-forense ainda é um ciência inexplorada que cresce aceleradamente e não possui nenhuma definição unificada, além de utilizar e explorar as ferramentas forenses. Percebemos isto pelo fato da literatura resumida, e da dificuldade de encontrar algo concreto e principalmente quanto a aplicações práticas. E entendemos que isto gera um grande problema, pois nem mesmo

sabemos exatamente quais os seus impactos no mundo computacional e nem tão pouco na forense computacional. Portanto, é de grande importância entender quais os problemas que as técnicas anti-forense podem trazer para o mundo computacional e principalmente com relação à perícia.

Diante do exposto, surgiu à necessidade de verificar em laboratório a aplicação e eficiência das técnicas anti-forense, para que fossem demonstradas neste trabalho, e como pergunta de pesquisa, saber exatamente quais os meios utilizados para a aplicação das técnicas anti-forense.

Este trabalho tem como objetivo geral, aplicar as técnicas anti-forense e avaliar os seus impactos na análise forense, mostrando quais os meios utilizados para a aplicação destas técnicas.

Os objetivos específicos deste trabalho serão:

- Definir os conceitos técnicos relevantes para o entendimento das aplicações das técnicas anti-forense, uma vez que estes não são encontrados com facilidade na literatura.

- Aplicar as técnicas anti-forense através do uso de softwares específicos e alternativos em uma máquina local desktop com sistema de arquivos NTFS.

- Avaliar os problemas caudados pelas técnicas anti-forense através de análises forenses.

Iniciamos o trabalho com a fundamentação teórica mostrando os históricos e evoluções da forense e anti-forense computacional, evidenciando as relações entre essas duas ciências além de uma formalização de conceitos da anti-forense.

No capítulo 2 mostraremos os conceitos técnicos que abrangem o funcionamento lógico do disco, desde a inicialização até a gerência dos arquivos em disco na qual são a base para o entendimento da aplicação das técnicas anti-forense.

No capítulo 3 descreveremos toda a metodologia empregada para a elaboração dos experimentos em um laboratório.

No capítulo 4 demonstraremos as técnicas anti-forense através dos experimentos de destruição, ocultação, eliminação e falsificação de evidências com as suas respectivas problemáticas no cenário de análise forense.

Finalizaremos o trabalho com as discussões sobre as características de cada técnica anti-forense aplicadas.

CAPÍTULO 1 - FUNDAMENTAÇÃO TEÓRICA

Este capítulo trata do histórico e evolução das ciências forense e anti-forense mostrando a relação entre elas.

1.1 Forense computacional

1.1.1 Histórico

O mundo está cada vez mais dependente com relação às tecnologias de sistemas digitais e de redes de computadores que crescem aceleradamente para atender as evoluções tecnológicas em massa. (DIGITAL SIGNAGE, 2009). Muito dessas tecnologias se assemelham as outras mudanças culturais que avançaram para modificar as nossas vidas, e a disponibilidade dessa tecnologia digital conduz inevitavelmente para a utilização indevida por prática ilegal do uso dos recursos digitais. E seja qual for às circunstâncias do crime – fraude, pedofilia, espionagem industrial, pirataria ou de corrupção, a tecnologia digital está envolvida (HONORATO, CARDOSO, 2008);

Charles Babbage, Alan Turing e Von Neumann não iriam compreender as conseqüências de suas ações no passado com a criação de seu computador, que hoje move o mundo moderno e também é uma arma para prática de crimes cibernéticos (VOGON, 2008).

Antigamente, qualquer informação era registrada em papel e exemplares eram cuidadosamente escritos por uma equipe de escriturários, e qualquer alteração era fácil de detectar quando comparado com o original. Depois da máquina de escrever e, em seguida, a fotocopiadora, economizando tempo e o trabalho de produção, a adulteração se tornava cada vez mais difícil de ser detectada. Daí, todo um ramo da polícia científica surgiu a partir deste problema para lidar com a questão de provar se um documento não era autêntico (VOGON, 2008).

Com a chegada de dispositivos de armazenamento de massa, os problemas foram aumentando, pois a disseminação de cópias era mais fácil e as informações contidas nestes dispositivos já não eram armazenadas com palavras legíveis, e sim codificadas com uma série de pulsos magnéticos gravados em fita e disco. (VOGON,

2008). Para que estas informações fossem lidas em forma legível, a primeira resposta foi criar uma cópia de todos os dados do disco e fita suspeitos tornando ser bem confiável, porém, com um custo elevado, devido à escassez de microcomputadores e mainframes (VOGON, 2008). No momento, a maneira mais óbvia de reproduzir os dados em um formato legível seria a impressão, mas depois caia no mesmo problema. Pois uma fita ou disco suspeito teria de serem comparados com as impressões originais, e para isto, essas mídias deveriam ser traduzidas da codificação magnética (VOGON, 2008). Para tentar resolver este problema, a única maneira foi de recorrer aos serviços de uma multiplicidade de peritos para recriar o sistema original e reproduzir as impressões, só que a um preço bastante alto (VOGON, 2008).

Felizmente para os investigadores daquela época, o acesso a computadores era limitado às grandes empresas e as incidências de dados suspeitos durante as investigações eram apenas restritas somente as estas empresas, e o volume de dados a serem periciados eram poucos (VOGON, 2008). Mas, com o advento do IBM/PC, as suas muitas variantes introduziram novos problemas no mundo da perícia, como a produção de um grande volume de dados, bem como a capacidade de alterar, ocultar e excluir dados (VOGON, 2008).

A informática foi disponibilizada para as todas as classes sociais que, naturalmente várias pessoas aproveitaram da tamanha complexidade do sistema para a prática dos crimes. E com essa nova gama de variáveis e complexidade, seria necessário um conhecimento bastante especializado e geral para se investigar esse novo cenário computacional. Com isso, surgiu à nova disciplina que podemos chamar de a arte da forense computacional.

A forense computacional é definida por Pires (2003, p. 2) como:

> Um conjunto de técnicas, cientificamente comprovadas, utilizadas para coletar, reunir, identificar, examinar, correlacionar e analisar e documentar evidências digitais processadas, armazenadas ou transmitidas por computadores.

Através de um conjunto metódico de procedimentos para manipulação e interpretação de evidências, é possível determinar um autor de um crime ou fraude no ambiente computacional.

Segundo Omar (2006, p. 3), “As primeiras fraudes documentadas são contra a contabilidade bancária, que eram cometidas por funcionários responsáveis pela área de informática da instituição, além de fraudes contra governos e usuários”. Neste tempo, o único método disponível para o perito forense iniciar uma investigação desta natureza, era obter um backup dos arquivos do disco suspeito e copiá-los para outro disco, para que então cada arquivo fosse analisado minuciosamente. Só que a simples cópia dos arquivos adultera os seus atributos de tempo (*mactimes*), que são uma das variáveis mais importantes para a perícia, pois contém os tempos de acesso, criação e modificação de cada arquivo (FARMER; VENEMA 2006).

Diante dos problemas com os *mactimes*, a perícia precisava desenvolver ferramentas funcionais e automatizadas para ajudar no processo, além de poder acessar arquivos apagados ou parcialmente sobrescritos. Com isso, o próximo passo foi examinar a mídia original através de um editor de disco hexadecimal, que dá acesso a cada setor mostrando a conversão dos binários em caracteres *American Standard Code for Information Interchange* (ASCII). Porém, a análise a este nível é complexa e cansativa, e às vezes sem sucesso (VOGON, 2008).

Como a tecnologia avançava mais problemas surgiam e a perícia forense tinha que refinar suas análises e definir certos padrões, para que pudessem ser adotados por outros especialistas, além de serem exigidos pelas legislações jurídicas com relação à integridade e autenticidade dos dados (VOGON, 2008).

Com as exigências da justiça com relação à integridade e autenticidade dos dados perante a definição de padrões e métodos, os peritos viram que não poderiam andar sozinhos neste avanço, uma vez que seria o suporte técnico da justiça para validar um crime cibernético. Logo, tinham que se apoiar nas legislações vigentes para que suas análises fossem aceitas no mundo jurídico, entretanto, um princípio surgia (VOGON, 2008).

"Nenhuma medida tomada por qualquer pessoa que realize uma investigação sobre um computador deve modificar os dados realizados no computador ou outros meios que possam posteriormente ser utilizadas como prova" (VOGON 2008, tradução nossa).

A partir deste princípio, o desafio da ciência forense era de preservar ao máximo a mídia original e assegurar que as evidências analisadas não poderiam ser alteradas ou contaminadas pelo processo de investigação. Mas com a adaptação

das tecnologias e juntamente com o desenvolvimento de softwares adequados, um disco original pôde ser copiado fielmente bit a bit sem que houvesse risco de corrompimento da mídia original e nem das evidências obtidas das imagens geradas. Com isso, a forense computacional começou a ter credibilidade e resultados nos quais promoveram esta ciência para o mundo, além de grandes mentores.

Segundo Jones (2007), Michael Anderson é considerado o pai da forense computacional por ser o pioneiro neste campo desde 1987, sendo um dos responsáveis diretos no desenvolvimento de procedimentos da prática forense para a formação e certificação de metodologias. Além de projetos e avaliações de softwares que se tornaram padrões utilizados pelos especialistas forenses de todo o mundo. Seus programas de pesquisas nesta área foram distribuídos mundialmente pelo *Internal Revenue Service* e ainda é um sustentáculo em algumas áreas de aplicação das leis da forense computacional (NTI, 2008).

Ele também foi o principal fundador do *New Technologies, Inc* (NTI), e treinou mais de 2500 policiais militares e especialistas forenses computacionais que compreendem grandes corporações como o *Federal Bureau of Investigation* (FBI), *National Security Agency,* serviço secreto, departamento de defesa, agência nacional de segurança e muitas outras agências governamentais (NTI, 2008).

Michael Anderson possui 25 anos de experiência como agente especial / especialista em computação forense, possuindo inúmeras certificações e conhecido mundialmente pelas suas contribuições na área da forense computacional. Atualmente é aposentado, mas ainda presta serviços de consultoria para juízes e procuradores (NTI, 2008).

Diante das contribuições do Michael Anderson e da disseminação de suas pesquisas entre os seus especialistas treinados, a perícia forense foi se espalhando pelo mundo e tornando-se cada vez mais importante pelo fato da demanda crescente de crimes digitais. Conseqüentemente, esta crescente demanda fez com que surgissem outros pesquisadores na área como cientistas, empresas e órgãos governamentais, que, no entanto, precisavam normalizar padrões e métodos universais no tratamento das evidências forenses.

1.1.2 Padrões e métodos

Com relação às outras ciências forenses já existentes há bastante tempo como a balística, criminalística, antropologia, química, odontologia, patologia, toxicologia e genética, a forense computacional é uma área relativamente nova e atualmente tornou-se uma prática investigativa importante tanto para empresas quanto para a polícia (FREITAS, 2006).

Apesar do atual estágio das pesquisas no campo da forense computacional, ainda existe muita carência de metodologias para o manuseio deste tipo de evidência com relação às outras ciências forenses (OLIVEIRA, 2002).

Segundo Noblett; Pollit; Presley (2000), uma pesquisa realizada pelo serviço secreto norte americano no ano de 1995, indicou que 48% das agências tinham laboratórios de forense computacional, nas quais, 68% das evidências encontradas foram encaminhadas aos peritos desses laboratórios. E no mesmo documento de pesquisa, 70% dessas agências fizeram seu trabalho pericial sem um manual de padronização.

Políticas de manipulação de uma evidência computacional devem ser estabelecidas, para que sejam desenvolvidos protocolos e procedimentos para uma formalização de padronizações para outras comunidades científicas, refletindo em resultados válidos. Contudo, a forense computacional é diferente das outras disciplinas forenses, uma vez que não se pode aplicar exatamente o mesmo método a cada caso específico, devido à diversidade de sistemas operacionais com estruturas lógicas diferentes e dentre várias tecnologias e aplicações envolvidas (NOBLETT; POLLIT; PRESLEY, 2000). Bem diferente de uma análise do Acido Desoxirribonucléico (DNA), em um sangue encontrado na cena de um crime, em que existem procedimentos e protocolos estabelecidos para este caso específico (NOBLETT; POLLIT; PRESLEY, 2000).

Segundo Guimarães et al (2002), as agências legais de vários países foram obrigadas a definirem métodos comuns para o tratamento de evidências eletrônicas, mas cada nação conta com sua legislação local e não seria possível a definição de uma norma universal. Contudo, as padronizações se aplicam apenas entre a troca de informações entre os países (GUIMARÃES et al, 2002).

Atualmente já existem padrões desenvolvidos pelo *Scientific Working Group on Digital Evidence* (SWGDE), que seguem o princípio de que todas as

organizações forenses devem manter um alto nível de confiança e exatidão na manipulação das evidências (GUIMARÃES et al, 2002). Este alto nível é conseguido através da elaboração das *Standard Operating Procedures* (SOPs), que devem conter os procedimentos para todo tipo de análise conhecida, assim como utilização de técnicas, equipamentos e materiais largamente aceitáveis na comunidade científica (SWGDE, 2000 apud GUIMARÃES et al, 2002).

Segundo Oliveira (2002, p.79):

> O desenvolvimento de documentos como SOPs dependem de técnicas específicas para cada tipo de situação e sistema operacional (SO). No entanto, alguns procedimentos relacionados à manipulação de sistemas de arquivos são gerais e não dependem do SO que está sendo analisado.

Podemos citar como SOPs gerais:

• Análise sobre cópias: Fazer cópias bit a bit (imagem) e não cópias comuns do disco original, para que todos os setores do disco sejam copiados. O procedimento de análise deve ser realizado nas cópias, para não contaminar o disco original.

• Assinaturas digitais: Utilizar assinaturas digitais para assegurar a confiabilidade das imagens e posteriormente os dados analisados, garantindo que são exatamente iguais aos do disco original.

• Sem permissão de escrita e execução: Examinar as cópias (imagens) com softwares que protejam o disco contra gravação e execução, evitando a alteração dos *mactimes*.

Após os padrões de procedimentos operacionais para o tratamento das evidências de uma mídia suspeita, é necessário que as imagens geradas sejam encaminhadas a um processo de investigação.

1.1.3 Processo de investigação

A partir das imagens geradas dentro das SOPs, as fases de condução de um processo de investigação em forense computacional segundo Pereira et al, (2007) pode-se dividir em quatro etapas, identificadas como: coleta dos dados, exame dos dados, análise das informações e interpretação dos resultados:

• Coleta de dados: Coletar o máximo de informações dentro dos mais variados tipos de dispositivos e aplicações, sem danificá-las. É importante que cada passo realizado na coleta seja documentado, assim como a execução das cópias dos originais devidamente assinados digitalmente.

• Exame de dados: Para cada tipo de dado coletado, existe uma ferramenta e técnica adequada que garanta a integridade dos dados relevantes como evidências.

• Análise das informações: Esta etapa analisa as evidências filtradas da etapa anterior, com o intuito de reconstruir a cena do crime.

• Interpretação dos resultados: Esta é a ultima etapa do processo de investigação, na qual é responsável por toda a documentação e organização das informações obtidas para a elaboração do laudo pericial final.

De acordo com Freitas, (2006), um laudo pericial é um relatório técnico que aponta os resultados obtidos a partir das provas, no qual é avaliado pela justiça. No entanto, todos os procedimentos da forense computacional devem estar baseados nas legislações locais.

Segundo Guimarães et al (2002, p.3):

> No Brasil não existem normas específicas para a forense computacional, no entanto, as normas gerais existentes abrangem todos os tipos de perícia que estão ditadas no Código de Processo Penal, podendo ser aplicadas na perícia computacional salvo algumas peculiaridades.

Podemos citar dois artigos importantes do código de processo penal, em que a forense computacional se baseia:

• Art. 170. Nas perícias de laboratório, os peritos guardarão material suficiente para a eventualidade de nova perícia. Sempre que conveniente, os laudos serão ilustrados com provas fotográficas, ou microfotográficas, desenhos ou esquemas.

• Art. 171. Nos crimes cometidos com destruição ou rompimento de obstáculo a subtração da coisa, ou por meio de escalada, os peritos, além de descrever os

vestígios, indicarão com que instrumentos, por que meios e em que época presumem ter sido o fato praticado.

Mesmo seguindo todos os procedimentos, o criminoso pode sair ileso devido à coleta e manipulação de dados de forma incorreta (PEREIRA, 2007). No entanto, é importante conhecer quais as ferramentas forenses que tem credibilidade com relação ao tratamento das evidências.

1.1.4 Ferramentas de perícia

Diante da evolução da forense computacional, as ferramentas de perícia sempre tiveram o intuito de oferecer maiores subsídios e credibilidade no tratamento das evidências. Atualmente, existem as mais diversas soluções, tanto proprietárias como abertas, nas quais, citaremos apenas as de código aberto por questões legais de direito de patentes e propriedade intelectual.

A seguir listaremos algumas ferramentas, nas quais são utilizadas neste trabalho.

• Helix: Segundo a Efense (2008), o *Helix* é uma distribuição personalizada do Linux Unbuntu, podendo funcionar como *live* CD ou dentro do próprio sistema operacional a ser periciado através de uma forense *on-line*. Possui proteção contra gravação em disco e gera imagens com assinaturas digitais. O *Helix* é um *framework* com as mais diversas ferramentas de código aberto, como podemos ver na Figura 1.

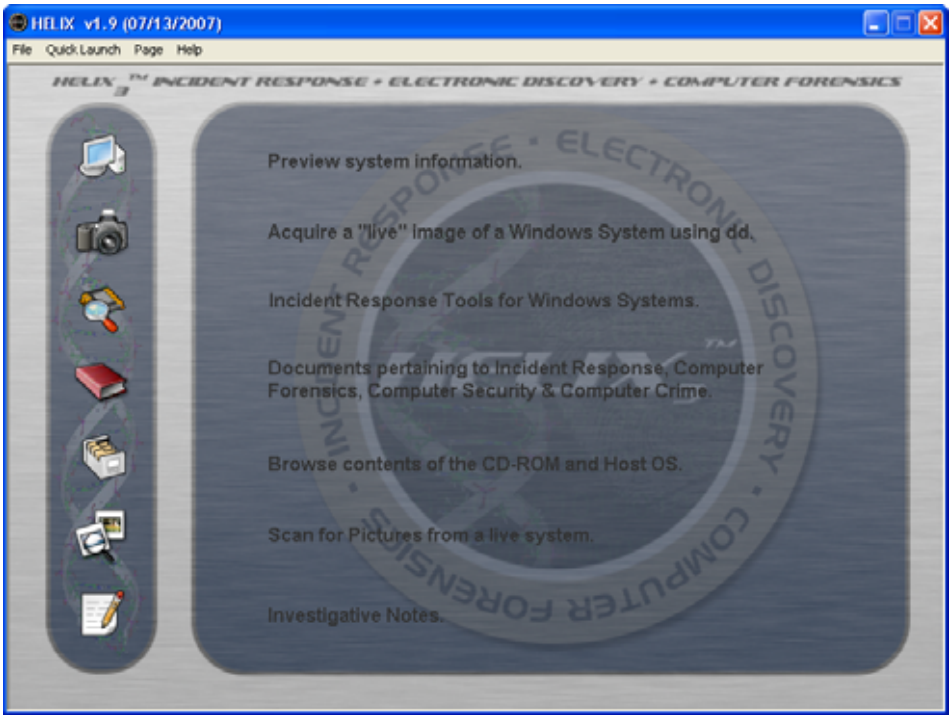


Figura 1 – Interface do Helix em forense on-line

Fonte: Ferramenta Helix

Na forense *on-line* com o *Helix*, podemos fazer uma análise das informações em qualquer mídia de armazenamento, bem como criar uma imagem dos dados para que seja posteriormente analisado através da forense *off-line* (sem o sistema operacional carregado). Na Figura 2, podemos ver que o *Helix* na forense *off-line* trás inúmeras ferramentas para coleta e análise de dados, e dentre outras que auxiliam na perícia.

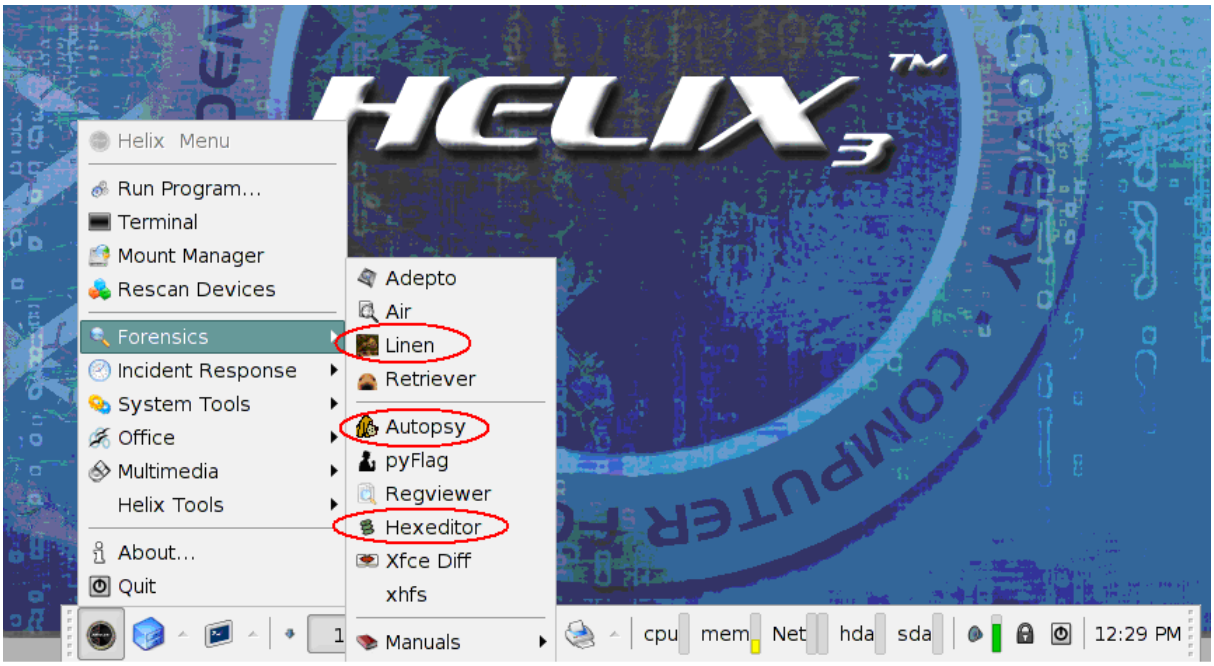


Figura 2 – Interface do Helix em forense off-line

Fonte: Ferramenta Helix

Além dos softwares, existem também *hardwares* específicos que auxiliam na perícia, com o intuito de melhorar as questões do tempo de aquisição, confiabilidade e análise que são bastante demorados em sistema convencionais.

Existem as mais diversas soluções em *hardwares*, desde pequenas maletas até sistemas modulares que se parecem até com um servidor. São sistemas de alto custo, mas que se fazem necessário em grandes corporações assim como outros ativos de redes.

A Figura 3 mostra um exemplo de um duplicador de imagens, que faz cópias fieis de discos garantindo a eficácia da prova além de gerar assinaturas digitais.

Figura 3 - Duplicador de imagens Forensics SF-5000
Fonte: http://www.logicube.com/media/images/product_3_d.jpg

A Figura 4 mostra um exemplo de um sistema robusto formado por um computador modular, que são mais específicos para a aquisição de grande quantidade de dados, como exemplo, uma matriz de discos que possuem capacidade de Terabytes obtida através de RAID, ou armazenamento distribuído através de um *cluster* de rede.

Figura 4 – Computador modular – Forensic Recovery of Evidence Device
Fonte: http://www.digitalintelligence.com/images/fredhw.jpg

Mesmo com as melhores práticas forenses e as mais avançadas ferramentas de aquisição e análise, é necessário que os peritos entendam as formas de como os crimes estão sendo executados, principalmente através das técnicas anti-forense para entender os seus efeitos na forense.

1.2 A origem da anti-forense

1.2.1 A motivação para o cibercrime

A partir da evolução do computador como uma ferramenta mais genérica, o mesmo propicia utilidades jamais pensadas, e como todo novo bem tecnológico possui duas faces, o mesmo pode ser usado tanto para o bem quanto para o mal. Podemos citar o exemplo do protocolo TCP, que no início foi projetado com o intuito de fazer apenas a comunicação entre computadores, mas, no entanto, certas pessoas conseguiram ver falha no protocolo. E diante disto, realizaram crimes cibernéticos como invasões ou ataques em massa para indisponibilizar servidores na internet. Mas que aos poucos, medidas de segurança foram acrescidas paralelamente ao protocolo tornando-o menos vulnerável aos seus problemas iniciais.

Assim como as tecnologias da informação, outras também sofrem com o seu uso indevido, como a engenharia, química e física, de forma que sempre haverá uma preocupação por parte dos cientistas criadores com o impacto de qualquer nova tecnologia em mãos alheias.

A gama de ferramentas disponíveis para a prática de crimes, anonimato, a falta de leis específicas para a punição, e o baixo custo para a prática do delito - uma vez que este pode ser realizado de um computador público, são os agentes motivadores para inclusão de criminosos de outros crimes paralelos, como roubos bancários, extorsões, pedofilia e etc. Com isso, o uso desordenado de computadores e a informatização mundial em massa sem uma infra-estrutura de segurança, trás, a cada dia, mais problemas sem precedentes. Refletindo em uma migração de crimes realizados de próprio punho para a forma cibernética, no qual se torna menos arriscado.

De posse dos dados obtidos pelo Centro de Atendimento a Incidentes de Segurança (CAIS) apresentados nas figuras 5 e 6 podemos entender a motivação do cibercrime através do aumento da quantidade de incidentes com o passar dos anos. Todavia, as estatísticas ainda estão abaixo da realidade, devido a muitos casos não serem reportados e mesmo que a quantidade de crimes tenha reduzido nos últimos anos, ainda é um número bastante elevado para uma pequena comunidade forense determinar os autores dos delitos.

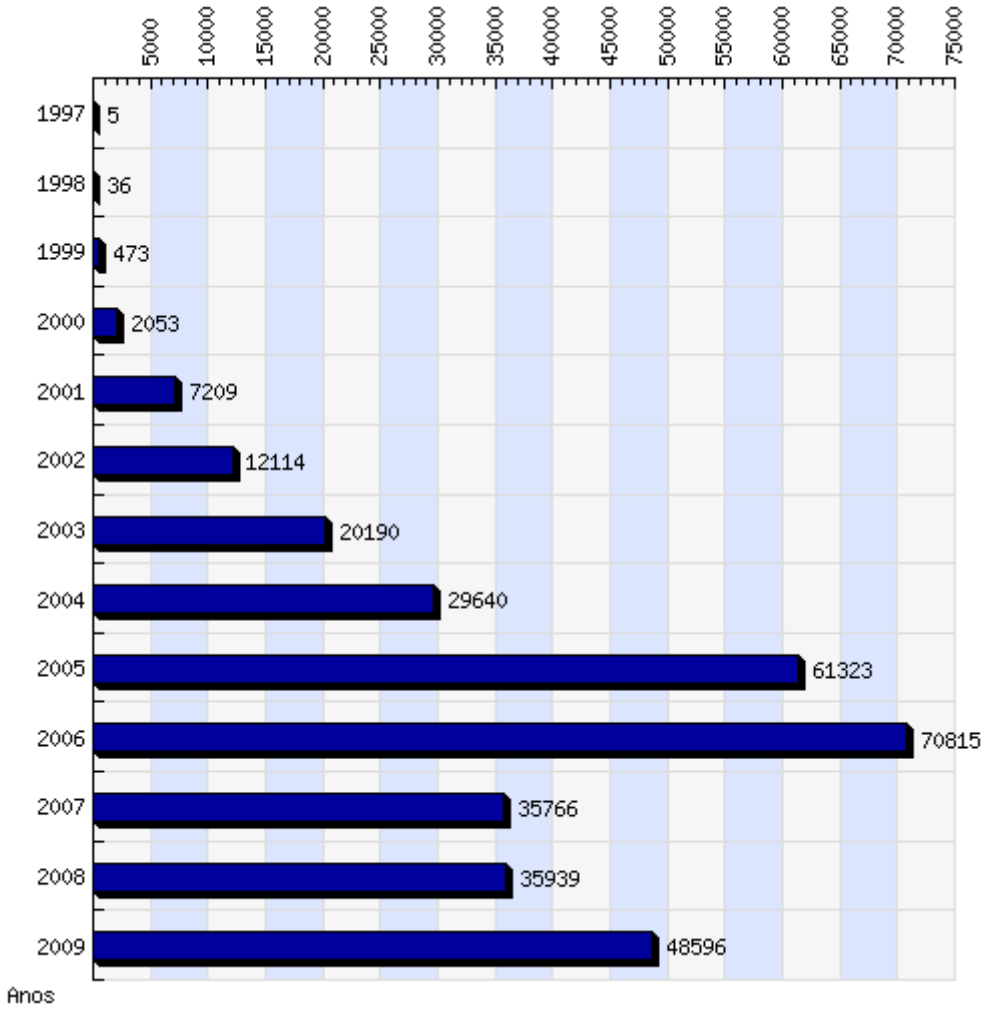


Figura 5 – Número de incidentes por ano registrado pelo CAIS

Fonte: http://www.rnp.br/cais/estatisticas/index.php

A Figura 5 mostra um aumento relevante no número de incidentes reportados que supera o ano de 2008, embora ainda estarmos no primeiro trimestre de 2009.

A Figura 6 mostra que o mês de março de 2009 foi o recorde em número de incidentes de todos os tempos.

