UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS DE ITAPAJÉ

CURSO DE GRADUAÇÃO EM SEGURANÇA DA INFORMAÇÃO

CARLOS GABRIEL DOS SANTOS ALVES

UM ESTUDO DE FERRAMENTAS OPEN-SOURCE PARA PERÍCIA FORENSE EM DISPOSITIVOS COM ANDROID 14

ITAPAJÉ

2024

CARLOS GABRIEL DOS SANTOS ALVES

UM ESTUDO DE FERRAMENTAS OPEN-SOURCE PARA PERÍCIA FORENSE EM DISPOSITIVOS COM ANDROID 14

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Segurança da Informação do Campus de Itapajé da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de Tecnólogo em Segurança da Informação.

Orientador: Prof. Dr. Israel Eduardo de Barros Filho.

ITAPAJÉ

2024

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

A478e Alves, Carlos Gabriel dos Santos.
Um Estudo de Ferramentas Open-Source para Perícia Forense em Dispositivos com Android 14 / Carlos Gabriel dos Santos Alves. – 2024.
64 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Itapajé, Curso de Análise e Desenvolvimento de Sistemas, Fortaleza, 2024.
Orientação: Prof. Dr. Israel Eduardo de Barros Filho.

1. perícia. 2. forense. 3. android. 4. dispositivos móveis. I. Título.

CDD 005.1

CARLOS GABRIEL DOS SANTOS ALVES

UM ESTUDO DE FERRAMENTAS OPEN-SOURCE PARA PERÍCIA FORENSE EM DISPOSITIVOS COM ANDROID 14

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Segurança da Informação do Campus de Itapajé da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de Tecnólogo em Segurança da Informação.

Aprovada em: 25/08/2024.

BANCA EXAMINADORA

Prof. Dr. Israel Eduardo de Barros Filho (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. João Henrique Gonçalves Medeiros Correia
Universidade Federal do Ceará (UFC)

Prof. Dr. Alyson Bezerra Nogueira Ribeiro
Universidade Federal do Ceará (UFC)

À minha família. Mãe, por ser a pessoa que cuidou de mim em toda essa trajetória e me proporcionou estar aqui e ser quem eu sou. Pai, que foi também quem me moldou a ser o homem que sou hoje, sei que estaria muito orgulhoso de ver até onde cheguei. Irmã, que foi minha maior motivação para seguir em frente, desde sempre.

AGRADECIMENTOS

Ao Prof. Dr. Israel Barros, pela excelente orientação.

Aos professores participantes da banca examinadora João Henrique e Alyson Bezerra pelo tempo, pelas valiosas colaborações e sugestões.

"O dever de um perito é dizer a verdade; no entanto, para isso é necessário: primeiro saber encontrá-la e, depois querer dizê-la. O primeiro é um problema científico, o segundo é um problema moral." (ROJAS, 1956)

RESUMO

O crescente uso de *smartphones* na última década traz consigo novos problemas. A utilização de dispositivos móveis para armazenar dados pessoas sensíveis cria uma nova vulnerabilidade a ser explorada por atacantes, que agora têm estes dispositivos como alvos. Tendo isso em vista, este trabalho tem como objetivo analisar ferramentas que se propõem a auxiliar em investigações destes dispositivos móveis, focando no sistema operacional mais utilizado, o Android. Esta avaliação é feita através de estudo de sete ferramentas *open-source* pré-selecionadas, sendo estas: AFLogical OSE, dd, Foremost, LiME - Linux Memory Extractor, AMExtractor, Volatility, Sistema IPED e Avilla Forensics; através de testes práticos e resultados concretos, baseados em experimentos e pesquisas. O estudo também demonstra diversos desafios e empecilhos encontrados ao tentar utilizar destas ferramentas, bem como guias para uma fácil instalação e execução das mesmas. Ao fim do trabalho, foi analisado que as ferramentas, em sua maioria, apresentaram dificuldade em obter êxito na execução de sua função. A ferramenta que obteve um melhor desempenho, baseado nos critérios estabelecidos foi, por fim, o AFLogical OSE.

Palavras-chave: perícia; forense; android; dispositivos móveis.

ABSTRACT

The increasing use of *smartphones* in the last decade brings with it new problems. The use of mobile devices to store people's sensitive data creates a new vulnerability to be exploited by adventurers, who now target these devices. With this in mind, this work aims to analyze tools that aim to assist in the investigation of these mobile devices, focusing on the most used operating system, Android. This evaluation is carried out through the study of seven pre-selected *open-source* tools, these being: AFLogical OSE, dd, Foremost, LiME - Linux Memory Extractor, AMExtractor, Volatility, IPED System and Avilla Forensics; through practical tests and concrete results, based on experiments and research. The study also demonstrates several challenges and peculiarities encountered when trying to use these tools, as well as guides for easy installation and execution of them. At the end of the work, the tools that, for the most part, had difficulty in successfully performing their function were analyzed. The tool that achieved the best performance, based on the defined criteria, was, finally, AFLogical OSE.

Keywords: expertise; forensics; android; mobile devices.

LISTA DE FIGURAS

SUMÁRIO

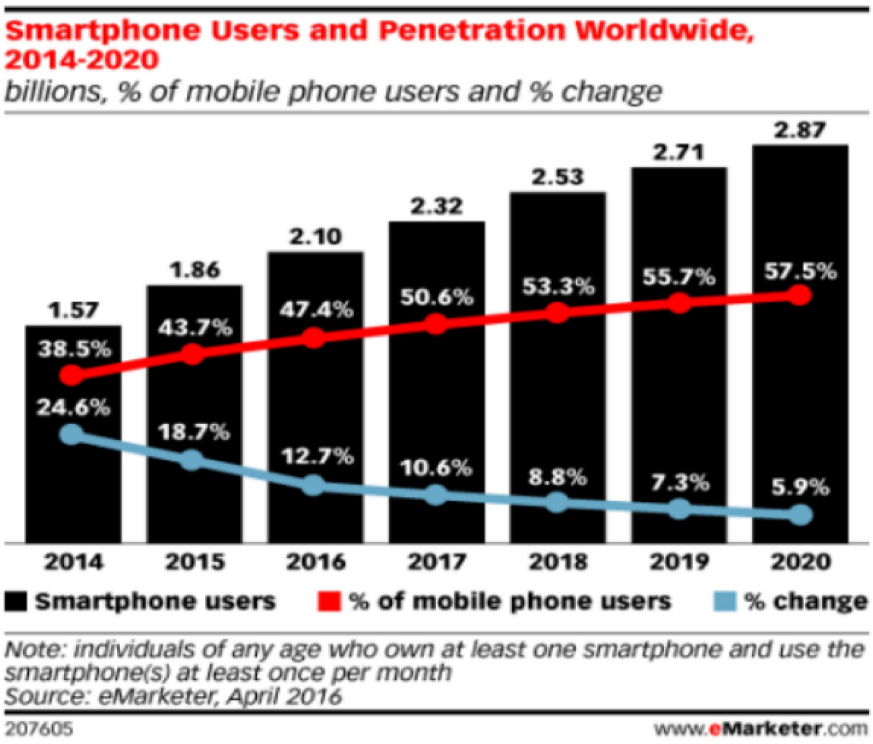
1 INTRODUÇÃO

O aumento do uso de *smartphones* entre a população nos dias atuais é evidente. Na Figura 1, este crescimento é evidenciado, mostrando um forte aumento no número de usuários de smartphones a partir do ano de 2014. Desde então, estes números seguem a tendência de crescimento. A principal razão disto é como os smartphones passaram a ser essenciais no dia a dia das pessoas. Sendo um dispositivo móvel que possui múltiplas funcionalidades como pesquisa, GPS, comunicação via texto ou via chamada de voz, e até mesmo administração e transação em contas bancárias.

Devido a estas funções dos smartphones, os mesmos passaram também a ser portadores de dados pessoais importantes, como número de documentos, cartões de crédito e conversas particulares. Tendo isto em vista, estes dispositivos tornaram-se "[...] fonte inestimável de evidência para investigações relativas a processos criminais, civis e até mesmo casos de alta importância" (Raquel Beatriz, 2016). O estudo da recuperação e análise dessas evidências digitais é chamado de **forense de dispositivos móveis**.

O procedimento de análise forense pode ser dividido em múltiplas etapas, com o objetivo de particionar um processo onde diversos desafios podem ser encontrados, aumentando

Figura 1 – Usuários de smartphones e sua penetração a nível mundial, 2014-2020



Fonte: eMarketer, 2016.

assim as chances de um resultado mais satisfatório. Além disso, existem várias ferramentas de coleta e análise de dados, que funcionam para diferentes cenários e diferentes etapas do processo forense.

Diferentes sistemas operacionais podem ser utilizados nos smartphones, como o Android, iOS, Windows, Blackberry, entre outros menos utilizados. Há também diferentes versões para cada um desses sistemas operacionais, dificultando ainda mais o processo de coleta e análise de dados nos dispositivos, visto que o procedimento em versões mais recentes apresenta mais desafios e complicações, em relação aos antecessores (Petter Anderson, 2020).

1.1 Objetivo geral

Este trabalho tem como objetivo realizar um estudo entre as ferramentas forense de coleta e análise de dados *open-source* selecionadas para este trabalho, testando-as e criando critérios para a comparação destas. O sistema operacional escolhido como foco para esta pesquisa, foi o Android, especificamente a versão 14, para verificar o quão bem as ferramentas operam em um sistema operacional moderno.

1.2 Objetivos específicos

Foram estabelecidos os seguintes objetivos específicos:

a) instalar e configurar com sucesso as ferramentas;

b) testar cada uma dessas ferramentas, dentro de um cenário pré-estabelecido;

c) buscar possíveis alternativas para as limitações das ferramentas;

d) elaborar conclusões baseadas nos resultados obtidos dos testes.

2 FUNDAMENTAÇÃO TEÓRICA

O presente trabalho tem como objetivo realizar uma análise abrangente sobre oito ferramentas de investigação forense já especificadas neste trabalho. Para tal, é essencial possuir o conhecimento a respeito da base de estudo e termos técnicos utilizados no ramo da ciência forense e afins, bem como a forma que isto afeta o processo de investigação.

2.1 Ciência Forense

As ciências forenses reúnem todos os conhecimentos científicos e técnicas utilizadas para investigar crimes e diversas questões jurídicas sendo estas civis, criminais ou administrativas.

A função do campo civil é o estudo e interpretação dos vestígios que caracterizam as infrações penais, a fim de esclarecer as infrações e cooperar com as autoridades responsáveis pela implementação da lei.

Nas investigações criminais, a principal tarefa do perito forense é confirmar o autor do crime ou excluir o envolvimento do(s) suspeito(s) - evitando a punição injusta de pessoas inocentes - através de métodos que permitam determinar com relativa precisão, por exemplo, se for um crime, uma pessoa estava no local do crime.

O principal papel desta ciência é auxiliar as investigações relativas às justiças civil e criminal, empregando métodos científicos para averiguar danos, mortes e crimes inexplicados. Ao estudar as provas coletadas durante a investigação, a perícia ajuda a identificar suspeitos e solucionar um crime específico, criando hipóteses sobre o ocorrido.

Na ciência forense, existem muitos campos que atuam separadamente, mas que finalmente se unem para fornecer resultados precisos e assim confirmar o autor do crime ou excluir o envolvimento dos suspeitos. Para que seu trabalho tenha autoridade, o perito deve ter experiência na sua área, mas para se tornar um especialista deve ter um conhecimento amplo e profundo, portanto, ser competente para formular seu relatório final.

2.2 Computação Forense

Também conhecida como forense digital, ciência forense da computação ou forenses cibernéticos, a computação forense combina a ciência da computação e a forense legal para coletar evidências digitais de maneira admissível em um tribunal de justiça.

Segundo Stanley Gusmão (2023), de forma semelhante a que os agentes responsáveis

pela aplicação da lei examinam as cenas dos crimes em busca de pistas, os investigadores forenses procuram nos dispositivos digitais provas que os advogados possam utilizar em investigações criminais, processos civis, investigações de crimes cibernéticos e outros assuntos relacionados com negócios e segurança nacional. E, tal como os seus homólogos responsáveis pela aplicação da lei, o investigador forense informático deve ser hábil não só na procura de provas digitais, mas também na recolha, manipulação e processamento das mesmas para garantir a sua exactidão e admissibilidade em tribunal.

Bem como as evidências físicas da cena do crime, as evidências digitais devem ser coletadas e tratadas corretamente. De outra forma, os dados e metadados poderão ser perdidos ou considerados inadmissíveis em um tribunal.

De acordo com André Jales (2023), as investigadores forenses computacionais são especialistas na busca e preservação de dados legalmente admissíveis. Eles sabem como coletar dados de fontes que a equipe interna de Tecnologia da Informação (TI) pode ignorar, como servidores externos e *Personal Computers* (PCs). Podem ajudar as organizações a desenvolver uma política forense de TI robusta que lhes poupará tempo e dinheiro na recolha de provas digitais, reduzindo o impacto do crime cibernético e protegendo as suas redes e sistemas contra ataques cibernéticos.

Além disso, especialistas em informática forense podem identificar falsificações de dados na tentativa de remover vestígios que conectem os fatos. Por exemplo, alterar a data e a hora das câmeras de segurança. E até mesmo documentos digitais fraudulentos.

Dentro do ambiente do tribunal, as provas forenses informáticas estão sujeitas aos requisitos habituais de provas digitais. Isto exige que as informações sejam autênticas, obtidas de forma confiável e aceitável. Diferentes países têm diretrizes e práticas específicas para recuperação de evidências.

Segundo o Protocolo de Minnesota sobre a Investigação de Mortes Potencialmente Ilícitas (2016), no Reino Unido, por exemplo, os examinadores geralmente seguem as orientações da Associação dos Chefes de Polícia, que ajudam a garantir a autenticidade e integridade das provas. Embora voluntárias, estas diretrizes são amplamente aceitas pelos tribunais do Reino Unido.

2.3 Dispositivos Investigados

A investigação forense digital busca englobar todos os dispositivos digitais que possam ser periciados, priorizando mais ainda dispositivos que possuem algum tipo de armazenamento de dados, indo desde computadores pessoais, até o dispositivos de Internet das Coisas (*Internet of Things* (IoT)), desde grandes aparelhos — geladeiras *smart*, carros autônomos — até objetos menores, como lâmpadas inteligentes e *smartwatches*.

Contudo, este trabalho busca abordar apenas as ferramentas e métodos de investigação forense que apliquem-se para os dispositivos com o sistema operacional Android, buscando entender como funciona todo o processo investigativo em volta deste.

