DISSERTAÇÃO DE MESTRADO PROFISSIONAL

FINGERPRINTS DE SISTEMAS OPERACIONAIS PARA IDENTIFICAR EQUIPAMENTOS IOT ESPÚRIOS NA AUSÊNCIA DE CONTROLE DE ADMISSÃO À REDE, UMA PROPOSTA E IMPLEMENTAÇÃO.

ATILA BATISTA BANDEIRA

Programa de Pós-Graduação Profissional em Engenharia Elétrica

DEPARTAMENTO DE ENGENHARIA ELÉTRICA
FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO PROFISSIONAL

FINGERPRINTS DE SISTEMAS OPERACIONAIS PARA IDENTIFICAR EQUIPAMENTOS IOT ESPÚRIOS NA AUSÊNCIA DE CONTROLE DE ADMISSÃO À REDE, UMA PROPOSTA E IMPLEMENTAÇÃO.

ATILA BATISTA BANDEIRA

Dissertação de Mestrado Profissional submetida ao Departamento de Engenharia Elétrica como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica

Banca Examinadora

Prof. Dr. Clóvis Neumann FT/PPEE/UnB _____
Orientador

Prof. Dr. Fábio Lúcio Lopes de Mendonca, ENE/UnB _____
Examinador Interno

Prof. Dr. Laerte Peotta de Melo, Banco do Brasil _____
Examinador Externo

Prof. Dr. Daniel Alves da Silva, PPEE/UnB _____
Suplente

FICHA CATALOGRÁFICA

BANDEIRA, ÁTILA BATISTA
FINGERPRINTS DE SISTEMAS OPERACIONAIS PARAIDENTIFICAR EQUIPAMENTOS IOT ESPÚRIOS NAAUSÊNCIA DE CONTROLE DE ADMISSÃO À REDE,UMA PROPOSTA E IMPLEMENTAÇÃO. [Distrito Federal] 2024.
xvi, 41 p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2024).
Dissertação de Mestrado Profissional - Universidade de Brasília, Faculdade de Tecnologia.
Departamento de Engenharia Elétrica

1. IoT	2. Internet das Coisas
3. fingerprints de SO	4. Dispositivos não autorizados
I. ENE/FT/UnB	II. Título (série)

PUBLICAÇÃO: PPEE.MP.067

REFERÊNCIA BIBLIOGRÁFICA

BANDEIRA, ÁTILA. (2024). *FINGERPRINTS DE SISTEMAS OPERACIONAIS PARAIDENTIFICAR EQUIPAMENTOS IOT ESPÚRIOS NAAUSÊNCIA DE CONTROLE DE ADMISSÃO À REDE,UMA PROPOSTA E IMPLEMENTAÇÃO.*. Dissertação de Mestrado Profissional, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 41 p.

CESSÃO DE DIREITOS

AUTOR: ATILA BATISTA BANDEIRA
TÍTULO: FINGERPRINTS DE SISTEMAS OPERACIONAIS PARAIDENTIFICAR EQUIPAMENTOS IOT ESPÚRIOS NAAUSÊNCIA DE CONTROLE DE ADMISSÃO À REDE,UMA PROPOSTA E IMPLEMENTAÇÃO..
GRAU: Mestre em Engenharia Elétrica ANO: 2024

ATILA BATISTA BANDEIRA
Depto. de Engenharia Elétrica (ENE) - FT
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP 70919-970 - Brasília - DF - Brasil

DEDICATÓRIA

Dedico este trabalho a minha querida esposa Gracielle Leal Vasconcelos Bandeira por acreditar na minha capacidade e ter paciência dando total apoio e incentivo.

Ao meu amigo Carlos Eduardo da Costa Gonçalves e Ana Cláudia Lemos pela ajuda com os testes e análise de resultados.

AGRADECIMENTOS

Os Autores agradecem ao apoio técnico e computacional do LATITUDE, da UnB, que conta com apoio do CNPq - (Outorgas 312180/2019-5 PQ-2 e 465741/2014-2 INCT em Cibersegurança), ao Mestrado Profissional em Engenharia Elétrica, na área de concentração: Segurança Cibernética – 1ª Turma para Profissionais do Setor de Inteligência (Outorga ABIN 01/2019) ao DPI da UnB (Outorga 7129 e do Projeto SISTER City (Outorga 625/2022) e a FAP/DF. A validação deste trabalho foi assegurada pelo Banco do Brasil S.A., que permitiu testes em laboratório, espelhando parte de sua infraestrutura para coleta e análise dos dados hipotéticos e imagens de tentativas reais de ataques, apresentados neste artigo.

Agradeço ao meu orientador, Clóvis Neumann, ao meu amigo e professor Dr. Fábio Lúcio Lopes de Mendonça, que me ajudou na orientação de forma profissional e amiga nas horas mais complicadas durante este trabalho e aturou tantas dúvidas e problemas relativos ao assunto e outros detalhes pertinentes à criação desta dissertação, mesmo nos momentos em que os inúmeros compromissos de trabalho pareciam levar o sonho deste Mestrado para mais longe.

Ao meu Coorientador, professor Dr. Georges Daniel Amvame NZE, que me ajudou de forma significativa na minha trajetória, com suas orientações e dicas relacionadas ao tema, ele que é uma grande referência para todos seus alunos.

Aos Professores do Programa de Pós Profissional em Engenharia Elétrica da UnB, Rafael Timóteo de Sousa Júnior, Rafael Rabelo, Edna Canedo, pelas grandes dicas, constante apoio, incentivo e amizade, essenciais para o desenvolvimento deste trabalho. Agradeço também aos membros da banca.

Ao Dr. Laerte Peotta de Melo que sempre me incentivou e contribuiu de forma fundamental para a conclusão deste trabalho: meus sinceros agradecimentos.

Agradeço, acima de tudo, a Deus!

RESUMO

A proliferação de dispositivos da Internet das Coisas (IoT) apresenta desafios significativos para a segurança das redes de computadores. A possibilidade de conexão cabeada em pontos de rede disponíveis fisicamente pode ser uma das técnicas utilizadas para a introdução de artefatos maliciosos, por meio dos quais diversos tipos de ataques cibernéticos podem ser realizados. Detectar e responder a conexões não autorizadas desses dispositivos é fundamental para se manter a integridade e a segurança das redes, principalmente em situações em que o controle de admissão de rede (NAC) ainda não foi implementado. Este trabalho propõe e avalia a utilização de fingerprints de sistemas operacionais (SO) para analisar características específicas de dispositivos conectados a uma rede cabeada. Os resultados obtidos mostram a capacidade da proposta de utilizar esses fingerprints para identificar dispositivos que não correspondem ao esperado, possibilitando medidas de contenção e aprimorando a segurança da rede mesmo na ausência de NAC, ou enquanto essa tecnologia de proteção não estiver implantada.

ABSTRACT

The proliferation of Internet of Things (IoT) devices presents significant challenges for computer network security. The possibility of a wired connection to physically available network points can be one of the techniques used to introduce malicious artifacts, through which various types of cyber attacks can be carried out. Detecting and responding to unauthorized connections from these devices is essential for maintaining network integrity and security, especially in situations where Network Admission Control (NAC) has not yet been implemented. This work proposes and evaluates the use of operating system (OS) fingerprints to analyze specific characteristics of devices connected to a wired network. The results obtained show the proposal's ability to use these fingerprints to identify devices that do not correspond to what is expected, enabling containment measures and improving network security even in the absence of NAC, or while this protection technology is not in place.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

A Internet das Coisas *IoT* tem emergido como uma força transformadora na sociedade moderna, marcando a evolução tecnológica com sua capacidade de conectar dispositivos inteligentes à internet (MENDONÇA 2019).

Esta integração abrange uma ampla gama de dispositivos, desde eletrodomésticos comuns até sistemas complexos de monitoramento industrial, cada um contribuindo para um ecossistema interconectado que promove a eficiência e a inovação.

A adoção desses dispositivos inteligentes tem sido notável em setores críticos como saúde, onde permitem o monitoramento remoto de pacientes, transporte, com sistemas avançados para gestão de tráfego, automação residencial que transforma a experiência de viver em casa, agricultura com soluções de precisão para cultivo, e na indústria, otimizando processos através da automação e coleta de dados em tempo real.

Conforme identificado por Zanella et al. (2014), essa tendência não só eleva a qualidade de vida como também introduz complexidades na gestão e segurança desses dispositivos.

A crescente conectividade trazida pela *IoT*, apesar de seus inúmeros benefícios, lança desafios significativos para a segurança das redes.

A natureza heterogênea dessas redes, compostas por uma variedade de dispositivos *IoT*, torna-as particularmente suscetíveis a ataques cibernéticos. Estes dispositivos, muitas vezes com segurança limitada, podem ser explorados para comprometer não apenas a privacidade dos usuários, mas também a integridade dos dados e a segurança física dos ambientes em que estão inseridos.

Assim, para um dispositivo que não corresponda ao esperado para a rede, ou cuja conexão tenha burlado um processo de gerenciamento de dispositivos pode ser chamado neste trabalho de "dispositivo não autorizado". (Lin e Tang 2018).

A detecção e controle de dispositivos conectados a uma rede de computadores é fundamental para mitigar os riscos associados, enfatizando a necessidade de estratégias de segurança robustas adaptadas à complexidade das redes *IoT* (Rocha, Melo e Sousa 2021).

1.1 DESCRIÇÃO DO PROBLEMA

A possibilidade de dispositivos não autorizados e que contenham artefatos *IoT* maliciosos serem conectados a uma rede de computadores representa uma ameaça considerável à integridade das redes que desafia os mecanismos tradicionais de segurança. Esses dispositivos podem ser utilizados para ataques cibernéticos como os de negação de serviço (DoS), espionagem de dados sensíveis e a participação em botnets e também podem conter certas características como endereços MAC desconhecidos ou *fingerprints* sistemas operacionais que não correspondem aos padrões esperados em uma rede.

A capacidade desses dispositivos de atuar sob o radar sublinha a importância de desenvolver métodos

de detecção e resposta eficazes para proteger as redes contra ameaças emergentes (Kumari e Jain 2023) Portanto, é prudente implantar mecanismos para identificar e responder a esses dispositivos não autorizados.

Embora o Controle de Admissão de Rede (*NAC*) seja uma solução comum para controlar o acesso de dispositivos, sua implementação pode ser complexa e demorada, especialmente em ambientes com muitos dispositivos em operação. Além disso, nem sempre é viável ou prático implementar o *NAC* em todas as redes existentes, uma vez que nem todos os equipamentos de rede são compatíveis com essa tecnologia, como alguns tipos de Switches. Diante desse cenário, novas abordagens são necessárias para detectar e responder a dispositivos não autorizados em redes sem o *NAC* ou em que esta tecnologia encontra-se em processo de implantação.

Um dos principais desafios que empresas, em especial algumas Instituições Financeiras (IF) brasileiras tem enfrentado é a tentativa de acesso não autorizado em suas redes de comunicação por meio de dispositivos conectados fisicamente aos pontos de rede interna de suas agências e Terminais de Autoatendimento (TAA/ATM), visando espionagem, roubo de credenciais e consecução de ataques e fraudes.

1.2 MOTIVAÇÃO

Algumas Instituições Financeiras (IF) no Brasil, vem sendo alvo de um tipo de ataque que, a princípio, seja de execução simples, mas cujo impacto e nível de dificuldade para detecção e resposta tem sido desafiador.

o ataque começa com a identificação de um ponto físico de rede no qual um equipamento espúrio, previamente preparado com artefatos maliciosos, será conectado. Essa técnica encontra-se descrita ne Seção "Modelo adversarial - MITRE ATTCK"neste trabalho. A conexão então é efetuada por um atacante, diretamente, ou por meio de colaboradores da Instituição (terceirizados, estagiários ou funcionários) os quais são aliciados para conectar o equipamento em uma agência bancária, por exemplo.

Em seguida, uma vez conectado a um ponto físico, o atacante se utiliza do dispositivo ora conectado para fazer um mapeamento da rede, de portas, configurações e de possíveis credenciais não protegidas. Caso não encontre credenciais desprotegidas, técnicas de engenharia social, cooptação e até filmagem de digitação são utilizados para captura de senhas.

De posse de senhas, são realizados ataques remotos emulando máquinas da Instituição Financeira. Nesse ataque "remoto", os dispositivos conectados são a ponte entre o agente malicioso e os sistemas atacados.

De forma mais detalhada, o ataque às Instituições Financeiras (IF) no Brasil envolve uma série de etapas meticulosamente planejadas e executadas pelos atacantes cujas etapas serão detalhadas a seguir.

Fase 1: Reconhecimento e Infiltração

O ataque começa com uma fase de reconhecimento, onde os atacantes identificam um ponto físico de rede vulnerável dentro da instituição financeira. Este ponto pode ser um terminal de computador em uma agência bancária ou qualquer dispositivo conectado à rede interna que possa ter um ponto de rede para ser

fisicamente conectado. O reconhecimento pode ocorrer de várias formas, incluindo a observação direta, o aliciamento de funcionários para obter informações internas para mapear a infraestrutura de rede a ser atacada.

Fase 2: Preparação do Dispositivo

Antes da execução do ataque, o dispositivo é cuidadosamente preparado com artefatos maliciosos. Este dispositivo pode ser um laptop, um Raspberry Pi ou qualquer outro hardware capaz de ser conectado à rede e executar softwares maliciosos (neste trabalho também chamado de dispositivo espúrio). Tal dispositivo é equipado com uma gama de ferramentas destinadas a explorar vulnerabilidades, capturar tráfego de rede, executar mapeamento de rede e comprometer sistemas através de explorações.

Fase 3: Inserção Física e Conexão à Rede

O dispositivo preparado é então inserido fisicamente na rede da instituição. Isso pode ser feito diretamente pelo atacante, que encontra uma maneira de acessar fisicamente as instalações, ou por meio de um colaborador interno da instituição que foi aliciado ou coagido a realizar essa ação. A inserção do dispositivo pode ser tão simples quanto conectar um USB em um computador desatendido ou tão complexo quanto instalar um hardware escondido dentro de uma sala de servidores.

No caso das Instituições Financeiras, os pontos de atendimento presencial (Agências) e os Terminais de Autoatendimeneto Presencial (TAA ou Caixas Eletrônicos), bem como os computadores utilizados pelos atendentes nas agências geralmente possuem proximidade com pontos de acesso físico à rede. Isso facilita a escolha do atacante quanto ao ponto no qual seu dispositivo espúrio será conectado.

Outro fator a ser considerado é que nem todas as instituições atacadas tem um controle de acesso de dispositivos (*NAC*) à rede totalmente implementado. No caso das IF, pode ser devido mais a questões de compatibilidade com o legado do que por falta de orçamento. Mesmo as que possuem capacidade de investimento ainda tem que lidar com incompatibilidades técnicas ou a necessidade de se mudar toda a arquitetura de rede atual para implantar ferramentas de detecção e resposta como o *NAC* (Helfrich et al. 2006) (Dildy).