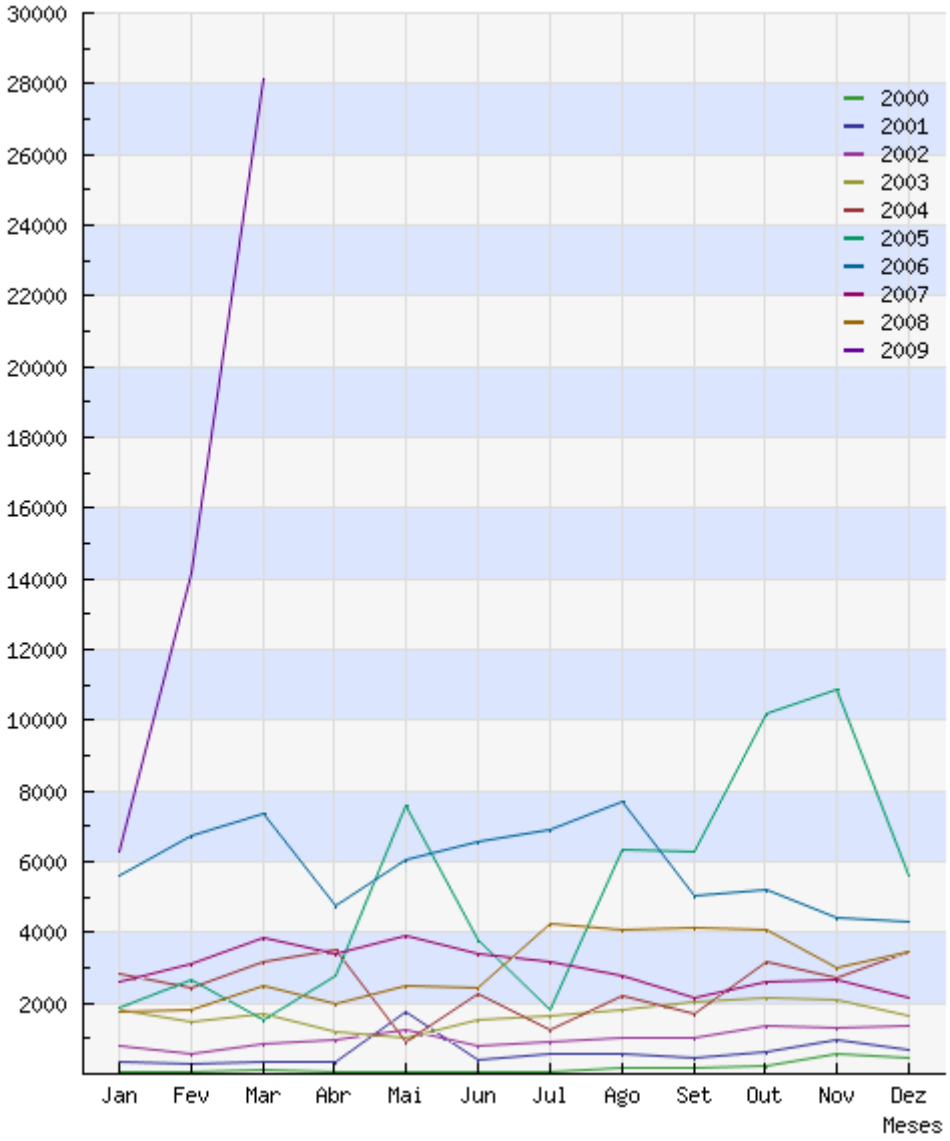


Figura 6 – Número de incidentes por mês registrado pelo CAIS

Fonte: http://www.rnp.br/cais/estatisticas/index.php

Sabendo que a segurança e a perícia tentam acompanhar os rastros deixados pelos crimes, a evolução destes tende sempre a deixar menos rastros possíveis. Ficando cada vez mais difícil a percepção pela perícia, e dando origem a ciência anti-forense, na qual deriva as técnicas anti-forense. Que é o novo grande desafio para a comunidade forense computacional.

1.2.2 A ciência anti-forense

Segundo o famoso princípio de Locard que é considerado o plano de fundo da ciência forense universal, estabelece que todas as pessoas deixam marcas de contato totalmente identificáveis. Baseado em fibras de roupas, carpetes e felpas, nos quais são suficientes para os criminalistas chegarem ao suspeito no local e conseguir a sua condenação (DISCOVERY, 2008).

Este princípio também se aplica na forense computacional, uma vez que todo evento realizado em um computador deixa alguma fonte de informação residente em algum tipo de memória, seja volátil ou não volátil, salvo a memória volátil após algum tempo de uso ou uma interrupção de energia elétrica no sistema.

A ciência forense se esforça para descobrir as fontes de informações deixadas bem como discernir o seu significado relacionando-o com o evento. O exame exige que as evidências sejam confiáveis e rigorosas para garantir o resultado correto, mas, no entanto, os criminosos podem utilizar de métodos anti-forense para trabalhar contra o processo ou interferir na fidelidade das evidências (HARRIS, 2006).

Com as técnicas anti-forense, os peritos devem ir além das evidências, e usar um maior grau de abstração para entender o verdadeiro problema, mas, ainda não existem referências gerais de existências de assinaturas das técnicas que permitam os peritos entenderem qual a solução anti-forense foi usada. A problemática está bem além do que se imagina, pois academicamente, ainda não existe uma definição concisa sobre a anti-forense. Do mesmo jeito que não existe um agrupamento geral que defina uma padronização ou categorias de técnicas utilizadas para compreender e determinar exemplos de mitigação e estratégias para a ajuda da resolução do problema (HARRIS, 2006).

Atualmente a anti-forense não possui uma definição unificada, sendo um reflexo de o campo ser ainda inexplorado e novo, mas já existem várias definições que estão disponíveis, e cada uma tem os seus méritos relativos, em que abrangem pontos específicos ou alguns segmentos da anti-forense (HARRIS, 2006).

Segundo a Figura 7, podemos ver que as estatísticas mostram que a anti-forense foi notada recentemente através do uso intensivo, mas nada impede que

antes desta data, as técnicas já tenham sido usadas em menor escala e complexidade como o exemplo da esteganografia.

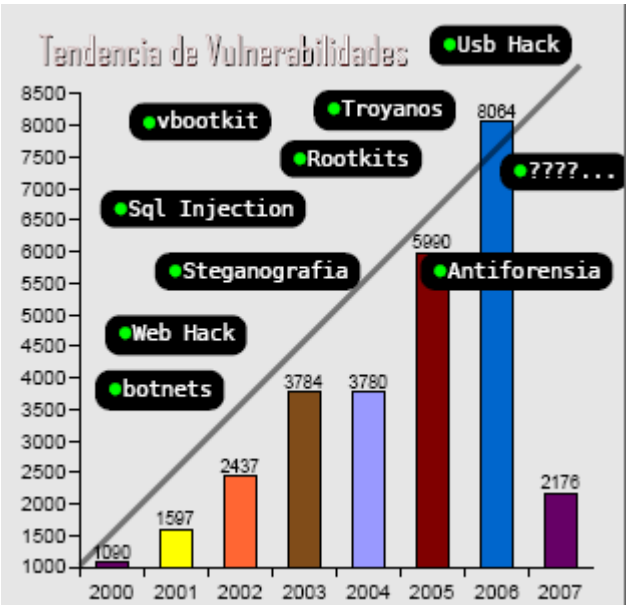


Figura 7 – Quantitativo do uso das técnicas anti-forense

Fonte: Fonte: Almanza, VII Jornada nacional de seguridad informática, 2007, p. 4.

Dentre os mais variados autores que definem a anti-forense, alguns simplesmente discutem as ferramentas que evitam a detecção, Foster e Liu (2005) apud Harris (2006), enquanto outros apenas relacionam com os sistemas de intrusões como Shirani (2002) apud Harris (2006). Seguindo o mesmo raciocínio de Harris (2006), no qual definiu concisamente a anti-forense baseado nas definições de Peron e Lergay (2005), e Grugg (2005), temos que a anti-forense “pode ser qualquer tentativa de comprometer a disponibilidade ou a utilidade de provas para o processo forense” (tradução nossa). Então a definição fica compreendida em comprometer a disponibilidade da apresentação de evidências, assim como escondê-las ou de alguma outra forma manipulá-las para garantir que não esteja mais ao alcance do perito.

A anti-forense originou-se devido à demanda dos ambientes coorporativos precisarem da ciência forense como uma ferramenta de apoio a resolução dos crimes cibernéticos, logo, surgiu à necessidade dos atacantes consolidarem a intrusão sem serem percebidos, onde, concentraram esforços em cima das normas e procedimentos da ciência forense e do sistema a ser atacado, para que os rastros sejam eliminados (HARRIS, 2006).

Segundo a Figura 8, podemos observar um ciclo para a consolidação de um ataque com o uso da anti-forense.

Figura 8 – Etapas de consolidação de um ataque utilizando técnicas anti-forense
Fonte: Almanza, VII Jornada nacional de seguridad informática, 2007, p. 5.

Com a redução das evidências ou até mesmo a eliminação definitiva, um administrador de sistemas mesmo que experiente, sente a dificuldade de perceber o seu sistema comprometido. Conseqüentemente, a administração deve ir além das ações básicas de monitoramento de *logs* ou de softwares específicos. O administrador deve acompanhar historicamente todo o processamento da máquina o mais próximo do funcionamento do sistema operacional / *hardware*. Ou seja, uma análise em camadas mais baixas, para que seja possível identificar as técnicas anti-forense utilizadas dentro das suas classificações.

1.2.3 Classificações da anti-forense

Vários pesquisadores tem proposto dividir as técnicas anti-forense em categorias, e cada autor difere de um ponto de vista na divisão das técnicas. Segundo (Perón; Legary, 2005 apud Harris, 2006), eles dividem as técnicas em quatro categorias, onde um atacante pode destruir, ocultar, manipular ou impedir a

criação de provas. Outras categorias são incluídas por Rogers (2005) apud Harris (2006) como esconder dados, uso de ferramentas de *wipe*, ofuscação de informações, ataques contra os processos e ferramentas.

Mesmo com diferenças na denominação das categorias entre os pesquisadores, algumas possuem o mesmo significado técnico, como podemos citar os dados escondidos de Rogers (2005) com a ocultação de Peron e Legary (2005). Seguindo o mesmo raciocínio de Harris (2006), iremos classificar as técnicas anti-forense em quatro grupos gerais:

1. Destruição de evidências
2. Ocultação de evidências
3. Eliminação das fontes de evidências
4. Falsificação de evidências.

1. Destruição: As evidências podem ser destruídas completamente ou parcialmente tanto em nível lógico como físico, evitando que sejam encontradas. Mas caso isso aconteça, terão pouca utilidade como prova forense (HARRIS, 2006). Dados sobrescritos em uma mídia magnética não podem ser recuperados em nível de software, salvo condições especiais como a análise magnética em *hardware* (VECCO, 2004 apud FARMER; VENEMA, 2006). Da mesma forma como a impossibilidade de recuperação de dados em memória DRAM quando o sistema é desligado, salvo outras condições especiais não documentadas pelos fabricantes dos chips de memória (GUTMANN, 2001).

Este tipo de técnica é bastante fácil quando aplicada a uma destruição total das informações, mas para uma destruição específica, se torna um pouco mais elaborado. A destruição física do disco através de elementos externos como campos magnéticos, choques ou queima intencional de algum dispositivo semicondutor, pode também inviabilizar a perícia.

2. Ocultação: A ocultação das evidências reduz as chances de um sucesso de investigação preliminar, exigindo que a perícia seja mais minuciosa (HARRIS 2006). O simples fato de ocultar um arquivo em um sistema de arquivos é considerado

como tal técnica, bem como utilizar ferramentas de criptografia e esteganografia, que podem ser facilmente usadas além de estarem livremente disponibilizadas na internet. Esconder informações em lugares não convencionais do sistema de arquivos – *slack space*, são estratégias de ocultação preferidas pelos atacantes. Em ultimo caso, pode ser usado várias estratégias em conjunto, como uma criptografia de uma informação esteganografada e armazenada nos *slack spaces*. Esta sem dúvida, é um dos estados da arte da anti-forense, e mesmo com uma perícia altamente sofisticada e minuciosa, a percepção fica muito difícil.

3. Eliminação das fontes de evidências: É o simples fato da remoção ou da não utilização de métodos para que as evidências não sejam criadas (HARRIS, 2006). Fazendo um paralelo a outras disciplinas forenses, é como usar luvas de borracha para cometer um crime. Logo, a falta de evidência pode ser questionada pelo perito como uma evidência de que o criminoso planejou cuidadosamente o crime (HARRIS, 2006). No mundo digital, isso pode ser considerado através do uso de memórias externas com programas portáveis, que faz com que sejam geradas poucas ou até nenhuma evidência no disco local, assim como *live* CDs.

4. Falsificação**:** Esta é uma das categorias mais complexas e elaboradas, sendo o maior desafio para a forense. A falsificação pode comprometer desde apenas um bit até certa quantidade de bits contidos em um disco, com o intuito de parecer qualquer outra coisa, menos a evidência real (Harris, 2006). Pode ser usado para incriminar inocentes, além de mudar códigos de processos legítimos, induzindo o perito ao erro. Mudança de strings em arquivos, alteração dos *mactimes*, atualização de *bad clusters* falsos no sistema de arquivos e dentre várias outras possibilidades que a criatividade exige, são exemplos de possíveis falsificações. Dependendo da elaboração desta técnica, os exemplos citados anteriormente podem ser executados sem deixar evidências no disco de que foram alterados propositalmente, uma vez, que as alterações podem ser realizadas a baixo nível do disco, ou seja, a nível binário. Com base na Figura 10, podemos entender as camadas de abstração de um disco rígido, e visualizar que a camada binária está abaixo do sistema de arquivos, todavia, qualquer acesso direto aos binários, faz com que o sistema de arquivos não consiga registrar nenhum evento e mudança de qualquer informação de um setor do disco.

De acordo com o princípio de Locard, sempre existirão evidências, só que neste caso, estarão residentes em memória DRAM e registradores do processador, na qual estão compreendidas em códigos de baixo nível para o acesso direto ao disco. Conseqüentemente, essas evidências podem ser sobrescritas facilmente, ou desaparecidas após um reset ou desligamento do sistema.

Para um atacante aplicar as técnicas anti-forense em sua totalidade, ou seja, destruir, ocultar, eliminar e falsificar evidências, o mesmo deve possuir um conhecimento profundo sobre a estrutura e funcionamento do mecanismo do sistema de arquivos, além da tecnologia de funcionamento da memória que o armazena. Analisando a Figura 9, para cada camada de abstração a partir da binária, existe uma técnica anti-forense mais adequada, com complexidade e eficiência maiores em camadas mais baixas.

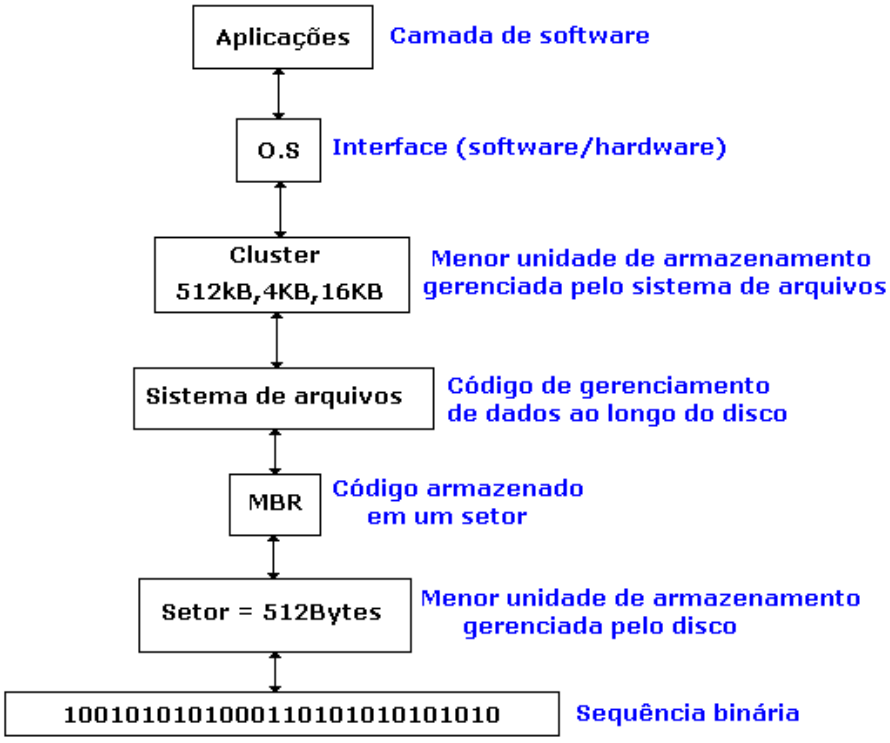


Figura 9 – Camadas de abstração de um disco rígido

A Figura 10 mostra um resumo dos métodos da anti-forense com os seus modos de operação.

Name	Destroying	Hiding	Eliminating source	Counterfeiting
MACE alterations	Erasing MACE information or overwriting with useless data			Overwriting with data which provides misleading information to investigators
Removing/wiping files	Overwriting contents with useless data	Deleting file (overwriting pointer to content)		
Data encapsulation		Hiding by placing files inside other files		
Account hijacking				Evidence is created to make it appear as if another person did the "bad act"
Archive/image bombs				Evidence is created to attempt to compromise the analysis of an image
Disabling logs			Information about activities is never recorded	

Figura 10 – Resumo das categorias anti-forense

Fonte: HARRIS, Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem, 2006, p. 2.

Os métodos anti-forenses são baseados na forma como os peritos conduzem qualquer tipo de perícia Grugq (2005) apud Harris (2006), além da dependência das ferramentas e processos (FOSTER; LIU, 2005 apud HARRIS, 2006). A anti-forense aproveita-se das limitações físicas e lógicas dos processos de investigação – tecnologias de *hardware* e *software*, fazendo com que a forense nunca possa se prevenir completamente da corrupção das evidências, mas pelo menos, tentar uma minimização da aplicação da anti-forense (ROGERS, 2005 apud HARRIS 2006).

A anti-forense exige muito mais rigor e dedicação por parte dos peritos além de ferramentas de investigação muito mais sofisticadas. Contudo, o elemento humano será determinante no sucesso da investigação, pois será necessário que o perito tenha mais agilidade, experiência, abstração e conhecimento para entender as medidas anti-forense que foram aplicadas (ROGERS, 2005 apud HARRIS 2006). A dependência da forense por ferramentas faz com que a anti-forense estude-as no intuito de encontrar vulnerabilidades, criando outros métodos que serão despercebidos por estas ferramentas (Harris, 2006). Para mitigar este problema, seria necessário utilizar uma variedade de ferramentas, ou forçar os fabricantes a melhorarem a precisão e eficácia delas, porém o custo é muito alto, além da

atualização e manutenção destas várias ferramentas, que, no entanto, não é nada atrativo para os departamentos ou empresas especializadas (HARRIS, 2006).

A Figura 11 mostra os exemplos dos métodos de exploração da anti-forense.

Name	Human element	Tool dependence	Physical/logical limitations
MACE alteration	Investigator may assume accuracy of dates and times	Tools may not function with invalid or missing dates and times	Invalid dates and times make collating information from multiple evidentiary sources difficult or impossible
Removing/wiping files	Investigator may fail to examine deleted files	Methods of restoring deleted files are specific to the tool – so effectiveness may vary	Time required to restore wiped file contents may outweigh the evidentiary value of the data it contained
Account hijacking	Investigator may fail to consider whether the owner of the account was actually the person at the keyboard	Tool may not be capable of extracting information that would aid investigator in determining who was in control of the account	Zombied computer accounts may produce so much indirection that it is almost impossible to actually find the origin of an attack. Lack of detailed information may keep investigator from determining actual account user
Archive/image bombs		Improperly designed software may crash	Useful information might be located in the bomb itself, but outside the logical limitations of the investigator's system
Disabling logs	Investigator may not notice missing log records	Software may not flag events that indicate logging was disabled	Missing data maybe impossible to reconstruct

Figura 11 – Métodos de exploração da anti-forense

Fonte: HARRIS, Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem, 2006, p. 3.

CAPÍTULO 2 - ORGANIZAÇÃO LÓGICA DO DISCO

Este capítulo aborda como o disco rígido organiza seus dados, bem como o funcionamento do sistema de arquivos NTFS, que é a base para o entendimento da aplicação e funcionamento das técnicas anti-forense neste sistema de arquivos.

2.1 Master Boot Record – MBR

O *Master Boot Record* (MBR) é um setor de inicialização de qualquer unidade de armazenamento, estando estrategicamente posicionado no primeiro setor do disco (TEODOROWITSCH, 2000). O MBR contém um código de inicialização juntamente com uma tabela de partições, na qual, constam todas as informações relevantes sobre o mapeamento de todas as partições do disco, assim como informações dos sistemas de arquivos instalados.

A principal função do MBR segundo Teodorowitcsh (2000) é fazer a transferência dos códigos de inicialização do sistema operacional para a memória DRAM, para que o próprio sistema tenha a própria autonomia de funcionamento.

O MBR é uma pequena porção de dados que contém apenas 512Bytes, sendo esta, a referência para os computadores da arquitetura IBM/PC e o valor padrão mínimo de armazenamento de um disco, conhecido como setor (TEODOROWITSCH, 2000).

Segundo a Figura 12, podemos visualizar a composição mais detalhada do MBR.

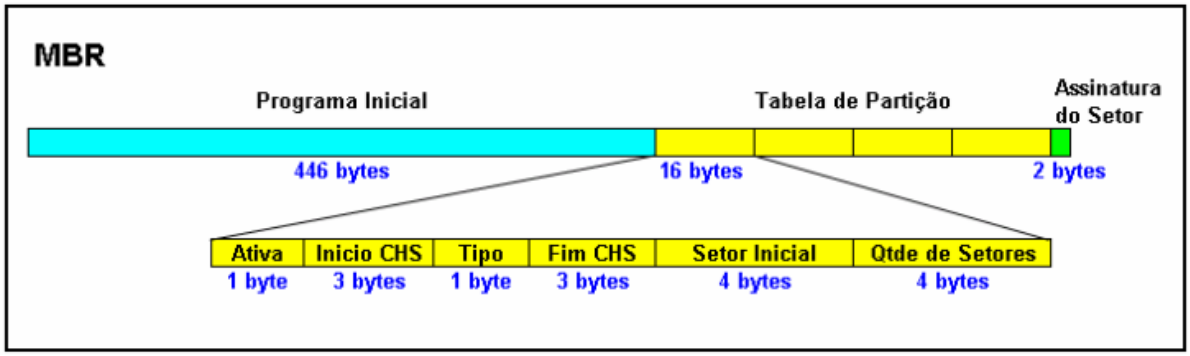


Figura 12 – Organização lógica do MBR

Fonte: SANTOS, Uma abordagem sobre recuperação de dados em disco rígido, 2002, p. 19.

Quando um computador é ligado, são executadas as rotinas de inicialização do *Basic Input Output System* (BIOS), e sua ultima instrução é chamada de *bootstrap*. Que tem por função, procurar pelo MBR e executar o programa inicial contido nele e carregá-lo para a memória DRAM. O programa inicial contido no MBR é conhecido como *bootloade*r, e tem como função, checar e interpretar todos os outros campos restantes do MBR apontando para o respectivo setor que possui a inicialização do sistema de arquivos. Caso o sistema de arquivos esteja corrompido, o próprio MBR se encarrega de gerar uma string contendo o erro de inicialização. A Figura 13 mostra os binários do MBR, com os seus respectivos campos de informações.

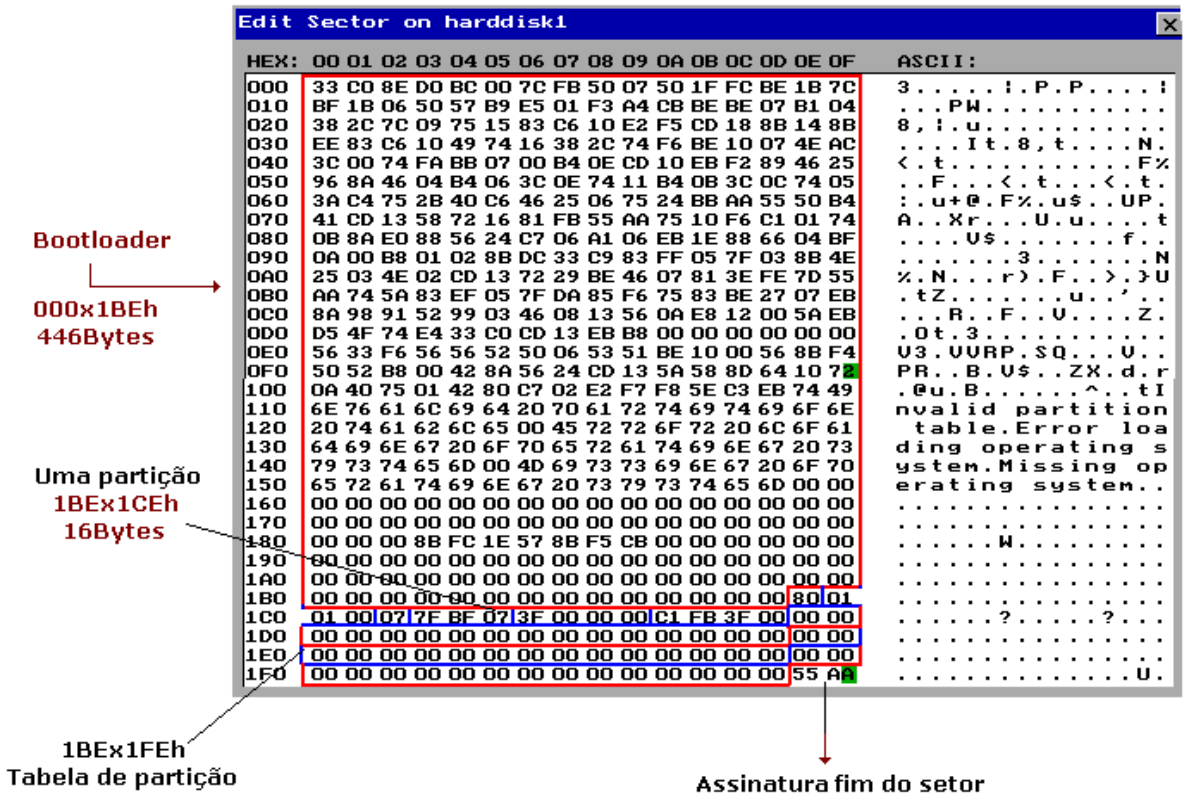


Figura 13 – Binários do MBR pelo editor de disco

Fonte: Software Partition Table Doctor

Com o MBR corrompido, ou com algum dado inconsistente, a inicialização não terá sucesso e sendo este, um dos grandes pontos para a utilização das técnicas anti-forense.

Através do acesso direto ao disco, o MBR pode ser alterado de forma a deixá-lo corrompido ou levar a inicialização de um outro falso sistema contido no disco. O

MBR tem uma grande importância direta em qualquer análise do disco, seja ela feita através de softwares forenses ou de qualquer ferramenta que avalie o disco de forma lógica. Entretanto, todos eles dependem de como o MBR está mapeado para que seja montada toda a organização lógica do disco. Consequentemente, para um MBR alterado, a perícia deve analisar o disco minuciosamente a nível binário para a procura de divergências de informações.

A Figura 14 mostra mais detalhes de um exemplo de mapeamento de uma partição no disco através da interpretação do trecho dos binários que trazem toda a informação. Esses binários indicadores são conhecidos como *flags*.

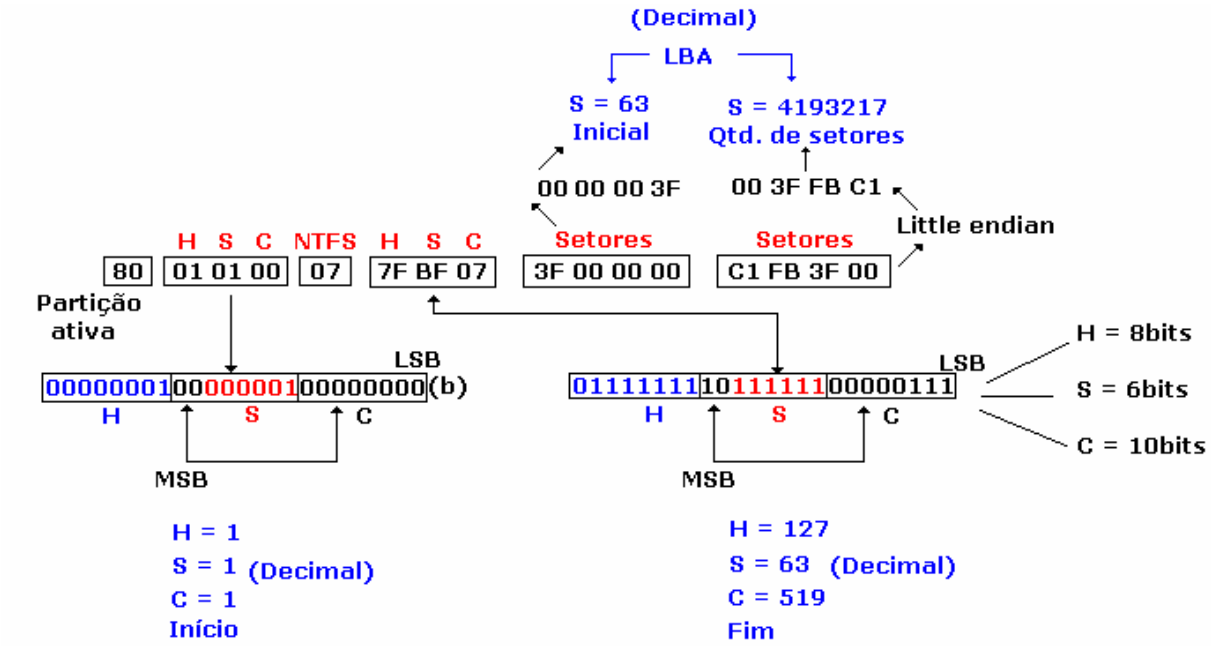


Figura 14 – Exemplo de flags de uma tabela de partição do MBR

A partir da figura anterior, podemos entender melhor que a alteração de qualquer binário pode levar a um falso mapeamento e também, a uma falsa análise lógica do disco. Com isso, podemos entender um pouco da gravidade das técnicas anti-forense, que podem levar a perícia a uma análise cada vez mais absoluta dos fatos. Resultando em um processo demorado e às vezes sem chances de sucesso.

A Figura 15 mostra um software que traduz os binários do MBR, automatizando o processo da perícia.

Com o entendimento do MBR, o próximo passo, é conhecer a organização do sistema de arquivos NTFS, no qual depende do MBR inicialmente para funcionar.

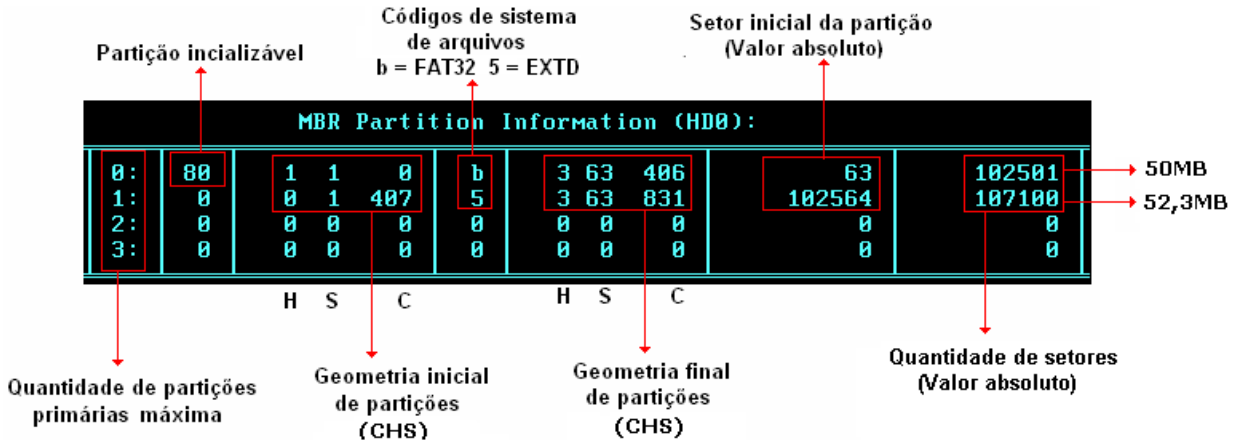


Figura 15 – Processo automatizado de interpretação dos binários do MBR

Fonte: Software MBR WORK

2.2 Sistema de arquivos NTFS

2.2.1 Conceitos preliminares

Um sistema de arquivos tem por função, garantir um método de organização para armazenar e recuperar dados a qualquer prazo, similar a um armário que contém inúmeros arquivos catalogados (CARRIER, 2005). Todos os arquivos contidos em um disco são referenciados por *flags*, que estão organizadas dentro de uma tabela de alocação de arquivos (TAA), para que estes sejam posteriormente localizados, além de outras informações substanciais.

