INSTITUTO FEDERAL
Catarinense

TECNOLOGIA E REDES DE COMPUTADORES

10ª EDIÇÃO

VANDERLEI FREITAS JUNIOR

INSTITUTO FEDERAL
Catarinense

TECNOLOGIA E REDES DE COMPUTADORES

10ª EDIÇÃO

VANDERLEI FREITAS JUNIOR

2024

Instituto Federal Catarinense

INSTITUTO FEDERAL
Catarinense

INSTITUTO FEDERAL CATARINENSE

REITOR
Rudinei Kock Exterckoter

PRÓ-REITORA DE ENSINO
Liane Vizzotto

PRÓ-REITOR DE EXTENSÃO, PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO
Cleder Alexandre Somensi

PRÓ-REITORA DE DESENVOLVIMENTO, INCLUSÃO, DIVERSIDADE E ASSISTÊNCIA À PESSOA
Jamile Delagnelo Fagundes da Silva

PRÓ-REITOR DE ADMINISTRAÇÃO
Jorge Luís de Souza Mota

CÂMPUS AVANÇADO SOMBRIO

DIRETOR GERAL
Victor Martins de Sousa

DIRETOR DE ENSINO, PESQUISA E EXTENSÃO
Mirian Rocho da Rosa Silveira

INSTITUTO FEDERAL CATARINENSE,
CÂMPUS AVANÇADO SOMBRIO
Av. Prefeito Francisco Lummertz Júnior, 931
CEP 88960-000 - Sombrio/SC
www.sombrio.ifc.edu.br

INSTITUTO FEDERAL
Catarinense

Direção Editorial	Vanderlei Freitas Junior
Capa e Projeto Gráfico	Claiton Andrade Junior
Editoração Eletrônica	Vanderlei Freitas Junior
Comitê Editorial	Armando Mendes Neto
	Cleber Luiz Damin Ferro
	Diego Monsani
	Guilherme Klein da Silva Bitencourt
	Jéferson Mendonça de Limas
	Joédio Borges Junior
	Marco Antônio Silveira de Souza
	Matheus Lorenzato Braga
	Sandra Vieira
	Vanderlei Freitas Junior
	Victor Martins de Sousa
Revisão	Gilnei Magnus dos Santos
Organizador	Vanderlei Freitas Junior

INSTITUTO FEDERAL
Catarinense

ISBN: 978-6-50126-310-6

9 786501 263106

Ficha catalográfica elaborada pela *Biblioteca do IFC Sombrio*

T255 Tecnologias e Redes de Computadores / Vanderlei Freitas Junior (org.). -- 10 ed.-- Sombrio : Instituto Federal Catarinense, 2024.
91 f.

ISBN 978-65-01-26310-6

1. Redes de computadores - Gerência. 2. Proteção de dados I.Freitas Junior, Vanderlei - 1980-. II. Título.

CDD: Ed. 21 -- 004.6

INSTITUTO FEDERAL
Catarinense

Agradecimentos

Agradecemos as valiosas contribuições de Claiton Andrade Junior, Diego Monsani, Coordenação do Sistema Integrado de Bibliotecas do IFC, além dos alunos e professores que contribuíram com suas pesquisas para o engrandecimento desta publicação.

Esta é uma publicação do Curso Superior de

Sumário de artigos

Sumário de Autores

Dez anos não são dez dias!

Vanderlei Freitas Junior

Há dez anos iniciávamos um trabalho de divulgação científica sem precedentes no Instituto Federal Catarinense, Câmpus Sombrio. Dávamos início à publicação de um livro digital que pudesse garantir visibilidade para toda a produção acadêmica de nossos estudantes, pesquisadores e docentes, entretanto nem nas nossas mais otimistas previsões poderíamos imaginar que chegaríamos à décima edição. E aqui está ela! Este trabalho exitoso, construído a muitas mãos, já foi capaz de levar conhecimento de qualidade para muito além dos muros institucionais, honrando nossa missão de interferir positivamente em nosso entorno, bem como nas comunidades em que estamos inseridos.

Mas, importante registrar que, para além das questões relacionadas à necessária extensão universitária, esta obra também trouxe motivação e reconhecimento para aqueles que, depois do árduo trabalho de pesquisa, viram reconhecidos os resultados de seus esforços, transformando alunos em autores de capítulo de um livro, com linhas a incluir em seus currículos acadêmicos!

Talvez de outra forma isto não seria possível.

Aproveito este momento para agradecer a todos aqueles que embarcaram conosco nesta jornada, que acreditaram, mesmo em uma época em que não se falava em publicação acadêmica digital no Brasil, de que era possível. E aqui está a prova! Dez anos demonstrando que estávamos certos.

Análise e avaliação de vulnerabilidades em ambiente virtualizado com o Openvas e Proxmox

Cristian Gomes Selau[1] Guilherme Klein da Silva Bitencourt[2] Marco Antônio Silveira de Souza[3]

[1] Acadêmico do Instituto Federal Catarinense – Campus Avançado Sombrio – 88960000 – Sombrio – SC – Brasil

[2, 3,] Docentes do Instituto Federal Catarinense – Campus Avançado Sombrio – 88960000 – Sombrio – SC – Brasil

cristiang02selau@gmail.com,
guilherme.bitencourt@ifc.edu.br,
marco.souza@ifc.edu.br

***Abstract**: This work aims to test and evaluate the security of the ProxMox Virtual Environment (ProxMox VE) hypervisor, an open-source tool with various features used in both academic and corporate environments for server management and virtualization. ProxMox VE optimizes the utilization of computational resources, reducing costs and time in server management. Vulnerability scanning was performed on ProxMox VE using the open-source tool OpenVAS (Open Vulnerability Assessment System). Its scanning capability can detect various security flaws, open ports, and generate detailed reports on how to mitigate the vulnerabilities found. Additionally, automation was implemented in the processes of updating OpenVAS and its vulnerability database through a shell script, increasing the reliability of the tool. This study evaluated the vulnerabilities found in three versions of ProxMox VE and discussed OpenVAS suggestions for mitigating these flaws, as well as the*

application of best practices in virtualized environments.

Resumo: *Este trabalho tem como objetivo testar e avaliar a segurança do hipervisor ProxMox Virtual Environment (ProxMox VE), ferramenta de código aberto com diversos recursos, utilizada em ambientes de estudo e corporativo na gestão e virtualização de servidores. O ProxMox VE otimiza a utilização dos recursos computacionais, reduzindo custos e o tempo no gerenciamento dos servidores. Foi realizado a varredura por vulnerabilidades no ProxMox VE utilizando a ferramenta open source OpenVAS (Open Vulnerability Assessment System). Sua varredura é capaz de encontrar diversas falhas de segurança, portas abertas e gerar relatórios detalhados sobre como mitigar as vulnerabilidades encontradas. Além disso, também foi feita a automação nos processos de atualização do OpenVAS e de seu banco de dados de vulnerabilidades por meio de um shell script, aumentando a confiabilidade da ferramenta. Nesse estudo foram avaliadas as vulnerabilidades encontradas em três versões do ProxMox VE e discutido sobre as sugestões do OpenVAS para mitigar essas falhas, bem como a aplicação de boas práticas em ambientes virtualizados.*

1. Introdução

O processo de virtualização de um servidor pode ser definido como uma tecnologia que realiza a segmentação de um servidor físico em dois ou mais servidores virtuais, além disso, cada servidor pode executar um sistema operacional distinto e independente dos demais (VMWARE, 2020).

Com a evolução da tecnologia da informação, as preocupações para melhorar a otimização dos servidores e reduzir os custos operacionais aumentaram. Nesse cenário, a virtualização surge como uma aliada, sendo capaz de otimizar o uso do *hardware* físico, aumentando o desempenho (SCHIMIGUEL e RIBEIRO, 2016).

Criada em 1967, a tecnologia executada na máquina física chamada de Virtual Machine Monitor (VMM) ou hipervisor é subdividida em dois tipos. O hipervisor é uma camada de software responsável por gerenciar os recursos de *hardware*, estabelecer os ambientes virtuais com a criação das máquinas virtuais (VMs) e executar os sistemas operacionais em cada VM. Sua gerência é feita através de uma interface gráfica acessível pelo navegador (SEO, 2009).

Durante um trabalho realizado em uma empresa da região sul de Santa Catarina, o autor identificou que muitos provedores de internet utilizavam hipervisores como o ProxMox Virtual Environment e o VMware ESXI em suas redes, com o intuito de virtualizar serviços e armazenar dados pessoais sobre seus clientes. Também, foi constatado a escassez de trabalhos voltados para a avaliação de segurança dessas ferramentas. Esses dois fatores motivaram o autor a realizar essa pesquisa.

Em um estudo realizado por Jacobsen *et al* (2015), após a utilização de experimentos práticos, realizou-se a comparação entre a ferramenta de código aberto OpenVAS (Open Vulnerability Assessment System), e o Nessus, que possui uma licença paga. Ambas as ferramentas desempenham a função de analisar vulnerabilidades em sistemas operacionais e outros serviços. O estudo mostrou que o OpenVAS por mais que tenha uma maior demora para mostrar os resultados, se utilizado em uma rede com um número pequeno de dispositivos conectados, se torna uma opção mais vantajosa, considerando que a ferramenta é de código aberto.

Nesse contexto, a proposta desse trabalho é promover o estudo da segurança em ambientes virtualizados, utilizando-se o OpenVAS para aplicar a varredura, análise e possibilidades de mitigação das vulnerabilidades encontradas no hipervisor ProxMox VE (Virtual Environment). Nas análises, foram escolhidas as versões 6.4-1, 7.4-1 e 8.0-1 por serem as principais disponibilizadas no site oficial da ferramenta e possivelmente as mais utilizadas.

Ao verificar as três versões será possível identificar se as vulnerabilidades encontradas em uma versão foram mitigadas na versão posterior, além de ajudar na avaliação da eficiência do OpenVAS nos processos de detecção e identificação de vulnerabilidades (JACOBSEN et al, 2015).

Com o objetivo de facilitar a gerência do servidor instalado com o OpenVAS, foi realizado a automação nos processos de atualização do OpenVAS e de seu banco de dados de vulnerabilidades por meio de um shell script, aumentando a confiabilidade da ferramenta.

1.1.1 Objetivo geral

O objetivo geral deste trabalho é avaliar a segurança de diferentes versões do ProxMox VE utilizando-se como ferramenta de avaliação o OpenVAS.

1.1.2 Objetivos Específicos

Este estudo possui os seguintes objetivos específicos:
a) Testar a segurança com a varredura de vulnerabilidades no hipervisor ProxMox VE (Virtual Environment).

b) Explorar as funcionalidades da ferramenta OpenVAS (Open Vulnerability Assessment System).

c) Análise e propostas de mitigação para as vulnerabilidades encontradas no ProxMox VE.

d) Automatizar processos do servidor OpenVAS utilizando shell script.

2 Virtualização: definição, vantagens e desvantagens

Segundo informações de RedHat (2023), a virtualização de servidores pode ser definida como uma tecnologia que realiza a segmentação de um servidor físico composto por CPU, memória RAM, armazenamento e conexões de rede em dois ou mais servidores lógicos distintos. Na virtualização é utilizado um software específico denominado hipervisor ou Virtual Machine Monitor (VMM), que permite o gerenciamento e a segmentação eficientes do servidor físico em várias máquinas virtuais separadas.

Os servidores virtuais, criados dentro de um servidor físico por meio desse processo, são denominados máquinas virtuais (ou VMs) e cada máquina virtual criada pode ser capaz de reproduzir um sistema operacional distinto e independente (IBM, 2020).

A virtualização fornece maior tolerância a falhas através do recurso de alta disponibilidade, isso significa que se um servidor físico falhar, as máquinas virtuais podem ser automaticamente migradas para outros servidores em funcionamento, reduzindo o tempo de inatividade e aumentando significativamente a disponibilidade do servidor. Além disso, existe a possibilidade de criar backups que salvam o estado e os dados de uma máquina virtual em um momento específico, além do status de energia (por exemplo, ligada, desligada, suspensa). Os dados desse backup incluem tanto os arquivos que fazem parte da máquina virtual, quanto as configurações de alocação de recursos, rede e de todos os arquivos salvos e modificados em disco dentro do sistema operacional que estiver instalado (VMWARE, 2020).

O estudo realizado por Ghannoum e Rodrigues (2018), mencionou algumas desvantagens a serem consideradas no uso da virtualização, tais como: o acúmulo de máquinas virtuais ociosas, baixa performance em algumas aplicações e a dificuldade em emular dispositivos, como placas externas e dispositivos USB. O excesso de máquinas virtuais ociosas impacta a eficiência de VMs importantes, e ocasiona o desperdício de recursos computacionais. Outra desvantagem mencionada é a baixa performance que pode ocorrer ao utilizar certas aplicações de forma virtualizada, pois muitas delas podem não ter total compatibilidade ou suporte a virtualização. Além disso, a dificuldade em emular dispositivos externos, como placas e dispositivos USB, representa um desafio na virtualização, levando a possíveis falhas e incompatibilidades no funcionamento desses dispositivos.

Por outro lado, o artigo de Barros *et al* (2019), revelou que em um cenário onde a virtualização é aplicada, é possível obter maior eficiência na alocação dos recursos computacionais, melhor gerenciamento, redução dos gastos com energia e com a compra de novos servidores. Com apenas um servidor físico, é possível criar vários outros virtuais, cada um deles executando os mesmos serviços e aplicações dos servidores físicos. Além disso, existe uma maior facilidade na aplicação de medidas de segurança por conta da centralização dos sistemas em execução.

3 Definição e tipos de hipervisores

O hipervisor (também conhecido como VMM, monitor de máquina virtual) desempenha um papel fundamental na virtualização. Essa ferramenta pode ser definida como um software responsável pela hospedagem, gerência e controle das máquinas virtuais e dos recursos, além disso a sua arquitetura é subdividida em dois tipos (MACEDO E SANTOS, 2014).

O tipo 1, conhecido como "*bare-metal*" (servidor dedicado), é uma designação para hipervisores que operam

diretamente no *hardware* do *host*. Nesse cenário, o *host* não possui um sistema operacional instalado, já que o software do hipervisor assume o papel de um sistema operacional leve, responsável pela gestão do ambiente virtualizado (AWS, 2023).

O termo "*hosted*"(hospedado) é utilizado para denominar o tipo 2, que são os hipervisores executados em um sistema operacional já existente no *host.* Eles são implementados de maneira semelhante a outros aplicativos e compartilham recursos e *hardware* com o sistema operacional principal (MACEDO E SANTOS, 2014).

O hipervisor tem a capacidade de carregar as imagens de cada máquina virtual para criar múltiplos sistemas operacionais em uma única máquina, atuando como um intermediário nesse processo. Em sua interface de gerência é possível configurar e alocar os recursos de *hardware,* como CPU, memória RAM e de armazenamento disponíveis para máquinas virtuais individuais conforme as necessidades, resultando em uma otimização dos recursos da infraestrutura de TI (AWS, 2023).

4 Histórico, funcionalidades e arquitetura do ProxMox VE (ProxMox Virtual Environment)

4.1.1 História e evolução do ProxMox VE

Lançada em abril de 2008 pela empresa ProxMox Server Solutions GmbH, a ferramenta de código aberto ProxMox Virtual Environment (VE) foi desenvolvida para implementar e gerenciar a virtualização de servidores. Essa ferramenta combina virtualização baseada em contêiner e o hipervisor KVM, seu gerenciamento é feito por uma interface gráfica acessível pela web (HÄSLER, 2015).