2.4 Procedimentos de Investigação Forense em dispositivos com Android

Segundo Eoghan Casey (2011), a análise forense de dispositivos Android é um processo meticuloso que começa com a identificação do dispositivo a ser investigado. Esta etapa é crucial para definir o objetivo da investigação e determinar as ferramentas e técnicas que serão necessárias. Uma vez identificado o dispositivo, ele é coletado e armazenado para evitar modificação de dados. A retenção de dados é um aspecto fundamental das investigações forenses, pois garante que os dados originais do dispositivo permaneçam intactos.

Após a coleta do dispositivo, os dados são extraídos. A extração de dados é um processo complexo que pode ser feito de diferentes maneiras, dependendo do tipo de dados que você procura e das capacidades do dispositivo. Por exemplo, se os dados desejados estiverem em um aplicativo específico, talvez seja necessário usar uma ferramenta de extração de dados especializada para esse aplicativo. Em outros casos, pode ser possível extrair os dados diretamente do sistema operacional do dispositivo. Depois que os dados são extraídos, eles são processados para torná-los legíveis e úteis para a investigação. O processamento de dados pode incluir a conversão de formatos de arquivo, decodificação de dados criptografados, análise de metadados e outras técnicas. O objetivo do processamento de dados é transformar os dados brutos extraídos em informações que possam ser facilmente analisadas e interpretadas.

Assim como Egohan Casey (2011) explica em seus artigos, a análise de dados é a etapa final do processo de investigação forense. Durante a análise, os dados processados são examinados para extrair informações relevantes para a investigação. A análise pode incluir a identificação de padrões nos dados, a correlação de dados de diferentes fontes, a identificação

de anomalias e outras técnicas. O objetivo da análise é fornecer informações que possam ajudar a solucionar o caso. Durante todo o processo de investigação forense, é essencial manter a integridade dos dados. Qualquer modificação dos dados pode comprometer a validade da pesquisa. Além disso, as investigações forenses devem ser conduzidas de forma ética e legal, de acordo com todas as leis e regulamentos aplicáveis.

Existem basicamente dois tipos de extração de dados em dispositivos Android: extração física e extração lógica. Segundo Alan Garbes (2024), a extração física é um método mais complexo que pode exigir conhecimentos especiais de hardware e eletrônica. Porém, tem a vantagem de poder recuperar arquivos excluídos do sistema. Por outro lado, a extração lógica utiliza a funcionalidade de software associada ao dispositivo. A extração lógica envolve a extração de dados usando o sistema operacional do dispositivo por meio de um conjunto de comandos familiares. Ambos os métodos têm vantagens e desvantagens, e a escolha entre eles depende das necessidades específicas da investigação. Existem muitas ferramentas disponíveis para facilitar a extração e análise de dados em dispositivos Android, como o software Mobiledit Forensic e a ferramenta Magnet Acquire. Essas ferramentas podem facilitar o processo de identificação, extração de dados e preservação de possíveis evidências digitais encontradas no dispositivo.

2.5 Evidência Digital

Mensagens, imagens, aplicações, arquivos e outros “artefatos” podem formar um ponto comum em múltiplos dispositivos, possibilitando identificar relacionamentos entre entidades comunicantes. De acordo com Alexsandro Menezes (2021), tais redes de relacionamento podem apontar grupos criminosos e revelar informações importantes para a resolução de casos. Cada objeto pode ser considerado prova digital e deve ser tratado adequadamente, garantindo sua integridade e atribuição ao autor. O tratamento das evidências digitais não apenas facilita a conexão dos dados com outros elementos, mas também permite a repetição dos processos de conhecimento, a confirmação dos resultados e a realização de uma segunda competência.

De acordo com Lucas Fernando (2023), a análise de evidências digitais é uma prática fundamental no campo da investigação forense digital, cujo objetivo é coletar, preservar e analisar informações eletrônicas para a resolução de crimes cibernéticos. Com o avanço da tecnologia, a quantidade de dados digitais disponíveis cresceu exponencialmente, tornando essencial o uso de técnicas especializadas para lidar com essas evidências de forma eficiente e segura.

O processo de análise de evidências digitais envolve a coleta armazenamento, análise e apresentação de dados eletrônicos de forma forense. Este processo segue diretrizes específicas para garantir a integridade e autenticidade das provas, garantindo que elas sejam admissíveis em juízo.

A criação de uma imagem forense de um dispositivo de armazenamento é um método comum de preservação de evidências porque mantém os dados originais intactos, evitando modificações acidentais ou intencionais.

2.6 Crimes Cibernéticos

Como menciona Tiago Faccini (2023), os crimes cometidos com equipamentos de informática, também chamados de crimes cibernéticos, são uma categoria de crimes que se tornou cada vez mais prevalente com os avanços da tecnologia. Estes crimes podem ser cometidos através da Internet ou de sistemas informáticos e incluem uma vasta gama de atividades ilegais.

O cibercrime pode ser definido como todos os crimes que podem ser cometidos através de computadores e assumir diferentes formas. Entre eles estão o acesso indevido a sistemas, o roubo de informações, a falsificação de documentos por meio de tecnologia, os danos a computadores e informações, a apropriação ilegal de segredos industriais ou comerciais, a cópia indevida de programas de computador, a violação de direitos autorais, a distribuição de vírus eletrônicos, etc. Além de crimes comumente cometidos na Internet, como crimes de indulto, pornografia infantil, crimes de honra e até crimes eleitorais.

A legislação brasileira foi adaptada para lidar com esses novos tipos de crimes. A Lei 14.155/2021, por exemplo, promoveu alterações no Código Penal no que diz respeito aos crimes cometidos com dispositivos eletrônicos, crimes de hacking de computador, roubo por fraude, peculato por fraude. A alteração mais importante diz respeito ao artigo 154.º-A do Código Penal introduzido na lei n.º. 12.737/2012 conhecida como Lei de Carolina Dieckmann, que criminaliza a invasão de dispositivo de informática.

O Código Penal Brasileiro é uma lei composta por um conjunto de normas sistemáticas de natureza punitiva. Seu objetivo é aplicar sanções, desencorajando a prática de crimes. O direito penal consagra as leis, caracteriza e garante os direitos dos cidadãos. Este documento tem como objetivo geral analisar todos os aspectos importantes do crime cibernético e do direito penal brasileiro no combate a esses crimes.

No entanto, apesar dos esforços envidados para combater o cibercrime, os desafios

são muitos. A natureza global da Internet e a velocidade com que a tecnologia avança muitas vezes tornam difícil para as autoridades rastrear e processar criminosos. Além disso, a falta de consenso internacional sobre o que constitui o crime cibernético e como deve ser punido complica ainda mais os esforços para combater estes crimes. Em conclusão, o cibercrime representa uma ameaça significativa na era digital. É essencial que a legislação continue a evoluir para acompanhar a tecnologia e garantir uma proteção adequada contra estes crimes. Além disso, é importante que os indivíduos sejam informados sobre os riscos associados ao uso da tecnologia e tomem medidas para proteger as suas informações pessoais e evitar serem vítimas de crimes cibernéticos.

2.7 Sistema Operacional Android

O Android é um sistema operacional móvel desenvolvido pelo Google. Foi lançado pela primeira vez em setembro de 2008 e desde então se tornou o sistema operacional mais utilizado no mundo em dispositivos móveis como smartphones e tablets.

A principal função do Android é fornecer um ambiente operacional completo para dispositivos móveis. Ele gerencia o hardware do dispositivo, como processadores, memória, tela sensível ao toque, câmera, sensores, conectividade, etc. O Android fornece uma interface de usuário intuitiva e interativa, que permite aos usuários acessar os aplicativos, serviços e funções do dispositivo.

O código do sistema operacional está disponível no Google sob uma licença de código aberto, embora a maioria dos dispositivos seja lançada com uma combinação de código aberto e software proprietário. Ele foi originalmente desenvolvido pela empresa Android, Inc., que o Google apoiou financeiramente.

2.7.1 Arquitetura do Sistema Operacional Android

Como descrito pela empresa proprietária do Android em sua documentação oficial, o sistema operacional Android é uma plataforma de código aberto baseada em Linux, projetada para uma ampla variedade de dispositivos e formatos. A arquitetura Android é composta por diversas camadas, cada uma com funções específicas, permitindo que o sistema operacional funcione de forma eficiente e segura.

O kernel Linux está no centro da arquitetura Android. O Android é baseado no

kernel Linux de código aberto, que fornece funcionalidades básicas comprovadas para o desenvolvimento do sistema operacional Android. O uso de um kernel Linux permite que o Android aproveite os principais recursos de segurança, possibilitando que os fabricantes de dispositivos desenvolvam *drivers* de *hardware* para um kernel conhecido. Acima do kernel do Linux está a camada de abstração de *hardware* (HAL). A HAL fornece interfaces padrão que expõem os recursos de hardware do dispositivo à estrutura de API Java de nível superior. O HAL consiste em vários módulos de biblioteca, cada um implementando uma interface para um tipo específico de componente de *hardware*, como uma câmera ou um módulo Bluetooth.

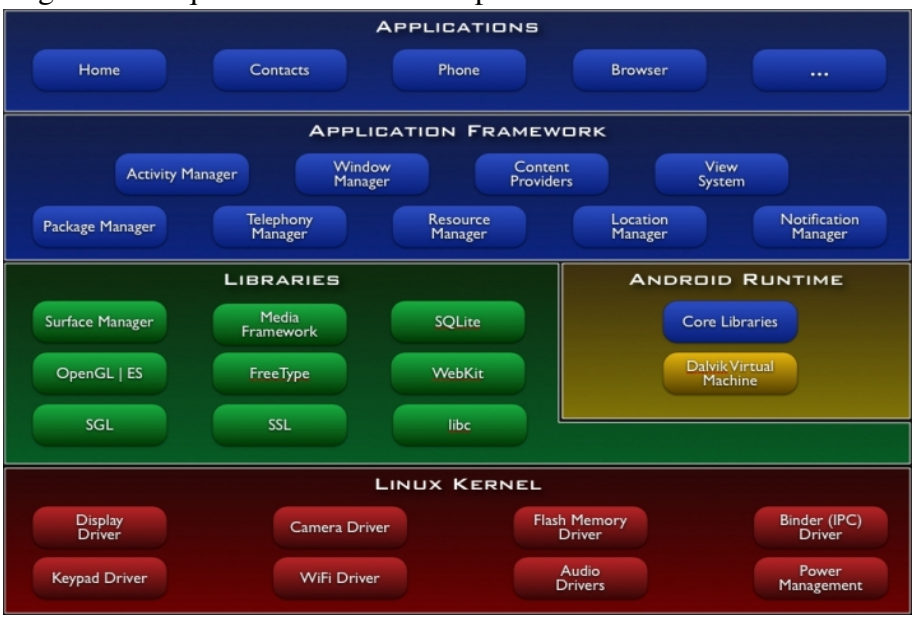
O tempo de execução do Android é outra parte essencial da sua arquitetura. Para dispositivos que executam o Android versão 5.0 (nível de API) ou superior, cada aplicativo é executado em seu próprio processo e com sua própria instância de tempo de execução do Android (ART). O ART foi projetado para executar várias máquinas virtuais em dispositivos com pouca memória, executando arquivos no formato Dalvik Executable (DEX), um formato de bytecode projetado especificamente para Android e otimizado para um consumo mínimo de memória.

Além disso, muitos componentes e serviços principais do sistema Android, como ART e HAL, são criados a partir de código nativo que requer bibliotecas nativas escritas em C e C++. A plataforma Android fornece APIs de estrutura Java para expor a funcionalidade de algumas dessas bibliotecas nativas aos aplicativos.

Finalmente, no topo da pilha de software Android, encontramos o *Application Framework* e os aplicativos. O *Application Framework* fornece os blocos de construção de alto nível necessários para criar aplicativos Android. Simplifica a reutilização de componentes e serviços do sistema, como a janela de gerenciamento do sistema, o pacote de gerenciamento e o gerenciamento de recursos. Aplicativos são programas com os quais os usuários finais interagem diretamente com aplicativos de telefone, contatos e navegadores.

Essa arquitetura em camadas permite que o Android seja flexível e adaptável a uma ampla variedade de dispositivos de hardware, mantendo a segurança e a eficiência. Além disso, permite que os desenvolvedores de aplicativos se concentrem na criação de experiências de usuário ricas sem se preocupar com a implementação de baixo nível.

Figura 2 – Arquitetura do Sistema Operacional Android



Fonte: DOLORES, 2011.

2.7.2 Funcionamento dos dispositivos Android e suas particularidades

De acordo com Milo Caspian (2024), o sistema Android consiste em um kernel baseado no kernel Linux, especialmente o branch, com suporte de longo prazo. Em janeiro de 2014, a maioria das versões do Android são modeladas a partir do kernel Linux versão 3.4 ou superior, mas a versão específica do kernel depende do dispositivo Android e do processador que ele usa. O Android usou vários kernels desde o primeiro.

O kernel Linux do Android contém mudanças arquitetônicas profundas que foram implementadas pelo Google fora do círculo típico de desenvolvimento do kernel Linux, como a inclusão de componentes como binder, ashmem, pmem, logger, wakelocks e uma saída de memória diferente, fora da memória (MOO).

Todas essas características do sistema operacional são importante de serem entendidas pelos profissionais em investigação forense, visto que as ferramentas desenvolvidas para auxiliar na perícia levam em consideração toda a estrutura do SO.

2.7.3 Tipos de dados armazenados em dispositivos Android que são relevantes para investigações forense

De acordo com Eoghan Casey (2011), a crescente adoção de dispositivos móveis equipados com sistema operacional Android trouxe desafios significativos para a análise forense.