Fase 4: Mapeamento da Rede e Captura de Credenciais

Uma vez que o dispositivo está conectado à rede, ele é ativado para realizar um mapeamento detalhado daquele segmento da infraestrutura de rede da instituição. Esse mapeamento inclui a identificação de servidores, terminais, dispositivos de rede (como roteadores e switches), portas disponíveis, serviços em execução e eventuais fragilidades que possam ser exploradas. Paralelamente, o dispositivo tenta capturar credenciais de rede eventualmente desprotegidas, seja interceptando tráfego, explorando vulnerabilidades conhecidas ou utilizando técnicas de engenharia social para coletar senhas.

Fase 5: Execução de Ataques Remotos

Uma vez que o dispositivo malicioso encontra-se conectado à rede, os atacantes podem então realizar ataques remotos, buscando por eventuais formas de acesso não autorizado a sistemas críticos, a execução de transações fraudulentas, o roubo de dados sensíveis ou a instalação de outros malwares para causar danos adicionais, incluindo o ataque de ransomware (Antonakakis et al. 2017).

A Figura 1.1, apresenta um esquema básico desse tipo de ataques:

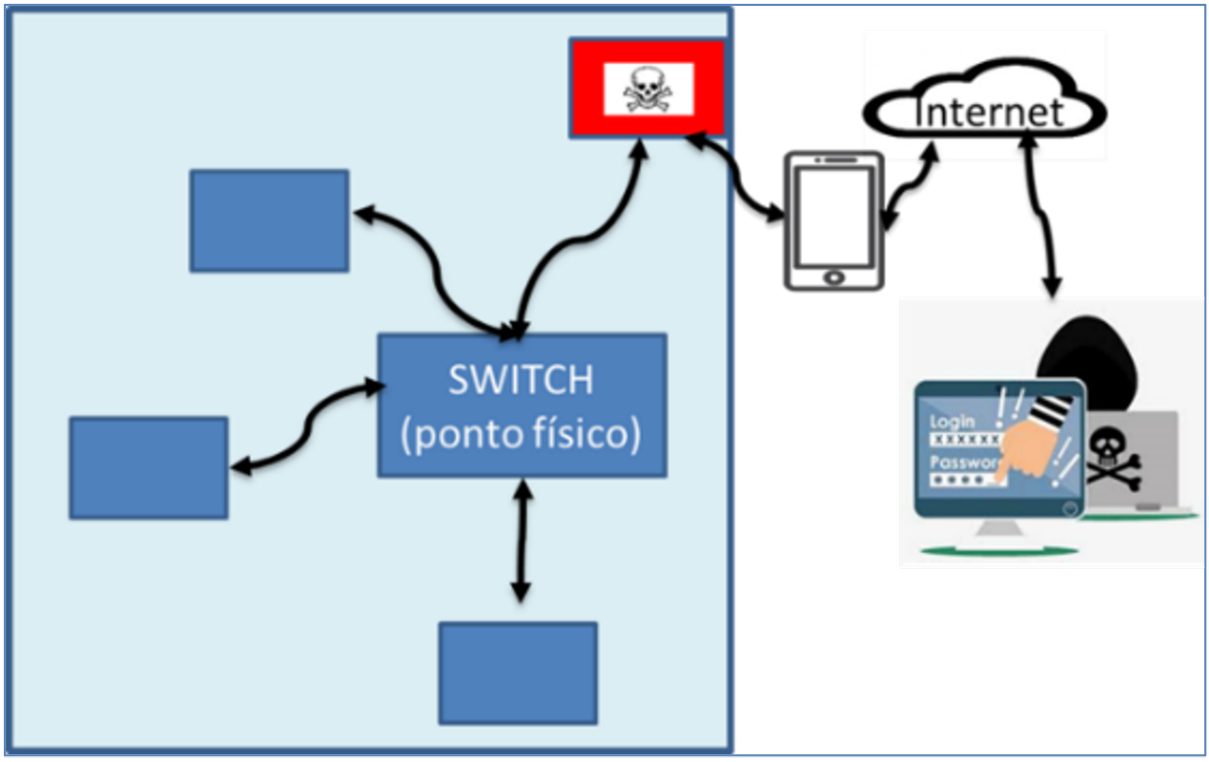


Figura 1.1: Representação do modelo de ataque, no qual um equipamento espúrio é conectado à rede.

Este tipo de ataque ilustra a complexidade dos desafios enfrentados pelas instituições financeiras na proteção de suas redes. A combinação de vulnerabilidades físicas, digitais e humanas requer uma abordagem de segurança multifacetada (Kumari e Jain 2023) que inclua:

- Vigilância Física: Monitoramento constante de todos os pontos de acesso físico à rede.
- Educação e Conscientização: Programas contínuos de treinamento em segurança cibernética para todos os funcionários, enfatizando a importância da vigilância e da proteção das credenciais de acesso.
- Segurança de Rede Robusta: Implementação de soluções de segurança avançadas, incluindo firewalls, sistemas de detecção e prevenção de intrusões, e análise de comportamento de rede para detectar atividades suspeitas.
- Resposta a Incidentes: Desenvolvimento de planos de resposta a incidentes eficazes que permitam uma ação rápida e coordenada em caso de detecção de um ataque. A rápida identificação e neutralização de dispositivos maliciosos conectados é crucial para limitar o impacto de tais intrusões.

Uma Instituiçãso Financeira brasileira cedeu imagens reais de uma tentativa de ataque detectada, mostrando alguns desses dispositivos conectados fisicamente a um ponto de rede de comunicação local,

As imagens mostram que os atacantes tem buscado a utilização de dispositivos *IoT* (Internet das coisas), seja pelo tamanho reduzido, capacidade de customização ou cujo comportamento se assemelhe a um *IoT*, para dificultar a detecção, conforme figuras 1.2, 1.3 e 1.4, a seguir.

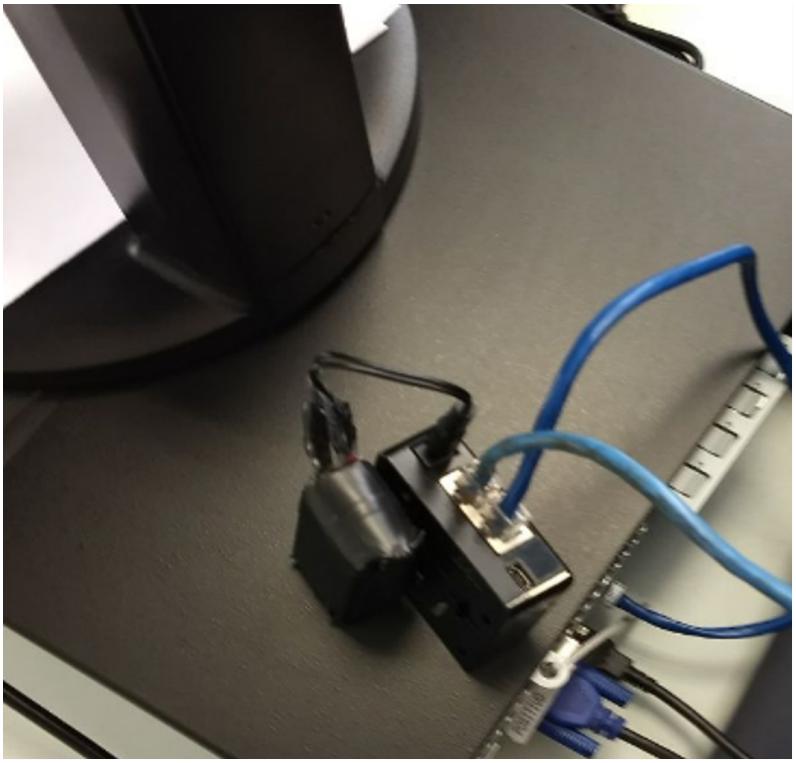


Figura 1.2: Dispositivo encontrado em uma tentativa de ataque real a uma empresa

Figura 1.3: Conexão física do dispositivo espúrio à rede corporativa

Figura 1.4: Outro micro dispositivo detectado.

Um dos principais mitigadores desse tipo de ataque é ter um controle sobre todos os dispositivos que são conectados a essa rede (MITRE 2021). No entanto, as tecnologias disponíveis exigem que sejam vencidos alguns obstáculos como orçamento (nem todas as empresas tem capacidade de investimento em *NAC*) ou complexidade (diferentes equipamentos no ecossistema daquela rede ou até mesmo dispositivos legados que não tem a capacidade de se conectar a uma solução de *NAC*).

Nesse contexto, este artigo propõe um modelo complementar para detectar e responder a conexões não autorizadas de dispositivos *IoT* na ausência do *NAC*.

O modelo utiliza técnicas combinadas de detecção e identificação de dispositivos, incluindo *fingerprints* de sistemas operacionais e técnicas ativas e passivas (Albanese, Battista e Jajodia 2015).

O modelo proposto neste artigo busca combinar, ao mesmo tempo, as técnicas passivas e ativas para detectar e responder a dispositivos não autorizados. A combinação dessas abordagens pode permitir uma detecção mais precisa, adaptável e eficiente, garantindo a segurança das redes. Ao mesmo tempo, buscou-se ferramentas e recursos que não fossem dependentes de orçamentos volumosos, ao mesmo tempo eficazes. Neste sentido, a impressão digital de dispositivos ou Device Fingerprinting foi a base para a proposta deste trabalho.

1.3 OBJETIVOS DO TRABALHO

1.3.1 Objetivo geral

O objetivo principal deste trabalho é propor e testar um modelo que seja capaz de detectar dispositivos, em especial *IoT*, espúrios em uma rede de comunicações.

1.3.2 Objetivos específicos

Para chegar no objetivo geral da proposta, foram elencados os seguintes objetivos específicos:

- Criar um ambiente virtualizado para a realização dos testes
- Propor um modelo para o desenvolvimento da solução proposta ;
- Testar os cenários e as ferramentas P0F e NMAP
- Detectar dispositivos com características de nãoo autorizados em ambiente controlado virtualizado.

1.4 TRABALHOS PUBLICADOS

Durante os estudos para esta dissertação, um artigo foi publicado abordando o tema de modelo para *fingerprints* de sistemas operacionais.

- IADIS International Journal on Computer Science and Information Systems, 2023

1.5 ORGANIZAÇÃO DO TRABALHO

O presente estudo é organizado em cinco capítulos, sendo este primeiro dedicado à introdução do tema e à contextualização do problema abordado.

Capítulo 2: Fundamentação Teórica

O segundo capítulo aborda a fundamentação teórica, tecnologias e técnicas relacionadas ao processo de fingerprint de dispositivos. Serão explorados conceitos fundamentais sobre identificação de dispositivos, incluindo métodos de fingerprinting, protocolos de rede relevantes e tecnologias de segurança empregadas nesse contexto.

Capítulo 3: Metodologia e Modelo Proposto

No terceiro capítulo, é apresentada a metodologia adotada para identificar e responder a conexões não autorizadas de dispositivos na rede. Além disso, será detalhado o modelo proposto para alcançar esse objetivo, destacando os conceitos aplicados e os desafios enfrentados durante sua elaboração e implementação.

Capítulo 4: Análise dos Resultados

O quarto capítulo consiste na análise dos resultados obtidos por meio da aplicação do modelo proposto. Serão examinados os dados coletados durante a execução da metodologia, avaliando a eficácia e a eficiência do modelo na detecção e resposta a conexões não autorizadas de dispositivos na rede.

Capítulo 5: Conclusão e Trabalhos Futuros

O quinto e último capítulo conclui o trabalho, fornecendo uma validação da proposta apresentada e discutindo os resultados obtidos. Além disso, serão indicadas direções para trabalhos futuros, destacando possíveis áreas de pesquisa e desenvolvimento que possam expandir e aprimorar o modelo proposto.

Esta estrutura organizacional visa fornecer uma visão abrangente do estudo realizado, apresentando a contextualização do problema, a fundamentação teórica, a metodologia adotada, a análise dos resultados e as conclusões alcançadas, além de abrir perspectivas para investigações futuras na área.

2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

2.1 TRABALHOS RELACIONADOS

A detecção e resposta a conexões não autorizadas de dispositivos têm sido amplamente exploradas em diversas pesquisas. Um estudo realizado por Albanese et al. (2015) propôs uma abordagem baseada em assinaturas para a detecção de dispositivos não autorizados. O método consiste em criar uma lista de assinaturas conhecidas de dispositivos legítimos e comparar essas assinaturas com os dispositivos presentes na rede. Embora essa abordagem tenha se mostrado eficaz em detectar dispositivos não autorizados, ela enfrenta limitações de escalabilidade devido à necessidade de atualizar constantemente as assinaturas à medida que novos dispositivos são introduzidos no mercado.

Outros pesquisadores, como Diro A. (2021) e De Caldas Filho et al. (2022) exploraram técnicas de aprendizado de máquina para a detecção de dispositivos. Eles aplicaram algoritmos de agrupamento para identificar padrões anormais de comportamento na rede que podem indicar a presença de dispositivos não autorizados. No entanto, essa abordagem requer grandes conjuntos de dados de treinamento e pode enfrentar desafios na lida com a variedade de dispositivos e suas características distintas.

Neste contexto existem estudos que exploram tanto técnicas passivas quanto ativas de impressão digital. Ferramentas como o P0F (Rumble, 2019) têm sido amplamente utilizadas para identificar o sistema operacional de dispositivos com base em seus comportamentos de rede. O P0F analisa padrões nos pacotes de dados, como o comportamento da pilha TCP/IP, para determinar o sistema operacional provável do dispositivo.

Além disso, o NMAP (Lyon, 2009), é outra ferramenta que continua sendo comumente empregada para realizar sondagens em dispositivos *IoT* não autorizados, examinando portas abertas, comportamento da pilha TCP/IP e outras características de rede (Lyon 2009).

Outra abordagem interessante foi apresentada por M.Bagaa et al. (2020) (Bagaa et al. 2020), que utilizaram técnicas de análise comportamental e de aprendizado de máquina para detectar dispositivos não autorizados. Eles desenvolveram um modelo que monitora o comportamento de dispositivos *IoT* em tempo real, identificando desvios em relação ao comportamento esperado e acionando alarmes em caso de atividades suspeitas.

Além disso, X. Yang et al. (2022) propuseram uma estrutura baseada em blockchain para verificar a identidade dos dispositivos *IoT* e garantir sua autenticidade. Essa tecnologia fornece um registro imutável e distribuído, aumentando a confiança e a segurança da rede (Yang et al. 2022).

Esses trabalhos fornecem uma visão abrangente das abordagens existentes para a detecção de dispositivos não autorizados em redes *IoT*. No entanto, muitos desses métodos enfrentam desafios, como a necessidade de atualizações constantes de assinaturas ou conjuntos de dados de treinamento extensos e, principalmente tempo de implementação.

O modelo proposto neste artigo busca superar essas limitações, utilizando *fingerprints* de sistemas operacionais e técnicas passivas e ativas de detecção para uma abordagem mais eficiente, flexível e rápida, adaptada tanto a redes simples quanto às mais complexas e com baixo custo e curto tempo de implementação.

Um dos desafios foi estabelecer uma proposta que não dependa totalmente da atualização de assinaturas externas, para isso, foi também proposta a utilização de uma base de assinaturas internas, ou seja, de padrões de dispositivos conhecidos na rede.