Segundo o CEO da empresa, Martin Maurer, o objetivo da ProxMox Server Solutions GmbH, era desenvolver programas para o Linux de forma eficiente e fácil gerenciamento.

Além disso, a meta também era dar a oportunidade para os usuários criarem ambientes seguros, estáveis e escaláveis. A partir de então, mais de 73.000 *hosts* de servidores em todo o mundo usavam ProxMox VE em 2015 (HÄSLER, 2015).

A cada atualização lançada, novas funcionalidades foram sendo implementadas no ProxMox VE. Em 2023, a versão 8.0 trouxe várias melhorias, fazendo o uso do *kernel* Linux 6.2 mais atual, e de versões atualizadas do QEMU 8.0.2, LXC 5.0.2, ZFS 2.1.12 e Ceph Quincy 17.2.6. Com essas mudanças o ProxMox VE ficou mais compatível com as tecnologias atuais de código aberto para ambientes virtuais (SMITH, 2023).

4.1.2. Funcionalidades e arquitetura de virtualização do ProxMox VE

Segundo Tolfo (2019), com o ProxMox é possível criar máquinas virtuais com diferentes sistemas operacionais, como por exemplo Microsoft Windows e GNU Linux simultaneamente e de forma isolada um do outro, ou seja, as ferramentas e serviços utilizados em uma máquina virtual não vão afetar o outro sistema operacional que estiver sendo virtualizado.

No estudo de Silva (2019), foram destacadas algumas vantagens importantes, como: recursos para segurança, grande flexibilidade, facilidade na gestão, entre outros fatores. Essas vantagens contribuem de forma positiva para as empresas que fazem implementação do ProxMox VE em sua infraestrutura.

A ferramenta possui de forma nativa algumas tecnologias para virtualização de código aberto, como por exemplo: KVM (Máquina Virtual Baseada em *Kernel*), que permite ao ProxMox VE criar máquinas virtuais (VMs) e executar sistemas operacionais e aplicativos com isolamento total em um único servidor físico; o OpenVZ, que melhora a eficiência de recursos e os containers LXC, que oferecem

inicializações rápidas e consumo mínimo de recursos, além da possibilidade de migração de dados em tempo real.

O ProxMox VE fornece opções que permitem o aumento da segurança do ambiente. É possível separar os serviços de um projeto, como por exemplo, um servidor web e seu banco de dados, e deixar cada um em uma VM distinta, caso ocorra um problema em algum dos processos em operação, os demais serviços não serão afetados. Além disso, é possível segmentar as VMs por VLAN, garantindo que em caso de falha ou comprometimento de alguma máquina virtual, o problema não se propague para outras partes da rede, mantendo os demais serviços protegidos (SILVA, 2019).

A possibilidade de alocar os recursos de *hardware* na criação de cada máquina virtual torna o ProxMox VE uma ferramenta muito flexível, pois facilita a gerência dos recursos por parte do administrador da rede de acordo com a necessidade em cada ambiente virtualizado (TOLFO, 2019).

O ProxMox VE fornece uma interface web intuitiva para gerência e centralização de recursos. Nessa interface é possível visualizar, criar e gerenciar recursos de cada máquina virtual criada em um único lugar. A necessidade de manutenção diminui, e o processo de controle fica mais bem otimizado, podendo ser feito em poucos segundos. Com isso, os administradores podem configurar suas VMs de forma mais fácil a partir de qualquer navegador.

A compatibilidade com diferentes servidores é um ponto notável a ser destacado no ProxMox VE, pois esse recurso simplifica consideravelmente o processo de migração, permitindo a transferência de ambientes virtualizados com facilidade. Com a capacidade de realizar backup de recursos e criar uma imagem na nova máquina virtual, a transição entre servidores se torna um procedimento mais eficiente e acessível, economizando tempo e recursos (OLIVEIRA, 2022).

5 Histórico, arquitetura e funcionalidades do OpenVAS (Open Vulnerability Assessment System)

5.1.1 Histórico do OpenVAS (Open Vulnerability Assessment System)

De acordo com informações disponíveis na documentação da comunidade *Greenbone* GitHub (2023), no ano de 2005 os desenvolvedores do scanner de vulnerabilidades *Nessus,* que até então possuía código aberto, optaram por comercializar a ferramenta e isso fez com que um grupo de desenvolvedores que contribuíram com o *Nessus* fundassem a empresa *Greenbone Networks,* em 2008. Por ter se tornado pago, surgiram vários *forks* do *Nessus* em 2006, porém o único que permaneceu sendo utilizado foi o framework Open Vulnerability Assessment System (OpenVAS).

Em 2009, a Greenbone implementou recursos para realizar o gerenciamento das vulnerabilidades, como a interface web e o serviço de gestão central. No mesmo ano, o OpenVAS *framework* teve seu nome oficialmente alterado para OpenVAS.

Em 2010, foi lançado a primeira versão do "*Greenbone Security Manager*", sendo esse o primeiro produto comercial lançado pela *Greenbone Networks*, baseado no OpenVAS. Essa solução oferece mais recursos e suporte, com foco nas empresas que exigem maior atenção com a segurança de suas infraestruturas. O produto passou por melhorias entre 2010 a 2016, e trouxe novas funções a cada atualização, assim como o OpenVAS, sua versão de código aberto, que também recebeu atualizações referentes ao banco de dados de vulnerabilidades e novas funcionalidades, como a inclusão de avisos de segurança atualizados diariamente (GITHUB, 2023).

Com a chegada da versão 22.4w, foi adicionado o *Notus Scanner.* Mora (2022), afirma que o Notus é um complemento do OpenVAS dentro do Greenbone Vulnerability Management

(GVM), executando verificações de vulnerabilidade em segundo plano sem a interação do usuário. Ele aprimora o desempenho ao comparar listas de software instalado nos sistemas alvos com listas de software vulnerável, simplificando o processo de identificação de vulnerabilidades.

5.1.2 Arquitetura e funcionalidades do OpenVAS (Open Vulnerability Assessment System)

O OpenVAS é uma ferramenta de código aberto utilizada para a detecção, avaliação e identificação de vulnerabilidades. O propósito central está na identificação e análise minuciosa de potenciais falhas de segurança presentes em sistemas operacionais, redes e aplicativos. A ferramenta se destaca pela sua capacidade de realizar varreduras detalhadas, ajudando na criação de estratégias cruciais para a mitigação assertiva dos riscos de segurança digital (FERNANDES, 2019).

A arquitetura do OpenVAS é formada pelas interfaces cliente CLI e web, além de dois módulos de serviço, *OpenVAS Manager e o OpenVAS Scanner*. A interface CLI possibilita ao usuário interagir com o sistema e executar tarefas de varredura e gerenciamento das vulnerabilidades, por meio de comandos no terminal. O sistema também conta com uma interface gráfica intuitiva acessível pela web, isso facilita o gerenciamento e realização das varreduras sem a necessidade de digitar os comandos (FERNANDES, 2019).

O OpenVAS possui o módulo de serviço *OpenVAS Manager*, responsável por gerenciar todas as funcionalidades da ferramenta, incluindo os resultados encontrados nas varreduras e o *OpenVAS Scanner,* que tem a função de descobrir as vulnerabilidades presentes dos *hosts* alvo. Juntos com ele, os chamados NVT (Network Vulnerability Tests) presentes no OpenVAS, são executados em conjunto, e são atualizados diariamente (FERNANDES, 2019).

Segundo Pastor (2022), o OpenVAS oferece recursos diversificados, tais como: verificação de vulnerabilidades

relacionadas a credenciais de acesso e falhas de segurança em sistemas que não requerem autenticação; descoberta de serviços e do sistema operacional em execução nos alvos; otimização para varreduras em larga escala; compatibilidade com grande variedade de protocolos, além da compatibilidade com o conjunto de ferramentas de segurança do OSSIM, expandindo sua capacidade de análise e correlação de vulnerabilidades, o que torna o OpenVAS uma ferramenta muito completa. Com todos esses recursos, é possível fazer uma varredura em computadores, servidores locais e remotos e ainda gerar relatórios detalhados contendo as vulnerabilidades encontradas, o nível de ameaça de cada uma e a descrição de como resolvê-las.

Para realizar a varredura de segurança em um *host*, é preciso atualizar a base de vulnerabilidades CVEs do OpenVAS. No processo de classificação de risco das vulnerabilidades, a ferramenta utiliza o padrão *Common Vulnerability Scoring System* (CVSS), que é utilizado nas avaliações do grau de risco oferecido por uma vulnerabilidade. Após a atualização, é necessário por meio da interface *web* configurar um alvo com os dados do *host* que será analisado. A ferramenta dispõe de várias opções para configuração, como por exemplo em relação ao grupo de portas que se deseja verificar e sobre o protocolo que o OpenVAS irá usar para fazer a comunicação com o *host* alvo. Em seguida, é preciso configurar uma tarefa, escolher o alvo configurado anteriormente e determinar a configuração de *scan* se vai ser completa ou parcial. Todas as tarefas criadas ficarão em uma aba do OpenVAS, junto com a porcentagem do progresso concluído. Depois que o teste for concluído 100% será possível abrir um relatório completo contendo as vulnerabilidades encontradas, portas abertas e qual o método deve ser feito para resolver as vulnerabilidades encontradas. Vale ressaltar que o número de falhas de segurança pode variar entre as configurações escolhidas para o teste (AZEVEDO et al.,

2022).

6 Metodologia

Para a elaboração desse trabalho, foi aplicado a metodologia de pesquisa experimental, sendo ela dividida nas seguintes etapas: criação do ambiente de testes, configuração da varredura de vulnerabilidades, coleta e relatórios dos dados, automatização do serviço OpenVAS, análise dos resultados e sugestões para mitigação das falhas encontradas utilizando os relatórios gerados pelo OpenVAS.

6.1.1 Criação do ambiente de testes

Para a montagem do ambiente de testes, foi realizada a instalação do OpenVAS em um notebook equipado com um processador Intel I3 de terceira geração, 4Gb memória RAM e com sistema operacional Kali Linux. O Kali é uma distribuição Linux com a proposta de facilitar os estudos e execução de práticas em segurança da informação. Ele conta com uma vasta biblioteca de programas de ataque, análise, exploração de serviços, aplicações e redes de computadores que já vem pré-instaladas no sistema.

Além da instalação do OpenVAS, as versões do ProxMox VE utilizadas nos testes foram adquiridas no site oficial dos desenvolvedores, depois foram instaladas em um disco rígido externo e executadas em um outro notebook equipado com um processador *Intel* I7 e 16Gb de memória RAM.

Para realizar a configuração do OpenVAS, foi aplicado a metodologia utilizada por Stefan em seu site Stefan (2022). A escolha dessa metodologia foi motivada pela abordagem prática e eficaz das informações contidas no site, além de ter sido eficaz para o propósito de implementar a ferramenta.

6.1.2 Configuração da varredura de vulnerabilidades

Para a realização das varreduras, o primeiro passo foi fazer a criação do alvo. Nessa parte foi configurado o endereço IP do servidor ProxMox VE. Na opção de lista de portas foi feito a escolha pela configuração "*All TCP and NMAP top 100*", para que sejam analisados o maior número de portas do alvo. A opção "*Alive Test*" no OpenVAS se refere a uma funcionalidade usada para determinar se um determinado *host* (computador ou dispositivo de rede) está ativo na rede alvo antes de iniciar a varredura de vulnerabilidades. É possível escolher o método que será usado para esse teste, nesse cenário foi determinado a opção "*Scan Config Default*.

As versões do ProxMox VE colocadas em teste foram: 6.4-1, 7.4-1 e a 8.0-1, e cada varredura foi executada separadamente. Foi criado um alvo para cada uma das versões, e cada alvo foi configurado da mesma forma para que a configuração não interferisse nos resultados. Após a configuração do alvo, foi feita a criação de uma tarefa para cada versão do ProxMox. Na tarefa é informado o alvo que será analisado, o nome do equipamento e configurado as opções *"Scanner" e "Scan config"*. Foi determinada a configuração "*OpenVAS Default*" na opção "*Scanner*", e escolhida a configuração "*Full and fast*" na opção "*Scan config*".

Para determinar a melhor configuração, foram realizadas diversas tentativas utilizando variações das opções de varredura disponíveis, e a configuração descrita anteriormente gerou o relatório de falhas mais completo em relação as demais que foram testadas. Além disso, a ferramenta apresenta opções que permitem personalizar as varreduras, sendo possível alterar o grupo de portas a serem verificadas, optar por descobrir apenas informações sobre o sistema operacional do alvo e configurar o grupo de endereços IP que serão analisados na rede.

6.1.3 Coleta e relatório de dados

Após a criação e configuração das tarefas, cada uma foi executada e a ferramenta inicia sua varredura. O tempo de término dos testes pode variar de sistema para sistema e depende de quais configurações foram configuradas. Cada tarefa em execução mostra seu progresso em forma de porcentagem. É possível verificar em um relatório quais falhas o OpenVAS já encontrou a partir dos 2% de progresso.

Foram realizados dois testes. Em um foi utilizado as ISOs do mesmo jeito que foram adquiridas no site oficial do ProxMox VE, já no segundo teste, foi realizado a execução do comando para atualizar os pacotes existentes nas três versões do ProxMox, dessa forma foi possível verificar se a atualização dos pacotes tem relevância ou não na quantidade de vulnerabilidades encontradas pelo OpenVAS.

Para a coleta dos dados, foi utilizado o relatório gerado pela própria ferramenta. Esse relatório informa: portas abertas, vulnerabilidades relacionadas à versão e autenticação, informação sobre qual sistema operacional está sendo executado, CVEs encontradas entre outras vulnerabilidades, seja em sistemas operacionais ou aplicações. Além disso, o relatório também informa o nível de gravidade de cada vulnerabilidade encontrada, isso ajuda a equipe de suporte a decidir as prioridades de melhoria da infraestrutura de rede.

6.1.4 Automatização do serviço OpenVAS

Para automatizar processos do servidor OpenVAS, tais como a atualização do sistema operacional e dos pacotes, atualização do OpenVAS e de seu banco de dados de vulnerabilidades, exclusão de pacotes não utilizados e a inicialização automática do serviço do OpenVAS nesse cenário, foi feito a implementação de um shell script, que foi configurado para ser executado juntamente com a inicialização do sistema.

O objetivo com essa automação é sugerir um método para facilitar a gerência e aumentar a praticidade da ferramenta OpenVAS, retirando a necessidade de o administrador de redes executar manualmente todos esses comandos. Além disso, é muito importante manter o OpenVAS e seu banco de dados atualizado, pois caso contrário, as varreduras poderão deixar de encontrar certas vulnerabilidades.

7 Aplicação dos testes de varredura no ProxMox VE e apresentação de resultados

7.1.1 Análise quantitativa das vulnerabilidades encontradas

Nos testes de varredura por vulnerabilidades foram utilizadas as versões 6.4-1, 7.4-1 e a versão 8.0-1. A escolha e objetivo por realizar o teste nessas versões do ProxMox VE foi para comparar a quantidade e a gravidade das vulnerabilidades encontradas entre essas versões, para que depois sejam discutidas métricas de mitigação e sejam evitados possíveis danos causados com a exploração maliciosa das falhas identificadas nos testes.

