CENTRO UNIVERSITÁRIO DO ESTADO DO PARÁ

ÁREA DE CIÊNCIAS EXATAS E TECNOLOGIA

BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

Wilson Antonio Cosmo Macêdo

TÉCNICAS ANTI-FORENSES EM COMPUTADORES PESSOAIS

Belém

2017

CENTRO UNIVERSITÁRIO DO ESTADO DO PARÁ

ÁREA DE CIÊNCIAS EXATAS E TECNOLOGIA

BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

Wilson Antonio Cosmo Macêdo

TÉCNICAS ANTI-FORENSES EM COMPUTADORES PESSOAIS

Trabalho de Curso na modalidade Monografia, apresentado como requisito parcial para obtenção do grau em Bacharelado em Engenharia da Computação do Centro Universitário do Estado do Pará – CESUPA, sob orientação do Professor Especialista Itamar Jorge Vilhena de Brito.

Belém

2017

Wilson Antonio Cosmo Macêdo

TÉCNICAS ANTI-FORENSES EM COMPUTADORES PESSOAIS

Trabalho de Curso apresentado na modalidade Monografia, apresentado como requisito parcial para obtenção do grau em Bacharelado em Engenharia da Computação do Centro Universitário do Estado do Pará – CESUPA.

Data da Defesa: 04/12/2017

Banca Examinadora:

_____

Prof. Especialista Orientador Itamar Jorge Vilhena de Brito - CESUPA

_____

Prof. Mestre Andrea Cristina Marques de Araujo - CESUPA

_____

Prof. Mestre Ricardo Melo Casseb do Carmo - CESUPA

Belém

2017

DEDICATÓRIA

Dedico este trabalho aos meus pais, irmã e a todos que se esforçam para tornar a sociedade mais ética e justa.

AGRADECIMENTOS

Agradeço especialmente aos meus pais, Marta Cosmo e Wilson Macêdo, pela dedicação e pela confiança que sempre depositaram em mim.

Aos meus colegas de grupo de pesquisa que comigo se esforçaram para demonstrar a importância da aplicação dos conhecimentos acadêmicos em benefício à sociedade.

Ao meu orientador, Professor Itamar Brito pela orientação e pelo apoio fornecido a mim e aos Estudantes de Engenharia da Computação durante o curso.

“É divina a arte da sutileza e do segredo! É através dela que se aprende a ser invisível e inaudível; enigmáticos como um deus – assim poderemos ter o destino do inimigo em nossas mãos.”

Sun Tzu

RESUMO

A evolução das tecnologias da informação e o aumento de usuários no meio digital resultou também no crescimento do número de crimes no meio virtual, bem como a complexidade dos mesmos, criando assim novos desafios para os profissionais envolvidos na forense computacional. O objetivo deste estudo é justamente analisar as formas que usuários mal-intencionados portadores de computadores pessoais possam utilizar para tentar obstruir a análise pericial em caso de suspeita de crime, seja ocultando ou destruindo dados, para que as informações obtidas ao longo das análises deste trabalho possam auxiliar em casos em que é constada a presença de alguma dessas técnicas. A abordagem utilizada para realização do estudo foi o uso de simulações de algumas dessas técnicas em um ambiente controlado próximo ao cenário real e a utilização de ferramentas voltadas para perícia e investigação digital nestes cenários para avaliar a eficiência e as principais características destas técnicas, gerando desta maneira informações técnicas que podem fornecer uma vantagem ao perito ou investigador digital em casos com ocorrências semelhantes aos presentes neste estudo.

Palavras-chave: Forense. Investigação. Digital. Computador.

ABSTRACT

The evolution of information technologies and the increase of users in the digital environment also resulted in the increase of the number of crimes in the virtual environment, as well as the complexity of them, thus creating new challenges for the professionals involved in the computer forensics. The purpose of this study is precisely to analyze the forms that malicious users with personal computers can use to try to obstruct the expert analysis in case of suspect of crime, either by hiding or destroying data, so that the information obtained during the analyzes of this work may help in cases where the presence of any of these techniques. The approach used to carry out the study was the use of simulations of some of these techniques in a controlled environment close to the real scenario and the use of tools focused on expertise and digital research in these scenarios to evaluate the efficiency and main characteristics of these techniques, generating in this way technical information that may provide an advantage to the expert or digital investigator in cases with similar occurrences to those present in this study.

Keywords: Forensic. Investigation. Digital. Computer.

LISTA DE FIGURAS

LISTA DE SIGLAS

ADS – Alternate Data Stream
ARM - Advanced RISC Machine
CPU - Unidade central de processamento
GNU – Acrônimo recursivo que significa “GNU is Not UNIX”
GPL – General Public Licence
HD – Disco Rígido
IoT – Sigla inglesa para Internet das Coisas
MMS - Serviço de mensagens multimídia
MS-DOS - MicroSoft Disk Operating System
PC – Computador Pessoal
RAM - Memória de acesso aleatório
ROM - Memória somente de leitura
SMS - Serviço de mensagens curtas
SO – Sistema Operacional
TI – Tecnologia da Informação
TSK – The Sleuth Kit

SUMÁRIO

1. INTRODUÇÃO

1.1. APRESENTAÇÃO DO PROBLEMA

Nos dias de hoje a presença dos meios computacionais é bastante evidente nos mais diversos locais e contextos, abrangendo desde de grandes centros industriais e empresariais a ambientes residenciais, fato este que proporcionou um grande avanço nos mais diversos processos, incluindo o armazenamento de informações e a comunicação, pois com a presença dos computadores e da Internet transmitir, receber ou armazenar dados se tornou uma tarefa bastante trivial.

Porém a grande facilidade de tráfego de arquivos e informações não trouxe apenas benefícios à sociedade, pois com a disseminação dessas tecnologias surgiu também toda uma nova categoria de crimes executados em meio virtual, ou mesmo com o auxílio do mesmo, o que ocorre tanto pelo fato desta própria tecnologia facilitar a coordenação de ações ilegais, devido à capacidade de realizar comunicação à distância de forma rápida e simples, quanto pelo fato das pessoas armazenarem cada vez mais informações pessoais nos meios digitais, o que torna esses meios alvos importantes para os criminosos virtuais.

Para combater tais práticas ilegais que passaram a migrar e evoluir nos meios digitais foi necessário desenvolver todo um conjunto de práticas e procedimentos com o intuito de investigar e identificar este tipo de ocorrência, bem como os seus responsáveis, o que resultou no advento da forense computacional em si, somado a métodos de prevenção, como o pentesting.

Com esse novo cenário ocasionado pelo mundo virtual, a informação, de uma maneira geral, tornou-se um dos recursos mais importantes e procurados na atualidade, pois a posse deste recurso na situação correta proporciona ao indivíduo ou instituição que o possui uma grande vantagem estratégica em relação ao que estiverem realizando, e para obter essa vantagem muitas empresas e governos realizam atualmente inúmeros processos para coleta de informações, e no âmbito da segurança da informação esta realidade é completamente válida, pois a quantidade de informação que ou o infrator ou o investigador digital possui sobre os processos

computacionais é vital para definir qual dos dois lados apresentará sucesso em alcançar o seu objetivo.

Tendo em vista esse contexto, a realização de pesquisas e estudos voltados à segurança da informação é algo bastante necessário, uma vez que esta área está em alteração constante, ou seja, o levantamento frequente de informações sobres os métodos utilizados para fins ilegais nos meios computacionais é algo primordial para que se possa desenvolver melhores formas de se analisar essas ações e identificar suas ocorrências.

1.2. OBJETIVOS

1.2.1. Objetivo Geral

O objetivo desta monografia é realizar um estudo referente às técnicas anti-forenses que são utilizadas para ocultar ou destruir dados importantes ou confidenciais para dificultar ou impedir que estes sejam recuperados em processos periciais.

1.2.2. Objetivos Específicos

- Conceituar investigação digital forense;
- Conceituar memória e os seus sistemas de arquivos;
- Conceituar Software Livre e Software de Código Aberto;
- Apresentar técnicas anti-forenses no contexto de ocultamento e destruição de dados;
- Analisar as técnicas anti-forenses apresentadas através de demonstração de aplicação;

- Avaliar o resultado e a eficiência das técnicas anti-forenses apresentadas executando ferramentas periciais as quais as mesmas servem de contramedida.

1.3. JUSTIFICATIVA

Conforme as tecnologias e os softwares utilizados para sustentar os meios virtuais evoluem e se atualizam para que determinados aspectos sejam modificados ou melhorados, surgem também, em reação, novas técnicas para burlar os novos meios ou para obstruir o trabalho da perícia digital. Portanto, o permanente estudo das práticas anti-forenses tem o intuito de reunir informações relevantes para contribuir em favor do próprio trabalho da perícia digital, pois o mapeamento dessas técnicas favorecerá bastante o processo pericial caso exista a necessidade de enfrentamento de uma situação onde seja constatada a presença de tais técnicas obstrutivas.

1.4. METODOLOGIA

Para embasar o estudo foi realizada uma pesquisa bibliográfica sobre os principais conceitos envolvendo o tema, levando-se em consideração a relevância e a importância das obras e seus autores no âmbito da perícia digital e da segurança da informação, e a partir deste embasamento, realizou-se uma pesquisa sobre as principais técnicas utilizadas para obstruir o processo pericial no contexto de recuperação de dados. Em sequência, foram realizadas análises sobre as técnicas identificadas com base nas informações levantadas na pesquisa bibliográfica.

Para o devido suporte metodológico e, paralelamente, visando dar um caráter de comprovação prático imediato das técnicas mencionadas, foram realizadas execuções das mesmas em ambiente controlado, porém contundente com o contexto de um computador pessoal comumente utilizado, bem como a aplicação de contramedidas periciais referentes às contramedidas anti-forenses para melhor

avaliação dos resultados das mesmas, tudo devidamente gravado por meio de registro das aplicações em arquivos de vídeo.

1.5. ESTRUTURA DO TRABALHO

O presente trabalho foi estruturado da seguinte forma: No capítulo 1 é realizada a apresentação do problema, são definidos o objetivo geral e os objetivos específicos e é apresentada a justificativa e a metodologia. No capítulo 2 são abordadas as definições dos conceitos teóricos necessários para realizar o devido embasamento deste trabalho. O capítulo 3 por sua vez define mais detalhadamente a metodologia utilizada e os softwares utilizados durante os testes. No capítulo 4 são analisadas técnicas de ocultamento de informações no âmbito dos computadores pessoais, e no capítulo 5 é testado um método de destruição de dados neste mesmo contexto. O capítulo 6 contém as considerações finais deste estudo, bem como as dificuldades encontradas e propostas para trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. INVESTIGAÇÃO DIGITAL FORENSE

O atual cenário de integração tecnológica o qual a sociedade se encontra bastante agregada devido à presença da tecnologia da informação tornou possível o surgimento de novas formas de crime voltadas explicitamente para serem realizadas no meio digital, ou com o auxílio do mesmo. Tais ações criminosas passaram a ser definidas como “crime cibernético” (ou “cyber crime”), e segundo Manson (2008, p. 1, tradução nossa) “É útil pensar em “crime cibernético” como qualquer crime em que um computador ou outro dispositivo digital desempenha um papel, o que significa que a evidência digital está envolvida”.

> Quando definidos da perspectiva das vítimas gerais do crime cibernético, definimos de forma operacional crimes cibernéticos como: Delitos cometidos contra indivíduos ou grupos de indivíduos com motivos criminosos para prejudicar intencionalmente a reputação da vítima ou causar danos físicos ou mentais à vítima diretamente ou indiretamente, usando redes de telecomunicações modernas, como Internet (salas de bate-papo, e-mails, quadros de avisos e grupos) e telefones celulares (SMS / MMS). (HALDER; JAISHANKAR, 2011, p. 15, tradução nossa).

Porém, com a grande presença de dispositivos digitais em vários aspectos do cotidiano da maioria dos indivíduos e nas mais diversas funcionalidades e objetivos, a presença de algum elemento digital na maioria dos crimes de qualquer natureza se tornou bastante frequente, mesmo quando não existe a intenção do criminoso de se aproveitar diretamente de um determinado meio computacional.

> Nesta era moderna, é difícil imaginar um crime que não possua uma dimensão digital. Criminosos, violentos e de colarinho branco, estão usando a tecnologia para facilitar suas ofensas e evitar a apreensão, criando novos desafios para advogados, juízes, agentes da lei, examinadores forenses e profissionais de segurança corporativa. (CASEY, 2011, p. 32, tradução nossa).

Sendo assim, a análise desses crimes cibernéticos está diretamente atrelada aos artefatos virtuais que podem ser usados como indícios de que a determinada atividade ilegal ocorreu de fato, o que é denominado como evidência digital. Segundo Carrier (2005, p. 12, tradução nossa) “A evidência digital é um objeto digital que contém informações confiáveis que suportam ou refutam uma hipótese”.

Com a diversidade de dispositivos digitais que estão presentes em diversos âmbitos, a abrangência de artefatos virtuais que podem ser considerados como evidência digital se tornou bastante ampla. De acordo com Chung, Park e Lee (2017, p. 1, tradução nossa) “Em várias investigações criminais recentes, funcionários da lei, peritos legais e peritos forenses tentaram usar dispositivos IoT ‘always-on’ como fontes de artefatos forenses semelhantes às caixas-pretas da vida humana”.

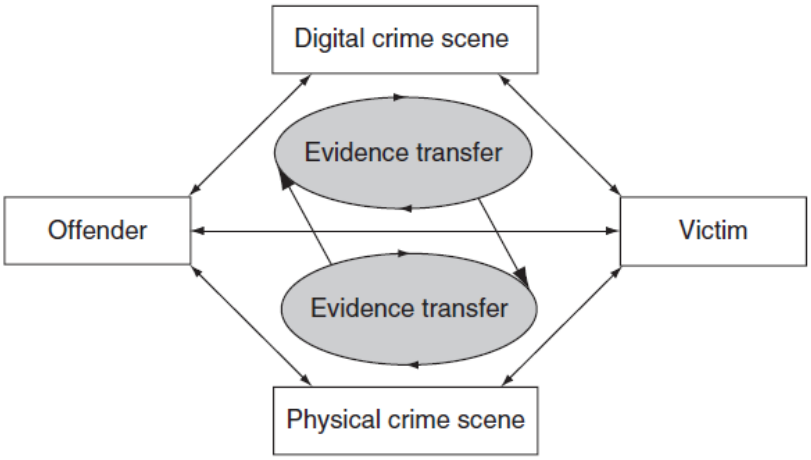
Tais evidências não necessariamente estão dispostas em um cenário em que possam ser identificadas e analisadas trivialmente, uma vez que qualquer tipo de arquivo ou dado virtual que é relevante para a comprovação de um determinado crime virtual pode ser considerado uma evidência digital, e a identificação da mesma e dos envolvidos “é uma tarefa complexa devido à possibilidade de anonimato dos contraventores e ao fato de que as evidências do crime podem estar distribuídas em diversos servidores espalhados pela Internet” (PEREIRA et. al., 2007, p. 2).

Nesse contexto, a investigação digital forense “é um processo que utiliza ciência e tecnologia para analisar objetos digitais e que desenvolva e teste teorias, que podem ser inseridas em um tribunal de justiça, para responder a perguntas sobre eventos ocorridos” (CARRIER, 2005, p. 13, tradução nossa), ou seja, é o processo de investigação com o objetivo de obter evidências digitais confiáveis para determinar a ocorrência de um crime cibernético. Ainda segundo Casey (2011, p. 38, tradução nossa) “Em essência, o processo de exame forense extrai e prepara dados para análise. O processo de exame envolve a tradução, redução, recuperação, organização e busca de dados”.

2.1.1. Processo de Investigação Digital

Tendo em vista as questões referentes ao crime cibernético, a Investigação Digital possui como principal objetivo encontrar elementos que possam demonstrar que existe interligações entre o criminoso, a vítima e a cena do crime, que nesse caso possui uma dimensão digital que deve ser levada em consideração, conforme a figura 1.

Figura 1 - Conexões entre os elementos de uma investigação digital

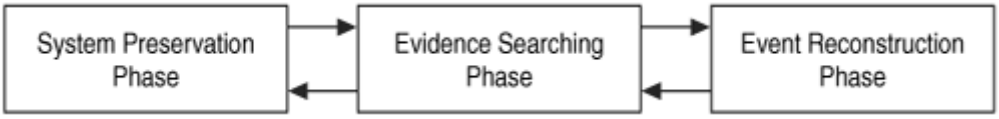


Fonte: Casey (2011, p. 17)

> De acordo com o Princípio de Trocas de Locard, o contato entre dois itens resultará em uma troca. Este princípio aplica-se a qualquer contato em uma cena do crime, inclusive entre um agressor e uma vítima, entre uma pessoa e uma arma, e entre as pessoas e a própria cena do crime. Em suma, sempre haverá evidências da interação, embora em alguns casos possa não ser facilmente detectada (...). Essa transferência ocorre nos domínios físicos e digitais e podem fornecer ligações entre eles. (CASEY, 2011, p. 16, tradução nossa).

Visto que em investigações digitais se sempre se apresentam uma “cena digital do crime que inclui o ambiente digital criado por software e hardware” (CARRIER, 2005, p. 13, tradução nossa) existem alguns métodos para se organizar a análise desse tipo de cenário, e ainda de acordo com o processo proposto por Carrier (2005, p. 13, tradução nossa) “O processo tem três fases principais, que são a preservação do sistema, pesquisa de evidências e reconstrução de eventos”, como exposto na figura 2.

Figura 2 - Principais fases de uma investigação digital



Fonte: Carrier (2005, p. 13)

Ao longo do tempo surgiram várias propostas de métodos de abordar uma investigação digital, que apesar de terem sidos propostos para atender as

especificidades de diferentes contextos apresentam similaridades essenciais, logo, de uma maneira geral, pode-se definir que uma investigação digital confiável e completa está geralmente subdividida em preparação, identificação, preservação, análise e apresentação (CASEY, 2011).

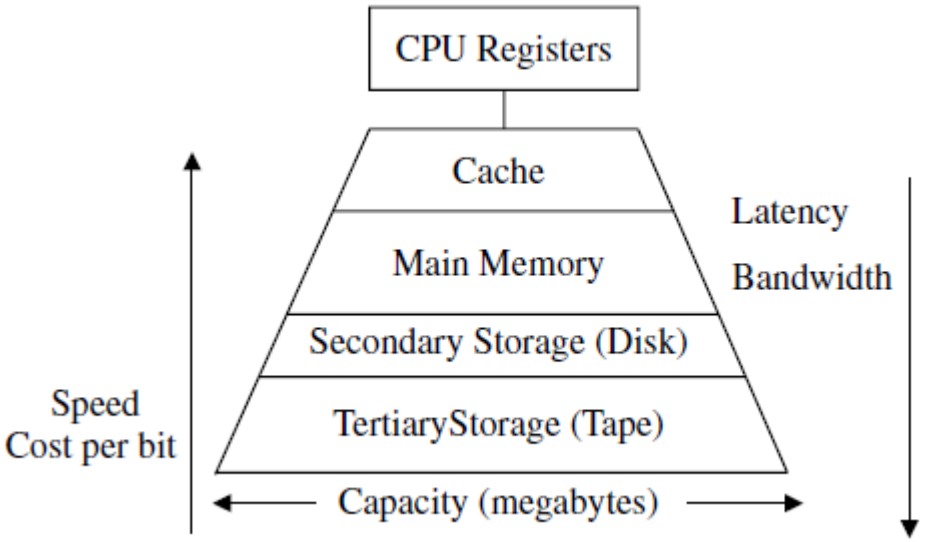
Ainda segundo Casey (2011) a preparação pode ser definida como a elaboração da melhor estratégia ou plano de ação para realizar a investigação, e os recursos necessários para realizar a mesma, a identificação é a etapa a qual são localizadas as fontes de evidência digitais em potencial, a preservação é a etapa aonde são tomadas medidas para preservar a integridade dos artefatos virtuais encontrados, inclusive através de coleta de dados e isolamento de sistemas, a análise é a etapa aonde os artefatos são interpretados para averiguar se os mesmos são evidências digitais relevantes à investigação e a apresentação é o relato das evidências encontradas em formato adequado ao contexto da investigação.

2.2. DEFINIÇÃO DE MEMÓRIAS

Por possuir uma relação direta com a existência e o armazenamento de qualquer artefato virtual em um computador, a compreensão do conceito de memória é um dos pontos mais importantes no entendimento de um processo pericial computacional, uma vez que é o elemento que irá portar em si as eventuais evidências virtuais necessárias para determinar os eventos que estão sendo averiguados por uma determinada investigação digital.

No contexto da tecnologia da informação, memória pode ser definida como “a parte do computador onde são armazenados programas e dados” (TANENBAUM, 2007, p. 39), e no que diz respeito da implementação das memórias em um computador segundo Abd-El-Barr e El-Rewini (2005, p. 107, tradução nossa) “foi reconhecido por Burks, Goldstine e Von Neumann que uma memória de computador deve ser organizada em uma hierarquia. Em tal hierarquia, memórias maiores e mais lentas são usadas para complementar as menores e as mais rápidas”, conforme exposto na figura 3.

Figura 3 - Hierarquia de memória típica de um computador

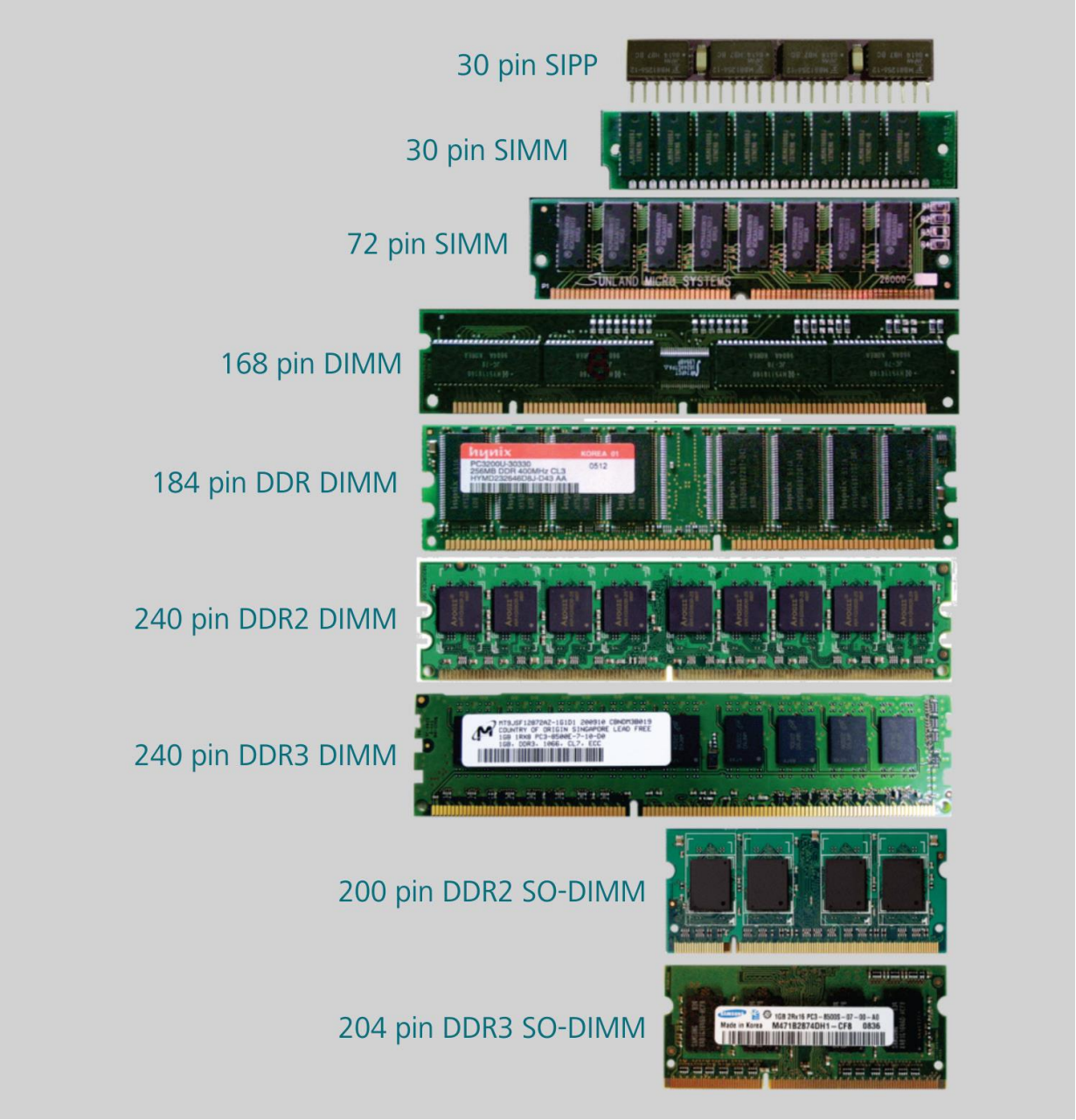


Fonte: Abd-El-Barr e El-Rewini (2005, p. 108)

Seguindo a hierarquia, logo após os registradores da CPU encontra-se a memória de cache, cuja a ideia é definida por Tanenbaum (2007, p. 44) “as palavras de memória usadas com mais frequência são mantidas na cache. Quando a CPU precisa de uma palavra ela examina em primeiro lugar a cache. Somente se a palavra não estiver ali é que ela recorre à memória principal”.

Logo em seguida apresenta-se a memória principal do computador (Main Memory), que é a memória primordial para o funcionamento do mesmo, e nessa categoria pode-se destacar as memórias de acesso aleatório (Random Access Memory ou RAM) e as memórias somente de leitura (Read Only Memory ou ROM). As memórias RAM possuem a característica de poderem sofrer operações de leitura e escrita, e de acordo com Abd-El-Barr e El-Rewini (2005, p. 156, tradução nossa) “a (memória de) acesso aleatório, bem como as memórias de cache, são exemplos de memórias voláteis. Um armazenamento volátil é definido como aquele que perde seu conteúdo quando a energia é desligada”. As memórias RAM são dispostas em módulos de memória que segundo Cristo, Preuss e Franciscatto (2013, p. 46) “é uma pequena placa onde são instalados os encapsulamentos de memória. Essa placa é encaixada na placa-mãe por meio de encaixes (slots) específicos para isso”, como exposto na figura 4.

Figura 4 - Principais módulos de memória



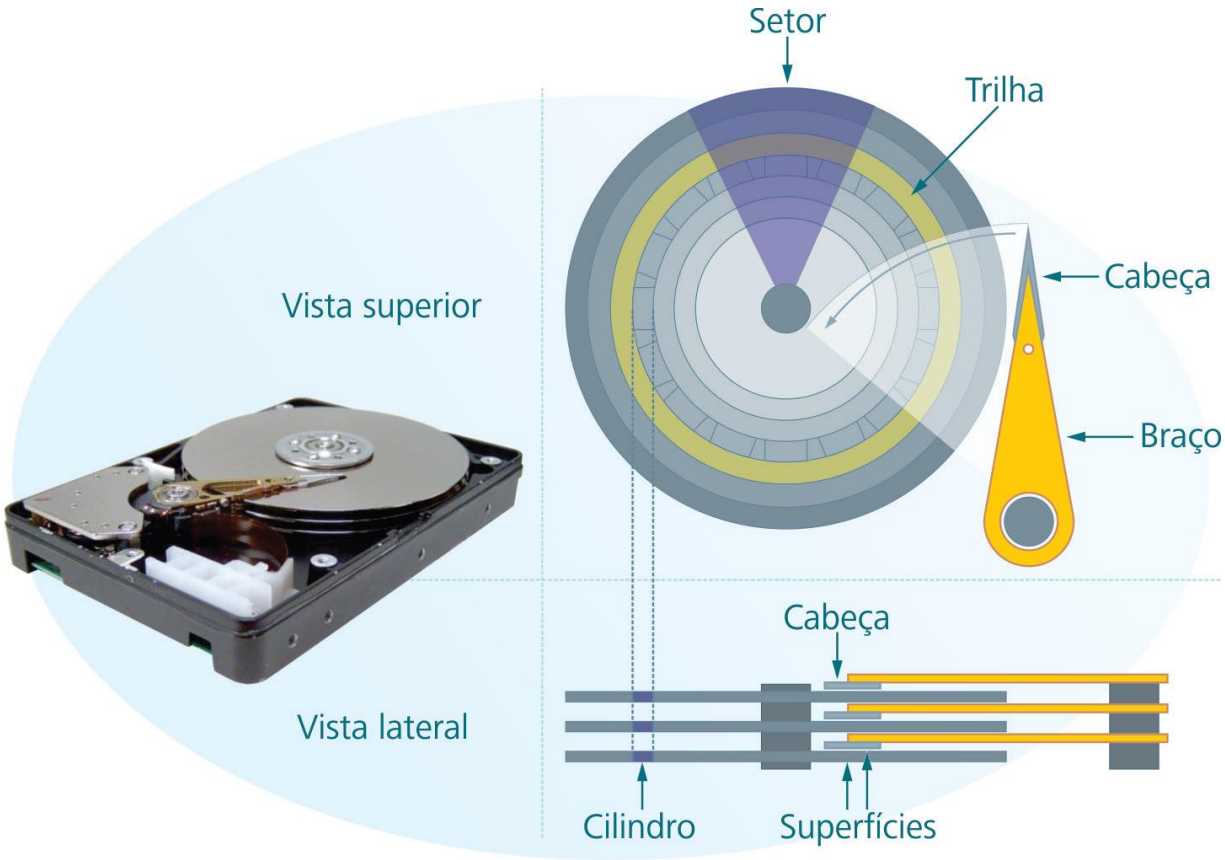
Fonte: Cristo, Preuss e Franciscatto (2013, p. 47)

Já a memória ROM “é um tipo de memória que, em uso normal, aceita apenas operações de leitura, não permitindo a realização de escritas” (CRISTO, PREUSS e FRANCISCATTO, 2013, p. 42) sendo que seu conteúdo geralmente é gravado durante a sua fabricação ou em situações específicas, e é classificada como uma memória não-volátil. Ainda segundo Abd-El-Barr e El-Rewini (2005, p. 156, tradução nossa) “As sub-rotinas de inicialização do sistema informático, o controle de

microcódigo e os cartuchos de videogames são alguns exemplos de software informático que requerem o uso de armazenamento não volátil".

Após a memória principal apresenta-se a memória secundária (Secondary Storage) que conforme Cristo, Preuss e Franciscatto (2013, p. 53) "é o local de armazenamento permanente do computador. Nela ficam depositados os programas e os arquivos dos usuários. A informação precisa ser carregada na memória principal antes de ser tratada pelo processador". A memória secundária geralmente é atrelada ao disco rígido, que é um dispositivo de armazenamento em massa não-volátil que possui em seu interior um ou mais discos magnéticos, como exposto na figura 5.

Figura 5 - Partes de um disco rígido



Fonte: Cristo, Preuss e Franciscatto (2013, p. 54)

> Um disco magnético é composto de um ou mais pratos de alumínio com um revestimento magnetizável (…). Quando uma corrente positiva ou negativa passa pelo cabeçote, ele magnetiza superfície logo abaixo dele, alinhando as partículas magnéticas para a esquerda ou para a direita, dependendo da polaridade da corrente. Quando o cabeçote passa sobre uma área magnetizada, uma corrente positiva ou negativa é induzida nele, o que possibilita a leitura dos bits armazenados antes. (TANENBAUM, 2007, p. 47)

As memórias terciárias (Tertiary Storage) são compostas pelos demais dispositivos externos com função de armazenar dados, tais como discos óticos, fitas magnéticas, cartões de memória e pen-drives (figura 6). Essas memórias são voltadas para o armazenamento em massa de arquivos, e assim como os discos rígidos são consideradas memórias não-voláteis, visto que preservam suas informações mesmo após serem retiradas do computador.

Figura 6 - Exemplos conhecidos de memória terciária

Fonte: Morimoto (2008, online)

2.3. SISTEMAS OPERACIONAIS

Uma vez que este é o elemento que define a maneira que um usuário opera o computador, e por consequência determina por meio do seu respectivo sistema de arquivos o modo o qual a informação existente é manipulada e armazenada, a definição de sistema operacional também é um ponto de alta relevância para um estudo relativo à investigação digital.