O *New Technologies File Systems* (NTFS) foi desenvolvido pela Microsoft, sendo o sistema de arquivos padrão entre os sistemas operacional NT, 2000, 2003, XP e VISTA (CARRIER, 2005). É muito mais complexo que o FAT devido a sua maior robustez, uma vez que foi projetado para dar confiabilidade, segurança e suporte para discos de grande capacidade, porém, o NTFS será o sistema de arquivos mais comum em investigações forenses (CARRIER, 2005).

Infelizmente não há uma especificação de publicação da Microsoft que descreve o *layout* do disco em baixo nível para o NTFS, mas existem algumas documentações que mostram a topologia em alto nível, porém, o que se sabe sobre

a estrutura do NTFS através de engenharia reversa, ainda não é exatamente comprovado pela Microsoft (CARRIER, 2005).

Para o reconhecimento e funcionamento do sistema de arquivos, o mesmo necessita inicialmente do MBR carregado em memória DRAM para que indique o setor de inicialização do sistema de arquivos.

Todo sistema de arquivos possui um setor de *boot* (inicialização) que serve para armazenar informações específicas sobre o próprio sistema de arquivos e proporcionar a inicialização do sistema operacional (CARRIER, 2005). De acordo com as Figuras 16 e 17, podemos entender a arquitetura do NTFS e as rotinas de inicialização do sistema.

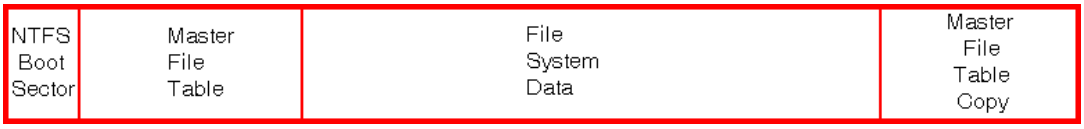


Figura 16 – Arquitetura de um sistema de arquivos NTFS

Fonte: Computer Forensics – NTFS File System, p. 4.

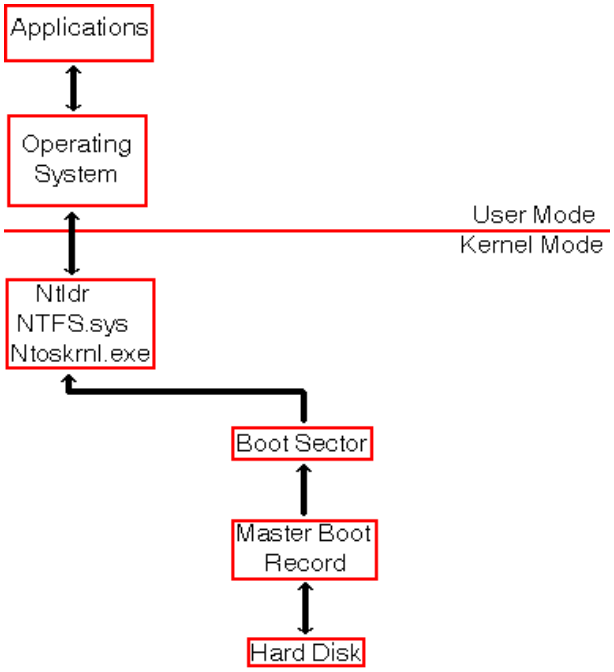


Figura 17 – Rotina de inicialização do sistema

Fonte: Computer Forensics – NTFS File System, p. 3.

A Figura 18 mostra os binários contidos no setor de *boot* do sistema de arquivos NTFS, através da ferramenta *Winhex.*

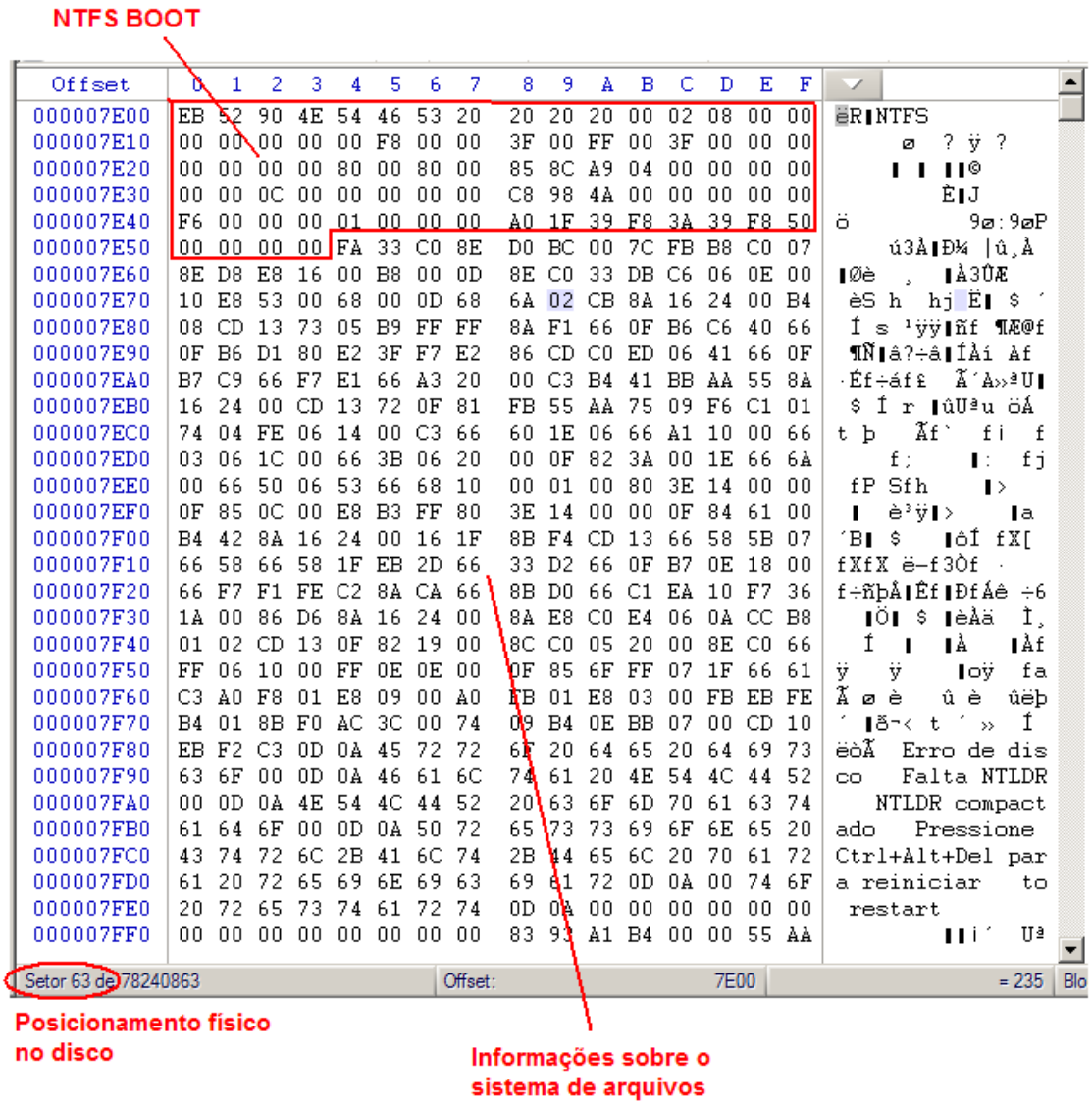


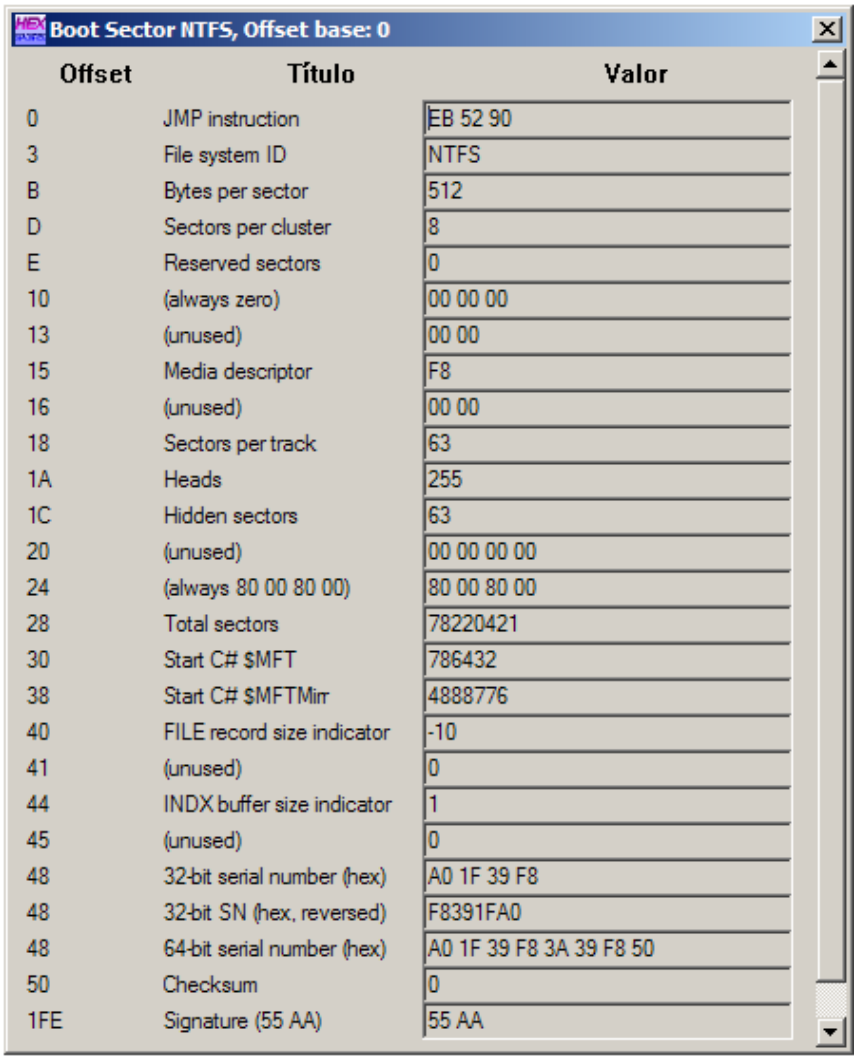
Figura 18 – Binários do setor de inicialização do sistema de arquivos NTFS

Fonte: Software Winhex

A partir da figura anterior, podemos notar que o setor de boot do sistema de arquivos procura por arquivos de incialização do sistema operacional para que então todo o processo de carga do sistema seja dado. No entanto, desde o primeiro *reset* da máquina que compreende as rotinas básicas do BIOS, até a carga do sistema operacional, existem várias sucessões de incializações que são indicadas por setores de *boot* específicos. Esta região de inicialização do sistema de arquivos compreende 6 setores.

Algumas técnicas anti-forense podem ser aplicadas no próprio setor de inicialização do sistema de arquivos, com a alteração de *flags* estratégicas que visam a enganar o trabalho da perícia. As mudanças podem alterar desde o nome do volume até o dimensionamento dos *clusters* que o sistema de arquivos gerencia, ou até mesmo deixar o sistema irreconhecível ou inoperável. Conseqüente, caímos no mesmo problema que pode acontecer no MBR, ou seja, qualquer informação pode está passível de adulteração, e a análise será cada vez mais complexa.

A partir da Figura 19, podemos entender a importância das informações que existem no setor de inicialização do sistema de arquivos NTFS, no qual, existem softwares que automatizam o processo de interpretação das *flags*, que tanto ajudam na perícia, assim como também para os que querem aplicar as técnicas anti-forense.



Offset	Título	Valor
0	JMP instruction	EB 52 90
3	File system ID	NTFS
B	Bytes per sector	512
D	Sectors per cluster	8
E	Reserved sectors	0
10	(always zero)	00 00 00
13	(unused)	00 00
15	Media descriptor	F8
16	(unused)	00 00
18	Sectors per track	63
1A	Heads	255
1C	Hidden sectors	63
20	(unused)	00 00 00 00
24	(always 80 00 80 00)	80 00 80 00
28	Total sectors	78220421
30	Start C# $MFT	786432
38	Start C# $MFTMirr	4888776
40	FILE record size indicator	-10
41	(unused)	0
44	INDX buffer size indicator	1
45	(unused)	0
48	32-bit serial number (hex)	A0 1F 39 F8
48	32-bit SN (hex, reversed)	F8391FA0
48	64-bit serial number (hex)	A0 1F 39 F8 3A 39 F8 50
50	Checksum	0
1FE	Signature (55 AA)	55 AA

Figura 19 – Interpretação automatizada dos binários do setor de inicialização do NTFS

Fonte: Software Winhex

Os dados mostrados anteriormente são como uma padronização do NTFS, e partir disto, é que as ferramentas forenses interpretam toda a estrutura montando o cenário lógico do disco. Uma vez alterados, essas ferramentas podem dar resultados incoerentes, e mais uma vez, a perícia necessita de uma análise binária das informações para confrontar resultados.

Na organização física do disco, nem sempre os dados são colocados de forma seqüencial. Como exemplo real, temos o MBR que está no primeiro setor absoluto do disco, e o *boot* do sistema de arquivos se encontra no setor 63, ou seja, já existem inicialmente 62 setores livres, nos quais podemos aproveitar como *slack spaces* para inserir dados. Além de vários outros setores vazios que vão se formando ao longo do disco, que, no entanto não podem deixar de existir, pois fazem parte da organização lógica do disco.

Com o entendimento do setor de inicialização do sistema de arquivos, o próximo item importante dentro do sistema é a TAA, que é formada pela *Master File Table* (MFT).

2.2.2 Master File Table - MFT

A *Master File Table* (MFT) é a TAA do sistema de arquivos NTFS, que é considerada o coração do sistema, pois contem todas as informações sobre a posição e atributos de todos os arquivos e diretórios (CARRIER, 2005). Para todo arquivo ou diretório, existe uma entrada na MFT, onde cada uma delas ocupa um espaço de 1024Bytes, que estão compreendidos em cabeçalho e atributos (CARRIER, 2005). A Figura 20 mostra a estrutura de uma entrada na MFT e a Figura 21 mostra os seus binários. A MFT além de possuir as informações de atributos dos arquivos, ela traz também as *flags* que apontam para os respectivos *clusters* do arquivo envolvido.

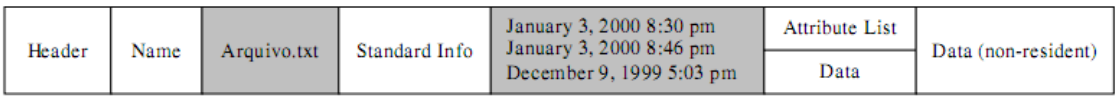


Figura 20 – Estrutura de uma entrada na MFT

Fonte: OLIVEIRA – Resposta a Incidentes e Análise Forense para Redes Baseadas em Windows 2000, 2002, p. 84.

Cada trecho de informação residente em uma entrada da MFT possui uma posição binária definida que facilita as possíveis alterações das informações através de técnicas específicas, assim como os tributos de tempos, que são uma das principais informações para a perícia (*mactimes).* Com a mudança dos atributos de tempo, existe uma distorção do cenário do crime, no qual o perito pode ser induzido ao erro na hora de relatar o histórico do crime.

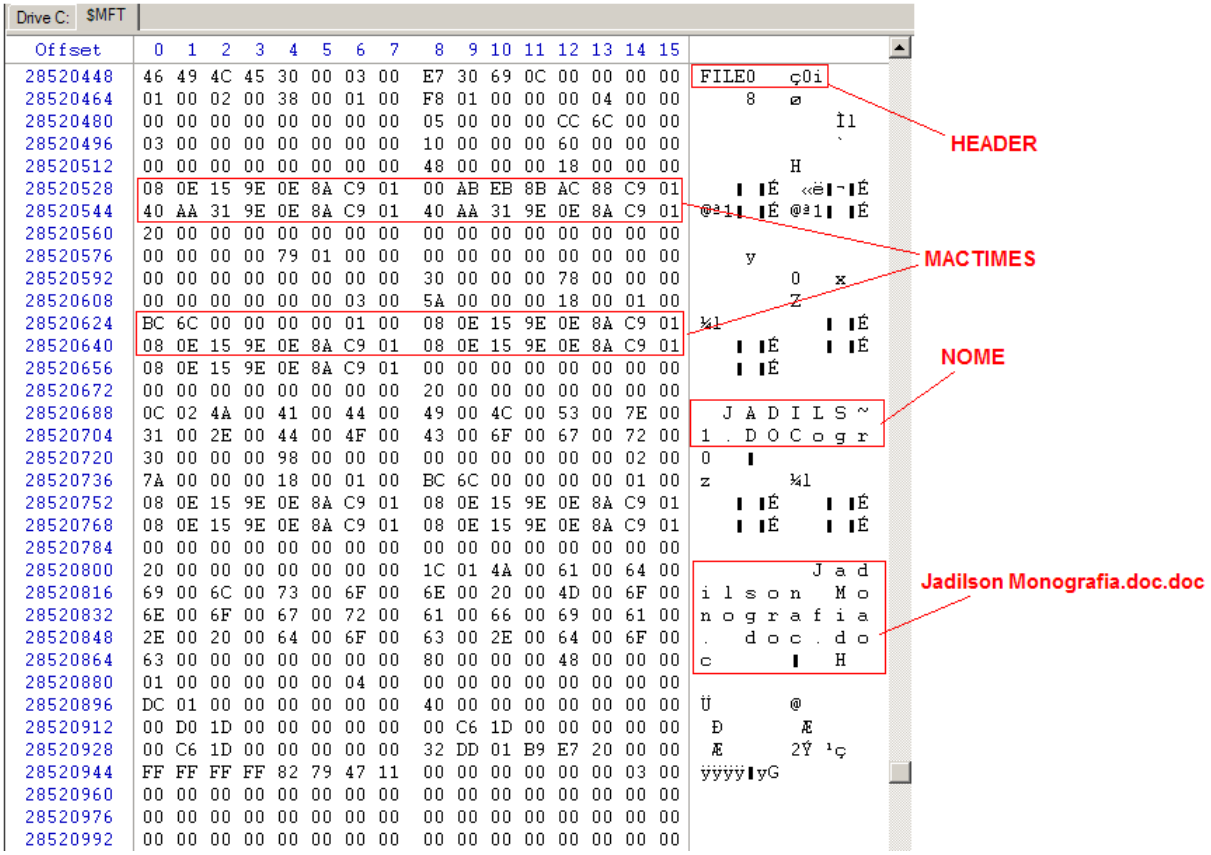


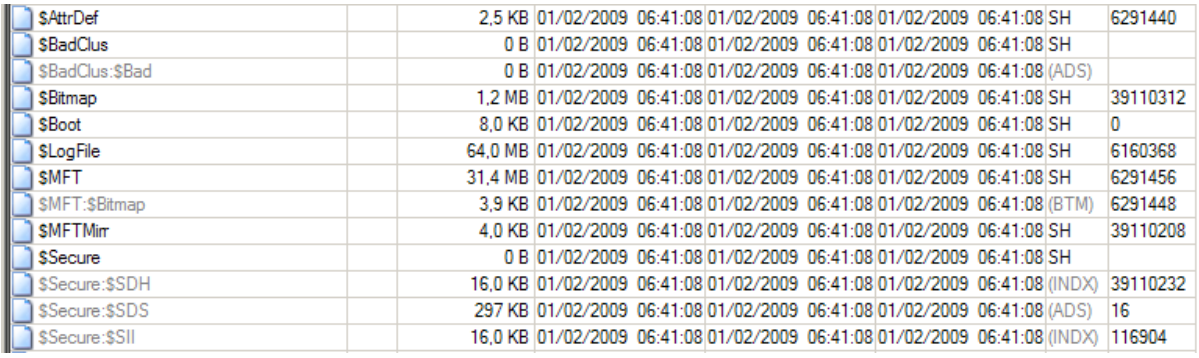
Figura 21 – Estrutura binária de uma entrada na MFT

Fonte: Software WinHex

Além da MFT, existem arquivos que contém meta informações usadas para implementar a própria estrutura do sistema de arquivos, que são mapeados no inicio da MFT, inclusive ela mesma (CARRIER, 2005). Para cada entrada de arquivo no registro da MFT, existe um número identificador no qual começa por zero – que é dado pela própria MFT, até o ultimo arquivo gravado no sistema. A partir dessas informações, as técnicas anti-forense podem adulterar qualquer uma delas, como exemplo, forjar falsos *bad clusters* para ocultar informações. Uma vez que esta meta

informação isola os clusters defeituosos a nível lógico, fazendo com que o sistema de arquivos não os acesse. Uma outra meta informação importante é o *journaling,* que traz todas as informações de transações efetuadas no disco, nas quais, são de grande avalia para a perícia.

De acordo com a Figura 22, podemos visualizar a MFT e os arquivos de meta informações.



$AttrDef		2,5 KB	01/02/2009 06:41:08	01/02/2009 06:41:08	01/02/2009 06:41:08	SH	6291440
$BadClus		0 B	01/02/2009 06:41:08	01/02/2009 06:41:08	01/02/2009 06:41:08	SH	
$BadClus:$Bad		0 B	01/02/2009 06:41:08	01/02/2009 06:41:08	01/02/2009 06:41:08	(ADS)	
$Bitmap		1,2 MB	01/02/2009 06:41:08	01/02/2009 06:41:08	01/02/2009 06:41:08	SH	39110312
$Boot		8,0 KB	01/02/2009 06:41:08	01/02/2009 06:41:08	01/02/2009 06:41:08	SH	0
$LogFile		64,0 MB	01/02/2009 06:41:08	01/02/2009 06:41:08	01/02/2009 06:41:08	SH	6160368
$MFT		31,4 MB	01/02/2009 06:41:08	01/02/2009 06:41:08	01/02/2009 06:41:08	SH	6291456
$MFT:$Bitmap		3,9 KB	01/02/2009 06:41:08	01/02/2009 06:41:08	01/02/2009 06:41:08	(BTM)	6291448
$MFTMirr		4,0 KB	01/02/2009 06:41:08	01/02/2009 06:41:08	01/02/2009 06:41:08	SH	39110208
$Secure		0 B	01/02/2009 06:41:08	01/02/2009 06:41:08	01/02/2009 06:41:08	SH	
$Secure:$SDH		16,0 KB	01/02/2009 06:41:08	01/02/2009 06:41:08	01/02/2009 06:41:08	(INDX)	39110232
$Secure:$SDS		297 KB	01/02/2009 06:41:08	01/02/2009 06:41:08	01/02/2009 06:41:08	(ADS)	16
$Secure:$SII		16,0 KB	01/02/2009 06:41:08	01/02/2009 06:41:08	01/02/2009 06:41:08	(INDX)	116904

Figura 22 – Arquivos de meta informações do sistema de arquivos NTFS

Fonte: Software WinHex

O Quadro a seguir mostra a função básica de cada meta informação

Quadro 1 – Função de todos os arquivos de meta informação

Nome do arquivo	Identificador da MFT	Descrição
$MFT	0	Master File Table
$MftMirr	1	Cópia dos 16 primeiros registros da MFT
$LogFile	2	Arquivo de log das transações efetuadas no disco
$Volume	3	Número de série do volume, data de criação e o dirty flag
$AttrDef	4	Definição dos atributos
$.	5	Diretório raiz do volume
$Bitmap	6	Representação do disco indicando que clusters estão sendo utilizados
$Boot	7	Setor de boot do volume
$BadClus	8	Clusters defeituosos
$Secure	9	Contém security descriptors únicos para todos os arquivos do volume
$Upcase	10	Mapeia caracteres minúsculos em seus correspondentes maiúsculos
$Extend	11	Usado por várias extensões opcionais, como quotas e reparse points
	12 - 15	Reservados para uso futuro.

Diante das técnicas anti-forense, qualquer dado dentro do disco está passível de adulteração, no entanto, o único limite para as mais variadas práticas da anti-forense é a criatividade.

CAPÍTULO 3 - METODOLOGIA

A pesquisa bibliográfica inicial foi realizada para a compreensão preliminar do que é estudado atualmente sobre anti-forense e com a esperança de encontrar algo concreto com relação a medidas práticas. Infelizmente, os títulos encontrados eram de ordem teórica e muito repetitiva, e sem demonstrações práticas. A obra mais concreta e atual sobre o tema foi a de Harris (2006), que abordava as técnicas anti-forense de forma apenas conceitual, no entanto, esta foi à base para a criação de uma metodologia prática para validar os conceitos apresentados por ele.

Na dependência de entender profundamente os possíveis mecanismos das técnicas, nos concentramos na forma de acesso ao disco em baixo nível, que resultou em consultas a obras clássicas de linguagem C, *assembly*, *hardware*, arquitetura de computadores, linguagem residente em BIOS, DOS, sistemas operacionais, tutoriais e artigos, que nos deram uma boa base para o começo da realização dos experimentos práticos para a aplicação das técnicas.

Como o objetivo geral deste trabalho era de aplicar as técnicas anti-forense em uma máquina local desktop para avaliar os seus impactos em análises forenses, foi criado um laboratório para que fossem realizados os experimentos baseados nas classificações de Harris (2006). Com isso, o laboratório foi composto por uma máquina desktop com três HDs físicos, que respectivamente foram usados como base do sistema da máquina real, sistema do ambiente virtual e repositório de imagens do ambiente virtual. Com exceção do HD de repositório de imagens, os outros dois possuíam o mesmo sistema operacional Windows XP montado sobre o NTFS, que foi o sistema de arquivos base para o estudo da aplicação das técnicas. As técnicas foram aplicadas no HD do ambiente virtual segundo as classificações de Harris (2006) na ordem de destruição, ocultação, eliminação e falsificação de evidências que posteriormente foram analisadas e avaliadas através do HD de repositório de imagens, para verificar o impacto de cada técnica na análise forense.

O ambiente virtual foi escolhido para esta metodologia devido à facilidade de coletar as telas para a demonstração neste trabalho.

As especificações da máquina utilizada no laboratório foram:

Processador Intel Celeron = 2GHz – 533MHz – LGA 775

Memória DRAM = DDR400 – 1GB

Placa mãe on-board = Vídeo, som, LAN

Três HDs = Samsung 32GB

Drive de CD

As experiências de destruição, ocultação, eliminação e falsificação de evidências foram realizadas repetidamente em torno de três vezes e comparado com um sistema real para que a análise no ambiente virtual não tivesse divergências.

Foram utilizadas ferramentas alternativas para a aplicação das técnicas, como softwares de formatação de baixo nível, particionamento, editor de disco e rotinas de programação em baixo nível. E como ferramentas específicas, as de criptografia e esteganografia.

A metodologia da aplicação das técnicas anti-forense bem como a escolha pelas ferramentas, se baseou primeiramente na tentativa de acesso ao disco em baixo nível para demonstrar a técnica de *slack space*. Sendo esta a mais motivadora e percussora de todas as outras técnicas subseqüentes. Com isso, os métodos foram se desenvolvendo de acordo com os experimentos para demonstrar o efeito prático dentro das classificações teóricas de Harris (2006).

A metodologia de análise forense foi baseada de acordo com toda a fundamentação teórica deste trabalho, que compreende em exame, análise e interpretação dos resultados, respeitando todos os procedimentos de coleta e manipulação das evidências através de imagens geradas para cada experimento na qual eram armazenadas no disco destinado a este fim.

Todas as ferramentas utilizadas no experimento estão livremente distribuídas na internet, nas quais, algumas são de código aberto e outras de versão para teste que funcionaram perfeitamente na demonstração dos impactos de cada técnica na análise forense.

CAPÍTULO 4 - O LABORATÓRIO

Este capítulo aborda todos os experimentos anti-forense realizados no laboratório de acordo com a metodologia proposta neste trabalho, bem como os seus resultados com relação aos impactos causados no cenário da forense computacional. Os experimentos estão ordenados dentro da classificação de Harris (2006) sobre a anti-forense.

4.1 Destruição de evidências

Dentro do que foi definida anteriormente no capítulo 1, a destruição se dá pela corrupção total ou parcial das evidências evitando que sejam encontradas, ou quando encontradas, não servirão de prova forense. Existem várias maneiras de se destruir uma evidência, que vão desde ações lógicas em nível de software, ou através de agentes externos para a destruição da mídia suspeita. A seguir, serão demonstradas as maneiras de se destruir evidências.

4.1.1 Formatação de baixo nível

Segundo a teoria, os *softwares* de formatação de baixo nível formatam o disco sobrescrevendo-o inteiramente através de codificações binárias, nas quais podem corresponder a “1s” ou “0s”. Todos os setores do disco estão envolvidos nesta formatação, inclusive o MBR que é sobrescrito nesta ação. Podemos dizer que esta é a verdadeira formatação de um disco, ao contrário das formatações lógicas que só apagam as *flags* da TAA que apontam para os arquivos ao longo do disco. No entanto, o conteúdo dos arquivos pode ser facilmente recuperado por *softwares* específicos para este fim, bem como as ferramentas forenses (SANTOS, 2002).

Cada fabricante de HD disponibiliza um formatador de baixo nível específico para os seus discos, que além da própria formatação, existe uma bateria de testes para certificar o estado físico e mecânico do disco.

Para demonstrar a ação do formatador de baixo nível, primeiramente visualizamos os binários do setor de *boot* do sistema de arquivos através do

software editor de disco que está mostrado na Figura 23 bem como os arquivos que estão contidos neste disco através da Figura 24, antes da formatação.