Após a varredura do OpenVAS ter sido executada pela primeira vez nas três versões, foi feita a comparação quantitativa das vulnerabilidades encontradas. Na versão 6.4-1 do ProxMox VE, foram encontradas 86 vulnerabilidades de gravidade alta, 43 de gravidade média, e 3 de gravidade baixa. Porém, na versão 7.4-1, a quantidade de falhas foi menor. Foram encontradas 10 vulnerabilidades de gravidade alta, 7 de gravidade média, e 3 de gravidade baixa. Na versão 8.0-1 foram encontradas 9 vulnerabilidades de gravidade alta, 6 de gravidade média e 3 de gravidade baixa, obtendo o menor número de vulnerabilidades entre as três, o que já é esperado sendo a versão mais atual entre elas.

Na **Tabela 1**, é possível visualizar a comparação da quantidade de vulnerabilidades encontradas em cada uma das três versões do ProxMox VE.

	6.4-1	7.4-1	8.0-1
Alta	**86**	**10**	**9**
Média	**43**	**7**	**6**
Baixa	**3**	**3**	**3**

Tabela 1 – Resultados da primeira varredura

7.1.2 Análise das vulnerabilidades encontradas no ProxMox VE 6.4-1

Ao analisar as vulnerabilidades da versão 6.4-1, das 132 falhas encontradas, 122 são do tipo DSA (Avisos de Segurança Debian), que segundo informações do site Debian (2023), se trata de um aviso de segurança contendo informações detalhadas sobre vulnerabilidades descobertas que afetam algum pacote Debian. Esse aviso contém informações sobre como resolver a falha de segurança, além disso os DSAs são publicados e podem ser visualizados no próprio site oficial do Debian.

Outra falha de segurança encontrada foi a *"ICMP Timestamp Reply Information Disclosure"*. Essa falha de segurança poderia ser usada para manipular geradores de números aleatórios, como geradores de chaves de criptografia. Isso pode resultar em chaves que são facilmente quebráveis ou reproduzíveis. Para mitigar esse problema, seria possível bloquear ou controlar os pacotes ICMP no servidor, especialmente os relacionados à carimbo de tempo, para isso pode ser realizado a criação de regras de firewall para filtrar e permitir apenas o tráfego ICMP necessário entre *hosts* confiáveis e bloquear o tráfego ICMP para redes consideradas não confiáveis.

A vulnerabilidade *"Weak MAC Algorithm(s) Supported (SSH)"* foi identificada. Segundo dados do site Virtue (2023),

essa falha está relacionada com métodos de geração de código de autenticação de mensagens (MAC) usados no protocolo SSH (*Secure Shell*), que são considerados fracos devido a certas características como a função de hash fraca ou tamanho da *tag* menor que 128 bits, por exemplo. Essa vulnerabilidade pode ser explorada por meio de ataques de força bruta, pois a senha pode ser quebrada mais facilmente e, para resolvê-la, é preciso manter o serviço SSH atualizado e configurar o servidor SSH para desativar explicitamente os algoritmos MAC considerados fracos ou inseguros. Isso pode ser feito no arquivo de configuração do SSH.

A vulnerabilidade "*TCP Timestamps Information Disclosure*", também foi identificada, segundo a análise do OpenVAS, ela pode expor informações sobre o tempo de atividade (*uptime*) do *host* remoto. Essa informação pode ser utilizada por possíveis invasores em planejamentos de ataques direcionados. Para sua mitigação é possível configurar a desativação dos pacotes TCP Timestamps.

A falha de segurança, denominada de "*SSL/TLS: Renegotiation DoS Vulnerability*", foi identificada nessa versão e ela permite que um invasor utilize as solicitações de renegociação manipuladas para esgotar os recursos do servidor SSL/TLS. Esse tipo de ataque poderia levar a uma indisponibilidade do serviço, interrompendo ou negando o acesso legítimo aos usuários. Para lidar com esse problema, até que ocorresse uma atualização de segurança, seria possível optar por desabilitar a renegociação no serviço SSL/TLS afetado por meio de configurações no servidor, porém vale ressaltar que ao desativar essa capacidade irá afetar a flexibilidade das conexões.

Além das vulnerabilidades citadas, o OpenVAS identificou como vulnerabilidade o fato de o ProxMox VE não estar em sua versão mais atual, identificando que a versão 6.4-1 chegou no final de sua vida útil. Também foi identificado a falta de algumas mitigações de falhas no *kernel* do sistema

operacional. Para resolver esse problema foi indicado algumas medidas como: atualizar a versão do *Kernel* Linux, instalar uma atualização de microcódigo e atualizar a BIOS da placa mãe. Também pode ser recomendável participar de discussões em fóruns sobre Linux, pois muitas pessoas compartilham a solução para vários problemas de segurança nessas plataformas.

7.1.3 Análise das vulnerabilidades encontradas no ProxMox VE 7.4-1

A varredura do OpenVAS realizada na versão 7.4-1 do ProxMox VE também identificou falhas do tipo DSA (Avisos de Segurança Debian), embora tenham sido encontradas apenas 13, uma significativa redução em relação às 122 falhas da mesma categoria encontradas na versão 6.4-1 do ProxMox VE. Parte das vulnerabilidades identificadas já existiam na versão anterior do ProxMox VE. As vulnerabilidades: *"Weak MAC Algorithm(s) Supported (SSH)"*, "*TCP Timestamps Information Disclosure*" e "*ICMP Timestamp Reply Information Disclosure*" que já foram discutidas anteriormente também foram identificadas na versão 7.4-1. A correção de falhas de segurança entre duas versões de uma aplicação deve ter uma atenção maior por parte dos desenvolvedores para que sejam solucionadas.

O OpenVAS identificou a ausência de 4 mitigações de *kernel* para as seguintes vulnerabilidades de *hardware*: *"Gather Data Sampling (Queda)", "L1TF - L1 Terminal Fault'", "MDS - Microarchitectural Data Sampling'"* e "*'Processor MMIO Stale Data*". Segundo informações do autor Masters (2018), a falha de segurança denominada *"L1TF - L1 Terminal Fault"*, pode ser explorada no Linux para comprometer a segurança do *kernel* e de outras aplicações, fazendo o uso da chamada de sistema "mprotect ( )" e criar uma entrada de tabela de página "não presente" para o endereço físico que desejar. Para mitigar essa falha, é possível definir que a criação da estrutura chamada PTE (Page Table Entry) pelo Linux, sejam escolhidas partes de

um endereço físico para sempre estar presentes, mesmo que a memória real correspondente não esteja lá. A mudança fará o processador identificar algo, mas na verdade está apontando para um lugar fora do alcance da memória real. Isso ajuda a evitar que um atacante consiga acessar informações sensíveis que não deveriam ser acessadas.

O relatório feito pelo OpenVAS indica que as demais vulnerabilidades encontradas podem ser mitigadas com a utilização da versão mais atualizada do *kernel* Linux, atualização da BIOS da placa mãe e com a atualização do microcódigo na CPU do dispositivo, que é um tipo de firmware do processador que tem a função de interpretar as instruções recebidas pela CPU e converter em ações físicas para que os circuitos internos da CPU possam executar (IPERIUS BACKUP BRASIL, 2019).

Manter o *kernel*, o microcódigo e a BIOS da placa mãe atualizados é muito importante, pois os recursos de segurança podem ser melhorados e atualizados. Manter o microcódigo da CPU atualizado pode ajudar na correção de falhas e defeitos na CPU, e no contexto das vulnerabilidades encontradas, essas atualizações são feitas para melhorar a forma como a CPU processa as informações, ajudando a evitar os ataques que exploram essas falhas de segurança.

7.1.4 Análise das vulnerabilidades encontradas no ProxMox VE 8.0-1

Ao realizar a análise da varredura no ProxMox VE 8.0-1, foi possível observar que foram encontradas 11 vulnerabilidades do tipo DSA (Avisos de Segurança Debian), duas a menos que sua versão anterior. Porém essa foi a única diferença entre as duas versões, pois o restante das vulnerabilidades encontradas na versão 7.4-1, também foram identificadas na versão 8.0-1 do ProxMox VE.

Na tentativa de resolver as vulnerabilidades de *hardware*

do sistema operacional, foi feita a atualização para a versão 6.2 do *kernel.* Após isso, a varredura foi realizada novamente, porém não houve mudanças nos resultados encontrados ao comparar com os resultados da varredura realizada antes da atualização do *kernel*. Para tentar diminuir as vulnerabilidades encontradas nas três versões do ProxMox VE, foi realizado a verificação mais detalhada das falhas do tipo DSA (Avisos de Segurança Debian). Nessa análise foi identificado que muitas delas notificaram a falta da atualização de um ou mais pacotes específicos, então em cada uma das três versões do ProxMox VE foram executados os comandos *"sudo apt update"* e *"sudo apt upgrade"*. Após a execução do comando, um segundo teste de varredura foi feito em cada uma das versões.

7.1.5 Comparativo das vulnerabilidades encontradas entre as duas varreduras

A versão 6.4-1, que no primeiro teste apresentava 86 vulnerabilidades de gravidade alta, após a atualização dos pacotes apresentou apenas uma, referente ao fim de sua vida útil. Apenas 6 vulnerabilidades de gravidade média foram identificadas, sendo 5 delas referentes a mitigações ausentes no *kernel* Linux e uma referente a vulnerabilidade de renegociação no SSL, além das 3 falhas de gravidade baixa que se mantiveram nos dois testes, na primeira varredura essa versão apresentou 43 vulnerabilidades de categoria média.

Em seu primeiro teste, foram encontradas 10 vulnerabilidades de gravidade alta, 7 de gravidade média e 3 de gravidade baixa na versão 7.4-1. Após a atualização dos pacotes, nenhuma vulnerabilidade de gravidade alta foi identificada, de gravidade média foram evidenciadas 3 falhas relacionadas a mitigações ausentes no *kernel* Linux e 3 vulnerabilidades de baixa gravidade, que também se mantiveram após a atualização dos pacotes.

Na versão 8.0-1, a quantidade de vulnerabilidades com

nível alto foi de 9 para 0, ao comparar os dois testes, além de identificar duas vulnerabilidades, a menos em comparação com a primeira varredura realizada, sendo os 4 referentes a falta de mitigação no *kernel* Linux. Porém, as 3 falhas de nível baixo, encontradas no primeiro teste, também se mantiveram na segunda varredura.

A **Tabela 2,** demonstra que a quantidade de vulnerabilidades encontradas na segunda varredura reduziu bastante se comparada com a primeira varredura realizada.

	6.4-1	7.4-1	8.0-1
Alta	**1**	**0**	**0**
Média	**6**	**4**	**4**
Baixa	**3**	**3**	**3**

Tabela 2 – Resultados da segunda varredura

Considerações finais

Ao final deste trabalho, considera-se que seu objetivo foi alcançado, uma vez que foram identificadas diferenças no nível de segurança nas versões do ProxMox VE que foram avaliadas pelo OpenVAS.

O estudo realizado do ProxMox VE (ProxMox Virtual Environment), demonstrou ser essa uma ferramenta de código aberto eficaz no gerenciamento e na virtualização de servidores. Sua interface intuitiva, flexibilidade e recursos de tolerância a falhas, tornam sua utilização viável para empresas que não querem investir em outros sistemas de virtualização comercializados. No entanto, embora seja altamente funcional, a questão da segurança exige a aplicação de medidas de segurança para esse ambiente, como a utilização de *softwares* para a varredura de vulnerabilidades.

A aplicação das varreduras e a análise detalhada das vulnerabilidades encontradas nas três principais versões disponíveis: 6.4-1, 7.4-1 e 8.0-1 do ProxMox VE, demonstrou uma tendência significativa de redução de falhas de segurança em versões mais recentes. Esse resultado demonstra a importância crítica de manter o ProxMox VE atualizado. Isso é importante para mitigar possíveis ameaças causadas por pacotes desatualizados e ressalta a relevância das correções de segurança na proteção dos sistemas.

Após a análise das vulnerabilidades encontradas nas diferentes versões do ProxMox VE, é perceptível que existe uma grande importância em realizar o monitoramento contínuo de vulnerabilidades em servidores. Esse processo de verificação e correção contínua é vital para garantir a segurança e a integridade dos sistemas virtualizados, especialmente considerando o cenário em constante evolução das ameaças cibernéticas.

Ao testar diferentes opções de varredura no OpenVAS, foi possível determinar a melhor configuração para o ambiente apresentado. A configuração correta permite o usuário obter as informações necessárias nas varreduras. No cenário apresentado, a intenção era obter o maior número de falhas de segurança possível, mas a ferramenta possui opções que entregam menos informações se necessário, por exemplo, é possível diminuir o número de portas analisadas ou optar por descobrir apenas o sistema operacional a ser executado no alvo.

Com a implementação da automação dos processos do OpenVAS, foi possível facilitar e garantir a atualização tanto da ferramenta OpenVAS, quanto do seu banco de dados de vulnerabilidades. A proposta e implementação abordadas neste trabalho se mostraram eficientes, pois remove a necessidade de realizar a execução manual dos comandos utilizados nesses processos, aumentando a praticidade da ferramenta.

Referências

Azevedo et al (2022) “AVALIAÇÃO DE VULNERABILIDADES EM ELEMENTOS DE AUTOMAÇÃO DE SUBESTAÇÃO”. Disponível em: https://repositorio.uninter.com/bitstream/handle/1/1276/Azevedo%2c%20Huliano%20F.%20M.%20de.pdf?sequence=1&isAllowed=y. Acesso em: 2 dez. 2023

AWS (2023) “O que é um Hipervisor?” Disponível em: https://aws.amazon.com/pt/what-is/hypervisor/. Acesso em: 24 out. 2023.

AWS (2023) “O que é virtualização? “Disponível em: https://aws.amazon.com/pt/what-is/virtualization/#. Acesso em: 1 dez. 2023.

BARROS et al (2019) “Virtualização de Servidores para otimização de recursos computacionais.” Disponível em: https://editoraessentia.iff.edu.br/index.php/citi/article/view/14729. Acesso em: 14 dez. 2023.

Darkcrizt (2023). “Proxmox VE 7.4 chega com grandes melhorias e atualizações. “Disponível em: https://blog.desdelinux.net/pt/proxmox-ve-7-4-llega-con-grandes-mejoras-y-actualizaciones/. Acesso em: 1 dez. 2023.

Debian (2004) “Debian Security Advisories.“ Disponível em: https://www.debian.org/doc/manuals/securingdebian-manual/dsa.en.html. Acesso em: 1 dez. 2023.

Ghannoum e Rodrigues (2018) “VIRTUALIZAÇÃO DE SERVIDORES: VANTAGENS E DESVANTAGENS.” Disponível em: https://revista.ueg.br/index.php/mirante/article/view/7612. Acesso em: 14 dez. 2023.

Häsler (2015) “Proxmox Celebrates 10-Year Anniversary.” Disponível em: https://www.proxmox.com/en/about/press-releases/10-years-of-proxmox. Acesso em: 1 dez. 2023.

IBM (2023) “O que são máquinas virtuais? |IBM.” Disponível em: https://www.ibm.com/br-pt/topics/virtualmachines.

Iperius Backup Brasil (2019) “O que são Microcódigos ou Firmwares e motivos para atualizá-los.” Disponível em: https://www.iperiusbackup.net/pt-br/o-que-sao-microcodigos-ou-firmwares-e-motivos-para-atualiza-los/#:~:text=Considere%20o%20microc%C3%B3digo%20como%20um. Acesso em: 4 dez. 2023.