Um sistema operacional (SO) de acordo com Tanenbaum (2007, p. 252) “é um programa que, do ponto de vista do programador, acrescenta uma variedade de

novas instruções e características, acima e além do que o nível ISA fornece", aonde ISA se refere à Instruction Set Architecture (Arquitetura do Conjunto de Instruções). Ainda segundo Tanenbaum (2009, p. 2) "os sistemas operacionais realizam basicamente duas funções não relacionadas: fornecer aos programadores de aplicativos (...) um conjunto de recursos abstratos claros em vez de recursos confusos de hardware e gerenciar esses recursos de hardware".

Segundo Deitel H, Deitel P e Choffnes (2005, p. 4) "sistemas operacionais podem ser encontrados em dispositivos que vão de telefones celulares e automóveis a computadores pessoais e computadores de grande porte (mainframes)", e por ser o elemento que fornece o maior suporte para que um usuário possa interagir de forma geral com um computador o mesmo é o componente o qual os indivíduos mais utilizam para realizar suas operações básicas de gerenciamento de informações. Atualmente, apesar da grande variedade de locais aonde um SO pode se apresentar, os sistemas operacionais mais conhecidos e comumente utilizados são os sistemas presentes em computadores pessoais (PC).

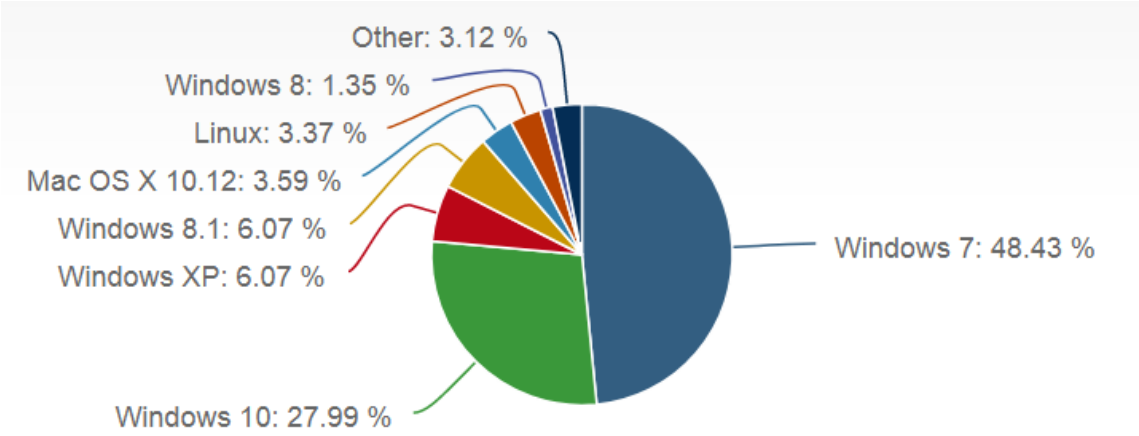
Os sistemas operacionais para PC são os mais conhecidos pelos usuários, tanto pelos comuns quanto por aqueles com conhecimentos específicos, visto que são os sistemas que são voltados para o próprio usuário e tem como objetivo fornecer ao operador do computador as funções necessárias para que este possa gerenciar suas informações digitais, executar aplicações em software e executar tarefas no meio virtual. "Seu trabalho é oferecer uma boa interface para um único usuário. São amplamente usados para processadores de texto, planilhas e acesso à Internet" (TANENBAUM, 2009, p. 21).

Dentre os sistemas operacionais voltados para PC um dos mais conhecidos e utilizados pelo público em geral são os sistemas da família Windows. Esses sistemas operacionais são baseados em um SO denominado MS-DOS (MicroSoft Disk Operating System), e são comercializados pela Microsoft. De acordo com Bellis (2017) este sistema operacional surgiu na década de 80 com a proposta de ser um sistema com interface gráfica compatível com os computadores IBM.

Os SOs da família Windows são largamente utilizados por terem a proposta de serem fáceis de serem operados no ponto de vista do usuário, sem requerer muito conhecimento específico para ser utilizados, e segundo a Net Market Share (2017), no que diz respeito a computadores pessoais (desktop), os sistemas da família Windows possuem a grande maioria de uso neste mercado, sendo que o SO

Windows 7 é o mais utilizado com 48,43% de participação, seguido do Windows 10 com 27,99% de uso, como exposto na figura 7.

Figura 7 - Sistemas Operacionais mais presentes no mercado de PC



Fonte: Net Market Share (2017, online)

Outro exemplo de sistema operacional voltado originalmente para PC é o Linux (ou GNU/Linux) que é um sistema operacional também bastante conhecido, que apesar de ter relativamente se popularizado por apresentar um grande número de distribuições geralmente ainda é utilizado de forma restrita, principalmente por usuários mais voltados para alguma área da TI. O Linux é conhecido também por ser um dos maiores exemplos de projetos de código aberto (open source), e é mantido pela Linux Foundation, uma instituição criada por Linus Torvalds (criador do sistema em questão) para oferecer suporte à comunidade usuária desse SO.

> Fundada em 2000, a Fundação Linux oferece suporte incomparável para comunidades de código aberto através de recursos financeiros e intelectuais, infraestrutura, serviços, eventos e treinamento. Trabalhando juntos, a Fundação Linux e seus projetos formam o investimento mais ambicioso e bem-sucedido na criação de tecnologia compartilhada. (LINUX FOUNDATION, 2017, online, tradução nossa)

O Linux foi desenvolvido com base no sistema UNIX, que é um SO “projetado para tratar múltiplos processos e usuários ao mesmo tempo (...) para ser usado em um ambiente o qual a maioria dos usuários é relativamente sofisticada e engajada em projetos de desenvolvimento de software” (TANENBAUM, 2009, p. 448), o que tornou o próprio Linux um sistema mais voltado para desenvolvedores e estudiosos da área. Atualmente o Linux se apresenta em numerosas distribuições que são

desenvolvidas em projetos separados, porém baseadas no núcleo Linux desenvolvido pela Linux Foundation.

2.4. SISTEMAS DE ARQUIVOS

Os arquivos podem ser definidos como “unidades lógicas de informações criadas por processos. (...) Os arquivos também são uma espécie de espaço de endereçamento, mas eles são usados para modelar o disco e não a memória RAM” (TANENBAUM, 2009, p. 158), logo são essas unidades que contêm as informações, e o modo o qual esses arquivos são armazenados, recuperados e reconhecidos é definido pelo sistema de arquivos, que está atrelado ao sistema operacional presente na unidade.

Sendo assim, segundo Carrier (2005, p. 129, tradução nossa) “os sistemas de arquivos fornecem um mecanismo para que os usuários armazenem dados em uma hierarquia de arquivos e diretórios”, esses mecanismos definem o modo o qual o computador dispõe os dados no sistema e a forma o qual os localiza. Conforme Deitel H, Deitel P e Choffnes (2005, p. 380) “sistemas de arquivo preocupam-se primordialmente com o gerenciamento do espaço de armazenamento secundário, particularmente armazenamento em disco, mas podem acessar dados de arquivos armazenados em outros meios”. Dentre as informações que compõe um sistema de arquivos existem dados que são considerados essenciais para a manipulação dos arquivos presentes, bem como dados que não são essenciais para esse objetivo.

> Os dados essenciais do sistema de arquivos são aqueles que são necessários para salvar e recuperar arquivos. Exemplos desse tipo de dados incluem os endereços onde o conteúdo do arquivo é armazenado, o nome de um arquivo e o ponteiro de um nome para uma estrutura de metadados. Os dados não essenciais do sistema de arquivos são aqueles que estão disponíveis por conveniência, mas não são necessários para a funcionalidade básica de salvar e recuperar arquivos. Os tempos de acesso e as permissões são exemplos deste tipo de dados. (CARRIER, 2005, p. 131, tradução nossa).

Tendo em vista essa classificação, apesar dos dados não essenciais dos sistemas de arquivos fornecerem informações adicionais sobre os arquivos em questão que podem ser relevantes para um processo de investigação digital, percebe-se que a confiabilidade dos dados essenciais tende a ser bem maior, uma vez que essas informações necessariamente precisam ser corretamente dispostas

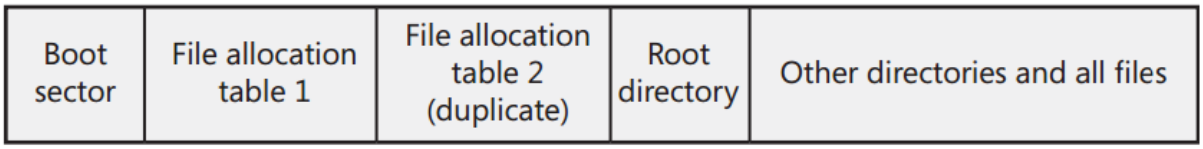
pelo sistema operacional, uma vez que este irá necessitar desses dados para realizar as devidas operações com os arquivos.

Além de dispor os dados de forma que o sistema operacional possa recuperar e manipular, os sistemas de arquivos também têm a função de fornecer ao usuário uma forma adequada de fazer referência aos seus dados, e de acordo com Deitel H, Deitel P e Choffnes (2005, p. 380) “sistemas de arquivos devem exibir independência de dispositivos — os usuários devem poder referir-se a seus arquivos por nomes simbólicos em vez de ter de utilizar nomes de dispositivos físicos”, visão essa que é definida principalmente de acordo com a proposta de cada SO, uma vez que este é o elemento que está mais ligado à forma o qual o sistema de arquivos está disposto no dispositivo de memória.

Dentre os sistemas de arquivos mais simples e comumente usados está o FAT, que “é caracterizado pela tabela de alocação de arquivos (FAT), que é na verdade uma tabela que reside bem no topo do volume” (MICROSOFT, 2016, online), sendo que esse sistema de arquivos, ainda segundo Microsoft (2016, online), “é alocado em clusters, cujo tamanho é determinado pelo tamanho do volume. Quando um arquivo é criado, uma entrada é criada no diretório, e o primeiro cluster contendo dados é estabelecido”, sendo a tabela FAT o componente o qual registra o cluster que foi alocado por último. Esses clusters podem ser definidos como “blocos endereçáveis que muitos formatos de sistema de arquivos usam. O tamanho do cluster é sempre um múltiplo do tamanho do setor (...) Os formatos do sistema de arquivos usam clusters para gerenciar o espaço em disco de forma mais eficiente” (RUSSINOVICH; SOLOMON, 2005, p. 689, tradução nossa).

De acordo com Russinovich e Solomon (2005, p. 692, tradução nossa) “Um volume FAT é dividido em várias regiões (...). A tabela de alocação de arquivos, que dá ao formato do sistema de arquivos FAT seu nome, tem uma entrada para cada cluster em um volume”, sendo as áreas de um volume FAT expostos na figura 8. Ainda segundo Russinovich e Solomon (2005, p. 692, tradução nossa) “como a tabela de alocação de arquivos é crítica para a interpretação bem-sucedida do conteúdo de um volume, o formato FAT mantém duas cópias da tabela”.

Figura 8 - Organização de um volume FAT



Fonte: Russinovich e Solomon (2005, p. 692)

Para que os registros da tabela FAT sejam atualizados é necessário que o leitor do disco retorne para o início físico do dispositivo de memória, operação essa que torna o gerenciamento dos arquivos uma tarefa mais demorada, o que faz os dispositivos com esse sistema de arquivos apresentarem uma perda de desempenho conforme o tamanho total da memória é maior, e de acordo com Microsoft (2016, online), “à medida que o tamanho do volume aumenta, o desempenho do FAT diminui rapidamente”.

Foram desenvolvidas versões do FAT com suporte para operar com clusters maiores, como o FAT16 e o FAT32, porém mesmo assim as limitações desse sistema de arquivos impediam o gerenciamento de memórias maiores, principalmente devido à perda de performance, e nesse contexto surgiu o NTFS, que segundo Carrier (2005, p. 199, tradução nossa) “foi projetado pela Microsoft e é o sistema de arquivos padrão para Microsoft Windows NT, Windows 2000, Windows XP (...). NTFS é um sistema de arquivos muito mais complexo do que o FAT porque possui muitos recursos e é muito escalável”.

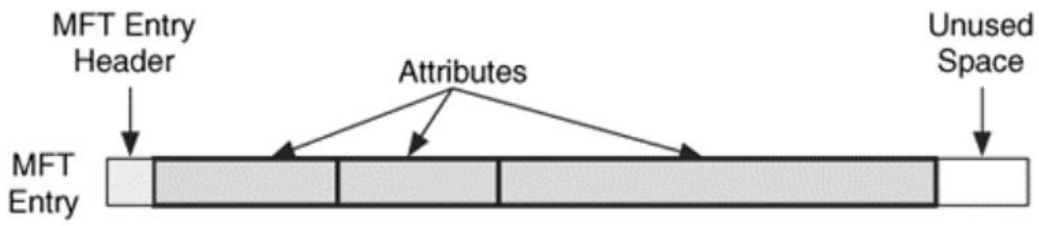
> O sistema de arquivos NTFS é o formato do sistema de arquivos nativo do Windows. O NTFS usa números de cluster de 64 bits. Essa capacidade oferece aos NTFS a capacidade de endereçar volumes de até 16 exabytes (16 bilhões de GB). No entanto, o Windows limita o tamanho de um volume NTFS ao endereçável com clusters de 32 bits, que é ligeiramente inferior a 256 TB (usando clusters de 64 KB). (...) NTFS também suporta 2^{32}-1 arquivos por volume. O formato NTFS permite arquivos com 16 exabytes de tamanho, mas a implementação limita o tamanho máximo do arquivo a 16 TB (RUSSINOVICH; SOLOMON, 2005, p. 694, tradução nossa).

Conforme Carrier (2005, p. 199, tradução nossa) “um dos conceitos mais importantes na compreensão do design do NTFS é que dados importantes são alocados aos arquivos. Isso inclui os dados básicos do sistema de arquivos que normalmente são ocultos por outros sistemas de arquivos”, o que permite a alocação da informação essencial do sistema de arquivos em qualquer área do volume, uma vez que essa informação também é considerada um arquivo, e ainda segundo

Carrier (2005, p. 199, tradução nossa) “um sistema de arquivos NTFS não possui um layout específico, como outros sistemas de arquivos. Todo o sistema de arquivos é considerado uma área de dados, e qualquer setor pode ser alocado para um arquivo”.

O NTFS possui uma estrutura importante que é denominada MFT (Master File Table) que “é o coração de NTFS porque contém as informações sobre todos os arquivos e diretórios. Cada arquivo e diretório tem pelo menos uma entrada na tabela” (CARRIER, 2005, p. 199, tradução nossa) e essa tabela é utilizada para manter os registros sobre a localização dos arquivos e as operações realizadas, cuja a entrada está representada na figura 9, e por segurança “no NTFS, são mantidas várias cópias (o número depende do tamanho do volume) da Tabela Mestra de Arquivos” (MICROSOFT, 2016, online), o que torna o NTFS um sistema de arquivos mais consistente e recuperável em comparação com os sistemas FAT, que possui somente uma cópia de segurança da Tabela de Alocação de Arquivos.

Figura 9 - Entrada padrão de uma MFT



Fonte: Carrier (2005, p. 200)

Os sistemas operacionais Linux possuem suporte para um grande número de sistemas de arquivos, porém existe um sistema de arquivo padrão que foi projetado especificamente para esse SO, que é o EXT. Segundo Both (2017, online, tradução nossa) “O sistema de arquivos EXT original (Extended) foi escrito por Rémy Card e lançado com Linux em 1992 para superar algumas limitações de tamanho do sistema de arquivos Minix”. O EXT foi baseado no UNIX File System (UFS), e herdou deste sistema várias características que proporcionaram uma boa confiabilidade e performance a esse sistema de arquivos. De acordo com Carrier (2005, p. 283, tradução nossa) “cópias de estruturas de dados importantes são armazenadas em todo o sistema de arquivos e todos os dados associados a um

arquivo estão localizados para que as cabeças do disco rígido não precisem viajar muito ao lê-los”.

> As informações de layout básicas do ExtX são armazenadas na estrutura de dados do superbloco, que está no início do sistema de arquivos. O conteúdo do arquivo é armazenado em blocos, que são grupos de setores consecutivos. Os metadados para cada arquivo e diretório são armazenados em uma estrutura de dados chamada inode, que tem um tamanho fixo e está localizada em uma tabela de inodes. Há uma tabela inode em cada grupo de blocos. O nome de um arquivo é armazenado em uma estrutura de entrada de diretório, que está localizada nos blocos alocados ao diretório pai do arquivo. As estruturas de entrada de diretório são estruturas de dados simples que contêm o nome do arquivo e um ponteiro para a entrada do inode do arquivo (CARRIER, 2005, p. 283, tradução nossa).

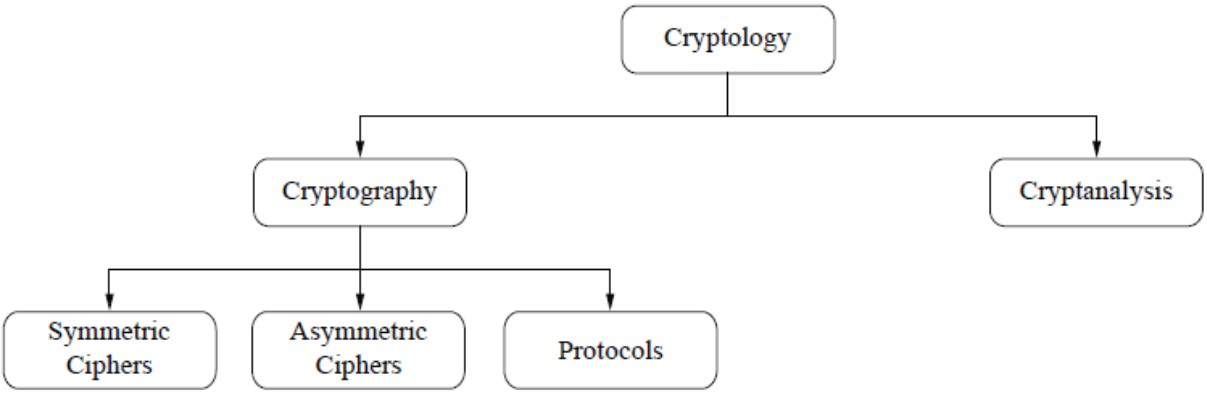
O EXT foi melhorado ao longo do tempo para atender às funcionalidades das novas distribuições Linux, e atualmente existem o EXT2, EXT3 e o EXT4 que é a versão mais atualizada desse sistema de arquivos a qual foi projetada para melhorar bastante a confiabilidade e performance, mesmo em sistemas que necessitem fazer operações em tempos mais críticos, e de acordo com Both (2017, online, tradução nossa) “para atender a vários requisitos de missão crítica, os timestamps do sistema de arquivos foram aprimorados com a adição de intervalos até nanosegundos”. O EXT4 também apresenta melhoras na forma de alocar o espaço em disco, sendo que “o EXT4 usa estratégias funcionais, como alocação atrasada, para permitir que o sistema de arquivos colete todos os dados que estão sendo gravados no disco antes de atribuir espaço a ele” (BOTH, 2017, online, tradução nossa).

2.5. DEFINIÇÃO DE CRIPTOGRAFIA

De acordo com Stallings (2014, p. 21) “O processo de converter um texto claro em um texto cifrado é conhecido como cifração ou encriptação (...). Os muitos esquemas utilizados para a encriptação constituem a área de estudo conhecida como criptografia”, ou seja, a criptografia é um conjunto de técnicas que visa tornar uma informação que está apresentada em um formato em que pode ser comumente interpretada para uma forma a qual o conteúdo só seja compreendido por interlocutores específicos, os quais possuem os dados necessários para reverter a encriptação.

Segundo Paar e Pelzl (2009, p. 2, tradução nossa) “a criptografia é um negócio bastante antigo, com exemplos iniciais que remontam há cerca de 2000 anos de idade, quando hieróglifos ‘secretos’ fora do padrão foram usados no antigo Egito”. Como exposto na figura 10 a criptografia juntamente com a criptoanálise compõem uma área maior denominada criptologia, sendo que a criptoanálise “é a ciência e às vezes a arte de quebrar criptosistemas” (PAAR; PELZL, 2009, p. 3, tradução nossa), e de acordo com Wobst (2007, p. 62, tradução nossa) “a criptoanálise destina-se a revelar tanta informação sobre o texto como possível sem saber a chave secreta”.

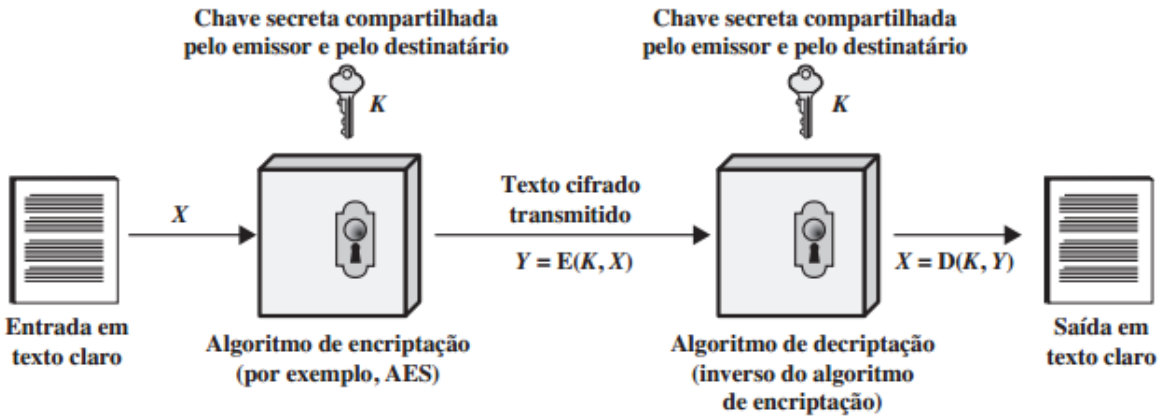
Figura 10 - As áreas que compõem a Criptologia



Fonte: Paar e Pelzl (2009, p. 3)

As cifras simétricas (Symmetric Ciphers) são a forma mais conhecida e usada de criptografia, e nesse metétodo segundo Paar e Pelzl (2009, p. 3, tradução nossa) “duas partes possuem um método de criptografia e descriptografia para o qual eles compartilham uma chave secreta. Toda criptografia dos tempos antigos até 1976 foi exclusivamente baseada em métodos simétricos”. O método da cifra simétrica, conforme exposto na figura 11, tem sua segurança atrelada à eficiência do algoritmo de encriptação, porém “com o uso da encriptação simétrica, o principal problema de segurança consiste de manter o sigilo da chave” (STALLINGS, 2014, p. 22).

Figura 11 - Modelo da encriptação simétrica



Fonte: Stallings (2014, p. 21)

A cifra Assimétrica (Asymmetric Ciphers), ou cifra de chave pública, é um tipo de criptografia que surgiu com uma proposta diferente, e segundo Paar e Pelzl (2009, p. 3, tradução nossa) “em 1976, um tipo de cifra completamente diferente foi introduzido por Whitfield Diffie, Martin Hellman e Ralph Merkle. Na criptografia de chave pública, um usuário possui uma chave secreta como na criptografia simétrica, mas também uma chave pública”.

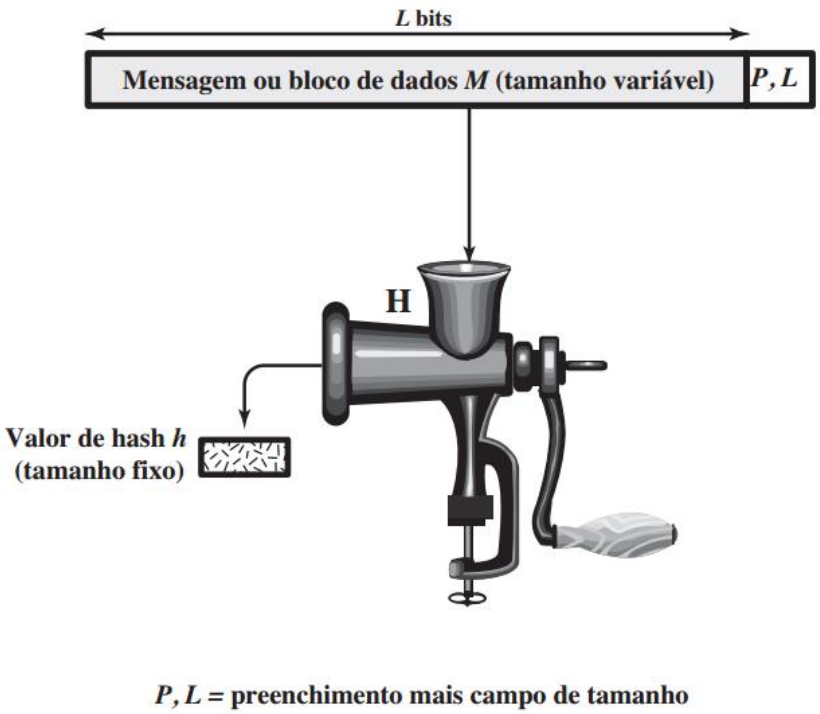
De acordo com Stallings (2014, p. 200) “Por um lado, os algoritmos de chave pública são baseados em funções matemáticas, em vez de substituição e permutação. Mais importante, a criptografia de chave pública é assimétrica, envolvendo o uso de duas chaves separadas”, ou seja, são usadas duas chaves no processo de comunicação, uma para criptografar a mensagem e outra para descriptografar as informações cifradas, sendo que uma das duas chaves é disposta em local acessível, sendo essa a chave pública.

Os protocolos criptográficos (Cryptographic Protocols) são os protocolos de aplicação dos algoritmos de criptografia, tanto dos simétricos quanto os assimétricos, visto que esses algoritmos “podem ser vistos como blocos de construção com quais aplicativos como a comunicação segura na Internet podem ser realizados” (PAAR; PELZL, 2009, p. 4, tradução nossa).

No sentido dos protocolos destaca-se a função hash, que ainda de acordo com Paar e Pelzl (2009, p. 293, tradução nossa) “para uma mensagem específica (...) o valor de hash podem ser vistos como a impressão digital de uma mensagem, ou seja, uma representação única de uma mensagem”, e segundo Stallings (2014, p.

247) “uma função de hash aceita uma mensagem de tamanho variável M como entrada e produz um valor de hash de tamanho fixo h = H(M)” como representado na figura 12, e essas propriedades tornam esse protocolo bastante importante na validação de dados e arquivos.

Figura 12 - Representação da função Hash



Fonte: Stallings (2014, p. 247)

2.6. DEFINIÇÕES DE SOFTWARE LIVRE E SOFTWARE DE CÓDIGO ABERTO

A definição de Software Livre e Código Aberto também é um ponto relevante para este estudo, uma vez que grande parte das ferramentas utilizadas para realizar as aplicações das técnicas forenses e anti-forenses apresentadas ao longo da monografia se enquadram nestas categorias por serem distribuídos de forma gratuita sob suas respectivas licenças para que possam ser usados livremente para a finalidade de seus usuários.

Segundo a Free Software Foundation (2017, online, tradução nossa) “O movimento do software livre foi iniciado em 1983 pelo cientista da computação Richard M. Stallman, quando lançou um projeto chamado GNU, que significa “GNU is not UNIX”, para substituir o sistema operacional UNIX”. Os softwares livres, segundo GNU (2017, online) são todos baseados nas quatro liberdades, que são a liberdade de executar o programa da forma que o usuário desejar, independente do propósito (liberdade 0), a liberdade de analisar e estudar o código fonte do programa (liberdade 1), a liberdade de distribuir cópias dos programas (liberdade 2) e a liberdade de modificar e distribuir cópias dos softwares modificados (liberdade 3), e de acordo com Stallman (2015, p. 3, tradução nossa) “"Software livre" significa software que respeita a liberdade e a comunidade dos usuários. Grosseiramente, isso significa que os usuários têm a liberdade de executar, copiar, distribuir, estudar, alterar e melhorar o software”.

De acordo com a Open Source Initiative (2012, online, tradução nossa) “A OSI foi fundada em conjunto por Eric Raymond e Bruce Perens no final de fevereiro de 1998, com Raymond como seu primeiro presidente, Perens como vice-presidente e um conselho de administração inicial”, com o objetivo bem semelhante ao da FSF, porém com algumas diferenças em relação às licenças dos softwares, que são mais brandas do que as licenças de software livre no que diz respeito à possibilidade de aplicação de restrições em versões modificadas dos softwares de código aberto, o que não é permitido no software livre.

A maior parte das diferenças entre os programas classificados como Software Livre e os de Código Aberto são mais referentes ao ponto de vista das duas comunidades e da filosofia das mesmas, conforme Stallman (2015, p. 76, tradução nossa) “o Código Aberto é uma metodologia de desenvolvimento; O software livre é um movimento social. Para o movimento do software livre, o software livre é um imperativo ético, um respeito essencial pela liberdade dos usuários”, porém a grande maioria dos programas que são abordados por uma das organizações também são amparados pela outra, o que resulta na cooperação de membros dos dois movimentos em projetos de desenvolvimento de softwares. “Na prática, o código aberto representa critérios um pouco mais fracos do que os de software livre (...). Todo o software livre existente poderia ser qualificado como código aberto. Quase todo o software de código aberto é um software livre, mas há exceções” (STALLMAN, 2015, p. 76, tradução nossa).

2.7. CONCLUSÃO

No presente capítulo foram abordadas as definições teóricas necessárias para poder se executar de forma correta e eficiente uma investigação digital, os conceitos essenciais para poder compreender a natureza dos sistemas de arquivos e como eles organizam as informações dos usuários, bem como a relevância da identificação do sistema operacional, uma vez que é este quem define o sistema de arquivos a ser aplicado na memória. Foram também realizados a caracterização da criptografia, e a forma como a mesma se apresenta no meio virtual atualmente, bem como dos programas de Código Aberto e os Softwares Livres, que possuem no momento uma grande relevância em todas as áreas de estudo da tecnologia da informação.

3. METODOLOGIA DA EXECUÇÃO DAS APLICAÇÕES

Neste capítulo serão apresentados as ferramentas e artefatos utilizados para realizar os experimentos referentes às técnicas anti-forenses para obstrução de recuperação de dados. De forma geral, foi utilizado um sistema operacional comumente usado em PCs para se aplicar as técnicas anti-forenses em seu ambiente e foi utilizado outro sistema voltado para perícia e segurança para realizar uma análise pericial no sistema anterior para poder avaliar a efetividade de cada técnica.

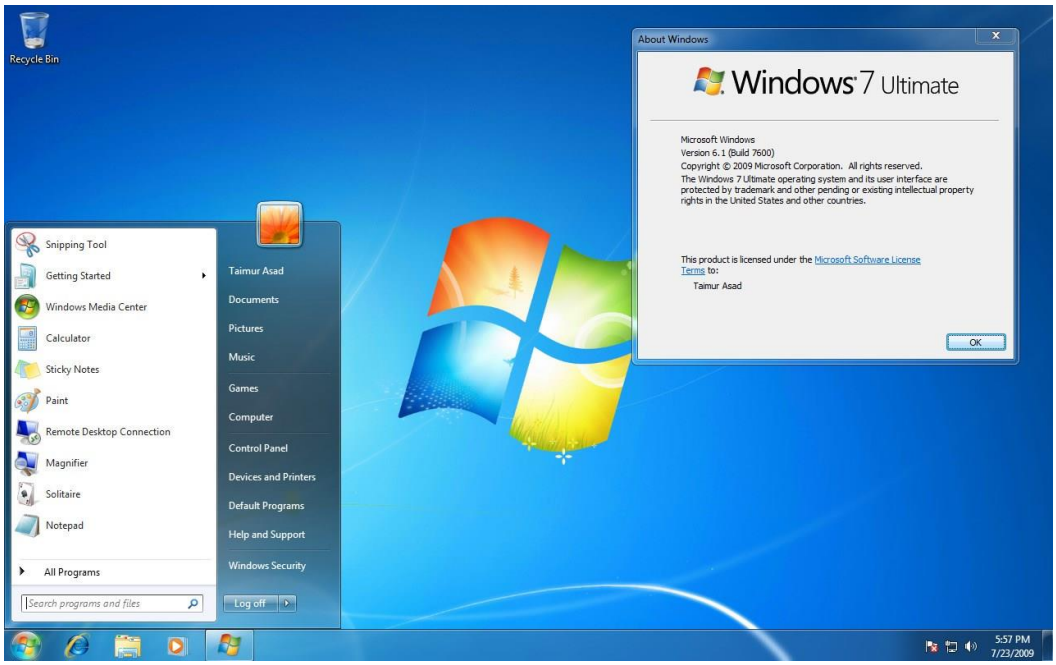
Para representar o computador pessoal foi utilizado o sistema operacional Windows 7, devido ao fato do mesmo ser o sistema operacional mais utilizado nesse contexto (ver figura 7), e neste PC serão utilizadas as técnicas de ocultamento de dados, através de esteganografia, uso do espaço de file slack e usando os Alternate Data Streams presentes no sistema de arquivos NTFS, e posteriormente um teste de destruição segura de dados utilizando o software Eraser.