Esses dispositivos, como *smartphones* e *tablets*, armazenam diversos dados importantes para investigações criminais e comerciais. Neste contexto, é fundamental compreender os tipos de informação que podem ser extraídas destes dispositivos:

a) metadados de comunicação: Os registros de chamadas e mensagens (SMS e MMS) contêm informações essenciais como datas, horários, duração e números de telefone envolvidos;

b) dados de contato: os contatos armazenados no seu dispositivo incluem nomes, números de telefone e endereços de e-mail. Esses dados podem ser importantes para identificar vínculos entre indivíduos;

c) histórico de navegação e atividade online: a análise do histórico de navegação revela URLs visitados, cookies e outras atividades online, fornecendo informações sobre o comportamento do usuário;

d) aplicativos e dados associados: A lista de aplicativos instalados e seus dados associados, como configurações e arquivos temporários, podem conter informações relevantes;

e) mensagens de aplicativos de mensagens instantâneas: Bate-papos em aplicativos como WhatsApp, Telegram e Facebook Messenger são fontes valiosas de evidências;

f) arquivos de mídia: fotos, vídeos e áudios armazenados no dispositivo podem conter informações importantes para investigações. Logs de eventos do sistema: Os logs de eventos, como reinicializações e desligamentos, podem ajudar a reconstruir o histórico de atividades;

g) dados de localização: histórico de localização, registros de GPS e informações sobre redes Wi-Fi e torres de celular são importantes para determinar os movimentos do usuário;

h) configurações e preferências: Os arquivos e preferências de configuração do sistema e do aplicativo podem fornecer informações sobre o uso do dispositivo;

i) backup e sincronização: O backup e a sincronização de dados com serviços em nuvem devem ser considerados;

j) memória não volátil: A memória não volátil é um tipo de armazenamento de computador que retém dados mesmo quando a energia é desligada. Ao contrário da memória volátil, que requer energia constante para manter a integridade dos dados, a memória não volátil é estável e não requer atualizações frequentes. Ele é usado para armazenar informações críticas, como sistema operacional, BIOS e outros dados essenciais. Exemplos comuns de memória não volátil incluem ROM (memória somente leitura), memória flash e discos rígidos. Essa característica torna a memória não volátil essencial para a confiabilidade e armazenamento de dados de sistemas computacionais;

k) memória volátil: A memória volátil é um tipo de memória de acesso aleatório (RAM) temporária em um computador. Ela mantém os arquivos de programas e o sistema operacional ativos quando o computador é ligado, mas não salva seu conteúdo quando o computador é desligado. Ao contrário da memória não volátil, a memória volátil requer energia constante para armazenar os dados armazenados. Resumindo, é a memória que permite acesso rápido aos dados durante a operação, mas não armazena esses dados quando o sistema está desligado.

A análise forense em dispositivos Android requer conhecimento técnico e ferramentas especializadas para extrair, interpretar e armazenar legalmente esses dados. A integridade das evidências é essencial para garantir resultados confiáveis nas investigações. A constante evolução destes dispositivos exige que os profissionais estejam atualizados e preparados para enfrentar os desafios desta área.

2.7.4 Desafios específicos na perícia de dispositivos Android

De acordo com Vinicius Oliveira (2021), a perícia em dispositivos Android apresenta desafios específicos devido à diversidade de modelos, sistemas operacionais e mecanismos de segurança. A variedade de modelos e versões de dispositivos Android exige que os peritos estejam atualizados sobre as particularidades de cada um. Além disso, cada versão do sistema operacional pode ter diferenças significativas em termos de estrutura de arquivos, configurações e aplicativos pré-instalados.

Os mecanismos de segurança utilizados nos dispositivos Android incluem senhas, PINs, padrões de desbloqueio e autenticação biométrica (como impressão digital e reconhecimento facial). Desvendar essas proteções requer conhecimento técnico e o uso de ferramentas especializadas.

A grande quantidade de aplicativos disponíveis na Play Store também complica a análise forense. Além dos aplicativos nativos, os peritos devem examinar aplicativos de terceiros, que podem conter dados relevantes para a investigação.

Outro desafio é a presença de dados em nuvem. Muitos dispositivos Android estão integrados a serviços em nuvem, como o Google Drive. Isso significa que parte dos dados pode estar armazenada remotamente, exigindo acesso a essas contas para uma análise completa.

Por fim, a fragmentação do sistema operacional Android, com diferentes versões em uso, dificulta a padronização dos procedimentos de perícia. Os peritos precisam adaptar suas técnicas a cada cenário específico.

2.8 Máquinas Virtuais

Uma máquina virtual (VM) é uma emulação de um sistema de computador que permite que um sistema operacional e aplicativos sejam executados como se fosse um computador físico. Segundo Ronaldo Gogoni (2019), a virtualização é a tecnologia subjacente que permite criar várias máquinas virtuais em uma única peça de hardware físico, chamada *host*. Cada VM é executada de forma independente, com sistema operacional, aplicativos e recursos alocados próprios, como processamento, memória e armazenamento.

O funcionamento de uma máquina virtual depende de um software denominado hipervisor, que gerencia e aloca os recursos físicos do host para as VMs. Existem dois tipos principais de hipervisores: Tipo 1, que roda diretamente em hardware físico, e Tipo 2, que roda em um sistema operacional host. O hipervisor tipo 1 é comumente usado em ambientes de servidor, enquanto o tipo 2 é mais comum em desktops e notebooks.

A virtualização oferece muitos benefícios, incluindo eficiência de recursos, flexibilidade e portabilidade. Com a virtualização é possível rodar diversos sistemas operacionais em um único dispositivo, maximizando o uso dos recursos disponíveis. Além disso, as máquinas virtuais podem ser facilmente movidas entre diferentes hosts, facilitando a manutenção e a escalabilidade dos sistemas.

Quanto às aplicações práticas, as máquinas virtuais são amplamente utilizadas em ambientes de desenvolvimento e testes, onde é necessário simular diferentes sistemas operacionais e configurações. Também são essenciais em infraestruturas em nuvem, que permitem a criação de ambientes isolados e seguros para execução de aplicações e serviços. A capacidade de isolar e gerenciar recursos com eficiência torna as máquinas virtuais uma ferramenta indispensável na computação moderna.

2.9 Aspectos técnicos e legais

Segundo Medeiros (2020), os aspectos técnicos da perícia forense computacional podem ser divididos em três categorias. A primeira é a coleta de evidências digitais, onde envolve a coleta de evidências em dispositivos eletrônicos, como computadores, dispositivos móveis e servidores. Neste caso, os peritos utilizam ferramentas especializadas para preservar e extrair dados relevantes, garantindo a integridade das provas.

A segunda é a análise de dados, onde os dados coletados são analisados minucio-

samente. Isso inclui examinar arquivos, registros, metadados e comunicações. Técnicas como recuperação de arquivos apagados, análise de tráfego de rede e identificação de padrões são aplicadas.

Por último, a recuperação de informações, no qual o objetivo é recuperar informações relevantes, mesmo que tenham sido excluídas ou ocultadas, o que envolve a busca por arquivos, históricos de navegação, mensagens e outros vestígios digitais.

De acordo com Eleutrio e Machado (2011), os aspectos legais podem ser igualmente separados em três distintas categorias. A primeira é a validade legal, visto que a perícia forense deve seguir procedimentos rigorosos para garantir a validade das evidências em tribunal. Os peritos devem cumprir as normas legais e éticas, evitando contaminação das provas.

A segunda é a cadeira de custódia, que documenta o controle das evidências desde a coleta até a apresentação em juízo. É fundamental para evitar questionamentos sobre a autenticidade e integridade das provas.

Por último, o sigilo e privacidade, essencial pois os peritos devem respeitar a privacidade dos envolvidos e seguir as leis de proteção de dados. Informações sensíveis devem ser tratadas com confidencialidade.

2.9.1 Lei Carolina Dieckman

A Lei 12.737/2012, conhecida como Lei Carolina Dieckmann, foi sancionada em 30 de novembro de 2012 pela presidente Dilma Rousseff. Essa legislação promoveu alterações no Código Penal brasileiro, classificando especificamente os chamados “crimes informáticos” ou crimes cibernéticos. Um dos principais aspectos que a lei trata é a invasão de dispositivos computacionais alheios, como celulares, notebooks e tablets. O texto legal considera crime a invasão de tais dispositivos, estejam ou não conectados a uma rede de computadores, violando injustamente um mecanismo de segurança. O objetivo é obter, modificar ou destruir dados ou informações sem a autorização expressa ou implícita do proprietário do dispositivo. Além disso, a lei também prevê sanções em caso de instalação de vulnerabilidades para obtenção de vantagem ilegal.

Segundo Tiago Fachini (2023), a Lei Carolina Dieckmann tem desempenhado papel fundamental na proteção de informações e dados individuais no ambiente digital brasileiro. Antes desta lei, o acesso a dispositivos privados não era considerado crime, sendo considerado apenas um ato preparatório. Com a sua passagem, a prática de hacking passou a ser tratada como

crime, contribuindo assim para a segurança e privacidade online. Além disso, esta lei serviu de base para o desenvolvimento de outras legislações relevantes, como a Lei Geral de Proteção de Dados (LGPD) e o Marco Civil da Internet. A lei de Carolina Dieckmann marcou, portanto, um passo importante na regulamentação do crime cibernético no Brasil.

2.9.2 Lei de Acesso a Informação

A Lei de Acesso à Informação (LAI), regulamentada pela lei n. 12.527/2011, visa garantir o direito constitucional de solicitar e receber informações de órgãos e entidades públicas. Esta norma, em vigor desde 16 de maio de 2012, criou mecanismos que permitem a qualquer pessoa, singular ou coletiva, aceder à informação pública sem ter de indicar um motivo específico. Por meio da LAI é possível pesquisar os dados produzidos ou em poder dos órgãos públicos, promovendo assim a transparência e o controle social da administração pública.

É importante notar que esta legislação estabelece procedimentos específicos para acesso à informação pública. Os órgãos e entidades públicas devem disponibilizar canais de comunicação, como o Serviço de Informação ao Cidadão (CIS), para receber pedidos de acesso. Além disso, a LAI define os prazos de resposta às solicitações e as possibilidades de recurso em caso de negação de acesso. Esta abordagem visa garantir a efetividade do direito à informação e a participação ativa da sociedade no monitoramento e controle das atividades governamentais.

2.9.3 Marco Civil da Internet

O Marco Civil da Internet, oficialmente conhecido como Lei nº. 12.965/2014, define os princípios, garantias, direitos e deveres para o uso da Internet no Brasil. Seu anúncio em 23 de abril de 2014 marcou um passo importante na regulamentação das atividades online no país. Fundado no respeito à liberdade de expressão e aos direitos humanos, o Marco Civil visa conciliar a escala global da rede com a proteção da privacidade, a preservação da neutralidade da rede e a promoção do acesso à informação e da participação dos cidadãos. Além disso, a lei estabelece diretrizes para a atuação de diversos entes federais nesta matéria, garantindo que suas disposições também se apliquem às empresas estrangeiras que prestam serviços ao público brasileiro.

Em sua finalidade, o Marco Civil da Internet visa promover o direito universal de acesso à Internet, bem como o acesso à informação e ao conhecimento. Além disso, busca incentivar a inovação e a adoção de padrões tecnológicos abertos, que possibilitem a comunicação e a

interoperabilidade entre aplicações e bancos de dados. A lei define também as responsabilidades dos diversos agentes que operam na rede, equilibrando a liberdade dos modelos de negócio com os princípios fundamentais estabelecidos. Assim, o Marco Civil da Internet desempenha um papel crucial na construção de uma Internet mais inclusiva, transparente e responsável no Brasil.

2.9.4 Lei das Perícias Oficiais

A Lei nº 12.030/2009, também conhecida como Lei das Perícias Oficiais, é um marco regulatório que estabelece as diretrizes e os procedimentos para a realização de perícias oficiais no âmbito criminal no Brasil. Seu objetivo é garantir a qualidade, a imparcialidade e a eficiência dos trabalhos periciais, contribuindo para a justiça e a segurança jurídica.

Os peritos que atuam sob esta lei têm autonomia técnica, científica e funcional. São profissionais especializados em áreas como medicina legal, medicina legal, odontologia legal e outras disciplinas forenses. Para ingressar na carreira é necessário ser aprovado em concurso público e ter formação acadêmica específica. Esta exigência visa assegurar a competência e a formação destes profissionais, que desempenham um papel crucial na investigação de crimes e na produção de prova pericial.

A Lei das Perícias Oficiais reconhece três categorias principais: os peritos criminais, os médicos-legistas e os odontolegistas. Cada um deles possui um campo de atuação específico. Por exemplo, os especialistas forenses são responsáveis por analisar pistas nas cenas do crime, como impressões digitais, amostras de sangue e fibras. Os médicos legistas realizam autópsias e avaliam as causas da morte. Já os dentistas forenses investigam questões relacionadas à odontologia legal, como a identificação de vítimas por meio de prontuários odontológicos.

Além disso, a Lei das Perícias Oficiais prevê um regime de trabalho específico para esses profissionais. Levando em consideração as peculiaridades de cada ente federado (Estados e Distrito Federal), a legislação define as regras quanto à jornada de trabalho, remuneração, infraestrutura e equipamentos necessários à realização dos exames. Esta abordagem visa garantir condições adequadas à realização de atividades especializadas, promovendo a excelência nos serviços oferecidos à sociedade.

3 FERRAMENTAS DE PERÍCIA FORENSE EM ANDROID

Segundo Eoghan (2014), as ferramentas forenses de perícia digital desempenham um papel crucial na investigação de crimes cibernéticos e na coleta de evidências digitais. Estas ferramentas são utilizadas para analisar dispositivos eletrônicos, como computadores e dispositivos móveis, para identificar atividades ilegais e recuperar informações relevantes. Eles permitem que especialistas examinem dados armazenados em discos rígidos, memória RAM, registros de rede e outros componentes digitais para reconstruir eventos, identificar os responsáveis e fornecer evidências sólidas para processos criminais.

A perícia digital envolve técnicas rigorosas, como recuperação de arquivos excluídos, identificação de *malware*, análise de logs de acesso e verificação da integridade dos dados. Estas ferramentas são essenciais para garantir a confiabilidade e validade das provas recolhidas durante investigações criminais e processos judiciais.

3.1 Critério de seleção das ferramentas a serem comparadas

Para realizar uma comparação mais justa entre as ferramentas, é necessário estabelecer critérios de avaliação objetivos, bem como dividi-las baseado em suas funcionalidades. Para os fins deste trabalho, as ferramentas serão dividas em três distintas categorias, sendo estas: ferramenta de análise para memória volátil, ferramenta para análise de memória não-volátil e ferramenta de múltiplas funcionalidades. Esta última possui diferentes abordagens de análise que serão especificadas na descrição de cada uma.

Os critérios de avaliação serão os mesmos para as três categorias, sendo estes: êxito na extração dos dados, especificação das dependências necessárias para execução da ferramenta, execução sem erros de síntaxe, atualização submetida nos últimos dois anos.

3.2 Memória Não-Volátil

Ferramentas de memória não-volátil.