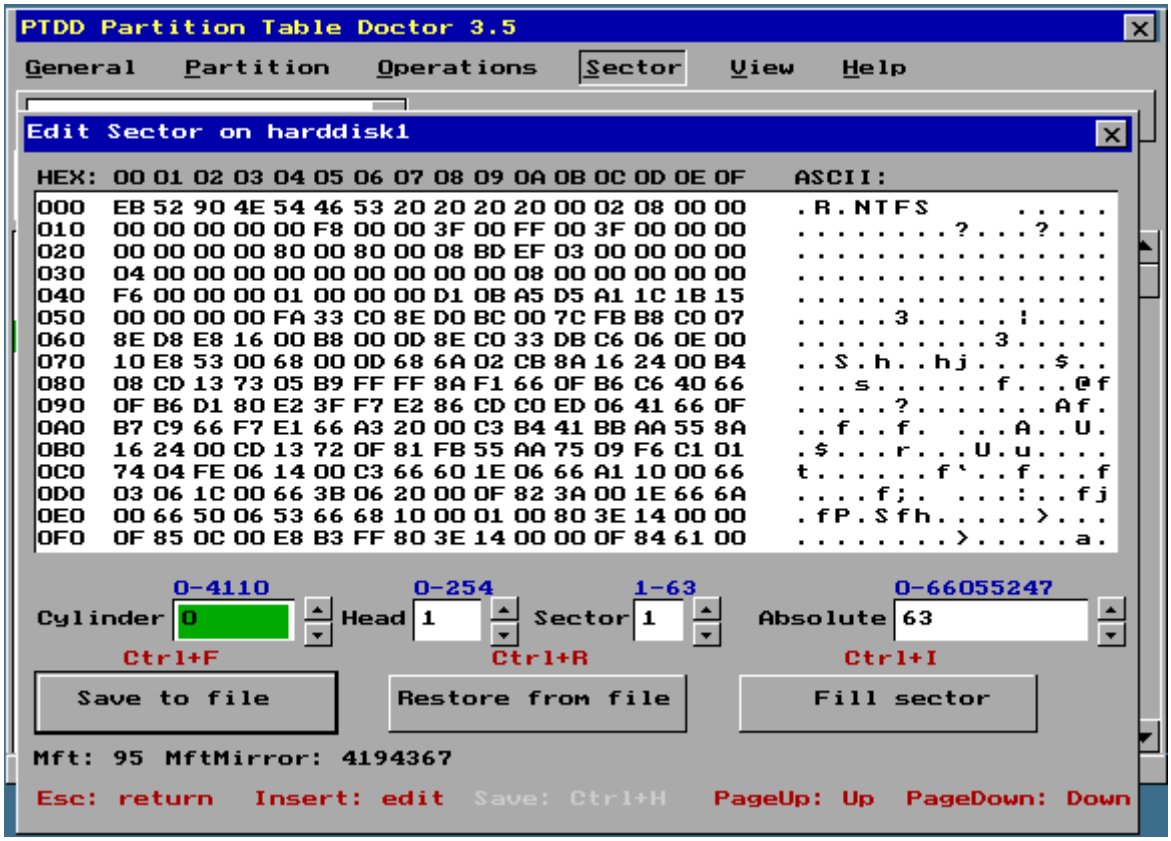


Figura 23 – Binários antes da formatação de baixo nível

Fonte: Software Partition Table Doctor

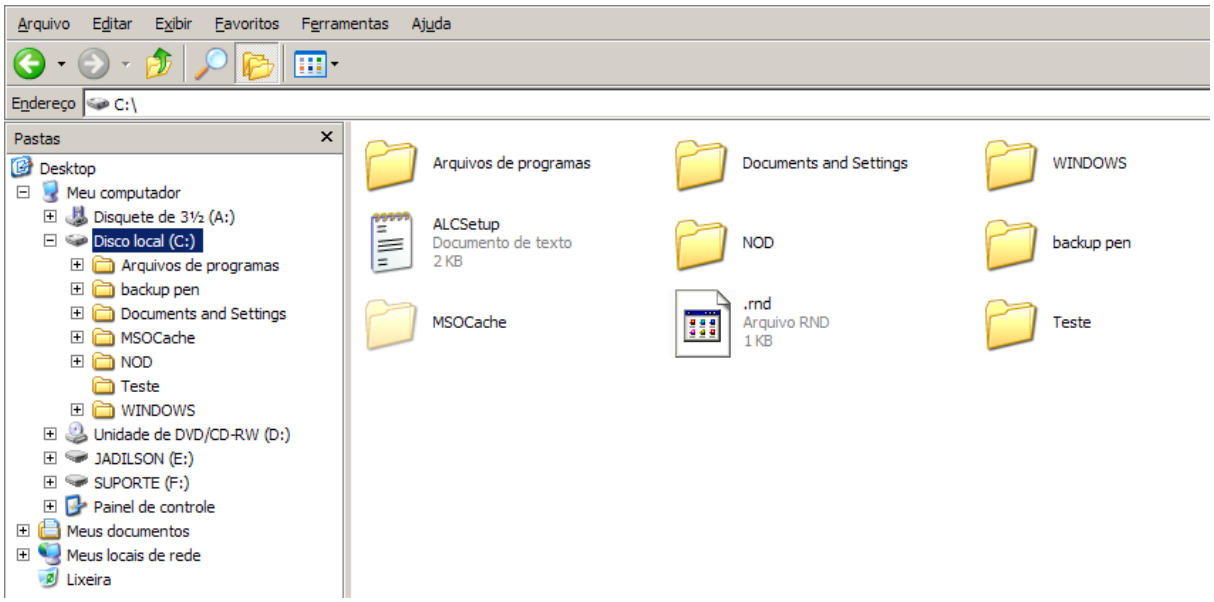


Figura 24 – Exemplo de conteúdo contido no disco antes da formatação

Com a ferramenta de formatação de baixo nível da Samsung, formatamos o disco, e verificamos através do mesmo editor de disco, que todos os dados dos setores foram sobrescritos com binários em zero. Comprovando o funcionamento da ferramenta de formatação. As Figuras 25 e 26 mostram as etapas.

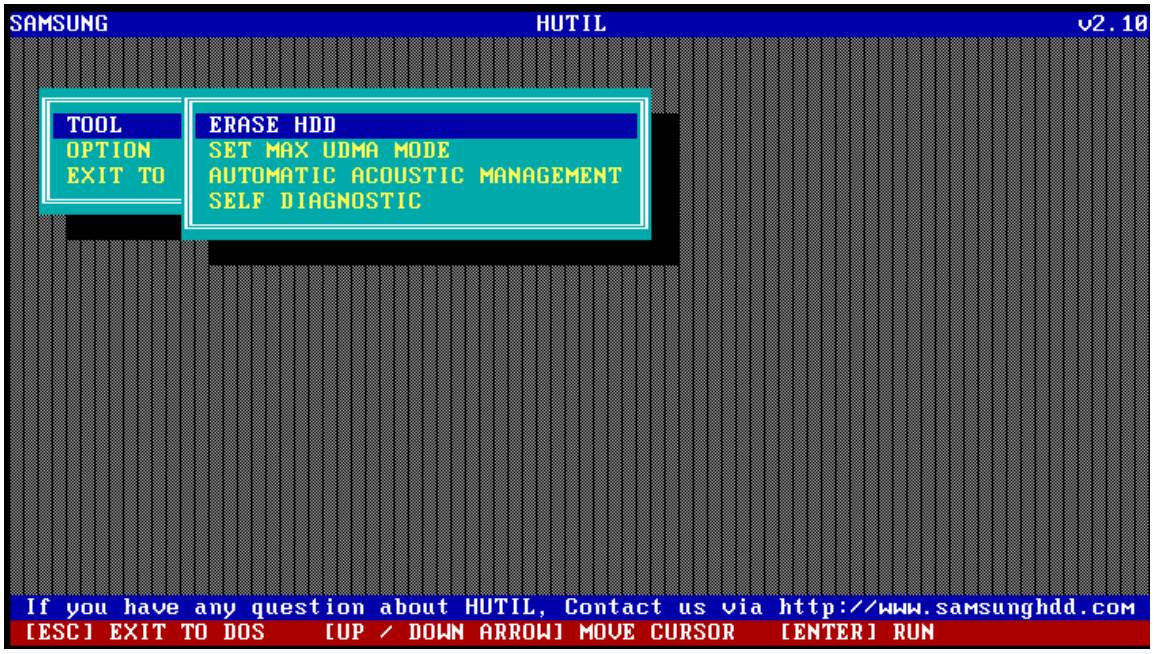


Figura 25 – Tela do software de formatação de baixo nível da Samsung

Fonte: Software Samsung Low Level Format

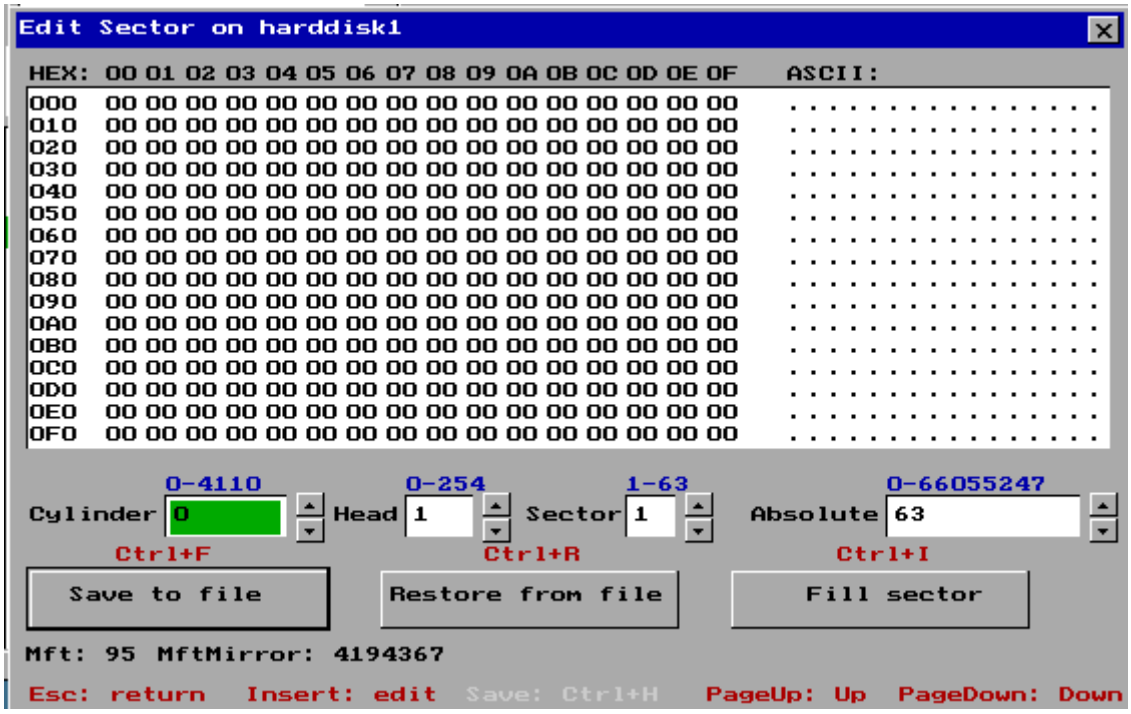


Figura 26 – Binários após a formatação de baixo nível

Fonte: Software Partition Table Doctor

Após a formatação de baixo nível, podemos concluir que toda ou qualquer informação contida no disco foi sobrescrita, e de acordo com (Vecco, 2004 apud Farmer; Venema 2006), os dados anteriores não podem ser recuperados em nível de software. No entanto, as evidências foram destruídas, assim como já podemos visualizar através do editor de disco Partition Table Doctor da Figura 26.

Para verificar o impacto desta técnica anti-forense e validar a destruição de evidências em nível de *software*, utilizamos o recuperador de arquivos *Get Back Data NTFS* na tentativa de encontrar qualquer resíduo de dados anterior. No entanto, encontramos um problema inicial devido ao disco está totalmente sem uma organização lógica. Logo, o sistema operacional bem como qualquer outra ferramenta, não vai enxergar o disco devido ao mesmo está sem o MBR e o sistema de arquivos. Com isso, a única maneira de análise do disco é através dos *softwares* editores de disco, que já mostramos anteriormente através da Figura 27.

Nesta primeira fase já podemos ver a dificuldade que esta formatação já causou, e para a tentativa de uma análise com este software de recuperação de arquivos, tivemos que formatar o disco em alto nível para que a MBR e o sistema de arquivos fossem compostos. Todavia o disco foi reconhecido pela ferramenta, mas, através desta formatação acabamos de sobrescrever mais uma parte do trecho do disco mais uma vez, sendo considerado pela forense um ato de imperícia.

A partir da Figura 28, podemos ver que o software mostrou apenas os arquivos do sistema NTFS compreendendo a MFT e totalizando apenas 132MB de um disco total de 32GB. Consequentemente, o *software* não conseguiu recuperar nenhum dado sobrescrito.

Para não questionar a eficácia do *software*, utilizamos à ferramenta forense Autopsy, que traz uma opção para recuperação de arquivos deletados. Porém, o mesmo não conseguiu também encontrar nenhum dado sobrescrito, exceto os do sistema de arquivos NTFS conforme o *software* anterior, como mostrado na Figura 28.

Conforme os resultados obtidos, provamos que em nível de *software*, não é possível uma recuperação de dados quando o mesmo é sobrescrito.

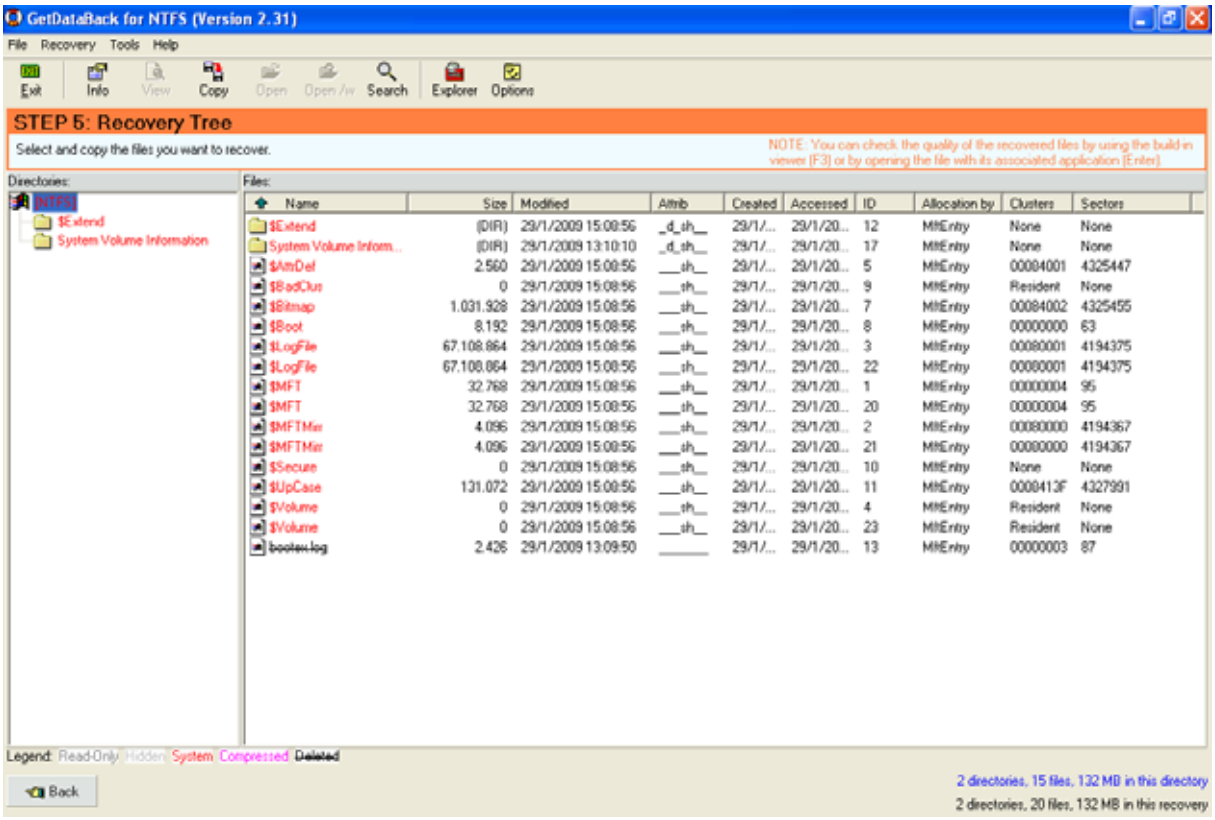


Figura 27 – Tela do Get Back Data Recovery NTFS

Fonte: Software Get Back Data Recovery NTFS

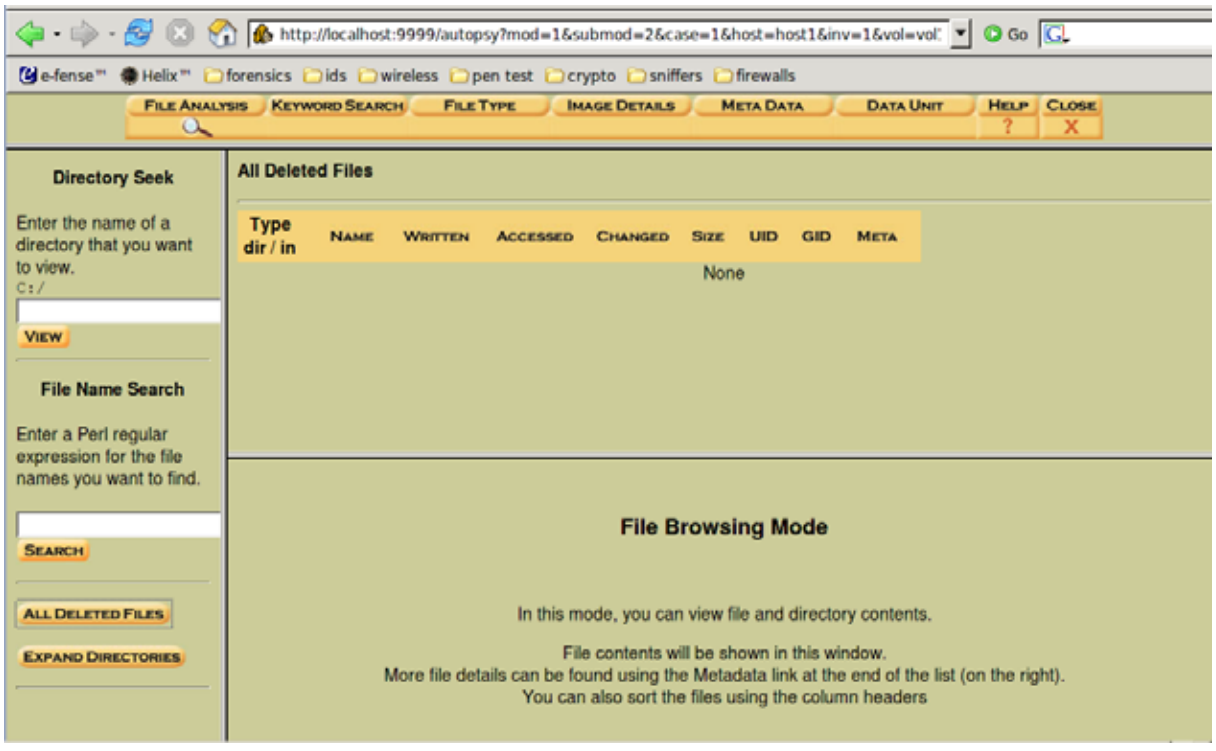


Figura 28 – Tela do Autopsy

Fonte: Software Autopsy

Logo, provamos à citação de (Vecco, 2004 apud Farmer; Venema 2006), na qual afirma que a única maneira de recuperar os dados sobrescritos é através de equipamentos, instalações, tempo e procedimentos sofisticados que não são viáveis na prática forense, mas que existe a possibilidade.

Esta teoria da possibilidade de recuperação é apresentada por Sobey (2004), que firma que, uma superfície de um disco pode armazenar dados anteriores. Isto é possível devido a pequenos desajustes da cabeça de leitura e gravação do disco, no qual, muda o seu curso de gravação dos dados de acordo com o funcionamento do disco (SOBEY, 2004). De acordo com a Figura 29 , podemos ver a superfície magnética do disco através de um microscópio de força atômica, que mostra os dados anteriores devido às imprecisões de gravação da cabeça de leitura e gravação do disco.

Segundo a *Recovery Lab* (2008) a recuperação dos dados pode ser possível até dez sobrescritas em um disco.

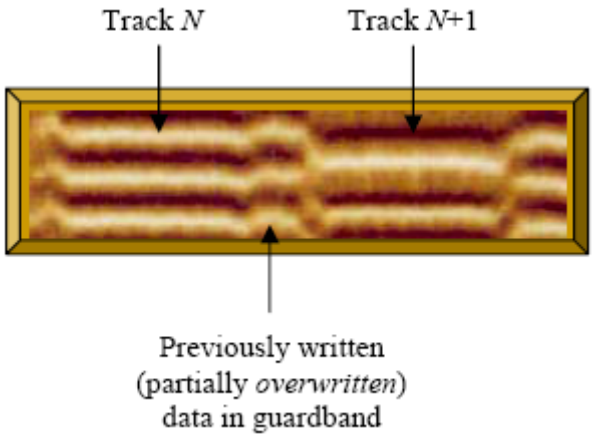


Figura 29 – Superfície magnética vista por um microscópio de força atômica
Fonte: SOBEY – Recovering Unrecoverable Data, 2004. p. 20.

Através da técnica *Magnetic Force Microscopy* (MFM) uma ponteira especial rastreia toda a superfície do disco catalogando os bits que compreendem os dados anteriores. Posteriormente, estes dados são avaliados para que sejam validados como informações (SOBEY, 2004).

Sobey (2004) afirma que mesmo com esta possibilidade teórica, ainda não viu comercialmente nenhuma empresa ou entidade praticar este tipo de método de forma viável. Além de cada dia, isto se tornar mais difícil devido ao aumento da

densidade de dados na superfície do disco e maior precisão de posicionamento das cabeças. Conseqüentemente, um alto custo de execução que torna inviável.

Isto consolida a facilidade da destruição de evidências com um todo, bem como a dificuldade de encontrá-las.

4.1.2 Wipe

As ferramentas de *wipe* foram criadas para destruir definitivamente os dados residentes em um disco devido à hipótese da recuperação de dados sobrescritos através da análise física. Esta ferramenta tem por finalidade sobrescrever o disco várias vezes com códigos binários “0s” “1s” ou algum pseudocódigo aleatório, com o propósito de ser uma ferramenta de deleção segura. Estes códigos são inspirados no trabalho de Gutmann (1996), que é um especialista em exclusão segura de dados, afirma que sobrescrever o disco 35 vezes com códigos aleatórios é o ideal.

Existem as mais diversas ferramentas de *wipe*, desde as proprietárias até as de código aberto, que podem sobrescrever o disco completamente ou algum dado específico. A Figura 30 mostra um exemplo de um formatador de disco que traz a opção de *wipe*, podendo escolher quantas vezes o disco pode ser sobrescrito. Esta ferramenta tem a mesma ação de um formatador de baixo nível, com a vantagem do *wipe*.

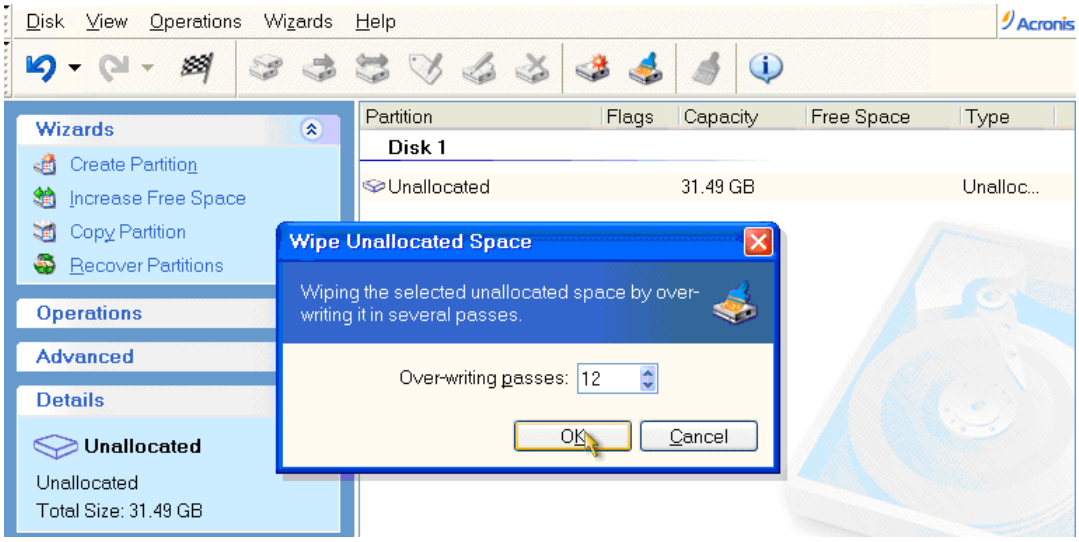


Figura 30 – Formatação de disco com wipe

Fonte: Software Acronis Disk Director Suíte

A Figura 31 mostra uma ferramenta que faz *wipe* de um arquivo específico no disco, sobrescrevendo os setores correspondentes ao arquivo. Lembrando que esta ferramenta é usada também pela perícia. Embora sendo uma excelente ferramenta, a perícia iria detectar que a mesma foi instalada no disco, porém, não iria saber o conteúdo do arquivo apagado, mas iria gerar indícios da prática anti-forense levando ao usuário ser suspeito.

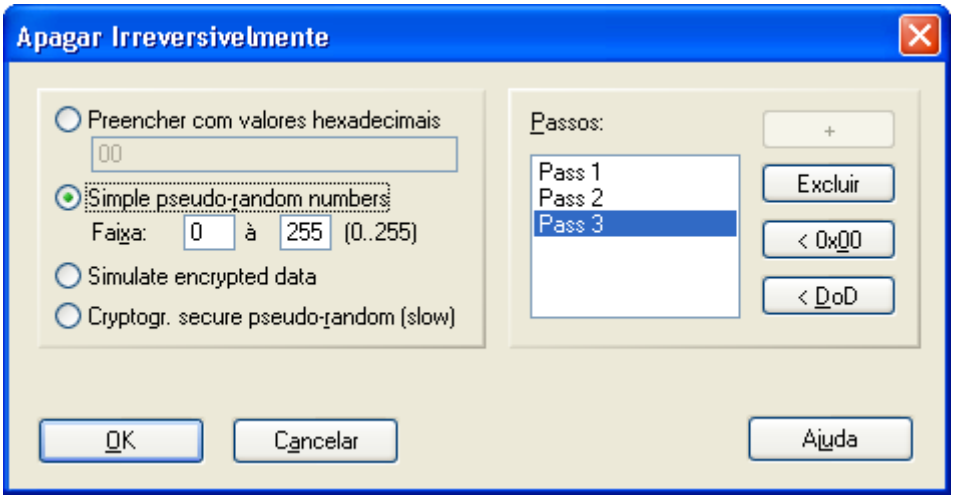


Figura 31 – Wipe de arquivos específicos com pseudocódigos randômicos
Fonte: Software WinHex

As duas ferramentas anteriores foram testadas através do *software* editor de disco, em que constatamos a sua eficiência dentro do esperado através das alterações dos binários.

Diante do exposto, podemos afirmar que as ferramentas de *wipe* são as mais seguras com relação à exclusão de arquivos. Mas também a simples ferramenta de formatação de baixo nível é eficiente, salvo considerarmos as variáveis de custo e tempo da perícia computacional afirmadas por Sobey (2004), que em nossa realidade, a recuperação de dados a este nível é impraticável.

4.1.3 Elementos externos

Os elementos externos como já foi citado no capítulo 1, é qualquer ação externa que evite o funcionamento do disco ou a destruição da mídia que contém os

dados. Dentro do que já foi exposto neste capítulo, qualquer ação que leve a perícia para uma análise física da mídia é válida no âmbito da anti-forense. No entanto, o disco pode ser destruído através da queima intencional de algum elemento semicondutor que está presente na placa controladora do disco. Fazendo com que o mesmo fique inacessível em qualquer computador. Mas, esta placa controladora pode ser consertada ou até mesmo ser trocada completamente por outra de um disco de modelo idêntico. Logo o melhor elemento externo para a destruição seria danificar a mídia literalmente ou desmagnetiza-la segundo algumas teorias. A Figura 33 mostra o chip principal de controle do HD pode ser queimado através de um curto-circuito em seus terminais através de uma chave de fenda com o disco ligado.

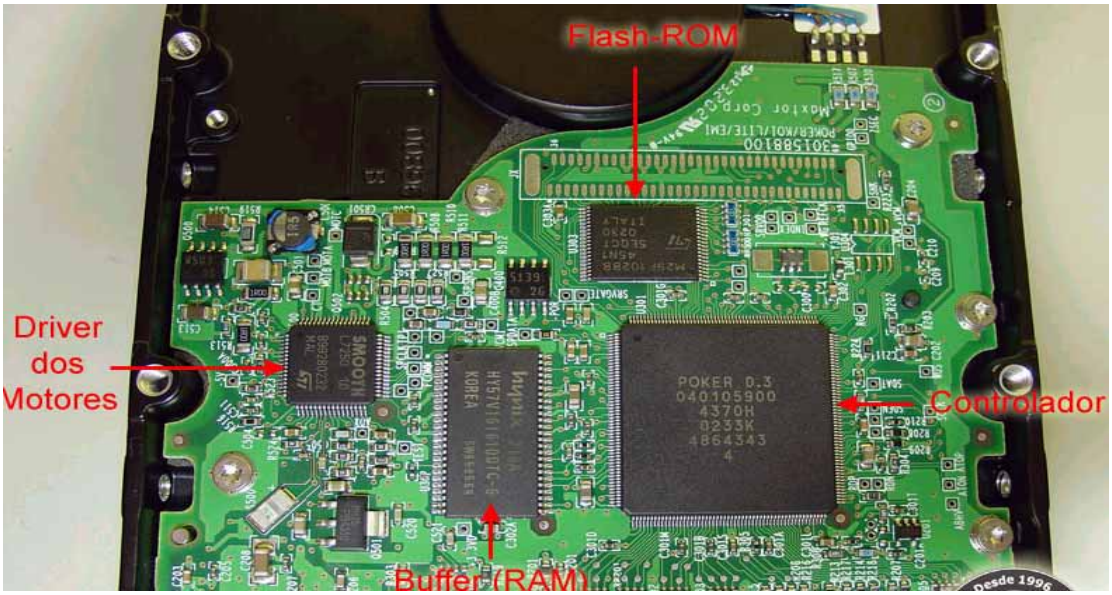


Figura 32 – Controlador do disco que deve ser curto-circuitado
Fonte: www.clubedo*hardware*.com.br

Dentro da técnica de desmagnetização, não fizemos nenhum experimento para comprovar tal função de acordo com que as teorias suponham. Segundo Guttman (1996) a desmagnetização se dá através da geração de um campo magnético para anular a força coercitiva residual dos dipolos do disco, que são as responsáveis por garantir os armazenamentos dos dados. Esta força residual existente nos dipolos é uma propriedade do material utilizado na camada superficial do disco.