A proposta combina técnicas de detecção e identificação, por meio de *fingerprints* de sistemas operacionais e técnicas de detecção ativa e passiva de dispositivos.

Os *fingerprints* de sistemas operacionais (OS *fingerprints*) são características únicas que podem ser utilizadas para identificar o sistema operacional de um dispositivo conectado à rede. A análise dessas impressões digitais pode fornecer informações valiosas para identificar dispositivos não autorizados na rede (H. Jafari et al. 2018).

2.2 INTERNET DAS COISAS - IOT

A Internet das Coisas (*IoT*) é uma rede de objetos e dispositivos ("coisas") conectados, equipados com sensores (e outras tecnologias) que os habilitam a receber e transmitir dados – de e para outras coisas e sistemas.

A arquitetura da IOT ele preve a conexão dos disposistivos conectados, um modulo de conexão, (middware) e uma aplicação para demosntração dos resultdos uma representação simplificada do middleware de *IoT* é uma camada de software entre a camada física (ambiente dos dispositivos, coleta de dados, processamento local de dados, execução de comandos com ações sobre o ambiente externo à *IoT*, dentre outros.) e a camada de aplicação (ambiente de tratamento consolidado dos dados, inferências, decisões, preparação de comandos, tarefas administrativas e coordenação da segurança). Como elemento de intermediação, o middleware proporciona um conjunto de abstrações, de modo a facilitar a integração e comunicação entre componentes heterogêneos.

O middleware desempenha um papel fundamental no ecossistema da Internet das Coisas (IoT), servindo como uma camada intermediária que abstrai as complexidades dos sistemas locais ou do hardware dos dispositivos, conforme a figura 2.1. Essa abstração permite que os desenvolvedores de aplicativos concentrem seus esforços nas tarefas a serem resolvidas, sem se preocuparem com os detalhes de comunicação e as complexidades associadas aos dispositivos subjacentes (MENDONÇA 2019)

O principal objetivo do middleware é ocultar os detalhes tecnológicos dos objetos físicos (coisas), proporcionando uma interface consistente e simplificada para os desenvolvedores de aplicativos. Ao fazer isso, o middleware oferece uma série de serviços que facilitam o desenvolvimento e a integração de aplicativos na infraestrutura da IoT. Além disso, o middleware também visa fornecer funções críticas que são comuns a todas as aplicações, permitindo que essas funções sejam fatoradas e gerenciadas de forma mais eficiente.

Um dos desafios enfrentados na IoT é a grande quantidade e variabilidade dos dispositivos envolvidos.

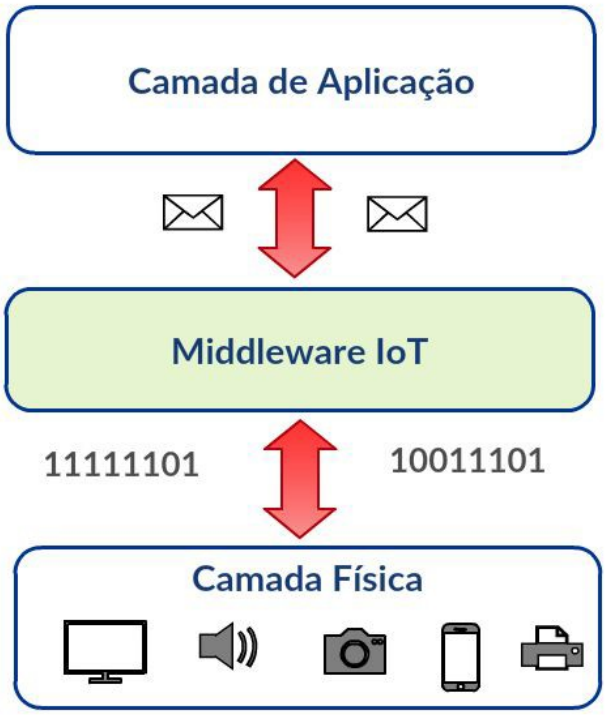


Figura 2.1: Middleware IoT - (Huacarpuma 2017)

É praticamente impossível para as aplicações terem conhecimento prévio de todos os possíveis dispositivos que podem participar da rede. Nesse contexto, o middleware desempenha um papel crucial ao fornecer mecanismos de descoberta e gerenciamento dinâmico de dispositivos, permitindo que novos dispositivos sejam facilmente integrados à rede sem a necessidade de modificação nos aplicativos existentes. Essa capacidade de adaptação e escalabilidade torna o middleware uma peça fundamental na infraestrutura da IoT, possibilitando o crescimento e a evolução contínua do ecossistema de dispositivos conectados (MENDONÇA 2019).

Em uma representação mais estruturada da instância de *IoT*, é necessário considerar a presença de múltiplos dispositivos, assim como múltiplas aplicações inclusive aquelas de auto sustentação e administração da instância de *IoT* (ambiente de tratamento consolidado dos dados, inferências, decisões, preparação de comandos, tarefas administrativas, coordenação da segurança etc.), além da possibilidade de o middleware de *IoT* ser acoplado e usar serviços de outros módulos associados de suporte, como por exemplo middleware de serviços de dados e módulos locais de serviços de segurança da *IoT*, conforme descritos respectivamente.

2.3 DEVICE FINGERPRINTING OU FINGERPRINTS

O device fingerprinting é uma técnica avançada de rastreamento online que identifica e monitora equipamentos conectados a uma rede de computadores por meio das características únicas de seus dispositivos. Essa prática, que capta informações como configurações de hardware, software ou Sistemas Operacionais,

pode ser utilizada para fins de segurança. Embora ofereça vantagens como a detecção de fraudes e a otimização da experiência do usuário, o fingerprinting também suscita preocupações significativas em relação à privacidade (Saraiva et al. 2014).

O conceito de device fingerprinting na rede expande o escopo tradicional do fingerprinting de dispositivos ao se concentrar na identificação e monitoramento de dispositivos através de suas interações com a rede.

Diferente do fingerprinting convencional, que analisa atributos do hardware e software do dispositivo individualmente, o fingerprinting de rede também se aprofunda nas características únicas apresentadas pelos dispositivos ao se comunicarem ou ao acessarem recursos de rede.

Fundamentos do Device Fingerprinting de Rede

O device fingerprinting de rede envolve a análise de padrões de tráfego, protocolos de comunicação, estilos de interação com a rede, e até mesmo os tipos de solicitações feitas aos servidores. Através da captura e análise desses dados, é possível identificar não apenas o tipo de dispositivo, mas também o sistema operacional, o navegador utilizado e, em casos mais avançados, até mesmo a identidade do usuário ou a aplicação específica em uso (Roy et al. 2022).

É um método para identificar um dispositivo utilizando uma combinação de atributos fornecidos pela configuração do dispositivo e de que forma será usado.

Técnicas e Metodologias (Albanese, Battista e Jajodia 2015)

Diversas técnicas podem ser empregadas no processo de fingerprinting de rede, incluindo:

Análise de Protocolos: Examina as especificidades nos protocolos de comunicação utilizados pelos dispositivos. Variações sutis na implementação de protocolos padrões podem revelar informações sobre o sistema operacional ou o software em uso (Albanese, Battista e Jajodia 2015).

Inspeção de Pacotes: A inspeção profunda de pacotes (DPI) permite analisar o conteúdo e o cabeçalho dos pacotes de dados transmitidos pela rede, oferecendo detalhes sobre as aplicações e serviços utilizados.

Comportamento de Rede: O estudo de padrões de comportamento, como horários de atividade, volume de tráfego, e destinos de comunicação, ajuda a distinguir entre tipos de usuários e dispositivos (Jafari et al. 2018).

2.4 O *NETWORK ADMISSION CONTROL* (*NAC*)

Uma tecnologia a se observar a respeito de segurança nas redes é a utilização do *NAC* na infraestrutura de rede para impor conformidade com políticas de segurança a todos os dispositivos que tentam acessar recursos de computação da rede. O *NAC* é projetado para permitir o acesso à rede apenas a dispositivos finais compatíveis e confiáveis, como PCs, servidores e PDAs, e pode restringir o acesso ou mesmo remediar dispositivos que não estejam em conformidade. (Helfrich et al. 2006)

O principal objetivo do *NAC* é proteger as redes contra ameaças de segurança emergentes, garantindo que apenas dispositivos seguros e autorizados possam acessar recursos da rede. Ele verifica a conformidade

dos dispositivos com as políticas de segurança da empresa antes de conceder acesso à rede, avaliando aspectos como a instalação de software antivírus, a aplicação de patches de segurança e a presença de softwares maliciosos.(Helfrich et al. 2006), (Press 2007)

Vantagens do *NAC* Melhora a Segurança da Rede: Ao garantir que apenas dispositivos em conformidade tenham acesso, o *NAC* reduz o risco de malware e ataques cibernéticos. Controle de Acesso Granular: Permite a aplicação de políticas de segurança diferenciadas com base no usuário, dispositivo ou função na rede. Detecção e Remediação de Ameaças: Identifica dispositivos não conformes e pode automatizar a correção de problemas de segurança antes de permitir o acesso. Visibilidade da Rede: Fornece um inventário detalhado dos dispositivos conectados à rede e seu estado de conformidade.

Desvantagens do *NAC* Complexidade de Implantação: A configuração e manutenção do *NAC* podem ser complexas, exigindo um planejamento cuidadoso e conhecimento técnico especializado. Custo: A implementação do *NAC* pode representar um investimento significativo em termos de hardware, software e recursos humanos. Possíveis Interrupções de Serviço: Se não configurado corretamente, o *NAC* pode bloquear inadvertidamente dispositivos legítimos, afetando a produtividade dos usuários. Manutenção Contínua: As políticas de segurança e a lista de dispositivos autorizados precisam ser atualizadas regularmente para manter a eficácia do *NAC*.

Em resumo, o *NAC* é uma ferramenta poderosa para melhorar a segurança das redes empresariais, oferecendo um controle rigoroso sobre quais dispositivos podem acessar a rede e garantindo que eles estejam em conformidade com as políticas de segurança estabelecidas. No entanto, sua eficácia depende de uma implementação cuidadosa e de uma gestão contínua para equilibrar segurança e acessibilidade.

Por outro lado, especialistas discutem a evolução da *NAC* e sua relevância para as empresas atualmente. Embora a *NAC* tenha sido fundamental para lidar com ameaças do tipo traga seu próprio dispositivo (BYOD) e garantir a integridade da rede, ainda há dúvidas sobre sua prontidão para todas as empresas (Dildy).

Para obter uma compreensão mais abrangente da implementação do *NAC* e de seu impacto na segurança corporativa, explorar essas análises detalhadas e estudos de caso pode fornecer informações valiosas sobre seus recursos e limitações.

A evolução constante das redes e a crescente integração de dispositivos da Internet das Coisas (*IoT*) exigem estratégias de segurança robustas e adaptáveis. Neste contexto, a identificação de dispositivos *IoT* na rede é um desafio significativo que necessita de atenção em pesquisas e trabalhos futuros. Uma das abordagens promissoras para abordar essa questão é a adoção do conceito de *Zero Trust* Network Access (*ZTNA*) (**??**).

O *ZTNA* é um modelo de segurança baseado no princípio de "nunca confiar, sempre verificar". Diferente das abordagens tradicionais de segurança, que assumem que tudo dentro da rede é confiável, o *ZTNA* trata todos os usuários e dispositivos como potencialmente hostis, exigindo verificação rigorosa antes de conceder acesso a recursos da rede. Esse conceito é particularmente relevante no cenário atual, onde a mobilidade dos usuários e a proliferação de dispositivos *IoT* expandem as fronteiras tradicionais da rede, introduzindo novas vulnerabilidades (Landry e Koger 2023).

A relação entre *ZTNA* e *Network Admission Control* (*NAC*) é complementar. Enquanto o *NAC* se

concentra na conformidade e na verificação de dispositivos ao tentar acessar a rede, garantindo que apenas dispositivos autorizados e em conformidade possam se conectar, o *ZTNA* vai além, aplicando políticas de acesso dinâmico baseadas na identidade do usuário, no contexto do dispositivo e na sensibilidade dos recursos acessados. Juntos, *NAC* e *ZTNA* fornecem uma abordagem de segurança em camadas que não apenas identifica e controla o acesso dos dispositivos *IoT*, mas também adapta os privilégios de acesso com base em avaliações contínuas de risco.

2.5 DETECÇÃO PASSIVA DE DISPOSITIVOS

A detecção passiva de dispositivos é uma abordagem que não gera tráfego adicional na rede, concentrando-se na análise do tráfego existente para identificar dispositivos e suas características. Uma abordagem comum nesse contexto é a análise de impressões digitais (fingerprinting) de dispositivos. O P0f é uma ferramenta de detecção passiva amplamente reconhecida, capaz de identificar sistemas operacionais e aplicativos com base em características passivas do tráfego de rede. O P0f analisa parâmetros como tamanhos de janela TCP, opções TCP/IP e TTL para inferir informações sobre os dispositivos na rede sem interromper o tráfego normal.

P0F é uma ferramenta sofisticada e avançada que se destaca no campo da segurança cibernética por sua habilidade em realizar a identificação passiva de sistemas operacionais (SO) em dispositivos conectados a uma rede. Desenvolvida por Michal Zalewski, *P0F* analisa os pacotes de rede de forma passiva, sem a necessidade de enviar pacotes adicionais para o dispositivo alvo. Esta abordagem minimiza o risco de detecção, tornando o *P0F* uma ferramenta valiosa para administradores de rede e profissionais de segurança que buscam monitorar e proteger suas infraestruturas de TI sem interferir no tráfego de rede normal.

Dentre as ferramentas e técnicas disponíveis para a obtenção e análise de *fingerprints* de sistemas operacionais, o *P0F* uma das mais conhecidas no campo da segurança cibernética. Desenvolvido e aprimorado ao longo dos anos, o *P0F*, conforme documentado por Rumble (2019), representa um marco na análise passiva de tráfego de rede, oferecendo aos profissionais de segurança uma ferramenta poderosa para a identificação de sistemas operacionais sem a necessidade de interações ativas ou invasivas.

O P0f utiliza um conjunto de técnicas para identificar dispositivos de forma não intrusiva:

Análise de Tamanhos de Janela TCP: O P0f examina os tamanhos de janela TCP nas conexões estabelecidas para determinar características específicas do sistema operacional. Diferentes sistemas operacionais implementam políticas de escalonamento de janelas TCP de maneiras distintas, permitindo que o P0f faça inferências precisas sobre o sistema operacional do dispositivo.

Análise de Opções TCP/IP: O P0f analisa as opções TCP/IP nos pacotes de rede para identificar características exclusivas de sistemas operacionais específicos. Essas opções incluem parâmetros como o tempo de vida (TTL) e as opções TCP MSS (Maximum Segment Size), que podem ser utilizadas para determinar o sistema operacional do dispositivo.

Análise de TTL: O P0f examina o TTL nos pacotes de rede para determinar a distância entre o dispositivo de origem e o destino. Com base nessa informação, o P0f pode inferir o sistema operacional do

dispositivo e sua localização na rede.

Essas técnicas permitem ao P0f identificar dispositivos na rede de forma precisa e eficiente, fornecendo informações valiosas sobre sistemas operacionais e aplicativos em uso.