3.2.1 AFLogical OSE

Segundo Carlos Cilleruelo (2024), o AFLogical OSE (*Open Source Edition*) é uma ferramenta forense digital de código aberto projetada para extrair dados de dispositivos móveis

Android. Desenvolvida pela empresa russa *Oxygen Forensics*, esta ferramenta é baseada no seu software comercial *Oxygen Forensic Detective*. AFLogical OSE permite que os examinadores extraiam registros de chamadas, contatos, mensagens MMS, partes MMS e mensagens SMS de dispositivos *Android*. Embora a versão completa do AFLlogical seja gratuita apenas para autoridades policiais, a edição de código aberto oferece recursos semelhantes para entusiastas do *Android* e especialistas em análise forense digital.

A edição de código aberto foi lançada para uso por não-policiais, entusiastas do Android e gurus forenses. Ele permite que um examinador remova chamadas *CallLog*, telefones de contato, mensagens MMS, MMSParts e mensagens SMS de dispositivos *Android*. O *software* AFlogical completo está disponível gratuitamente para o pessoal responsável pela aplicação da lei.

3.2.2 dd

O dd é uma ferramenta de linha de comando amplamente usada em sistemas operacionais *Unix* e *Unix-like* para converter e copiar arquivos. Seu principal objetivo é criar imagens de disco, principalmente para fins forenses. Ao criar uma imagem de um dispositivo de armazenamento, como um disco rígido ou uma unidade flash USB, o dd copia setores inteiros, incluindo dados, partições e estruturas do sistema de arquivos. Isso permite especificar o tamanho do bloco, gerenciar erros de leitura e gravar em um arquivo de saída, que pode ser analisado para investigação.

Adicionalmente, existem variantes dd otimizadas para tarefas forenses, como dc3dd e dcfldd. Essas versões aprimoradas fornecem recursos adicionais, como *hashing* em tempo real, verificação de integridade, relatórios detalhados de *status* e compartilhamento de arquivos de saída. O dcfldd, em particular, foi projetado especificamente para aplicativos de segurança e forenses, oferecendo recursos como verificação de imagem, limpeza flexível de disco e saída simultânea para vários arquivos.

3.2.3 Foremost

O Foremost é um programa forense de recuperação de dados para sistemas Linux. Ele recupera arquivos com base em seus cabeçalhos, rodapés e estruturas de dados por meio de um processo chamado file carving. Embora tenha sido originalmente desenvolvido para uso pelas autoridades policiais, o código-fonte do programa está disponível gratuitamente e pode ser

usado como uma ferramenta geral de recuperação de dados. Ele foi projetado principalmente para ignorar o tipo de sistema de arquivos subjacente e ler e copiar diretamente partes do disco para a memória do computador. Ele procura tipos de cabeçalho de arquivo definidos em seu arquivo de configuração e, quando encontra uma correspondência, grava o cabeçalho e os dados subsequentes em um arquivo, parando quando encontra um tamanho de página final ou atinge o limite de tamanho do arquivo.

O Foremost não possui interface gráfica e é usada apenas na linha de comando. É capaz de recuperar tipos de arquivos específicos como jpg, gif, png, pdf, doc, zip e outros, podendo ser aplicado em imagens de disco ou diretamente em discos físicos com sistemas de arquivos ext3, NTFS ou FAT.

3.3 Memória Volátil

Ferramentas de memória volátil.

3.3.1 LiME - Linux Memory Extractor

Linux Memory Extractor (LiME) é um módulo de kernel carregável (LKM) projetado para extrair memória não volátil em sistemas Linux e dispositivos baseados em Linux, como Android. Sua principal função é possibilitar a captura total da RAM, mantendo a integridade forense no processo. O LiME minimiza a interação entre o espaço do usuário e os processos do kernel, resultando em uma captura mais confiável e robusta em comparação com outras ferramentas de captura de memória.

Como explica Adam Outler (2012), o LiME é carregado como um módulo do kernel, permitindo que ele seja executado no espaço do kernel. Isto dá acesso privilegiado à memória do sistema, incluindo áreas protegidas. Entretanto, o LiME foi projetado para minimizar a interação com outros processos do kernel, reduzindo o risco de corrupção ou manipulação. Durante a aquisição, o LiME evita alterações desnecessárias de memória. Não altera o conteúdo da RAM, mantendo assim a integridade dos dados. Além disso, o LiME minimiza a atividade do kernel, evitando que outros processos interfiram na captura. O LiME usa mecanismos de acesso controlado para ler a memória. Ele não ignora as proteções de acesso do kernel, mas opera dentro dessas restrições. Isto garante que a aquisição seja cientificamente sólida e não comprometa a integridade das evidências.

3.3.2 AMExtractor

AMExtractor é uma ferramenta de análise de memória para dispositivos Android. Ao contrário de outras ferramentas baseadas em módulos do kernel (como o popular LiME), o AMExtractor não requer código-fonte do kernel ou suporte a módulos. Ele opera no espaço do kernel usando o dispositivo especial /dev/kmem. AMExtractor foi testado em vários dispositivos Android, incluindo Galaxy Nexus, Nexus 4, Nexus 5 e Samsung Galaxy S4.

AMExtractor requer acesso ROOT no dispositivo Android para funcionar. Isso significa que o usuário deve ter privilégios de superusuário, o que pode ser um obstáculo para alguns usuários ou em dispositivos onde o ROOT não é possível. Embora o AMExtractor tenha sido testado em muitos dispositivos Android, a compatibilidade pode variar. Alguns dispositivos podem não ser suportados, resultando em falhas ou resultados incompletos na recuperação de memória.

Além disso, é importante observar que AMExtractor não é uma ferramenta completa de análise de memória. Ele se concentra na extração de memória física, mas não oferece funções avançadas de análise ou visualização de dados. Portanto, os analistas devem complementar a sua utilização com outras ferramentas e técnicas.

3.3.3 Volatility Framework

O Volatility Framework é uma coleção de ferramentas totalmente de código aberto, implementadas em Python sob a Licença Pública Geral GNU, para extrair objetos digitais de amostras de memória não volátil (RAM). As técnicas de extração são realizadas independentemente do sistema estudado, mas proporcionam visibilidade do estado de execução do sistema. A estrutura visa apresentar as técnicas e complexidades associadas à extração de artefatos digitais de amostras de memória não volátil e fornecer uma plataforma para pesquisas futuras nesta área interessante.

O Volatility suporta várias imagens de memória, incluindo Windows (XP, Vista, 7, 8, 10 e servidores), Linux (vários kernels) e Mac OSX. É amplamente utilizado em investigações de *malware*, resposta a incidentes e análise forense digital, tornando-se uma ferramenta essencial para analistas e pesquisadores interessados em análise de memória volátil.

3.4 Ferramentas Multifuncionalidades

Ferramentas que realizam múltiplas funções.

3.4.1 Sistema IPED

O Sistema IPED é uma ferramenta desenvolvida pela Polícia Federal brasileira para indexação e processamento de evidências digitais. Lançado em 2019 como um projeto de código aberto, o IPED foi originalmente utilizado internamente a partir de 2012. Seu foco é a velocidade de processamento, o que permite pesquisas eficazes e eficientes de palavras-chave em grandes volumes de dados. A indexação cria um catálogo de palavras identificadas, acelerando as respostas da pesquisa.

O Sistema IPED possui vários módulos essenciais para análise de evidências digitais. O módulo de indexação cria um catálogo de palavras-chave a partir dos dados coletados, o que permite pesquisas eficientes. O IPED processa dados brutos, extrai metadados e gera relatórios detalhados sobre os artefatos encontrados. Este módulo analisa imagens e vídeos para obter informações relevantes. O IPED pode rastrear atividades nas redes sociais, identificando perfis e conexões. Analisa e-mails, anexos e metadados. O IPED verifica os arquivos em busca de conteúdo relevante.

3.4.2 Avilla Forensics

Avilla Forensics é uma ferramenta gratuita de análise forense móvel, lançada em fevereiro de 2021, projetada para ajudar os investigadores a recuperar informações e evidências de dispositivos móveis. Desenvolvida pelo Delegado de Polícia do Estado de São Paulo, Daniel Avilla, essa ferramenta permite a extração de dados lógicos e a conversão de cópias de segurança em formatos compatíveis com análises forenses detalhadas, como IPED Physical Analyzer ou Cellebrite A versão 3.7 do Avilla Forensics introduziu melhorias significativas e novos recursos para extração e análise de dados móveis, incluindo um sistema de integridade que gera logs criptografados AES-256 (.avilla) com hashes de arquivos armazenados. Além disso, o arquivo .avilla possui assinatura HMAC, que cria uma segunda camada de proteção para a integridade do arquivo. A ferramenta também oferece interação com dispositivos móveis por meio da interface Android Debug Bridge (ADB), tornando-se uma opção versátil para comunicação com dispositivos. O módulo APK Downgrade é o principal recurso, que permite a coleta de dados de

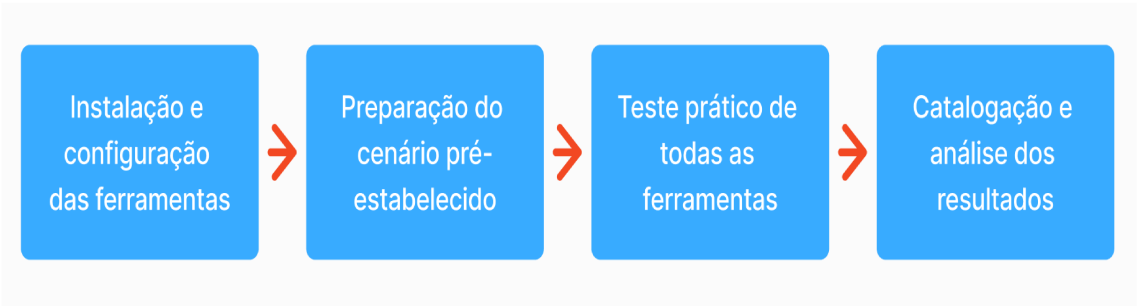
mais de 15 aplicações, tornando o Avilla Forensics indispensável para investigações forenses.

Avilla Forensics enfrenta desafios comuns no campo da análise forense móvel. A crescente adoção da criptografia em dispositivos móveis dificulta a extração de dados. Avilla Forensics deve gerenciar senhas, chaves de criptografia e proteções de privacidade para acessar informações relevantes. O ecossistema móvel é diversificado, com diferentes fabricantes, modelos e sistemas operacionais (iOS, Android, etc.). Avilla Forensics deve ser compatível com uma ampla gama de dispositivos e versões de sistemas operacionais. Quando novos dispositivos e versões de sistemas operacionais são lançados, o Avilla Forensics deve ser atualizado para garantir eficácia contínua. Isto requer pesquisa e desenvolvimento constantes de funções compatíveis.

4 METODOLOGIA

O propósito deste estudo é analisar as ferramentas *open-source* selecionadas e entender como as mesmas funcionam dentro de um cenário determinado nesta seção.

Figura 3 – Fluxo de Trabalho



Fonte: Criado pelo autor.

4.1 Instalação e configuração das ferramentas

A primeira etapa do trabalho consiste em instalar e configurar cada uma das ferramenatas. Essa etapa foi realizada seguindo os passos do que é apresentado na documentação de cada ferramenta. As instruções específicas do que deve ser feito varia de acordo com a ferramenta.

4.2 Preparação do cenário pré-estabelecido

O aparelho com Android utilizado como alvo deste estudo foi um Samsung A04e, dispositivo lançado no ano de 2023, que utiliza atualmente o sistema operacional Android 14 e foi disponibilizado para análise forense livre por um voluntário. Para preservar a privacidade do proprietário do dispositivo, todos os dados extraídos aqui mostrados terão informações censuradas.

O aparelho não tem acesso de root (acesso que permite acesso completo à raiz do Android e o controle do sistema). Apesar de ser previsto que determinadas ferramentas funcionaram melhor com o acesso de root, atualmente é necessário formatar o dispositivo de volta às configurações de fábrica para conceder o acesso de root, excluindo assim os dados que seriam extraídos. É também obrigatório que o dispositivo esteja com a função "Depuração USB"ativada para funcionar devidamente, sendo assim, a mesma está ativa neste cenário utilizado.

Os testes seriam inicialmente realizados em uma máquina de virtualização com o sistema "Santoku Linux". Contudo, devido a contratempos encontrados melhores descritos no

Apêndice A deste trabalho, a divisão ocorreu da seguinte forma: AFLogical OSE foi testado no Santoku Linux em uma máquina virtual; dd foi testado no Kali Linux que estava rodando diretamente em um dispositivo USB Flash; todas as outras ferramentas foram testadas no sistema operacional Kali Linux, sendo executado diretamente de um dispositivo Solid State Drive (SSD), com exceção do Sistema IPED e Avilla Forensics, dos quais foram testados no sistema operacional Windows 11, ideal para ambas as ferramentas.

4.3 Teste prático de todas as ferramentas e catalogação e análise dos resultados

Por fim, todos os testes foram realizados, levando em consideração a necessidade de corrigir certos erros, sendo assim, houve a tentativa de ajustar as ferramentas à medida das necessidades. Ao fim dos testes, todos os resultados foram devidamente catalogados dentro de uma tabela da qual contém todos os critérios posteriormente apresentados neste estudo, verificando se as ferramentas cumprem ou não os critérios determinados.

5 RESULTADOS

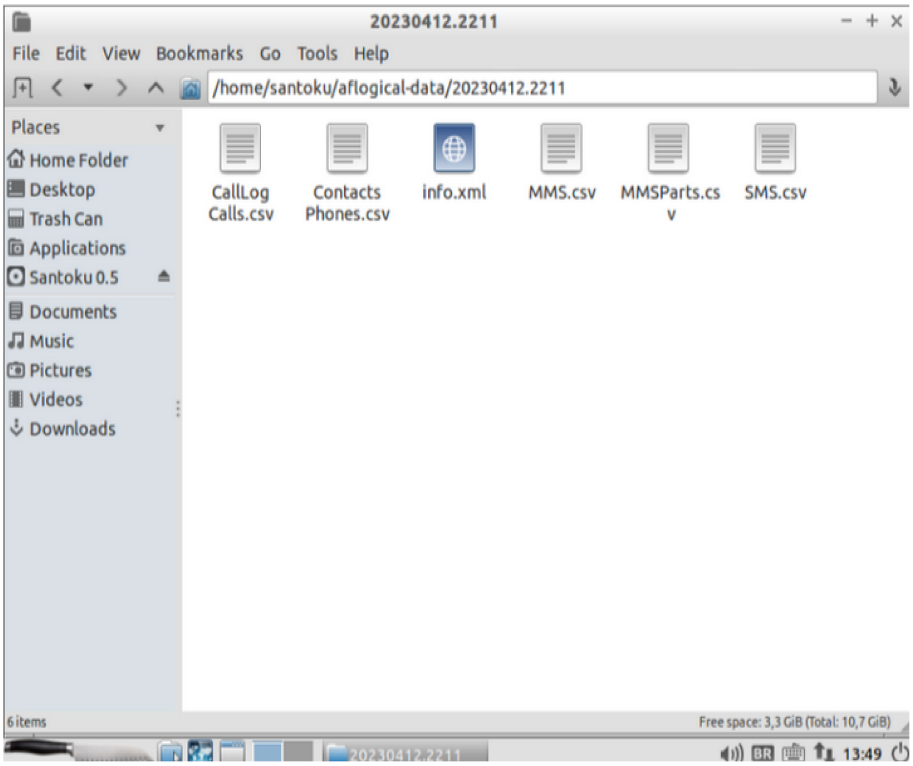
A seção de resultados deste trabalho demonstra o que foi obtido ao fim de cada teste com as ferramentas, bem como uma Tabela que foi preenchida baseado nos critérios estabelecidos no capítulo 3.