Para testar a eficiência das técnicas anti-forenses usadas no PC de teste foi utilizado o sistema operacional Kali Linux, devido ao fato deste ter sido projetado, dentre outras finalidades, para realização de perícias e análises, possuindo assim um grande número de ferramentas e pacotes para estas tarefas, as quais serão usadas algumas ferramentas relacionadas à análise de memórias e arquivos e à recuperação de arquivos.

3.1. WINDOWS 7

O Windows 7 é um sistema operacional da família Windows criado pela Microsoft, sendo lançado após o Windows Vista. É ainda o sistema operacional mais utilizado em PCs atualmente, mesmo após o lançamento do Windows 8, Windows 8.1 e do Windows 10. Assim como os outros sistemas operacionais da sua família, o Windows 7 é baseado no MS-DOS, e o seu sistema de arquivos é o NTFS, sendo a sua interface representada na figura 13.

Figura 13 - Tela Inicial do Windows 7

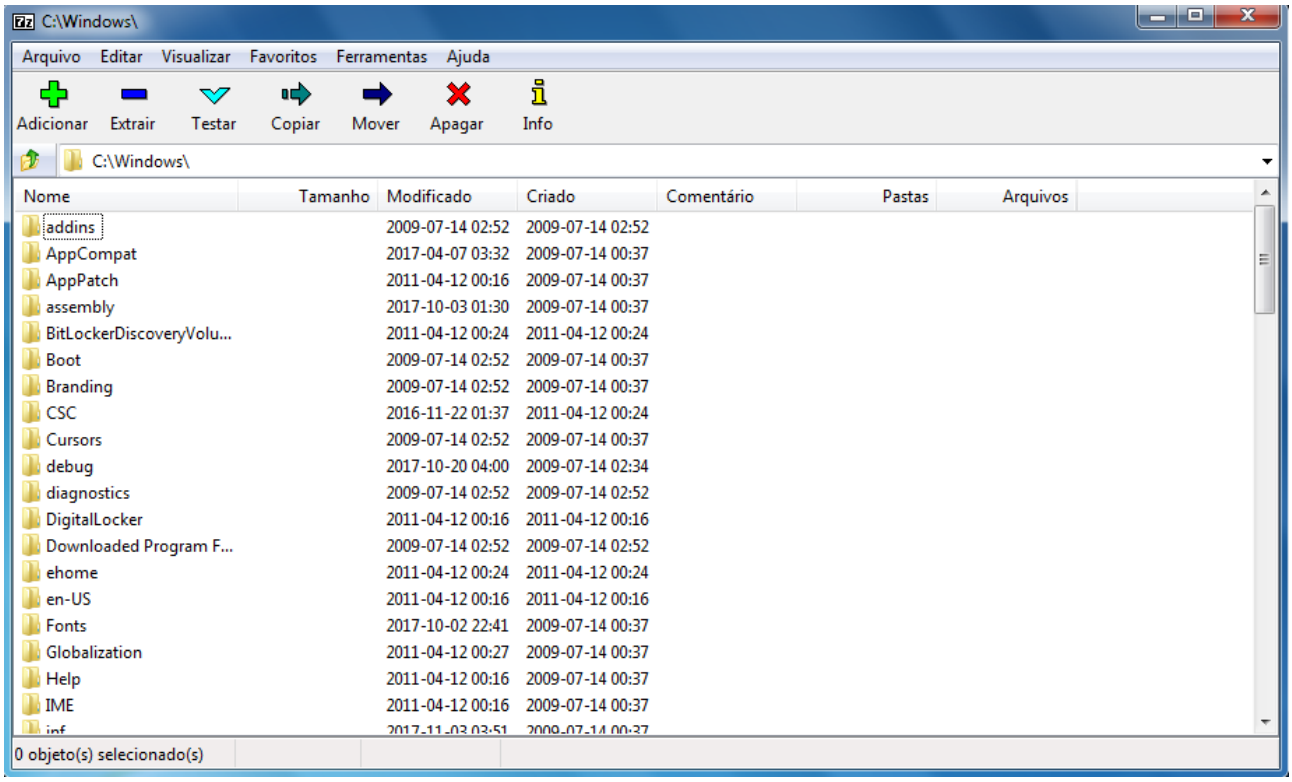


Fonte: alternativeTo (2017, online)

3.1.1. 7zip

O 7zip é um programa compactador de arquivos, e de acordo com Pavlov (2016, online) é um programa de Código Aberto disponível para várias plataformas, incluindo os sistemas baseados em Linux e o Windows, sendo que o 7zip é capaz de trabalhar com uma grande quantidade de tipos de arquivos compactados. Ainda segundo Pavlov (2016, online) além da eficiência deste compactador ser superior do que a maioria dos outros compactadores comerciais, o 7zip fornece a possibilidade de aplicar criptografia AES-256 em arquivos compactados em formatos ZIP e 7Z, e sua interface está representada na figura 14.

Figura 14 - Interface do gerenciador do 7zip no Windows 7

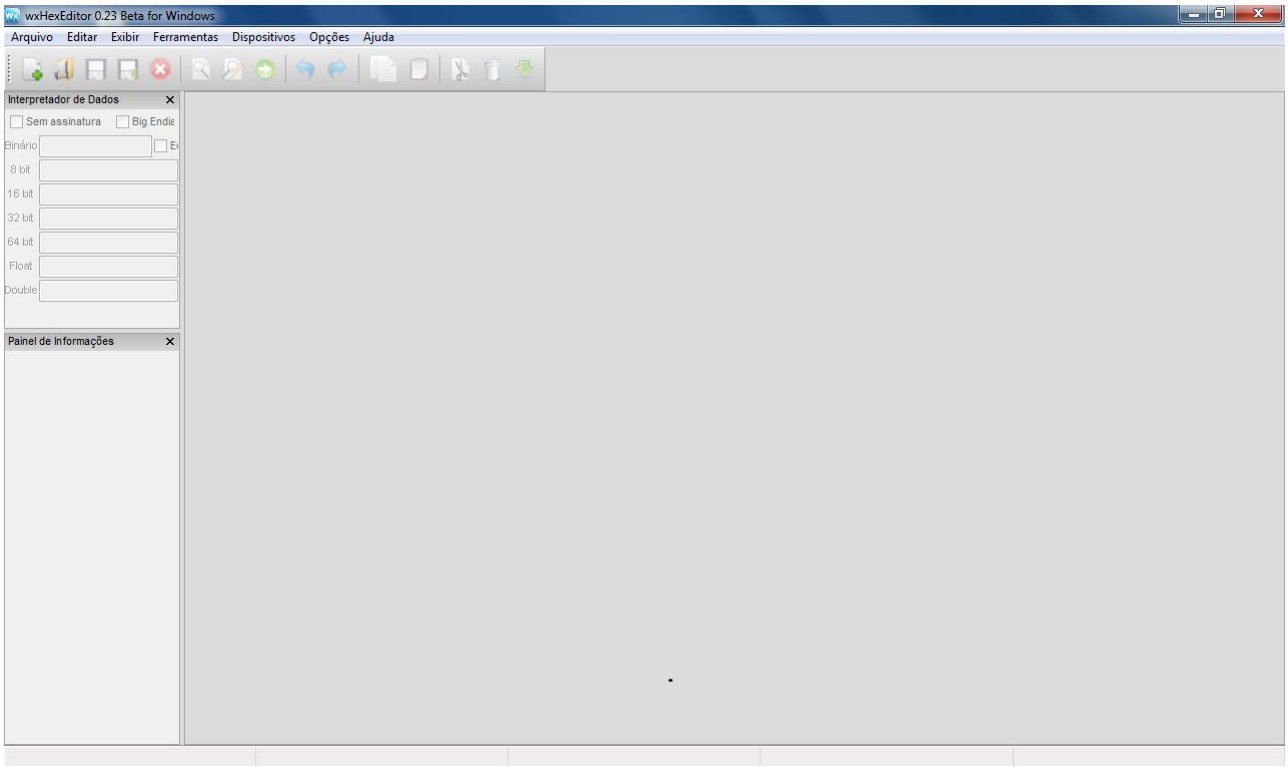


Fonte: Autor (2017)

3.1.2. wxHexEditor

O wxHexEditor é um software que permite editar arquivos e memórias por meio de códigos hexadecimais. Este programa, cuja a interface está representada na figura 15, traduz o conteúdo aberto pelo mesmo para valores em hexadecimal que podem ser alterados pelo operador, o que permite que o mesmo modifique este conteúdo no nível mais baixo da informação, sendo deste modo uma ferramenta importante para análise avançada de arquivos e memórias, bem como para engenharia reversa de software. O wxHexEditor está disponível sob a licença GPL para as plataformas Linux, Windows e OSX.

Figura 15 - Tela inicial do wxHexEditor no Windows 7

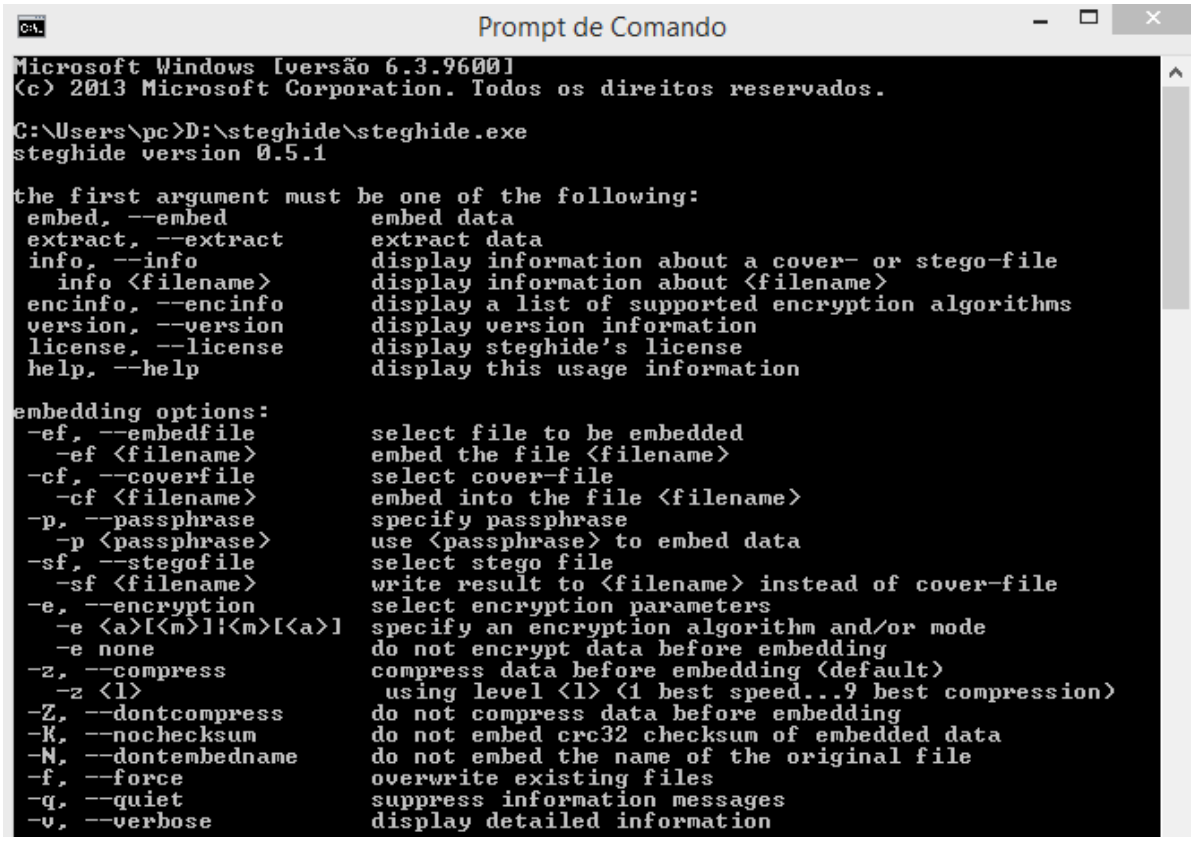


Fonte: Autor (2017)

3.1.3. Steghide

De acordo com Sourceforge (2003, online, tradução nossa) “Steghide é um programa de esteganografia que é capaz de ocultar dados em vários tipos de arquivos de imagem e áudio”, sendo um considerado um Software Livre por ser distribuído de acordo com a GPL. O software não possui interface gráfica está disponível para os sistemas baseados em Linux, aonde pode ser operado pelo Terminal, e para Windows, aonde pode ser utilizado pelo prompt de comandos, como mostra a figura 16.

Figura 16 - Steghide no Windows

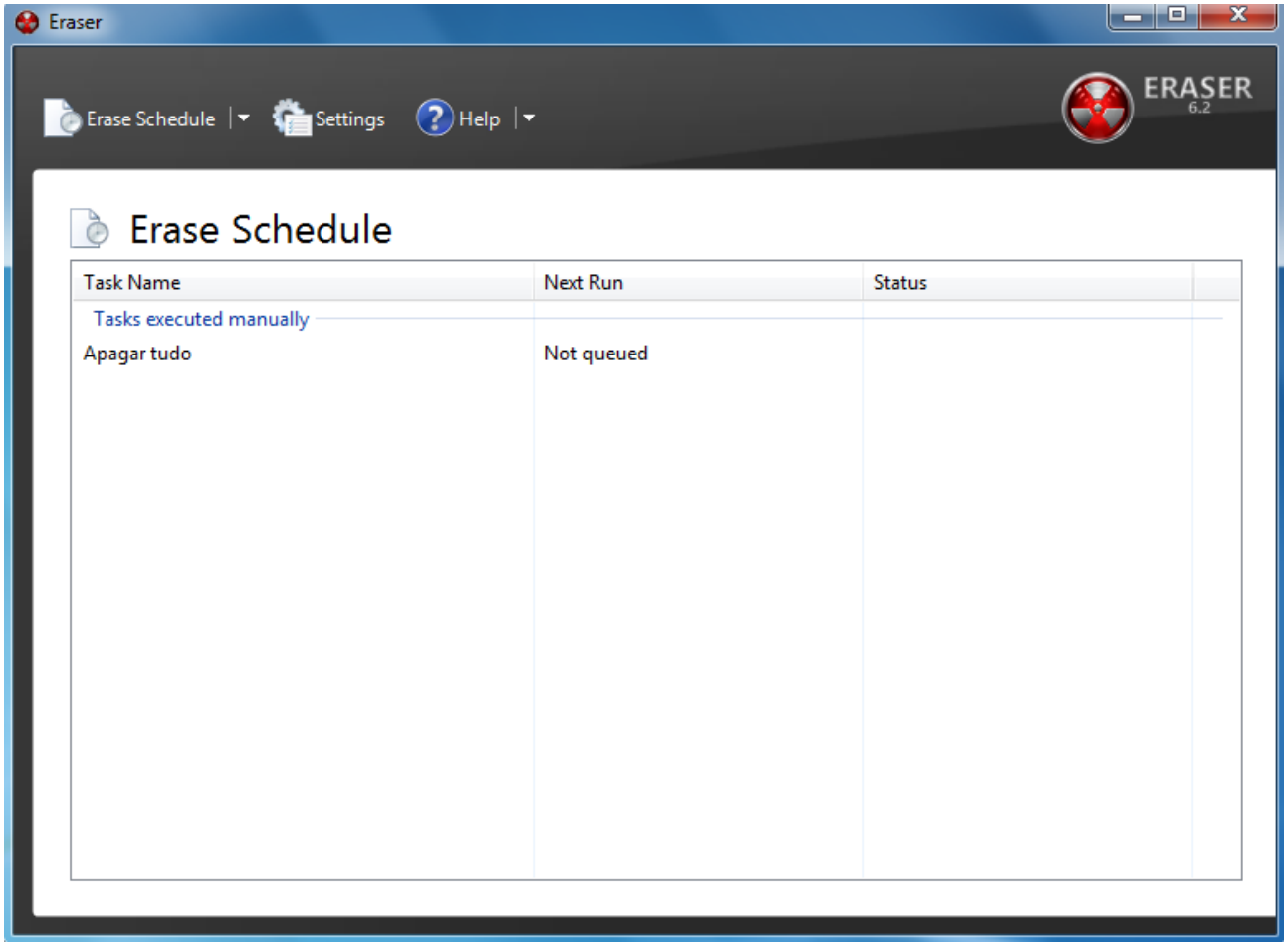


Fonte: Autor (2017)

3.1.4. Eraser

O Eraser “é uma ferramenta de segurança avançada para o Windows que permite que você remova completamente os dados confidenciais do seu disco rígido substituindo-o várias vezes com padrões cuidadosamente selecionados” (ERASER, 2017, online, tradução nossa). Com sua interface representada na figura 17 este programa possui várias funções referentes à destruição de dados e limpeza de memória, e apesar de ser um software desenvolvido propriamente para os SOs da família Windows este programa é um Software Livre distribuído nos termos da GPL.

Figura 17 - Eraser no Windows 7



Fonte: Autor (2017)

3.2. KALI LINUX

O Kali Linux é um sistema operacional baseado no SO Debian, e é um sistema voltado para Pentesting e segurança digital em diversos aspectos. De acordo com Offensive Security (2017, online, tradução nossa) “Kali contém várias centenas de ferramentas, voltadas para várias tarefas de segurança da informação, como teste de penetração, pesquisa de segurança, forense computacional e engenharia reversa”, sendo um sistema projetado para facilitar a execução de auditorias, pentestings, perícias e outras análises que suas ferramentas permitem realizar, e sua interface está representada na figura 18.

Figura 18 - Kali Linux (versão 2017.2)

Fonte: Autor (2017)

Além das ferramentas específicas que o Kali Linux contém, o mesmo também possui suporte para executar a grande maioria dos programas que são suportados pelo SO Debian, o que permite uma grande possibilidade de customização do sistema para melhor atender as necessidades do usuário, e o sistema operacional é distribuído em variantes para computadores 32 bits, 64 bits, arquitetura ARM e para máquinas virtuais (VMware e Virtual Box).

Assim como a maioria dos sistemas voltados para a segurança e perícia, o Kali Linux pode ser utilizado em um computador sem a necessidade de ser instalado no disco rígido, ele pode ser executado a partir de uma memória externa bootável, desde que a BIOS do computador suporte boot para esse tipo de memória, sendo que ele pode ser inicializado de formas específicas conforme o objetivo do seu uso, incluindo um “modo forense”, como exposto na figura 19.

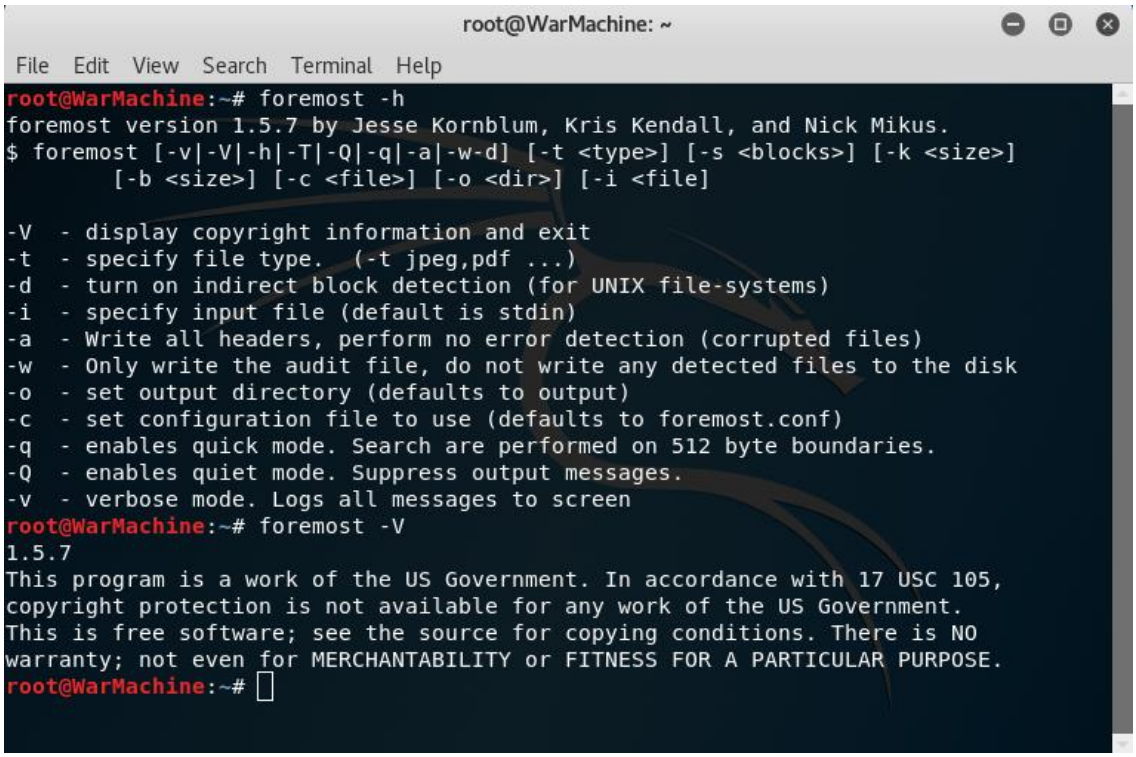
Figura 19 - Tela de inicialização de memória de boot com Kali Linux

Fonte: Autor (2017)

3.2.1. Foremost

O programa Foremost de acordo com Offensive Security (2014, online, tradução nossa) "é um programa forense para recuperar arquivos perdidos com base em seus cabeçalhos, rodapés e estruturas internas de dados. O Foremost pode funcionar em arquivos de imagem (...) ou diretamente em uma unidade". É um programa sem interface gráfica executado pelo Terminal de comandos, como mostra a figura 20, e está sob a licença de domínio público.

Figura 20 - Ferramenta Foremost

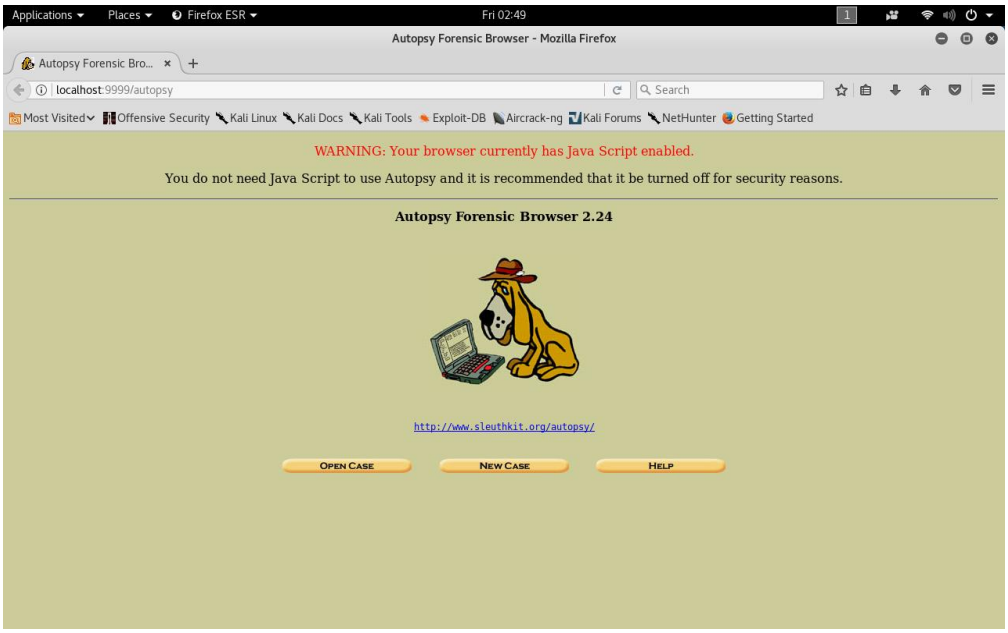


Fonte: Autor (2017)

3.2.2. Autopsy

Autopsy é uma ferramenta de perícia digital que fornece uma plataforma para análise de uma memória ou uma imagem de uma memória, sendo uma interface gráfica para um conjunto de ferramentas denominada Sleuth Kit, como demonstrado na figura 21, que segundo Carrier (2017, online, tradução nossa) “é uma biblioteca e uma coleção de ferramentas de linha de comando que permitem que você investigue imagens de disco. A funcionalidade principal do TSK permite analisar o volume e os dados do sistema de arquivos”.

Figura 21 - Tela inicial do Autopsy



Fonte: Autor (2017)

Essa ferramenta permite localizar arquivos existentes em uma memória, inclusive arquivos que foram apagados de formas comuns pelos usuários, e permite também realizar análises com base em hashs que o usuário possa possuir, extração de arquivos referentes à possíveis navegadores presentes na memória, busca de arquivos por termos relevantes, dentre outras funções, sendo que a distribuição do Autopsy é realizada embasada por uma licença de software livre.

3.3. CONCLUSÃO

As ferramentas apresentadas possuem funcionalidades que abrangem os testes definidos na metodologia, o que torna possível a execução das técnicas de modo que estas aplicações possam ser utilizadas para embasar os tópicos abordados nesse estudo. Do mesmo modo, o sistema operacional escolhido para criar o cenário comum para um computador pessoal torna o ambiente criado para a simulação bastante próximo do que realmente está presente nesse contexto, uma vez que o mesmo é largamente utilizado pela grande maioria dos usuários de computadores pessoais.

4. CAMUFLAGEM DE DADOS

No presente capítulo serão abordas as aplicações das técnicas anti-forenses voltadas para camuflagem e ocultamento de informação confidencial. Tais técnicas consistem em principalmente associar uma informação ou um arquivo a um outro arquivo de uma forma que torne dificultosa a detecção deste conteúdo para os indivíduos que não são os portadores originais desses dados, geralmente adicionando criptografia para tornar a averiguação externa do conteúdo mais improvável e difícil.

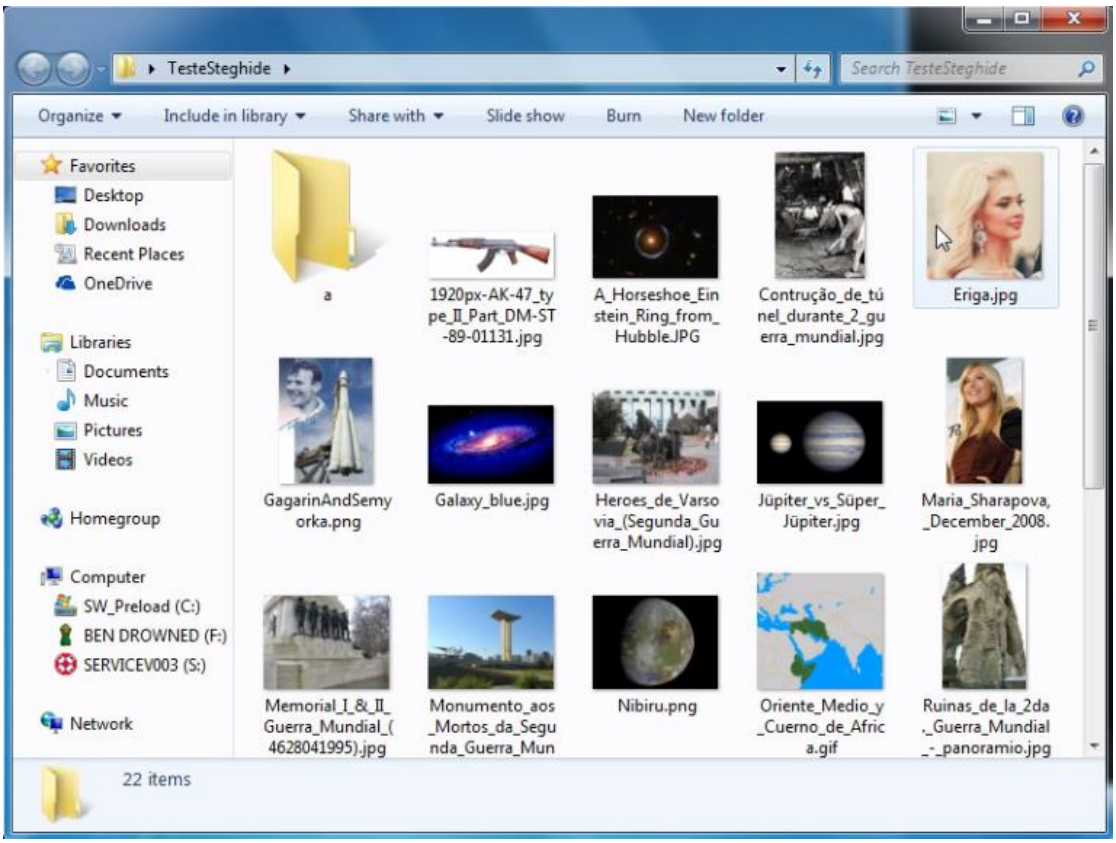
4.1. ESTEGANOGRAFIA

Esteganografia é uma das técnicas de ocultação de informação mais antigas existente, e segundo Julio, Brazil e Albuquerque (2007, p. 55) “a palavra significa a arte da escrita escondida (estegano = esconder e grafia = escrita)”. Foi aplicada de diversas formas em diversos contextos ao longo da história, e a mesma manteve a sua usabilidade de forma eficiente ao ser aplicada no meio digital para esconder informação digital em outros tipos de arquivos para evitar a visualização destes dados por indivíduos que não conhecem a existência dos mesmos.

4.1.1. Aplicação da Esteganografia

Para o presente teste foi utilizado o software Steghide para ocultar informações em uma imagem no formato JPG. Como exposto na figura 22, foi escolhida uma imagem dentre várias separadas para o teste, denominada “Eriga.jpg”.