Dentro da teoria de Guttman (1996) seria necessário um campo magnético na ordem de 2200 Oersted que seria produzido através de uma bobina enrolada ao redor do disco, para que então fosse alimentada por uma fonte de tensão e produzir o desejado campo magnético segundo a lei de Faraday. No entanto, mesmo com esta montagem para a possível desmagnetização, não se garante que todos os dipolos serão desmagnetizados (GUTTMAN, 1996). Ficando apenas a ultima alternativa da destruição literal do disco, que leva a um grande trabalho dispendioso. A Figura 33 mostra as disposições de dipolos magnetizados e desmagnetizados em um disco através da forca coercitiva.

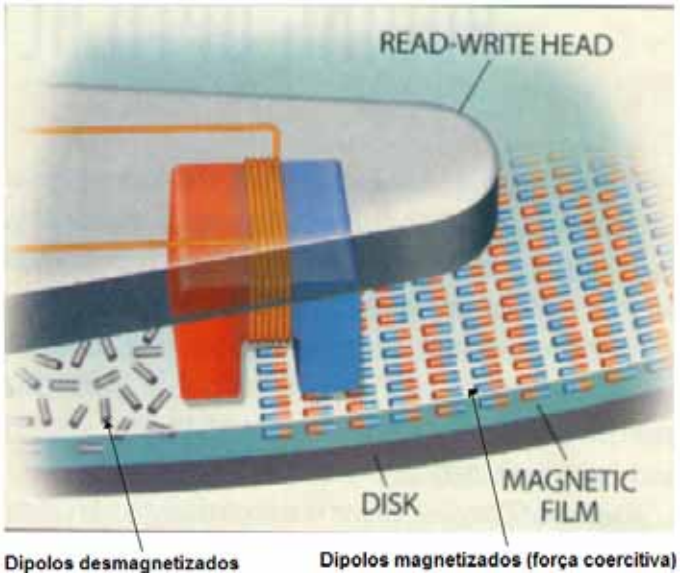


Figura 33 – Ação da força coercitiva nos dipolos do disco

Fonte: Desconhecida

Concluímos que dentro dos itens de destruição de evidências, a técnica de *wipe* se torna a mais eficiente mesmo perante aos elementos externos, devido à vantagem de garantir totalmente a destruição das evidências, bem como a sua facilidade de uso e por satisfazer a teoria de Guttman (1996).

Para inviabilizar completamente a perícia, as técnicas anti-forense podem ser trabalhadas em conjunto, ou seja, dentro do que já experimentamos, podemos combinar a ferramenta de *wipe* com o uso de elementos externos. Do mesmo jeito, funcionam as ferramentas de esteganografia e criptografia nas quais iremos demonstrar na próxima sessão, como uma das técnicas para a não destruição das evidências, mas sim, ocultá-las.

4.2 Ocultação de evidências

A ocultação de evidências se dá pela redução das chances de um sucesso de investigação preliminar, exigindo que a perícia seja mais minuciosa (HARRIS, 2006). Dependendo do tipo de ocultação, podem se levar anos para descobrir o que realmente estava ocultado ou até mesmo nunca descobrir.

Dentro das técnicas de ocultações, podemos citar a esteganografia, criptografia, ADS e *slack space*, nas quais serão demonstradas a seguir.

4.2.1 Esteganografia

Segundo Petri (2004) a esteganografia é uma arte de esconder informações dentro de uma outra que funciona como portadora, com o objetivo da comunicação em segredo.

Existem os mais diversos softwares e técnicas de esteganografia que vão desde as simples até as mais complexas, porém, este trabalho não irá tratar individualmente cada técnica e ação, mas irá mostrar a maneira mais eficiente de esteganografar uma informação.

Utilizamos à ferramenta *Puff* versão 1.01 que esconde qualquer tipo de informação em uma portadora de imagem ou áudio. Através de uma *passphrase* de 256 bits, a informação fica protegida dentro da portadora.

Para entendermos o funcionamento mais detalhado desta ferramenta, a Figura 34 ilustra as propriedades de um arquivo de imagem na qual será a portadora de nossa informação a ser ocultada. Através da ferramenta *Puff* v1.01 adicionamos no portador um arquivo TXT que contém uma frase secreta com capacidade de 26Bytes. A Figura 35 mostra o arquivo secreto sendo adicionado e criptografado por uma chave e algoritmo que não conhecemos. A partir da Figura 36 notamos que o arquivo portador aumentou de tamanho em 100%, embora o TXT ter apenas 26Bytes e a compressão da esteganografia ter sido máxima.

No entanto, a perícia poderia desconfiar que um arquivo JPG com 158kBytes nestas condições seria um portador de alguma informação. Com isso, através de técnicas de esteganálise seria possível a perícia descobrir que este arquivo possui alguma informação embutida.

Figura 34 – Imagem de arquivo e propriedades antes da esteganografia

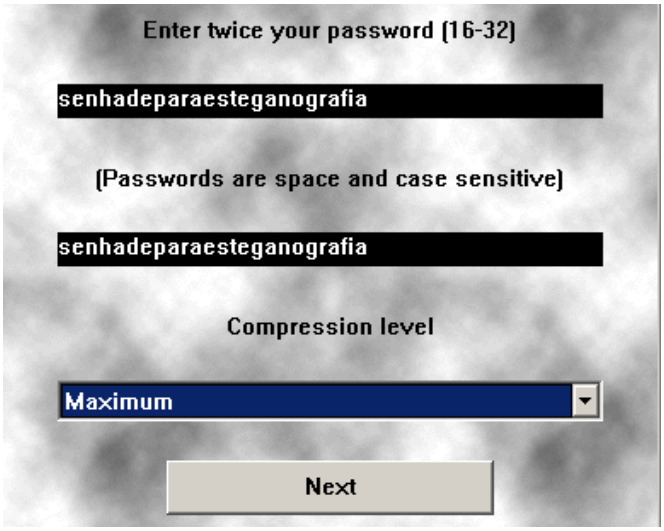


Figura 35 – Tela do software de esteganografia

Fonte: Software Puff v1.01

Figura 36 – Imagem de arquivo e propriedades depois da esteganografia

Para avaliarmos a possibilidade de descobrir se um arquivo possui mensagem oculta, utilizamos o software *Outguess* que detecta padrões de

esteganografia tais como JPEG-JSTEG, JP Hide&Seek e F5 (CARVALHO, 2005). A ferramenta forense *Autopsy* não possui nenhum recurso pra encontrar supostos arquivos esteganografados, por isso, optamos pelo *Outguess*. A partir da Figura 37 percebemos que o software não conseguiu detectar a mensagem oculta, embora termos calibrado a sensibilidade da ferramenta em comparação com um arquivo que não possui mensagem oculta para não gerar um falso positivo.

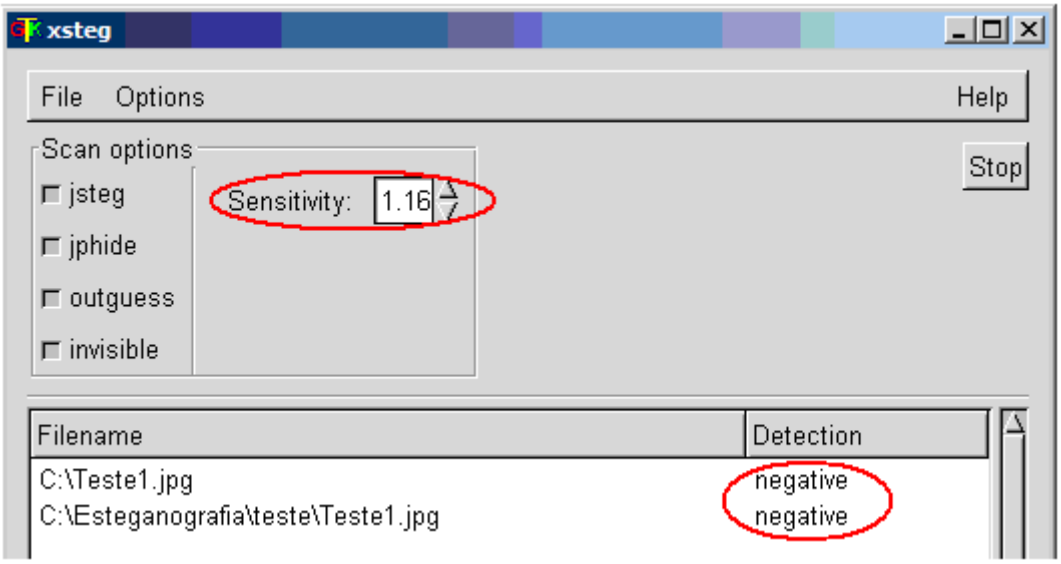


Figura 37 – Ferramenta de steganálise

Fonte: Software Outguess

Diante dos resultados, concluímos que a ferramenta de esteganografia *Puff* utiliza um algoritmo que não é utilizado pelo *Outguess*, com isso, o mesmo não conseguiu detectar a mensagem oculta. Mas partindo do pressuposto que a perícia iria desconfiar deste arquivo JPG pela sua capacidade, fizemos o mesmo experimento com um arquivo portador BMP. A Figura 38 mostra a comparação entre os dois BMPs antes e depois da esteganografia com a mesma mensagem oculta anterior de 26Bytes. Logo percebemos a vantagem de trabalhar com BMP uma vez que a capacidade não foi alterada. Com isso, entendemos que quanto maior o arquivo portador, mais difícil será para a perícia detectar.

Segundo Carvalho (2005) os algoritmos de esteganografia atuam dentro do bit menos significativo (LSB) do arquivo portador. Consequentemente, com a alteração destes bits, a capacidade total do arquivo não é mudada. Todavia, o arquivo portador tem que ter um tamanho bem superior ao que se quer ocultar, para

que exista uma quantidade suficiente de LSBs para acomodar a mensagem assim como não alterar a visualização da imagem de forma visível ao olho nu. Já para um arquivo portador JPG, não tivemos o mesmo efeito devido ao mesmo ser uma compressão do próprio BMP, logo, qualquer alteração dentro dos LSBs do JPG, poderia ser notado visualmente a diferença na imagem. Com isso, o algoritmo de esteganografia adiciona bits proporcionais ao que se quer ocultar em arquivos JPG.

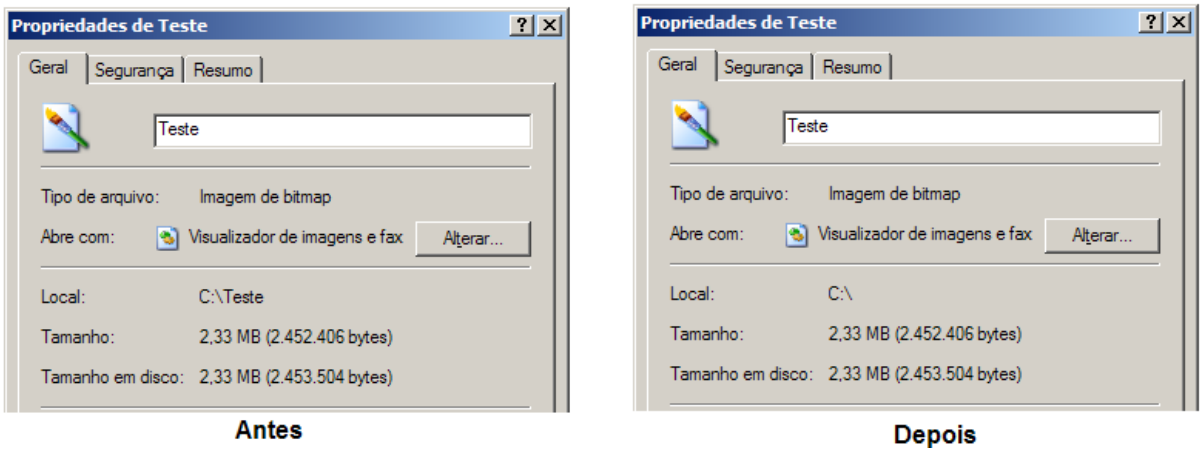


Figura 38 – Comparações dos arquivos BMP antes e depois da esteganografia

A Figura 39 mostra a comparação entre um trecho dos binários dos arquivos BMP com e sem esteganografia, nos quais podemos perceber claramente a alteração do bit LSB, dentro do Byte FF(16) = 1111111111111111(2) para FE(16) = 1111111111111110(2) seguindo uma entropia e freqüência de alterações que determina o algoritmo usado pelo software de esteganografia.

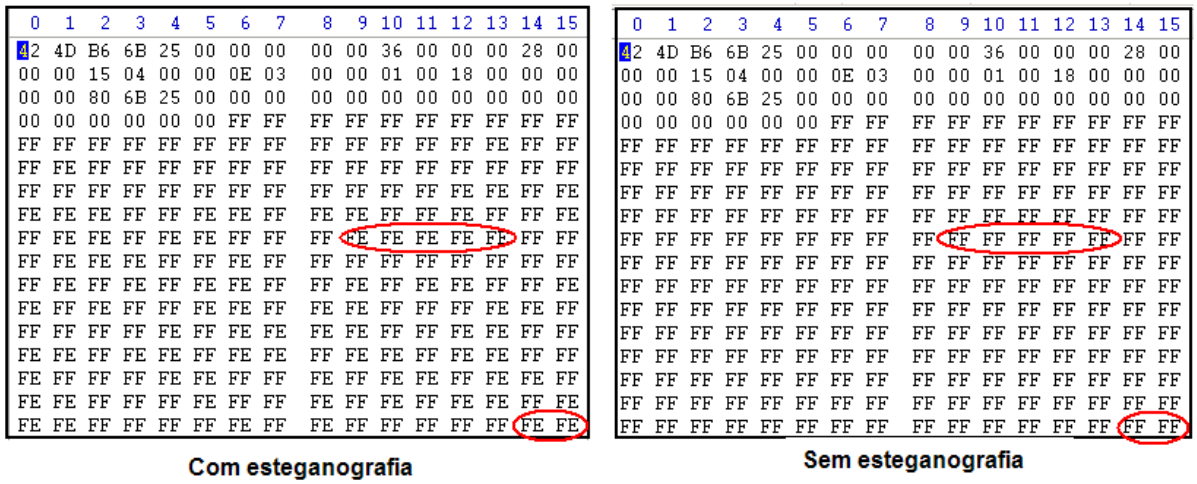


Figura 39 – Alterações dos binários LSB do arquivo com esteganografia

Através de ataques estatísticos nos quais determinam a entropia e freqüência de alterações dos LSBs do portador, é possível determinar o tamanho da mensagem oculta assim como dizer se um arquivo é portador (OLIVEIRA, 2007). Como foi citada anteriormente, a ferramenta *Outguess* executa estas estatísticas de forma automatizada, porém, não tivemos sucesso dentro do algoritmo usado pela ferramenta *Puff.*

Mesmo com a certeza de um arquivo ser portador, existe a dificuldade de extrair a mensagem oculta devido a sua compressão e criptografia envolvida, no entanto, não conseguimos nenhuma outra ferramenta que fizesse este trabalho com sucesso. Para a perícia realmente conseguir a informação oculta, seria necessário que a mesma descobrisse como a ferramenta *Puff* funciona em sua totalidade ou tentar quebrar por força bruta a *passphrase* de 256bits, ou então, descobri-la de alguma outra forma.

Sabendo que todos os processos realizados no computador estão em memória DRAM, verificamos através de um *dump* da própria memória pela ferramenta *Winhex*, se a *passphrase* fica armazenada em texto plano mesmo com o fechamento do programa *Puff*. A Figura 40 mostra à presença da *passphrase* decodificada em ASCII que foi utilizada na Figura 35 para esteganografar a mensagem da Figura 34. No entanto, através de uma forense *on-line*, seria possível que a perícia detectasse a *passphrase* se a mesma fosse feita antes que o sistema operacional não sobrescrevesse essa região de memória.

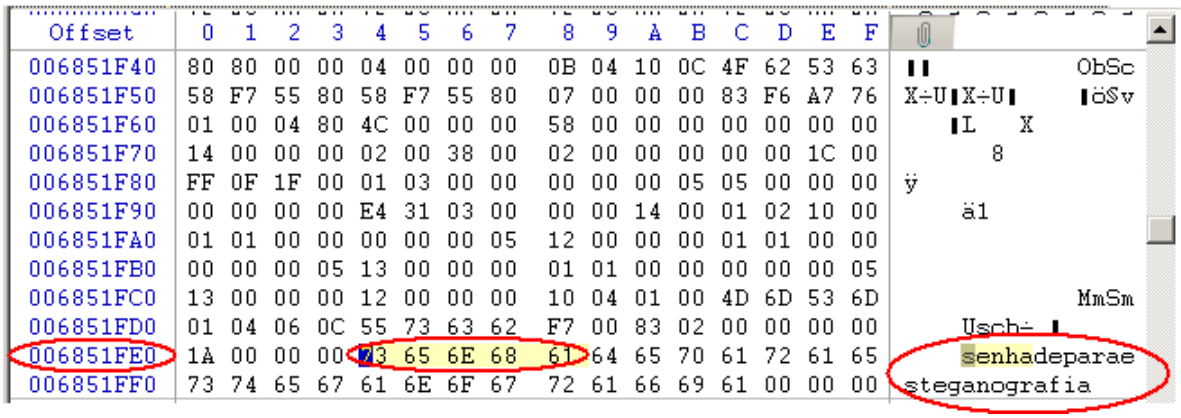


Figura 40 – Passphrase encontrada na memória DRAM

Fonte: Software Winhex

Para uma senha não perceptível facilmente, seria necessário que a *passphrase* fosse formada totalmente por caracteres especiais, evitando os da tabela ASCII tradicional. Com isso, realizamos um novo teste com uma senha de caracteres especiais (*&%@$#<>^) e logo após a operação, desligamos e religamos o computador para verificar se realmente a senha tinha sido volatilizada. E como era esperado dentro da teoria de funcionamento da memória DRAM, nada foi encontrado que comprovasse a *passphrase.*

Mas segundo alguns pesquisadores como Halderman et al (2008) afirmam que a memória DRAM ainda persiste com os seus dados por um tempo mesmo após serem desligadas. Os dados vão se dissipando gradualmente a cada segundo e não instantaneamente como estamos acostumados a ler nas literaturas. E quando as memórias são submetidas a temperaturas mais baixas, os dados são guardados por mais tempo, na ordem de minutos ou até horas. Gerando uma porta de vulnerabilidade para quebrar senhas de aplicações, assim como a esteganografia e criptografia estarão vulneráveis a um ataque físico desta natureza (HALDERMAN et al 2008).

A técnica proposta por Halderman et al (2008) consiste em congelar o módulo de memória ligado e levá-lo para um outro computador com sistema operacional personalizado para verificar o conteúdo da mesma e partir daí, conseguir decifrar as senhas. Esta monografia não irá abordar com detalhes este tipo de ataque, porém, estamos preocupados com esta nova modalidade assim como tentar sobressair desta, uma vez que pode ser usada pela perícia. Para maiores detalhes, consulte Halderman et al (2008) que está referenciado no final desta monografia.

Diante desta nova possível técnica de congelamento que a perícia pode utilizar, verificamos que esta somente é válida para a memória em funcionamento. Logo, após alguma possível ação de esteganografia ou de qualquer outra aplicação, devemos desligar o computador para que os dados devam ser totalmente volatilizados depois de certo tempo, no qual foi proposto por Halderman et al (2008). Para comprovar esta ação, podemos verificar através da Figura 41 que a *passphrase* anterior (senhadeparaesteganografia) não está mais carregada em memória após o sistema ter sido desligado e religado imediatamente.

Na pior das hipóteses, o módulo de memória pode ser trocado, assim como destruído ou descarregado eletrostaticamente através de papel alumínio envolvido no módulo e aterrado em alguma rede elétrica. Todavia o trabalho de Halderman et

al (2008) trouxe novas dimensões ao funcionamento da DRAM em que não conhecíamos, e que a partir daí, novos seguimentos de pesquisas irão surgir.

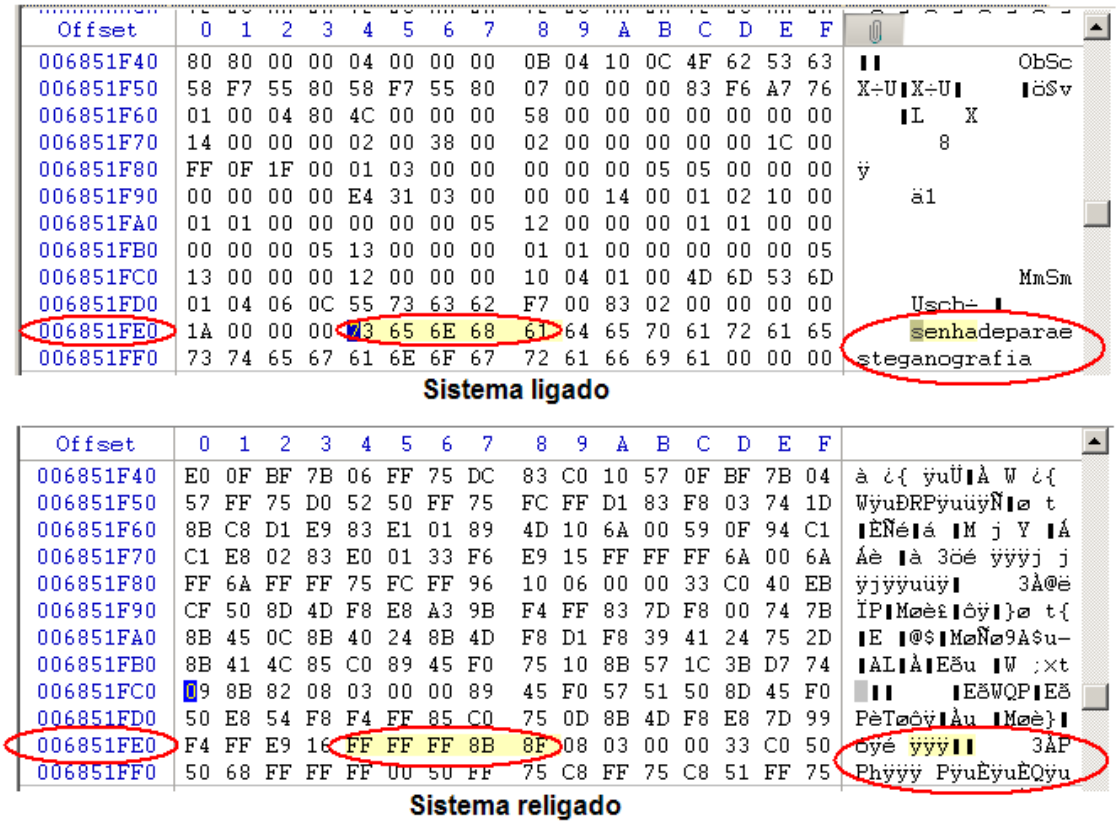


Figura 41 – Comparações do dump de memória antes e depois do sistema desligado

Fonte: Software Winhex

Mesmo com a possibilidade de a perícia extrair a mensagem esteganografada, Carvalho (2005) afirma que a possibilidade dessa extração diminui exponencialmente quando é utilizado um arquivo de vídeo como portador. Como o vídeo é uma sucessão de *frames*, então a complexidade aumenta exponencialmente de acordo com o número de *frames* de um vídeo. Utilizando o exemplo proposto por Carvalho (2005), o mesmo afirma que a probabilidade de extrair a informação oculta é de $1.76884.10^{-168}$ para um vídeo de duração de 4 segundos resultando em 120 quadros, na condição que saibamos o algoritmo de esteganografia utilizado. No entanto, para esta técnica de esteganografia em vídeo, fica extremamente difícil a perícia extrair a informação, bem como se revela o método mais seguro de ocultar informações dentro da esteganografia.

Infelizmente não conseguimos nenhuma ferramenta de esteganografia em vídeo, porém, seguindo o mesmo raciocínio, podemos dizer que a esteganografia

em áudio é mais segura se comparada a uma simples imagem. A mesma ferramenta *Puff* possui a opção de arquivos de áudio, que trabalha em cima dos LSBs, evitando que o tamanho do arquivo portador seja alterado bem como uma percepção auditiva de ruído depositado pelas alterações dos LSBs.

4.2.2 Criptografia

Segundo Takagi (2003) a criptografia é um estudo matemático de métodos para cifrar e decifrar uma mensagem para que apenas o destinatário consiga interpretar o conteúdo. Existem os mais diversos métodos e técnicas que atualmente são bastante complexas, que vão desde as criptografias assimétricas até as quânticas. Diante disto, o foco deste trabalho é mostrar uma maneira mais simples de aplicar uma criptografia segura e verificar sua eficiência perante a forense.

A criptografia é considerada um dos métodos mais seguros, logo, esta pode ser também considerada a melhor técnica anti-forense para a ocultação de evidências. Pois mesmo diante de uma perícia, a quebra da criptografia pode se levar anos dependendo da chave, algoritmo de criptografia e do poder computacional utilizado. Ou seja, dependem diretamente dos fatores tempo x processamento, uma vez que os algoritmos de criptografia utilizados atualmente requerem uma grande demanda de processamento. Com isso, um arquivo criptografado pode ser facilmente detectado pela perícia, mas o tempo para decifrá-lo pode inviabilizar a perícia.

A evolução dos algoritmos criptográficos se dá diretamente pela demora de sua quebra através de criptoanálise ou força bruta, que em teoria, toda a criptografia pode ser quebrada, desde que se tenha poder computacional suficiente para isso (NOGUEIRA, 2008).

Segundo Nogueira (2008) uma máquina com freqüência de processamento de 1600MHz gastaria 1713698 anos para quebrar uma chave criptográfica de 56bits por foca bruta. Todavia, já existem chaves de até 2048bits, que reforça a citação de Harris (2006) em que afirma que técnicas anti-forense se aproveitam das limitações de *software* e *hardware*, que consequentemente, correspondem diretamente à técnica de criptografia.

Uma vulnerabilidade para a criptografia foi apresentada por Halderman et al (2008) através da sua técnica de congelamento da memória DRAM. Como foi mostrada anteriormente, esta técnica permite trabalhar com a presença dos binários da memória DRAM para que então sejam utilizados ataques para decifrar possíveis chaves utilizadas na criptografia. Mas este método não tem mais sentindo quando o sistema é desligado e esperado o tempo de volatilização necessário dos dados da memória DRAM. No entanto, as únicas maneiras de quebrar a criptografia são através de criptoanálise ou força bruta.

Dentro das mais consagradas ferramentas de criptografia, usamos a *Truecrypt* que está livremente distribuída na internet *(open source)* além de ser uma excelente ferramenta. Funciona basicamente através da criação de um volume lógico no qual pode ser inserido qualquer tipo de dado para ser criptografado. Esta ferramenta possui inúmeros algoritmos criptográficos, nas quais são protegidos por uma *passphrase* ou chave de até 256bits. O volume lógico pode ser um arquivo ou uma partição inteira de qualquer memória de massa residente no computador.

A Figura 42 mostra uma unidade lógica de 1MByte sendo criada pela ferramenta *Truecrypt* e associada a um arquivo.

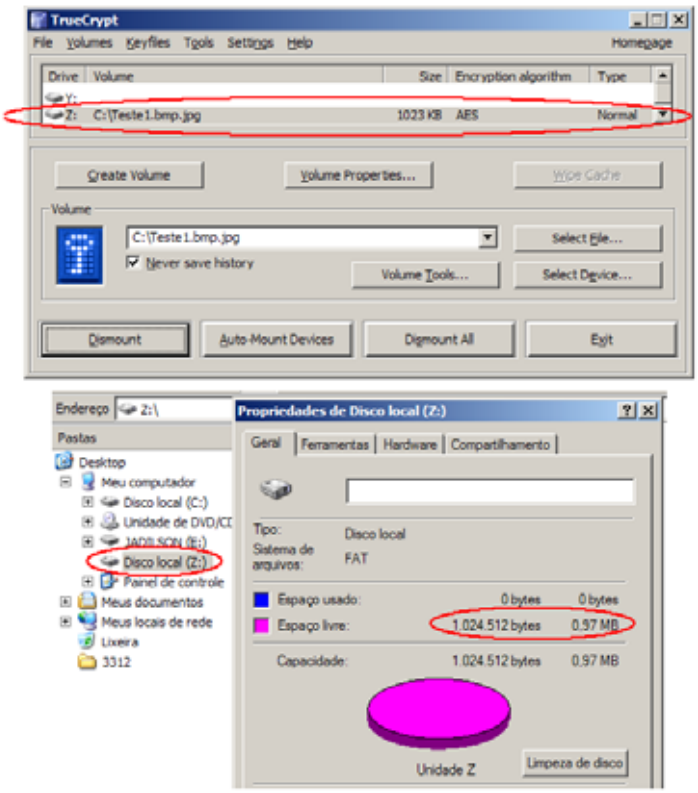


Figura 42 – Criação de volume lógico para criptografia

Fonte: Ferramenta Truecrypt

Após a criação do volume lógico, verificamos o dump da memória DRAM para certificar se a chave utilizada na criptografia poderia estar em texto plano no qual foi encontrado com a ferramenta de esteganografia *Puff*. Logo, nada concreto foi encontrado, uma vez que a possível chave ou os seus mecanismos estavam todos criptografados de forma que não entendemos. Diante disso, a ferramenta *Truecrypt* se mostra segura neste ponto, salvo, as condições propostas por Halderman et al (2008).

Podemos concluir que a criptografia pode inviabilizar o trabalho da perícia desde que sejam utilizadas grandes chaves criptográficas como a partir de 128bits e que o sistema seja desligado após qualquer criptografia para volatilizar as chaves na DRAM. Infelizmente por falta de poder computacional, não conseguimos quebrar nenhuma criptografia proposta pela ferramenta *Truecrypt.*

4.2.3 Alternate Data Stream – ADS

É um mecanismo nativo de sistemas de arquivos NTFS de qualquer sistema operacional que o utilize. Este mecanismo faz com que um arquivo seja embutido dentro de outro, de forma que o arquivo original não tenha o seu tamanho alterado (OLIVEIRA, 2002). Na verdade, para cada arquivo no sistema NTFS, existe um outro arquivo em branco sem nome *(unnamed stream)* que pode ser usado como um *stream* do arquivo original. Ou seja, um arquivo legítimo relacionado a um arquivo *stream* com nome e conteúdos diferentes, que é chamado de *alternate data streams* (ADS) (OLIVEIRA, 2002).

Os ADSs são invisíveis para as ferramentas de exploração convencionais, no entanto, esta é a grande motivação para esconder informações ou programas maliciosos como *trojans* que podem ser executados nos sistemas. Mas através de softwares forenses ou específicos, estes *streams* são facilmente visíveis.

Esta técnica é bem problemática devido a não precisar instalar nenhum tipo de software específico, pois já vem nativo do sistema além de ser legítimo, bastando apenas a console de linha de comandos para executar a ação.