5.1 Análise das Ferramentas de Memória Não-Volátil

Análise realizada a partir dos resultados das ferramentas de memória não-volátil.

5.1.1 AFLogical OSE

Figura 4 – Resultados obtidos da extração



Fonte: Criado pelo autor.

Ao analisar os 5 arquivos que foram gerados, pode-se concluir que o resultado foi satisfatório visto que as informações desejadas foram devidamente extraídas e separadas de forma organizada para uma análise facilitada.

5.1.2 dd

Ao executar o comando, o resultado obtido não foi o esperado. Como demonstrado na Figura abaixo, o dd não pôde acessar o dispositivo alvo por ter tido a permissão negada. Considerando a mensagem exibida, pode-se presumir que o sistema de segurança do dispositivo com Android bloqueou o acesso do dd, que não possui recursos para contornar este bloqueio, impedindo assim de que a extração fosse realizada.

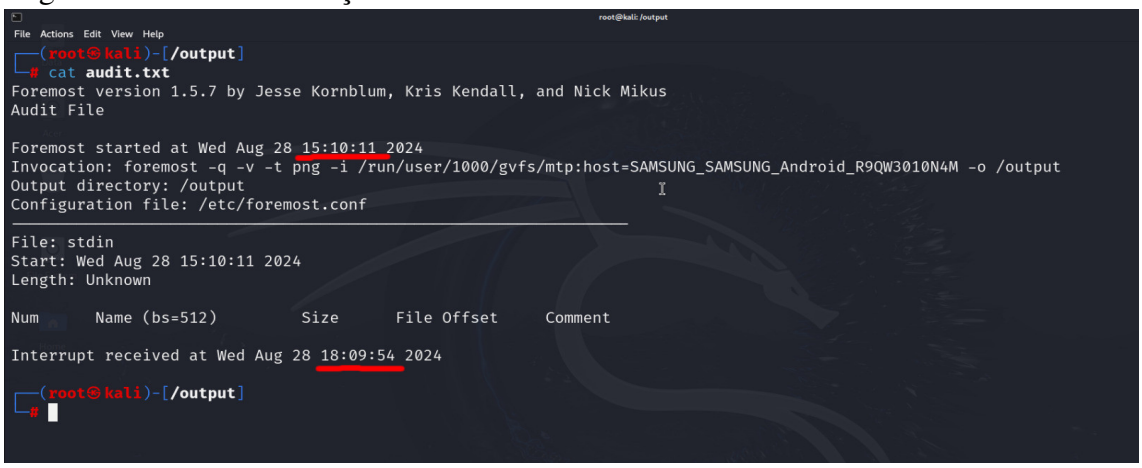
Figura 5 – Execução da ferramenta dd

Fonte: Criado pelo autor.

5.1.3 Foremost

Mesmo que a execução do programa não tenha apresentado nenhum progresso nos primeiros minutos, a ferramenta prosseguiu em execução por 3 horas inteiras, como pode-se ver na Figura abaixo, onde o processo começou às 15h10 e foi manualmente cancelado apenas às 18h. O arquivo "audit.txt"que está sendo visualizado é gerado para mostrar o log do que foi extraído.

Figura 6 – Primeira execução da ferramenta Foremost



Fonte: Criador pelo autor.

Visto que a ferramenta não conseguiu extrair nenhum arquivo, um teste similar foi

realizado, mas dessa vez alterado o tipo de arquivo alvo de "png" para "joint photographic experts group" (jpeg) para checar se os resultados seriam diferentes.

Figura 7 – Tentativa de extração de arquivos da extensão jpeg

```
┌──(root㉿kali)-[/output]
└─# cat audit.txt
Foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Mikus
Audit File

Foremost started at Thu Aug 29 15:18:00 2024
Invocation: foremost -q -v -t jpeg -i /run/user/1000/gvfs/mtp:host=SAMSUNG_SAMSUNG_An
droid_R9QW3010N4M/ -o /output
Output directory: /output
Configuration file: /etc/foremost.conf
------------------------------------------------------------------
File: stdin
Start: Thu Aug 29 15:18:00 2024
Length: Unknown

Num      Name (bs=512)         Size      File Offset     Comment

Interrupt received at Thu Aug 29 15:26:13 2024
```

Fonte: Criador pelo autor.

Como é possível ver na Figura acima, o resultado foi o mesmo, ainda não conseguindo realizar a devida extração. Neste momento, mais uma ideia de cenário para teste surgiu, então foi realizado um novo teste, desta vez digitando um diretório inexistente. Repare que na Figura abaixo, o diretório alvo não possui a última letra, representando assim um diretório alvo que não existe.

Figura 8 – Tentiva de extração a partir de um diretório inexistente

```
┌──(root㉿kali)-[/home/gabriel]
└─# cat /output/audit.txt
Foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Mikus
Audit File

Foremost started at Thu Aug 29 15:29:27 2024
Invocation: foremost -q -v -t jpeg -i /run/user/1000/gvfs/mtp:host=SAMSUNG_SAMSUNG_An
droid_R9QW3010N4 -o /output
Output directory: /output
Configuration file: /etc/foremost.conf
------------------------------------------------------------------
File: stdin
Start: Thu Aug 29 15:29:27 2024
Length: Unknown

Num      Name (bs=512)         Size      File Offset     Comment

Interrupt received at Thu Aug 29 15:33:50 2024
```

Fonte: Criador pelo autor.

Apesar de estar tentando realizar a extração de um diretório que não existe, a ferramenta Foremost se mantém na mesma tela, não relatando nenhum erro ou ao menos parando a execução, dificultando ao perito o entendimento do que está acontecendo no processo de extração.

Como último caso de teste, foi alterado o diretório do dispositivo alvo, de modo apontar diretamente para uma pasta da qual tinha-se, com certeza, arquivos da extensão jpeg. O diretório específico tomado como alvo está censurado na Figura 9 para preservar a privacidade do detentor do objeto de estudo.

Figura 9 – Tentativa de extração com diretório objetivo

```
┌──(root㉿kali)-[/home/gabriel]
└─# cat /output/audit.txt
Foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Mikus
Audit File

Foremost started at Thu Aug 29 16:02:58 2024
Invocation: foremost -q -v -t jpeg -i /run/user/1000/gvfs/mtp:host=SAMSUNG_SAMSUNG_An
droid_R9QW3010N4M/Armazenamento_interno/[redacted] -o /output
Output directory: /output
Configuration file: /etc/foremost.conf
-------------------------------------------------------------
File: stdin
Start: Thu Aug 29 16:02:58 2024
Length: Unknown

Num      Name (bs=512)         Size      File Offset     Comment

Interrupt received at Thu Aug 29 16:05:17 2024
```

Fonte: Criado pelo autor.

Mesmo com esta última alternativa, os resultados obtidos não foram satisfatórios, concluindo-se não ser possível realizar a extração.

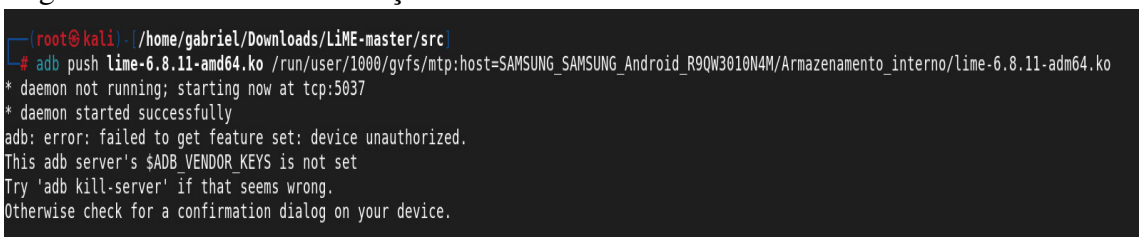
5.2 Análise das Ferramentas de Memória Volátil

Análise realizada a partir dos resultados das ferramentas de memória volátil.

5.2.1 LiME - Linux Memory Extractor

Ao executar o comando, o LiME apresentou um erro, como demonstrado na Figura abaixo. Neste caso, a ferramenta não conseguiu acesso aos arquivos do dispositivo por falta de autorização, visto que não pôde burlar a segurança do dispositivo alvo. Após tentativas frustradas de corrigir o erro exibido ao executar o comando exibido na mensagem de erro, foi constatado que o LiMe não obteve êxito na extração.

Figura 10 – Tentativa de extração com LiME

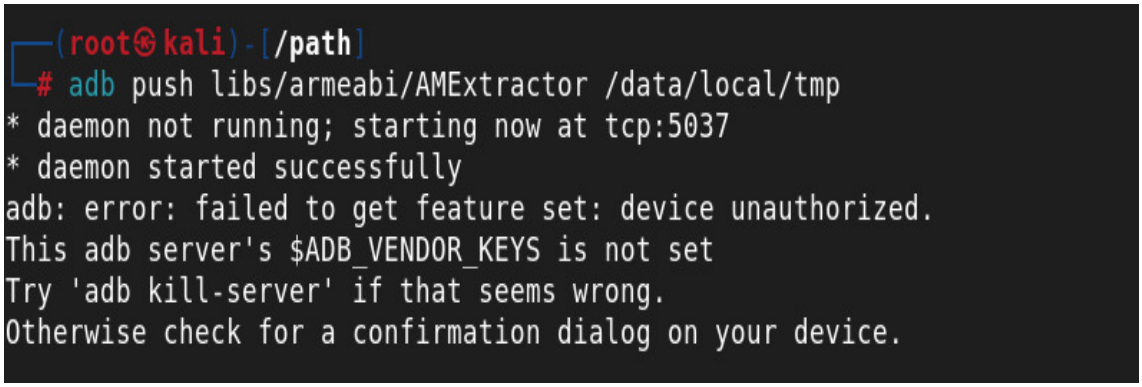


Fonte: Criado pelo autor.

5.2.2 AMExtractor

O processo do "daemon", essencial para a execução da ferramenta, inicializa normalmente, mas um erro de dispositivo não autorizado é acusado, impedindo a extração de acontecer, como demonstrado na Figura abaixo. Após tentativas frustradas de resolver o problema acusado, concluiu-se que a ferramenta falhou em extrair os dados do dispositivo alvo.

Figura 11 – Tentativa de execução da ferramenta AMExtractor

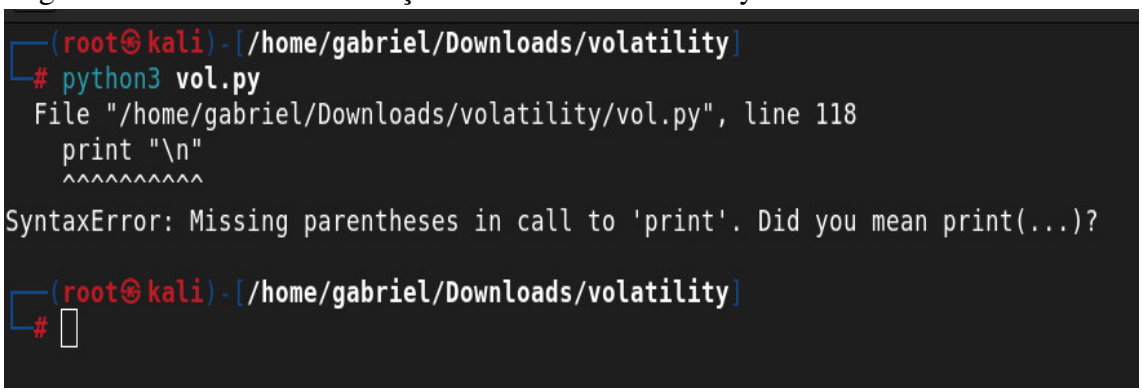


Fonte: Criado pelo autor.

5.2.3 Volatility Framework

O procedimento de instalação é simples, contudo, a ferramenta apresentou um erro muito fora do esperado até então. Ao executar o arquivo principal da ferramenta, "vol.py", é relatado um erro na síntaxe do programa, como é possível visualizar na Figura abaixo.

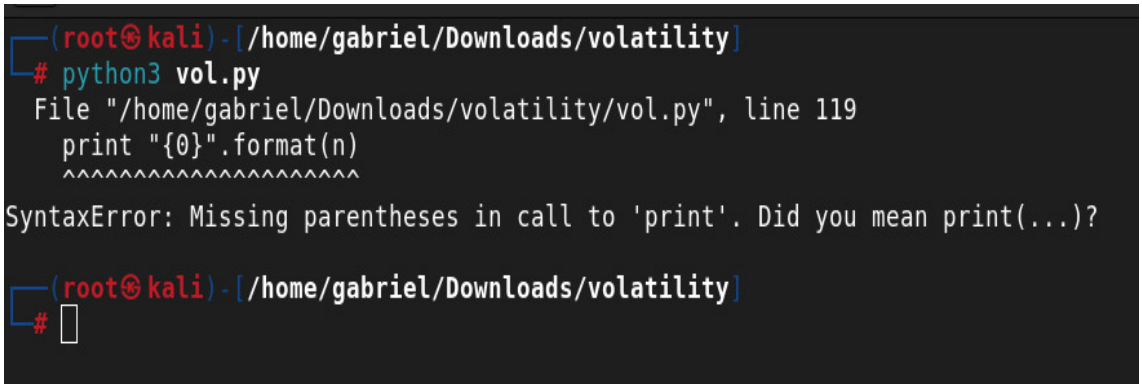
Figura 12 – Tentativa de execução da ferramenta Volatility



Fonte: Criado pelo autor.

Como aparenta ser um erro simples de inserir parênteses no comando "print", é possível utilizar um programa de edição de texto para consertar o erro demonstrado na execução da ferramenta. Apesar disso, a ferramenta começa a apresentar um novo problema de síntaxe, semelhante ao apresentado anteriormente, como é possível ver na Figura seguinte.