2.5.1 *P0F*

O *P0F* utiliza técnicas passivas de análise de pacotes, o que significa que ele é capaz de identificar características dos sistemas operacionais simplesmente observando o tráfego de rede, sem a necessidade de enviar pacotes adicionais ou realizar qualquer forma de scanning ativo. Isso é particularmente valioso em ambientes onde a minimização da intrusividade é crítica, como em redes corporativas ou infraestruturas críticas, onde atividades de scanning ativo podem ser consideradas intrusivas ou até mesmo hostis.

A capacidade do *P0F* de determinar o sistema operacional de um dispositivo na rede baseia-se em uma análise detalhada de várias características do tráfego, incluindo, mas não se limitando a, o comportamento da pilha TCP/IP, o tamanho dos pacotes, tempos de vida (TTL), opções de TCP, e peculiaridades específicas na formação de pacotes. Esses elementos, quando combinados, oferecem uma assinatura única que pode ser associada a sistemas operacionais específicos e até suas versões.

Por meio de uma extensa base de dados que contém padrões de tráfego associados a diferentes sistemas operacionais, o *P0F* compara as características observadas no tráfego de rede com as assinaturas conhecidas. Isso permite não apenas a identificação do sistema operacional mas também fornece insights sobre configurações específicas, versões do sistema, e até mesmo possíveis patches de segurança aplicados. Esta capacidade de discernir entre diversas versões do mesmo sistema operacional é de imenso valor para a análise de segurança, permitindo uma avaliação mais precisa da postura de segurança de dispositivos na rede.

Além de sua aplicação em segurança cibernética, o *P0F* também tem sido utilizado em tarefas de administração de sistemas para monitoramento de rede, detecção de configurações anormais, e suporte na implementação de políticas de segurança adaptativas. Sua natureza passiva e a riqueza de informações que pode extrair do tráfego de rede o tornam uma ferramenta indispensável para profissionais de TI e segurança cibernética, facilitando a detecção precoce de dispositivos potencialmente vulneráveis ou mal configurados, bem como a identificação de atividades suspeitas que possam indicar tentativas de intrusão ou comprometimento de rede.

A contribuição de Rumble (2019) ao desenvolvimento e documentação do *P0F* reforça a importância de ferramentas sofisticadas de detecção passiva na construção de defesas cibernéticas robustas. Ao capacitar os profissionais de segurança com meios para identificar proativamente possíveis ameaças baseadas na análise de tráfego de rede, o *P0F* desempenha um papel crucial na manutenção da segurança e integridade de sistemas operacionais em ambientes de rede cada vez mais complexos e heterogêneos.

2.5.2 Capacidades técnicas do *P0F*

P0F utiliza uma abordagem sofisticada para identificar sistemas operacionais, versões, e, em alguns casos, configurações específicas de dispositivos na rede. A ferramenta analisa os detalhes contidos nos

cabeçalhos dos pacotes TCP/IP, que incluem, mas não se limitam a:

Comportamento da Pilha TCP/IP: *P0F* examina como diferentes sistemas operacionais respondem a vários cenários de rede, incluindo a forma como iniciam conexões (por exemplo, a sequência de flags TCP), como respondem a pacotes inesperados ou malformados, e como encerram conexões. Essas respostas são distintas entre diferentes famílias de sistemas operacionais e até entre versões de um mesmo sistema operacional.

Tamanho de Pacotes e Opções TCP: O tamanho do pacote e as opções selecionadas no cabeçalho TCP, como o tamanho da janela de recepção e o uso de opções como NOP (No Operation) e MSS (Maximum Segment Size), podem indicar certos sistemas operacionais. *P0F* mantém uma base de dados de assinaturas derivadas desses padrões para correlacionar as observações com sistemas operacionais específicos.

Outros Parâmetros: *P0F* também considera parâmetros como o TTL (Time To Live) inicial, a presença e o ordenamento de certas opções TCP, e até mesmo peculiaridades na forma como alguns sistemas operacionais fragmentam pacotes IP. O TTL (Time to Live) é um valor presente em pacotes de rede que é utilizado para evitar a circulação infinita de pacotes na rede. Ele é decrementado a cada roteador que o pacote passa, e quando chega a zero, o pacote é descartado. O valor inicial do TTL é definido pelo sistema operacional que envia o pacote, e pode ser usado para identificar qual sistema operacional está enviando o pacote.

Alguns exemplos de valores padrão de TTL para SO são: Windows: 128 Linux: 64 macOS: 64 iOS: 64 Android: 64 Cisco IOS: 255

Esses valores podem ser alterados pelo administrador de rede ou pelo sistema operacional, ou pelo atacante como técnica de evasão. Ainda assim, eles podem ser uma referência sobre qual sistema operacional está sendo usado.

2.5.3 Aplicação e relevância

Ao identificar o sistema operacional de um dispositivo, o *P0F* pode ser utilizado para diversos propósitos de segurança e administração de redes, incluindo:

Detecção de Anomalias: Identificar dispositivos que não deveriam estar presentes em uma rede específica. Por exemplo, a presença de um sistema operacional tipicamente usado por dispositivos *IoT* em uma rede corporativa onde tal dispositivo não é esperado.

Avaliação de Segurança: Avaliar o perfil de segurança de dispositivos na rede identificando sistemas operacionais desatualizados ou vulneráveis.

Mitigação de Ameaças: Auxiliar na resposta a incidentes de segurança ao fornecer informações detalhadas sobre os sistemas operacionais envolvidos em atividades suspeitas.

2.5.4 Considerações técnicas

A eficácia do *P0F* depende significativamente da precisão e da atualização de sua base de dados de assinaturas. A diversidade de dispositivos e a rápida evolução dos sistemas operacionais exigem atualiza-

ções regulares para manter a relevância da ferramenta. Além disso, técnicas avançadas de evasão podem ser utilizadas por atacantes para mascarar ou alterar características dos pacotes, desafiando a capacidade do *P0F* de identificar corretamente os sistemas operacionais.

2.6 DETECÇÃO ATIVA DE DISPOSITIVOS

A detecção ativa de dispositivos é um processo que envolve a geração de tráfego na rede por meio do envio de pacotes de sondagem para determinar a presença e as características dos dispositivos na rede. Uma das ferramentas mais proeminentes e versáteis para essa finalidade é o Nmap (Network Mapper), um utilitário de código aberto amplamente utilizado na comunidade de segurança da informação. O Nmap oferece uma variedade de técnicas de varredura, incluindo TCP SYN scan, TCP connect scan e ICMP echo request.

TCP SYN Scan: Esta técnica envia pacotes SYN para os hosts na rede e analisa as respostas para determinar quais hosts estão ativos e quais portas estão abertas. O Nmap cria um handshake TCP parcial, não concluindo a conexão, o que torna a técnica mais sigilosa em comparação com outros métodos.

TCP Connect Scan: Nesta abordagem, o Nmap estabelece uma conexão TCP completa com os hosts alvo, tentando abrir uma conexão em cada porta. Isso permite determinar quais portas estão abertas, fechadas ou filtradas.

ICMP Echo Request: Esta técnica envia pacotes ICMP echo request (ping) para os hosts na rede e analisa as respostas para determinar se os hosts estão vivos. Embora não forneça informações detalhadas sobre portas abertas, é útil para identificar hosts ativos na rede.

Essas técnicas permitem identificar hosts ativos, serviços em execução e até mesmo sistemas operacionais subjacentes em uma rede, fornecendo uma visão abrangente da topologia e da segurança da rede.

2.6.1 *NMAP Port Scan*

Além disso, abordagens ativas também podem ser utilizadas para a obtenção de impressões digitais de sistemas operacionais. O *NMAP* (Lyon, 2009) é uma ferramenta popular que realiza sondagens em dispositivos para obter informações sobre suas características e identificar o sistema operacional. O *NMAP* envia pacotes específicos para o dispositivo-alvo e analisa as respostas recebidas para determinar o sistema operacional e outras informações relevantes.

2.6.2 Funcionalidades Técnicas e Aplicações do *NMAP*

O *NMAP* executa a descoberta de hosts e serviços através do envio de pacotes específicos para dispositivos-alvo, utilizando uma variedade de técnicas para sondar portas, identificar serviços rodando e determinar versões de software. As respostas recebidas são analisadas para inferir detalhes sobre os sistemas operacionais dos dispositivos, configurando uma poderosa ferramenta para administradores de rede e profissionais de segurança.

Descoberta de Hosts: O *NMAP* pode ser usado para identificar dispositivos ativos em uma rede, utilizando técnicas como ping sweep para mapear rapidamente os hosts. O "ping sweep"(Fyodor 2009)é uma técnica utilizada para descobrir quais endereços IP estão ativos em uma determinada rede. No contexto do Nmap (Network Mapper), o ping sweep é uma funcionalidade que permite verificar a disponibilidade dos hosts em uma rede através do envio de pacotes de ping (ICMP Echo Request) para múltiplos endereços IP.

Sondagem de Portas: Fundamental na identificação de portas abertas, o *NMAP* permite aos usuários entender quais serviços estão expostos em um dispositivo, fornecendo insights sobre possíveis vulnerabilidades.

Detecção de Versão de Serviço: Através da análise de respostas a sondagens específicas, o *NMAP* pode estimar qual software e qual versão estão rodando em portas abertas, permitindo a identificação de serviços potencialmente vulneráveis.

Identificação de Sistema Operacional: Utilizando uma técnica conhecida como fingerprinting de pilha TCP/IP, o *NMAP* compara as características das respostas do dispositivo com uma base de dados de assinaturas conhecidas para determinar o sistema operacional.

Scripts de *NMAP* (Nmap Scripting Engine - NSE): Uma das funcionalidades mais poderosas do *NMAP* é o NSE, que permite aos usuários executar scripts para automatizar uma ampla gama de tarefas de monitoramento e análise de segurança, incluindo a detecção de vulnerabilidades, a auditoria de configurações de segurança e a descoberta de informações adicionais sobre os hosts e serviços.

2.6.3 Aplicação e relevância do *NMAP*

Ao utilizar o *NMAP*, é crucial considerar aspectos éticos e legais, pois a sondagem ativa de redes pode ser interpretada como intrusiva por alguns hosts ou redes. Além disso, o *NMAP* deve ser usado com cuidado em ambientes de produção, pois algumas das técnicas de sondagem podem impactar o desempenho dos dispositivos-alvo ou mesmo resultar em interrupções para serviços sensíveis.

Por esses motivos, o *NMAP* é uma ferramenta indispensável para a segurança cibernética, fornecendo dados críticos que ajudam na proteção de redes contra acesso não autorizado e ataques cibernéticos. A capacidade do *NMAP* de realizar uma análise detalhada e multifacetada de redes faz dele uma escolha primária para profissionais de segurança que buscam uma compreensão profunda da postura de segurança de suas infraestruturas de TI. Sua aplicação na identificação de sistemas operacionais é apenas uma faceta de seu amplo espectro de utilidades, demonstrando sua importância fundamental na análise de segurança de redes modernas, cujas funcionalidades encontram-se descritas na tabela 2.1, a seguir:

No entanto, Albanese et al. (2015) apresenta estratégias avançadas para ocultar informações sobre sistemas operacionais de ferramentas de fingerprinting, como o *NMAP*, essenciais tanto para profissionais de segurança quanto para indivíduos mal-intencionados. São abordadas várias técnicas e soluções específicas de sistemas operacionais, como módulos de kernel para Linux e patches para BSD, que podem modificar o comportamento da pilha TCP/IP para simular diferentes sistemas operacionais.

O objetivo é dificultar que atacantes identifiquem e explorem vulnerabilidades específicas do sistema operacional. Contudo, ressalta-se a importância de manter um ambiente de segurança robusto além

Funcionalidade	Descrição
Port Scanning	Detecta quais portas estão abertas em um host.
OS Detection	Identifica o sistema operacional do host alvo.
Service Version Detection	Determina quais serviços estão sendo executados em cada porta aberta.
Scripting Engine	Permite a execução de scripts personalizados para automação e extensão de funcionalidades.
Host Discovery	Encontra hosts ativos na rede.
Ping Sweeping	Detecta hosts ativos usando ping ICMP, TCP ou UDP.
Output Formatting	Oferece várias opções para formatação de saída, incluindo texto simples, XML e formato grepable.
Scan Types	Suporta uma variedade de tipos de varredura, incluindo TCP SYN, TCP Connect, UDP e SCTP.
Timing Options	Permite ajustar o tempo de espera, o intervalo entre pacotes e outras configurações relacionadas ao tempo.

Tabela 2.1: Principais funcionalidades do Nmap

de simplesmente ocultar informações do sistema operacional, sublinhando que a segurança por obscuridade não é uma estratégia suficiente por si só. A discussão inclui o impacto da revelação do tipo de sistema operacional em potenciais ataques e a importância da privacidade e da imagem da empresa (Albanese, Battista e Jajodia 2015).

2.7 COMPARATIVO DAS ABORDAGENS

Tabela 2.2: Comparativo entre a abordagem ativa e passiva

Característica	Abordagem Ativa	Abordagem Passiva
Método de Detecção	Envia sondas ou pacotes para identificar dispositivos.	Analisa o tráfego de rede existente para identificar dispositivos.
Intrusividade	Intrusiva: pode gerar tráfego adicional na rede.	Não intrusiva: não gera tráfego adicional.
Visibilidade	Detecta dispositivos que podem estar ocultos ou não respondem a sondas.	Detecta dispositivos que estão ativos e enviando tráfego na rede.
Detecção de Dispositivos Ocultos	Efetiva para identificar dispositivos que não respondem a sondas.	Limitada para detectar dispositivos ocultos.
Segurança	Pode ser interpretada como um ataque por sistemas de detecção de intrusão.	Geralmente não é interpretada como um ataque.
Impacto na Rede	Pode gerar congestionamento de rede e afetar o desempenho.	Não afeta o desempenho da rede.
Identificação de Serviços	Pode identificar serviços em dispositivos ativos.	Não é eficaz na identificação de serviços em dispositivos ocultos.

2.8 JUSTIFICATIVA PARA A ESCOLHA DAS FERRAMENTAS

Para este estudo, a escolha das ferramentas, P0f e Nmap, baseia-se nas características apresentadas tanto na seção de detecção ativa quando passiva, descritas neste estudo e sitetizadas na Tabela 2.2. Também foram consideradas a aceitação dessas ferramentas na comunidade de segurança da informação (Roy et al. 2022). Estas ferramentas são frequentemente citadas na literatura especializada como referências essenciais para atividades de detecção e exploração de redes.

3 METODOLOGIA

Este capítulo aborda a metodologia utilizada na realização do trabalho e detalha as características principais como delimitação do tema, tipo de investigação e o limite do estudo. São expostas, também, as etapas de realização do trabalho, bem como os dispositivos e ferramentas necessários, os cenários de simulação.

3.1 DELIMITAÇÃO DO TEMA

Considerando os tipos de conexão à rede, o presente trabalho trata das conexões cabeadas. Este trabalho demonstra formas de avaliar *fingerprints* de sistemas operacionais para detecção de dispositivos conectados a essas redes cabeadas.