Jacobsen et al (2015) “Um Estudo Sobre a Análise de Vulnerabilidades em Sistemas Computacionais – das Ferramentas ao Uso.” Disponível em: https://www.periodicos.unesc.net/ojs/index.php/sulcomp/article/view/1791. Acesso: 15 dez de 2023.

MACEDO E SANTOS (2014) “Hipervisor: Segurança em ambientes virtualizados.” Disponível em: https://www.devmedia.com.br/hypervisor-seguranca-em-ambientes-virtualizados/30993. Acesso em: 1 dez. 2023.

Masters (2018) “Understanding L1 Terminal Fault aka Foreshadow: What you need to know.” Disponível em: https://www.redhat.com/en/blog/understanding-l1-terminal-fault-aka-foreshadowwhat-you-need-know. Acesso em: 2 dez. 2023.

Mora (2022) “CAPACIDADES TÉCNICAS, LEGALES Y DE GESTIÓN PARA EQUIPOS BLUE TEAM Y RED TEAM. “Disponível em: https://repository.unad.edu.co/bitstream/handle/10596/51

776/odgomezmo.pdf?sequence=1&isAllowed=y. Acesso em: 2 dez. 2023.

Oliveira (2022) “Vantagens do Proxmox: por que utilizar este software?” Disponível em: https://nova.escolalinux.com.br/blog/vantagens-do-proxmox-por-que-utilizar-este-software-1. Acesso em: 2 dez. 2023.

Pastor (2022) “Estudio comparativo entre OpenVAS y Wazuh.” Disponível em: https://repositorio.upct.es/bitstream/handle/10317/11663/tfg-bar-est.pdf?sequence=1&isAllowed=y. Acesso em: 18 nov. 2023.

Red Hat (2023) “O que é um hipervisor?” Disponível em: https://www.redhat.com/pt-br/topics/virtualization/what-is-a-hypervisor. Acesso em: 1 dez. 2023.

SCHIMIGUEL e RIBEIRO (2016) “ANÁLISE DE DESEMPENHO DE HIPERVISORES NO CONTEXTO DOS SISTEMAS OPERACIONAIS WINDOWS E LINUX.” Disponível em: https://revistas.anchieta.br/index.php/RevistaEngenho/article/view/853. Acesso em: 14 dez de 2023.

SEO (2009) “Virtualização - Problemas e desafios.” Disponível em: https://www.ic.unicamp.br/~ducatte/mo401/1s2009/T2/008278-t2.pdf. Acesso em: 1 dez. 2023.

Silva (2019) “DSpace DECEx: Página de Busca.” Disponível em: https://bdex.eb.mil.br/jspui/simplesearch?query=ESTUDO+DE+CASO+DA+IMPLEMENTAÇÃO+DA+FERRAMENTA+PROXMOX+&sort_by=score&order=desc&rpp=10&etal=0&filtername=author&filterquery=Silva%2C

+Carlos+André+Rodrigues+da&filtertype=equals. Acesso em: 2 dez. 2023.

Smith (2023) “Versão estável do Proxmox VE 8.0 agora” disponível. Disponível em: https://www.storagereview.com/PT/NEWS/PROXMOX-VE-8-0-STABLE-RELEASE-NOW-AVAILABLE. Acesso em: 1 dez. 2023.

(STEFAN, 2022) Install OpenVAS on Kali Linux - Easy Step-by-Step Tutorial. Disponível em: < https://www.ceos3c.com/security/install-openvas-kali-linux/?expand_article=1>. Acesso em:12 nov. 2023.

Controle de acesso em ambientes institucionais utilizando tecnologia RFID: desenvolvimento e aplicação

João Vitor Bendo de Oliveira[1], Norton Santos Pereira[2] , Jeferson Mendonça de Limas[3], Sandra Vieira[4]

[1, 2,] Acadêmicos do Instituto Federal Catarinense – Campus Avançado Sombrio – 88960000 – Sombrio – SC – Brasil

[3, 4,] Docentes do Instituto Federal Catarinense – Campus Avançado Sombrio – 88960000 – Sombrio – SC – Brasil

joaobendo2012@gmail.com, norton.p@hotmail.com, jeferson.limas@ifc.edu.br, sandra.vieira@ifc.edu.br

Abstract. *This work proposes an access control system for the laboratories of the Instituto Federal Catarinense - Campus Sombrio, currently relying on a single key. The solution utilizes RFID technology and a web system connected to a NodeMCU ESP8266 microcontroller, developed with PHP. The system includes features such as user registration, RFID tag recording, and access control. Integration is achieved through the microcontroller's Wi-Fi. The project successfully met all specified requirements.*

Resumo. *Este trabalho propõe um sistema de controle de acesso para os laboratórios do Instituto Federal Catarinense - Campus Sombrio, que atualmente dependem de uma única chave. A solução utiliza tecnologia RFID e um sistema web conectado a um*

microcontrolador NodeMCU ESP8266, desenvolvido com PHP. O sistema inclui funcionalidades como cadastro de usuários, registro de tags e controle de acesso. O sistema é integrado pelo Wi-Fi do microcontrolador. O trabalho atingiu com êxito todos os requisitos.

1. Introdução

O avanço constante na área de Tecnologia da Informação (TI) proporcionou diversas inovações, dentre estas, a Internet das Coisas (IoT), que surgiu como uma necessidade de automatizar e otimizar processos, sendo capaz de integrar e conectar dispositivos ou "coisas" à internet, permitindo a coleta, troca e análise de dados de forma eficiente. Neste sentido, a IoT vem transformando a maneira como interagimos com a internet, e influenciando vários setores como a agricultura, com sensores de análise do solo; na saúde, com monitoramento remoto ou na segurança com sistemas de detecção e controle de acesso. (OLIVEIRA, 2017, p. 15).

Neste contexto, a segurança física ganha destaque, especialmente em locais como o Instituto Federal Catarinense - Campus Sombrio (IFC-Campus Sombrio), onde a necessidade de criar um controle de acesso se tornou evidente. Atualmente, a instituição enfrenta desafios com uma única chave para todos os laboratórios, resultando em possíveis transtornos e outras vulnerabilidades.

Diante dessa realidade, propõe-se um sistema de controle de acesso de baixo custo baseado em tecnologia RFID para os ambientes do Instituto, com foco nos laboratórios, onde encontra-se o maior obstáculo. A ausência de um controle efetivo pode acarretar problemas como acesso não autorizado, interrupção de atividades acadêmicas e inconveniências no caso supracitado, como dois ou mais docentes necessitarem da chave no mesmo momento. Desta forma, o sistema proposto visa

oferecer flexibilidade aos usuários, monitoramento eficaz e segurança, a fim de reduzir riscos de acesso não autorizado e proporcionar benefícios em termos de organização e agilidade.

O presente trabalho consiste em um sistema web para monitorar e registrar os acessos, gerenciar usuários, dispositivos e registros, interligado a um microcontrolador com sensor RFID para controlar o acesso físico. A escolha da tecnologia RFID se deu por seu menor custo em comparação a outras opções, como biometria, reforçando a viabilidade econômica do projeto.

Em resumo, este trabalho busca abordar a importância da segurança física em instituições educacionais, apresentando uma solução prática e funcional para o controle de acesso, a fim de contribuir para a integridade dos ambientes e a otimização dos recursos institucionais.

3. Fundamentação Teórica

Aqui serão descritos os equipamentos e tecnologias utilizadas neste trabalho.

3.1 Sistemas Embarcados

Sistemas embarcados podem ser definidos como microcontroladores interligados a outros sistemas para desempenhar funções específicas. Esses sistemas são cuidadosamente projetados para executar tarefas dedicadas, porém com recursos limitados, o que contrasta com computadores pessoais, que são desenvolvidos para executar várias tarefas.

Segundo CUNHA (2007):

> Diferente dos computadores, que rodam sistemas operacionais como base para que outros aplicativos diversos sejam instalados e utilizados (cada um para uma aplicação diferente), os

> sistemas embarcados são construídos para executar apenas uma tarefa pré-determinada.

Bons exemplos de sistemas embarcados são as máquinas de lavar roupas ou televisões, que se enquadram nesta categoria, pois são dedicados a realizar tarefas limitadas, como reproduzir canais de TV, ou lavar e secar roupas (SILVA et al. 2018).

Assim como as arquiteturas tradicionais de sistemas de computação, sistemas embarcados também precisam de entrada, processamento e saída de dados. Neste sentido, pode-se dividi-los em dois tipos, reativos e em tempo real. No reativo, o funcionamento é influenciado por fatores externos, ou quando o usuário interage com o sistema. Em tempo real, há restrições temporais para execução das tarefas, como leitura de sensores, ou emissão de sinais para atuadores, no entanto esse modo de operação é cíclico e pode tomar decisões mesmo na ausência imediata de sinais de entrada para realizar as atividades (CUNHA, 2007).

Geralmente esses sistemas são caracterizados por possuírem um tamanho compacto e baixo consumo de energia, tornando-se adequados a serem alimentados por baterias ou fontes de energia limitada. E no âmbito de compacidade, são projetados para ocupar o menor espaço possível, diferente de computadores pessoais, que podem ser relativamente grandes.

O hardware microcontrolador foi implementado seguindo estes conceitos.

3.2 NodeMCU - ESP8266

O NodeMCU é uma plataforma de desenvolvimento de código aberto integrado com chip ESP8266. que conta com módulo Wi-Fi, possui conversor USB serial integrado e suporta os protocolos mais recentes, como TCP/IP, sendo um microcontrolador versátil para uma variedade de aplicações.

Esse microcontrolador é desenvolvido em linguagem C, e permite a comunicação através de GPIO (General Purpose Input/Output), que são portas responsáveis por fazer a comunicação de entrada e saída de dados, recebendo funções via programação. Além disso existe a possibilidade de realizar a programação via OTA (Over the Air), através da comunicação Wi-Fi (OLIVEIRA, 2016).

Para melhor compreensão das partes do microcontrolador, a Figura 1 a seguir, mostra uma breve ilustração da placa NodeMCU:

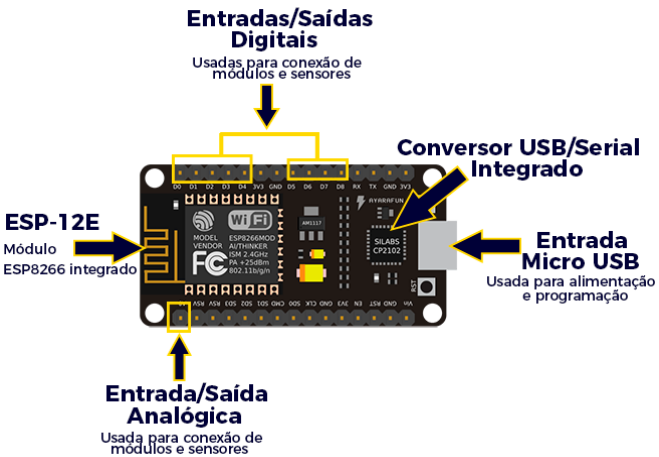

Figura 1 - NodeMCU com ESP8266

A implementação é bastante facilitada, podendo ser realizada de maneira simples, usando a linguagem de programação LUA ou a IDE (plataforma de desenvolvimento) do Arduino. Isso é possível adicionando-se uma extensão ou complemento diretamente na plataforma Arduino, e a programação pode ser realizada de forma simples através de cabo micro-USB. Pode-se citar vantagens também como baixo

custo e baixo consumo de energia, sendo encontrado em torno de 5 dólares em marketplaces, e facilmente disponível no Brasil.

A escolha do NodeMCU se deu devido à sua fácil implementação e à capacidade de programação na IDE Arduino, utilizando principalmente as mesmas linguagens, possuir comunicação Wi-Fi, ter um custo acessível e curva de aprendizagem rápida.

3.3 RFID (Radiofrequency Identification)

A tecnologia RFID surgiu na década de quarenta, durante a Segunda Guerra Mundial. Na ocasião, havia a necessidade dos militares para identificação de aeronaves consideradas "inimigas ou amigas". No entanto, enfrentavam problemas a serem solucionados. A capacidade de reconhecimento das aeronaves era limitada, e existia a dificuldade de identificar a origem do objeto capturado pelo sensor. "A simples descoberta da presença da aeronave não garantia estarem livres de ataques, pois muitas vezes havia falha na transmissão dos dados de identificação. Não se sabia de que lado estava a aeronave, se ela era inimiga", explica (GREFF, 2009, p.14).

Com o passar das décadas e o decorrer da evolução da tecnologia, o sistema de RFID tomou direções opostas ao uso bélico, para o qual inicialmente foi projetado. Instituições, no entanto, iniciaram o processo de padronização da tecnologia para fins de uso e estudo, conforme discutido por Pinheiro (2006):

> Massachusetts Institute of Technology (MIT) e outros centros de pesquisa no mundo iniciaram o estudo de uma arquitetura de identificação automática que utiliza os recursos da transmissão por radiofrequência para servir como modelo no desenvolvimento de novas aplicações de rastreamento e localização de produtos.

O RFID (Radio Frequency Identification - identificação por radiofrequência) é um método de comunicação e transferência de dados via ondas de rádio. Essas ondas são eletromagnéticas e possuem frequências menores que a frequência da luz visível, estando compreendidas no intervalo de 3 KHz a 300 Ghz. O RFID utiliza faixas de frequência estabelecidas e padronizadas por organizações para operar a comunicação entre leitor, antena e tag. A ISO e a EPCglobal são as duas organizações mais envolvidas no desenvolvimento de padrões para tecnologia RFID. “Os padrões ISO e EPCglobal representam um papel crucial no desenvolvimento da tecnologia RFID, proporcionando compatibilidade e interoperabilidade entre os diversos componentes envolvidos em um sistema RFID” (ZANLOURENSI, 2011, p.10).

Essa tecnologia consiste em uma combinação de etiquetas e leitores: as etiquetas armazenam e transmitem dados para os leitores usando ondas de rádio. Os leitores coletam dados fornecidos pelas etiquetas e os enviam de volta para o servidor para análise e processamento.

A identificação por radiofrequência traz consigo uma grande variedade de aplicações. Segundo OLIVEIRA e PEREIRA (2006, p.7), "O RFID pode ser encontrada em diversas áreas, como controle de acesso, controle de tráfego de veículos, lavanderia, indústrias, controle de contêineres, monitoração de pacientes, identificação de animais, monitoração de bagagem e passageiros nos aeroportos."

Pode-se dividir o funcionamento do RFID em três partes principais: sensor ou leitor, tag RFID e Antena.

A ilustração na Figura 2, mostra alguns dos componentes do RFID:

Figura 2 - Componentes do RFID

Os leitores são dispositivos que emitem sinais de radiofrequência para ativar as tags ou etiquetas nas proximidades e coletar dados delas. Quando um leitor envia um sinal, as etiquetas RFID dentro do alcance desse sinal são ativadas e começam a transmitir seus dados de identificação e outros dados armazenados. "Uma vez que os sinais do receptor sejam corretamente recebidos e decodificados, são usados algoritmos para decidir se o sinal é uma repetição de transmissão de uma etiqueta" (ZANLOURENSI, 2011, p.8).

Após a verificação dos dados transmitidos pela tag ou etiqueta, o leitor tem a responsabilidade de realizar a autenticação da mesma e garantir a integridade dos dados fornecidos. Consequentemente, faz o tratamento dessa informação de forma correta com o respectivo uso da mesma, seja para controle de acesso, identificação de objetos ou pessoas ou qualquer função da variedade de possibilidades que a tecnologia oferece.