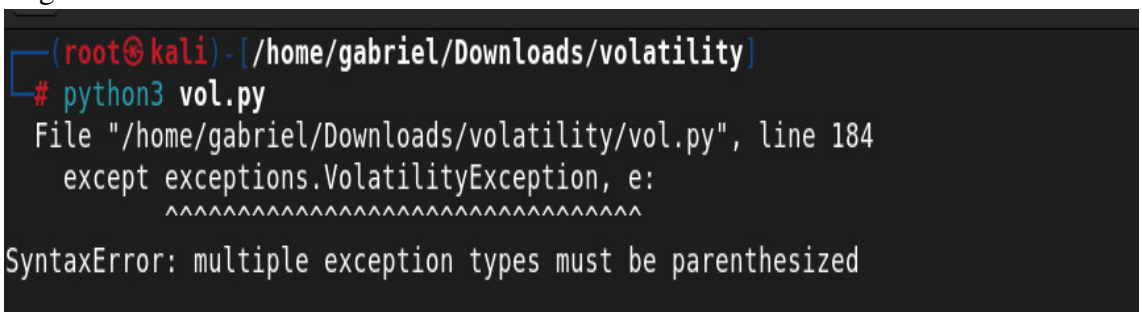
Figura 13 – Erro de síntaxe da ferramenta



Fonte: Criador pelo autor.

O código ainda apresentava o mesmo problema de síntaxe em alguns trechos, que foram manualmente corrigidos ao longo das tentativas. Após corrigir todos os erros em comandos "print", experimentou-se uma nova execução que apresentou um novo erro de síntaxe, como demonstrado na Figura abaixo.

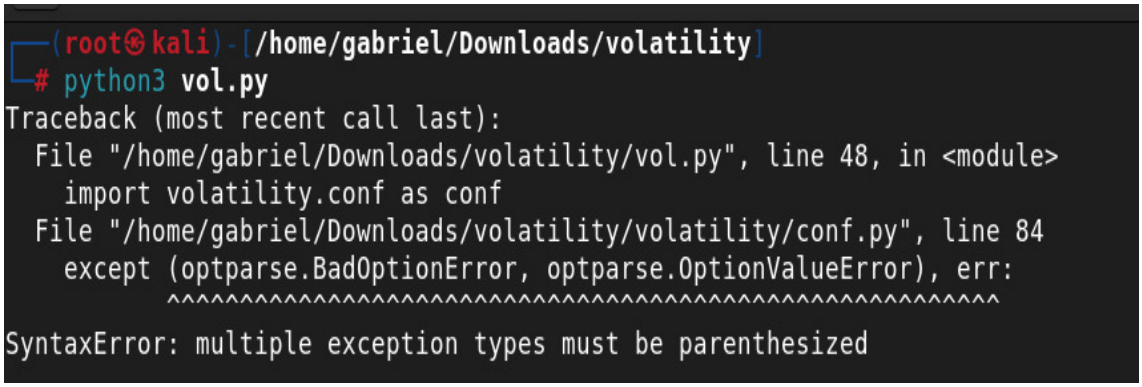
Figura 14 – Novo erro de síntaxe da ferramenta



Fonte: Criador pelo autor.

Este novo erro apresentado ainda parecia simples e ocorria apenas cerca de quatro vezes ao longo do código. Sendo assim, ainda foi possível consertar estes erros apenas incluindo parênteses nos argumentos do comando "exceptions". Após uma nova tentativa de executar o ferramenta e verificar se estava funcionando adequedamente, um novo erro foi apresentado, desta vez um erro de síntaxe em outro arquivo que é utilizado em uma *call* (chamada) do arquivo feita pelo arquivo "vol.py", como é possível visualizar na Figura abaixo.

Figura 15 – Erro de síntaxe em outro arquivo chamado pelo código principal



Fonte: Criado pelo autor.

Com este último erro, foi definido que a ferramenta não estava em condições de funcionamento adequado. A causa exata dos erros de síntaxe é desconhecida, apesar de ser possível presumir, dadas as evidências, que o Volatility esteja lidando com um problema de versionamento, onde a atual versão da ferramenta está em conflito com atualizações das suas dependências, como a linguagem "python".

5.3 Análise das Ferramentas Multifuncionalidades

Análise realizada a partir dos resultados das ferramentas de múltiplas funções.

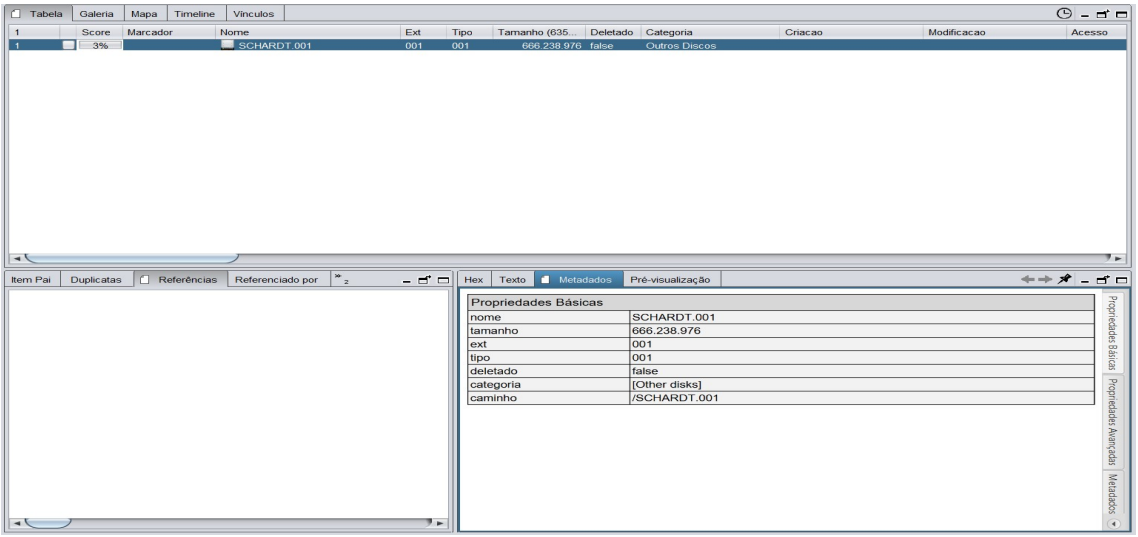
5.3.1 Sistema IPED

Figura 16 – Análise do arquivo SCHARDT001

Fonte: Criado pelo autor.

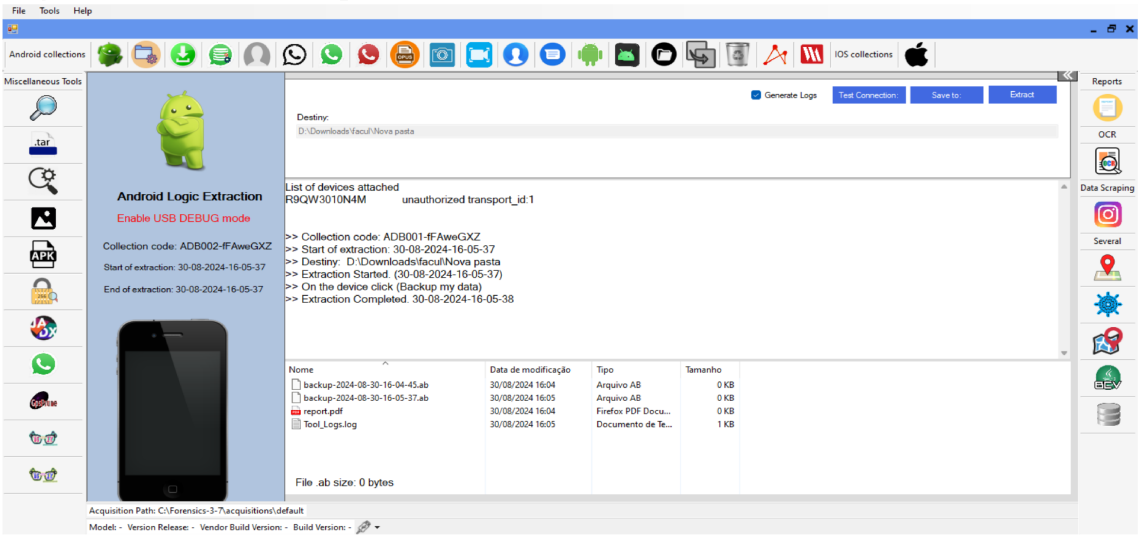
Ao verificar o resultado da análise na interface gráfica do Sistema IPED, constatou-se que não foram analisados dados relevantes da imagem alvo, além de alguns metadados que apareciam no canto da tela. Ao verificar o *hex* (leitura em hexadecimais) da extração, os dados pareciam corrompidos, demonstrando um possível problema na análise e assim não apresentando resultados satisfatórios.

Figura 17 – Supostos dados analisados pela ferramenta



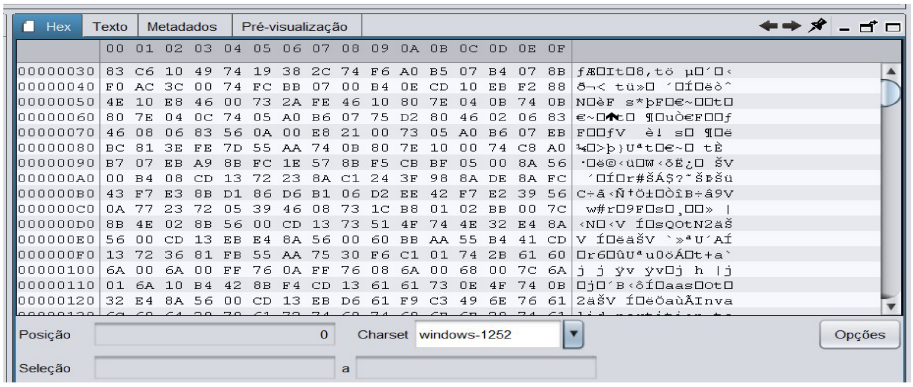
Fonte: Criado pelo autor.

Figura 19 – Tela de backup do Avilla Forensics



Fonte: Criado pelo autor.

Figura 18 – *Hex* da extração



Fonte: Criado pelo autor.

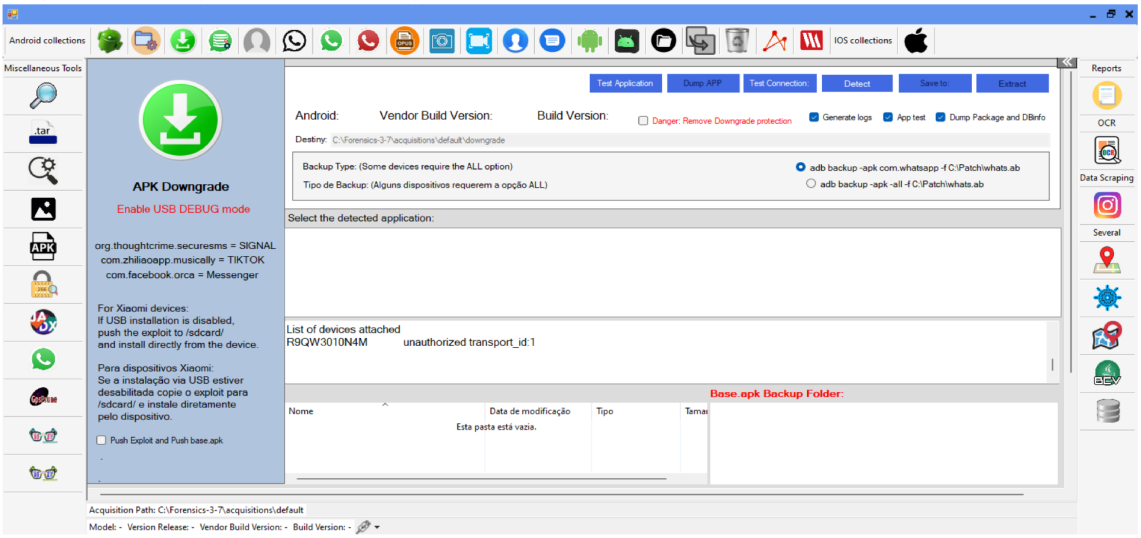
5.3.2 Avilla Forensics

Ao clicar na opção de detectar dispositivo, a ferramenta não consegue detectá-lo devidamente, não podendo nem ao mesmo nomear o dispositivo, mostrando uma mensagem de "transporte não autorizado". Ainda assim, a opção de realizar backup continua disponível e pode ser executada.

Ao executar o backup do dispositivo, um arquivo de backup é gerado, porém o mesmo não ocupa espaço nenhum, o que indica que seja apenas um arquivo vazio ou até mesmo corrompido. O erro não pode ser corrigido mesmo após repetidas tentativas. Apesar de não ser seguro seguir com a análise sem um backup adequado, foi tomada a decisão de seguir para a próxima etapa do teste.

A próxima etapa consiste em selecionar a opção "APK Downgrade", a terceira opção

Figura 20 – Seção "APK Downgrade"do Avilla Forensics



Fonte: Criado pelo autor.

da fileira, função da qual detecta e desativa as funções de proteção dos softwares do dispositivo alvo, como *WhatsApp* e *Instagram*. Para executar essa função, deve-se primeiro clicar na opção "*Detect*"que irá detectar o dispositivo conectado. Nesta etapa, o mesmo erro de "transporte não autorizado"aparece, impedindo assim de prosseguir e realizar a extração do dispositivo alvo. O teste então termina com um resultado não satisfatório.

5.4 Tabela de resultados

Para uma comparação mais justa entre as ferramentas, os critérios terão diferentes pesos, baseado na importância que aquele critério tem para o procedimento de extração. O êxito na extração tem um peso maior por ser o critério mais importante dentre os quatro, enquanto a especificação de dependências possui um peso menor por ser um critério que não afeta diretamente na parte prática. A divisão fica da seguinte forma:

a) Êxito na extração: 2 pontos;

b) Especificação das dependências: 0,5 pontos;

c) Execução sem erros de síntaxe: 1 ponto;

d) Atualização submetida recentemente: 1 ponto.

Figura 21 – Comparação final dos critérios entre as ferramentas.

Ferramentas / Critérios	Êxito na extração dos dados	Especificação das dependências	Execução sem erros de síntaxe	Atualização submetida nos últimos dois anos
AFLogical OSE	✓	✗	✓	✗
dd	✗	✓	✓	✗
Foremost	✗	✓	✓	✗
LiME	✗	✗	✓	✗
AMExtractor	✗	✗	✗	✗
Volatility	✗	✓	✗	✓
Sistema IPED	✗	✓	✓	✓
Avilla Forensics	✗	✓	✓	✓

Fonte: Criado pelo autor.