3.2 TIPO DE INVESTIGAÇÃO

A pesquisa será metodológica e aplicada. com o tratamento dos cenários domésticos e de laboratório, por meio da execução de procedimentos e simulações. Quanto aos meios, a pesquisa será de laboratório e bibliográfica.

3.3 COLETA E TRATAMENTO DE DADOS

A coleta de dados será realizada por pesquisa bibliográfica e coleta de evidências de casos reais em ambiente de empresa do setor financeiro.

3.4 CENÁRIOS MITTRE

Para alcançar os objetivos desta dissertação, na fase inicial foi realizada a identificação e caracterização do problema, além do levantamento da hipótese da solução com um conjunto de *softwares*, em seguida a verificação das ferramentas que possibilitassem a virtualização dos cenários e os testes. Testes estes que foram realizados tanto em cenários reais, com máquinas físicas, como em cenários virtualizados.

Foram também elaborados cenários físicos e virtuais, criação ou adaptação de scripts e conexões de dispositivos com as mesmas características dos utilizados em ataques reais. Análise dos resultados obtidos e validação da arquitetura proposta.

Neste trabalho iremos realizar um estudo de uma proposta de identificação e resposta a tentativas de

invasão por técnicas de conexão física de dispositivos espúrios IoT em redes de comunicação. Como premissa, a rede não dispõe de ferramenta de controle de acesso à rede (NAC).

3.4.1 Catálogo de dispositivos

Um script de teste, constante na seção Apendice deste trabalho foi implementado para viabilizar a detecção de dispositivos. Por meio de uma combinação de varreduras de rede, uso de protocolos padrão como SNMP, e armazenamento estruturado de dados, o script fornece uma solução robusta para a detecção e catalogação de dispositivos, contribuindo significativamente para a gestão de segurança de redes.

O codigo, escrito na linguagem Python, é um script avançado para detecção de dispositivos em uma rede, incluindo impressoras e switches, utilizando varreduras de rede e consultas SNMP. Inicializa registrando exceções não tratadas no syslog, suporta modo debug e processamento de argumentos de linha de comando para funções específicas. Define duas classes principais, HostSearch para detecção de hosts e Printers/Switches para identificar dispositivos específicos. HostSearch emprega fping para identificar hosts ativos e arping para capturar endereços MAC. Printers e Switches usam snmpget para coletar informações detalhadas sobre os dispositivos, como modelo e fornecedor, salvando os resultados em arquivos CSV. O script é configurável para escanear faixas de IP específicas e registrar suas descobertas, otimizando a gestão de rede e a segurança.

3.4.2 Descrição da Proposta

A arquitetura combina fingerprints de sistemas operacionais e técnicas passiva e ativa para uma detecção mais precisa e adaptável de dispositivos não autorizados em redes sem a implementação do NAC. Essa abordagem permite a identificação precoce de dispositivos não autorizados.

A detecção e resposta a dispositivos não autorizados exige uma compreensão detalhada das características dos dados coletados.

3.4.3 Dados de tráfego de rede e demais características

Os dados de tráfego de rede contêm informações sobre a comunicação entre os dispositivos e a infraestrutura de rede. Esses dados podem ser coletados em diferentes pontos da rede, como switches, roteadores ou sensores de rede distribuídos.

Além disso, é importante considerar a diversidade de dispositivos presentes na rede. Esses dispositivos podem ter diferentes protocolos de comunicação, taxas de transmissão, padrões de tráfego e requisitos de largura de banda. Portanto, é necessário realizar uma análise abrangente e adaptável dos dados de tráfego, levando em consideração as características específicas de cada tipo de dispositivo (X. Yang et al., 2022)(Yang et al. 2022).

Outra característica relevante é a velocidade dos dados. Em redes IoT, a comunicação entre os dispositivos pode ocorrer em tempo real, exigindo uma análise rápida e eficiente dos dados de tráfego.

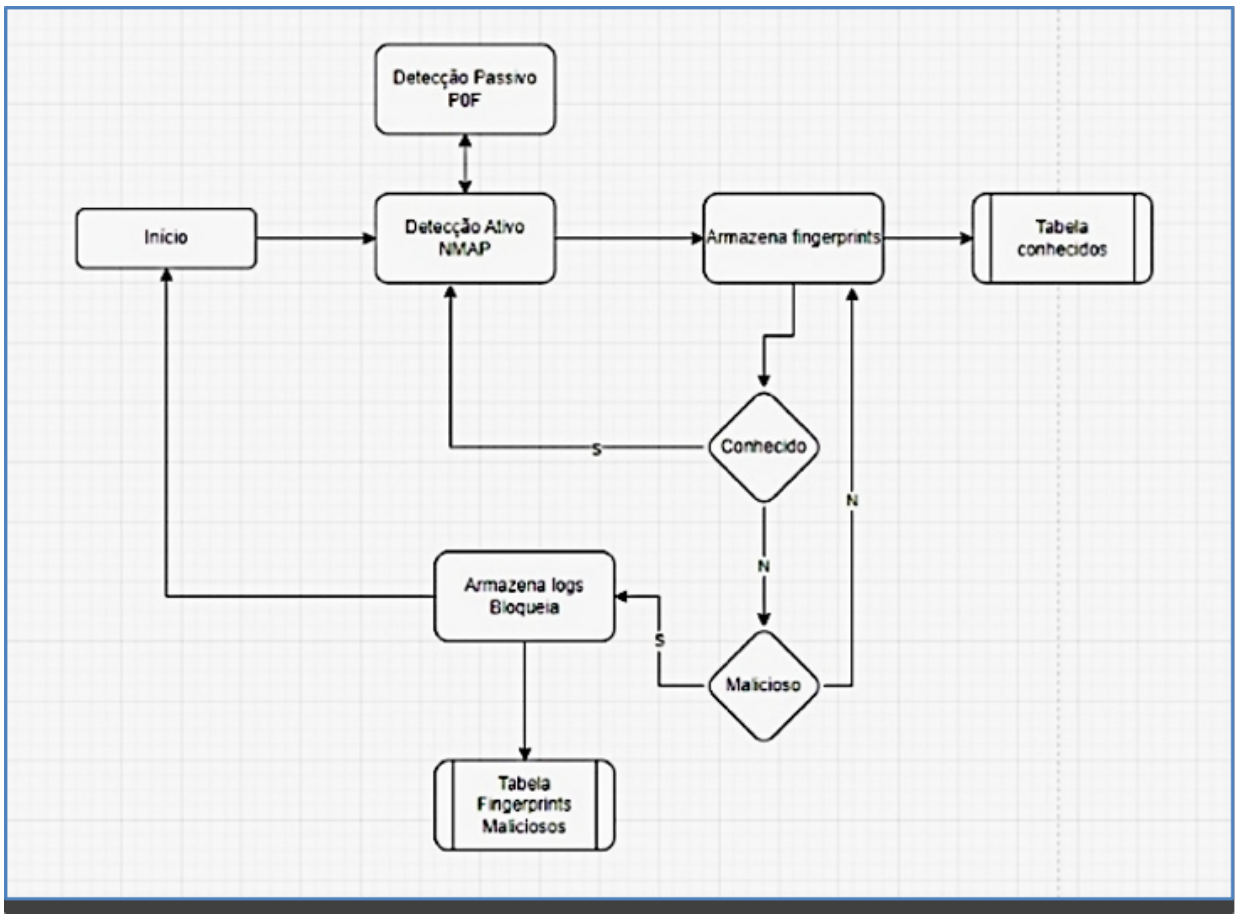


Figura 3.1: Arquitetura proposta

Os dados de comportamento dos dispositivos fornecem informações cruciais para a detecção de dispositivos não autorizados. Esses dados podem incluir informações sobre padrões de comunicação, tempos de resposta, volumes de dados transmitidos e outros indicadores de comportamento.

É essencial estabelecer um perfil de comportamento normal para cada dispositivo IoT na rede. Esse perfil é baseado nas atividades regulares do dispositivo, como horários de comunicação, protocolos de rede utilizados e volumes de dados transmitidos. Qualquer desvio significativo desse perfil pode indicar a presença de um dispositivo não autorizado (Diro et al. 2021). Neste estudo o tratamento de falsos positivos foi realizado por meio de observação deireta não automatizada.

Outra característica importante é a contextualização dos dados de comportamento. É essencial considerar o contexto em que os dispositivos operam, como o ambiente físico, a finalidade do dispositivo e as interações com outros dispositivos (Soares Francisco Lopes de Caldas Filho 2022).

Para a verificação da proposta, figura 3.1, considerou-se que, numa rede sem NAC, um dispositivo havia sido fisicamente conectado pelo atacante, utilizando a técnica descrita.

O diagrama contido na figura 3.1 representa um modelo de detecção de dispositivos com base em fingerprints de sistema operacional, utilizando as ferramentas P0F e NMAP. Cada fase está descrita a seguir:

Início: O ponto de partida do processo de detecção.

Detecção Passiva - P0F: Esta etapa envolve a utilização da ferramenta P0F, que realiza uma detecção passiva de dispositivos ao monitorar tráfego de rede e identificar sistemas operacionais sem enviar pacotes ativos.

Detecção Ativa - NMAP: Em paralelo à detecção passiva, o NMAP é utilizado para realizar uma varredura ativa, utilizando técnicas como o envio de pacotes TCP/IP para identificar dispositivos e seus sistemas operacionais.

Armazena fingerprints: Os fingerprints, ou características identificadoras dos sistemas operacionais detectados pelo P0F e NMAP, são armazenados para análise.

Tabela conhecidos: Uma tabela pré-existente de fingerprints conhecidos é consultada para determinar se o dispositivo identificado é reconhecido ou não.

Conhecido?: Um ponto de decisão que avalia se o fingerprint armazenado corresponde a um dispositivo conhecido. Se sim (S), o processo segue para armazenar os logs; se não (N), segue para a detecção de um possível dispositivo malicioso.

Armazena logs Bloqueia: Se o dispositivo for conhecido, os logs são armazenados e o acesso do dispositivo é provavelmente permitido ou simplesmente registrado.

Malicioso: Se o dispositivo não for conhecido, ele é então tratado como potencialmente malicioso.

Tabela Fingerprints Maliciosos: Os fingerprints considerados maliciosos são armazenados em uma tabela específica para referência futura e ações de segurança apropriadas, como bloqueio ou alerta.

Um aspecto fundamental na construção de um sistema de detecção e resposta eficaz contra dispositivos não autorizados é compreender as estratégias e táticas empregadas pelos adversários cibernéticos. Como premissa para o teste, foi utilizado um modelo do adversário com base na sub-técnica T1200 do Mitre ATTCK Framework, que aborda o comportamento de "Contornar Defesas"especificamente na área de "Hardware Additions"(Mittre, 2021a) (MITRE 2021).

Essa tática permite que os adversários ganhem acesso não autorizado à rede e executem atividades maliciosas sem serem detectados.

3.5 MODELO ADVERSARIAL - MITRE ATT&CK

O MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) é um recurso compreensivo que cataloga táticas e técnicas baseadas em observações reais de incidentes cibernéticos. Desenvolvido pela organização sem fins lucrativos MITRE Corporation, o framework foi lançado com o objetivo de fornecer uma linguagem comum para a comunidade de segurança cibernética discutir, documentar e combater ameaças. Desde sua concepção, tornou-se um padrão de fato na indústria para a análise de ameaças, avaliação de segurança e planejamento de defesa.

O framework é utilizado por defensores para melhor entender as ameaças, identificar pontos fracos em suas defesas e orientar o desenvolvimento de estratégias mais robustas de segurança. Para atacantes, serve como um compêndio de métodos eficazes, enquanto educadores e pesquisadores o usam como uma fonte

rica de estudos de caso e análises de tendências.

Dentro do framework, a técnica T1200 (MITRE 2021) é categorizada sob a tática de "Contornar Defesas"e especificamente aborda o uso de "Adições de Hardware"por adversários. Essa técnica descreve um vetor de ataque onde dispositivos físicos são introduzidos intencionalmente em ambientes para facilitar ou conduzir atividades maliciosas. Tais dispositivos podem variar desde simples pendrives USB que contêm malware até dispositivos mais complexos projetados para interceptar ou alterar dados em trânsito.

Exemplos de implementação desta técnica incluem: Keyloggers físicos que capturam e transmitem cada tecla pressionada por um usuário. Dispositivos USB maliciosos que, quando conectados, executam automaticamente código prejudicial. Implantes em hardware de rede, como roteadores ou switches modificados, que podem ser usados para criar backdoors permanentes em redes.

A detecção de adições de hardware maliciosas pode ser desafiadora, exigindo uma combinação de medidas de segurança física e digital. Isso inclui a implementação de políticas rigorosas de controle de acesso físico, monitoramento de ativos de rede e a utilização de soluções de segurança que possam identificar comportamentos anômalos indicativos de dispositivos comprometidos.

O entendimento profundo do MITRE ATTCK Framework e de técnicas específicas como a T1200 é crucial para o desenvolvimento de uma estratégia de defesa cibernética holística. Por meio da análise detalhada de táticas e técnicas usadas por adversários, organizações podem aprimorar suas capacidades de prevenção, detecção e resposta a incidentes, fortalecendo sua postura de segurança contra uma variedade de vetores de ataque. A técnica T1200, em particular, destaca a importância de não subestimar a segurança física na proteção contra ameaças cibernéticas.

Tabela 3.1: Subtécnica T1200 Mittre - (MITRE 2021)

ID	Nome	Descrição da técnica
G0105	DarkVishnya	*DarkVishnya*, *Raspberry Pi*, *netbooks* ou *laptops* de baixo custo para conectar fisicamente à rede local.

ID	Mitigação	Descrição da técnica
M1035	*Limite o acesso a recursos na rede*	*Estabeleça políticas de controle de acesso à rede, como o uso de certificados de dispositivos e o padrão 802.1x. Restrinja o uso de DHCP a dispositivos registrados para evitar que dispositivos não registrados se comuniquem com sistemas confiáveis.*
M1034	*Limite a instalação de hardware*	*Bloqueie dispositivos e acessórios desconhecidos por meio de configuração de segurança de endpoint e agente de monitoramento.*

3.6 INFRAESTRUTURA PARA REALIZAÇÃO DO TESTE

Para a realização dos testes foi preparada uma infraestrutura (Figura 3.2) que simula uma rede corporativa simples, constituída de uma rede virtual, com servidor Dell R920 e Proxmox como hypervisor.

Foram configuradas VMs na arquitetura amd64, na seguinte configuração: uma VM com pfSense para

atuar como roteador e máquinas Windows e Linux.

Para a máquina Windows foi utilizada a versão Windows 10 Pro, e para a máquina Linux foi utilizado Ubuntu 22.

A máquina espúria foi simulada com uma VM com o sistema operacional RouterOS. Para essa máquina foi definido um endereço MAC cujo identificador é comumente utilizado por equipamentos de rede. Uma máquina com a distribuição Kali Linux foi incluída na rede para receber todo o tráfego de rede e executar as ferramentas de teste.

O emprego do Python como linguagem de programação neste projeto decorre de diversos motivos, dos quais alguns pontos fortes foram determinantes para a implementação do protótipo de validação. Primeiramente, destaca-se a altíssima portabilidade da linguagem, o que significa que o desenvolvimento de classes ou soluções pode ser facilmente migrado para outras plataformas, exigindo pouco esforço adicional.