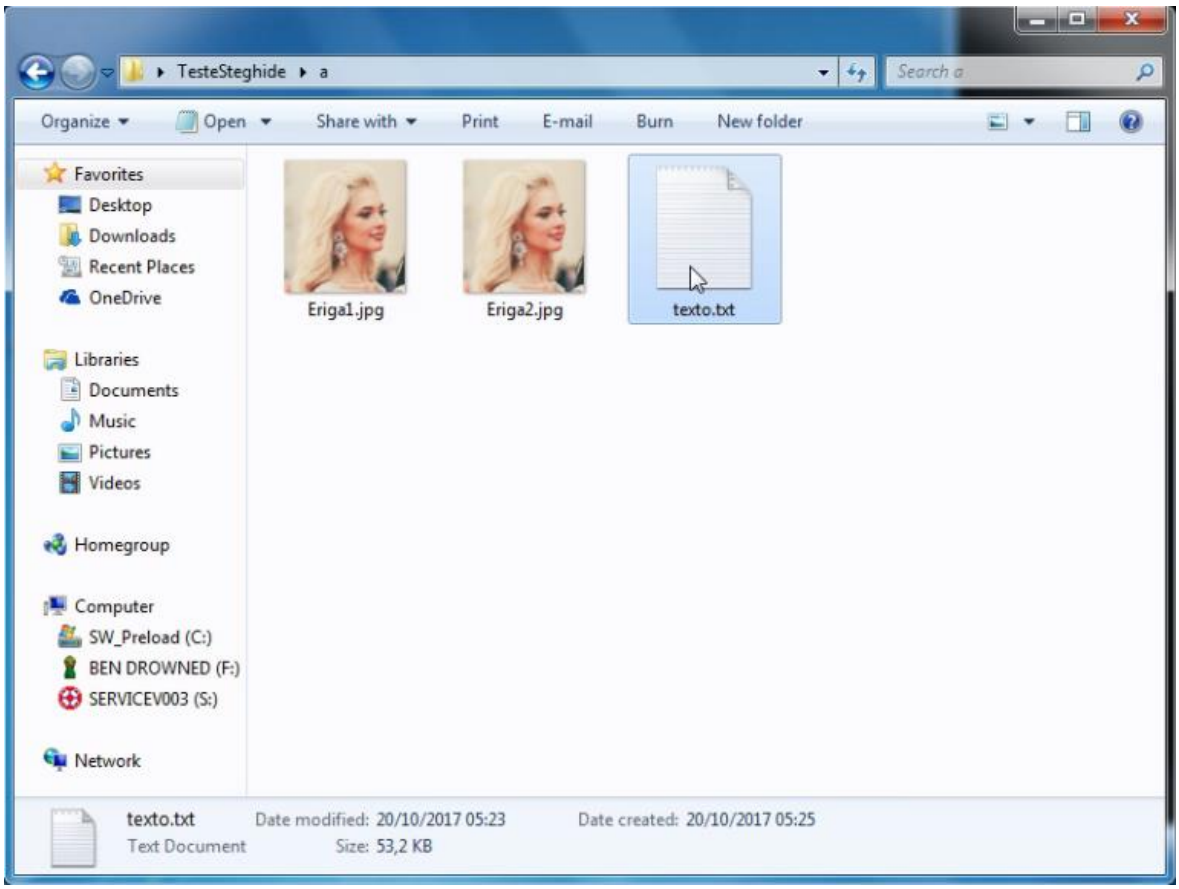
Figura 22 - Imagem selecionada para o processo de esteganografia



Fonte: Autor (2017)

Em seguida, criou-se duas cópias da imagem “Eriga.jpg” dentro do diretório “a” que se encontra dentro da pasta reservada para o teste, e as cópias dentro do diretório “a” foram renomeadas como “Eriga1.jpg” e “Eriga2.jpg”, aonde o primeiro arquivo será alvo da esteganografia e o segundo será preservado para fins de comparação. Em seguida copiou-se também um arquivo denominado “texto.txt” que possui 53,2KB de tamanho, contendo a informação a ser ocultada, fazendo com que o diretório “a” possuísse esses três arquivos, como exposto na figura 23.

Figura 23 - Diretório com os arquivos reunidos para o teste



Fonte: Autor (2017)

Após reunir esses arquivos nesse diretório, utilizou-se o prompt de comandos do Windows para poder executar o Software Steghide, inicializando-se primeiramente o diretório o qual os referidos arquivos se encontram, utilizando o seguinte comando:

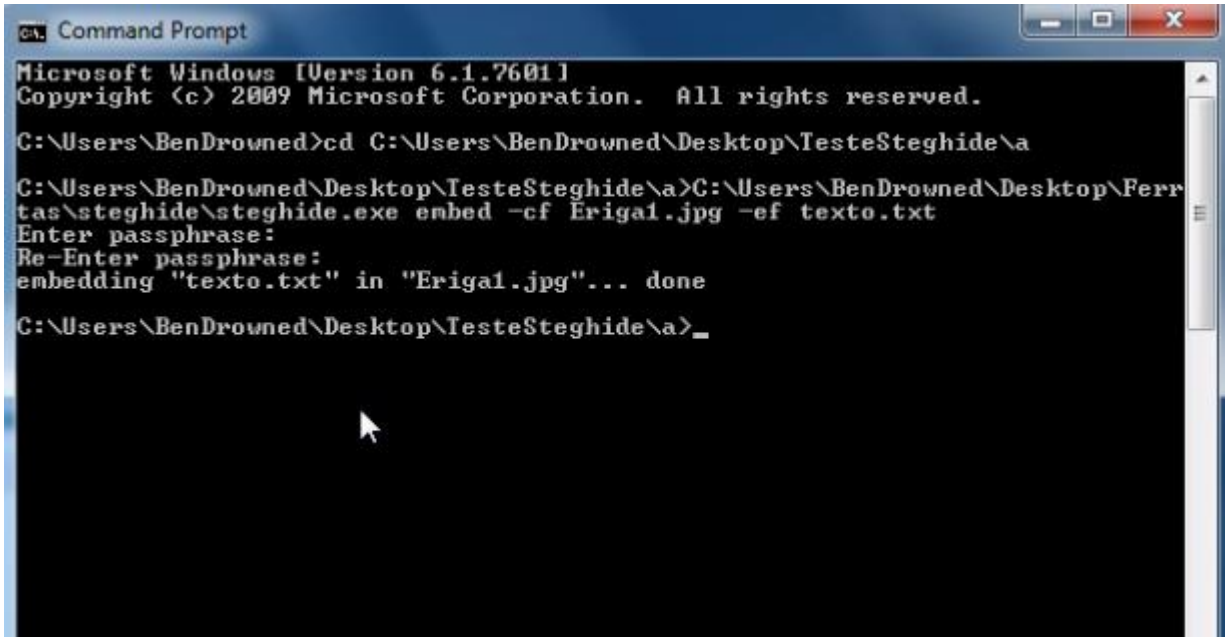
```
cd C:\Users\BenDrowned\Desktop\TesteSteghide\a
```

Após esse comando inicializou-se o Software referenciando o caminho o qual o mesmo se encontrava no computador no prompt de comandos, ativando o comando de esteganografia no arquivo “Eriga1.jpg” através da seguinte entrada, aonde o parâmetro “embed” inicializa a função de gerar a esteganografia, o parâmetro “-cf” define o arquivo que irá conter a mensagem oculta e o parâmetro “-ef” define o arquivo que contêm a mensagem a ser escondida:

```
C:\Users\BenDrowned\Desktop\Ferramentas\steghide\steghide.exe
embed –cf Eriga1.jpg –ef texto.txt
```

Após essa entrada, o Steghide requisita que o operador crie uma senha para criptografar o conteúdo no arquivo de destino, e após inserir a senha o aplicativo realiza a esteganografia, ocultando a informação nos pixels da imagem alvo de acordo com a senha introduzida, sendo o feedback das operações realizadas mostradas na figura 24, observando-se que o prompt de comando não exibe na tela a senha que o usuário insere.

Figura 24 - Entrada no prompt de comandos para utilizar o Steghide



Fonte: Autor (2017)

Para verificar o sucesso da operação, copiou-se o arquivo “Eriga1.jpg” de volta para o diretório anterior e direcionou-se o terminal para este diretório com o seguinte comando:

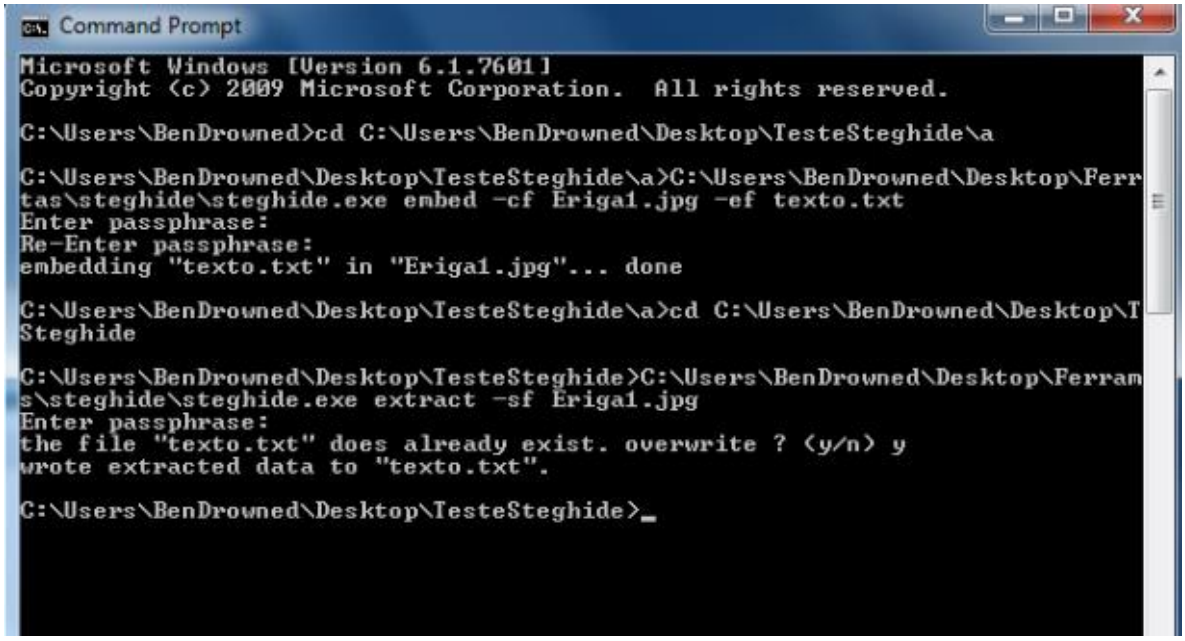
```
cd C:\Users\BenDrowned\Desktop\TesteSteghide
```

Em seguida invocou-se o Steghide novamente para extrair os dados que foram embutidos na imagem, utilizando a seguinte entrada, sendo o parâmetro “extract” o indicador para o software realizar a extração e o parâmetro “-sf” o indicador do arquivo que contêm a esteganografia:

```
C:\Users\BenDrowned\Desktop\Ferramentas\steghide\steghide.exe
extract –sf Eriga1.jpg
```

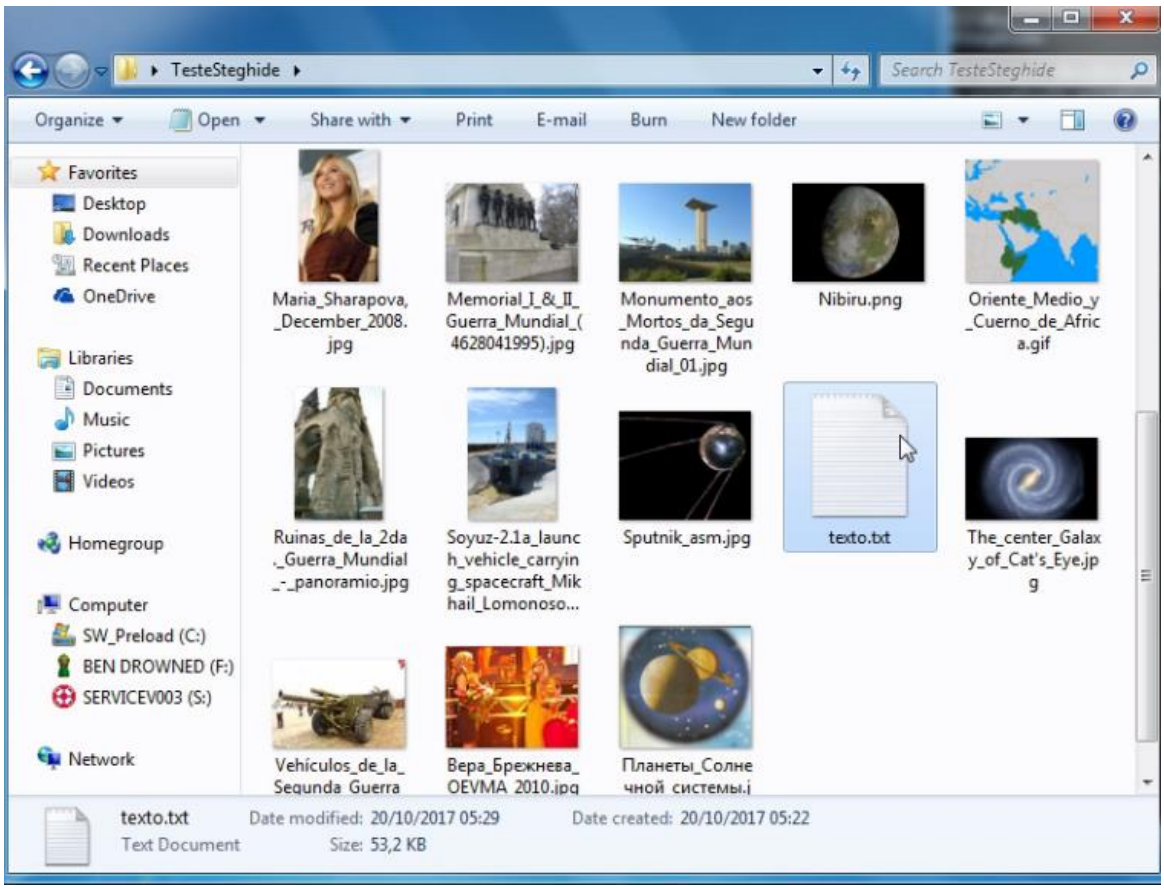
Após o usuário inserir a senha o programa realiza o processo de extração da informação, notando-se que o programa é capaz de preservar o nome original do arquivo que foi inserido na imagem, como exposto na figura 25 que expõe o resultado final da operação no prompt de comando e o momento em que o software pergunta se pode sobrescrever o arquivo original que ainda se encontrava neste diretório, e a figura 26 mostra o resultado final do diretório em questão, aonde o arquivo de texto recuperado possui o mesmo nome e tamanho do original (“texto.txt” com 53,2KB de tamanho).

Figura 25 - Resultado da operação no prompt de comandos após a extração de dados



Fonte: Autor (2017)

Figura 26 - Arquivo extraído com o mesmo nome de origem e tamanho

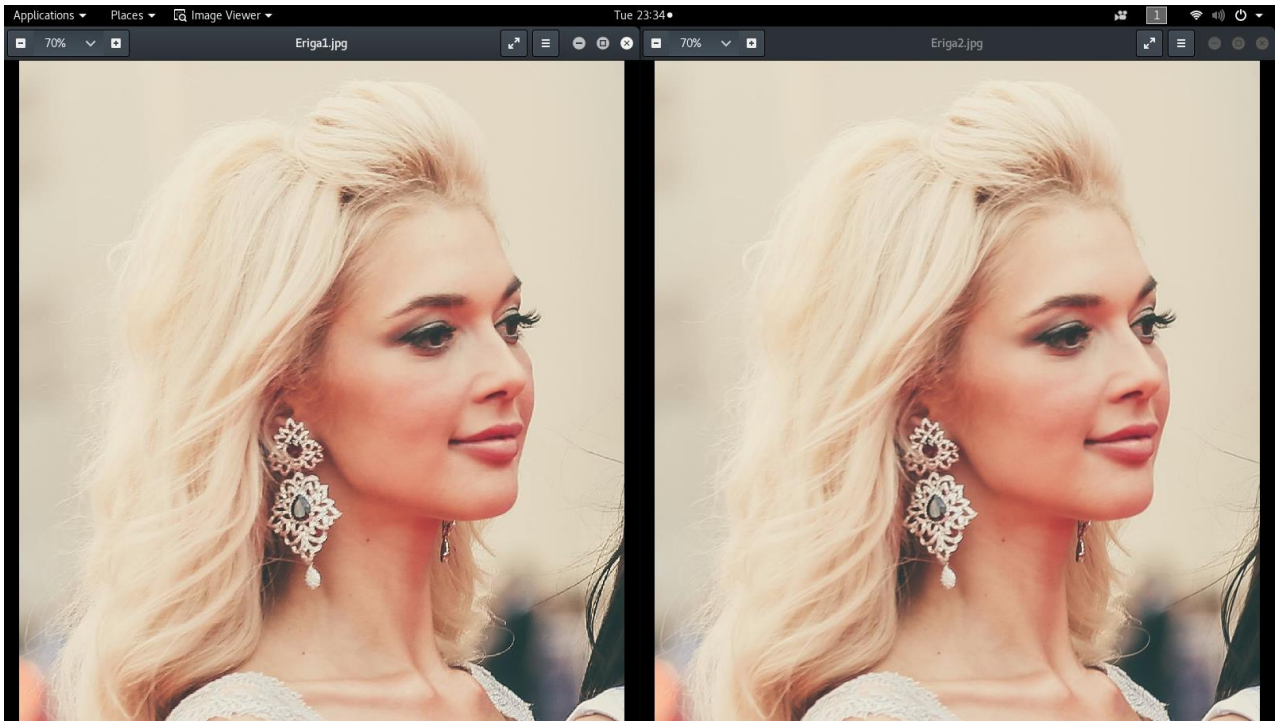


Fonte: Autor (2017)

4.1.2. Análise da Esteganografia

Identificar a presença de esteganografia em uma imagem é uma tarefa que pode ser bastante dificultosa, uma vez que a informação quase sempre é inserida nos pixels menos significativos da imagem, o que torna a diferenciação a olho nu muito improvável, quando não impossível em alguns casos. Na figura 27 temos a comparação entre o arquivo usado no teste com o Steghide já contendo a informação oculta com o seu estado original.

Figura 27 - Comparação visual da imagem com esteganografia com a imagem original



Fonte: Autor (2017)

Uma das formas mais eficientes de se analisar um arquivo de imagem que possa possuir esteganografia é por meio de um ataque visual, uma técnica que visa realçar os bits menos significativos dos pixels, pois segundo Almeida (2011, online) “partindo-se do princípio de que a informação foi embutida nos bits menos significativos das cores dos pixels, a análise visual consiste em deslocá-los para a posição mais significativa da cadeia binária”, aonde o mesmo apresenta um exemplo de script na linguagem python com esse objetivo, que está exposto na figura 28.

Figura 28 - Exemplo de implementação de ataque visual em python

```
# coding: utf-8

import Image

def ataque_visual(imagem, destino):
    # Abre a imagem, obtém seus atributos e carrega os pixels para a memória
    img = Image.open(imagem)
    largura, altura = img.size
    pix = img.load()

    for x in xrange(largura):
        for y in xrange(altura):
            # Desloca os bits menos significativos para a posição mais significativa
            pix[x, y] = tuple(int(bin(cor)[-1] + '1111111', 2) for cor in pix[x, y])
    img.save(destino, 'PNG', quality=100)
```

Fonte: Almeida (2011, online)

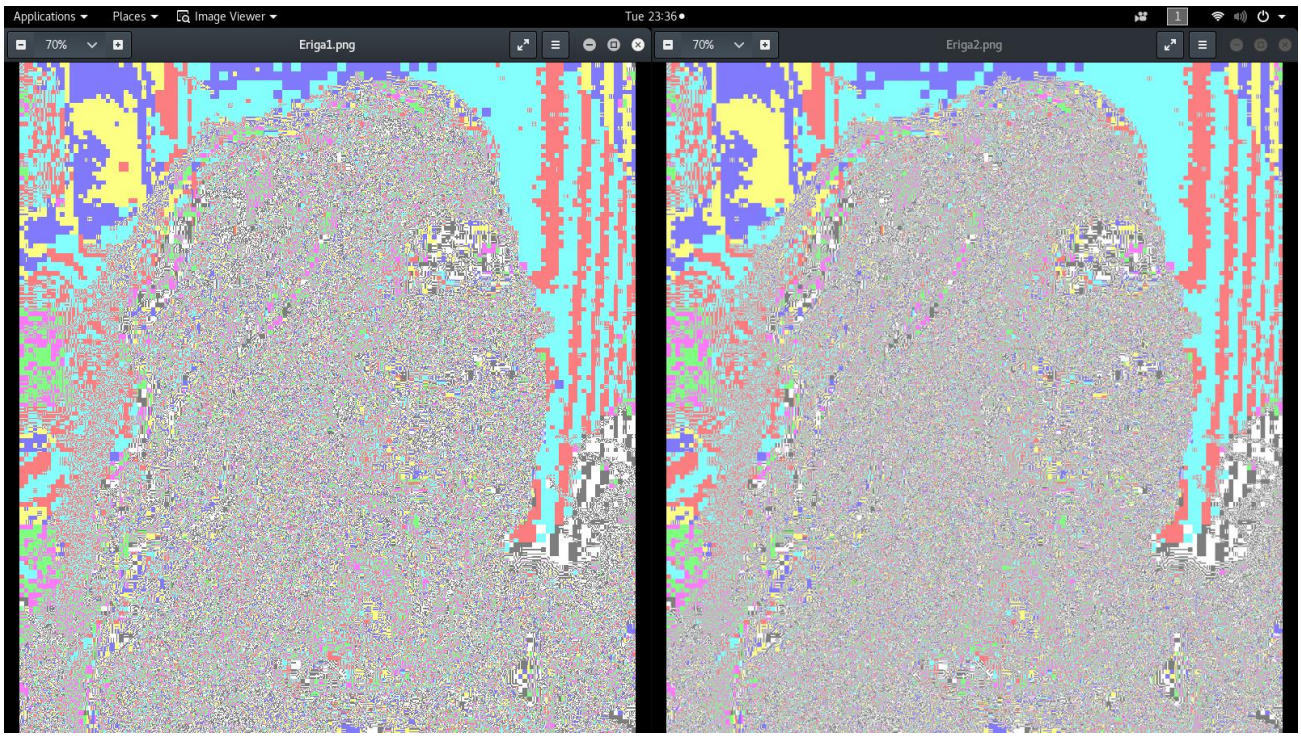
Com base nesse exemplo de script, foi criado um código semelhante para o propósito da análise deste trabalho, o qual foi modificado para pegar duas imagens de uma vez, com o intuito de analisar uma imagem com esteganografia e a sua versão antes deste processo, e então gerar dois arquivos de imagem com os bits dos pixels modificados para realizar a comparação, sendo o script final usado no teste exposto na figura 29.

Figura 29 - Implementação do ataque visual para o teste realizado

Fonte: Autor (2017)

Após executar o script, o mesmo realiza o processo de deslocamento dos bits e salva o resultado em um diretório de saída, nesse caso gerando o arquivo “Eriga1.png” para armazenar o resultado da imagem com esteganografia e criando o arquivo “Eriga2.png” para guardar o resultado da imagem no seu estado original, sendo a comparação dos resultados exposto na figura 30.

Figura 30 - Comparação entre os resultados dos dois ataques visuais



Fonte: Autor (2017)

Nota-se que existem alguns pixels em alguns lugares na imagem “Eriga1.png” que não correspondem com os pixels da imagem original “Eriga2.png”, mas no caso em questão os mesmos não são tão evidentes e dificilmente levantariam tanta suspeita se a imagem original não estivesse disponível para realizar uma comparação, e mesmo após a constatação da presença da esteganografia haveria a necessidade de coleta de mais informações para definir

4.2. FILE SLACK

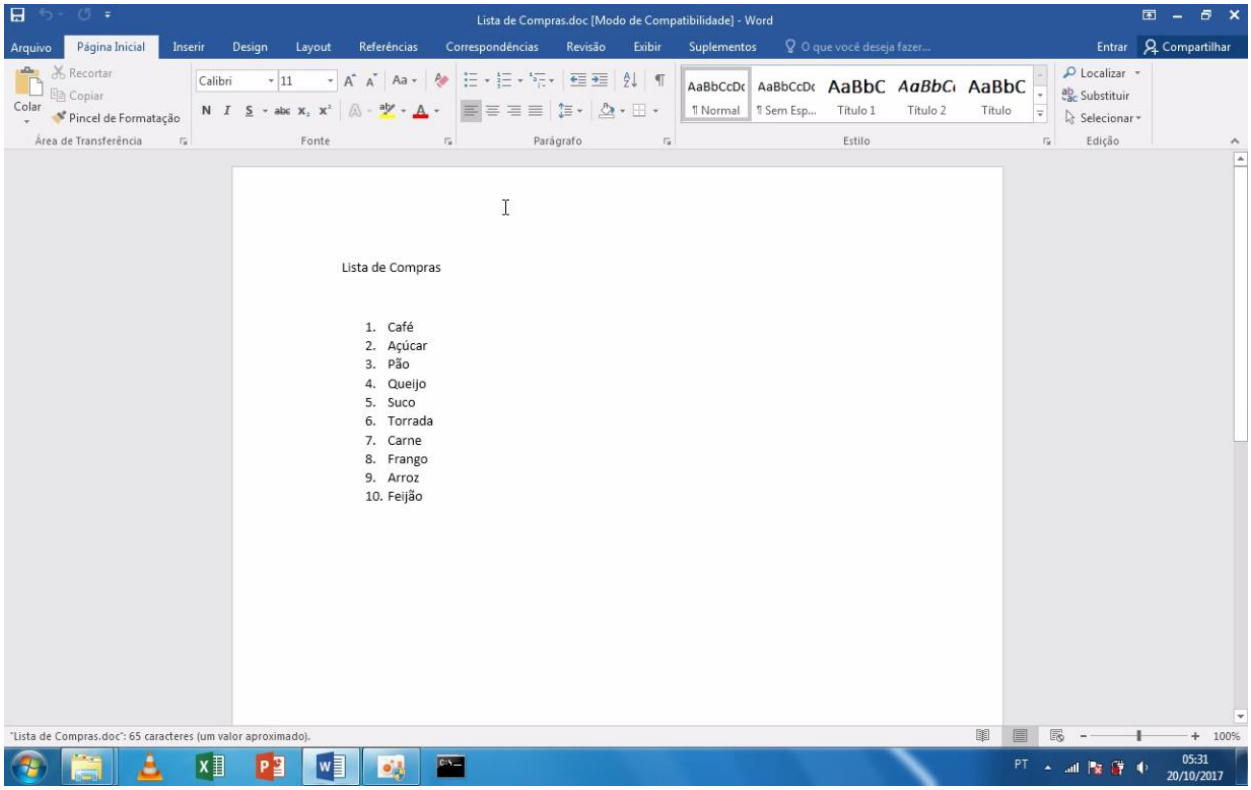
Em um disco rígido os dados armazenados são armazenados em setores no mesmo, que estão contidos nas trilhas do disco, porém o sistema operacional Windows não aloca os arquivos em setores de disco, e sim em clusters, que são conjuntos de setores, sendo o tamanho do cluster definido pelo sistema de arquivos (FAT ou NTFS). Como consequência, os arquivos alocados possuem dessa forma

possuem dois tamanhos, o tamanho lógico, que corresponde ao tamanho levando em consideração somente os setores alocados, e o tamanho físico, que corresponde ao tamanho real dos clusters no disco rígido, sendo a diferença desses dois tamanhos denominada “File Slack”, que é uma região que apesar de estar alocada na memória para o arquivo não contém de fato informações que fazem parte do mesmo, o que torna possível a ocultação de informação nesse espaço, visto que a mesma não se manifestará no arquivo referido.

4.1.1. Aplicação de ocultação de informações no File Slack

Para o presente teste criou-se um arquivo DOC chamado “Lista de Compras.doc” contendo um determinado texto, como exposto na figura 31:

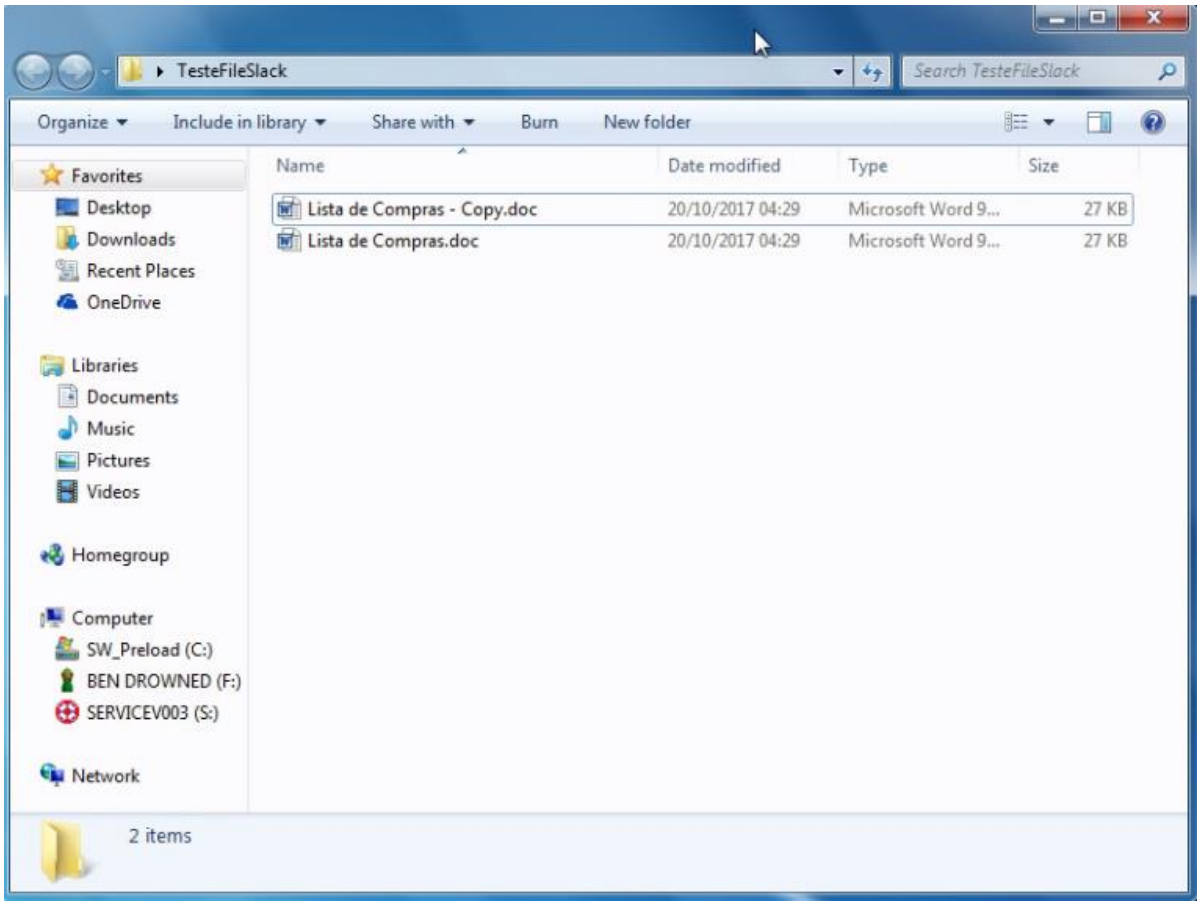
Figura 31 - Documento criado para o teste de File Slack



Fonte: Autor (2017)

Realizou-se então uma cópia do arquivo no mesmo diretório para preservar o estado do arquivo original, como exposto na figura 32.

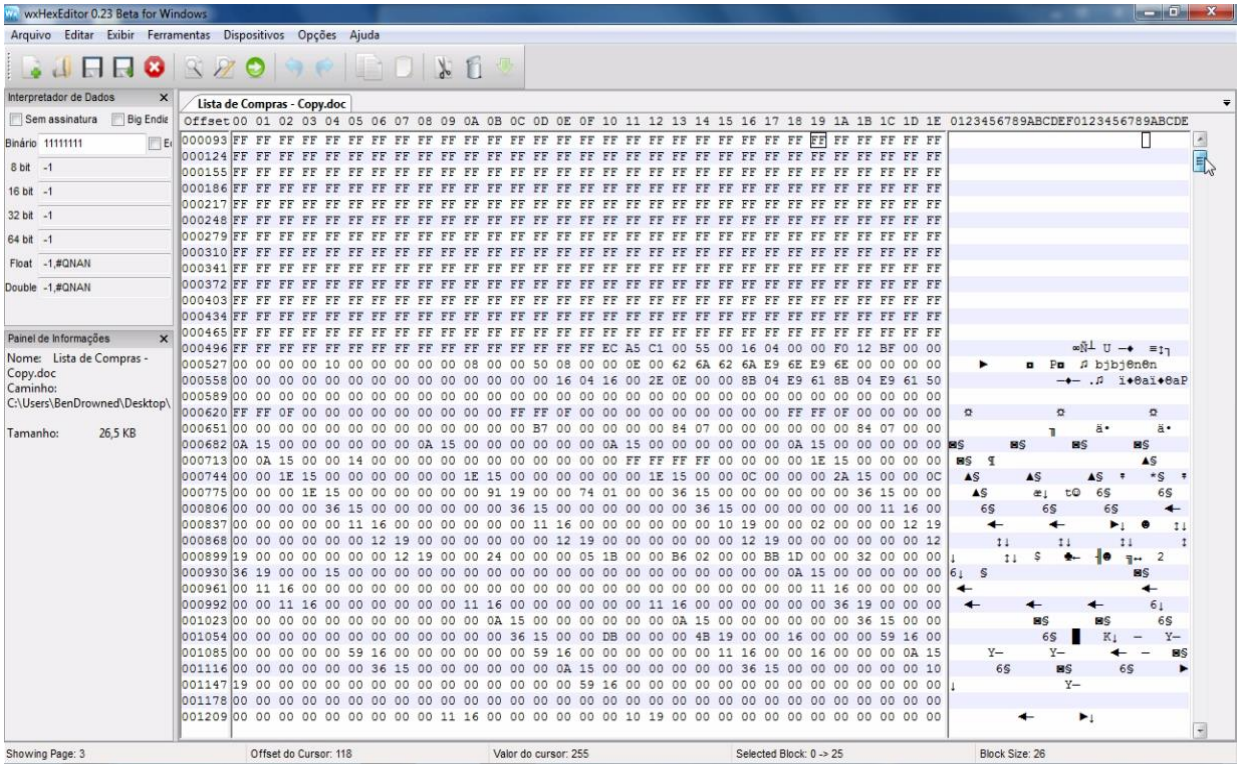
Figura 32 - O arquivo de teste e sua cópia de segurança



Fonte: Autor (2017)

Após a cópia do arquivo, o arquivo “Lista de Compras - Copy.doc” foi aberto por meio do software wxHexEditor, que é um editor hexadecimal de arquivos, como demonstrado na figura 33.