Outro resultado que não era esperado, mas foi consequência das análises, ocorreu no computador utilizado para realizar os testes, do qual teve um erro que impediu o seu funcionamento, só sendo consertado após uma formatação completa. Este erro, conhecido popularmente como "tela preta do fim", aconteceu apenas dois dias após os testes realizados, não tendo o computador apresentado nenhum defeito de *software* ou *hardware* anteriormente a isto.

6 CONCLUSÕES E TRABALHOS FUTUROS

Como conclusão para este trabalho, após analisar a tabela de resultados, pode-se concluir que, dentre as ferramentas selecionadas para análise, a ferramenta AFLogical OSE foi a que obteve melhor desempenho geral, sendo esta a única que conseguiu realizar a extração de dados com êxito.

Ainda que uma ferramenta tenha sido eleita como a melhor, existe um evidente problema nas ferramentas para perícia forense analisadas neste trabalho. A incapacidade de cumprir com êxito suas respectivas funções pode ocorrer por diversos motivos, dificultando assim a tarefa de apontar um erro específico. Contudo, é presumível que um dos principais motivos seja a incapacidade destas ferramentas de acompanhar a evolução da tecnologia Android, do qual possui atualizações constantes, até mesmo com atualizações maiores de versionamento, do qual foi da versão 11 para a versão 14 em um intervalo de quatro anos.

Era particularmente inesperado que as ferramentas "Sistema IPED"e "Avilla Forensics" fossem apresentar problemas tão graves, visto que estas são ferramentas profissionais utilizadas em casos de grande visibilidade no Brasil, e ambas das quais possuem atualizações constantes até os dias atuais.

Para futuros trabalhos, deseja-se desenvolver uma ferramenta única que terá como objetivo sanar todos os problemas analisados neste trabalho. Esta ferramenta seria de código aberto, com o intuito de manter atualizações constantes que acompanhem a evolução da tecnologia Android, assim tornando-a utilizável e com bom funcionamento a longo prazo.

Ademais, se faz interessante realizar um estudo para entender o motivo de os *smartphones* não poderem ser detectados e reconhecidos em máquinas virtuais. Esse estudo se faz muito importante, pois a utilização da máquina virtual ajuda a preservar a integridade do computador utilizado para análise e, por consequência, o aparelho alvo.

Outra alternativa interessante seria um estudo para compreender a razão pelas quais essas ferramentas não funcionam de modo adequado, visto que o motivo, como mencionado anteriormente, pode variar de ferramenta para ferramenta. Este suposto estudo poderia também abordar novas ferramentas, visto que algumas destas podem obter versões atualizadas, criadas por desenvolvedores que querem continuar com o trabalho das ferramentas que já não recebem atualizações há muito tempo.

REFERÊNCIAS

ACHINI, T. **Lei Carolina Dieckmann: Tudo o que você precisa saber sobre.** Projuris, [S.l.], 2023. Disponível em: https://www.projuris.com.br/blog/lei-carolina-dieckman-tudo-o-que-voce-precisa-saber-sobre/. Acesso em: 08 set. 2024.

AFD. **Ferramentas para Análise Forense em Android.** Disponível em: https://academiadeforensedigital.com.br/ferramentas-para-extracao-de-dados-em-android/. Acesso em: 08 set. 2024.

ANDERSON, T. **Open source Android Forensics app and framework.** Disponível em: https://github.com/nowsecure/android-forensics. Acesso em: 08 set. 2024.

ANDROID Open Source Project. **Platform Architecture.** [S.l.], Disponível em: <https://developer.android.co
br>. Acesso em: 08 set. 2024.

AURELIO, M. **Ferramenta Avilla Forensics: Explorando Métodos Alternativos de Instalação de Aplicativos em Dispositivos Xiaomi com MIUI.** Disponível em: https://apecof.org.br/index.php/artigos/27-ferramenta-avilla-forensics-explorando-metodos-alternativos-de-instalacao-de-aplicativ
Acesso em: 08 set. 2024.

AVILLA, D. **(NEW) Avilla Forensics 3.7.** Disponível em: https://github.com/AvillaDaniel/AvillaForensics. Acesso em: 08 set. 2024.

BARROS, F. **Ciências forenses**: princípios éticos e vieses. Disponível em: https://www.scielo.br/j/bioet/a/GYNrWJgbtfwQskD5TR7dCGN/#. Acesso em: 08 set. 2024.

BRASIL. Ministério da Justiça. **Curso de investigação de crimes de homicídio e drogas ilegais: aspectos legais, técnicos e investigativos dos exames periciais.** Brasília, DF: Ministério da Justiça. [S.d.] Disponível em: <https://www.passeidireto.cm/arquivo/93285470/aspectos-legais-tecnicos-e-investigativos-dos-exames-periciais>. Acesso em: 11 out. 2024.

CASEY, E. **Digital Evidence and Computer Crime.** Disponível em: https://rishikeshpansare.wordpress.com/wp-content/uploads/2016/02/digital-evidence-and-computer-crime-third-edition.pdf. Acesso em: 08 set. 2024.

CASPIAN, Milo. **Android Archiestrutura**: Camadas de aplicação, estrutura, componente. [S.l.], 26 set. 2024. Disponível em: <https://www.guru99.com/pt/android-architecture.html>. Acesso em: 08 set. 2024.

CILLERUELO, C. **¿Qué es AFLogical OSE?** Disponível em: https://keepcoding.io/blog/que-es-aflogical-ose-open-source-edition/. Acesso em: 08 set. 2024.

CORRELAÇÃO de Evidências Digitais e Forense Computacional. Centro de Ciências Forenses , [s.d.]. Disponível em: <https://ccf.c3sl.ufpr.br/correlacao-de-evidencias-digitais-e-forense-computacional/>. Acesso em: 08 set. 2024.

DARYUS, C. **Computação forense**: O que é e como funciona? Disponível em: https://blog.daryus.com.br/computacao-forense-o-que-e-e-como-funciona-2/. Acesso em: 08 set. 2024.

GARBES, A. **Tipos de perícia no processo criminal**: ferramentas essenciais para a elucidação de crimes. 2024. Disponível em: https://www.jusbrasil.com.br/artigos/tipos-de-pericia-no-processo-criminal-ferramentas-essenciais-para-a-elucidacao-de-crimes/2345523747. Acesso em: 08 set 2024.

GONÇALVES, M. et al. **Perícia Forense Computacional: Metodologias, técnicas e ferramentas. Mato Grosso. Revista Científica Eletrônica de Ciências Sociais Aplicadas da EDUVALE,** nov, 2012. Disponível em: http://www.eduvalesl.revista.inf.br/imagens/arquivos/arquivos/destaque/LXkEA5E
12-19-2-33-33.pdf. Acesso em: 08 set. 2024.

GUSMÃO, S. **Extração de Dados em Dispositivos Móveis – Técnicas Avançadas.** Disponível em: https://academiadeforensedigital.com.br/extracao-de-dados-em-dispositivos-moveis-tecnicas-avancadas/
Acesso em: 08 set. 2024.

IBM. **O que é computação forense?** Disponível em: https://www.ibm.com/br-pt/topics/computer-forensics. Acesso em: 08 set. 2024.

iMHLv2. **An advanced memory forensics framework.** Disponível em: https://github.com/volatilityfoundation/volatility. Acesso em: 08 set. 2024.

JALES, A. **Crimes Virtuais e Perícia computacional Forense**: Um estudo sobre... Jus Navigandi, jun. 2023. Disponível em: <https://jus.com.br/artigos/104611/crimes-virtuais-e-pericia-computacional-forense-um-estudo-sobre-a-atividade-de-investigacao-criminal-no-ambiente-digital>. Acesso em: 08 set 2024.

KALI ORG. **Foremost**: Tool Documentation. Disponível em: https://www.kali.org/tools/foremost/. Acesso em: 08 set. 2024.

MEDEIROS, Lamartine de Oliveira; TORRES, Anderson Barros. **Análise de ferramentas open source utilizadas para a perícia forense computacional.** 2020. 25 f. Trabalho de Conclusão de Curso. Escola de Aperfeiçoamento de Oficiais, . Brasília, DF, 2020. Acesso em: 08 set. 2024.

MENEZES, A. **Análise Forense em Dispositivos Móveis com o Sistema Operacional Android.** Disponível em: https://www.marcelolau.com.br/wp-content/uploads/2020/03/An%C3%A1lise-Forense-em-Dispositivos-M%C3%B3veis-com-Sistema-Operacional-Android.pdf. Acesso em: 08 set. 2024.

NASSIF, L. **IPED Digital Forensic Tool.** Disponível em: https://github.com/sepinf-inc/IPED. Acesso em: 08 set. 2024.

OUTLE, A. **LiME Forensics Kernel Module for Raw Memory Snapshots.** Disponível em: https://www.xda-developers.com/lime-forensics-kernel-module-for-raw-memory-snapshots/. Acesso em: 08 set. 2024.

RESPER. **Evidências Digitais**: Um Guia Completo De Forense Digital. Disponível em: https://forense.io/evidencias-digitais/. Acesso em: 08 set. 2024.

SYLVE, J. **LiME Linux Memory Extractor.** Disponível em: https://github.com/504ensicsLabs/LiME. Acesso em: 08 set. 2024.

APÊNDICE A – REPOSITÓRIOS UTILIZADOS

Inicialmente, foi planejado que os testes seriam feitos através de uma ou mais máquina virtuais. Com um programa de virtualização de máquina instalado, sendo o *Oracle VirtualBox* o programa usado para este primeiro teste, o sistema operacional (SO) pode ser instalado. O SO escolhido foi o *Santoku Linux*, visto que este é voltado para fins de perícia forense e até mesmo já possui uma das ferramentas deste trabalho pré-configurada. A máquina foi configurada para possuir 4 GB (*gigabytes*) de memória RAM, ainda que 2 GB poderiam ser suficientes. Utiliza 2 núcleos do processador e tem 15 GB de disco rígido. É importante ressaltar que para a instalação do Santoku Linux, faz-se interessante possuir, no mínimo, 8.3 GB de espaço de armazenamento livre, assim como recomendado pelo próprio sistema operacional, a fim de evitar problemas na instalação.

Para evitar certos problemas que foram encontrados serão descritos ao longo deste trabalho, é importante adicionar alguns repositórios ao sistema operacional utilizado. Vale ressaltar que estes repositórios funcionam apenas para os sistemas operacionais baseados em Debian Linux, visto que estes foram os utilizados neste trabalho. Estes repositórios, assim como outros, estão disponíveis através do site oficial Debian Wiki. Os repositórios podem ser adicionados através dos seguintes comandos, que devem ser executados com acesso root do terminal:

```
1  echo "deb http://deb.debian.org/debian bullseye main"
       >> /etc/apt/sources.list
2
3  echo "deb-src http://deb.debian.org/debian bullseye
       main" >> /etc/apt/sources.list
4
5  echo "deb http://deb.debian.org/debian-security/
       bullseye-security main" >> /etc/apt/sources.list
6
7  echo "deb-src http://deb.debian.org/debian-security/
       bullseye-security main" >> /etc/apt/sources.list
8
9  echo "deb http://deb.debian.org/debian bullseye-updates
```

```
        main" >> /etc/apt/sources.list
10
11      echo "deb-src http://deb.debian.org/debian bullseye-
           updates main" >> /etc/apt/sources.list
```

De modo adjacente, é possível utilizar um editor de texto como *vim* ou *nano* para editar o arquivo "sources.list" e adicionar manualmente os *links* acima listados, linha após linha.

Após adicionais todos os repositórios ao arquivo "souces.list", utilize o seguinte comando para que o sistema operacional verifique e atualize efetivamente os repositórios:

```
1       sudo apt-get update
```

É também vital para a instalação das ferramentas, possuir mais três recursos mencionados no comando abaixo.

```
1       sudo apt-get install git gcc make
```

APÊNDICE B – AFLOGICAL OSE

O primeiro software selecionado para ser testado e avaliado é o AFLogical OSE. A ferramenta é facilmente acessível através do sistema operacional Santoku Linux, e que a possui já instalada e pré-configurada para uso. O software AFLogical OSE está incluso na categoria de Ferramenta de Memória Não-Volátil, sendo assim, suas funcionalidades serão avaliadas de acordo com os objetivos desta categoria.

Figura 22 – Tela inicial do Santoku Linux

Fonte: Criador pelo autor.

Ao entrar no ambiente gráfico, é possível perceber um executável no canto superior esquerdo que ao realizar um duplo clique nele, irá instalar todos os recursos do Santoku Linux, bem como a ferramenta necessária para este estudo.

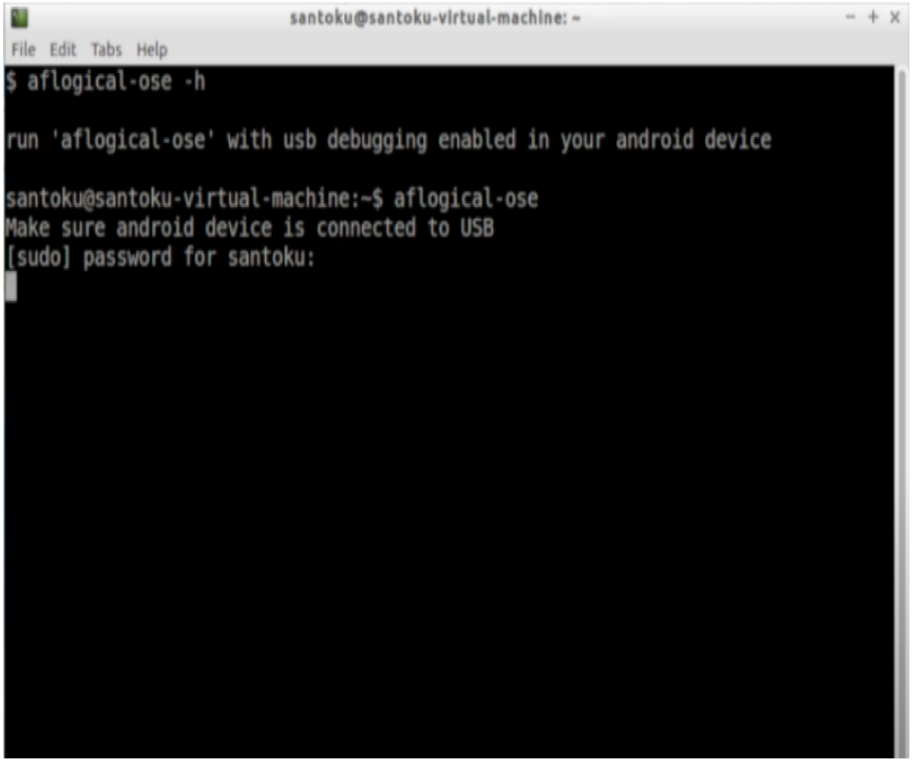
Para fazer a execução da ferramenta, é preciso apenas abrir o menu, clicando na faca no canto inferior esquerdo, ir até o submenu ”Santoku”, depois o submenu ”Device Forensics”e clicar na ferramenta ”AFLogical OSE”. Com o terminal da ferramenta aberto, agora é necessário estar com o dispositivo Android pronto. O dispositivo Android deve estar conectado via USB e com a depuração USB habilitada nas configurações do Android. Para verificar se a entrada USB está sendo reconhecida, pode ser verificado na seção ”Removable Devices” do VMWare. Com

tudo pronto, resta executar o seguinte comandono terminal:

```
1    aflogical-ose
```

A senha de root será requisitada, sendo essa a mesma de login do sistema.