Outro aspecto relevante é a natureza dinâmica do Python, o que permite sua aplicação em uma ampla variedade de cenários, como aplicações web, aplicativos Win32, redes, aplicações científicas, aplicativos móveis, entre outros.

Python foi concebido com a filosofia de priorizar o esforço do programador sobre o esforço computacional, enfatizando a legibilidade do código em detrimento da velocidade de execução. Sua sintaxe é concisa e clara, e sua biblioteca padrão oferece recursos poderosos, além de contar com módulos e frameworks desenvolvidos por terceiros. Portanto, é uma escolha adequada para o propósito de validação desta tese, pois se adapta facilmente a cenários com múltiplos módulos de software distribuídos e paralelos que necessitam de coordenação. A versão utilizada neste projeto foi o Python 3.10.

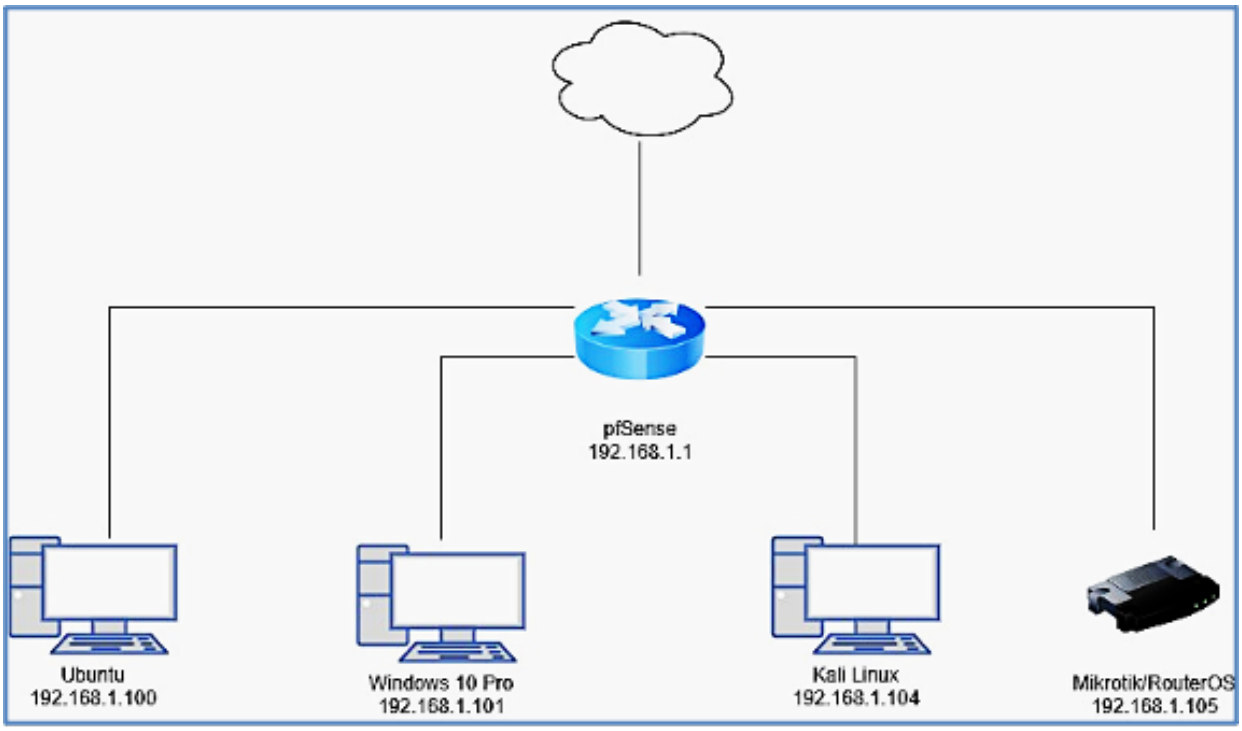


Figura 3.2: Infraestrutura para a realização dos testes

Um dos objetivos do teste foi avaliar a capacidade da proposta em identificar e responder a um dispositivo não autorizado conectado à rede. Comandos foram emitidos para o dispositivo não autorizado, e sua resposta foi monitorada e avaliada por meio da realização de experimento e execução de protótipo, cujo algoritmo específico encontra-se sintetizado na arquitetura modelo, a seguir:

O dispositivo não autorizado deve ser detectado com base em suas características e comportamento. Assim que o dispositivo é detectado, devem ser tomadas medidas adequadas, como isolar o dispositivo da rede ou bloquear seu acesso, para evitar potenciais riscos de segurança.

Para o teste também foi simulada a ausência do equipamento a ser detectado, a fim de comparar os tempos de detecção.

Em alguns casos, a arquitetura proposta deverá lidar com dispositivos não autorizado que utilizam técnicas de bloqueio de firewall para evitar sua detecção. Para esse caso é fundamental a detecção passiva, ou seja, o equipamento espúrio precisa utilizar a rede para se comunicar, assim a detecção passiva é capaz de capturar esses pacotes de rede e determinar qual tipo de equipamento.

O modelo também foi testado com sucesso em um ambiente controlado de uma Instituição Financeira. Embora os dados reais não tenham sido disponibilizados por questões de segurança e identificação de padrões da rede analisada. Mesmo assim, cabe ressaltar que foi possível reproduzir esta avaliação aplicada à detecção de dispositivos não autorizados, posteriormente adotado em conjunto com outras técnicas em cenário real.

4 RESULTADOS E ANÁLISES

Os testes realizados trazem uma abordagem promissora à proposto de detecção e resposta a dispositivos não autorizados.

Foi possível identificar o dispositivo não autorizado e permitir iniciar ações apropriadas, como isolamento ou bloqueio.

No teste, foram utilizadas as técnicas de detecção passiva utilizando *P0F* e detecção ativa utilizando *NMAP* para identificar dispositivos não autorizados na rede. Além disso, foram coletadas impressões digitais dos sistemas operacionais (SO) dos dispositivos detectados (Figura 4.1).

Analisando o log fornecido do teste do *NMAP* e do *P0F* foi possível extrair as seguintes informações:

4.1 DETECÇÃO ATIVA

O escaneamento foi realizado usando o Nmap versão 7.93. O alvo do escaneamento foi o intervalo de endereços IP de 192.168.1.1/24, ou seja, todos os hosts na faixa de IP de 192.168.1.1 a 192.168.1.254.

O host 192.168.1.105 foi relatado como "up"e apresentou os seguintes serviços e portas abertas:

- Porta 21/tcp: Aberta, serviço identificado como "ftp"(MikroTik router ftpd 7.9.2).
- Porta 22/tcp: Aberta, serviço identificado como "ssh"(MikroTik RouterOS sshd).
- Porta 23/tcp: Aberta, serviço identificado como "telnet"(Linux telnetd).
- Porta 80/tcp: Aberta, serviço identificado como "http".
- Uma das respostas HTTP indica um servidor MikroTik rodando RouterOS.

Foi observado que um serviço não foi reconhecido, apesar de retornar dados. Uma impressão digital desse serviço foi fornecida para envio de feedback. O host 192.168.1.105 também é acessível e apresenta serviços como FTP (porta 21), SSH (porta 22), Telnet (porta 23), servidor web HTTP (porta 80), teste de largura de banda (porta 2000) e um serviço desconhecido (porta 8291), diferente do padrão esperado para os demais dispositivos presentes na rede.

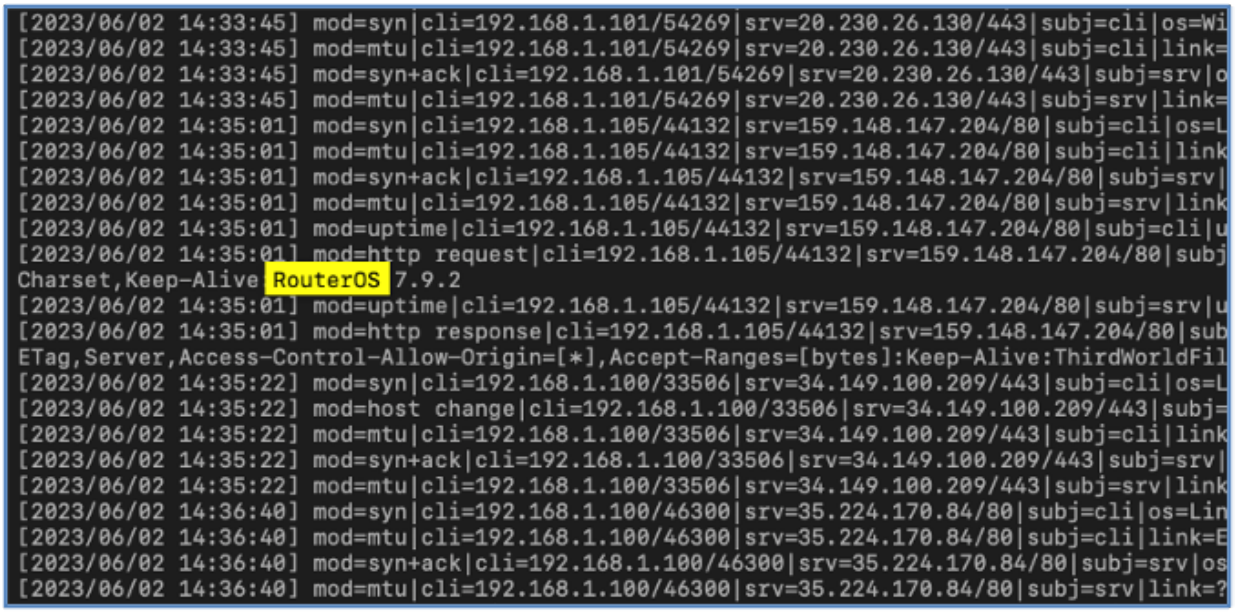


Figura 4.1: Exemplo de detecção ativa

4.2 DETECÇÃO PASSIVA

Tabela 4.1: Resultados filtrados do *P0F*

HOST	DETECTADO	OS	RAW_SIG
192.168.1.100	Linux 2.2.x-3.x	Ubuntu	4:64+0:0:1460:mss*44,7:mss,sok,ts,nop,ws:df,id+:0
192.168.1.101	Windows NT kernel	Windows 10	4:128+0:0:1460:mss*44,8:mss,nop,ws,nop,nop,sok:df,id+:0
192.168.1.101	Windows NT kernel 5.x	Windows 10	4:128+0:0:1460:65535,8:mss,nop,ws,nop,nop,sok:df,id+:0
192.168.1.101	Windows 7 or 8	Windows 10	4:108+20:0:1286:8192,8:mss,nop,ws,nop,nop,sok:df,id+:0
192.168.1.104	Linux 2.2.x-3.x	Kali	4:64+0:0:1460:mss*44,7:mss,sok,ts,nop,ws:df,id+:0
192.168.1.104	Linux 2.2.x-3.x	Kali	4:64+0:0:1460:mss*44,7:mss,sok,ts,nop,ws:df:0
192.168.1.105	Linux 2.2.x-3.x	RouterOS	4:64+0:0:1460:mss*44,7:mss,sok,ts,nop,ws:df,id+:0

Os resultados da detecção passiva mostram que o *P0F* revelou algumas informações sobre os dispositivos, identificando apenas se o dispositivo utiliza sistema operacional Windows ou Linux, e em alguns casos reportando mais de uma possibilidade diferente para o mesmo host.

Analisando as requisições DHCP, foi possível identificar o hostname padrão do dispositivo espúrio (MikroTik), e uma consulta do endereço MAC desse host revelou que o fabricante seria Routerboard.com.

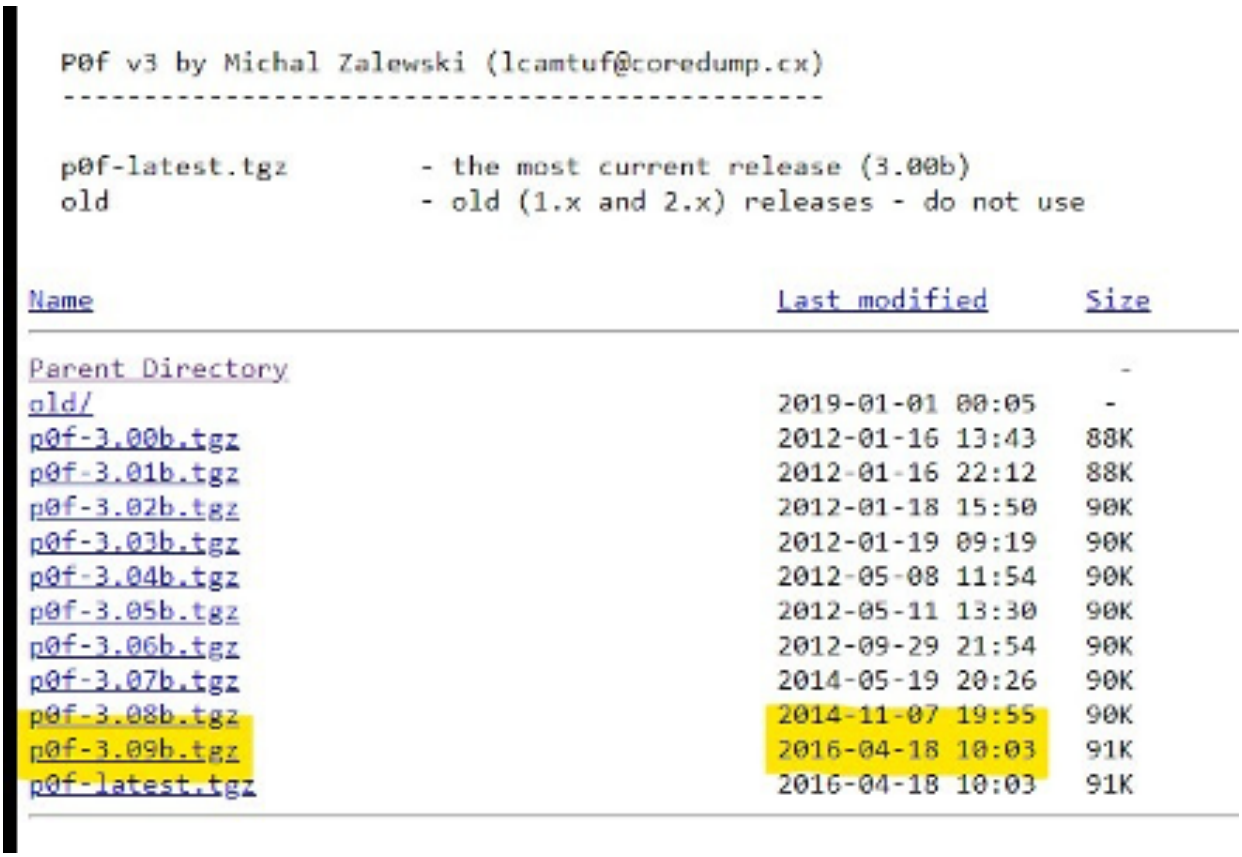
Já o *NMAP* conseguiu identificar hosts com portas abertas, e no caso do dispositivo não autorizado, conseguiu identificar corretamente o modelo e sistema operacional. Os resultados consolidados de detecção do *NMAP* e *P0F* estão na tabela 4.2:

Tabela 4.2: Resumo comparativo de detecção ativa e passiva

Host	Sistema detectado	P0F	NMAP	Hostname detectado
192.168.1.100	Ubuntu / Linux 2.2.x-3.x	Sim	Sim	user-Standard-PC-i440FX-PIIX-1996
192.168.1.101	Windows 10 Pro Windows NT kernel Windows NT kernel 5.x Windows 7 or 8	Sim	Sim	user-Standard-PC-i440FX-PIIX-1996
192.168.1.104	Kali Linux	Sim	Sim	gepro
192.168.1.105	**RouterOS**	**Não**	**Sim**	**MikroTik**

Vale ressaltar que as informações detectadas podem ser alteradas por um atacante, evidenciando a necessidade de somar mais fontes de informações, como banners e identificadores em comunicações não criptografadas, informações do dispositivo no domínio (em redes Windows), algoritmos criptográficos aceitos, entre outras.