Inicialmente o ADS foi tratado como falha de segurança em meados de julho de 1998 e que se faz presente até hoje nos sistemas Win2003, XP e VISTA, e foi

criado para que houvesse compatibilidade entre compartilhamento de arquivos entre os sistemas da Microsoft e MAC. (OLIVEIRA, 2002).

A Figura 43 mostra os comandos de criação de uma *stream* juntamente com um arquivo TXT válido. Na verdade são criados dois arquivos TXT, um com o nome “jadilson.txt” e outro como *stream* de nome “secreto”.

Para editar o arquivo “secreto” basta apenas utilizar o comando da Figura 43, que abrirá o *notepad* para a edição da *stream* “secreto”, ficando o arquivo “jadilson.txt” inalterado.

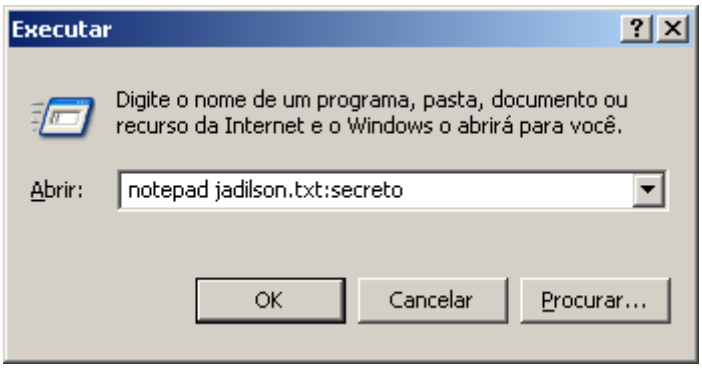


Figura 43 – Criação da alternate data streams

Depois de criado o arquivo juntamente com a *stream*, podemos perceber na Figura 44 que o sistema apresenta apenas o arquivo “jadilson.txt” com 0Bytes de espaço, uma vez que não foi editado, e sua *stream* permanece invisível para o usuário. Porém o arquivo “jadilson.txt” pode ser editado naturalmente sem que altere os dados armazenados em sua *stream,* logo, os conteúdos desses dois arquivos não estão relacionados.

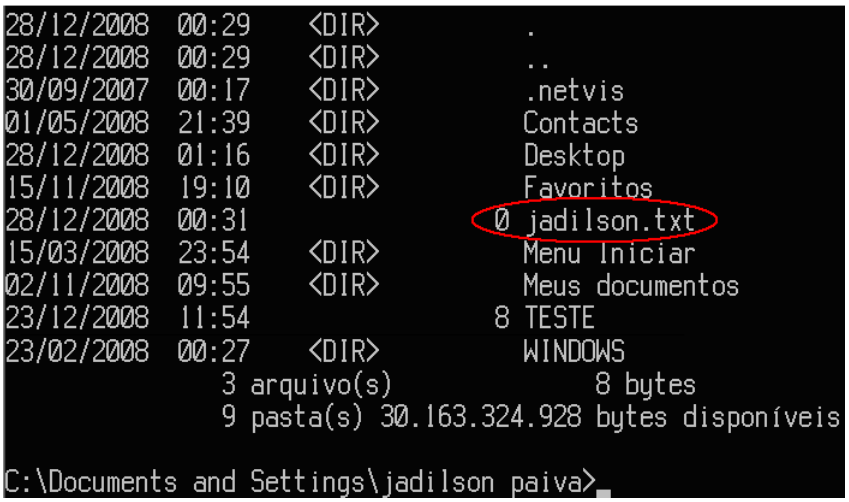


Figura 44 – Alternate data streams não visualizado

Fonte: Console de comandos

Partindo desta mesma filosofia, é possível criar também uma *stream* com um arquivo executável relacionado com um arquivo TXT, e partir daí, o mesmo pode ser executado. Este é um bom recurso para esconder alguns programas maliciosos. A Figura 45 mostra o comando para a criação de uma *stream* executável do programa *netcat* que estará relacionado com o arquivo teste.doc.

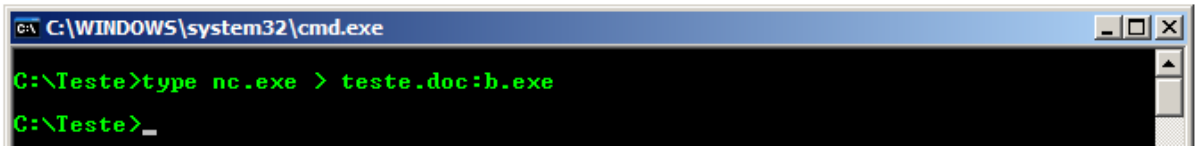


Figura 45 – Criando uma stream executável

Fonte: Console de comandos

Do mesmo jeito que foi feito anteriormente, o sistema só apresenta o arquivo “teste.doc” com a sua *stream* “b.exe” invisível. A Figura 46 mostra o comando para executar a *stream*.

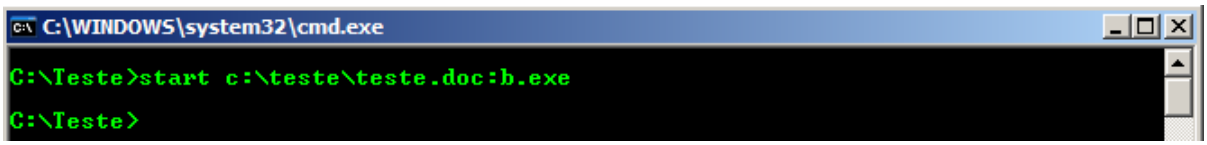


Figura 46 – Executando uma stream

Fonte: Console de comandos

Para demonstramos que mesmo com a invisibilidade do *stream* pelo sistema operacional, esta técnica é bastante fácil de ser detectada pela perícia. A Figura 47 mostra através da ferramenta *Autopsy* a *stream* da Figura 44 que era invisível para o usuário.

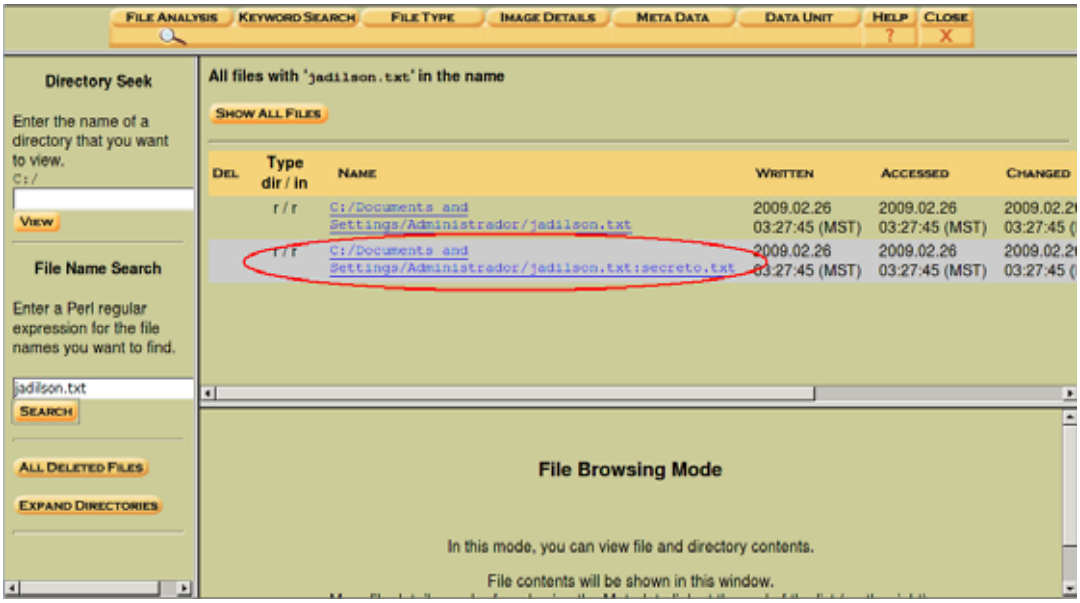


Figura 47 – Stream detectado pelo Autopsy

Fonte: Ferramenta Autopsy

Embora a ferramenta *Autopsy* ter detectado a *stream*, o processo não é automatizado, requerendo maiores atenções e busca por parte do perito. No entanto, a ferramenta *lns* que é específica para este fim, é bem mais prática. A Figura 48 mostra a ação desta ferramenta.

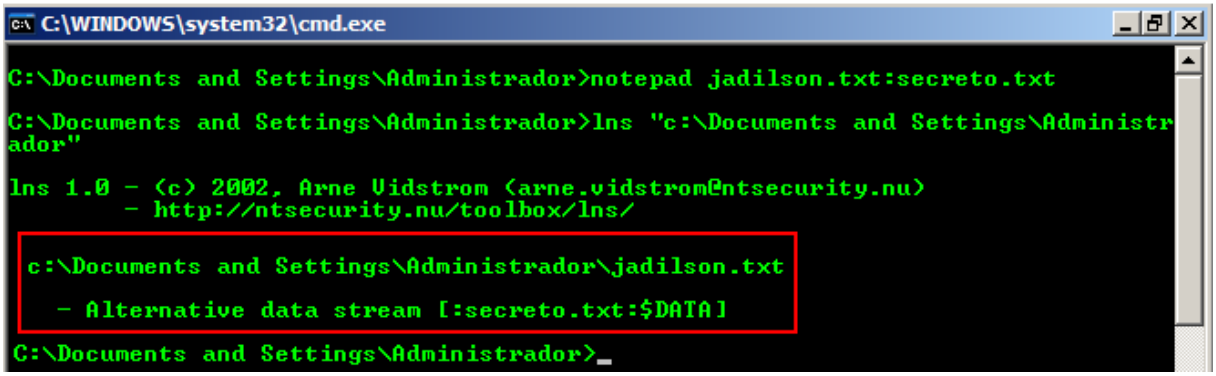


Figura 48 – Stream detectado pela ferramenta lns 1.0

Fonte: Console de comandos

Diante do exposto, verificamos que a técnica de ADS para ser usada como anti-forense só é válida para ser tratada entre usuários, no entanto, não deve ser usada permanentemente para a ocultação de informações, uma vez que é facilmente detectável pela perícia.

4.2.4 Slack Space

Segundo Silva (2003) o *slack space* é o espaço em disco entre o final e o início de um grupo de *clusters* que compõe um arquivo. Definimos também que são espaços em disco em que não são gerenciáveis pelo sistema de arquivos, ou seja, setores que não possuem *flags* relacionadas com a TAA do sistema de arquivos. Esta técnica aproveita-se dos inúmeros setores sem alocação existentes no disco, que já começam posicionados antes do sistema de arquivos, no qual foi mencionado na seção 3.2.1 e espalhados entre o final de cada arquivo ou a cada partição existente. A Figura 49 mostra a formação de um *slack space*, e que geralmente se faz presente também ao final de cada arquivo no disco, uma vez que o espaço ocupado pelo arquivo não consegue preencher todos os clusters envolvidos.

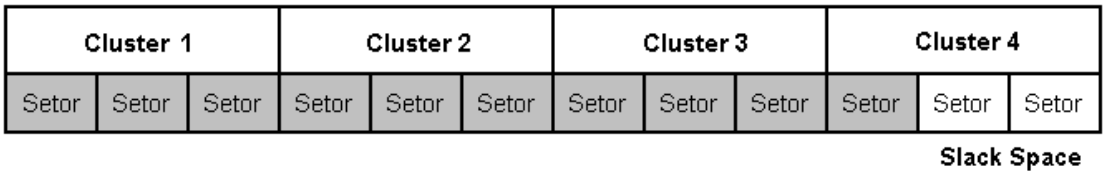


Figura 49 – Formação de slack spaces

A técnica de armazenamento de dados nos *slack spaces* como ocultação de informações é a mais famosa entre as técnicas anti-forense, uma vez que necessita de meios diretos para a comunicação com o disco assim como não é facilmente visível para o perito em uma análise posterior.

As ferramentas forenses não detectarem diretamente os *slacks*, pois estas trabalham de forma automatizada na camada do sistema de arquivos passando por cima deles. Todavia a perícia tem que recorrer às ferramentas editoras de disco, nas quais vão ao nível mais baixo do disco rígido, gerando maiores trabalhos para a exploração das possíveis informações, e que às vezes podem ser sem sucesso.

Para acessar os *slacks* ou qualquer setor do disco sem a dependência do sistema de arquivos, é necessário que o mesmo seja feito através de rotinas de baixo nível que acesse o disco diretamente sem intermediários. Através de um disco de *boot* em sistema de arquivos FAT32 e com a ferramenta nativa do DOS o *debug*, conseguimos através das linhas de comando em *assembly* mostradas na Figura 50 acessar o primeiro setor do disco que corresponde ao MBR apresentado na Figura 51.

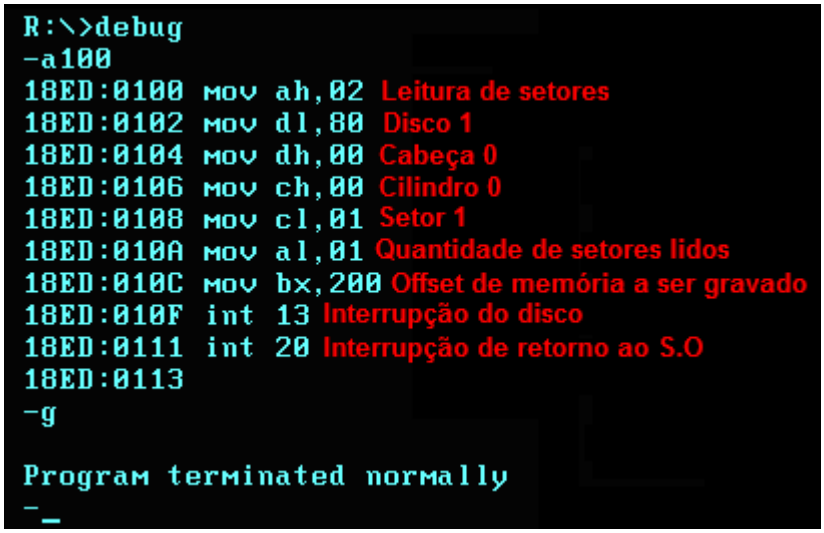


Figura 50 – Comandos assembly de acesso ao disco

Fonte: Debug do DOS

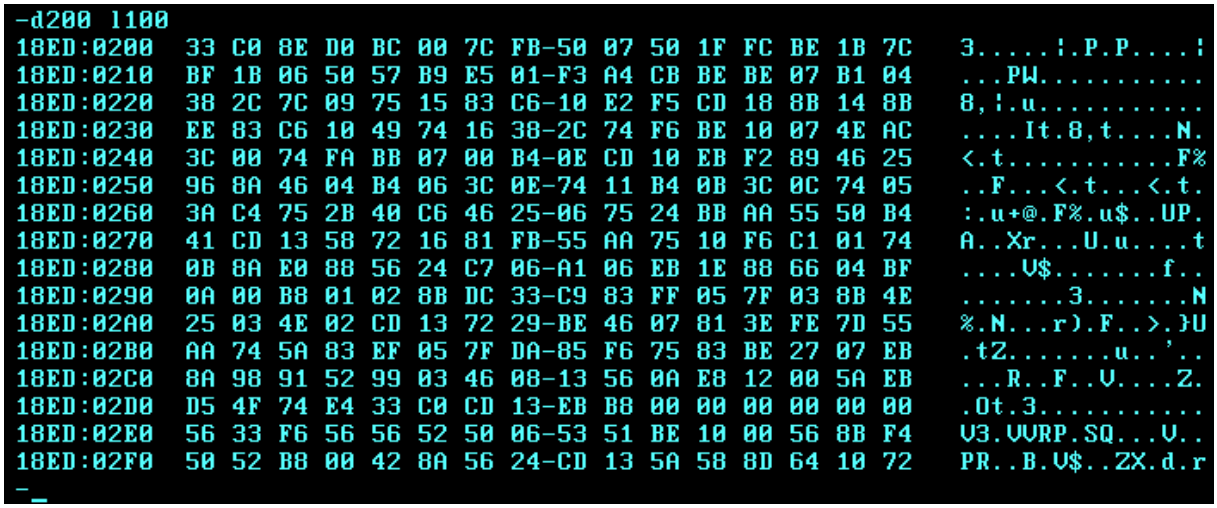


Figura 51 – Visualização do MBR através do dump da DRAM

Fonte: Debug do DOS

A partir destas figuras, podemos entender claramente a independência do *assembly* com relação aos sistemas de arquivos, uma vez que o sistema empregado

no disco é NTFS, e o boot carregado é FAT32. Consequentemente, o FAT32 não monta discos NTFS, mas por *assembly,* foi possível acessa-lo. Estes comandos não podem ser executados na console de comandos do Windows XP devido ao mesmo bloquear esta forma de acesso ao disco, por isso, que esta técnica só foi realizada através de um disco de *boot* em DOS.

Para demonstrar a técnica de *slack space*, inserimos um pequeno programa de *wipe* de 2,3kBytes a partir do segundo setor do disco, onde já sabemos que o mesmo não é alocado pela TAA. Como este programa é de 2,3kBytes e cada setor do disco é de 512Bytes, então o mesmo ocupará exatamente 5 setores começando a partir do setor 2 e indo até o setor 6.

Primeiramente, com o programa *wipe* contido já no *boot* do FAT32, carregamos o seu código binário para a DRAM, para que depois seja adicionado no *slack*, conforme mostram as Figuras 52 e 53.

Mesmo com este programa embutido, nenhuma ferramenta de anti-virus funcionando ao nível de sistema de arquivos conseguiu detecta-lo, devido ao mesmo não ser mapeado pela TAA. E também mesmo após a uma formatação lógica, este programa ainda persistia no disco, devido à rotina desta formatação ser apenas a sobrescrita do sistema de arquivos. No entanto com o programa embutido, o mesmo pode ser executado ou trazido para a camada do sistema de arquivos. A Figura 54 mostra o programa *wipe* gravado a partir do segundo setor do disco.

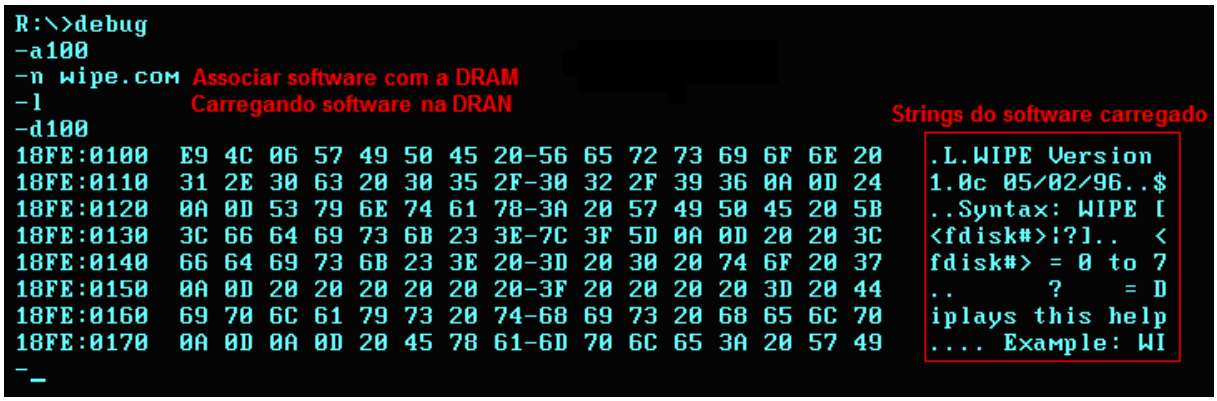


Figura 52 – Carregando um código de programa na DRAM

Fonte: Debug do DOS

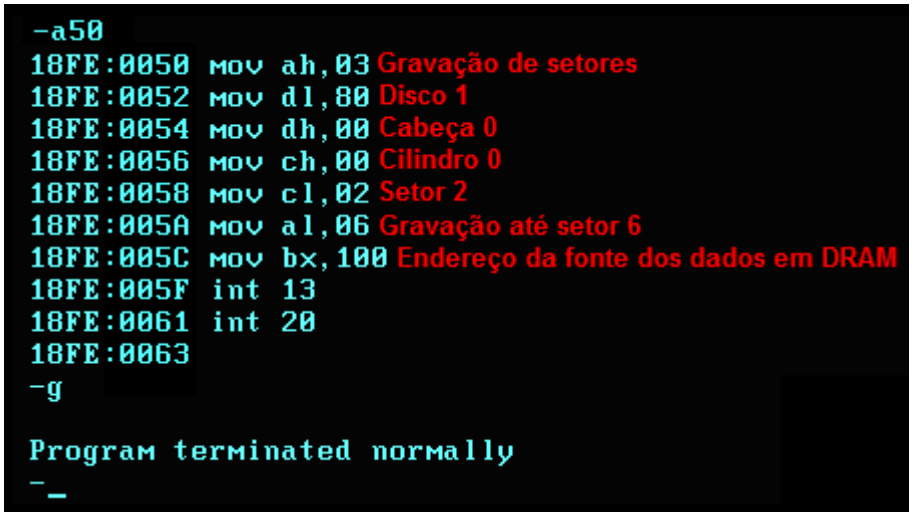


Figura 53 – Gravando um código de programa da DRAM para o disco

Fonte: Debug do DOS

Embora o *debug* ter permitido a inclusão de dados em *slack spaces*, este código permite apenas o acesso ao disco até 8GB de capacidade, devido a razões de limitações de código por ainda trabalhar em modo anterior ao *logical block access* (LBA), ou seja, modo não extendido.

Infelizmente, não conseguimos desenvolver nenhum código em *assembly* que desse suporte a toda a capacidade do disco, devido a maior complexidade deste ser trabalhada em modo extendido. Este novo modo de comunicação faz com que a tradução do disco seja feita sequencialmente de setor a setor, sem uma dependência das características de cabeças, cilindros e setores. Sendo esta tradução conhecida como LBA.

Todavia, com a pequena capacidade de acesso ao disco, o código realizado em *assembly* é eficiente, porém, bastante trabalhoso para a prática desta técnica e mais complexo ainda seria trabalhar no modo LBA. No entanto, para ter acesso total ao disco e incluir dados em qualquer local, usamos a mesma ferramenta que é usada pela perícia, ou seja, à ferramenta editora de disco, que já foi utilizada anteriormente em outras seções desta monografia. Esta ferramenta tem a grande vantagem de ser totalmente automatizada, permitindo a alteração mínima de até 1 bit em um disco inteiro, de uma maneira muito mais fácil em relação aos códigos em *assembly*.

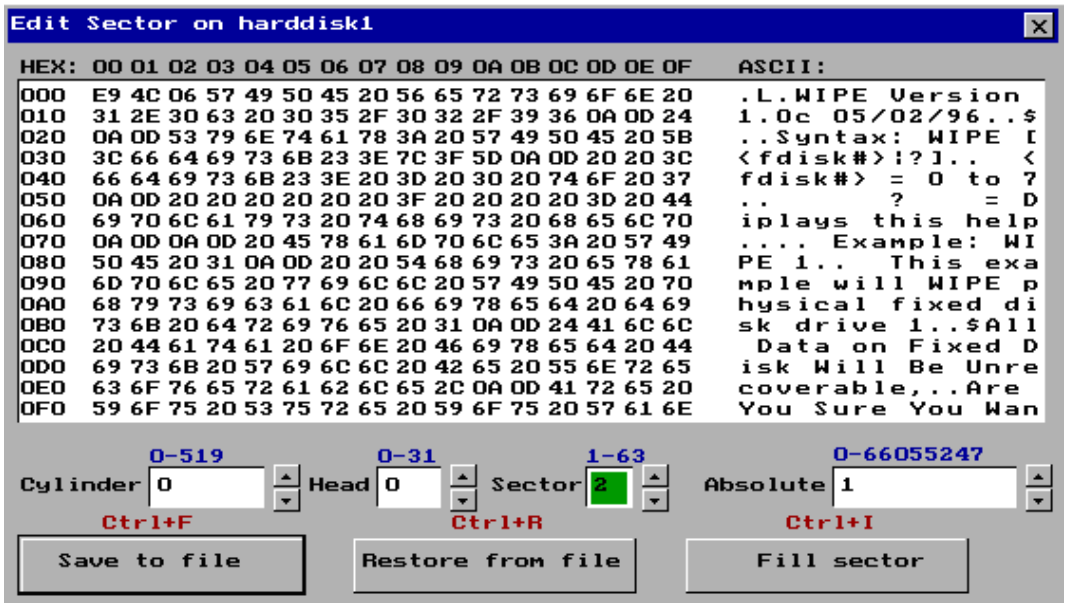


Figura 54 – Slack space com binários gravados

Fonte: Software Partition Table Doctor

Com a ferramenta de edição do disco, o atacante pode modificar quaisquer dados existentes, assim como o MBR, de forma a mapear partições inexistentes, corromper sistemas de arquivos ou qualquer outra ação que tente despistar a perícia. A Figura 55 mostra que através desta ferramenta, podemos facilmente corromper o MBR do disco simplesmente alterando um 1Byte. Fazendo com que o uso normal das ferramentas forenses não consiga montar o disco como unidade lógica. Logo, os peritos têm que recorrer a outros softwares que podem recompor o MBR em função dos binários encontrados no disco, ou dependendo do nível de corrompimento dos binários, a perícia tem que verificar setor a setor os binários a procura de irregularidades. Gerando maiores trabalhos para a perícia.

Com isso, acabamos de reforçar uma das citações de Harris (2006) que afirma que as mesmas ferramentas utilizadas pela perícia são utilizadas também para a prática da anti-forense.

A partir do próprio editor, podem ser inseridas também as maiores variedades de códigos bem como mensagens a serem ocultadas, e para piorar ainda a situação, o atacante pode inserir códigos ou mensagens criptografadas. Portanto, a perícia vai ter que vasculhar todo o disco a procura destes slack *spaces* e talvez sem chances de sucesso, devido aos códigos estarem criptografados. Uma vez que a perícia pode interpretá-los como lixo de formatações anteriores.

Uma outra dificuldade que a perícia pode encontrar com os slack spaces, é a fragmentação, pois o atacante pode esconder essas informações de forma não seqüencial, logo, códigos ou mensagens podem ser espalhadas entre setores aleatórios do disco.

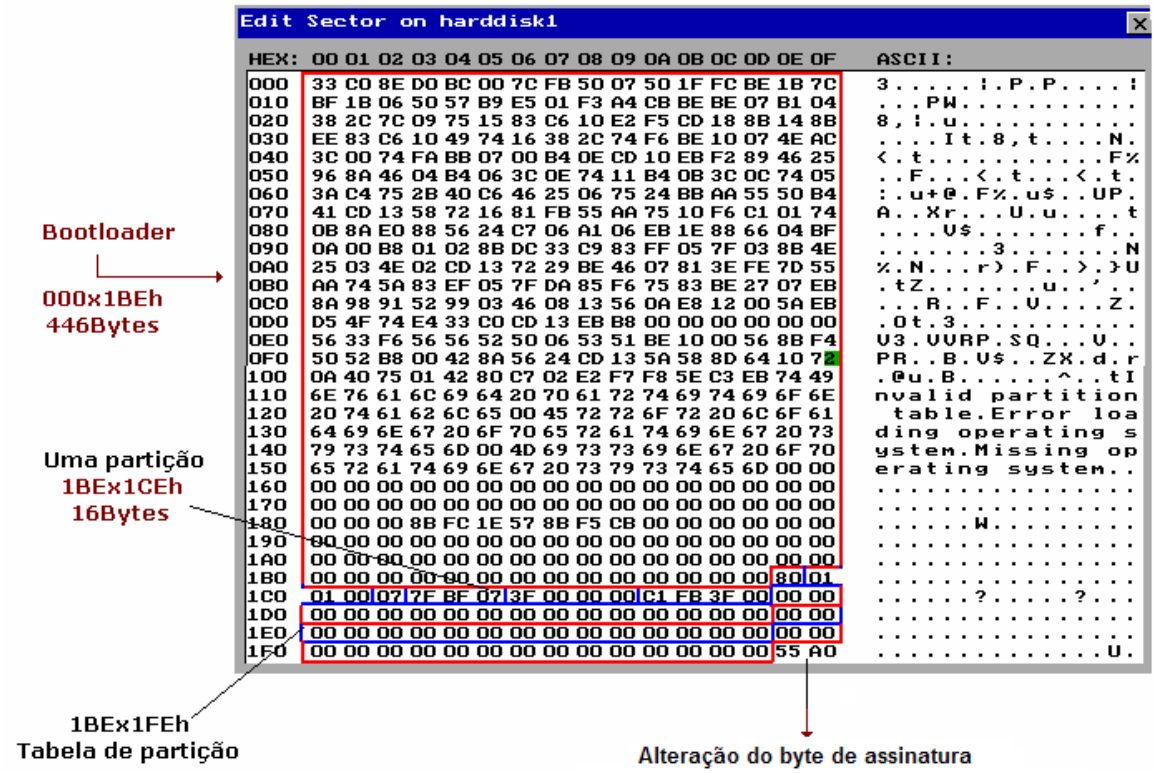


Figura 55 – Alteração de um Byte do MBR

Fonte: Software Partition Table Doctor

Dentro das técnicas anti-forense, existe a possibilidade de não deixar rastros na mídia, no entanto, a tarefa deve ser realizada somente em memória DRAM, em que chamamos de método de eliminação de fontes de evidências.

4.3 Eliminação das fontes de evidências

Segundo Harris (2006) a eliminação das evidências se dá através da utilização de métodos para que as evidências não sejam criadas, mas o principio de Locard diz que sempre existirão evidências em algum local do crime. Logo, demonstramos que sempre existirão evidências, mas de acordo com o método utilizado, estas serão apenas armazenadas em memória DRAM ou ROM da placa

controladora do disco. Logo, o melhor método para a não geração de evidências em disco (mídia) - que corresponde ao maior percentual de coleta de evidências em uma perícia, é o uso de live CDs.

Mesmo com dispositivos portáteis como HDs externos ou *pen drivers*, existirão algumas chamadas no sistema operacional que revelam a ação do uso destas memórias. Ficando registradas no disco local e dando brecha para a perícia investigar. A partir da Figura 56, demonstramos que o uso de qualquer memória externa como método de não deixar evidências não é eficiente, pois as chamadas do dispositivo ficam no disco local e são facilmente visíveis pela perícia através do *timeline.* Embora existirem as chamadas do sistema, não haverá o arquivo propriamente dito, mas a perícia vai entender que houve mídias removíveis na ação.