No contexto das tags, é possível classificá-las de acordo com seu funcionamento e com a proposta desejada. Existem dois tipos de tags: Ativas ou Passivas.

As tags ativas têm uma bateria embutida que desempenha duas funções diferentes. Primeiro, alimenta o circuito da etiqueta. Segundo, fornece energia para a etiqueta enviar sinais para a antena da estação base. A vantagem de uma tag ativa em relação a passiva é o maior alcance que esta proporciona. A desvantagem fica por conta do maior tamanho da etiqueta e do custo um pouco mais alto.

Em contraste às tags ativas, tags passivas não possuem bateria própria. Elas obtêm a energia que precisam do sinal enviado pela antena/base leitora. Isso significa que elas dependem do sinal da base para transmitir suas informações. "A tag passiva contém, normalmente, memória do tipo Read Only Memory (ROM) e apenas responde ao sinal emitido pela antena ligada ao leitor" (TEIXEIRA, 2011, p.22).

O próximo componente são as antenas, que por sua vez, são responsáveis por transmitir o sinal de radiofrequência entre a etiqueta RFID e o leitor RFID. As antenas podem ser internas ou externas. Possui (quem? As antenas? Então o verbo no plural)a capacidade de realizar a leitura e escrita de uma informação, e principalmente responsável(veis?) pela troca de informações entre o tag e o leitor.

O exemplo utilizado na construção deste projeto é o Módulo RFID RC522 compatível com microcontroladores. O RC522 tem a capacidade de ler e escrever nas tags ou etiquetas de acordo com a norma ISO/IEC 14443. Isso significa que o chip é capaz de interagir com esses cartões de forma sem fio, usando ondas eletromagnéticas.

No âmbito deste projeto, a tecnologia RFID foi utilizada com etiquetas passivas e encarregada da identificação dos usuários vinculados ao sistema. Cada usuário será atribuído a

uma etiqueta relacionada ao seu registro no sistema, possibilitando a liberação de acesso ao ambiente físico das salas.

3.4 Trabalhos Relacionados

Alguns estudos relacionados à aplicação da tecnologia RFID para controle de acesso foram previamente publicados, destacando-se aqueles que desempenharam um papel mais relevante na fundamentação e contextualização do presente trabalho.

Bianca Stephanie Guimarães Morais contribuiu significativamente com o trabalho intitulado "*SISTEMA DE CONTROLE DE ACESSO UTILIZANDO TECNOLOGIA RFID - GPACESS*". Neste projeto, propôs-se a criação de um sistema que integra a tecnologia RFID com o microcontrolador NodeMCU ESP8266, visando fortalecer a segurança e monitorar o acesso para preservar o patrimônio do ambiente. O estudo foi conduzido com a utilização de tags RFID nos objetos do laboratório, com a perspectiva de implementação no Grupo de Pesquisa em Sistemas Críticos de Segurança (GPSiCS) da Universidade Federal Rural do Semi-Árido (UFERSA), localizado no Centro Multidisciplinar de Pau dos Ferros. Os resultados alcançados pelo sistema embarcado foram bem-sucedidos em atender às funcionalidades planejadas, evidenciando êxito em todos os serviços. O sistema web, por sua vez, atendeu a todos os requisitos estabelecidos.

Conforme Tiago Teixeira, publicado em 2011 em o "*CONTROLE DE FLUXO DE PESSOAS USANDO RFID"*, é proposto a elaboração de um aplicativo para monitorar o fluxo de pessoas e demonstrar a viabilidade do uso de um sistema RFID. O autor enfrentou desafios durante o desenvolvimento do aplicativo, destacando-os em seu trabalho juntamente com os resultados obtidos. No decorrer do projeto, foi possível superar as dificuldades, culminando na bem-sucedida implementação do aplicativo com todas as funcionalidades previamente projetadas.

Este estudo evidencia que, com o devido conhecimento sobre o assunto, a implementação do RFID torna-se mais intuitiva e eficiente.

A partir destes trabalhos, foi possível idealizar o desenvolvimento do projeto em questão, nos modelos propostos, há abordagens um pouco distintas sobre o assunto, porém com finalidades relacionadas, o uso do RFID para controle de acesso. Esses artigos foram essenciais para o embasamento científico na construção do projeto, trazendo várias ideias, e sendo possível desenvolver melhorias quanto aos trabalhos já existentes, com objetivo de criar algo de característica própria, porém tendo estes artigos como inspiração para o seguimento do trabalho.

3.5 Escolha do RFID

A escolha do RFID como controle de acesso, bem como seus componentes e o microcontrolador NodeMCU, foi fundamentada por uma série de razões que se alinham aos objetivos e requisitos do projeto. Algumas das considerações para esta etapa, foram:

- Custo acessível, onde acessibilidade financeira é crucial, considerando a aplicação em um ambiente educacional.

Desta forma, a partir da Tabela 1, sugere-se o seguinte orçamento obtido através de pesquisas em marketplaces:

Tabela 1 - Orçamento

Item	Descrição do Item	Quantidade	Preço Un.	Total
NodeMCU ESP8266 v3	Microcontrolador utilizado no projeto	1	R$ 32,00	R$ 32,00
Módulo Leitor Rfid Mfrc522	Responsável pela Leitura dos cartões	1	R$ 17,00	R$ 17,00
Tranca Elétrica	Responsável pela segurança	1	R$ 169,00	R$ 169,00
Protoboard	Interligação dos componentes	1	R$ 15,00	R$ 15,00
Jumpers	Fazer as ligações	40	R$ 0,37	R$ 14,80
Total				R$ 247,80

Além das motivações de acessibilidade de custo, há outras como:

- A facilidade de implementação, para garantir uma adoção suave, especialmente em ambientes onde a infraestrutura possa precisar ser implementada ou atualizada.
- Desempenho comprovado: conforme abordado nos trabalhos relacionados ao longo deste artigo em que se demonstrou seu desempenho confiável, visto os resultados obtidos, e também a disponibilidade de recursos e pesquisas sobre a tecnologia utilizada;

Segundo Loureiro et al. (2015), as vantagens do RFID incluem maior confiabilidade, capacidade de reduzir ou eliminar erros humanos, durabilidade das etiquetas passivas e ativas, também ressalta a possibilidade de reutilização, além da otimização de processos resultantes da implementação da tecnologia RFID. Contudo, conforme os autores mencionam no artigo “RFID Identificação por Rádio Frequência” publicado em 2015, existem desafios a serem superados, como a interferência

por metais, que há possibilidade de queda de performance se possui algum obstáculo magnético, e além disso, apesar de apresentar um custo mais barato em relação à outras soluções mais sofisticadas, possui um custo mais alto que soluções em códigos de barras.

Diante disso, a compreensão abrangente dessas perspectivas proporciona uma visão crítica e equilibrada sobre a decisão de utilizar tecnologia RFID como proposta.

4. Metodologia

A pesquisa tecnológica pode ser realizada utilizando-se de várias metodologias, e o DSRM (Design Science Research Methodology) ou Ciência Design, é uma delas. Em muitos casos, a pesquisa tecnológica envolve o desenvolvimento de soluções práticas ou sistemas que podem se beneficiar da aplicação do DSRM. A metodologia oferece uma estrutura para a criação sistemática de soluções inovadoras e sua avaliação ocorre em contextos do mundo real.

Este trabalho empregou o método do DSRM. Esta metodologia constitui uma abordagem específica voltada para investigações em sistemas de informação e ciência da computação, concentrando-se na criação e elaboração de artefatos. Esses artefatos podem abranger sistemas, modelos, métodos ou teorias, sendo a pesquisa direcionada à resolução de problemas práticos e de soluções aplicáveis.

Conforme discutido por Freitas Júnior (2016): “Assim como definido em pesquisa tecnológica, o artefato previsto pela DSRM não necessariamente é um objeto concreto, mas um constructo, um modelo ou mesmo um método.”

Dessa maneira, no presente estudo adaptou-se integralmente a ordem proposta por Peffers *et al.*, que descreve seis etapas (apud Junior *et al*., 2016). Segmenta-se, então, neste projeto, cinco etapas. Na Figura 3 abaixo, apresenta-se

visualmente o fluxo e a progressão dessas etapas, fornecendo uma visão clara do desenvolvimento e implementação do sistema de controle de acesso.

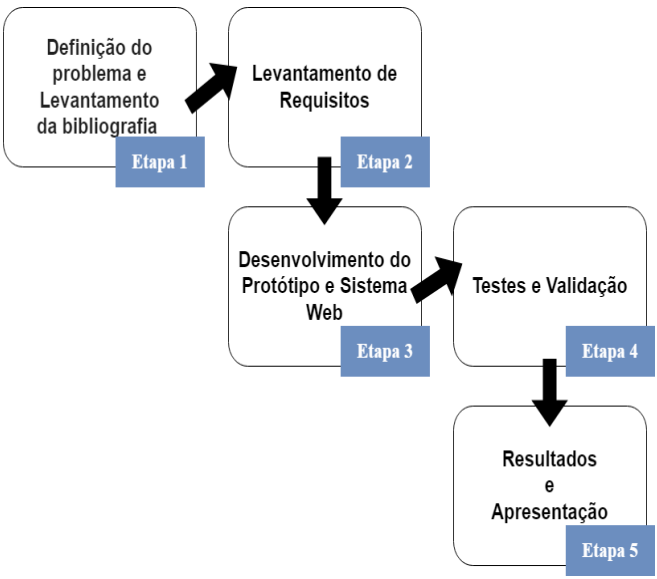

Figura 3 - Etapas do processo metodológico

A primeira etapa iniciou-se com a definição do problema. O objetivo foi compreender a dinâmica atual da segurança nos laboratórios e salas do IFC-CAS. A problemática central que emergiu desse levantamento foi a distribuição de uma única chave para todos os laboratórios, resultando em possíveis transtornos e outras vulnerabilidades. Após a definição do problema, foi realizado o levantamento bibliográfico, empregando a metodologia de pesquisa exploratória, fazendo-se consultas a artigos científicos, livros e blogs especializados. Essa etapa foi essencial para a construção do embasamento teórico e escolha do hardware que fundamentou o desenvolvimento do sistema.

Na segunda etapa, já com o problema evidenciado, buscou-se então o levantamento dos requisitos necessários para a elaboração do artefato, desta forma foi decidido quais os componentes, ferramentas e linguagem de programação seriam utilizados para a construção do sistema, e também a viabilidade de implementação do projeto.

A terceira e quarta etapa do projeto foram divididas e elaboradas em conjunto, sendo o desenvolvimento do protótipo, testes e validação para obter a versão final do sistema. Durante esse processo, integrou-se hardware e software, realizando-se uma série de testes e avaliações. Essas avaliações verificaram a viabilidade de conexão com o servidor web e as demais funções e também foram corrigidos vários bugs e problemas de lógica de programação.

Também, foi realizada uma minuciosa averiguação em um ambiente de teste controlado, buscando-se alcançar a máxima aproximação com o resultado desejado. Isso destaca a eficácia do sistema em condições simuladas, aproximando-nos ao máximo do desempenho esperado.

Na última etapa encontra-se a versão finalizada do sistema de controle de acesso. Desta forma, a etapa cinco, dos resultados e apresentação, será discutida no próximo tópico.

5. Resultados e Discussões

Nesta seção é delineada as fases e desdobramentos dos resultados obtidos por meio da execução deste projeto. A análise a seguir oferece uma visão abrangente e detalhada das diversas etapas pelas quais o desenvolvimento do sistema de controle de acesso passou, revelando os desafios e os benefícios observados durante seu progresso.

5.1 Implementação

O desenvolvimento do protótipo de controle de acesso foi segmentado em três fases: construção do protótipo físico, criação do sistema web e integração entre o hardware e o software.

Na primeira etapa da construção, optou-se por executar testes de funcionamento básico em relação ao hardware, com o objetivo de verificar se apresentava o funcionamento de forma correta e averiguar possíveis problemas que poderiam surgir durante o desenvolvimento.

Para realizar a comunicação entre o sensor RFID e o microcontrolador NodeMCU, foi utilizada a plataforma de programação IDE disponibilizada pela própria Arduino, que, em sua essência, utiliza a linguagem de programação C e C + +. Para fins de teste, executou-se funções simples, como leitura e reconhecimento dos cartões RFID, com o intuito de identificar se a leitura ocorreria de forma correta. Com todos os testes realizados para determinar o funcionamento do conjunto de hardware, passou-se, então, para o desenvolvimento do sistema web.

5.1.1 Protótipo

A montagem do hardware foi uma etapa fundamental na concretização do sistema de controle de acesso, desempenhando um papel central na materialização dos objetivos propostos. Ao unir os componentes físicos, como o microcontrolador NodeMCU e o sensor RFID MFRC 522, construiu-se a espinha dorsal do sistema. Neste estágio, não apenas conectou-se o hardware, mas foi moldada a base sobre a qual a segurança e eficácia do controle de acesso foram estabelecidas.

No contexto da programação do hardware, mencionada anteriormente, optou-se pelo desenvolvimento utilizando a linguagem de programação C, a qual é amplamente utilizada na

IDE do Arduino. Essa escolha foi motivada pela familiaridade e eficácia dessa linguagem para programação de microcontroladores. Para garantir o funcionamento eficiente e uma comunicação sem complicações, foram empregadas algumas bibliotecas de hardware disponíveis gratuitamente.

A comunicação entre o NodeMCU e a internet é gerenciada pelo módulo WiFi ESP8266 integrado à placa. Para essa finalidade, foi utilizada a biblioteca ESP8266 WiFi, que facilita a configuração e gerenciamento da conexão WiFi. Além disso, em conjunto com essa biblioteca, incorporou-se a WiFiManager, uma ferramenta que possibilita a abertura de um portal de configuração. Esse portal permite ao usuário conectar o dispositivo à rede WiFi desejada e identificar o endereço IP do servidor de forma intuitiva.

A seguir uma ilustração do aplicativo WiFiManager, na Figura 4:

Figura 4 - Tela do WiFi Manager

A comunicação entre NodeMCU ESP8266 e o módulo MFRC522, utilizado para leitura de cartões RFID, geralmente ocorre por meio da interface SPI (Serial Peripheral Interface), em que ambos os dispositivos, o NodeMCU ESP8266 e o módulo MFRC522, precisam ser configurados corretamente. A ESP8266 é configurada para atuar como um mestre SPI, enquanto o MFRC522 é configurado como um dispositivo escravo. O NodeMCU ESP8266, atuando como mestre, envia comandos para o módulo MFRC522 usando a interface SPI. Esses comandos podem incluir instruções para iniciar a leitura de cartões, verificar a presença de cartões, ou realizar operações específicas no módulo. O NodeMCU ESP8266 processa os dados recebidos do módulo MFRC522 de acordo com a lógica do sistema.

A Figura 5 demonstra a interligação entre os componentes de hardware e ilustra o esquema do protótipo.