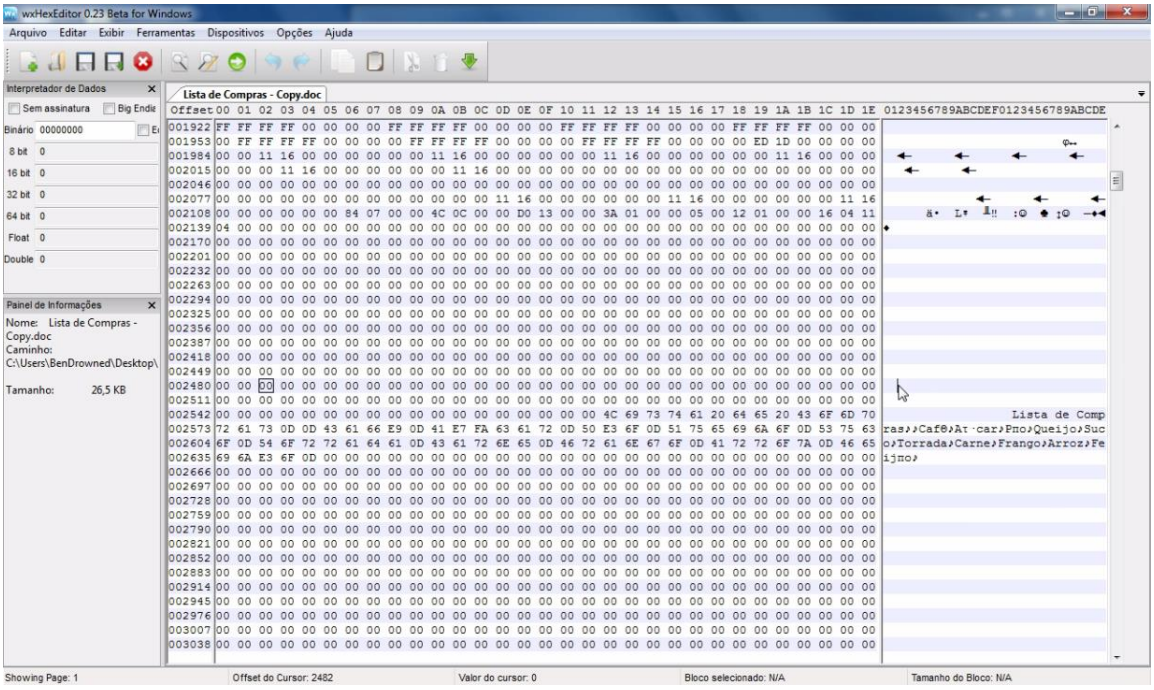
Figura 33 - O arquivo escolhido para o teste aberto no editor hexadecimal



Fonte: Autor (2017)

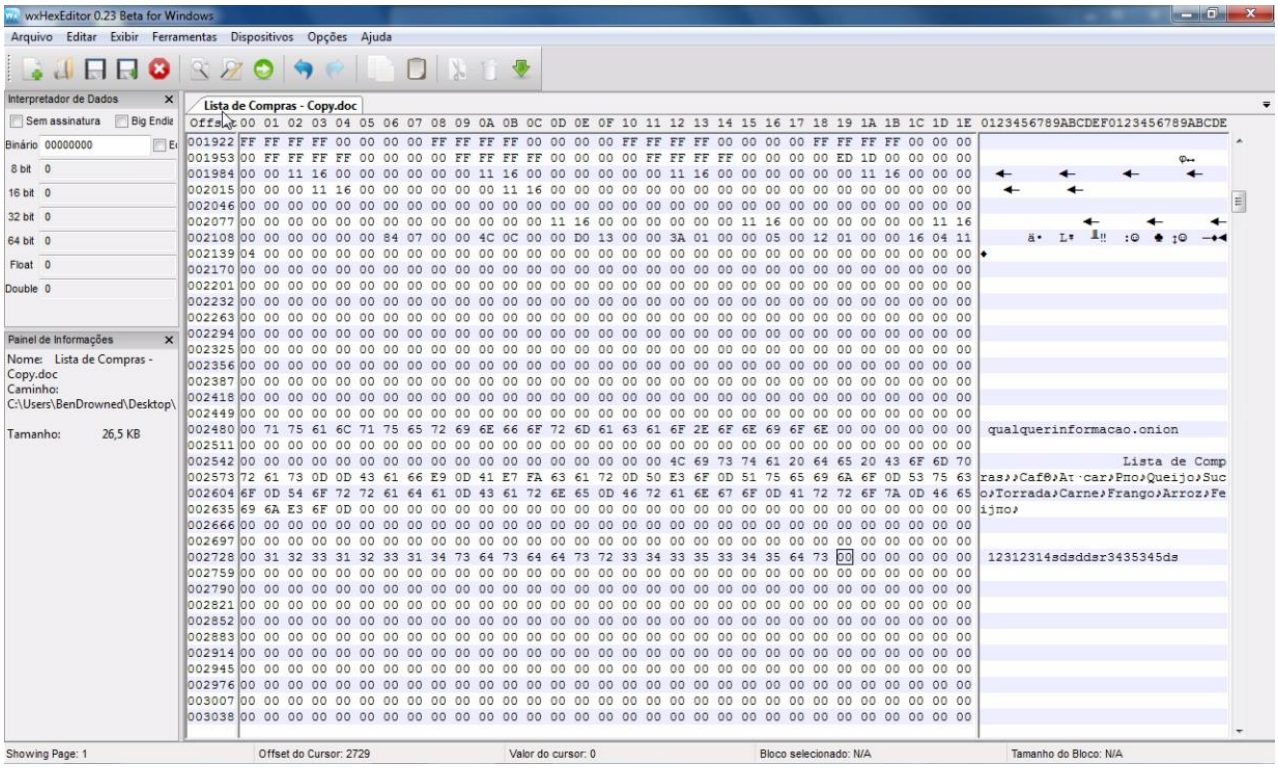
Utilizando este programa, navegou-se pelo conteúdo do arquivo até localizar o texto que estava presente no arquivo, como exposto na figura 34, e foram escolhidas duas regiões vazias (em relação ao arquivo) e foram introduzidas informações em texto nessas regiões, como mostra a figura 35.

Figura 34 - Localização do conteúdo do arquivo em hexadecimal



Fonte: Autor (2017)

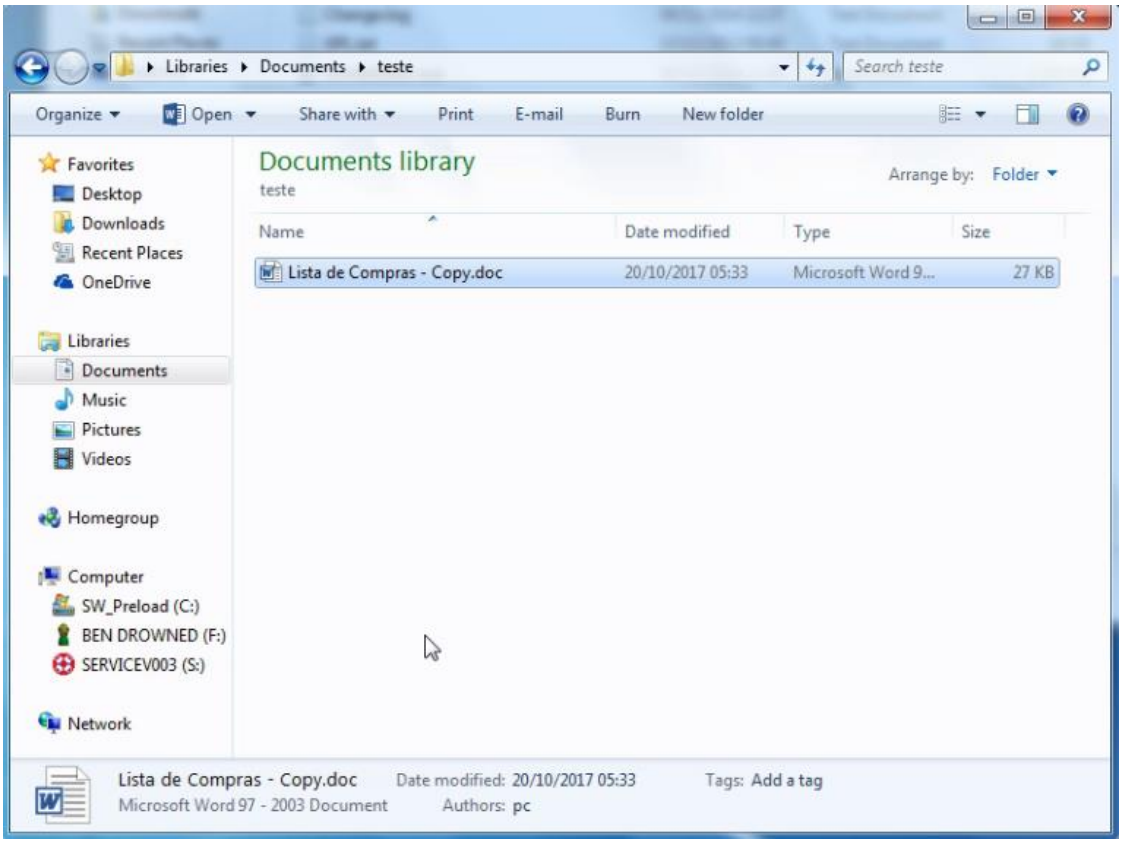
Figura 35 - Informação inserida em espaços "vazios" no arquivo



Fonte: Autor (2017)

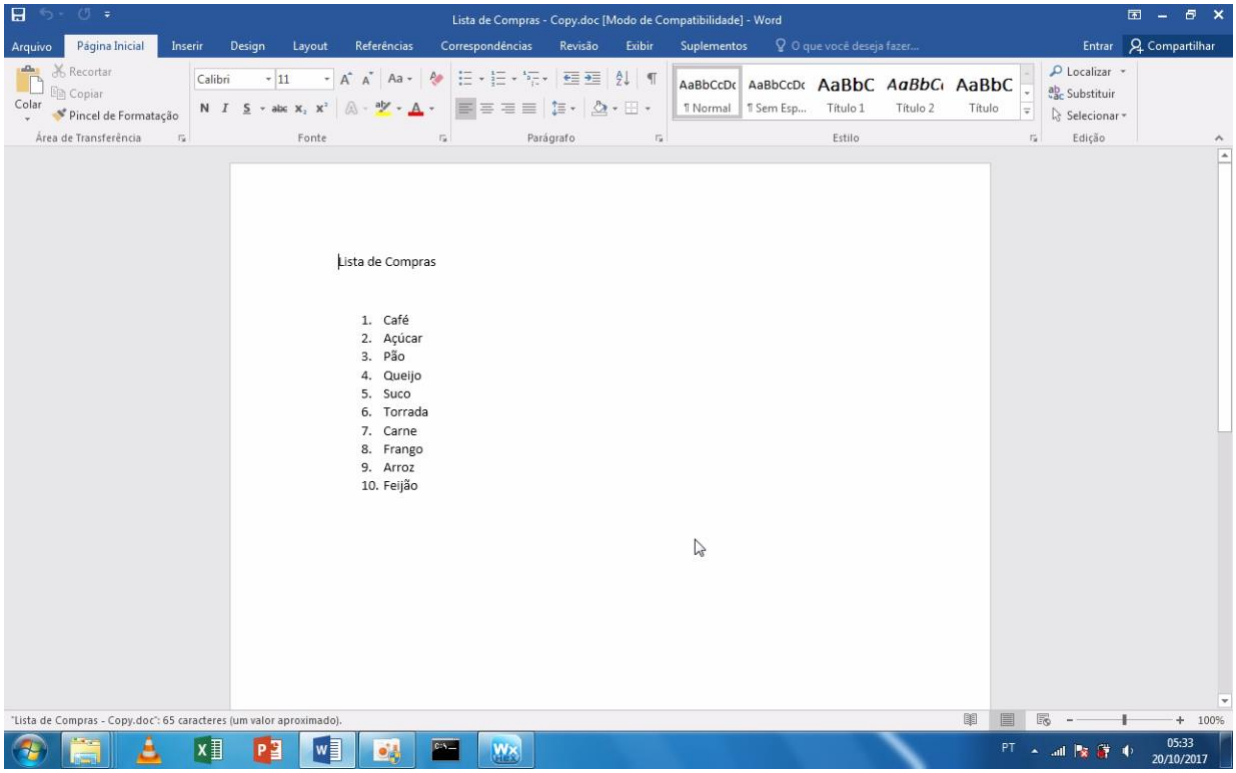
Após salvar as modificações no arquivo em questão, foi feita uma cópia do arquivo alterado para um outro diretório como demonstrado na figura 36, então o arquivo foi aberto normalmente para averiguar se o mesmo demonstrou alguma alteração, e como consta na figura 37 o mesmo não sofreu nenhuma alteração, permanecendo visualmente igual ao original.

Figura 36 - Arquivo com as informações escondidas copiado para outro diretório



Fonte: Autor (2017)

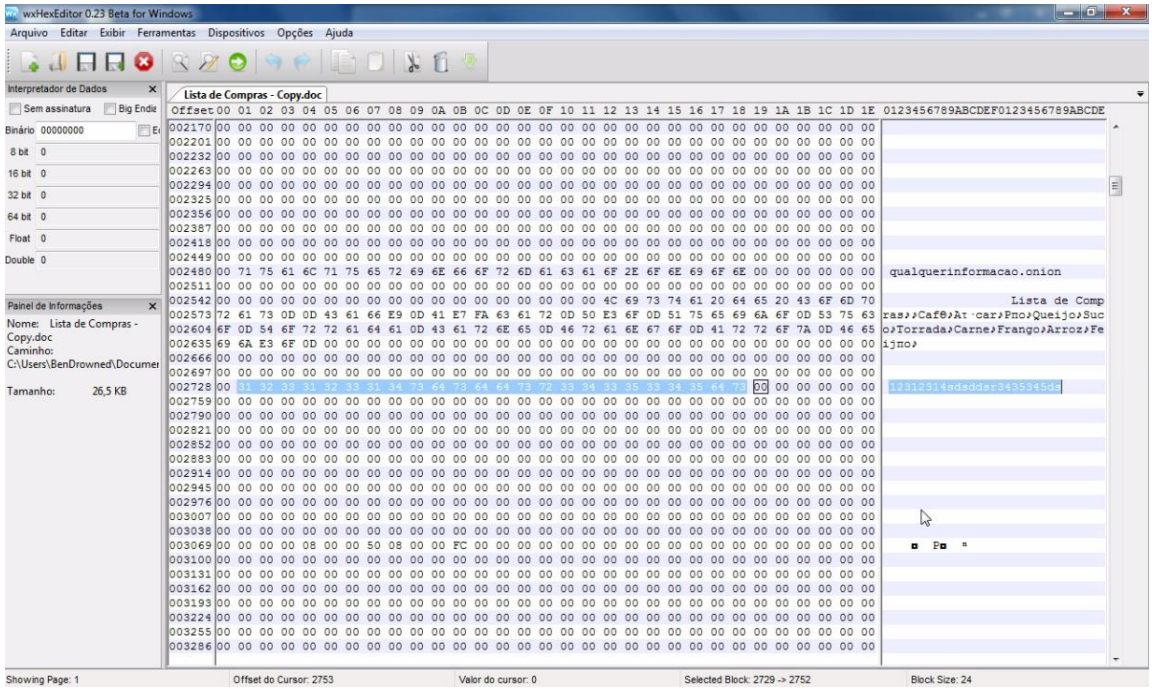
Figura 37 - Arquivo reaberto para averiguar possíveis alterações de conteúdo



Fonte: Autor (2017)

Porém ao abrir esse arquivo novamente pelo editor de hexadecimal é constatado que as informações introduzidas permanecem no mesmo local sem perder sua integridade, como exposto na figura 38.

Figura 38 - Informações ocultas ainda presentes no arquivo

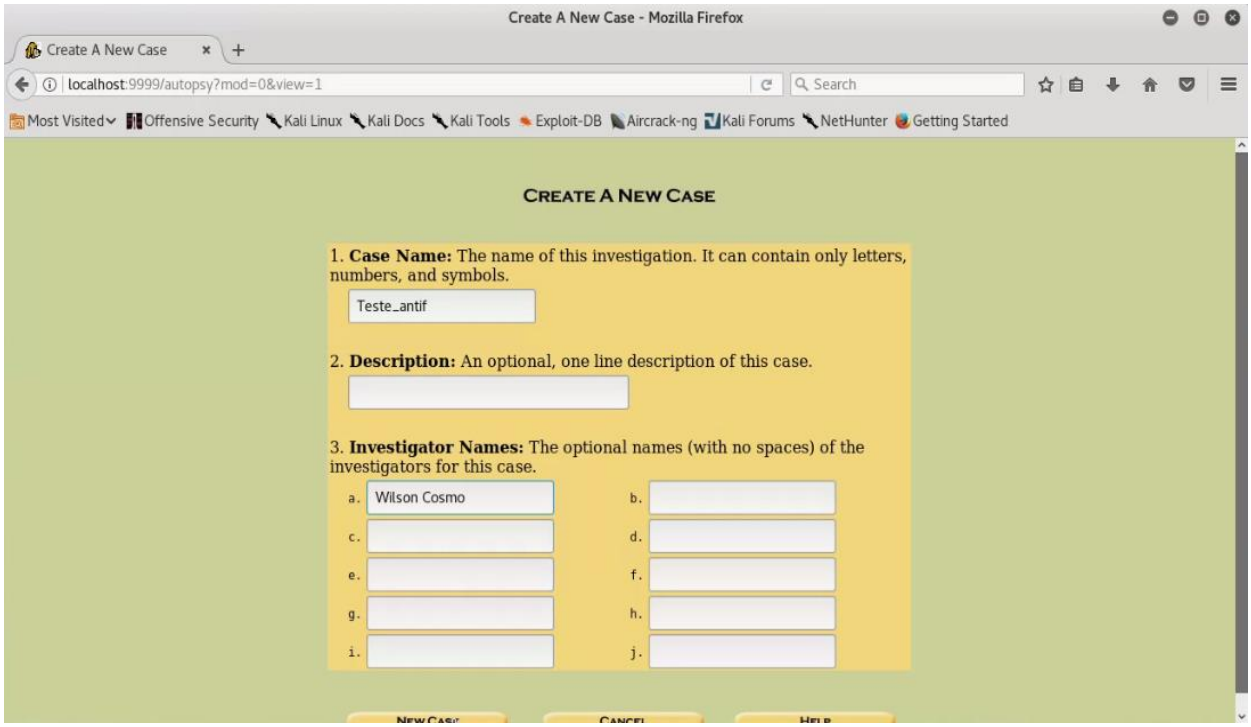


Fonte: Autor (2017)

4.2.2. Análise do File Slack

Para realizar a análise dos presentes arquivos utilizou-se o software Autopsy presente no SO Kali Linux. Para iniciar a análise, iniciou-se o programa e criou-se um novo caso, como exposto na figura 39, sendo essa a forma a qual o mesmo utiliza para organizar as análises.

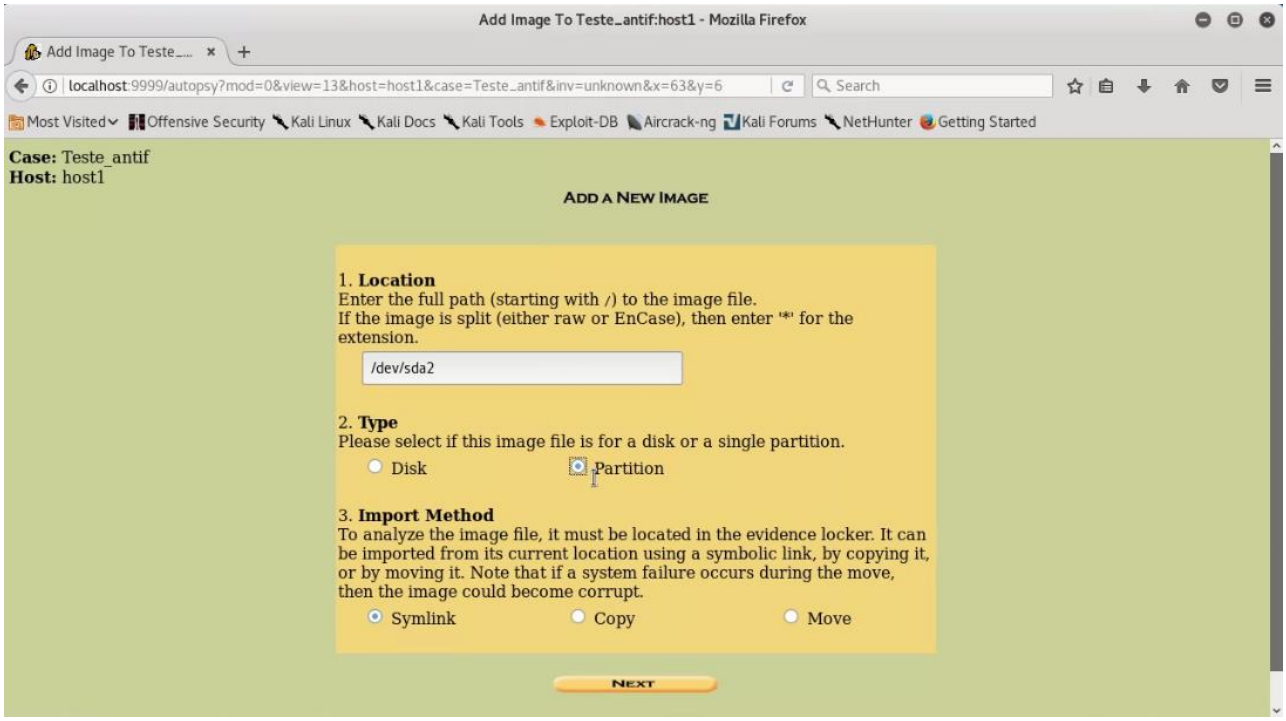
Figura 39 - Tela de criação de um novo caso no Autopsy



Fonte: Autor (2017)

Após o operador introduzir as informações necessárias para categorizar e organizar o caso, o software irá requisitar o caminho para o objeto da análise que neste caso é uma partição que contém o sistema que ocorreram os testes, então repassou-se para o software a localização da partição, a indicação de que o objeto representa uma partição e o método de importação dos dados, que pode ser realizado através de links simbólicos, através de uma cópia dos dados ou movendo o conteúdo para um diretório de análise, como mostra a figura 40.

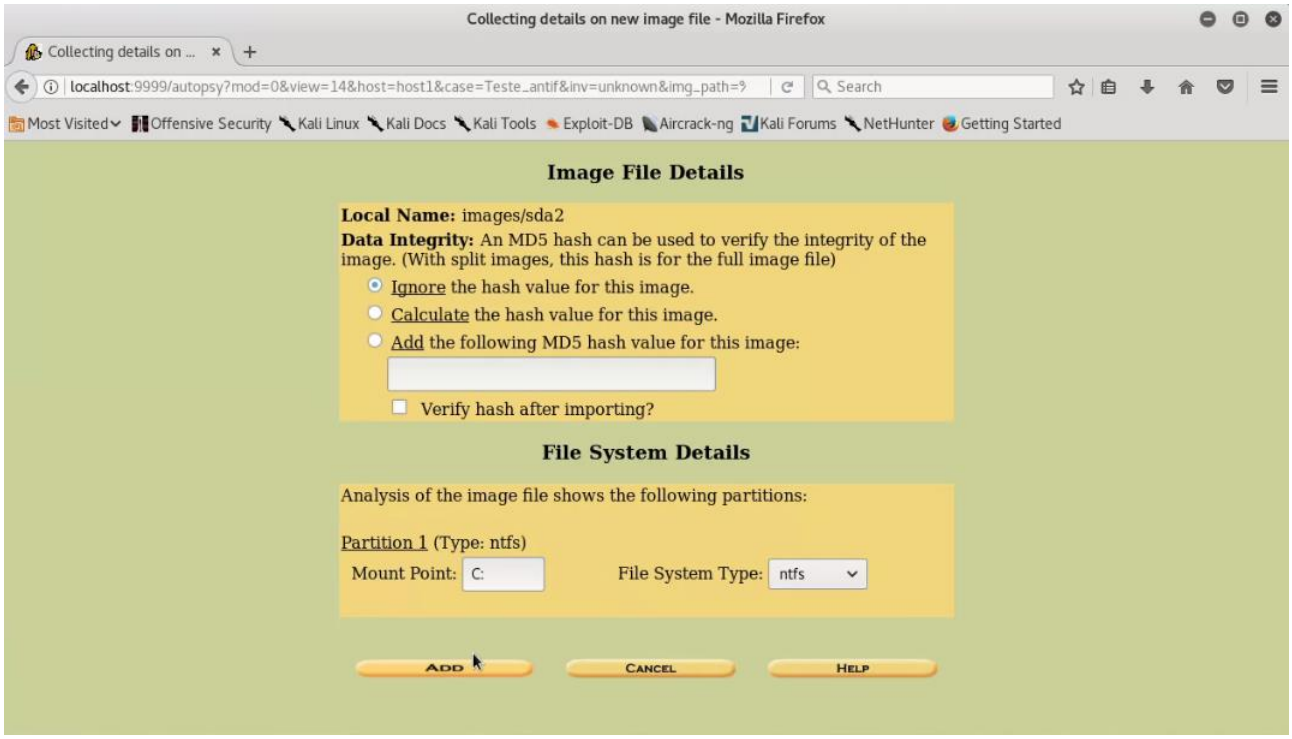
Figura 40 - Tela de definição do objeto de análise



Fonte: Autor (2017)

Após essa etapa o usuário pode optar por adicionar um hash MD5 para verificar a integridade da imagem analisada, caso fosse necessário, e também indica o ponto de montagem da partição analisada e o seu sistema de arquivos, como mostra a figura 41.

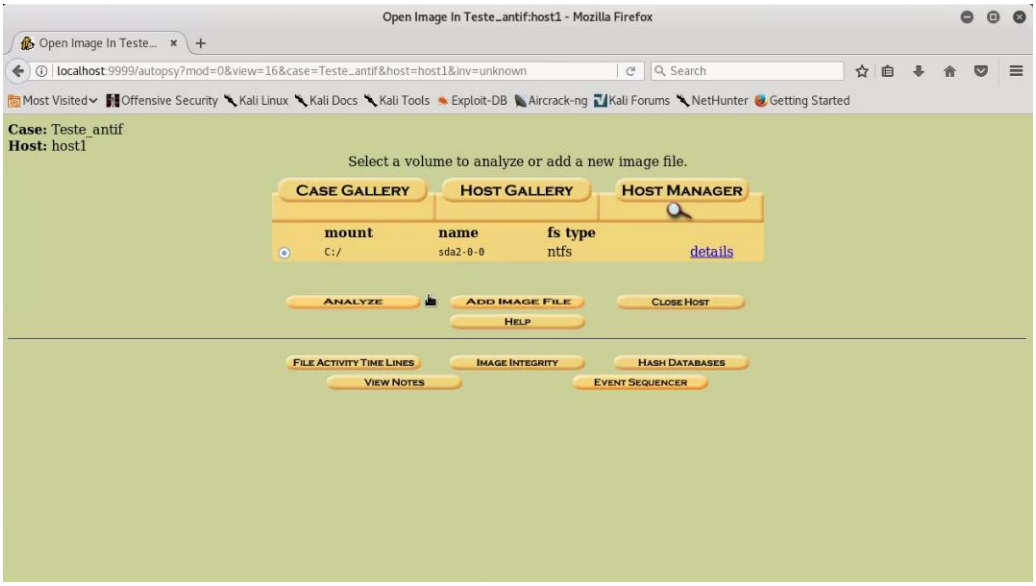
Figura 41 - Opções adicionais sobre o objeto de análise



Fonte: Autor (2017)

Após esta operação o usuário será direcionado para uma tela aonde ele poderá adicionar mais imagens, partições ou unidades para o seu caso, como exposto na figura 42, porém o mesmo já possui a opção de realizar análise possuindo pelo menos um objeto de estudo registrado no caso.

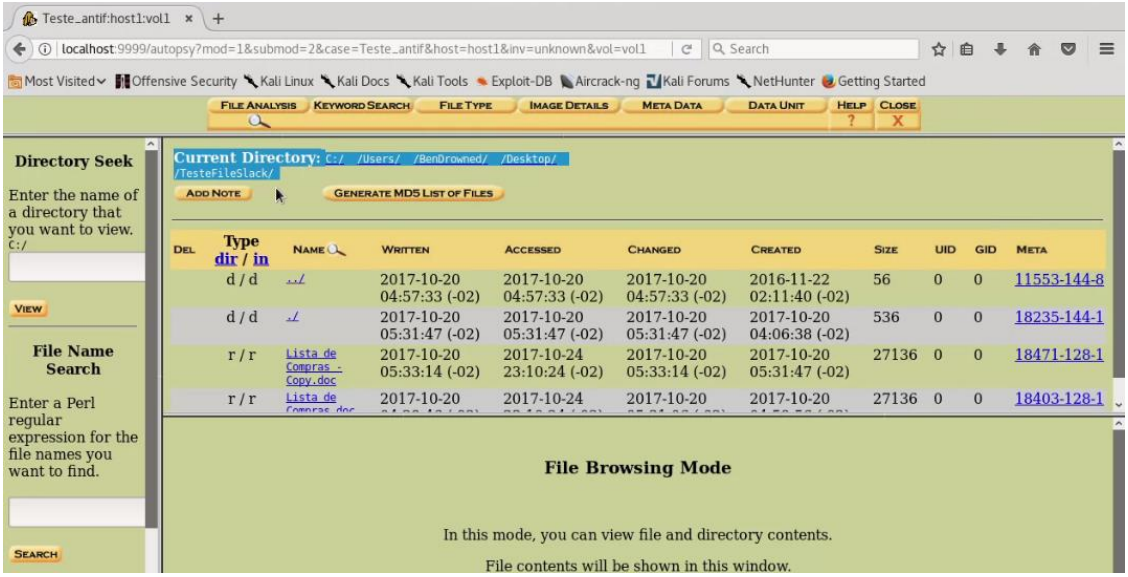
Figura 42 - Lista de memórias registradas no caso disponíveis para análise



Fonte: Autor (2017)

Após inicializar a opção de análise na partição definida, navegou-se até o diretório o qual foi realizado os testes de ocultação de dados com File Slack, como exposto na figura 43, observando-se que tanto o arquivo com o conteúdo oculto quanto o arquivo original possuem o mesmo tamanho na memória.