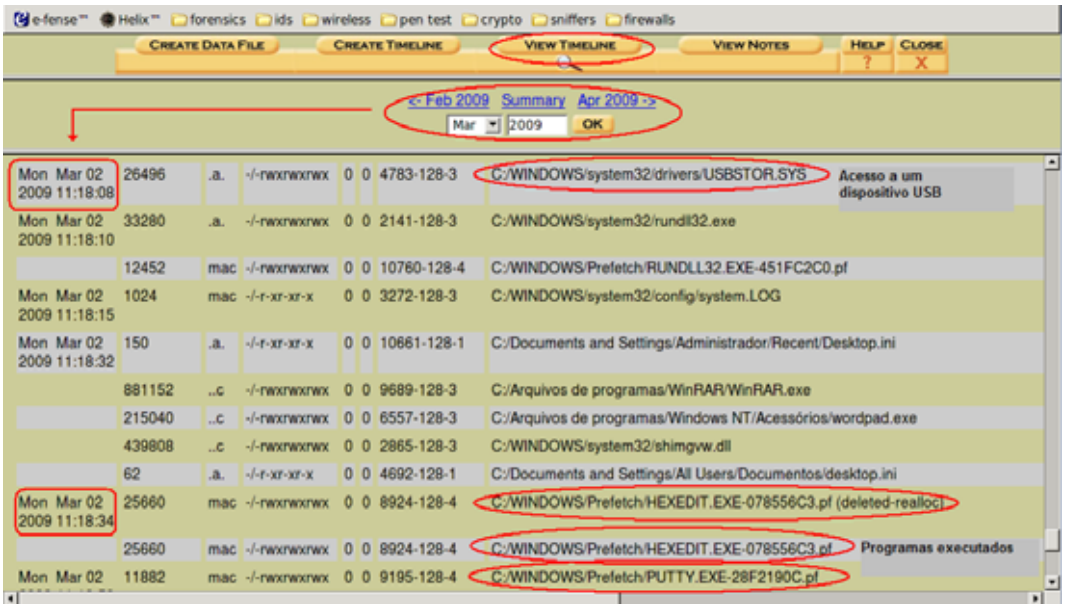


Figura 56 – Detecção do uso de memórias externas

Fonte: Software Partition Table Doctor

Como melhor alternativa, utilizamos o próprio *live* CD do *Helix*, em que demonstramos deste modo, a geração de evidências apenas em memória DRAM ou ROM da placa controladora do disco. Preservando todos os binários da mídia sem nenhuma alteração.

Para demonstrar este fato, inicialmente carregamos o live CD do *Helix* e utilizamos a ferramenta para calcular o *hash* do disco. Após esta operação, recalculamos mais duas vezes para certificar que o próprio live CD do *Helix* não alterava nenhum binário do disco. A Figura 57 mostra o *hash* calculado.

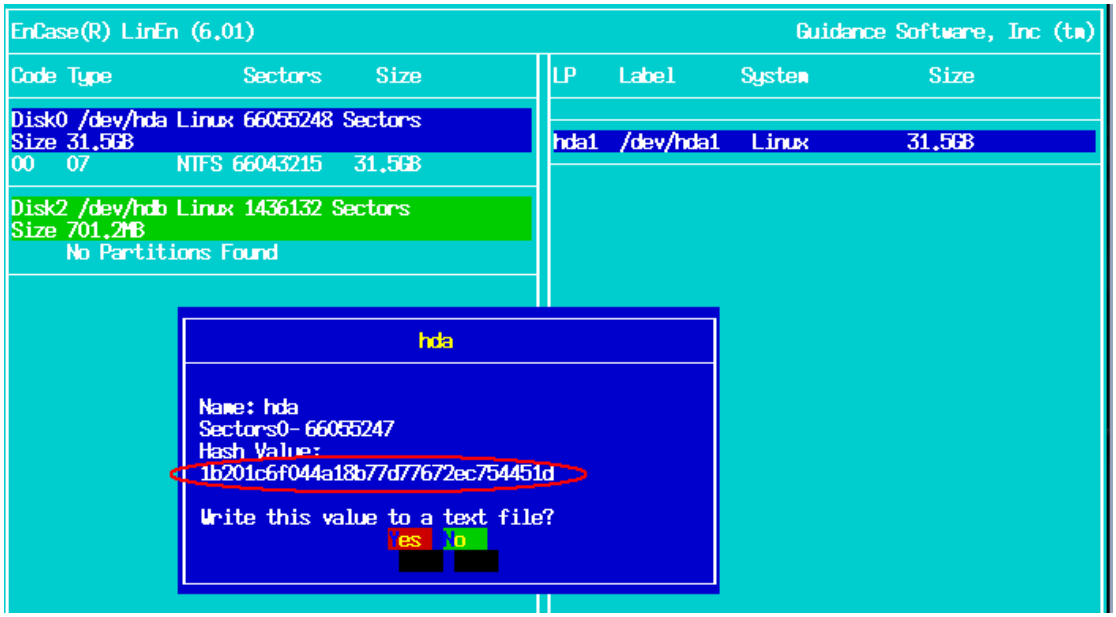


Figura 57 – Cálculo do hash no disco

Fonte: Software Linen

A partir do *hash* calculado, fizemos inúmeras ações, desde consultas na internet até *downloads* de fotos que ficaram armazenadas na unidade virtual criada pelo *live* CD. Posteriormente, recalculamos o *hash* novamente e vimos que este foi igual ao anterior, comprovando que o *live* CD do *Helix* realmente trabalha com o disco em modo apenas leitura.

Com isso, a técnica de eliminação de evidências na mídia com uso de *live* CDs é imune a uma forense *off-line.* E para verificar este fato diante do nosso CD de *boot* em FAT32 em que usamos para fazer o acesso ao *slack space*, verificamos também que pelo motivo do sistema FAT32 não montar partições NTFS, o disco fica inalterado. Consequentemente, dando o mesmo *hash* recalculado.

No entanto, isso mostra que será possível alterar dados dentro de um disco através de *assembly* ou ferramentas editora de disco sem que existam provas no próprio disco de como foram alteradas. Originando a técnica de falsificação de evidências que demonstraremos na próxima seção.

Outra ação bastante problemática é a copia indevida de informações, que podem ser copiadas facilmente através de ferramentas geradoras de imagens bit a bit, clonagem ou até mesmo trechos específicos através das chamadas em *assembly* ou pela ferramenta editora de disco. Com isso, o disco inteiro ou informações sigilosas podem ser copiados sem que exista nada comprobatório na

mídia do disco que leve a consumação do fato. Exceto, com relação aos dados armazenados no *Self Monitoring Analysis and Reporting Technology* (SMART).

Além das evidências geradas em memória DRAM, a placa controladora do disco possui uma memória ROM na qual é responsável por armazenar um *log* de eventos que acontece com o disco. Este *log* é gerado por uma ferramenta nativa do próprio disco que monitora suas ações mecânicas como método de predição de problemas ao disco. Este *log* é acessado por ferramentas específicas, e dentro dos seus vários atributos de informações, existem duas que são importantíssimas para a avaliação da perícia. Como o tempo de uso do disco, e quantas vezes o mesmo foi ligado e desligado.

Mesmo sendo informações indiretas que não comprovem a ação propriamente dita do autor do crime, esta, é a única fonte de evidência gerada que pode ser detectada pela perícia *off-line*. Todavia, esta teria maior validade caso a perícia tivesse essa base de tempo anterior como referência. Logo, os mesmos iriam entender que o disco foi utilizado posteriormente, mas, não saberiam o que foi feito no disco. A Figura 58 mostra uma ferramenta que acessa o *log* do SMART com os seus atributos.

Concluímos que sem uma política de segurança mais severa para a coleta destas informações assim como falta de câmeras de segurança locais, o atacante estará imune à perícia *off-line*. E também imune ao realizar o ataque de falsificação de evidências, que demonstraremos a seguir.

Descrição do dispositivo

SAMSUNG SP0411N (S01JJ10WB20797)

Minutos

ID	Descrição de atributos	Intervalo	Valor	Pior...	Dados	Estado
01	Raw Read Error Rate	51	100	100	23	OK (o valor é normal)
03	Spinup Time	0	74	1	4800	OK (sempre funcionará)
04	Start/Stop Count	0	99	99	1777	OK (sempre funcionará)
05	Reallocated Sector Count	10	86	86	31	OK (o valor é normal)
07	Seek Error Rate	51	253	253	0	OK (o valor é normal)
08	Seek Time Performance	0	253	253	0	OK (sempre funcionará)
09	Power-On Time Count	0	99	99	1093087	OK (sempre funcionará)
0A	Spinup Retry Count	49	253	253	0	OK (o valor é normal)
0C	Power Cycle Count	0	99	99	1017	OK (sempre funcionará)
C2	Temperature	0	139	115	33	OK (sempre funcionará)
C3	Hardware ECC Recovered	0	100	100	141681946	OK (sempre funcionará)
C4	Reallocation Event Count	0	23	23	167	OK (sempre funcionará)
C5	Current Pending Sector Count	10	253	253	0	OK (o valor é normal)
C6	Offline Uncorrectable Sector Count	10	38	38	135	OK (o valor é normal)
C7	Ultra ATA CRC Error Rate	51	100	100	0	OK (o valor é normal)
C8	Write Error Rate	51	100	100	0	OK (o valor é normal)
C9	Soft Read Error Rate	51	100	100	2	OK (o valor é normal)

Figura 58 – Informações do SMART do disco

Fonte: Software Everest

4.4 Falsificação de evidências

Com base no que já foi exposto sobre as técnicas anti-forense, a falsificação pode ser caracterizada pelo misto das técnicas anteriores, como a junção de *slack space* e eliminação das fontes de evidências.

A falsificação tem o intuito de fazer com que as evidências pareçam qualquer outra coisa menos a evidência real. Como já foi citado na seção 1.3.3, esta técnica visa levar o perito a um falso positivo ou até mesmo incriminar inocentes.

Esta técnica é muito problemática para a perícia devido ao atacante não deixar rastros no disco de como foi realizada a falsificação, logo, esta técnica também é imune a uma forense *off-line*.

Para demonstrarmos que esta problemática acontece, utilizamos o disco em que foi calculado o *hash* anteriormente mostrado na Figura 57, e alteramos todo o MBR do em baixo nível, de forma a obter um novo *hash*. A Figura 59 mostra este novo valor.

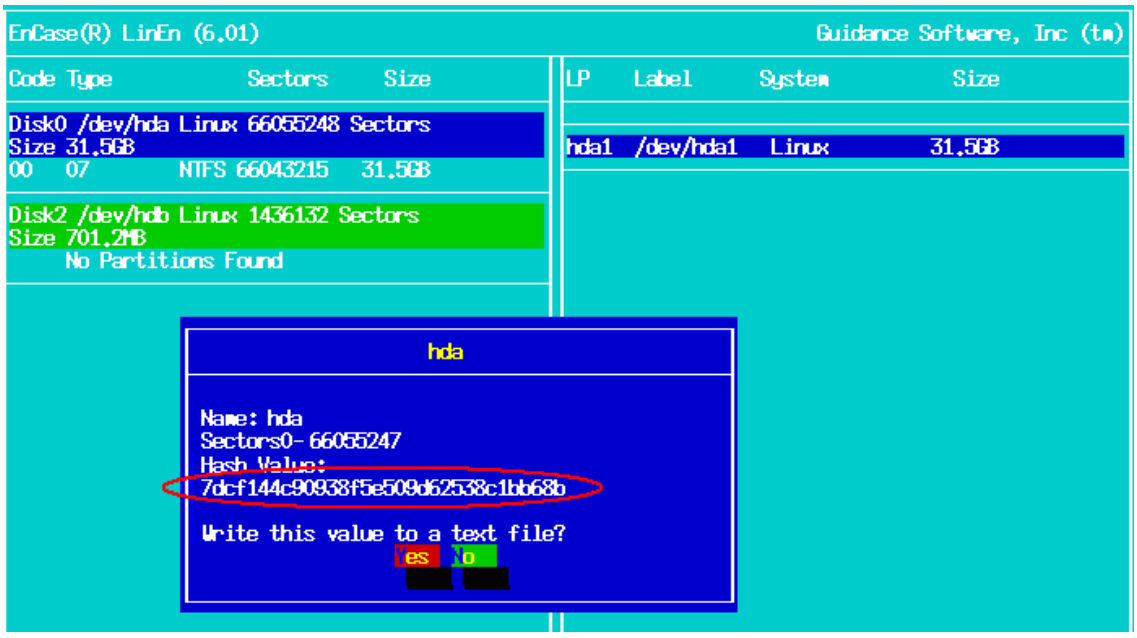


Figura 59 – Novo hash calculado após alteração do MBR

Fonte: Software Linen

A partir daí, voltamos aos binários anteriores do MBR e batermos novamente o *hash*, em que o resultado foi igual ao da Figura 58, provando que o acesso em baixo nível altera apenas os binários nos quais queremos falsificar. No entanto, isto

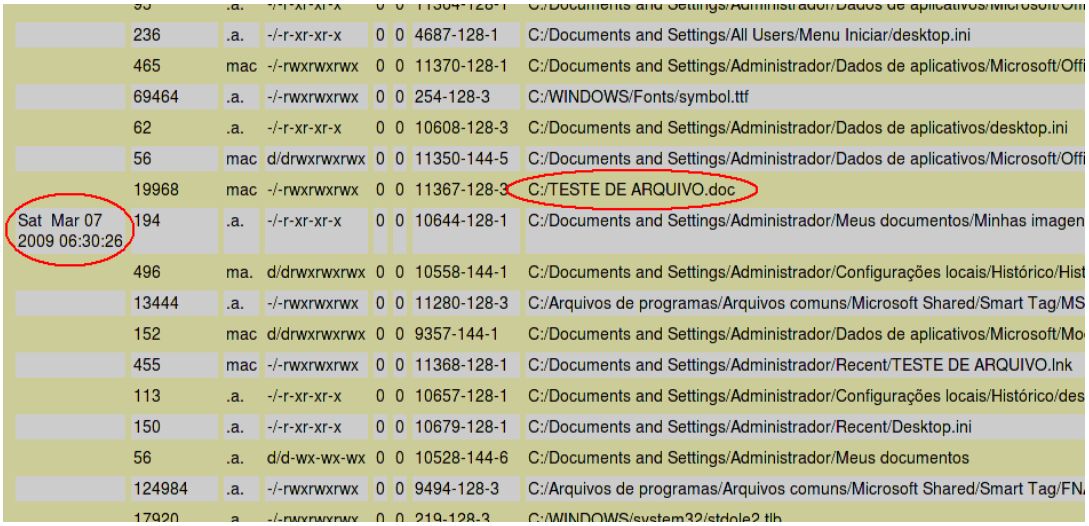
mostra que não vai existir nenhuma evidência em disco de que alteramos e voltamos novamente os binários do MBR, pois conseguimos obter o mesmo *hash* anterior. Apenas irá existir a informação que o disco foi utilizado por um tempo maior através dos atributos encontrados no SMART, porém, não haverá nada comprobatório do que foi feito.

Este resultado seria bem diferente caso fosse realizado em nível de sistema de arquivos através de alguma ferramenta, pois qualquer evento realizado neste nível deixa comprovações de chamadas no sistema, além da atualização dos *mactimes* dos arquivos envolvidos.

Como já foi citado na seção anterior, o atacante com o acesso em baixo nível pode alterar qualquer estrutura binária do disco e utiliza-la para a falsificação de nomes de arquivos, conteúdos, *mactimes*, implantar falsas evidências e etc.

Para comprovar um destes ataques, criamos um documento chamado “TESTE DE ARQUIVO.DOC” em que usamos como piloto para a prática dos ataques.

A Figura 60 mostra através da linha de tempo *(timeline)* do *Autopsy* a criação do arquivo “TESTE DE ARQUIVO.DOC” juntamente com outros arquivos de sistema e arquivos vinculados.



	236	.a.	-/-r-xr-xr-x	0	0	4687-128-1	C:/Documents and Settings/All Users/Menu Iniciar/desktop.ini
	465	mac	-/-rwxrwxrwx	0	0	11370-128-1	C:/Documents and Settings/Administrador/Dados de aplicativos/Microsoft/Offi
	69464	.a.	-/-rwxrwxrwx	0	0	254-128-3	C:/WINDOWS/Fonts/symbol.ttf
	62	.a.	-/-r-xr-xr-x	0	0	10608-128-3	C:/Documents and Settings/Administrador/Dados de aplicativos/desktop.ini
	56	mac	d/drwxrwxrwx	0	0	11350-144-5	C:/Documents and Settings/Administrador/Dados de aplicativos/Microsoft/Offi
	19968	mac	-/-rwxrwxrwx	0	0	11367-128-3	C:/TESTE DE ARQUIVO.doc
Sat Mar 07 2009 06:30:26	194	.a.	-/-r-xr-xr-x	0	0	10644-128-1	C:/Documents and Settings/Administrador/Meus documentos/Minhas imagen
	496	ma.	d/drwxrwxrwx	0	0	10558-144-1	C:/Documents and Settings/Administrador/Configurações locais/Histórico/Hist
	13444	.a.	-/-rwxrwxrwx	0	0	11280-128-3	C:/Arquivos de programas/Arquivos comuns/Microsoft Shared/Smart Tag/MS
	152	mac	d/drwxrwxrwx	0	0	9357-144-1	C:/Documents and Settings/Administrador/Dados de aplicativos/Microsoft/Mo
	455	mac	-/-rwxrwxrwx	0	0	11368-128-1	C:/Documents and Settings/Administrador/Recent/TESTE DE ARQUIVO.lnk
	113	.a.	-/-r-xr-xr-x	0	0	10657-128-1	C:/Documents and Settings/Administrador/Configurações locais/Histórico/des
	150	.a.	-/-r-xr-xr-x	0	0	10679-128-1	C:/Documents and Settings/Administrador/Recent/Desktop.ini
	56	.a.	d/d-wx-wx-wx	0	0	10528-144-6	C:/Documents and Settings/Administrador/Meus documentos
	124984	.a.	-/-rwxrwxrwx	0	0	9494-128-3	C:/Arquivos de programas/Arquivos comuns/Microsoft Shared/Smart Tag/FN

Figura 60 – Criação do arquivo mostrada no timeline

Fonte: Software Autopsy

As Figuras 61 e 62 mostram as características do arquivo, desde os *mactimes* até a sua posição em disco.

Figura 61 – Características de entrada do arquivo na $MFT

Fonte: Software Autopsy

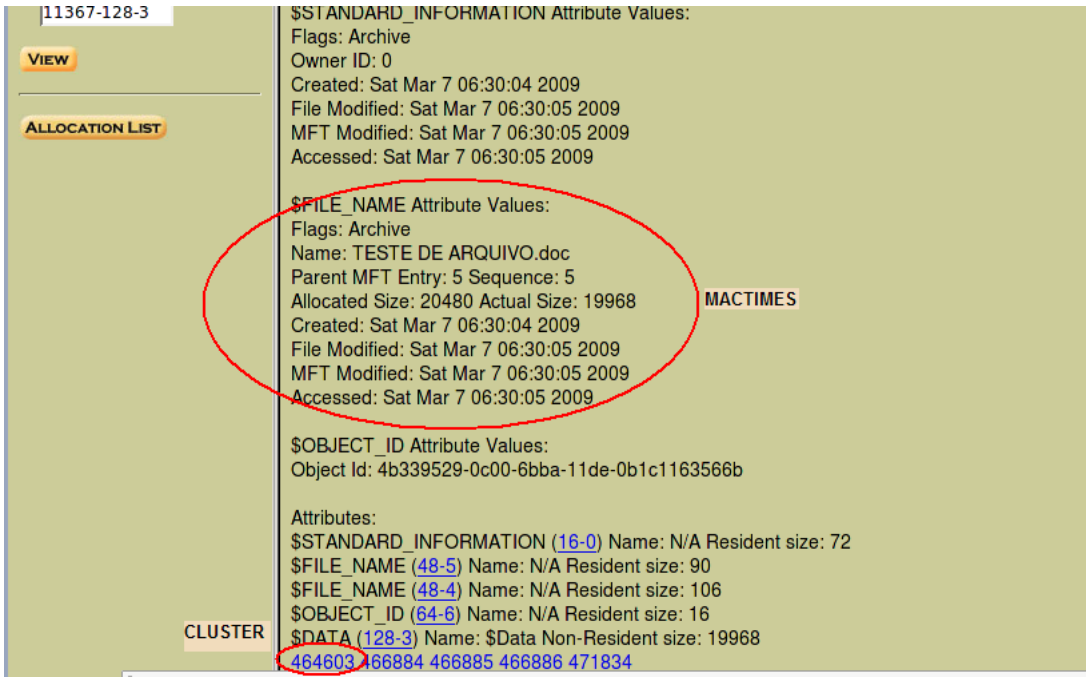


Figura 62 – Mactimes e cluster do arquivo

Fonte: Software Autopsy

A Figura 63 mostra os binários do arquivo em disco que correspondem ao conteúdo do arquivo.

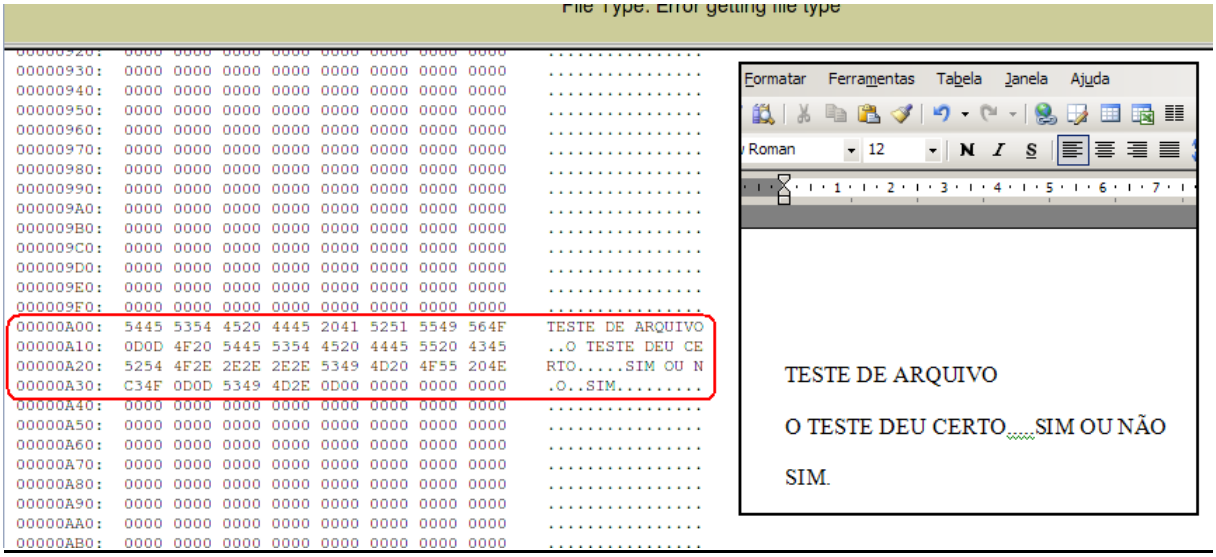


Figura 63 – Binários do arquivo

Fonte: Software Autopsy

A partir das informações da análise forense que foram bastante úteis para o ataque, e principalmente do valor do *cluster* 464603 apresentado na Figura 63, chegamos ao setor absoluto do disco em que corresponde ao conteúdo do arquivo preterido através das seguintes condições:

1 – Um *cluster* em sistema de arquivos NTFS tem 8 setores conforme a Figura 20.
2 – Multiplicando o cluster (464603) de localização do arquivo pela quantidade de setores (8) por cluster do NTFS, teremos a posição do setor 3716824 como posição relativa do disco.
3 – Sabendo que existem 62 setores livres antes do sistema de arquivo, e que a inicialização do sistema de arquivos compreende 6 setores conforme mostrado na seção 2.2.1, adicionamos mais 68 setores ao setor relativo, obtendo o setor absoluto 3716892 que compreende a localização exata do conteúdo do arquivo.

A Figura 64 mostra através do editor de disco a localização do setor absoluto e os respectivos binários que correspondem aos da Figura 64.

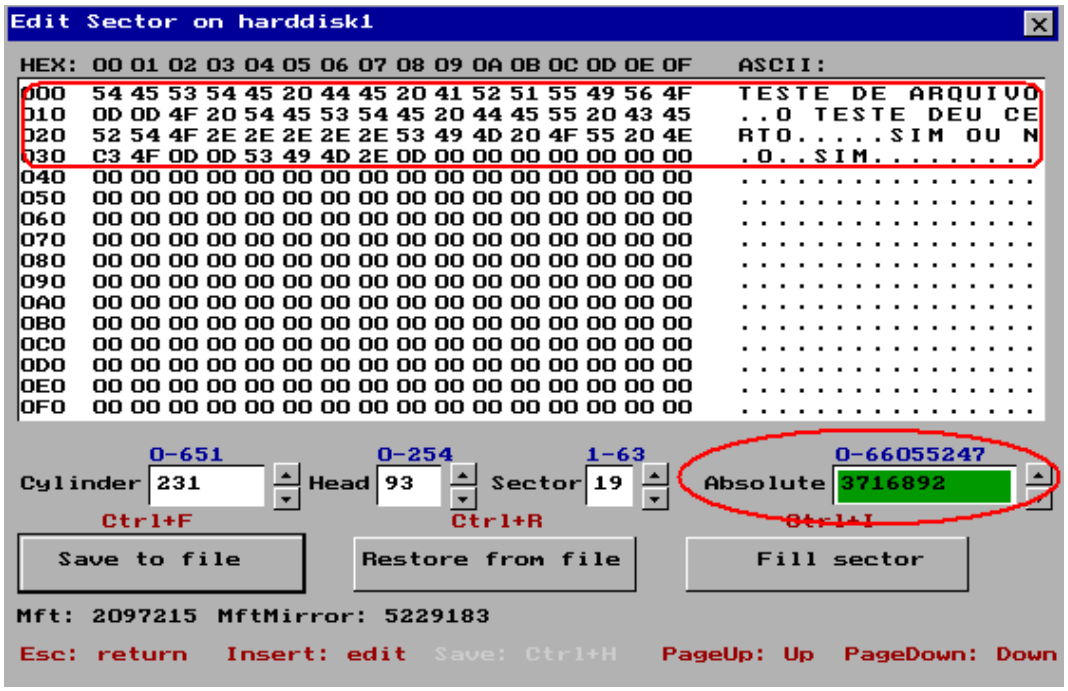


Figura 64 – Localização absoluta do conteúdo do arquivo

Fonte: Software Partition Table Doctor

Utilizando o mesmo software editor de disco, mudamos alguns caracteres como forma de falsificação do arquivo como mostrado na Figura 65. A partir daí, para validar o efeito da falsificação sem evidências em disco, analisamos o arquivo novamente através da análise forense com o *Autopsy*, e verificamos que todas as características do arquivo se conservaram, exceto os binários do conteúdo conforme alteramos. A Figura 66 mostra as características do arquivo idênticas aos das Figuras 61 e 62, exceto pelo *hash* diferente que identifica que o arquivo foi modificado. Ou seja, através da perícia *off-line* sem uma referência de dados anteriores (*hash*), as evidências levariam ao resultado que o autor do arquivo realmente tinha escrito o conteúdo falsificado.

Para reforçar mais ainda esta afirmação, utilizamos à ferramenta *Winhex* na opção de *string search* na qual procuramos pelo texto “TESTE DE ARQUIVO” em todo o disco local, para ver quais as entradas geradas deste arquivo no sistema. Na hipótese de existir alguma cópia do arquivo antes de ser falsificado.

A Figura 67 mostra que existem 9 entradas armazenadas em disco, mas nenhuma destas possuem evidências do arquivo antes de ser falsificado.