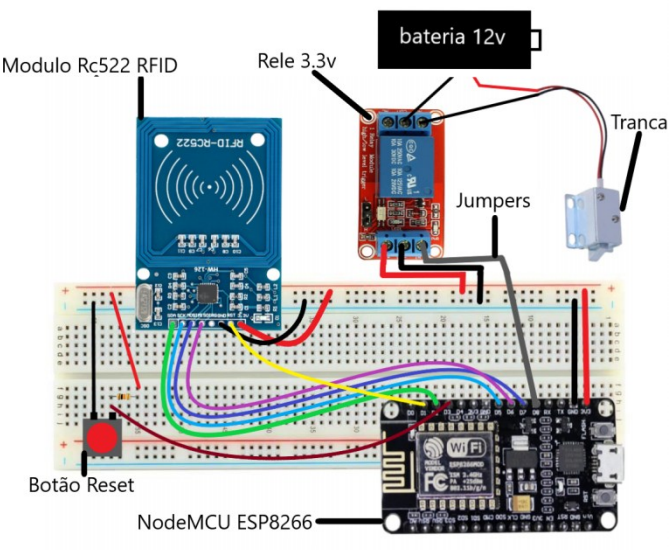

Figura 5 - Sketch do Protótipo

Com uma breve introdução ao funcionamento do sensor RFID em conjunto com o microcontrolador NodeMCU ESP8266 v3, é possível agora esclarecer o processo operacional do projeto apresentado. Após a conexão física entre o sensor e o microcontrolador, iniciou-se o desenvolvimento da comunicação entre hardware e servidor, essencial para estabelecer o funcionamento adequado da ferramenta.

A comunicação entre o microcontrolador e o servidor ocorre por meio de uma solicitação HTTP Post direcionada a um link ou caminho previamente definido na programação. Este link é responsável por encaminhar as informações enviadas para um arquivo PHP, que, por sua vez, é encarregado de processar essas informações conforme designado no código do sistema web. Esse arquivo PHP desempenha um papel crítico ao armazenar o ID da tag lida pelo sensor RFID. Além disso, quando solicitado, o PHP também registra o Mac Address do dispositivo, ampliando as informações armazenadas e permitindo uma gama mais completa de funcionalidades no sistema. Este processo, portanto, cria uma interligação dinâmica entre o microcontrolador, sensor RFID e o sistema web, proporcionando uma base sólida para o controle de acesso e registro de atividades no ambiente.

No entanto, reconhecendo a possibilidade de enfrentar eventualidades, o sistema de hardware foi projetado com uma funcionalidade de resetar incorporada em sua configuração. Essa característica permite que, em situações não previstas, o sistema seja reconfigurado, possibilitando assim a retomada do funcionamento adequado do projeto.

O protótipo do sistema em sua fase final é apresentado na Figura 6:

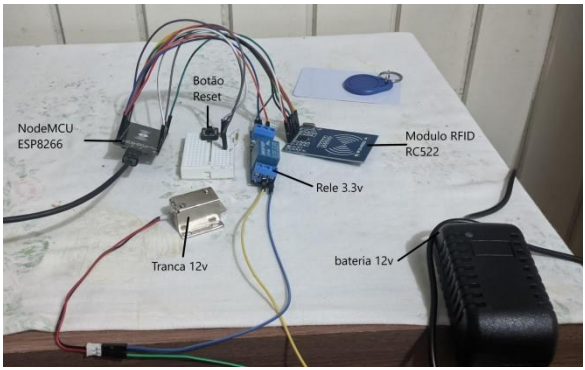

Figura 6 - Protótipo final

5.1.2 Sistema WEB

A fase de desenvolvimento web envolveu a criação de um sistema web capaz de gerenciar usuários, dispositivos e monitorar acessos. Além disso, o sistema é projetado para fornecer relatórios específicos aos ambientes que implementam o protótipo. Desta forma, foram conduzidos levantamentos detalhados para identificar os requisitos necessários para atingir os objetivos delineados no projeto.

5.1.2.1 Requisitos Funcionais

- Integração com Dispositivos de Controle de Acesso:
    - Garantir a comunicação entre microcontrolador NodeMCU com o sensor RFID MFRC 522 e a trava de segurança;
- Gestão de usuários:
    - Incluir no sistema uma fácil implementação de adição, edição e exclusão de usuários ao sistema, e associação destes aos ambientes aos quais serão garantidos os acessos;
- Gestão de Ambientes:

    - Gerenciar os ambientes associados às travas eletrônicas utilizando *MAC Address* como identificador único.

- Autorização e Controle de Acesso:
    - Implementar controle de acesso aos ambientes no qual estarão instalados os sistemas de segurança, responsável pela verificação se determinado usuário faz parte do ambiente.
- Registro de Atividades (Log):
    - Oferecer um monitoramento eficiente de registro dos acessos e tentativas de acesso que ocorreram durante o funcionamento.

Na construção do sistema web, baseado em estudos, análises, conhecimentos da área e extensa pesquisa, foram identificadas quais ferramentas seriam mais úteis e proporcionariam maior funcionalidade ao que se está desenvolvendo. Dentre as escolhidas para o desenvolvimento web estão: HTML (HyperText Markup Language - Linguagem de Marcação de HiperTexto), PHP, Javascript, JQuery, JSON, e também a utilização da ferramenta de banco de dados MySQL.

Nos primeiros momentos, foi desenvolvido um sistema simples de cadastro de usuários e edição, com o objetivo de testar o funcionamento e a comunicação com o banco de dados. Após estabelecer a comunicação entre o banco de dados e o sistema web, implementou-se as demais funcionalidades. Realizou-se não apenas a implementação do cadastro de usuários, mas também o cadastro de ambientes ou salas que seriam associadas a travas eletrônicas, identificadas através do seu MAC Address. Durante esse processo, implementou-se a associação do usuário ao dispositivo de trava.

A parte de estrutura e programação do sistema foi desenvolvida utilizando-se a linguagem de programação PHP.

Isso inclui tanto a construção do sistema quanto a comunicação com o banco de dados. Para efetuar o processamento dos dados recebidos através do NodeMCU, foi utilizado jQuery. Esta biblioteca ficou responsável por realizar todo o tratamento da informação, como o recebimento do ID (número de identificação do cartão) e o processamento do endereço MAC do dispositivo atualmente em uso. Dessa forma, o jQuery é encarregado de executar as funções na página web, tais como a leitura do ID e sua colocação no campo correspondente. Além disso, ele também desempenha um papel crucial na identificação da tag.

Nesse processo, verifica-se a existência da tag no banco de dados, a associação a algum usuário e, consequentemente, se o usuário pertence ao ambiente específico. Essa abordagem, utilizando PHP para a lógica de servidor e jQuery para o processamento dinâmico na interface web permitiu uma implementação eficiente e integrada, assegurando o correto funcionamento das operações relacionadas à identificação de cartões RFID e controle de acesso.

5.1.3 Distribuição de Telas do Sistema e Funcionamento

A abordagem a seguir destaca os principais pontos relacionados às telas e ao funcionamento do sistema web desenvolvido, trazendo as partes que são essenciais e com maior propósito no funcionamento do sistema web.

Na Figura 7 encontra-se a primeira etapa, a tela de login, responsável por tornar o acesso às configurações do sistema exclusivas ao administrador, geralmente o responsável pelo TI. Desta forma, mesmo que alguém encontre ou acesse o servidor do sistema, necessitará das credenciais do responsável para o acesso.

Figura 7 - Tela de Login

Na Figura 8 e Figura 9 ilustra-se a tela de cadastro de usuários e registros de acesso respectivamente, duas das partes essenciais do sistema. A primeira responsável por registrar novos usuários e a segunda por monitorar todos os acessos concebidos.

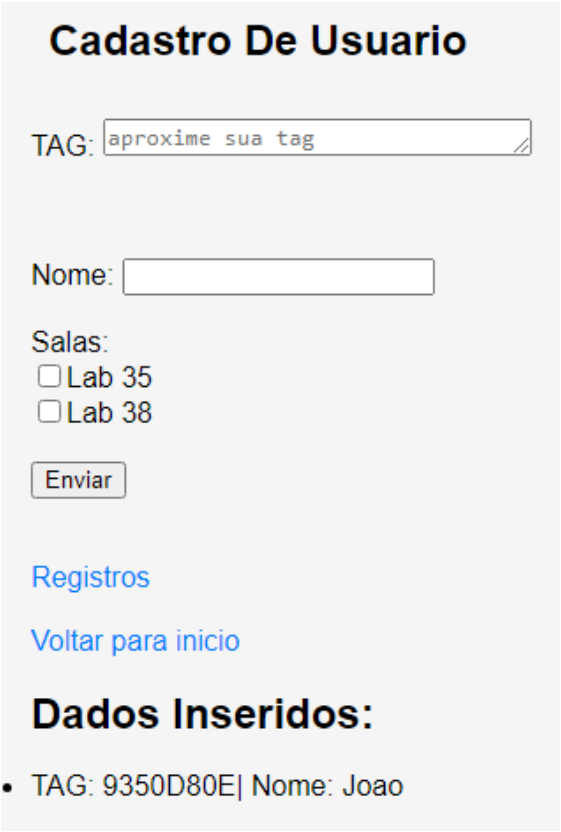

Figura 8 - Tela de cadastro de usuários

Figura 9 - Tela de Registros de Acesso

Por fim, na Figura 7, encontra-se a tela de leitura, seção fundamental para o sistema, incumbindo-se do controle de acesso dos usuários e dispositivos, além de encaminhar as informações para a tela de registros. Aqui, o sistema permanece constantemente à espera da leitura da tag para executar as funções previamente programadas, conforme exemplificado no artigo.

Assim que uma solicitação é recebida, inicia-se o processo de verificação. Primeiro, se a tag existe e está associada a um usuário e, em seguida, se está vinculada ao dispositivo solicitado.

Figura 7 - Tela de Registros

No final do processo, além das telas expostas, o sistema de controle de acesso desenvolvido também é capaz de filtrar os registros, gerar relatórios em PDF e cadastrar os dispositivos (microcontroladores).

6. Considerações Finais

A partir do trabalho exposto, a utilização da tecnologia RFID torna-se uma solução viável para a construção de projetos relacionados também ao IoT. Por ser uma tecnologia com alto potencial, ela permanece em constante evolução, como explica Zanlourensi (2011, p.49): “A tecnologia está amadurecendo, as padronizações estão acontecendo, conforme a evolução dos padrões ISO e EPCglobal, mas a tecnologia do RFID ainda tem que superar algumas adversidades para definitivamente ser aceita com sucesso no mercado”.

Com todas as informações reunidas sobre a tecnologia RFID, as escolhas definidas para a construção desse artigo e do projeto do protótipo estão diretamente relacionadas à facilidade de implementação, usabilidade e baixo custo da tecnologia. Por isso, foi estabelecido o uso do sensor RC522 e a utilização das tags passivas, ou seja, sem necessidade de alimentar as tags com bateria, deixando assim um menor custo, juntamente com o uso do microcontrolador NodeMCU ESP 8266 v3.

Neste trabalho foi desenvolvido um sistema de controle de acesso baseado em RFID e de baixo custo para solucionar a problemática de haver uma única chave para todas as salas dos laboratórios, e também da ausência de não haver um controle de acesso efetivo na instituição. No entanto, é válido ressaltar que o sistema desenvolvido pode ser utilizado em outras salas restritas do Instituto, como a Sala de TI, coordenação, diretoria, entre outras.

É possível observar que os objetivos propostos foram alcançados, com a implementação de um sistema de controle de acesso baseado na tecnologia RFID. O sistema desenvolvido é capaz de gerenciar usuários, dispositivos e monitorar acessos, garantindo maior segurança e controle nos ambientes onde é instalado. O sistema desenvolvido apresenta um potencial significativo para ser utilizado em diversos ambientes, como empresas, escolas, instituições públicas e privadas. Com a expansão e aprimoramento do sistema, é possível adicionar novas funcionalidades e melhorar ainda mais o seu desempenho.

Algumas das dificuldades encontradas foram a escolha do microcontrolador, pois há muitas opções no mercado, o desenvolvimento do sistema web com relação ao uso das bibliotecas (frameworks), e vincular id da tag e mac address do dispositivo, e por fim a ativação da tranca da fechadura elétrica.

Com base na experiência adquirida durante o desenvolvimento e implementação do sistema proposto, é viável

sugerir algumas expansões e melhorias que poderiam ser realizadas no futuro. Alguns exemplos a serem considerados incluem: a implementação de uma VLAN para reforçar a segurança na rede, de modo que separe a rede dos microcontroladores das demais redes, e entre outras recomendações e boas práticas em cibersegurança; a integração de compatibilidade do protocolo IPv6 no NodeMCU ESP8266; adição de sistemas de alertas e notificações; uma possível integração com outros sistemas utilizados no instituto, como sistemas de gestão acadêmica (SIGAA) ou bancos de dados dos funcionários; e por fim, melhorias na interface visual por meio da implementação do Bootstrap, proporcionando uma experiência mais aprimorada e intuitiva.

7. Referências

Cunha, Alessandro F. (2007). “O que são sistemas embarcados”. Saber Eletrônica, v. 43, n. 414, p. 1-6.

https://files.comunidades.net/mutcom/ARTIGO_SIST_EMB.pdf

De Almeida Greff, Ponciano (2009). “Especificação de um Sistema para Monitoramento de Atividades de Natação usando RFId”.

https://wiki.sj.ifsc.edu.br/images/0/06/ProjetoFinal_Ponciano.pdf

De Oliveira, Sérgio (2017). “Internet das coisas com ESP8266, Arduino e Raspberry PI”. Novatec Editora. p. 15-32, 64.