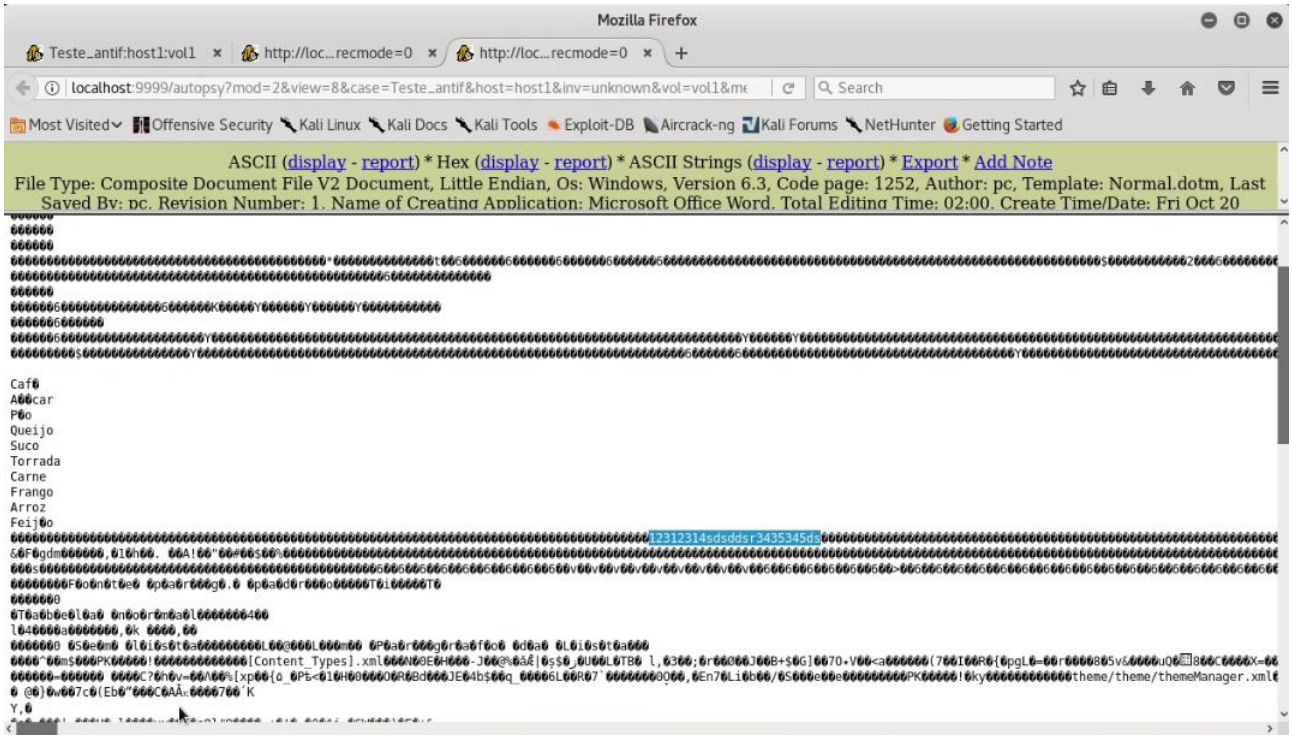
Figura 43 - Autopsy posicionado no diretório do teste de file slack



Fonte: Autor (2017)

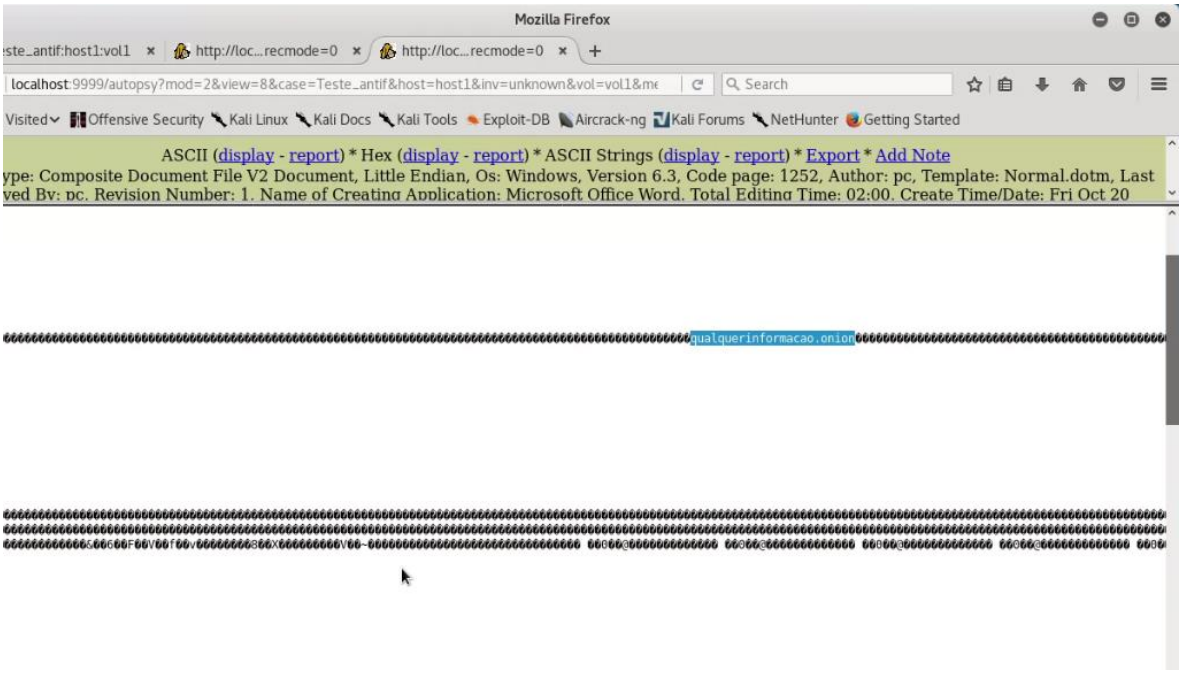
Após navegar até este diretório, os dois arquivos presentes tiveram seu conteúdo analisado pelo software, sendo que cada um foi aberto em uma aba diferente do navegador. O arquivo “Lista de Compras - Copy.doc” teve o conteúdo oculto identificado sem muitas dificuldades no modo de visualização de conteúdo ASCII, como exposto nas figuras 44 e 45, conteúdo que também pôde ser localizado pela visualização de conteúdo hexadecimal, como representado na figura 46.

Figura 44 - Primeira parte da informação escondida por File Slack localizada no arquivo



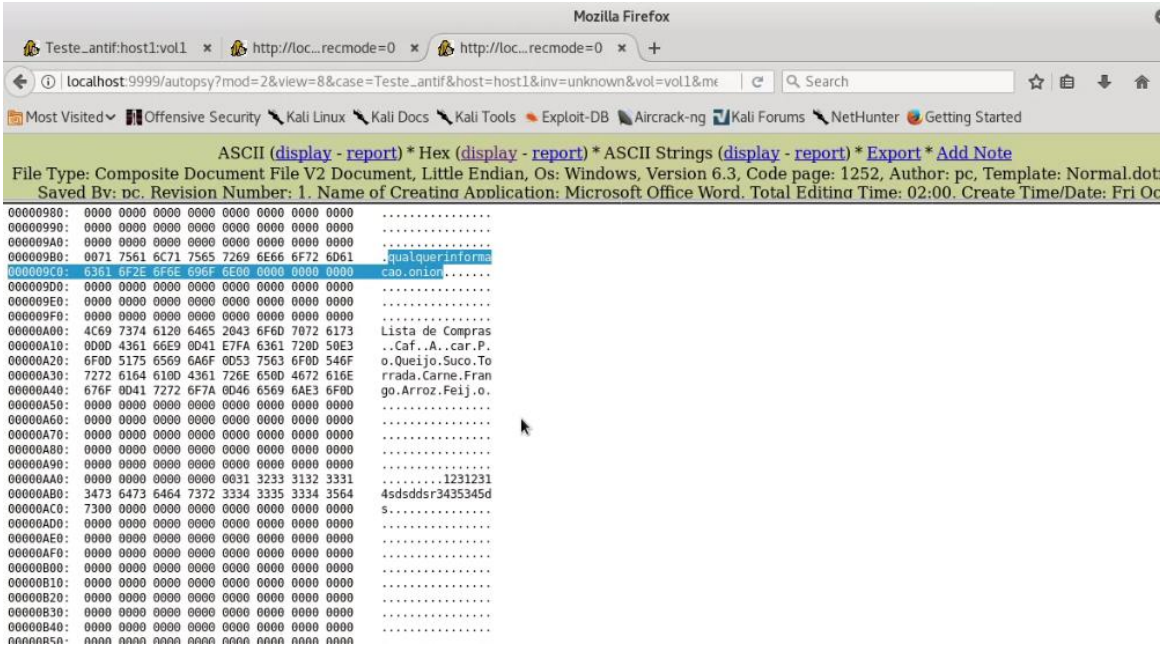
Fonte: Autor (2017)

Figura 45 - Segunda parte da informação escondida por File Slack localizada no arquivo



Fonte: Autor (2017)

Figura 46 - Informações escondidas por File Slack identificadas pelo visualizador hexadecimal



Fonte: Autor (2017)

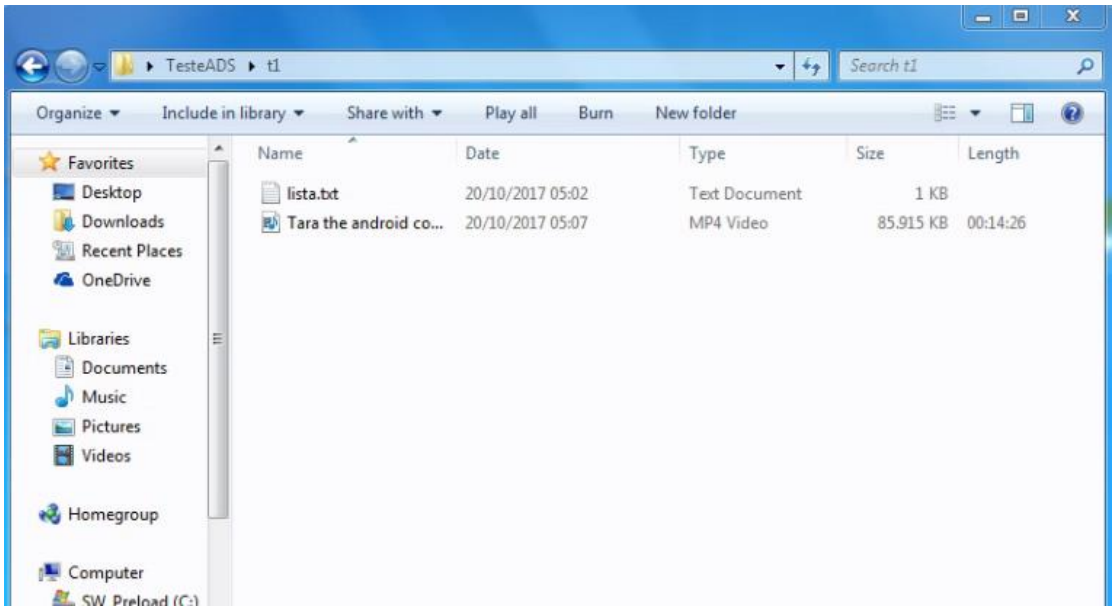
4.3. ALTERNATE DATA STREAMS

De acordo com Arntz (2016, online, tradução nossa) “Os Alternate Data Streams (ADS) são um atributo de arquivo encontrado apenas no sistema de arquivos NTFS”, sendo uma funcionalidade criada para implementar um sistema de bifurcação originalmente projetado para armazenar ícones e arquivos adicionais de um arquivo em um espaço associado ao próprio arquivo, tornando-se uma funcionalidade muito interessante no contexto de ocultamento de arquivos.

4.3.1. Aplicação de ocultação de arquivos no ADS

Para o experimento de ocultação de arquivos em ADS foram colocados em um diretório dois arquivos, um no formato TXT denominado “lista.txt” e um arquivo de vídeo denominado “Tara the android compilation.mp4”, como exposto na figura 47.

Figura 47 - Diretório contendo os arquivos para a técnica ADS



Fonte: Autor (2017)

Em seguida, direcionou-se o prompt de comandos para o referido diretório por meio do comando “cd”, com a seguinte entrada:

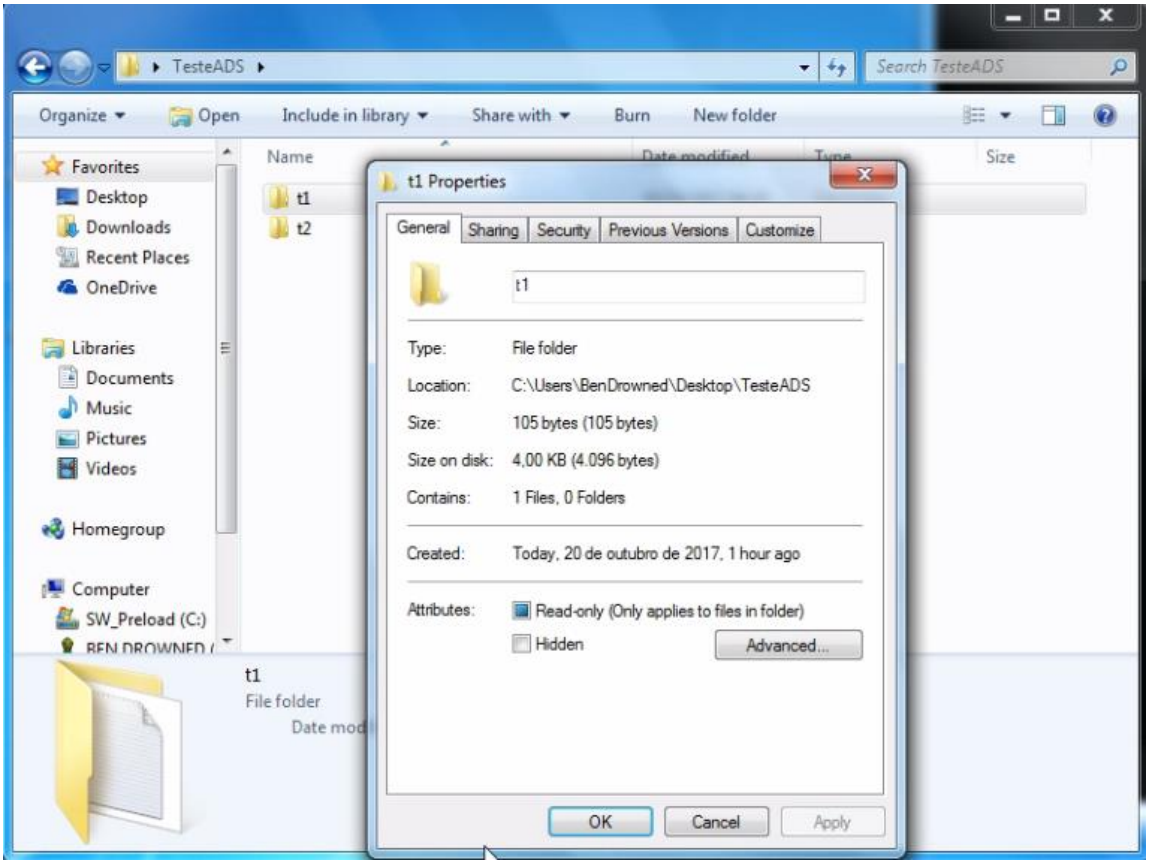
```
cd C:\Users\BenDrowned\Desktop\TesteADS\t1
```

Para alocar o arquivo de vídeo em uma ADS atrelada ao arquivo de texto ultilizou-se a seguinte entrada no prompt de comandos, aonde o arquivo original “Tara the android compilation.mp4” foi copiado para um novo arquivo denominado “a.mp4” criado em uma ADS atrelado ao arquivo “lista.txt”:

```
type “Tara the android compilation.mp4” > lista.txt:a.mp4
```

Após o prompt de comandos processar a entrada anterior, o arquivo “Tara the android compilation.mp4” foi movido para a lixeira e o peso e a quantidades de arquivos do diretório foi averiguado pela tela “Propriedades” da referida pasta, sendo que a mesma retorna para o operador a informação de que só existe um arquivo na pasta (que no caso é o arquivo “lista.txt”) e também informa que o peso total da pasta é o peso deste arquivo de texto, como exposto na figura 48.

Figura 48 - Informações da pasta que contêm o arquivo que possui ADS



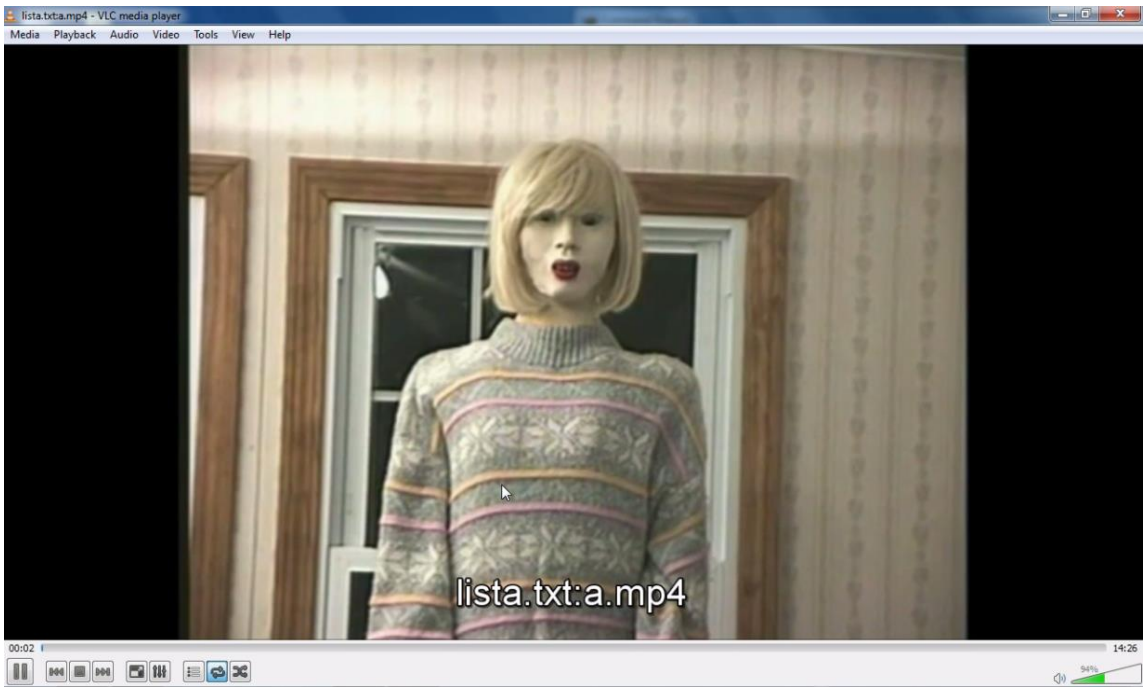
Fonte: Autor (2017)

Para abrir o arquivo presente na ADS é necessário inicializar o programa que é compatível com o arquivo oculto no prompt de comandos, no presente teste foi utilizado o programa VLC Media Player, e em seguida declara-se o arquivo a ser aberto, como na seguinte entrada:

```
"C:\Program Files\VideoLAN\VLC\vlc.exe" lista.txt:a.mp4
```

Após este comando, o arquivo de vídeo “a.mp4” é inicializado normalmente pelo VLC Media Player, como exposto na figura 49.

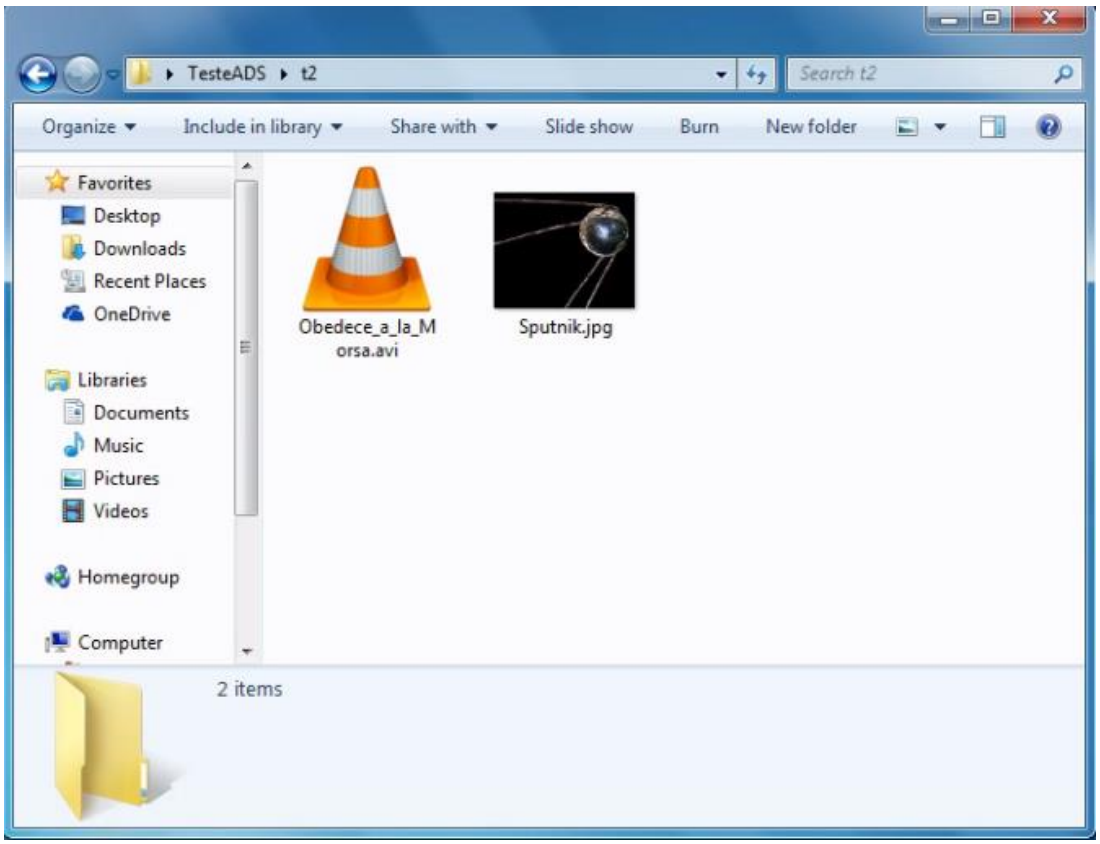
Figura 49 - Arquivo oculto por ADS inicializado



Fonte: Autor (2017)

Uma observação importante sobre a técnica apresentada é que o arquivo que está oculto não está criptografado, logo poderia ser facilmente acessado caso o perito identificasse a presença do mesmo no outro arquivo, bem como o seu nome. Para contornar esta questão é possível criptografar o arquivo antes de ocultar o mesmo via ADS. Para exemplificar este caso realizou-se um segundo teste com ADS, reunindo-se os arquivos “Sputnik.jpg” e “Obedece_a_la_Morsa.avi” em um novo diretório, como exposto na figura 50.

Figura 50 - Arquivos reunidos no diretório para o segundo teste com ADS



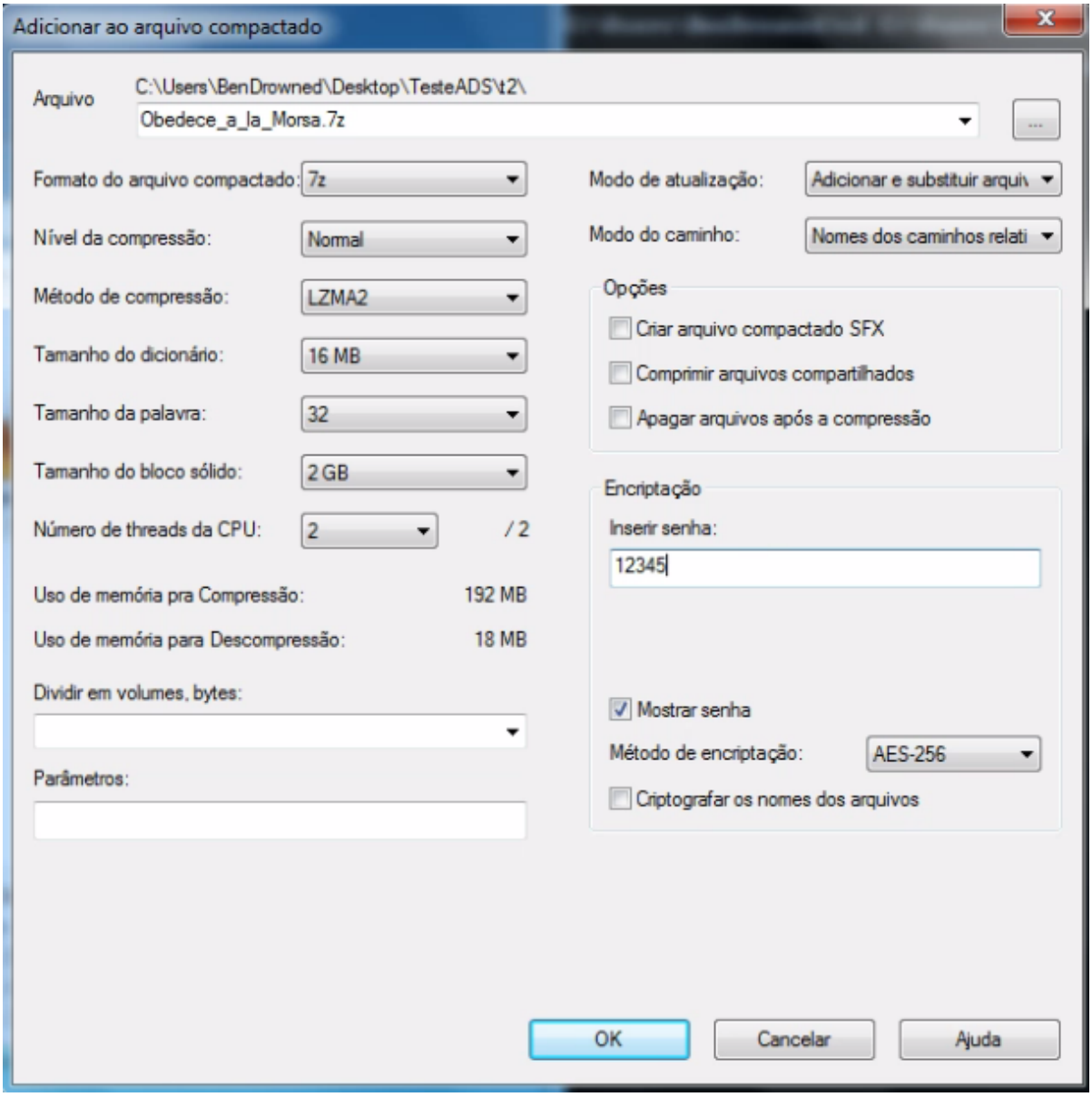
Fonte: Autor (2017)

Como no teste anterior, direcionou-se o prompt de comandos para este diretório utilizando-se o comando “cd”, usando a seguinte entrada:

```
cd C:\Users\BenDrowned\Desktop\TesteADS\t2
```

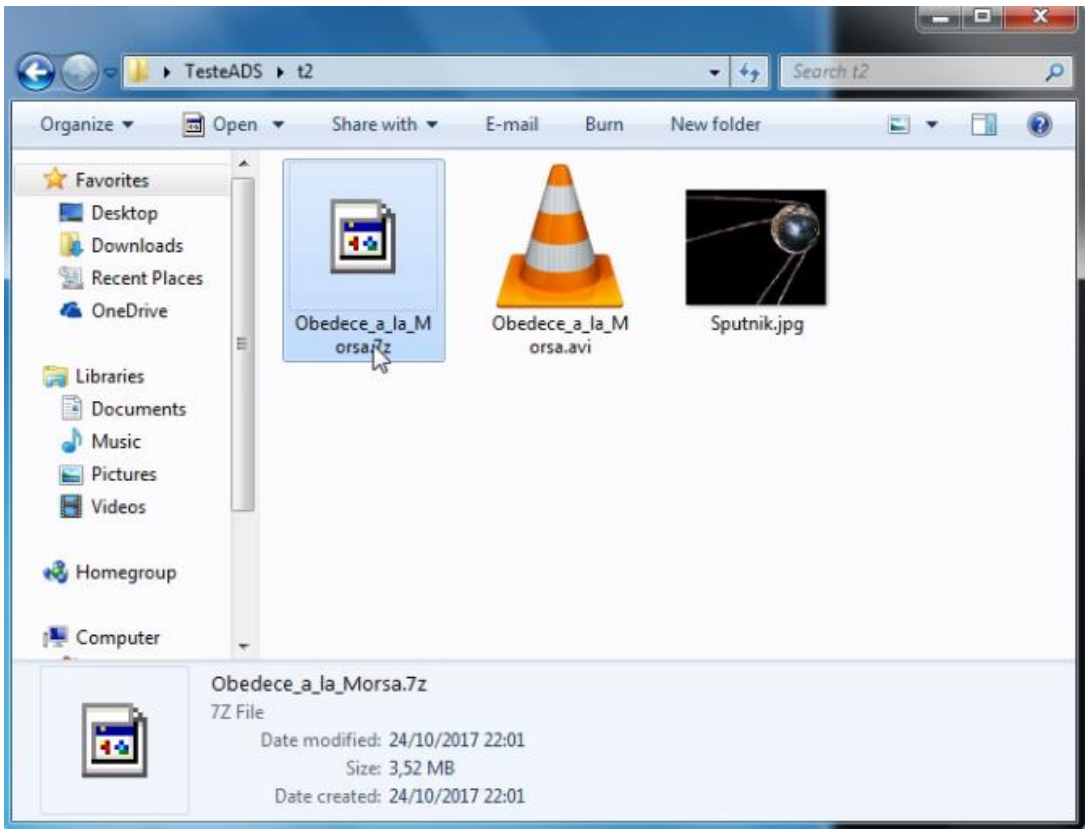
Antes de prosseguir com o uso do ADS utilizou-se o Software 7zip para criar um contêiner compactado e criptografado contendo o arquivo “Obedece_a_la_Morsa.avi” como demonstrado na figura 51, gerando então um novo arquivo denominado “Obedece_a_la_Morsa.7z”, que foi criado no mesmo diretório que os outros arquivos, como mostra a figura 52.