Figura 23 – Terminal do AFLogical OSE



Fonte: Criado pelo autor.

O dispositivo Android irá solicitar permissão do usuário para que a ferramenta colete as informações. Após cedida a permissão, o software irá realizar o resto do processo automaticamente e armazenar as informações em uma pasta reservada.

APÊNDICE C – DD

Apesar de o dd poder ser normalmente baixado e executado no *Santoku Linux*, existe um grande empecilho que impede que sua execução seja devidamente realizada. Assim como outras ferramentas, o dd utiliza do diretório de montagem do dispositivo para fazer a extração. Existem múltiplas maneiras de realizar a montagem de um dispositivo Universal Serial Bus (USB). Contudo, os dispositivos com Android apresentam uma particularidade. Ao utilizar o seguinte comando, seria esperado visualizar o dispositivo Android sendo listado:

```
1    lsblk -l
```

Os dispositivos com Android não se comportam da forma esperada de um aparelho conectado via USB, e por isso não aparecem nesta listagem, como é possível visualizar na Figura. Devido a isso, é necessário utilizar um programa capaz de operar arquivos em dispositivos de *Media Transfer Protocol* (MTP) conectados via USB à máquina local.

Existem múltiplas ferramentas que podem realizar este trabalho, bem como: go-mtpfs, simple-mtpfs, jmtpfs, dentre outras. Apesar de serem amplamente utilizadas com o propósito de realizar a montagem de dispositivos com Android, o resultado esperado não foi obtido ao executar cada uma das ferramentas. Elas apresentaram diferentes erros, em sua maioria, de não identificar nenhum dispositivo USB conectado, mesmo que o programa de virtualização de máquina estivesse indicando que o aparelho utilizado para o teste estivesse devidamente conectado.

Devido aos problemas apresentados no Santoku Linux, bem como no funcionamento das ferramentas de montagem de partição, foi tomada a decisão de experimentar um sistema operacional alternativo. Com isto, foi criada uma nova máquina virtual com as mesmas especificações da anterior, mas agora utilizando o sistema operacional Kali Linux, uma distribuição Linux baseada em Debian.

Era esperado que o Kali Linux realizasse a montagem de partição do aparelho de teste no momento em que este fosse conectado, contudo, ao conectar o aparelho e permitir acesso aos dados através do celular, estes dados não apareciam na máquina virtual. Mesmo ao tentar utilizar as mesmas ferramentas de montagem de partição previamente utilizadas, os resultados continuavam sendo os mesmos.

Com isto, foi utilizada uma abordagem com o sistema operacional Arch Linux, já que este tem a ferramenta "simple-mtpfs"disponível em seu diretório principal, facilitando assim sua

instalação e instalação de dependências necessárias para o funcionamento adequado. No sistema operacional Arch Linux, o simple-mtpfs pode ser instalado utilizando o seguinte comando:

```
pacman -S simple-mtpfs
```

Ainda assim, os erros persistiam, mesmo neste sistema operacional. Com isto, foi levantada a teoria de que a confirmação adicional que o aparelho de teste requisitava causava um problema no programa de virtualização de máquinas, fazendo com que a permissão concedida para acessar os dados não fosse aceita pelo host (máquina real) ou pela máquina virtual, ainda que o suporte para USB do programa Oracle VirtualBox estivesse atualizado na versão mais recente.

Para comprovar a teoria levantada, foi novamente instalado o sistema operacional Kali Linux, mas desta vez em uma partição do Solid State Drive (SSD - dispositivo de armazenamento do computador) da máquina real.

Ao realizar a instalação e configuração básicas do sistema operacional e conectar o aparelho de teste, a montagem de partição foi feita de forma automática e sem problemas, necessitando apenas abrir o menu suspenso com o botão direito do *mouse* e clicar na opção "Exibir no terminal" para poder verificar o diretório de montagem de partição e obter esta informação essencial para a posterior execução das ferramentas. O dd é uma ferramenta presente de forma nativa em boa parte das distribuições Linux baseadas em Debian, inclusive no Kali Linux e Santoku Linux, o que significa que não é necessário fazer instalação. Caso o sistema operacional que esteja sendo utilizado seja baseado em Debian mas não tenha o dd instalado de forma nativa, é necessário apenas executar o seguinte comando:

```
sudo apt-get install dd
```

Para executar o dd, é necessário utilizar um comando no terminal Linux que requisita dois argumentos principais: o diretório do dispositivo a ser extraído e o diretório de saída. No cenário de teste, o comando ficou da seguinte de forma:

```
sudo dd if=/run/user/1000/gvfs/mtp:host=
   SAMSUNG_SAMSUNG_Android_R9QW3010N4M of=/root/copia.
   img bs=2048
```

Na estrutura de comando do dd, o argumento em "if"corresponde ao diretório do

dispositivo alvo, enquanto o argumento em "of"corresponde ao diretório do arquivo de saída e o argumento "bs"representa em quantos blocos a imagem deve ser analisada e copiada, utilizada por padrão com o valor 2048.

APÊNDICE D – FOREMOST

A ferramenta Foremost não está nativamente instalada no Kali linux, mas felizmente pode ser facilmente instalada através do seguinte comando:

```
1    sudo apt-get install foremost
```

Para executar o comando da ferramenta Foremost, são necessários três argumentos principais, sendo estes: o tipo de arquivo que deve ser extraído, o diretório do dispositivo alvo e o diretório de saída. No cenário de teste, o comando ficou da seguinte de forma:

```
1    sudo foremost -q -v -t png -i /run/user/1000/gvfs/mtp:
       host=SAMSUNG_SAMSUNG_Android_R9QW3010N4M -o /output
```

Após executar este comando, a ferramenta irá começar o processo de extração. No caso deste primeiro teste, foi realizada uma tentativa de extração de arquivos com a extensão portable network graphic (png), uma conhecida extensão de imagens.

APÊNDICE E – LIME

A ferramenta LiME não está disponível nos repositórios Debian mencionados anteriormente, sendo assim, ela deve ser baixada a partir da sua página oficial no GitHub. Os recursos podem ser baixados e configurados através dos seguintes comandos:

```
1    apt-get install adb
2    git clone https://github.com/504ensicsLabs/LiME.git
3    cd LiME/src
4    make
```

Com isto, o LiME está pronto para execução, sendo necessário como argumento para o seu comando de execução, apenas o arquivo gerado após o último comando e o diretório alvo da extração, repetindo o arquivo .ko ao fim para definir aquele diretório como o local onde este arquivo será enviado.

```
1    adb push lime-6.8.11-amd64.ko /run/user/1000/gvfs/mtp:
       host=SAMSUNG_SAMSUNG_Android_R9QW3010N4M/lime
       -6.8.11-amd64.ko
```

APÊNDICE F – AMEXTRACTOR

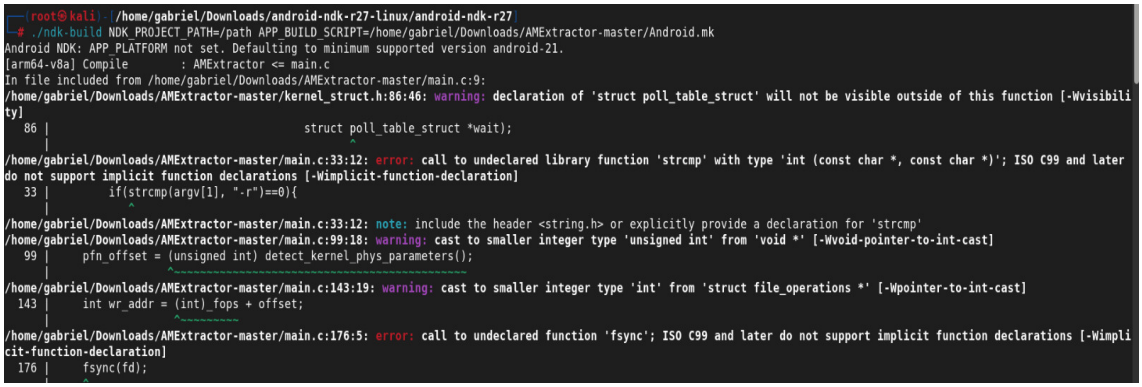
Assim como a ferramenta previamente analisada, o AMExtractor não está disponível nos repositórios Debian, por isso deve ser baixado de maneira semelhante ao LiME, através da sua página no Github. Para este, também é necessário ter o "NDK-Build"que pode ser baixado diretamente no site do NDK.

Após baixar ambos os arquivos e extrai-los, o AMExtractor pode ser executado através dos seguintes comandos:

```
cd android-ndk-r27-linux/android-ndk-r27
./ndk-build NDK_PROJECT_PATH=/path APP_BUILD_SCRIPT=/
    home/user/Downloads/AMExtractor-master/Android.mk
```

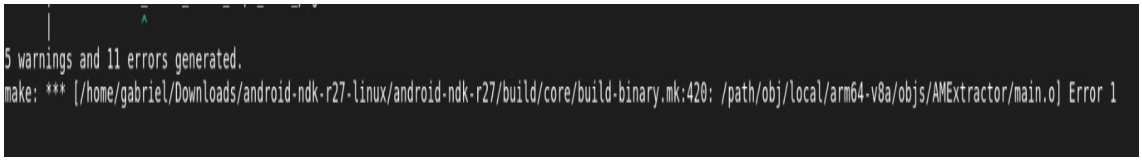
A ferramenta apresentou alguns avisos, bem como erros ocorridos na execução do código, como é possível visualizar na Figura. Contudo, o procedimento pode seguir normalmente.

Figura 24 – Execução do NDK



Fonte: Criado pelo autor.

Figura 25 – Avisos de erro



Fonte: Criado pelo autor.

O próximo passo é executar um comando "adb", executando um método muito semelhante ao usado na ferramentas anteriormente analisada.

```
cd /path
```

```
2    adb push libs/armeabi/AMExtractor /data/local/tmp
```

APÊNDICE G – VOLATILITY

A instalação da ferramenta Volatility se dá de modo semelhante ás ferramentas anteriormente mostradas, no qual os arquivos da ferramenta devem ser baixados diretamente da sua página no GitHub.

```
1  git clone https://github.com/volatilityfoundation/
       volatility.git
2  cd volatility/
3  setup.py build
4  setup.py install
```

APÊNDICE H – SISTEMA IPED

O Sistema IPED, bem como a próxima ferramenta a ser analisada, possuem apenas uma versão feita para o sistema operacional Windows. Por conta disso, o restante da análise será feita no mesmo dispositivo utilizado até então, mas dessa vez, no sistema operacional Windows 11.

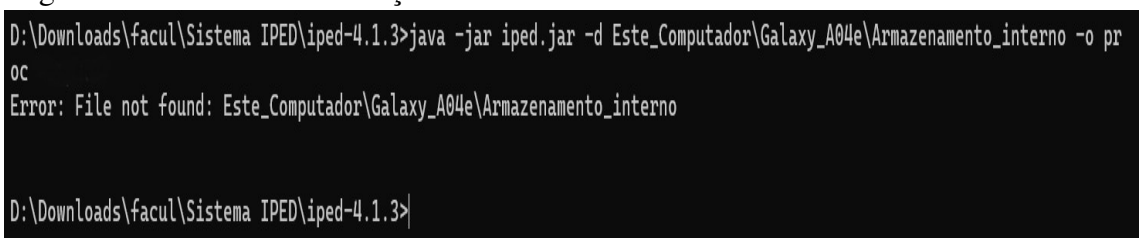
Para baixar a ferramenta, é necessário apenas buscar por sua última versão na sua página do GitHub, assim como feito outras vezes ao longo deste estudo. Além disso, é necessário ter o software "Java", especificamente, em sua versão 11. Esta versão em específico pode ser baixada de forma gratuita e de fácil acesso através do site da Liberica JDK.

Para executar o Sistema IPED, basta navegar até a pasta onde ele foi baixado e utilizar o seguinte comando:

```
1    java -jar iped.gar -d Este_Computador\Galaxy_A04e\
        Armazenamento_interno -o proc
```

O diretório do dispositivo alvo pode mudar a depender do dispositivo alvo. Ao executar o comando, o Sistema IPED acusa um erro de não ter encontrado o dispositivo referido, como apresentado na Figura.

Figura 26 – Tentativa de extração com o Sistema IPED



Fonte: Criado pelo autor.

Isso possivelmente se deve ao fato da ferramenta não ser propriamente preparada para extração de arquivo, apenas análise. Como consequência de as outras ferramentas feitas para extração de memória terem falhado, chegou-se à conclusão de que esta ferramenta deveria ser analisada com sua execução realizada a partir de um arquivo de imagem já existente. É possível conseguir um arquivo para testes de forma prática no *National Institute of Standards and Technology*. O arquivo, nomeado de "SCHARDT.001"pode ser baixado ao clicar no mesmo.

Para executar a ferramenta, usa-se o mesmo comando, desta vez alterando o diretório alvo para o diretório onde o arquivo "SCHARDT.001"se encontra.

APÊNDICE I – AVILLA FORENSICS

A ferramenta Avilla Forensics pode ser facilmente baixada através da sua página no GitHub. O assistente de instalação é bem intuitivo, facilitando assim o procedimento.

Com a ferramenta instalada, múltiplas funções são apresentadas na tela, visto que esta ferramenta tem várias utilidades. O primeiro passo é utilizar a função localizada na barra de cima, na ponta da esquerda, para realizar um backup do dispositivo. Este procedimento é essencial para manter os dados originais do aparelho alvo.