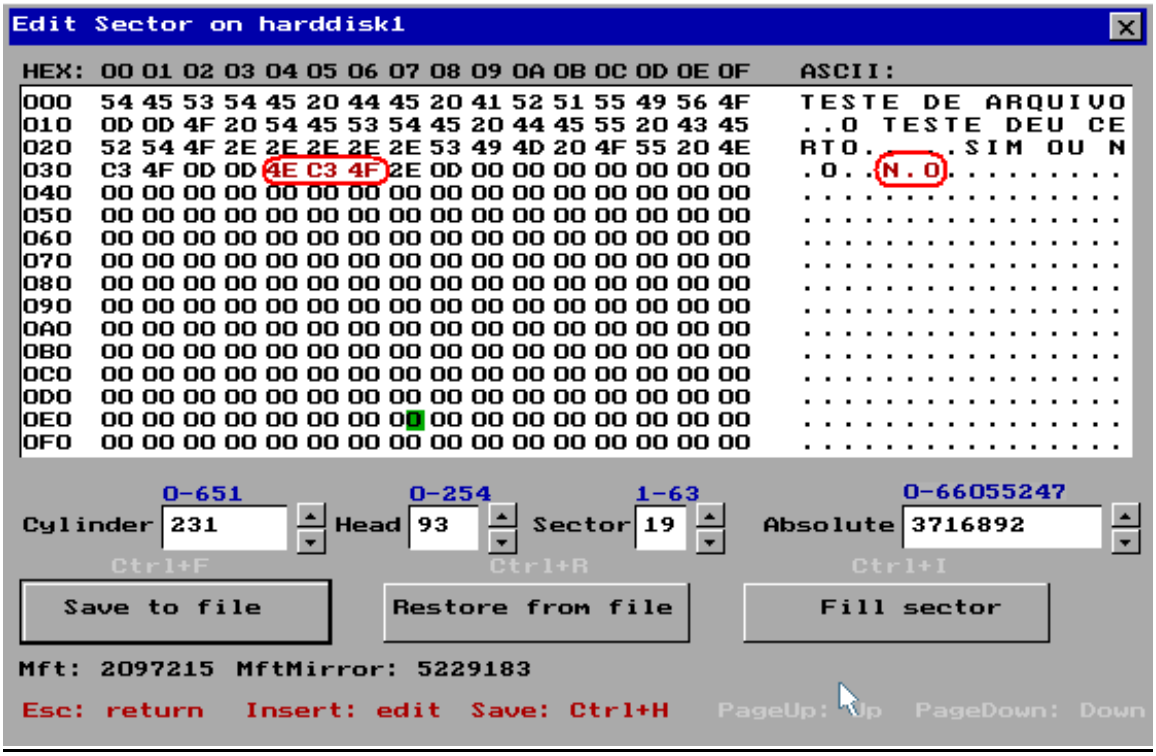


Figura 65 – Falsificação do arquivo

Fonte: Software Partition Table Doctor

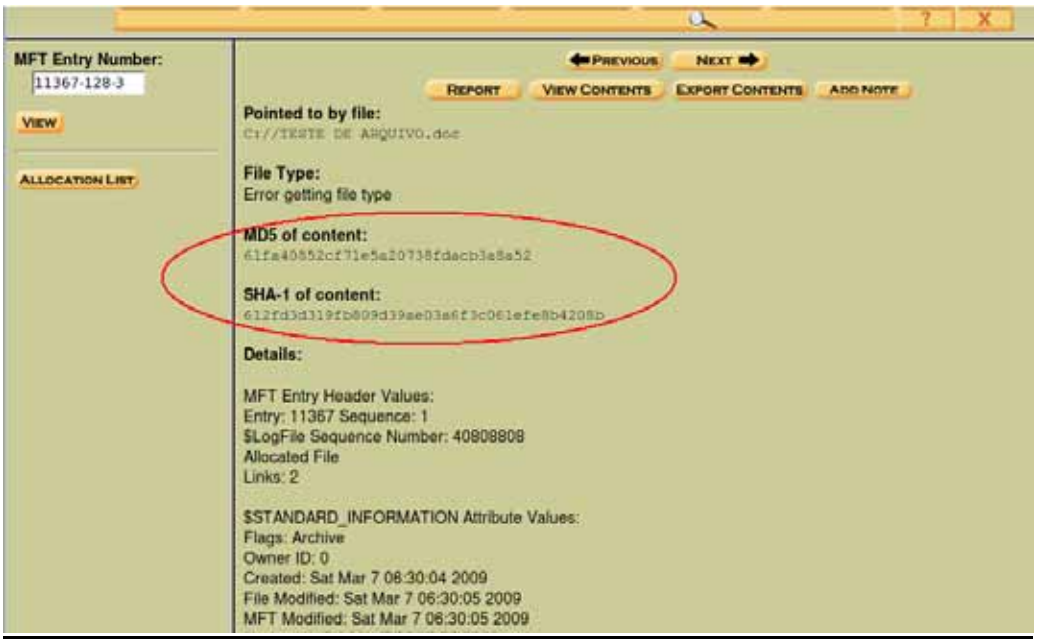


Figura 66 – Falsificação do arquivo detectada apenas pelo hash do conteúdo

Fonte: Software Partition Table Doctor

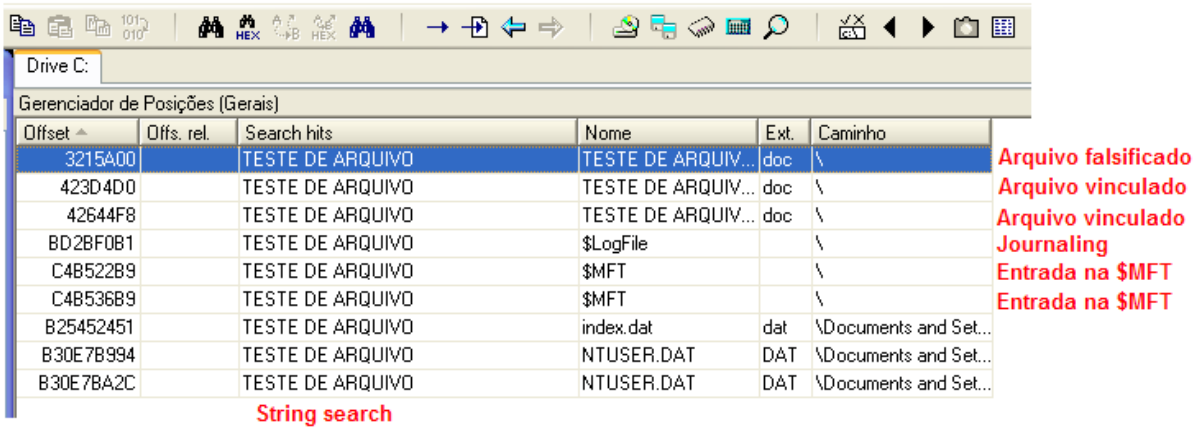


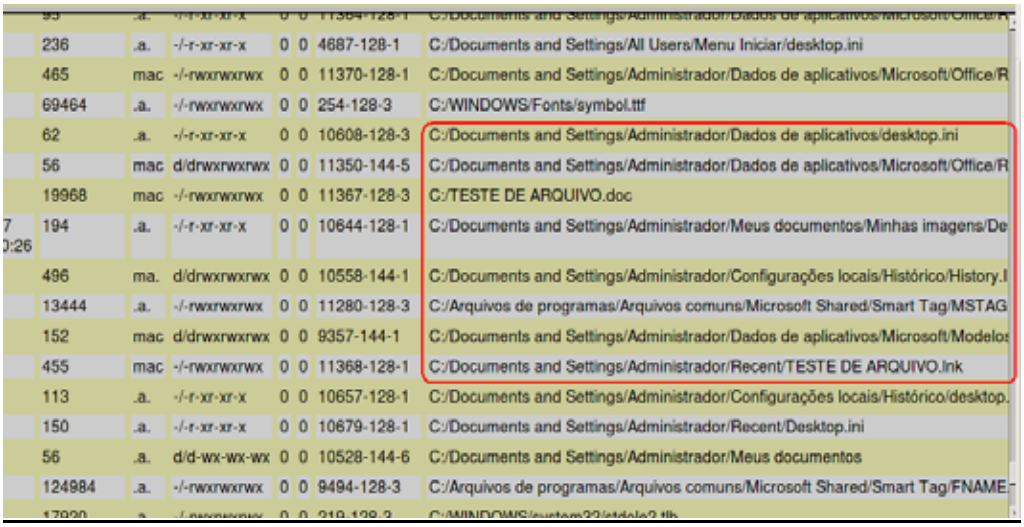
Figura 67 – String search pela ferramenta Winhex

Fonte: Software Winhex

Uma outra maneira interessante de levar a perícia para um falso relato é alterar os *mactimes* em baixo nível. Como já vimos anteriormente, os *mactimes* são variáveis importantes para a determinação do histórico de uso do computador.

Este tipo de ataque é bem mais complexo, porém, mesmo em baixo nível, se as alterações não forem bem aplicadas, a perícia pode detectar incoerências entre os tempos. No caso do texto anterior que foi falsificado, mudamos apenas alguns binários do conteúdo do arquivo, permanecendo os mesmos *mactimes.* Já no caso da alteração dos tempos, devemos mudar as *flags* da $MFT que correspondem aos *mactimes*.

Analisando cuidadosamente a Figura 68, entendemos que existem vários arquivos de sistema além de outros que estão vinculados à criação do arquivo “TESTE DE ARQUIVO.DOC”, e cada um deles tem um *mactime* associado. Logo, se mudarmos apenas o *mactime* do arquivo propriamente dito, vão existir outros arquivos com o *mactime* anterior que serão percebidos pela perícia. No entanto, para uma alteração sem deixar dualidade de interpretação para a perícia, todos esses arquivos devem ter seus *mactimes* adulterados. Ou na pior das hipóteses, removidos.



	236	.a.	-/-r-xr-xr-x	0	0	4687-128-1	C:/Documents and Settings/All Users/Menu Iniciar/desktop.ini
	465	mac	-/-rwxrwxrwx	0	0	11370-128-1	C:/Documents and Settings/Administrador/Dados de aplicativos/Microsoft/Office/R
	69464	.a.	-/-rwxrwxrwx	0	0	254-128-3	C:/WINDOWS/Fonts/symbol.ttf
	62	.a.	-/-r-xr-xr-x	0	0	10608-128-3	C:/Documents and Settings/Administrador/Dados de aplicativos/desktop.ini
	56	mac	d/drwxrwxrwx	0	0	11350-144-5	C:/Documents and Settings/Administrador/Dados de aplicativos/Microsoft/Office/R
	19968	mac	-/-rwxrwxrwx	0	0	11367-128-3	C:/TESTE DE ARQUIVO.doc
7 0:26	194	.a.	-/-r-xr-xr-x	0	0	10644-128-1	C:/Documents and Settings/Administrador/Meus documentos/Minhas imagens/De
	496	ma.	d/drwxrwxrwx	0	0	10558-144-1	C:/Documents and Settings/Administrador/Configurações locais/Histórico/History.I
	13444	.a.	-/-rwxrwxrwx	0	0	11280-128-3	C:/Arquivos de programas/Arquivos comuns/Microsoft Shared/Smart Tag/MSTAG
	152	mac	d/drwxrwxrwx	0	0	9357-144-1	C:/Documents and Settings/Administrador/Dados de aplicativos/Microsoft/Modelos
	455	mac	-/-rwxrwxrwx	0	0	11368-128-1	C:/Documents and Settings/Administrador/Recent/TESTE DE ARQUIVO.lnk
	113	.a.	-/-r-xr-xr-x	0	0	10657-128-1	C:/Documents and Settings/Administrador/Configurações locais/Histórico/desktop.
	150	.a.	-/-r-xr-xr-x	0	0	10679-128-1	C:/Documents and Settings/Administrador/Recent/Desktop.ini
	56	.a.	d/d-wx-wx-wx	0	0	10528-144-6	C:/Documents and Settings/Administrador/Meus documentos
	124984	.a.	-/-rwxrwxrwx	0	0	9494-128-3	C:/Arquivos de programas/Arquivos comuns/Microsoft Shared/Smart Tag/FNAME

Figura 68 – Arquivos de sistema associados a criação de arquivo

Fonte: Software Autopsy

Para mudar os *mactimes* mais facilmente, a maioria dos atacantes utiliza ferramentas que adulteram atributos de qualquer arquivo ou utilizam o artifício de mudança do relógio do sistema. Todavia, ficarão evidências de que este tipo de estratégia foi usado para este fim. Contudo, utilizamos umas destas ferramentas para saber como era o funcionamento a nível binário de alteração das *flags* da $MFT com relação aos *mactimes*.

A partir da ferramenta *Attribute Changer* e da arquitetura da $MFT mostrada anteriormente na seção 2.2.2 Figura 21, conseguimos entender o funcionamento de alteração dos binários dos *mactimes*. A Figura 69 mostra os binários caracterizados em uma entrada de arquivo na $MFT.

Cada atributo de tempo está associado a grupos de 64bits que estão dispostos no formato *little-endian* e nos *offsets* mostrados na figura anterior, onde cada combinação binária do bit menos significativo segundo Carrier (2005), equivale a 100ηs a partir de 1 de janeiro de 1601 (UTC). Logo, os *mactimes* são montados em cima de uma grande somatória de tempos de 100ηs. Com isso, a tarefa de alterar um ano, um mês ou um dia não é trivial.

Tentamos estabelecer uma seqüência lógica binária com relação aos *mactimes*, mas devido ao tempo e a complexidade da montagem do modelo, preferimos optar pelo uso paralelo das ferramentas automatizadas de alterações de atributos. Uma vez que já sabíamos os *offstes* das *flags* binárias, ficou mais fácil à manipulação.

Portanto, com as ferramentas que já alteram estes *mactimes* automatizadamente como a *Attribute Changer,* utilizamos o mesmo padrão binário gerado por esta ferramenta em um outro computador de acordo com um *mactime* falsificado, e utilizamos estes mesmos binários na máquina alvo em baixo nível para alterar o *mactime* do arquivo “TESTE DE ARQUIVO.DOC”

A Figura 70 mostra a tela da ferramenta *Attribute Changer.*

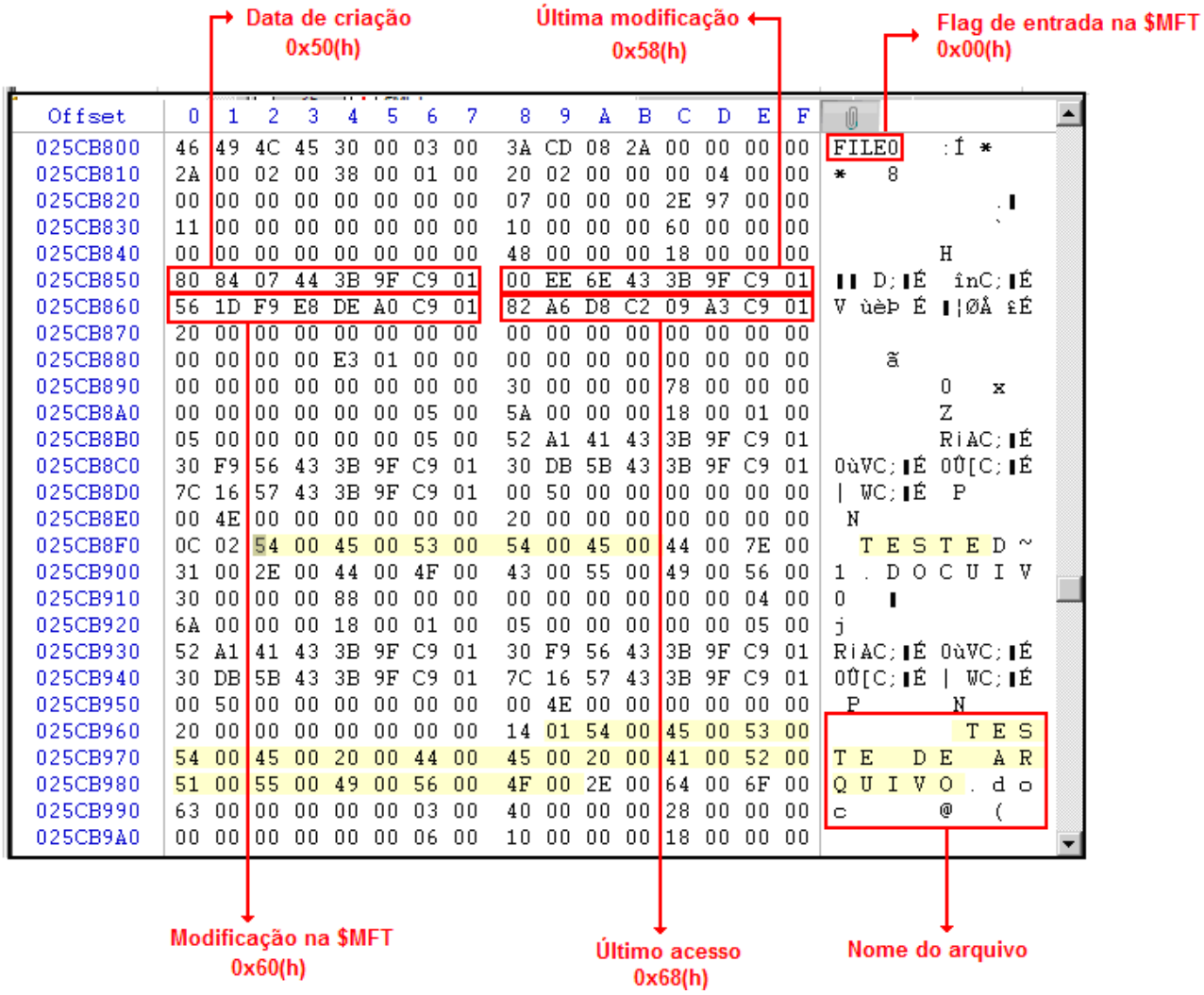


Figura 69 – Binários dos mactimes de uma entrada na $MFT

Fonte: Software Winhex

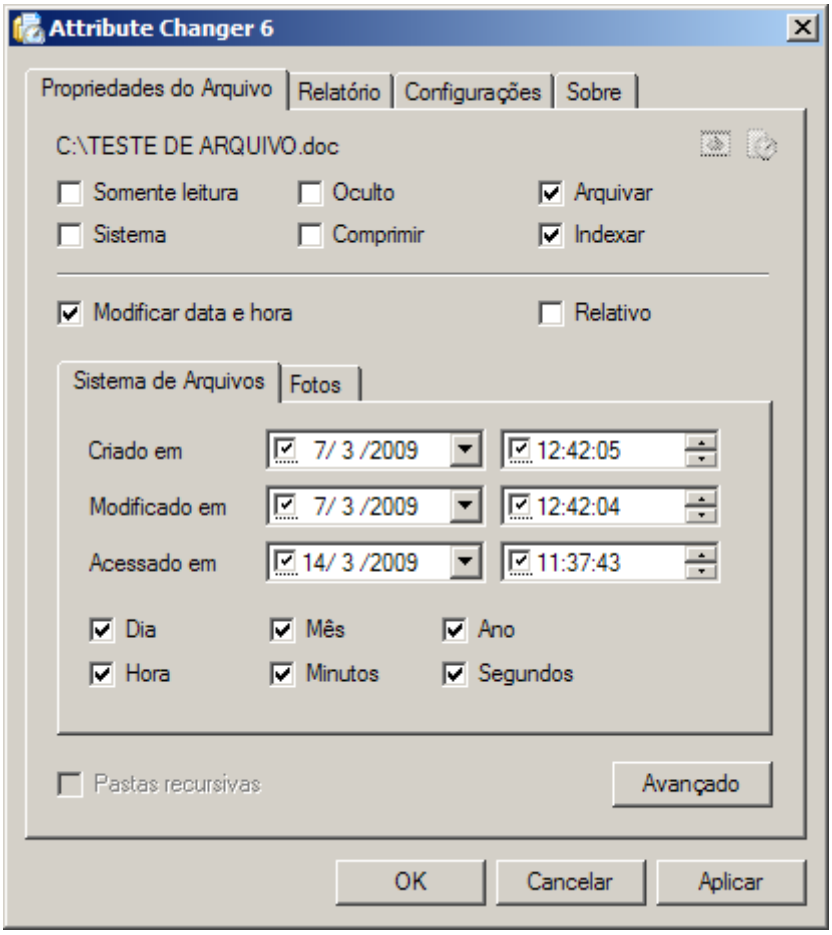


Figura 70 – Alterações de atributos de um arquivo

Fonte: Software Attribute Changer

Mesmo com todas as opções da ferramenta *Attribute Changer*, a mesma não consegue alterar a *flag* de tempo de modificação na $MFT por não possuir esta opção. Portanto, a alteração desta *flag* ficou por nossa conta na qual utilizamos o mesmo padrão de tempo utilizado nas outras flags que foi proposta pela própria ferramenta. Com isso, conseguimos alterar qualquer *mactime* de qualquer arquivo dentro do disco, sem que exista alguma evidência em disco da forma como alteramos.

A tarefa de falsificar qualquer evidência é muito trabalhosa e requer muitos detalhes que devem ser percebidos pelo atacante para não deixar dúvidas para a perícia. No entanto, é a única maneira de não gerar evidências na mídia do disco de como foram falsificadas as informações.

E quanto mais sofisticada for à falsificação, mais complexa a mesmo será de ser executada e interpretada.

DISCUSSÃO

Diante do avanço da forense computacional para responder as ações ilegais do uso do computador, a anti-forense tenta está a um passo a frente para dificultar e inviabilizar as respostas que os peritos procuram. Por ser uma nova ciência, a anti-forense ainda não é bem disseminada nos meios profissionais e nem tão pouco na academia, sendo esta, uma de suas grandes vantagens, além da falta de políticas de segurança severas e das leis que punem os criminosos.

Comprovamos que os seus impactos podem deixar a perícia sem respostas, ou na pior das hipóteses, com falsas respostas que levam inocentes a serem culpados. Uma vez que os dados estão passíveis de corrompimentos e adulterações sem que exista uma comprovação absoluta de quem o fez.

Dentro dos investimentos de segurança que visam principalmente evitar ataques de redes de terceiros, a zona perimetral (DMZ) fica sempre com a maior fatia, no entanto, verificamos que em uma rede local, um atacante pode livremente praticar técnicas anti-forense através de ataques físicos. E com fracas políticas de segurança de acesso física, o atacante fica motivado a praticar técnicas de destruição, ocultação, eliminação e falsificação de evidências.

A partir dos experimentos, descobrimos que toda uma operação em baixo nível não deixa rastros de como foram modificadas as informações no disco, exceto na memória DRAM ou através de informações indiretas do uso quantitativo do disco pelo SMART que está residente em memória ROM na placa controladora do disco. Comprovando o principio de Locard, que afirma em que toda uma ação seja ela qual for, é gerada evidências.

Verificamos que o impacto de uma técnica realizada em baixo nível pode ser difícil de ser detectada pela perícia, considerando que o atacante desligue a máquina após qualquer operação anti-forense e espere o tempo necessário para a volatização das informações em memória DRAM. E que não exista nenhuma segurança local como câmeras ou alguma ferramenta que registre as horas de funcionamento totais de uso do computador trabalhando em conjunto com o SMART. Logo, com essa falha de segurança local, o próprio usuário do computador pode ser lesado de algo que ele não praticou além de dados ou informações estarem passíveis de falsificações ou de cópias não autorizadas, seja por algum colega de sala ou até mesmo pelos funcionários da TI.

Dentre as várias técnicas anti-forense, verificamos que a destruição de evidências mostrou-se como uma das de maior impacto, pelo fato das provas serem totalmente destruídas em nível de software, sem que nada neste nível consiga detectá-las. Exceto pela análise da mídia em *hardware*, que infelizmente possui um custo elevado, no entanto, a problemática se agrava com as ferramentas de *wipe* que elimina todas as possibilidades de recuperação.

Para a técnica de ocultação de evidências, utilizamos maior elaboração para os ataques mais específicos como *slack spaces* e ADS, onde a ADS foi facilmente detectada pelas ferramentas forenses e não sendo uma das preferidas pelos atacantes. Já o *slack spaces* se mostrou bastante eficiente por sua detecção ser muito difícil quando usado em conjunto com a criptografia e gravado de forma fragmentada. Por outro lado, sua aplicação é complexa, logo, o ataque tradicional de *slack space* sem criptografia e seqüencial é mais preferível, porém, a perícia detecta-o. E como uma melhor alternativa de ocultação, as ferramentas de esteganografia e criptografia são as mais eficazes, se considerarmos a questão da facilidade de implementação. E para as ferramentas de criptografia, estas são consideradas legítimas dentro dos aspectos da segurança, sendo este o grande agente motivador para o uso deliberado. Todavia, a criptografia e esteganografia podem ser detectadas, mas a perícia dependerá do esforço computacional para decifrar a mensagem ocultada, que poderá levar anos.

Dentro de todas as técnicas anti-forense experimentadas, a eliminação das fontes de evidências foi a mais fácil de ser praticada e eficiente pelo fato de não existir evidências na mídia do disco devido aos dados ficar inalterados. O uso de *live* CDs forenses estimula esta prática além de possuírem ferramentas que copiam o disco fielmente bit a bit, no qual pode ser utilizado para fazer cópias indevidas. Ou seja, informações sigilosas podem ser facilmente copiadas sem que haja provas de que elas foram copiadas.

A falsificação de evidências foi considerada a técnica mais complexa dentro dos experimentamos e a de maior impacto devido a não existir provas concretas de que um inocente não tenha falsificado um documento. Contudo, a falsificação requer muitos cuidados do atacante para não deixar dualidade de informações em disco e DRAM para a perícia. Mas satisfazendo estas condições, nem mesmo as informações do SMART são concretas para culpar ou inocentar alguém de algum fato.

Diante do exposto, concluímos que sem uma segurança severa local como câmeras ou políticas de monitoramento de *logs* de uso do computador e certificação de validade das informações em disco através de *hashes*, não há como se precaver dos ataques da anti-forense, já que o próprio sistema permite esta ação. Atualmente alguns laptops e até mesmo desktops já vem com um recurso de proteger o disco contra gravação, porém, não evita a cópia indevida que pode ser facilmente visualizada em um clone. Mesmo com maiores custos de implementação de segurança física local que não investidos, as técnicas anti-forense podem gerar impactos de ordens sem precedentes além de perdas incalculáveis.

REFERÊNCIAS

ALMANZA, Andrés R.**VII Jornada Nacional de Seguridad Informática.** Disponível em < www.acis.org.co/fileadmin/Base_de_Conocimiento/VII_JornadaSeguridad/VIIJNSI_AAlmanza.pdf>. Acesso em: 22 dez. 2008.

AMORIN, Paulo Henrique. **Policia Federal não consegue decifrar HDs.** Disponível em <http://www.paulohenriqueamorim.com.br/forum/Post.aspx?id=668>. Acesso em: 28 out. 2008.

CARRIER, Brian. File System Forensic Analysis. 1 ed. Addison Wesley, 2005. 199-282p.

CARVALHO, Diego Fiori de. **Exploração Tecnológica para Esteganografia em Vídeos Digitais.** 2005. 39f. (Monografia - Graduação em Ciência da Computação). Universidade de São Paulo, São Carlos.

Computer Forensics. **NTFS File System**. Disponível em < www.cse.scu.edu/~tschwarz/coen152_05/PPtPre/NTFSFS.ppt>. Acesso em: 1 out 2008.

DIGITAL SIGNAGE. **Digital Signaging, uma perspectiva histórica.** Disponível em < http://www.portaldigitalsignage.com.br/sinaging.php>. Acesso em: 18 abr. 2009.

DISCOVERY. **Crime e Prática Forense.** Disponível em < http://www.discoverybrasil.com/guia_crime/crime_analise/crime_fibras/index.shtml>. Acesso em: 05 jan. 2009.

PEREIRA, Evandro Della Vecchia et al. **Forense Computacional: Fundamentos, tecnologias e desafios atuais.** Disponível em < www.sbseg2007.nce.ufrj.br/documentos/Minicursos/minicurso_forense.pdf>. Acesso em: 10 nov. 2008.

EFENSE, **Software Helix.** Disponível em <http://www.e-fense.com/helix/> Acesso em: 16 nov. 2008.

FARMER, Dan; WIETSE, Venema. **Perícia Forense Computacional: Teoria e prática aplicada.** 1ed. São Paulo: Prentice Hall. 189p.

FREITAS, Andrey R. de. **Perícia Forense Aplicada à Informática**. 1. ed. Rio de Janeiro: Brasport, 2006. 216p. Bibliografia: p. 1

GUIMARÃES, Célio Cardoso et al. **Forense Computacional: Aspectos Legais e Padronização.** Disponível em: <http://www.las.ic.unicamp.br/paulo/papers/>. Acesso em: 12 dez. 2008.

GUTMANN, Peter. **Secure Deletion of Data from Magnetic and Solid-State Memory.** Disponível em: <http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html>. Acesso em: 9 jan. 2009.

GUTMANN, Peter. **Data Remanence in Semiconductor Devices.** Disponível em: <http://www.cs.auckland.ac.nz/~pgut001/>. Acesso em: 9 jan. 2009.

HALDEMAN, J. Alex. **Lest We Remember: Cold Boot Attacks on Encryption Keys.** Disponível em: <http://citp.princeton.edu/memory/>. Acesso em: 1 fev. 2009.

HARRIS, Ryan. **Arriving at an anti-forensics consensus: Examining how to define and control the anti-forense problem.** Disponível em: <http://www.dfrws.org/2006/proceedings/6-Harris.pdf>. Acesso em: 19 dez. 2008.

HONORATO, Lívia N; Cardoso, Antônio L.M.S. **Crimes cometidos pela internet.** Disponível em: < http://www.scribd.com/doc/7630692/Crimes-cometidos-pela-Internet>. Acesso em 18 abr. 2009.

JONES, Sara. Computer Forensics. Disponível em: < www.middlesexcc.edu/faculty/Steven_Zale/Computer_%20Forensics.ppt>. Acesso de 22 dez. 2008.

LIMA, Gisele Truzzi. **Crimes virtuais aumentam, mas ainda falta regulamentação.** Disponível em: <http://ultimainstancia.uol.com.br/noticia/57166.shtml>. Acesso em: 27 out. 2008.

MILAGRE, José. **A Nova Armadilha Cracker ICEPAK: Análise Forense Computacional de Malwares e Códigos embutidos em websites.** Disponível em: <http://imasters.uol.com.br/artigo/10147>. Acesso em: 7 jan. 2009.

NOBLET, Michael G; POLLIT, Mark M; PRESLEY, Lawrence A. **Recovering and Examining Computer Forensic Evidence.** Disponível em: <http://www.fbi.gov/hq/lab/fsc/backissu/oct2000/computer.htm>. Acesso em: 10 nov. 2008.

NOGUEIRA, Márcio Luiz Machado**. Criptografia Clássica**, 2 de ago. de 2008. 26 28f. Notas de Aula.

NTI. **Michael R, Anderson.** Disponível em: <http://www.forensics-intl.com/mra.html>. Acesso em: 22 dez. 2008.

OLIVEIRA, Flávio de Souza. **Resposta a Incidentes e Análise Forense para Redes Baseadas em Windows 2000.** 2002. 150p. (Dissertação em Ciência da Computação). Universidade Estadual de Campinas. São Paulo.

OMAR, Emran. **Perícia Forense.** 25f**.** Notas de Aula.

PEREIRA, Marcos Nascimento Borges. **Aspectos da análise forense.** Disponível em: <www.security.usp.br/palestras/forense.pdf> Acessado em: 25 dez. 2008.

PETRI, Marcelo. **ESTEGANOGRAFIA,** 2004. 62f (Trabalho de conclusão de curso – Bacharel em sistemas de informação) – Instituto Superior Tupy, Joinville, Santa Catarina.

PIMENTA, Flávio Aparecido. **Perícia forense computacional baseada em sistema operacional Windows XP Professional.** 2007. 122f (Trabalho de conclusão de curso – Especialização em Segurança de Redes e Sistemas) – Centro Universitário SENAC, Sorocaba, São Paulo.

PIRES, Paulo S. da Motta. **FORENSE COMPUTACIONAL: UMA PROPOSTA DE ENSINO.** Disponível em: <http://www.dca.ufrn.br/~pmotta/trabalhos.html>. Acesso em 22 dez. 2008.

RECOVERY LABS, **CANCELAMENTO DE DADOS.** Disponível em <http://www.recoverylab.com.br/cancelamento.htm>. Acesso em: 30 jan. 2009.

RNP, **Rede Nacional de Pesquisa**. Disponível em: < http://www.rnp.br/cais/estatisticas/index.php > Acesso em: 18 abr. 2009.

SANTOS, Edmilson Porto. **UMA ABORDAGEM SOBRE RECUPERAÇÃO DE DADOS,** 2002. 93f (Monografia – Tecnologia em Processamento de dados) – Universidade Tiradentes, Aracajú, Sergipe.

SILVA, Luís Miguel. **Anti-Análise Forense.** Disponível em: < lms.ispgaya.pt/documentacao/seminario_ubi_anti-analise_forense.ppt>. Acesso em: 1 nov. 2008.

SOBEY, Charles H. **Recovering Unrecoverable Data.** Disponível em: < www.actionfront.com/whitepaper/Drive-Independent%20Data%20Recovery%20Ver14Alrs.pdf> Acesso em 20 jan. 2008.

TAKAGI, Nilton Hideki. **Fundamentos Matemáticos da Criptografia Quântica.** 2003. 80f (Monografia – Ciência da Computação) – Universidade Federal do Mato Grosso, Cuiabá, Mato Grosso.

TEODOROWITSCH, Roland. **Bootlib: uma linguagem para definição de setores de inicialização.** Disponível em: <gravatai.ulbra.tche.br/~roland/bootlib/clei2000.pdf>. Acesso em: 1 dez. 2008.

Vogon, Disponível em: <http://www.vogon-investigation.com/evidential_systems-01.htm>. Acesso em 15 nov. 2008.

WIRELESSBR. **Crimes Digitais (8) - "Ecos" da audiência pública + Novo artigo de Fernando Botelho.** Disponível em: <www.wirelessbrasil.org/wirelessbr/index2.php?option=com_content&do_pdf=1&id=607 ->. Acesso em 18 abr. 2009.