Figura 51 - Opções de compactação e encriptação fornecidas pelo 7zip



Fonte: Autor (2017)

Figura 52 - Arquivo compactado e criptografado gerado



Fonte: Autor (2017)

Após gerar o arquivo “Obedece_a_la_Morsa.7z” prosseguiu-se para o processo de ocultação por ADS, usando a seguinte entrada:

```
type Obedece_a_la_Morsa.7z > Sputnik.jpg:b.7z
```

Em seguida, após o prompt de comando processar a cópia do “Obedece_a_la_Morsa.7z” para o arquivo “b.7z” que está no ADS atrelado ao “Sputnik.jpg”, copiou-se o referido arquivo JPG para outro diretório, que foi inicializado por meio do comando cd da seguinte forma:

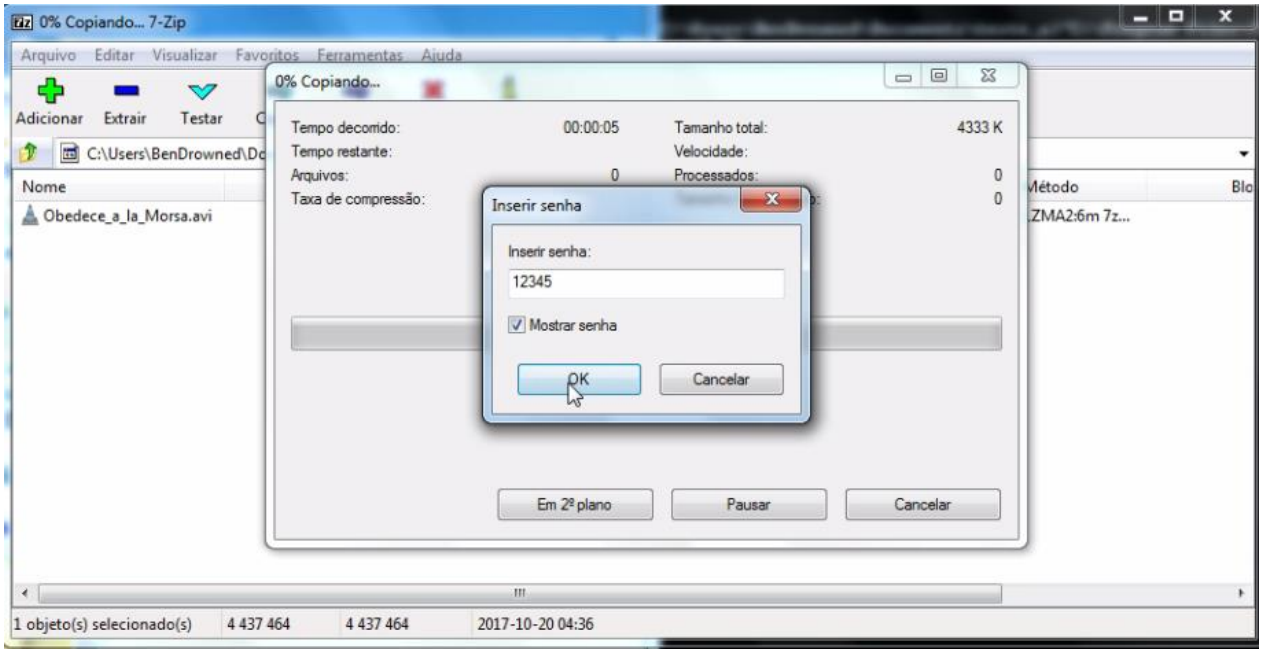
```
cd C:\Users\BenDrowned\Documents\teste_a
```

Após esta entrada, inicializou-se o software 7zip declarando o caminho do mesmo no prompt de comando para que o mesmo abrisse o arquivo “b.7z” que está contido no “Sputnik.jpg”, utilizando a seguinte entrada:

```
“C:\Program Files\7-Zip\7zFM.exe” Sputnik.jpg:b.7z
```

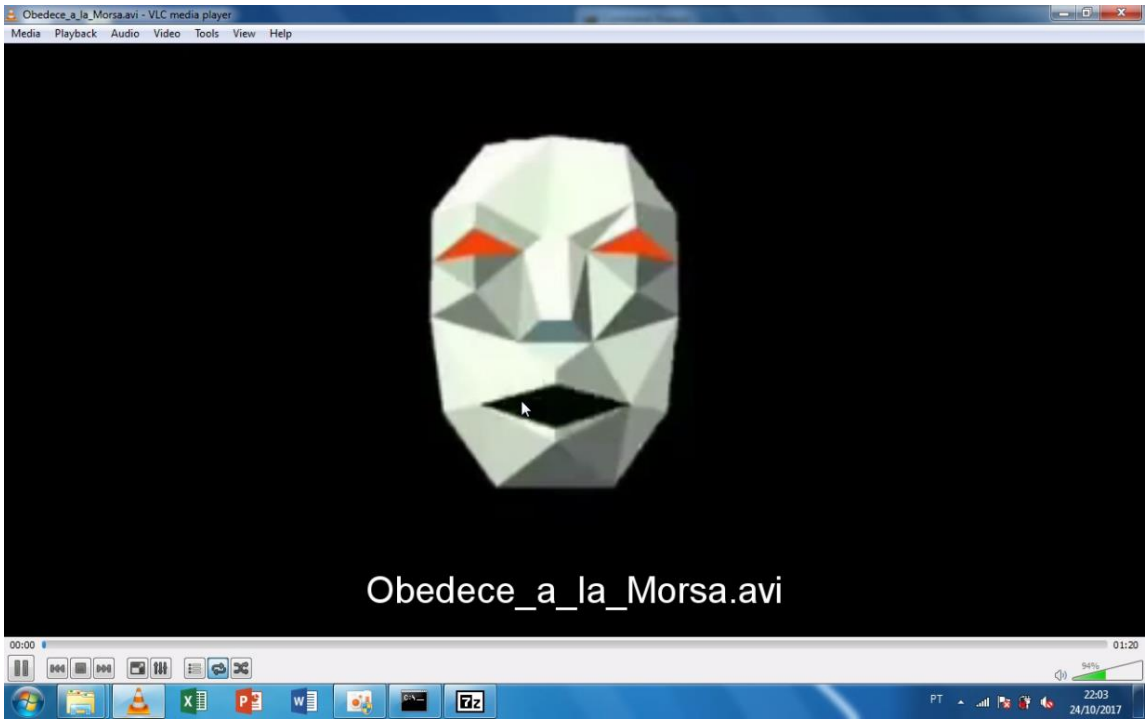
Logo em seguida o 7zip abre o arquivo compactado, e quando o usuário tenta visualizar o arquivo dentro do mesmo a senha é requisitada, como mostra a figura 53. Após a senha ser inserida, o arquivo pode ser visualizado normalmente, como exposto na figura 54.

Figura 53 - Arquivo compactado oculto em ADS sendo inicializado



Fonte: Autor (2017)

Figura 54 - Vídeo dentro do arquivo 7Z sendo inicializado após o fornecimento da senha

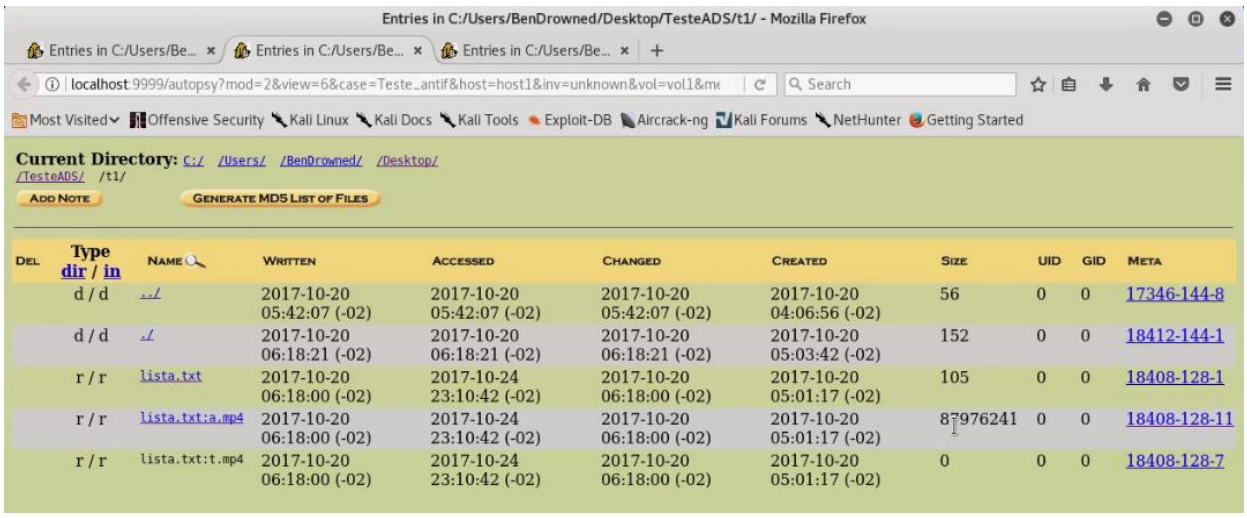


Fonte: Autor (2017)

4.3.2. Análise no ADS

Repetindo-se o processo de inicialização de análise de partição com o Autopsy realizado na situação anterior, navegou-se para o diretório o qual o teste de ocultação de dados por ADS foi realizado, e logo constata-se que o próprio Autopsy revela as bifurcações por ADS existentes como na figura 55.

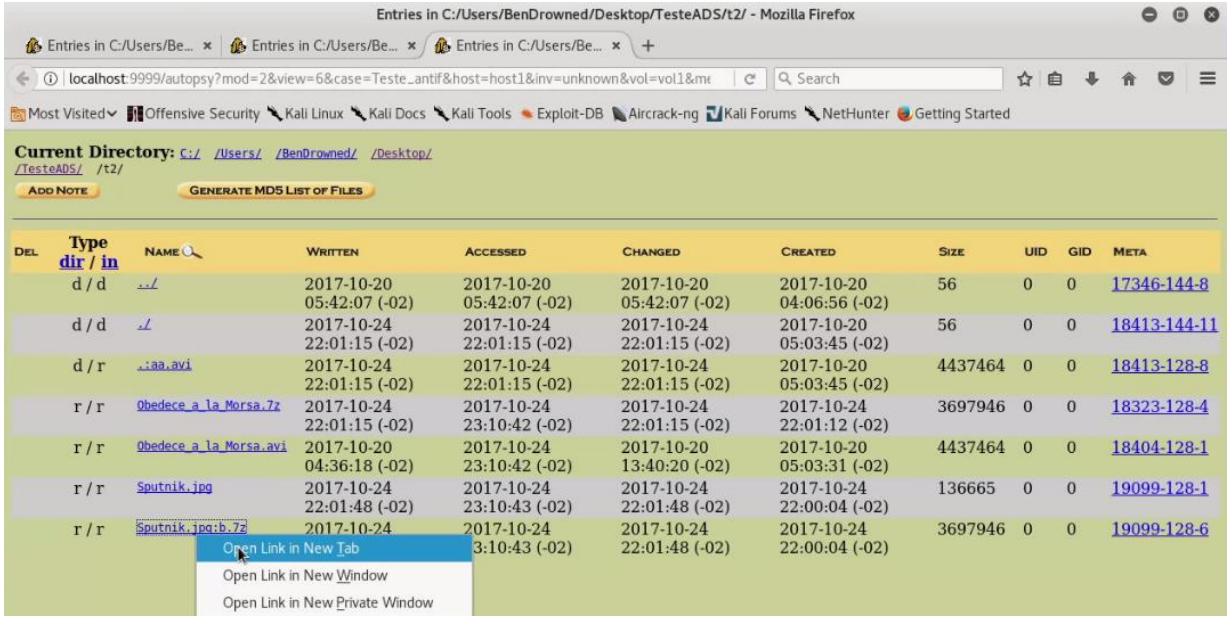
Figura 55 - Diretório o qual o teste de ADS foi realizado exposto no Autopsy



Fonte: Autor (2017)

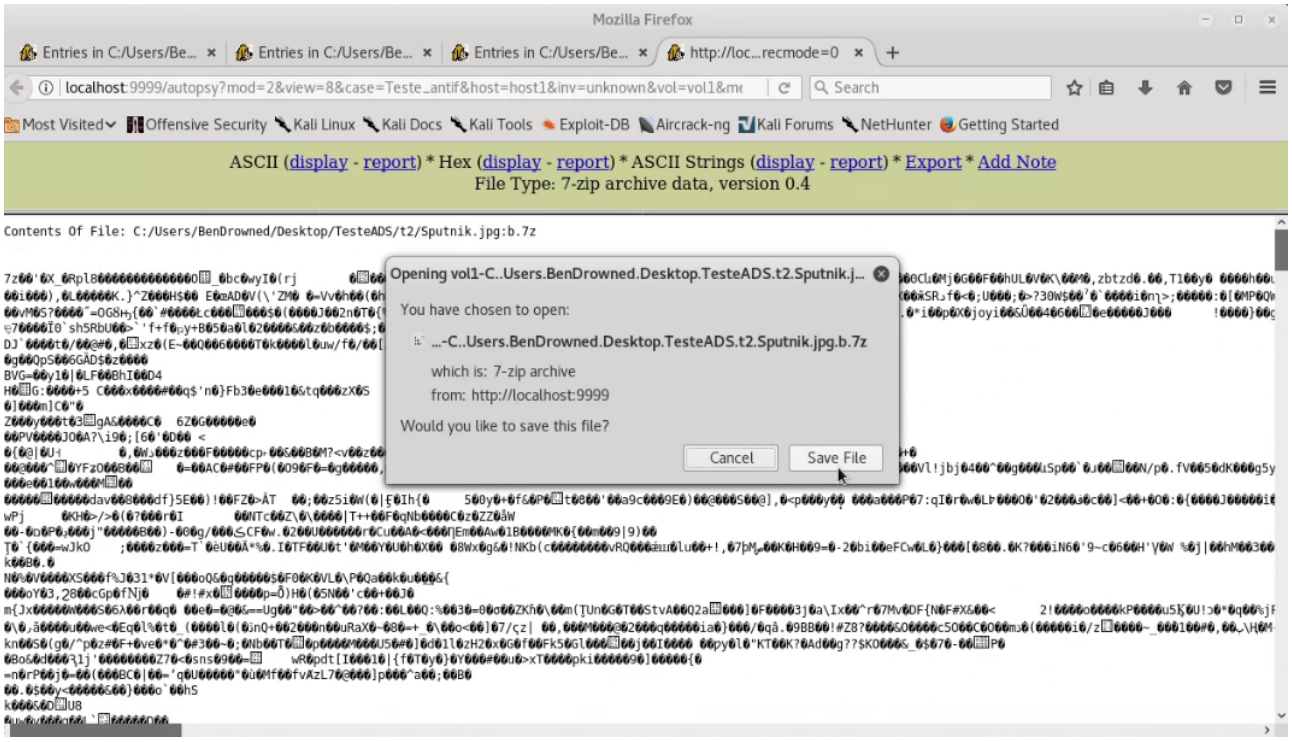
Percebe-se que o Autopsy permite também que se exporte diretamente esses arquivos para análise sem a necessidade de comandos adicionais para referenciar o ADS, como exposto nas figuras 56 e 57, aonde o arquivo “b.7z” embutido originalmente no arquivo “Sputnik.jpg” pôde ser analisado e extraído diretamente.

Figura 56 - Arquivo oculto em ADS sendo visualizado e acessado diretamente por meio do Autopsy



Fonte: Autor (2017)

Figura 57 - Arquivo oculto por ADS sendo exportado diretamente pelo Autopsy



Fonte: Autor (2017)

4.4. CONCLUSÃO SOBRE OS TESTES

De acordo com o teste, a esteganografia se mostrou uma técnica bastante desafiadora no que diz respeito à sua detecção e análise, visto que por manipular os bits menos significantes das cores dos pixels a mesma muito dificilmente seria detectada sem o uso de algum mecanismo, como o que foi apresentado. Além disso, mesmo após a detecção da aplicação desta técnica, a extração do conteúdo oculto na imagem em questão dependeria da interpretação de todos os bits de pixels modificados na imagem, sendo que essa identificação em si já representaria uma tarefa muito improvável de ser realizada sem a presença do arquivo de imagem original, com o agravante de o conteúdo oculto nesses pixels pode estar comprimido e criptografado, como é o caso do teste realizado, o que torna bem menos viável a recuperação destas informações. Nota-se, porém, que o usuário necessita possuir o

software usado para ocultar a mensagem para poder acessar o conteúdo embutido na imagem, logo em situações em que há suspeita do uso desta técnica a forma mais viável de recuperar estas informações escondidas é procurando no computador periciado possíveis evidências que apontem para o software usado para essa finalidade.

No que diz respeito ao teste do uso do file slack verificou-se que os dados inseridos na área não alocada do arquivo ficam alocados em segurança no mesmo e sua presença não causa nenhuma interferência no conteúdo original do arquivo, torna esta técnica bastante interessante nesse aspecto, porém observa-se que a quantidade de informação inserida não pode ultrapassar os limites do slack. Observou-se também que o conteúdo oculto pode ser recuperado por meio de qualquer editor hexadecimal já que a informação só pode ser inserida em formato de texto, porém existe uma boa possibilidade desse aspecto não ser analisado, pois em uma situação real um computador pessoal pode conter um número muito grande de arquivos, o que tornaria bastante improvável que um profissional de perícia optasse por analisar individualmente cada arquivo no modo hexadecimal sem ter algum indício de alguma informação foi inserida deste modo.

Em relação ao ADS observou-se que esta técnica esconde efetivamente arquivos de qualquer tamanho (respeitando-se os limites disponíveis na memória em questão) em arquivos muito menores sem que haja qualquer indício imediato de que exista uma bifurcação em relação ao arquivo que está ocultando outro, porém notou-se que, além da mesma só funcionar em sistemas de arquivos NTFS, com o uso da ferramenta correta não é difícil detectar a presença destas bifurcações e recuperar o conteúdo oculto.

5. DESTRUIÇÃO DE DADOS

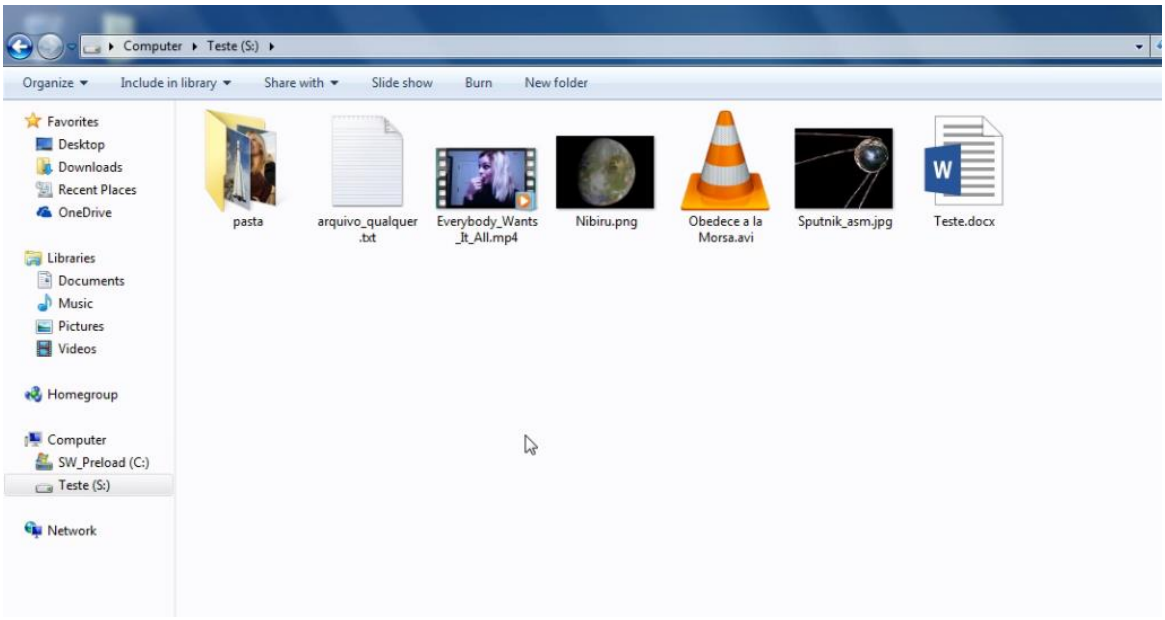
Destruição de dados é uma das formas mais diretas de impedir que determinada informação seja encontrada ou recuperada por uma terceira parte, porém quando um arquivo é apagado de forma “convencional” pelo sistema operacional Windows as informações geralmente permanecem em grande maioria no mesmo lugar até o momento que a região que continha o arquivo seja sobrescrevida com outra informação, como afirma Cozma (2011, online, tradução nossa) “os arquivos que são excluídos no Windows não são realmente excluídos - apenas os links para onde os arquivos estão localizados em seu disco rígido são removidos”.

Neste capítulo é realizado um teste de destruição de dados o qual um conjunto de arquivos contidos em uma partição é apagado primeiro normalmente e depois com um software especializado, e então a efetividade destas formas de destruição de dados é testada por meio de um software próprio para recuperação de arquivos.

5.1. DESTRUIÇÃO DE DADOS “CONVENCIONAL” E TENTATIVA DE RECUPERAÇÃO DE DADOS

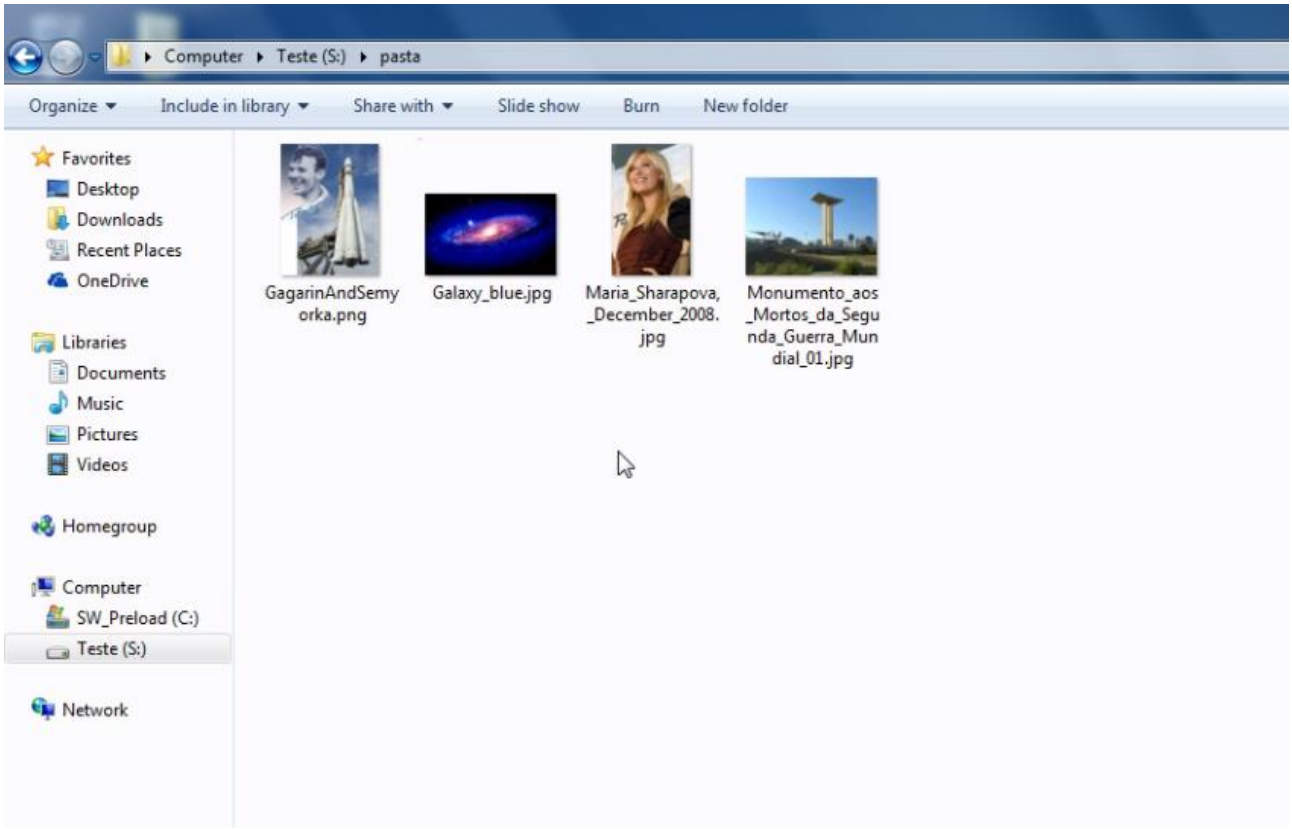
Para o primeiro teste reuniu-se um determinado conjunto de arquivos em uma partição reservada para o teste, sendo que este conjunto é composto por seis arquivos na raiz da partição (“arquivo_qualquer.txt”, “Everybody_Wants_It_All.mp4”, “Nibiru.png”, “Obedece a la Morsa.avi”, “Sputnik_asm.jpg” e “Teste.docx”), conforme a figura 58, e quatro arquivos dentro de um diretório denominado “pasta” (“GagarinAndSemyorka.png”, “Galaxy_blue.jpg”, “Maria_Sharapova_December_2008.jpg” e “Monumento_aos_Mortos_da_Segunda_Guerra_Mundial_01.jpg”), conforme a figura 59, resultando em um total de dez arquivos.

Figura 58 - Arquivos localizados na raiz da partição de teste



Fonte: Autor (2017)

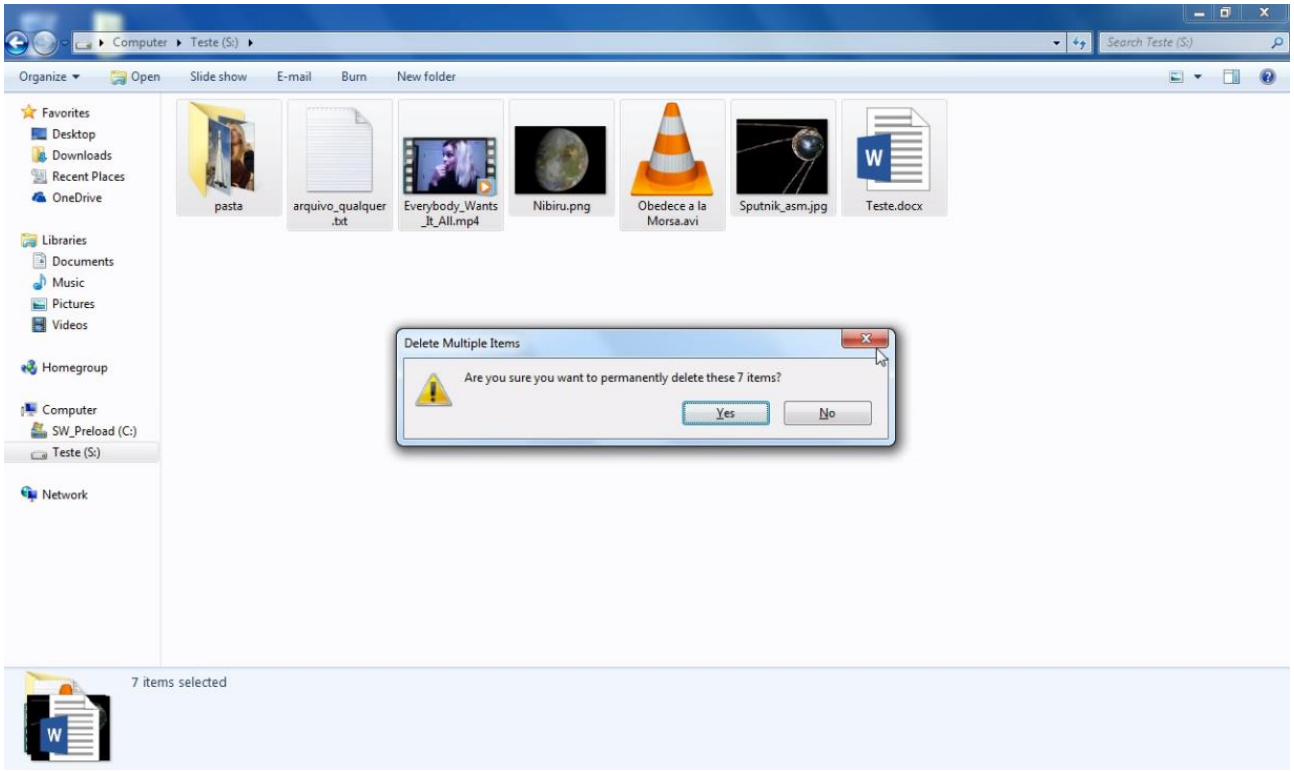
Figura 59 - Arquivos localizados no diretório "pasta" dentro da partição de teste



Fonte: Autor (2017)

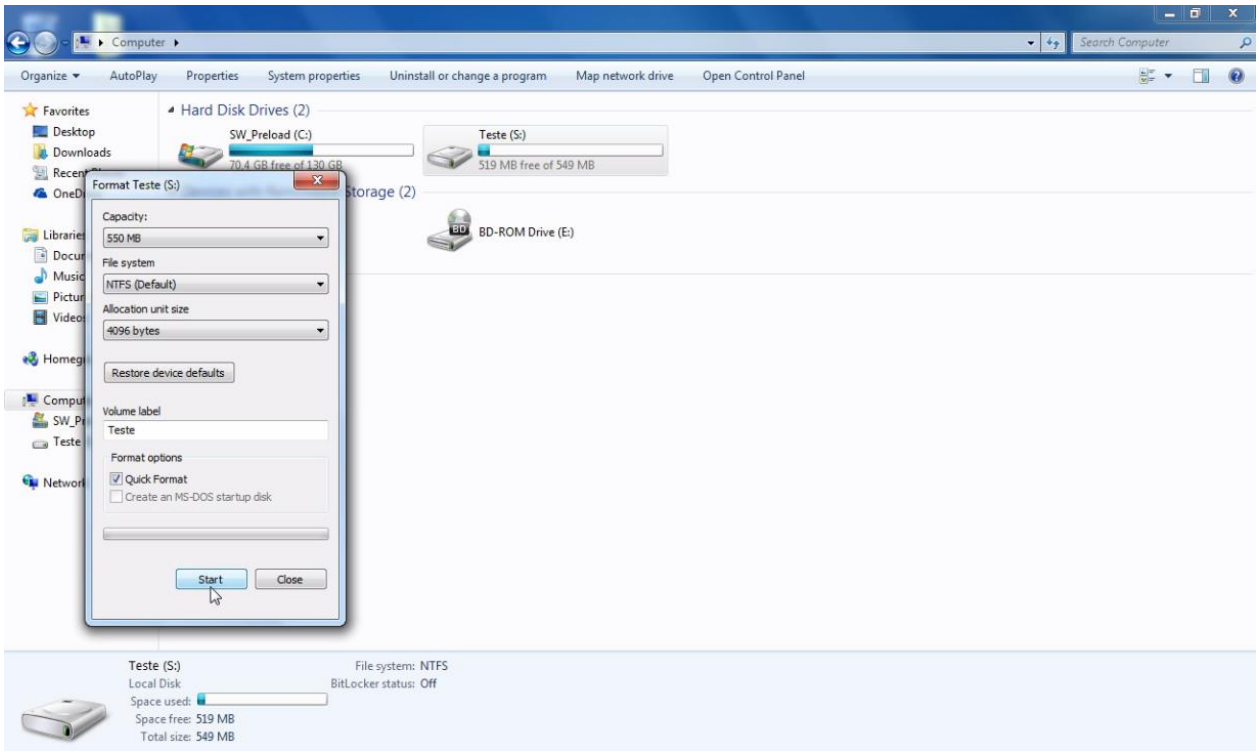
Em seguida, a partir do diretório raiz da partição todo o conteúdo contido na mesma é selecionado e deletado de forma “permanente” (pressionando-se as teclas SHIFT e Delete ao mesmo tempo), como exposto na figura 60, e em seguida é realizada uma formatação rápida na partição para observar se os dados poderão ser recuperados, como mostra a figura 61.