Dos Santos Pinheiro, José Maurício et al (2006). “Identificação por Radiofrequência: Aplicações e Vulnerabilidades da Tecnologia RFID”. Cadernos UniFOA, v. 1, n. 2, p. 18-32.

https://revistas.unifoa.edu.br/cadernos/article/view/889/733

Junior, Vanderlei Freitas et al. (2017). “Design Science Research Methodology Enquanto Estratégia Metodológica para a Pesquisa Tecnológica”. Revistas Espacios 38 (6), p. 25.

https://www.revistaespacios.com/a17v38n06/a17v38n06p25.pdf

Morais, Bianca Stephanie Guimarães (2020). Sistemas de controle de acesso utilizando tecnologia RFID-GPACESS.

https://repositorio.ufersa.edu.br/handle/prefix/7527

Nambiar, Arun N (2009). “RFID technology: A review of its applications”. In: Proceedings of the world congress on engineering and computer science. Hong Kong, China: International Association of Engineers, p. 20-22.

https://www.iaeng.org/publication/WCECS2009/WCECS2009_pp1253-1259.pdf

OLIVEIRA, Greici (2016). NodeMCU: Uma plataforma com características singulares para o seu projeto IoT. Blog MasterWalkerShop.

https://blogmasterwalkershop.com.br/embarcados/nodemcu/nodemcu-uma-plataforma-com-caracteristicas-singulares-para-o-seu-projeto-iot. Acesso em 29/11/2023

Oliveira, Alessandro de Souza; Pereira, Milene Franco (2006). “Estudo da tecnologia de identificação por radiofreqüência-RFID”.

https://bdm.unb.br/bitstream/10483/829/1/2006_AlessandroeMilene.pdf

Ossada, Jaime Cazuhiro et al. (2012). “GERSE: Guia de Elicitação de Requisitos para Sistemas Embarcados”. In: WER.

https://www.researchgate.net/publication/259758053_GERSE_Guia_para_Elicitacao_de_Requisitos_de_Sistemas_Embarcados/link/0deec52da8012cb44b000000/download

Parihar, Yogendra Singh et al (2019). “Internet of things and nodemcu”. journal of emerging technologies and innovative research, v. 6, n. 6, p. 1085.

https://www.researchgate.net/profile/Yogendra-Singh-Parihar/publication/337656615_Internet_of_Things_and_Nodemcu_A_review_of_use_of_Nodemcu_ESP8266_in_IoT_products/links/5e29767b4585150ee77b868a/Internet-of-Things-and-Nodemcu-A-review-of-use-of-Nodemcu-ESP8266-in-IoT-products.pdf

Teixeira, Tiago (2011). “Controle de Fluxo de Pessoas Usando RFId”. Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina - Campus São José, São José-SC.

https://wiki.sj.ifsc.edu.br/images/f/fa/TCC_TiagoTeixeira.pdf

Zanlourensi, Luis Guilherme (2011). “Identificação por rádio freqüência-RFID”. Disponível em < http://repositorio.utfpr.edu.br/jspui/bitstream/1/19986/2/CT_TELEINFO_XIX_2011_14.pdf>

Análise de detecção de ataques DDoS baseado em machine learning

Rodrigo Schwartzhaupt Nunes[1], Matheus Lorenzato Braga[2], Vanderlei Freitas Junior[3]

[1]Instituto Federal Catarinense – Campus Avançado Sombrio
Caixa Postal 88960-000 – Santa Catarina – SC – Brazil

[2]Instituto Federal Catarinense – Campus Avançado Sombrio
Caixa Postal 88960-000 – Santa Catarina – SC – Brazil

[3]Instituto Federal Catarinense – Campus Avançado Sombrio
Caixa Postal 88960-000 – Santa Catarina – SC – Brazil

rodrigosnuness@gmail.com, matheus.braga@ifc.edu.br,
vanderlei.freitas@ifc.edu.br

Abstract. *This study addresses the application of machine learning techniques in network traffic anomaly detection, aiming to enhance cybersecurity. The methodology encompassed a comprehensive analysis of the proposed problem, the selection of the most efficient machine learning techniques, their execution on a specific dataset, and a comprehensive analysis of the obtained results. The primary objectives were to assess the effectiveness of machine learning techniques in anomaly detection and identify potential limitations and suggestions for future enhancements in the detection system. Methodological procedures followed established guidelines from the academic literature, drawing from related works that substantiated the efficiency of the selected techniques. The dataset used was the DDoS Evaluation Dataset (CIC-DDoS2019), containing features extracted from*

network traffic. The results demonstrated the effectiveness of artificial intelligence in predicting attacks, achieving an accuracy rate exceeding 99% for both evaluated techniques. This study contributes to advancing cybersecurity by providing an efficient approach to network traffic anomaly detection. The conclusions underscore the feasibility and effectiveness of the employed techniques, emphasizing the importance of a thorough problem analysis and the need to consider specific characteristics when developing anomaly detection solutions.

Resumo. *O presente trabalho aborda a aplicação de técnicas de machine learning na detecção de anomalias no tráfego de rede, com o objetivo de melhorar a segurança cibernética. A metodologia adotada incluiu uma análise detalhada do problema proposto, a seleção das técnicas de machine learning mais eficientes, a execução dessas técnicas em uma base de dados específica e uma análise abrangente dos resultados obtidos. Os principais objetivos foram avaliar a eficácia das técnicas de machine learning na detecção de anomalias e identificar possíveis limitações e sugestões para melhorias futuras no sistema de detecção. Os procedimentos metodológicos seguiram as diretrizes estabelecidas na literatura acadêmica, com base em trabalhos relacionados que comprovaram a eficiência das técnicas selecionadas. A base de dados utilizada foi a DDoS Evaluation Dataset (CIC-DDoS2019), que contém recursos extraídos do tráfego de rede. Os resultados obtidos demonstraram a eficácia da inteligência artificial na previsão de ataques, com acurácia superior a 99% para ambas as técnicas avaliadas. Este estudo*

contribui para o avanço da segurança cibernética, fornecendo uma abordagem eficiente para a detecção de anomalias no tráfego de rede. As conclusões destacam a viabilidade e a eficácia das técnicas utilizadas, bem como a importância da análise detalhada do problema e a necessidade de considerar características específicas ao desenvolver soluções de detecção de anomalias.

1. Introdução

A área da Segurança da Informação se insere como um dos campos cruciais no âmbito da Tecnologia da Informação e Comunicação, desempenhando um papel fundamental na identificação e desenvolvimento de estratégias para salvaguardar todas as informações e dados que transitam pela rede. Seu foco primordial reside na minimização dos riscos associados e na asseguração da continuidade dos serviços de uma organização. É notório que, nos últimos anos, temos presenciado uma crescente incidência de ataques DDoS (*Distributed Denial of Service*) perpetrados por grupos de usuários maliciosos, com consequências severas, tais como a indisponibilidade de serviços, prejuízos financeiros e reputacionais, bem como o risco iminente de exposição de dados sensíveis, conforme destacado pelo CERT.br (CERT.BR, 2016).

Neste contexto desafiador, o propósito principal deste estudo é apresentar uma ferramenta de Inteligência Artificial (IA) desenvolvida por meio da aplicação de técnicas de ML (Machine Learning), incluindo a utilização de Árvore de Decisão e Floresta Aleatória. Ambas as abordagens foram implementadas com sucesso na linguagem de programação Python, respaldadas por resultados promissores previamente evidenciados em pesquisas anteriores (FIGUEIREDO, 2012; CLARINDO; SILVA, 2022; GARCIA, 2022).

Este trabalho visa apresentar uma solução de detecção de ataques DDoS com base em IA, fortalecendo a segurança cibernética por meio de técnicas avançadas de aprendizado de máquina e processamento de dados em tempo real. Sendo então, não apenas analisar as ameaças imediatas, mas também contribuir para o avanço contínuo da defesa cibernética, alinhando-se com as demandas crescentes e complexas do cenário digital contemporâneo.

O presente artigo está estruturado da seguinte maneira: na seção subsequente, abordaremos detalhadamente a metodologia empregada, destacando os procedimentos e técnicas utilizados na criação da ferramenta de inteligência artificial. Em seguida, apresentaremos os resultados alcançados e a discussão a eles relacionada. Finalmente, na última seção, concluiremos o artigo, resumindo as principais descobertas e delineando possíveis direções futuras para a pesquisa nesta área crítica da Segurança da Informação.

2. Objetivos

A partir do estágio obrigatório supervisionado e determinado o tema de pesquisa proposto para este trabalho de conclusão de curso, sob o título “Análise de ataques DDoS baseado em Machine Learning", definiram-se os seguintes objetivos.

2.1. Objetivo Geral

A pesquisa tem como objetivo desenvolver e implementar um sistema de detecção robusto e eficiente capaz de identificar com precisão os ataques DDoS em redes de computadores, utilizando técnicas avançadas de aprendizado de máquina e processamento de dados em tempo real.

A proposta busca contribuir significativamente para a segurança cibernética, analisando ameaças de ataques DDoS e

fortalecendo a resiliência das infraestruturas de rede contra essas ameaças, cada vez mais sofisticadas.

2.2. Objetivos Específicos

Diante do objetivo geral definido para o trabalho, os objetivos específicos foram definidos como segue:

- Investigar os principais desafios, limitações e questões relacionados à Inteligência Artificial e às vulnerabilidades envolvendo DDoS.
- Implementar e Treinar Modelos de Machine Learning para aprimorar a capacidade do sistema em identificar padrões associados a ataques DDoS.
- Realizar testes e simulações para avaliar a eficácia da abordagem na detecção e classificação de ataques DDoS.
- Contribuir para o conhecimento geral na área de segurança cibernética, de acordo com às necessidades específicas do trabalho.

3. Referencial Teórico

Este estudo se apoia em uma variedade de técnicas e recursos já consolidados na literatura acadêmica. Com o propósito de proporcionar ao leitor uma compreensão inicial sobre cada um desses elementos, apresentaremos nos parágrafos subsequentes uma breve síntese das abordagens selecionadas.

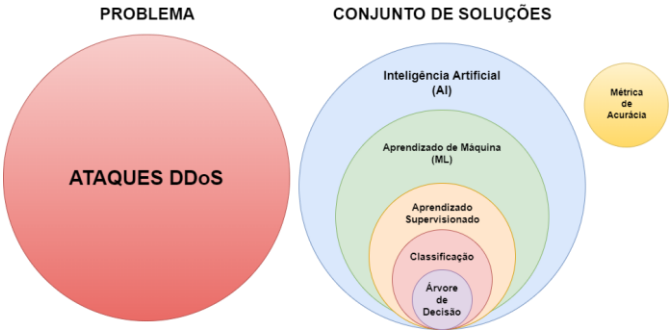

Figura 1. Essa figura demonstra de forma organizacional as técnicas e recursos utilizados.

Conforme ilustrado na Figura 1, é possível observar a divisão dos conjuntos de técnicas e recursos em duas grandes categorias: o problema e o conjunto de soluções. Dentro do conjunto de soluções, destacam-se duas subdivisões: a primeira engloba a Inteligência Artificial (IA) e todos os assuntos relacionados ao seu universo, enquanto do outro lado temos a Métrica de Acurácia, que serve para quantificar a precisão de determinadas medidas ou resultados obtidos. Esta estruturação proporciona uma visão clara da organização dos elementos envolvidos, facilitando a compreensão e análise do cenário abordado na pesquisa.

3.1. Ataques DDoS

De acordo com Nazario (2008) e Mello (2010), os ataques Distribuídos de Negação de Serviço (DDoS) representam ações maliciosas com o intuito de sobrecarregar uma infraestrutura-alvo, excedendo sua capacidade de processamento e recursos disponíveis por meio de uma alta carga de tráfego de dados. Isso culmina na inacessibilidade do sistema ou servidor para os usuários legítimos, acarretando prejuízos financeiros e danos à reputação da vítima.

Os ataques DDoS, também conhecidos como *Distributed Denial of Service*, são frequentemente perpetrados por indivíduos ou grupos com a finalidade de prejudicar organizações, empresas ou mesmo serviços online em grande escala. O objetivo primordial desses ataques consiste em saturar os recursos de rede e computacionais da vítima com tráfego malicioso, impossibilitando o acesso ou uso legítimo por parte dos usuários (Mirkovic, 2004).

Diversas técnicas são empregadas na execução de ataques DDoS, caracterizando-se por sua natureza distribuída. Isso envolve uma rede de computadores comprometidos, denominados "bots" ou "zumbis", que são controlados por um atacante central, frequentemente sem o conhecimento dos proprietários dessas máquinas. Esse cenário torna a detecção e a mitigação desses ataques um desafio significativo (Lau, 2000).

Segundo Mirkovic (2004), os alvos de ataques DDoS abrangem uma ampla gama, desde pequenos sites e fóruns online até organizações de grande porte, instituições financeiras e até mesmo infra estruturas críticas, como sistemas de energia e telecomunicações. Além disso, os motivos que instigam esses ataques podem variar, incluindo vingança, competição empresarial, motivações políticas e até mesmo extorsão, na qual os atacantes ameaçam continuar os ataques a menos que um resgate seja pago.

Para fortalecer a proteção contra ataques DDoS, as organizações necessitam adotar uma abordagem de várias camadas que compreenda a implementação de sistemas de detecção e mitigação de ataques, configurações apropriadas de firewalls e sistemas de filtragem, bem como a coordenação com provedores de serviços especializados em segurança DDoS. A capacidade de resposta rápida desempenha um papel crucial na minimização dos danos durante um ataque (Laufer, 2005).

Além disso, a conscientização em segurança cibernética e a aplicação das melhores práticas, como a distribuição geográfica de servidores e o uso de sistemas de balanceamento de carga, podem contribuir para a redução da vulnerabilidade a ataques DDoS. Importa ressaltar que a ameaça de ataques DDoS continua a evoluir, tornando essencial a colaboração entre organizações, provedores de serviços de internet e especialistas em segurança para mitigar esse tipo de ameaça em constante evolução.

3.2. Ataque de Reflexão Tipo SYN

O ataque de reflexão tipo SYN representa uma variante dos ataques DDoS que se vale de vulnerabilidades nos sistemas, a fim de sobrecarregar as infra estruturas alvo. Nessa categoria de ataques, perpetradores enviam um volume considerável de solicitações de conexão SYN (*Synchronize*) falsas para servidores, os quais, por sua vez, respondem com um SYN-ACK (*Synchronize-Acknowledge*), aguardando a confirmação final (Lau, 2000; Laufer, 2005).

No entanto, no contexto do ataque de reflexão tipo SYN, os atacantes deliberadamente não concluem a conexão, deixando os servidores em estado de espera por uma confirmação que nunca chegará. Isso resulta na exaustão dos recursos dos servidores, incluindo sua capacidade de processamento e memória, tornando-os inacessíveis para os usuários legítimos.

Conforme Carvalho (2020) destacou em sua pesquisa, o DoSSEC é uma proposta que visa à detecção e mitigação de ataques SYN Flood em redes SDN, ressaltando a importância de abordagens específicas para combater essa ameaça.

Adicionalmente, os estudos realizados por Fox et al. (2019) contribuíram significativamente para a detecção de ataques syn-flooding em redes definidas por software, fornecendo valiosos insights para o desenvolvimento de técnicas de proteção e prevenção eficazes.

O ataque de reflexão tipo SYN revela-se particularmente eficaz, uma vez que amplifica seu impacto, permitindo que os atacantes utilizem servidores de terceiros como "amplificadores". Isso faz com que o tráfego malicioso pareça originar-se de múltiplas fontes, tornando a identificação e a mitigação do ataque ainda mais desafiadoras. A compreensão aprofundada desse tipo de ataque é crucial para a implementação de medidas de segurança eficazes, visando a proteção de sistemas e redes contra essa ameaça específica.

3.3. Machine Learning

Machine Learning ou Aprendizado de Máquina, uma subárea da Inteligência Artificial, desempenha um papel fundamental na revolução da ciência de dados. Esta disciplina é dedicada ao desenvolvimento de algoritmos e modelos capazes de aprimorar seu desempenho por meio da análise de dados, em oposição à abordagem tradicional de programação explícita para tarefas específicas (Mitchell, 1997). O processo de Machine Learning é composto por três etapas interligadas e cruciais.

Na fase de treinamento, o modelo é alimentado com um conjunto de dados de treinamento que contém exemplos relevantes e informações relacionadas à tarefa desejada. Durante essa etapa, o modelo é exposto a uma ampla variedade de dados, permitindo-lhe aprender a reconhecer padrões e relações nos dados. Essa capacidade de "aprendizado" é o que distingue o Machine Learning de abordagens mais convencionais (Bishop, 2006).

Após o treinamento, o modelo entra na fase de teste e avaliação. Aqui, ele é confrontado com um conjunto de dados de teste separado, projetado para verificar sua capacidade de generalização. Essa etapa é fundamental para determinar a precisão e o desempenho do modelo em condições do mundo real, garantindo sua confiabilidade e utilidade prática (Hastie et al., 2009).

Uma vez que o modelo tenha passado com sucesso pelas etapas de treinamento e teste, ele está pronto para ser implantado em situações reais. Isso permite que o modelo faça previsões, tome decisões e forneça resultados com base em novos dados não observados anteriormente. O Machine Learning emergiu como uma ferramenta poderosa para lidar com dados complexos e dinâmicos, possibilitando análises mais profundas e informadas

3.4. Aprendizado Supervisionado - Classificação

O Aprendizado Supervisionado é uma das pedras angulares do Machine Learning e desempenha um papel vital na análise de dados. Essa técnica visa treinar modelos capazes de atribuir categorias ou rótulos a novos dados, com base em exemplos previamente rotulados (Nasteski, 2017). Dentro do âmbito do aprendizado supervisionado, destacam-se dois principais enfoques: a classificação e a regressão. Neste contexto, nossa atenção se concentra na análise da classificação, uma vez que essa área abrange uma ampla gama de técnicas consolidadas e relevantes para a pesquisa em questão.