No caso do *P0F*, a versão mais recente da ferramenta é a 3.09b, lançada em 2016. A falta de atualizações da base de assinaturas pode ter contribuído para a dificuldade de a ferramenta realizar uma identificação assertiva do sistema operacional do dispositivo.



P0f v3 by Michal Zalewski (lcamtuf@coredump.cx)

p0f-latest.tgz - the most current release (3.00b)
old - old (1.x and 2.x) releases - do not use

Name	Last modified	Size
Parent Directory		-
old/	2019-01-01 00:05	-
p0f-3.00b.tgz	2012-01-16 13:43	88K
p0f-3.01b.tgz	2012-01-16 22:12	88K
p0f-3.02b.tgz	2012-01-18 15:50	90K
p0f-3.03b.tgz	2012-01-19 09:19	90K
p0f-3.04b.tgz	2012-05-08 11:54	90K
p0f-3.05b.tgz	2012-05-11 13:30	90K
p0f-3.06b.tgz	2012-09-29 21:54	90K
p0f-3.07b.tgz	2014-05-19 20:26	90K
p0f-3.08b.tgz	2014-11-07 19:55	90K
p0f-3.09b.tgz	2016-04-18 10:03	91K
p0f-latest.tgz	2016-04-18 10:03	91K

Figura 4.2: Lista de versões disponíveis da ferramenta.

Uma forma que poderia contornar essa limitação seria converter a base de dados de assinaturas do *NMAP*, que recebe atualizações constantes para o padrão do *P0F*, de modo a permitir detecção mais precisa de *fingerprints* de dispositivos, mesmo na abordagem passiva. Esse pode, inclusive, ser um tema para trabalhos futuros.

Superadas as limitações relacionadas às assinaturas desatualizadas, a criação e armazenamento de padrões de SO, permite visualizar instantaneamente dispositivos fora do perfil da rede, permitindo a criação de scripts de bloqueio automático.

4.2.1 Análise forense de um dispositivo espúrio real encontrado

Avaliação dos artefatos testados frente ao método de ataque.

A imagem apresenta externamento os equipamentos que estão sendo utilizados para o cometimento dos atos ilícitos. Trata-se de equipamento originalmente desenvolvido para conexão sem fio, produzido pela empresa Mikrotik. Este específicamente é do modelo MAP Lite (código: RBmAPL-2nD)

No entanto, a utilização de técnicas de descobrimento de rede, o portscanner Nmap idenfica um conjunto de serviços que o equipamento dispõe.

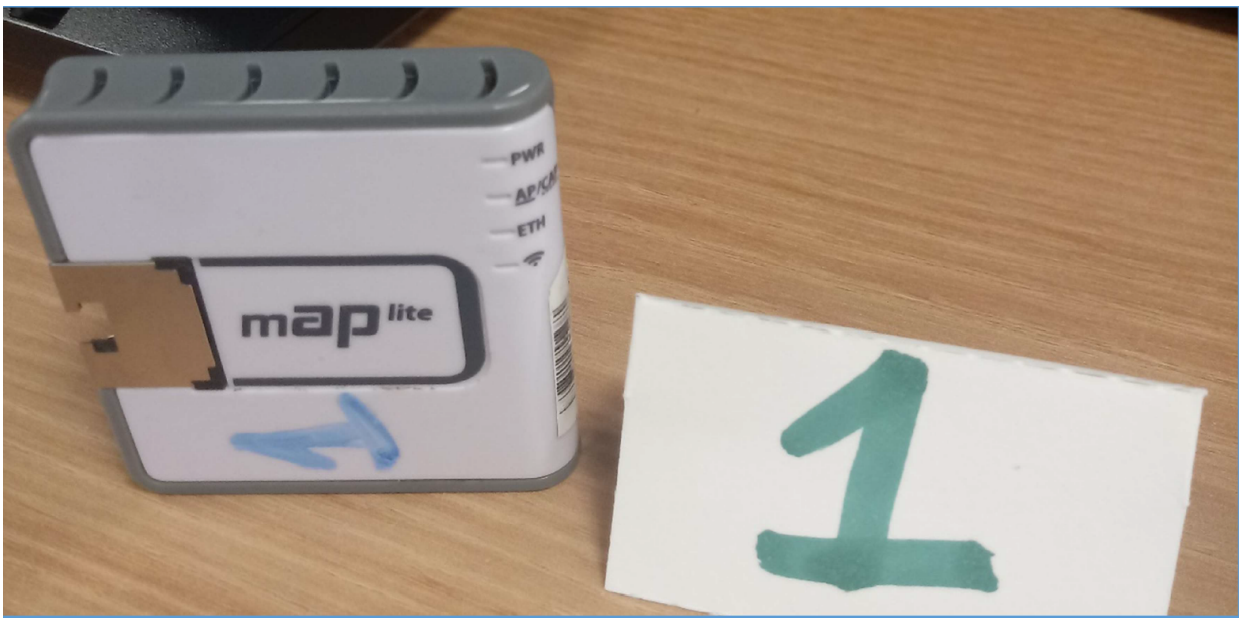


Figura 4.3: Análise de Artefato Roteador Mikrotik.

Em uma das imagens é possível notar que setem como resultado o MAC ADDRESS 6C:3B:6B:E9:F2:03, no qual os três primeiros bytes do conjunto hexadecimal fazem referencia ao Vendor Mikrotik (6C:3B:6B).

O acesso ao dispositivo é feito por interface WEB. Com o objetivo de dificultar a análise forense os atacantes utilizaram de senhas de acesso, dessa forma, com o objetivo de avançar na perícia do dispositivo foi utilizado técnicas de "exploit" onde uma vulnerabilidade é utilizado permitindo acesso ilimitado.

Os atacantes se utilizam de um acesso direto à internet em um endereço IP fixo e único utililizando o protocolo de conexão l2tp, ou seja, é criada uma conexão fechada através da internet com a rede atacada. E é neste ponto que o *P0F* parece ser mais promissor, pois consegue trazer informações de comportamento.

L2TP (Layer 2 Tunneling Protocol) (Townsend e Simpson 1999) é um protocolo de túnel utilizado para criar redes privadas virtuais (VPNs). Ele opera no nível de enlace de dados do modelo OSI (camada 2) e é utilizado em conjunto com protocolos de autenticação, como o IPsec (Internet Protocol Security), para fornecer um ambiente seguro para comunicação através de redes não confiáveis, como a internet.

O L2TP permite a criação de túneis ponto a ponto entre dois dispositivos de rede, como um cliente e um servidor VPN. Ele encapsula os pacotes de dados originais em datagramas UDP (User Datagram Protocol) para transporte através da rede pública. Além disso, o L2TP utiliza o protocolo de controle L2TP para estabelecer, manter e encerrar sessões de túnel, enquanto os dados do usuário são encapsulados usando o protocolo PPP (Point-to-Point Protocol) dentro do túnel L2TP.

Essas conexões L2TP fornecem um meio para estabelecer comunicações seguras entre redes remotas ou entre usuários remotos e uma rede corporativa, garantindo a confidencialidade, integridade e autenticidade dos dados transmitidos.

Equipamento 2

Porta 8291 (Acesso via Winbox) Exploit Router Mikrotik – Porta 8291 Interface VPN utilizada – 177.75.*.*

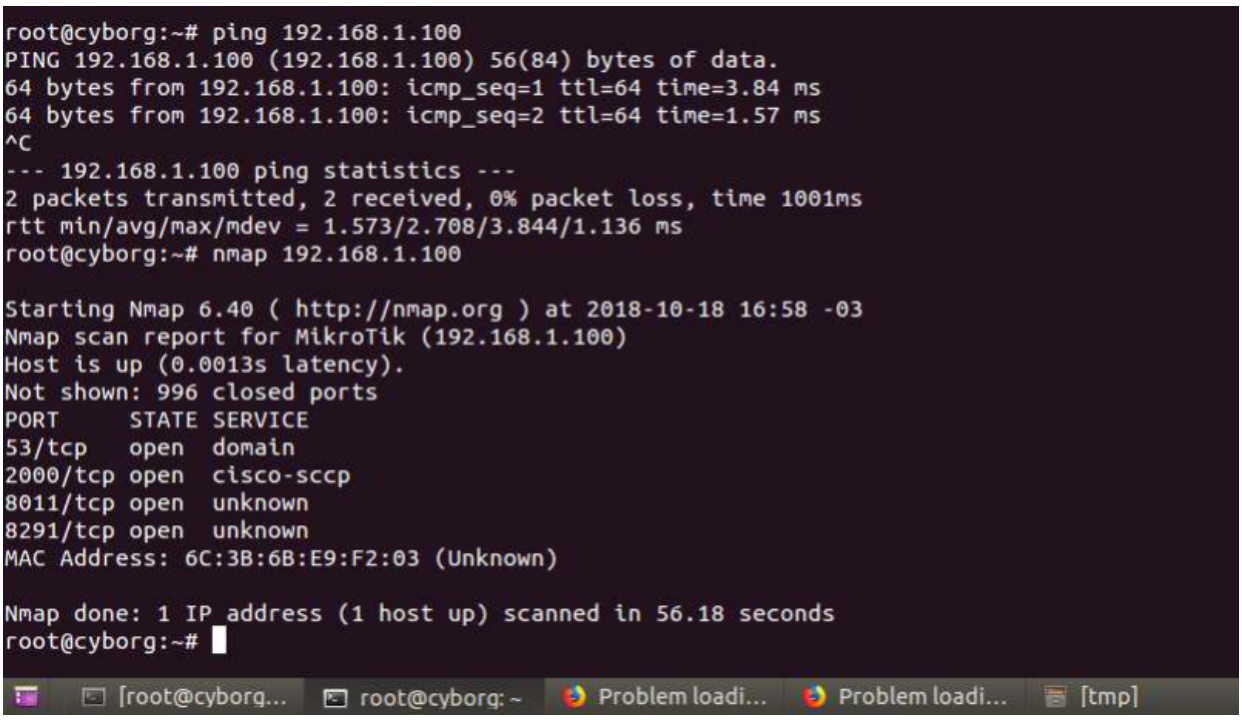


Figura 4.4: Nmap para descoberta de portas liberadas

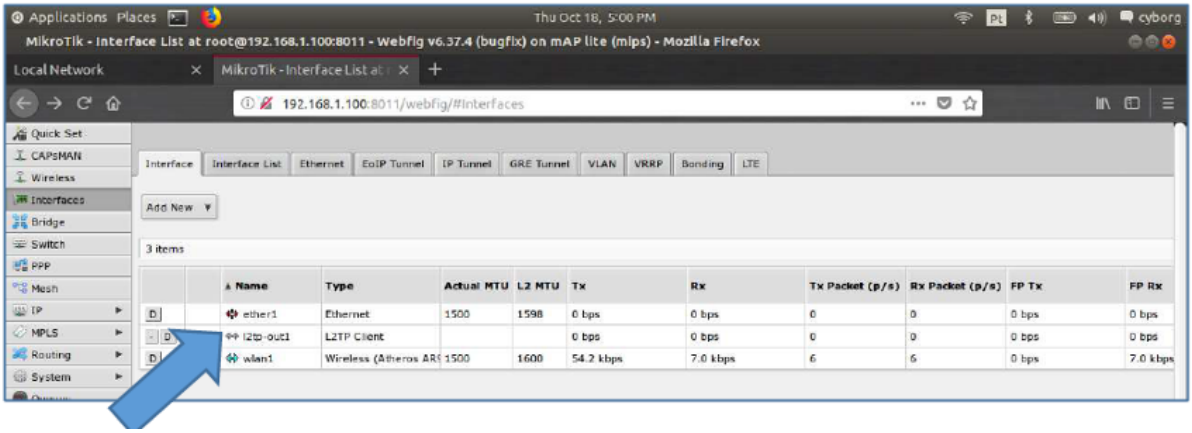


Figura 4.5: Análise de Artefato Roteador Mikrotik - Lista de interfaces

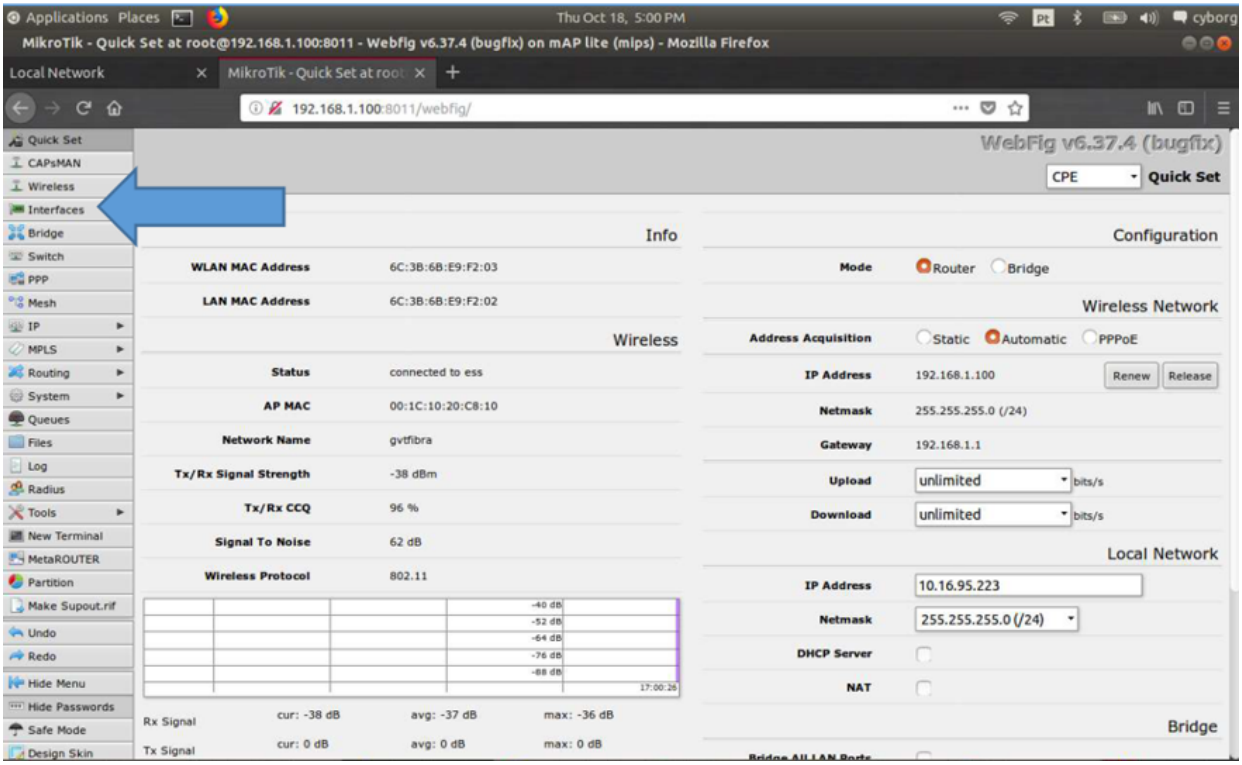


Figura 4.6: Análise de Artefato Roteador Mikrotik - Interface Web

Equipamento 3

Figura 4.7: Análise de Artefato Roteador Mikrotik 2.