Figura 60 - Arquivos sendo deletados de forma "definitiva" para o Sistema Operacional



Fonte: Autor (2017)

Figura 61 - Partição de teste sendo formatada



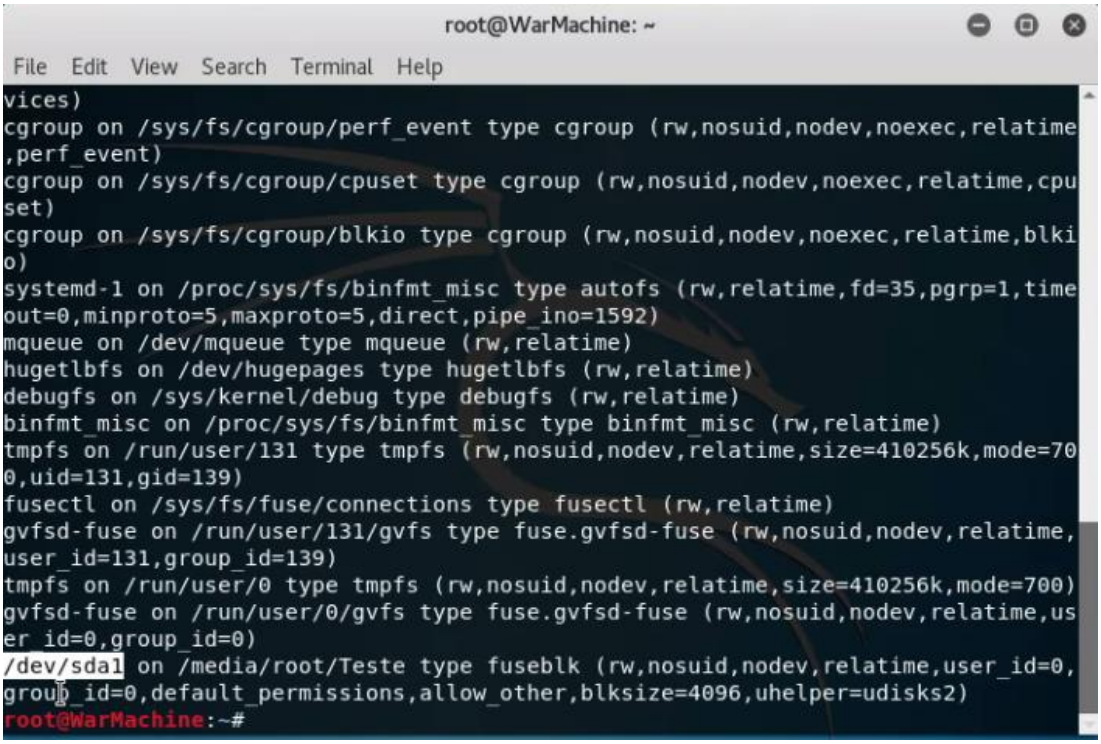
Fonte: Autor (2017)

Após a realização da formatação, procedeu-se para o sistema utilizado para as análises, inicializando-se a ferramenta Foremost presente no mesmo. Após identificar a localização da partição em questão por meio do comando mount (figura 62) foi utilizado a seguinte entrada no Terminal para invocar a ferramenta:

```
foremost -i /dev/sda1 -o '/root/Desktop/analises/3' -v
```

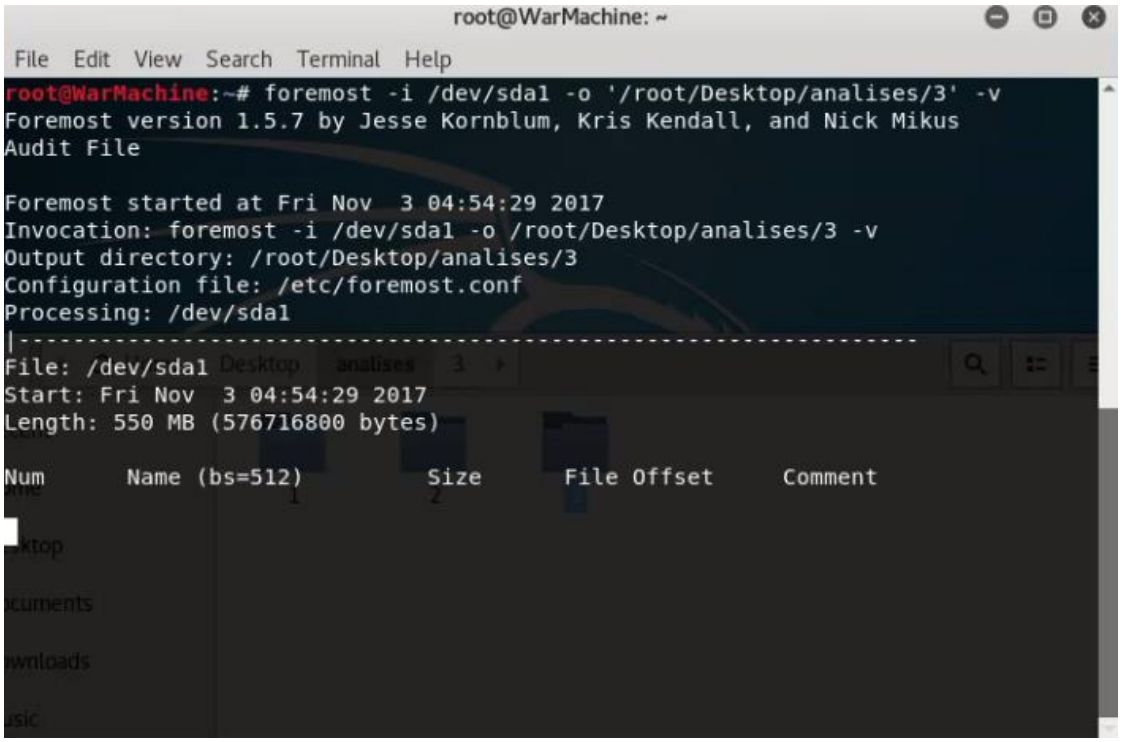
Aonde o comando “foremost” inicializa o software, o parâmetro “-i” define a unidade, partição ou arquivo a ser analisado, o parâmetro “-o” define o diretório de saída aonde os arquivos recuperados serão alocados e o parâmetro “-v ” faz a ferramento rodar em modo “verboso”, exibindo todas as informações do processo no Terminal, como exposto na figura 63.

Figura 62 - Local de montagem da partição de teste através do comando "mount"



Fonte: Autor (2017)

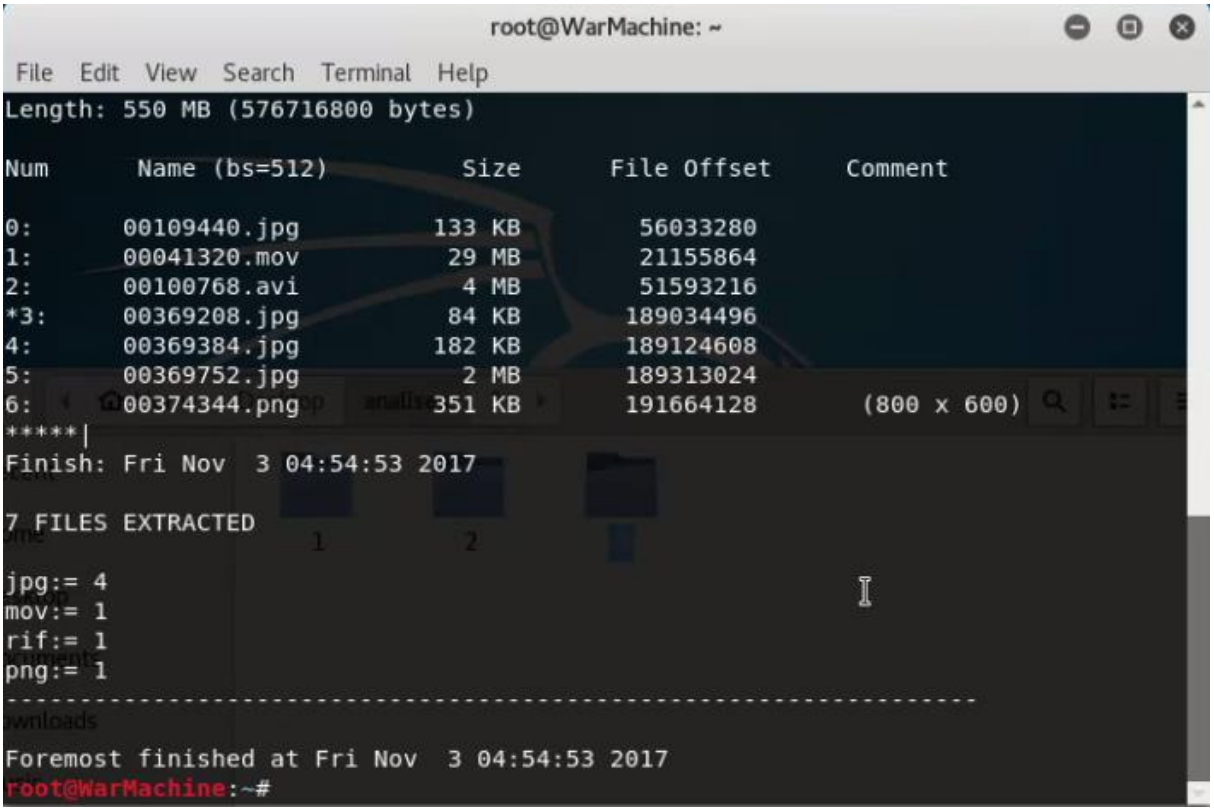
Figura 63 - Inicialização do foremost no primeiro teste de destruição de dados



Fonte: Autor (2017)

Após a ferramenta concluir o processo de recuperação de arquivos da partição nota-se que foram recuperados sete dos dez arquivos presentes na partição, como exposto na figura 64, sendo quatro arquivos JPG, um no formato PNG, um no formato AVI (que a ferramenta indica no final como sendo do formato RIF) e um arquivo MOV.

Figura 64 - Resultado do foremost no primeiro teste de destruição de dados



Fonte: Autor (2017)

Nota-se que uma parte do conteúdo da partição foi impedida de ser recuperada deste modo, porém a maior parte dos arquivos que foram destruídos foram resgatados apesar da formatação. Observa-se, porém, que o foremost não conserva os nomes originais dos arquivos recuperados, o que apesar de não ser um obstáculo que possa evitar o uso dos arquivos recuperados como evidências pode requisitar na necessidade da realização de análises adicionais de outras ferramentas caso mais informações sobre os arquivos seja necessária.

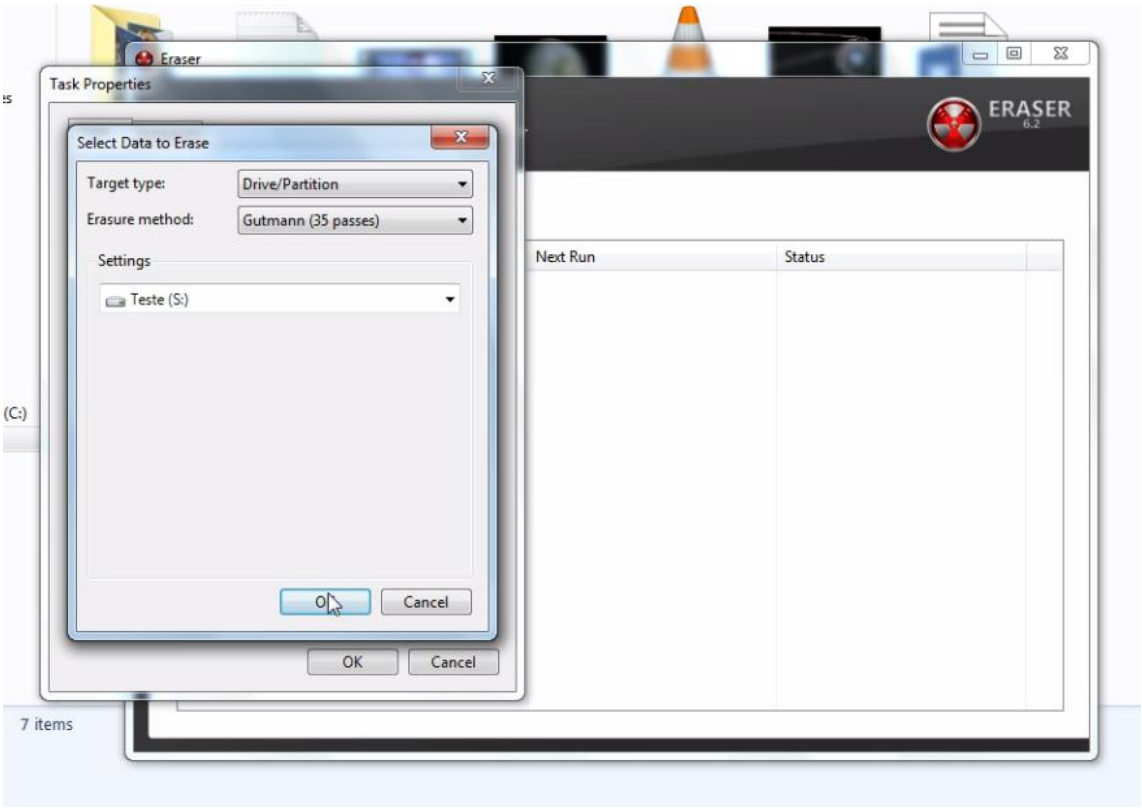
5.2. DESTRUIÇÃO DE DADOS COM O ERASER E TENTATIVA DE RECUPERAÇÃO DE DADOS

O Eraser, como explicado no capítulo 3, é um software designado para destruir dados confidenciais de forma segura, contendo vários algoritmos de destruição de dados. Para o teste em questão será utilizado o método Gutmann, que consiste em trinta e cinco passos no qual cada um caractere é usado para sobrescrever a memória e segundo Fisher (2017, online, tradução nossa) “O método Gutmann usa um caractere aleatório para os 4 primeiros e os últimas 4 passos, mas usa um padrão complexo de substituição do passo 5 ao passo 31”.

É importante ressaltar que a técnica não é considerada a mais otimizada para a realização da destruição de dados, pois segundo o próprio Gutmann (1996, online, tradução nossa) “se você estiver usando uma unidade que usa tecnologia de codificação X, você só precisa executar os passos específicos para X, e você nunca precisará executar todos os 35 passos”, logo boa parte das 35 etapas do método não são necessárias para o processo, porém, devido ao foco do trabalho estar voltado mais para eficiência das técnicas, esta será utilizada no experimento devido ao fato da mesma abranger a grande maioria das codificações existentes devido ao grande número de passos, tornando bastante improvável a possibilidade de algum arquivo não ser sobrescrito por completo.

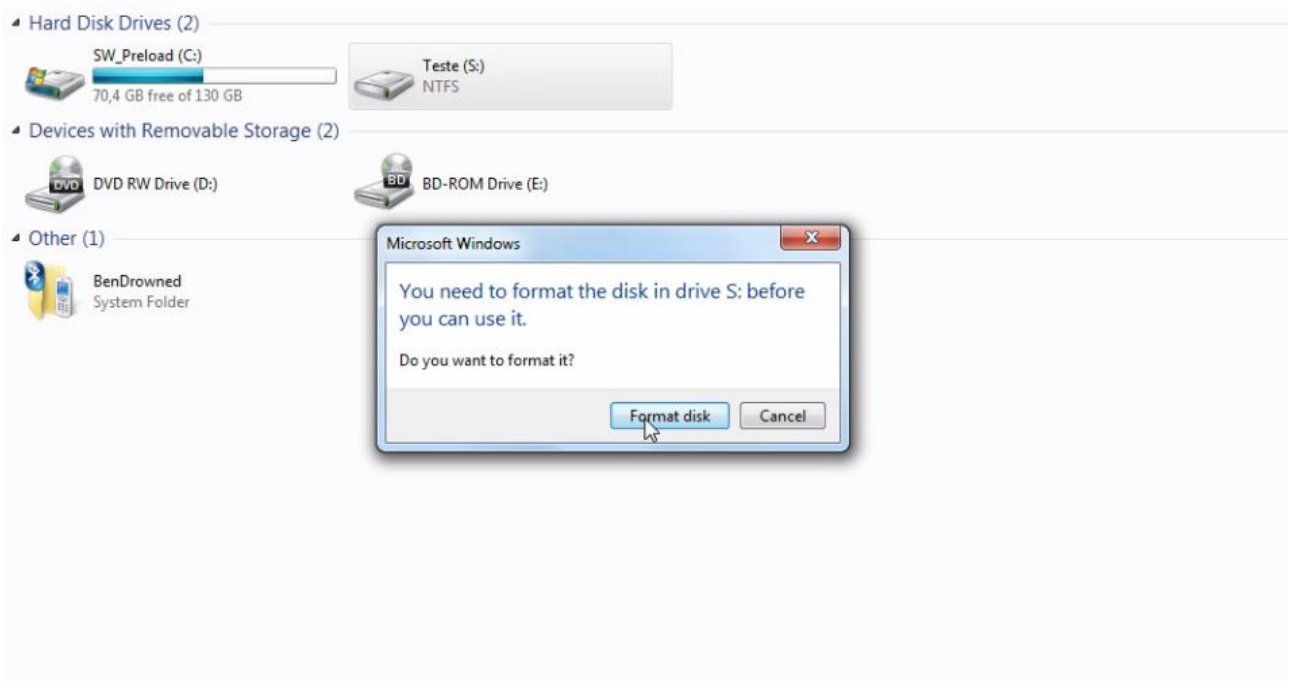
Tendo essas informações em vista, reuniu-se novamente os mesmos arquivos utilizados no experimento de destruição de dados anterior na partição de teste, e então inicializou-se o programa Eraser e adicionou-se uma nova tarefa à lista de tarefas do programa, cujo o objetivo é da mesma é apagar o conteúdo da partição de teste através do método Gutmann, como exposto na figura 65, e após o processo ser concluído a partição requisita ser formatada novamente para poder ser utilizada, como mostra a figura 66, e então é realizada uma formatação rápida do mesmo tipo da que foi usada no teste anterior.

Figura 65 - Criação de uma nova tarefa no Eraser



Fonte: Autor (2017)

Figura 66 - Requisição do sistema para formatar a partição de teste

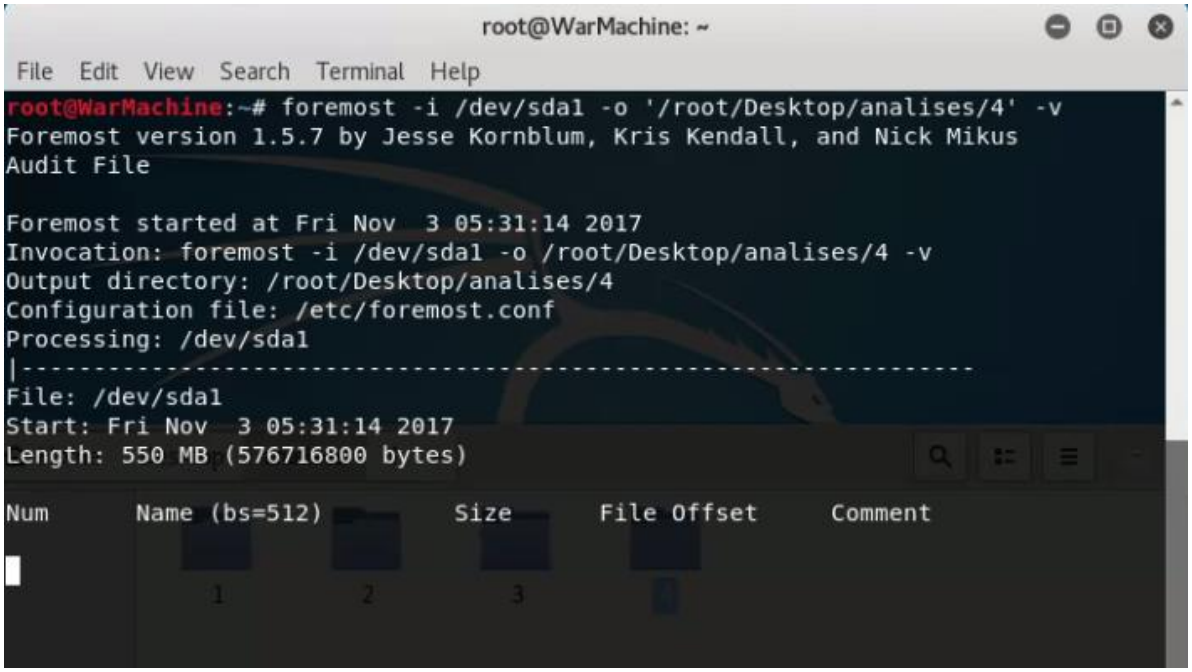


Fonte: Autor (2017)

Após esta formatação a partição de teste retorna ao estado vazio, e então retorna-se ao Kali Linux para realizar a análise, utilizando-se novamente o Foremost da mesma forma que no teste anterior, alterando-se somente o diretório de saída, como mostra a imagem 67, de acordo com a seguinte entrada:

```
foremost -i /dev/sda1 -o ‘/root/Desktop/analises/4’ -v
```

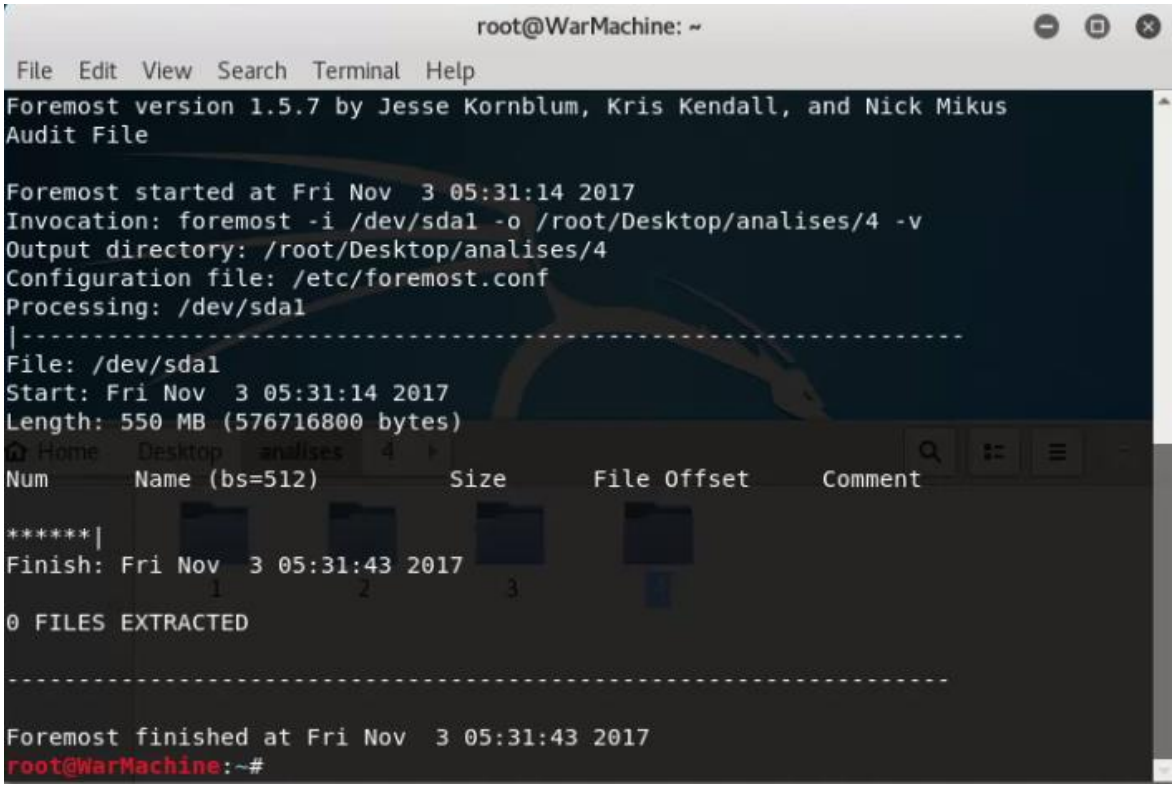
Figura 67 - Inicialização do foremost no segundo teste de destruição de dados



Fonte: Autor (2017)

Após a ferramenta terminar de executar o processo de recuperação de dados nota-se que dessa vez não houveram arquivos recuperados, como mostra a figura 68, o que indica que as informações foram todas sobrescritas com sucesso.

Figura 68 - Resultado do foremost no segundo teste de destruição de dados



Fonte: Autor (2017)

5.3. CONCLUSÃO SOBRE OS TESTES

Observou-se que o uso de um método de destruição de dados seguro fez uma grande diferença no que pode ser recuperado da partição a qual os arquivos foram deletados, pois quando os arquivos são deletados normalmente pelo sistema operacional, mesmo quando de forma “definitiva”, geralmente o sistema operacional apenas remove a referência da existência do arquivo em questão do registro do sistema de arquivos, porém o conteúdo permanece fisicamente nos setores do disco rígido até ser sobrescrito, que é exatamente o que o método de destruição de dados do Eraser realiza, visto que ele substitui o conteúdo apagado por outros valores de acordo com o algoritmo selecionado pelo usuário para a operação.

6. CONSIDERAÇÕES FINAIS

Conforme os meios digitais tornam-se mais integrados à sociedade como um todo, a quantidade de delitos referentes a esses meios, ou que tangenciam os mesmos, continua a crescer e fazer vítimas. Portanto é importante que a investigação digital forense tenha efetividade em identificar a ocorrência desses crimes, a fim de que o meio virtual não se torne um ambiente hostil, onde essas práticas indevidas ocorram livremente.

Este trabalho visou justamente analisar técnicas utilizadas para evitar a detecção de informações pelos métodos periciais, ou seja, técnicas que oferecem barreiras à investigação que busca identificar a relação de causa e efeito e comprovar delitos cometidos no ambiente virtual.

Para que o conhecimento gerado por meio deste estudo possa contribuir na execução de perícias realizadas no ambiente proposto, é necessária uma abordagem fortemente calcada em ferramentas tecnológicas de análise computacional, uma vez que as aplicações das técnicas anti-forenses que foram realizadas revelaram, quando confrontadas com determinadas ferramentas de perícia, características específicas que indicam qual técnica foi utilizada e as suas particularidades.

Sempre que é possível essa forma de operacionalização, as informações periciais obtidas podem fornecer uma grande vantagem tática ao profissional que está realizando a investigação digital forense, caso o mesmo se encontre em um cenário como o proposto na monografia.

6.1. DIFICULDADES ENCONTRADAS

Os maiores desafios encontrados durante a realização do presente estudo se deram em relação da execução das técnicas e da avaliação das mesmas utilizando ferramentas especializadas, uma vez que não haveria modo eficiente de avaliar a eficácia e as particularidades das técnicas analisadas sem as ferramentas de perícia. Porém, esta dificuldade foi superada devido às comunidades de

desenvolvimento de Software Livre e Software de Código Aberto, que criam inúmeros programas, muitos deles considerados bastante eficientes, para as mais variadas aplicações na computação, incluindo perícia digital. Na maioria das vezes, os citados programas permitem, por meio das suas licenças, que os usuários utilizem os mesmos para qualquer propósito, tornando possível a realização desta pesquisa (e certamente de muitas outras) com a execução de experimento prático das aplicações virtuais, sem a necessidade de gastos adicionais com softwares proprietários.

6.2. PROPOSTAS PARA TRABALHOS FUTUROS

Para futuros trabalhos uma possibilidade de proposta seria, além de reavaliar as análises aqui realizadas, utilizar os resultados dos testes para agregar melhorias nas próprias ferramentas que se enquadram em software livre ou código aberto, ou mesmo utilizar essas informações para desenvolver uma ferramenta completamente nova, que possua uma eficiência melhor em algum aspecto que aquelas adotadas neste trabalho, as quais possam não ter a abrangência para trabalhos de natureza mais complexa. Outra possibilidade de continuação deste estudo seria de se expandir a análise, que foi realizada somente no sistema operacional mais utilizado atualmente, para outros sistemas operacionais, principalmente aqueles que integram diferentes sistemas de arquivos concorrentes, pois isso poderia expor mais aspectos das técnicas em diferentes cenários e certamente geraria mais informação que poderia ser utilizada em favor das investigações digitais.

REFERENCIAS

ABD-EL-BARR, Mostafar e EL-REWINI, Hesham. **Fundamentals of Computer Organization and Architecture**, New Jersey: John Wiley & Sons, 2005

ALMEIDA, Rafael, **Esteganografia e Esteganálise: transmissão e detecção de informações ocultas em imagens digitais**, 2011. Disponível em: <https://www.vivaolinux.com.br/artigo/Esteganografia-e-Esteganalise-transmissao-e-deteccao-de-informacoes-ocultas-em-imagens-digitais/?pagina=4>. Acesso em 15/09/2017

ALTERNATIVETO, **Windows 7 Alternatives and Similar Software**, 2017. Disponível em: <https://alternativeto.net/software/windows-7>. Acesso em: 02/10/2017

ARNTZ, Pieter. **Introduction to Alternate Data Streams**. 2016. Disponível em: <https://blog.malwarebytes.com/101/2015/07/introduction-to-alternate-data-streams/>. Acesso em 30/09/2017

BELLIS, Mary, **The Unusual History of Microsoft Windows**, 2017. Disponível em: <https://www.thoughtco.com/unusual-history-of-microsoft-windows-1992140>. Acesso em 25/09/2017

BOTH, David, **An introduction to Linux's EXT4 filesystem**, 2017. Disponível em: <https://opensource.com/article/17/5/introduction-ext4-filesystem>. Acesso em 25/09/2017

CARRIER, Brian. **File System Forensic Analysis**, Upper Saddle River, NJ: Addison-Wesley, 2005

_____. **The Sleuth Kit**, 2017. Disponível em: <https://www.sleuthkit.org/sleuthkit/index.php>. Acesso em 30/09/2017

CASEY, Eoghan. **Digital Evidence and Computer Crime**, 3ª Edição, Waltham, MA: Elsevier, 2011

CHUNG, Hyunji, PARK, Jungheum e LEE, Sangjin, **Digital forensic approaches for Amazon Alexa ecosystem**, 2017. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1742287617301974>. Acesso em 07/09/2017

COZMA, Nicole. **Permanently delete files in Windows 7**, 2011. Disponível em: <https://www.cnet.com/how-to/permanently-delete-files-in-windows-7/>. Acesso em 30/09/2017

CRISTO, Fernando de, PREUSS, Evandro e FRANCISCATTO, Roberto. **Arquitetura de Computadores**, Frederico Westphalen: Universidade Federal de Santa Maria, Colégio Agrícola de Frederico Westphalen, 2013

DEITEL, Harvey M., DEITEL, Paul J. e CHOFFNES, David R. **Sistemas operacionais**, 3ª edição, São Paulo: Pearson Prentice Hall, 2005

ERASER, **Eraser – Erase Files form Hard Drives**, 2017. Disponível em: <https://eraser.heidi.ie/>. Acesso em 25/09/2017
FISHER, Tim, What is the Gutmann Method?, 2017. Disponível em: <https://www.lifewire.com/gutmann-method-2625891>. Acesso em 25/09/2017

FREE SOFTWARE FOUNDATION. **What is Free Software**, 2017. Disponível em: <http://www.fsf.org/about/what-is-free-software>. Acesso em 25/09/2017

GNU. **O que é o software livre?** 2017. Disponível em: <https://www.gnu.org/philosophy/free-sw.pt-br.html>. Acesso em 25/09/2017

GUTMANN, Peter. **Secure Deletion of Data from Magnetic and Solid-State Memory**, 1996. Disponível em: <https://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html>. Acesso em 25/09/2017

HALDER; JAISHANKAR. **Sistemas Operacionais Modernos**. São Paulo: Information Science Reference, 2011

JULIO; BRAZIL; ALBUQUERQUE. **Esteganografia e suas Aplicações**, 2007. Disponível em: <http://ceseg.inf.ufpr.br/anais/2007/minicursos/cap2-esteganografia.pdf>. Acesso em 30/09/2017

LINUX FOUNDATION, About - **The Linux Foundation**, 2017. Disponível em: <https://www.linuxfoundation.org/about/>. Acesso em 15/09/2017

MANSON, Don. **The Dimensions of "Cyber Crime"**. 2008. Disponível em: <https://mafiadoc.com/download/the-dimensions-of-cyber-crime-university-of-mississippi_59d0ab7d1723ddd7865bf1b3.html>. Acesso em 25/09/2017

MICROSOFT, **Visão geral dos sistemas de arquivo FAT, HPFS e NTFS**, 2016. Disponível em: <https://support.microsoft.com/pt-br/help/100108/overview-of-fat--hpfs--and-ntfs-file-systems>. Acesso em 25/09/2017

MORIMOTO, Carlos E. **O fim dos pendrives**. 2008. Disponível em: <http://www.hardware.com.br/dicas/fim-dos-pendrives.html>. Acesso em 02/10/2017

NET MARKET SHARE. **Desktop Operating System Market Share**. 2017. Disponível em: <https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0>. Acesso em 25/09/2017

OFFENSIVE SECURITY. **Foremost | Penetration Testing Tools**, 2014. Disponível em: <https://tools.kali.org/forensics/foremost>. Acesso em 30/09/2017

_____. **What is Kali Linux? | Kali Linux**, 2017. Disponível em: <https://docs.kali.org /introduction/what-is-kali-linux>. Acesso em 25/09/2017

OPEN SOURCE INITIATIVE, **History of the OSI**, 2012. Disponível em: <https://opensource.org/history>. Acesso em 30/09/2017

PAAR, Christof; PELZL, Jan. **Understanding Cryptography**, Heidelberg: Springer, 2009

PAVLOV, Igor, **7-zip**, 2016. Disponível em: <http://www.7-zip.org/>. Acesso em 02/10/2017

PEREIRA et. al, **Forense Computacional: fundamentos, tecnologias e desafios atuais**, 2007. Disponível em: <http://ceseg.inf.ufpr.br/anais/2007/minicursos/cap1-forense.pdf>. Acesso em 07/09/2017

RUSSINOVICH, Mark; SOLOMON, David. **Microsoft Windows Internals**, 4ª Edição, Washington: Microsoft Press, 2005

SOURCEFORGE, **Steghide**, 2003. Disponível em: <http://steghide.sourceforge.net/>. Acesso em 02/10/2017

STALLINGS, William. **Criptografia e segurança de redes: princípios e práticas**, 6ª Edição, São Paulo: Pearson Education do Brasil, 2015

STALLMAN, Richard M. **Free Software, Free Society**, 3ª Edição, Boston: Free Software Foundation, 2015

TANENBAUM, Andrew. **Organização Estruturada de Computadores**, 5ª Edição. São Paulo: Pearson Prentice Hall, 2007

_____. **Sistemas Operacionais Modernos**, 3ª Edição. São Paulo: Pearson Prentice Hall, 2009