5 CONCLUSÃO E TRABALHOS FUTUROS

A proposta, que aproveita as impressões digitais dos sistemas operacionais, pode oferecer uma abordagem promissora para detectar e responder a conexões não autorizadas de dispositivos de *IoT* na ausência do controle de admissão de rede, em especial pelo seu custo benefício e necessidade de se ter uma resposta efetiva, mesmo sem o *NAC*.

Por meio de alguns testes realizados, foi identificado que a proposta pode identificar dispositivos não autorizados e permitir ações apropriadas para manter a segurança da rede.

Os testes realizados podem trazer elementos para permitir a detecção de dispositivos não autorizados no cenário descrito como o problema inicial, em especial para as Instituições Financeiras. Embora o *P0F* não tenha sido capaz de detectar todos os *fingerprints*, o modelo mostrou-se interessante para os casos de necessidade emergencial na detecção desse tipo de ataque.

Devido à dinâmica dos dispositivos informáticos, as bases de assinaturas devem possuir atualização constante. Para aumentar a precisão da detecção, várias fontes de dados podem ser obtidas e correlacionadas. As bases de assinaturas do *NMAP* se mostraram mais assertivas, enquanto o *P0F* teve maior dificuldade de determinar o sistema operacional do dispositivo.

No entanto, por comparar as assinaturas esperadas na rede interna (dados das capturas anteriores) com assinaturas desconhecidas ou inexistentes na base capturada, a inserção de um dispositivo diferente do esperado pode ser prontamente sinalizada (conforme script constante no apêndice deste trabalho). E esta é uma importante vantagem do modelo: manter uma base de assinaturas dos dispositivos autorizados internos permite uma comparação rápida de dispositivos espúrios.

As direções futuras incluem utilização de recursos de aprendizado de máquina, adicionalmente à arquitetura proposta para melhorar a precisão da detecção.

Além disso, a detecção de anomalias é uma característica fundamental dos dados de comportamento. A detecção de anomalias pode ser realizada por meio de técnicas de aprendizado de máquina, que identificam padrões incomuns nos dados de comportamento e sinalizam possíveis dispositivos não autorizados (Antonakakis et al., 2017)(Antonakakis et al. 2017). Neste sentido, podem ser considerados equipamentos fora do padrão utilizado pela rede, MAC address que não corresponde ao equipamento, como um MAC Address para DVR que no fingerprint responde como windows, MAC que se relaciona da rede mas não responde a comandos esperados, como um DVR que deveria ter um webserver e não tem.

Não obstante a proposta ter sido testada com sucesso também na infraestrutura de uma empresa de grande porte do ramo financeiro (onde capturou as tentativas de conexão espúria em ambiente real), dadas as limitações das ferramentas utilizadas, como o *P0F*, por exemplo, uma evolução interessante do modelo em trabalhos futuros seria a utilização de outras características a serem capturadas por outras ferramentas complementares, sendo agrupadas para formarem um score de risco de dispositivo, de forma a não dependerem somente de assinaturas ou *fingerprints*.

Referências Bibliográficas

Albanese, Battista e Jajodia 2015 ALBANESE, M.; BATTISTA, E.; JAJODIA, S. A deception based approach for defeating os and service fingerprinting. In: *2015 IEEE Conference on Communications and Network Security (CNS)*. [S.l.]: IEEE, 2015.

Antonakakis et al. 2017 ANTONAKAKIS, M.; APRIL, T.; BAILEY, M.; BERNHARD, M.; BURSZTEIN, E.; COCHRAN, J.; DURUMERIC, Z.; HALDERMAN, J. A.; INVERNIZZI, L.; KALLITSIS, M.; KUMAR, D.; LEVER, C.; MA, Z.; MASON, J.; MENSCHER, D.; SEAMAN, C.; SULLIVAN, N.; THOMAS, K.; ZHOU, Y. Understanding the mirai botnet. In: *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC: USENIX Association, 2017. p. 1093–1110. ISBN 978-1-931971-40-9. Disponível em: <https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis>.

Bagaa et al. 2020 BAGAA, M.; TALEB, T.; BERNABE, J. B.; SKARMETA, A. A machine learning security framework for iot systems. *IEEE Access*, Institute of Electrical and Electronics Engineers (IEEE), v. 8, p. 114066–114077, 2020. ISSN 2169-3536.

Dildy DILDY, T. J. Network access control: Has it evolved enough for enterprises? *ISACA Journal*, ISACA, v. 2016, n. 4. Disponível em: <https://www.isaca.org/resources/isaca-journal/issues/2016/volume-4/network-access-control-has-it-evolved-enough-for-enterprises>.

Diro et al. 2021 DIRO, A.; CHILAMKURTI, N.; NGUYEN, V.-D.; HEYNE, W. A comprehensive study of anomaly detection schemes in iot networks using machine learning algorithms. *Sensors*, MDPI AG, v. 21, n. 24, p. 8320, dez. 2021. ISSN 1424-8220.

Fyodor 2009 FYODOR. Nmap network scanning: The official nmap project guide to network discovery and security scanning. Insecure.Com LLC, 2009.

Helfrich et al. 2006 HELFRICH, D.; RONNAU, L.; FRAZIER, J.; FORBES, P. *Cisco Network Admission Control, Volume I: NAC Framework Architecture and Design*. Cisco Press, 2006. Accessed: 2024-02-11. Disponível em: <https://www.ciscopress.com/title/1587052415>.

Huacarpuma 2017 HUACARPUMA, C. *Proposição de um modelo e sistema de gerenciamento de dados distribuídos para internet das coisas – GDDIoT*. Tese (Doutorado) — Universidade de Brasília, 2017.

Jafari et al. 2018 JAFARI, H.; OMOTERE, O.; ADESINA, D.; WU, H.-H.; QIAN, L. Iot devices fingerprinting using deep learning. In: *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*. [S.l.]: IEEE, 2018.

Kumari e Jain 2023 KUMARI, P.; JAIN, A. K. A comprehensive study of ddos attacks over iot network and their countermeasures. *Computers amp; Security*, Elsevier BV, v. 127, p. 103096, abr. 2023. ISSN 0167-4048.

Landry e Koger 2023 LANDRY, B.; KOGER, M. Exploring zero trust network architectures for building secure networks. In: . [S.l.: s.n.], 2023.

Lin e Tang 2018 LIN, D.; TANG, B. Detecting unmanaged and unauthorized devices on the network with long short-term memory network. In: *2018 IEEE International Conference on Big Data (Big Data)*. [S.l.: s.n.], 2018. p. 2980–2985.

Lyon 2009 LYON, G. Nmap network scanning: The official nmap project guide to network discovery and security scanning. *Insecure.com LLC*, 2009.

MENDONÇA 2019 MENDONÇA, F. L. L. d. *Proposição de um modelo de interoperação peer-to-peer para internet das coisas – P2PIoT*. Tese (Doutorado) — Universidade de Brasília, 2019.

MITRE 2021 MITRE. Mitre attck framework. *MITRE ATTCK Framework,*, 2021.

MITRE 2021 MITRE. Mitre attck framework,t1200 - defense evasion. *MITRE ATTCK Framework,TTP,*, 2021.

Press 2007 PRESS, C. *Cisco Network Admission Control, Volume II: NAC Framework Deployment and Troubleshooting*. Cisco Press, 2007. Accessed: 2024-02-11. Disponível em: <https://www.ciscopress.com/title/1587052423>.

Rocha, Melo e Sousa 2021 ROCHA, B. C. da; MELO, L. P. de; SOUSA, R. T. de. Preventing apt attacks on lan networks with connected iot devices using a zero trust based security model. In: *2021 Workshop on Communication Networks and Power Systems (WCNPS)*. [S.l.]: IEEE, 2021.

Roy et al. 2022 ROY, S.; SHARMIN, N.; ACOSTA, J. C.; KIEKINTVELD, C.; LASZKA, A. Survey and taxonomy of adversarial reconnaissance techniques. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 55, n. 6, dec 2022. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/3538704>.

Saraiva et al. 2014 SARAIVA, A.; FEITOSA, E.; ELLERES, P.; CARNEIRO, G. *SBC*, Belo Horizonte - MG, p. 49–98, 2014.

Soares Francisco Lopes de Caldas Filho 2022 SOARES FRANCISCO LOPES DE CALDAS FILHO, M. F. S. F. L. L. d. M. E. D. C. e. R. T. d. S. J. S. C. M. Arquitetura de detecÇÃo de intrusÃo por anomalias com federated learning em redes iot. *Conferências IADIS Ibero-Americanas Computação Aplicada e WWW/Internet 2022*, 2022.

Townsend e Simpson 1999 TOWNSEND, K.; SIMPSON, W. Layer two tunneling protocol “l2tp”. *RFC 2661*, IETF, 1999. Disponível em: <https://tools.ietf.org/html/rfc2661>.

Yang et al. 2022 YANG, X.; YANG, X.; YI, X.; KHALIL, I.; ZHOU, X.; HE, D.; HUANG, X.; NEPAL, S. Blockchain-based secure and lightweight authentication for internet of things. *IEEE Internet of Things Journal*, Institute of Electrical and Electronics Engineers (IEEE), v. 9, n. 5, p. 3321–3332, mar. 2022. ISSN 2372-2541.

APÊNDICES

I. APÊNDICE

Nesta seção apresentamos algumas partes dos scripts utilizados para identificação de dispositivos e criação de banco de dados dos resultados obtidos.

Aprofundando nas funcionalidades e no fluxo de operação do código, detalhando cada componente e sua contribuição para a finalidade do script: detecção e catalogação de dispositivos na rede.

Explicação detalhada do script:

Obtenção de endereços IP da subrede local: A função get$_{l}ocal_{s}ubnet_{a}ddresses()utilizaocomandoipconfigparaobt$

Detecção passiva de dispositivos usando P0F: A função detect$_{d}evices_{p}assive()executaoP0Fparacapturarotrfegoc$

Detecção ativa de dispositivos usando NMAP: A função detect$_{d}evices_{a}ctive(subnet_{a}ddresses)executaoNMAPpa$

Atualização do banco de dados de dispositivos: A função update$_{d}atabase(new_{d}evices)verificasehdispositivosnov$

Função principal: A função main() coordena todo o processo de detecção de dispositivos não autorizados. Chama as funções de detecção ativa e passiva de dispositivos. Exibe os dispositivos detectados em ambas as abordagens. Chama a função de atualização do banco de dados com os fingerprints dos dispositivos detectados.

Este script pode ser executado periodicamente para verificar se há dispositivos não autorizados na rede, atualizando o banco de dados de dispositivos sempre que novos dispositivos são detectados. Isso permite o acompanhamento das mudanças na composição da rede ao longo do tempo.

I.1 SCRIPTS

I.1.1 Detecção de dispositivos e Armazenamento de fingerprints

```
1
2
3
4  import subprocess
5  import json
6  import os
7  import ipaddress
8
9  # Fun  o para obter endere os IP da subrede local
10 def get_local_subnet_addresses():
11     # Obter o endere o IP da m quina local
12     ipconfig_output = subprocess.check_output(["ipconfig"]).decode()
13     local_ip = ipconfig_output.split("IPv4 Address. . . . . . . . . . . : ")[1].
```

```
            split("\n")[0]
14
15      # Obter a m scara de subrede
16      subnet_mask = ipconfig_output.split("Subnet Mask . . . . . . . . . . . : ")
            [1].split("\n")[0]
17
18      # Calcular o prefixo da subrede
19      network = ipaddress.IPv4Network(f"{local_ip}/{subnet_mask}", strict=False)
20      return [str(ip) for ip in network.hosts()]
21
22  # Fun  o para detec  o passiva de dispositivos usando P0F
23  def detect_devices_passive():
24      try:
25          # Executar o P0F para capturar o tr fego de rede e obter os fingerprints
                dos dispositivos
26          output = subprocess.check_output(["p0f", "-f", "network_traffic.pcap", "-o
                ", "json"])
27          devices = json.loads(output)
28          return devices
29      except subprocess.CalledProcessError:
30          print("Erro ao executar o P0F.")
31          return []
32
33  # Fun  o para detec  o ativa de dispositivos usando NMAP
34  def detect_devices_active(subnet_addresses):
35      try:
36          # Executar o NMAP para escanear a subrede e obter os fingerprints dos
                dispositivos
37          output = subprocess.check_output(["nmap", "-O", "-oX", "nmap_results.xml",
                 ",".join(subnet_addresses)])
38          devices = subprocess.check_output(["xsltproc", "--xpath", "//host", "
                nmap_results.xml"])
39          return devices
40      except subprocess.CalledProcessError:
41          print("Erro ao executar o NMAP.")
42          return []
43
44  # Fun  o para atualizar o banco de dados de dispositivos
45  def update_database(new_devices):
46      # Verificar se o arquivo de banco de dados existe
47      if os.path.exists("device_database.json"):
48          # Carregar o banco de dados existente
49          with open("device_database.json", "r") as f:
50              database = json.load(f)
51      else:
52          # Criar um novo banco de dados se n o existir
53          database = []
54
55      # Verificar se h  dispositivos novos
56      for device in new_devices:
57          if device not in database:
58              # Adicionar o novo dispositivo ao banco de dados
```

```
59              database.append(device)
60              print(f"Novo dispositivo detectado: {device}")
61
62      # Atualizar o arquivo de banco de dados
63      with open("device_database.json", "w") as f:
64          json.dump(database, f, indent=4)
65
66  # Fun  o principal
67  def main():
68      # Obter os endere os IP da subrede local
69      subnet_addresses = get_local_subnet_addresses()
70
71      # Detec  o ativa de dispositivos na subrede local
72      active_devices = detect_devices_active(subnet_addresses)
73      print("Dispositivos detectados de forma ativa:")
74      for device in active_devices:
75          print(device)
76
77      # Detec  o passiva de dispositivos
78      passive_devices = detect_devices_passive()
79      print("\nDispositivos detectados de forma passiva:")
80      for device in passive_devices:
81          print(device)
82
83      # Atualiza  o do banco de dados
84      update_database(passive_devices + active_devices)
85
86  # Executar a fun  o principal
87  if __name__ == "__main__":
88      main()
```