Existem diversos algoritmos de classificação disponíveis, e neste estudo, utilizaremos duas técnicas de Machine Learning amplamente reconhecidas: a Árvore de Decisão e a Floresta Aleatória. Essas abordagens foram escolhidas com base em sua eficiência comprovada em tarefas de classificação e detecção de Ataques Distribuídos de Negação de Serviço (DDoS) (Figueiredo, 2012; Clarindo; Silva, 2022; Garcia, 2022). Esses algoritmos são treinados com base em dados rotulados, o que lhes permite aprender padrões e relações que, por sua vez, possibilitam a classificação precisa de novos dados.

O processo de treinamento envolve o uso de um conjunto de dados rotulados, que é tipicamente dividido em dois subconjuntos: o conjunto de treinamento e o conjunto de teste.

O modelo é treinado utilizando o conjunto de treinamento e, em seguida, é avaliado com o conjunto de teste para verificar seu desempenho. Métricas de avaliação, tais como precisão, recall, F1-score e matriz de confusão, são frequentemente empregadas para medir quão bem o modelo está executando a tarefa de classificação de dados.

O Aprendizado Supervisionado, em especial a classificação, desempenha um papel crucial em uma variedade de domínios, incluindo processamento de linguagem natural, visão computacional, diagnóstico médico, detecção de fraudes e muito mais. Essas aplicações contribuem significativamente para a tomada de decisões informadas, proporcionando insights valiosos com base na análise de dados rotulados (Nasteski, 2017; Kotsiantis et al., 2007).

3.5. Árvore de Decisão

Uma Árvore de Decisão é uma técnica de aprendizado de máquina que se baseia em estruturas de árvore para modelar decisões e previsões a partir de dados. É uma abordagem intuitiva e altamente interpretável, amplamente utilizada em problemas de classificação e regressão. Como Ali *et al.* (2012) afirmam, as Árvores de Decisão são eficazes na construção de modelos preditivos, devido à sua capacidade de representar decisões de maneira clara e hierárquica.

A construção de uma Árvore de Decisão envolve a divisão dos dados em subconjuntos com base em características relevantes. A cada divisão, a árvore procura maximizar a homogeneidade das classes em cada ramo, resultando em uma estrutura que pode ser facilmente interpretada (Fratello & Tagliaferri, 2018). Essa capacidade de interpretabilidade é particularmente valiosa em domínios nos quais a justificação das decisões é essencial.

A aplicação das Árvores de Decisão abrange problemas de classificação, nos quais rótulos ou categorias são atribuídos a

instâncias com base em características, bem como problemas de regressão, nos quais valores numéricos são previstos. Esse poderoso enfoque é útil em diversos contextos, como diagnósticos médicos, onde uma Árvore de Decisão pode ser usada para classificar pacientes em grupos de diagnóstico com base em sintomas clínicos.

3.6. Floresta Aleatória

A Floresta Aleatória é uma técnica avançada de aprendizado de máquina que se baseia no conceito de conjuntos (ensemble) de Árvores de Decisão. Como Ali *et al.* (2012) destacam, essa abordagem tem ganhado destaque devido à sua capacidade de melhorar a precisão e a robustez das previsões em problemas de classificação e regressão. A Floresta Aleatória é uma extensão inteligente das Árvores de Decisão, que visa mitigar algumas das limitações associadas a modelos de decisão únicos.

Essa técnica opera criando múltiplas Árvores de Decisão independentes e combinando suas previsões para chegar a um resultado final. Fratello e Tagliaferri (2018) explicam que a combinação das previsões de várias árvores resulta em uma estimativa mais robusta e menos sujeita a overfitting, tornando-a adequada para lidar com conjuntos de dados complexos.

A Floresta Aleatória encontra aplicações em uma variedade de domínios, desde diagnósticos médicos até previsão de preços em finanças. Sua capacidade de lidar com múltiplas classes e características, bem como sua resistência a ruído nos dados, a tornam uma escolha atraente para problemas do mundo real.

Em resumo, a Floresta Aleatória representa uma evolução das Árvores de Decisão, visando melhorar a precisão e a generalização das previsões. Ela se destaca como uma técnica robusta e versátil que desempenha um papel significativo no campo do aprendizado de máquina.

3.7. Métrica de Acurácia Utilizadas

A avaliação do desempenho de técnicas de inteligência artificial, como algoritmos de aprendizado de máquina, é fundamental para determinar a eficácia e a confiabilidade desses sistemas em resolver tarefas específicas. Uma das métricas mais comuns e importantes usadas para avaliar o desempenho é a acurácia. Conforme Junior *et al.* (2022) explicam, a acurácia é uma medida que indica a proporção de previsões corretas feitas por um modelo em relação ao número total de previsões realizadas.

A acurácia é calculada com base em quatro elementos-chave:

Verdadeiro Positivo (VP): se refere aos casos em que o modelo previu corretamente uma classe positiva, ou seja, quando o resultado real é positivo e o modelo previu positivo.

Verdadeiro Negativo (VN): representa os casos em que o modelo previu corretamente uma classe negativa, ou seja, quando o resultado real é negativo e o modelo previu negativo.

Falso Positivo (FP): são os casos em que o modelo previu incorretamente uma classe positiva, ou seja, quando o resultado real é negativo, mas o modelo previu positivo.

Falso Negativo (FN): se refere aos casos em que o modelo previu incorretamente uma classe negativa, ou seja, quando o resultado real é positivo, mas o modelo previu negativo.

A acurácia mede, portanto, a capacidade geral do modelo de classificar corretamente tanto os exemplos positivos quanto os negativos. Ela é uma métrica fundamental e amplamente utilizada para avaliar a qualidade das previsões feitas por modelos de inteligência artificial.

4. Metodologia

A metodologia adotada para este trabalho consiste nas seguintes etapas: (a) Análise do Problema Proposto, (b) Seleção das Técnicas de Machine Learning, (c) Execução das Técnicas em uma Base de Dados para Testes, (d) Análise dos Resultados Obtidos, (e) Captura de Tráfego em Cenário Simulado e (f) Teste da IA com Tráfego Capturado. Na etapa (a) realizou-se uma análise detalhada do problema proposto, compreendendo suas características essenciais, desafios e requisitos. Isso incluiu a identificação das principais variáveis envolvidas, a compreensão das relações entre elas e uma avaliação das necessidades específicas que a solução deve atender. Essa análise inicial serviu como base para orientar as decisões nas etapas subsequentes do projeto. A etapa (b), consiste na seleção através do levantamento de trabalhos relacionados encontrados, em que a eficiência na detecção de anomalias em tráfego de rede sejam comprovadas. Nesta etapa, as técnicas de Árvore de Decisão e Floresta Aleatória foram as selecionadas (Figueiredo, 2012; Clarindo e Silva, 2022; Garcia, 2022). A etapa (c) foi realizada utilizando a base de dados DDoS Evaluation Dataset (CIC-DDoS2019) (UNB, 2023), a qual contém recursos extraídos do tráfego capturado pelo CICFlowMeter-V3 (UBN, 2023), para o processo de treinamento da inteligência artificial realizado. Na etapa (d), os resultados obtidos são analisados de forma abrangente, sendo feita uma avaliação da eficácia das técnicas de ML implementadas na detecção de anomalias no tráfego de rede, considerando métricas como porcentagem de precisão. A análise também inclui uma interpretação dos resultados à luz das necessidades e características específicas do problema proposto, para posterior identificação de possíveis limitações e sugestões para melhorias futuras no sistema de detecção de anomalias. Na etapa (e), foi elaborado um cenário simulando ataques DDoS do tipo reflexão Syn, e utilizado ferramentas para captura e armazenamento desse tráfego. A etapa (f) foi realizada a partir

da IA treinada na etapa (c). Assim, para analisar o tráfego capturado na etapa (e) e validar a capacidade e eficácia da IA em identificar ataques cuja origem não fazia parte de seu treinamento.

5. Desenvolvimento

5.1. Análise do Problema Proposto

No âmbito desta etapa, empreendeu-se uma análise minuciosa do problema apresentado. Esse processo abrangeu a identificação de variáveis-chave, a compreensão das relações entre essas variáveis e a determinação de necessidades específicas relacionadas ao contexto em questão. A condução dessa análise serviu como fundamento para nortear as decisões tomadas nas etapas subsequentes do projeto, assegurando uma abordagem embasada e coerente no desenvolvimento do presente trabalho.

5.2. Seleção das Técnicas de Machine Learning

Na fase de escolha das técnicas de aprendizado de máquina, optou-se pela utilização da Árvore de Decisão e da Floresta Aleatória, conforme respaldado por autores como Figueiredo (2012), Clarindo e Silva (2022) e Garcia (2022). Essa decisão fundamentou-se na evidência de que tais abordagens apresentam resultados particularmente significativos na classificação e previsão das características de tráfego de redes relacionadas a ataques DDoS.

5.3. Execução das Técnicas em Base de Dados para Teste

Durante a execução das técnicas na base de dados destinada aos testes, constatou-se a presença de ataques DDoS reflexivos, tais como PortMap, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS e SNMP. Optou-se por utilizar a tabela "Syn.csv" do conjunto de dados disponibilizados. Nesta tabela,

foram identificadas mais de 80 características, as quais foram extraídas utilizando o CICFlowMeter-V3. Posteriormente, realizou-se uma filtragem para selecionar as características relevantes para a técnica de classificação, resultando em 79 características que foram disponibilizadas para o treinamento e teste da Inteligência Artificial em uma proporção de 80% destinados ao treinamento e 20% aos testes.

5.4. Análise dos Resultados Obtidos

A análise dos resultados obtidos empregou a métrica de acurácia, apresentada tanto de forma percentual quanto por meio de representações gráficas.

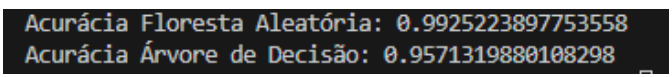

Figura 2. Essa figura demonstra o percentual de acurácia de cada técnica de machine learning.

Conforme ilustrado na Figura 2, foi possível mensurar a taxa percentual de acurácia de cada técnica de aprendizado de máquina, evidenciando que ambas alcançam uma taxa de acerto superior a 95% na fase de testes. Essa representação foi

elaborada graficamente para enfatizar os resultados.

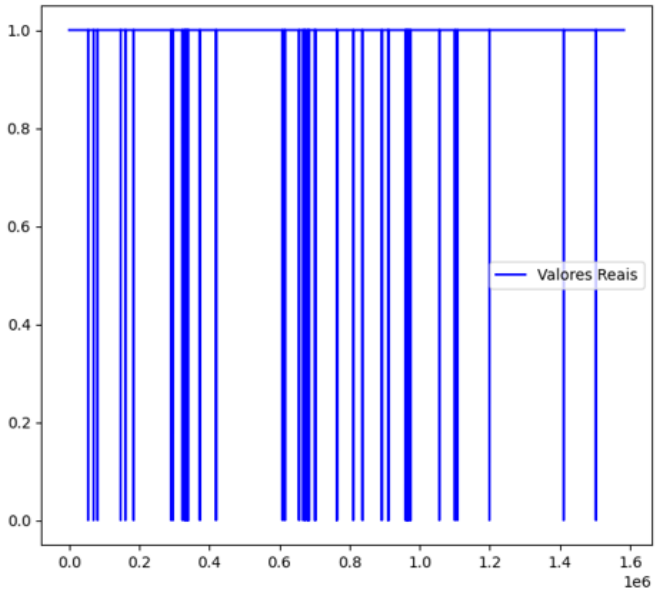

Figura 3. Essa figura demonstra os valores Reais da coluna a ser prevista pela AI.

Na Figura 3, é possível observar que os valores da coluna a ser prevista pela IA situam-se numa escala de 0 a 1. O valor 1 indica a confirmação de um ataque DDoS, enquanto o valor 0 sugere a ausência de um ataque DDoS.

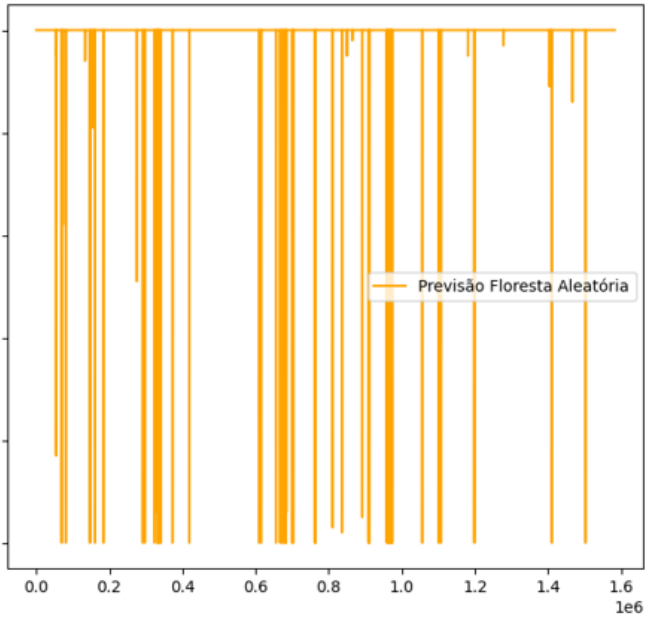

Figura 4. Essa figura demonstra os valores de teste da AI após o treinamento

Conforme evidenciado na Figura 4, conseguimos identificar de maneira gráfica as previsões realizadas pela Máquina de Aprendizado Floresta Aleatória em relação à última coluna da tabela.

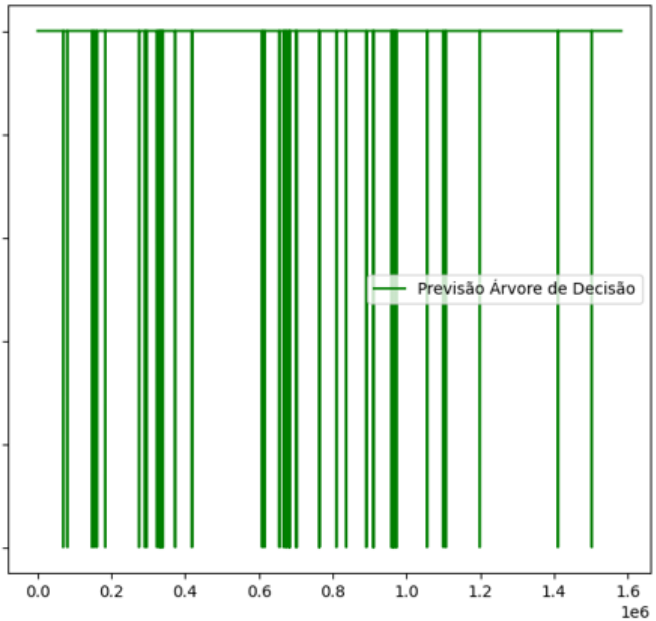

Figura 5. Essa figura demonstra os valores de teste da AI Árvore de Decisão após o treinamento

Conforme ilustrado na Figura 5, foi possível identificar de maneira gráfica as previsões realizadas pela Máquina de Aprendizado Árvore de Decisão em relação à última coluna da tabela.

5.5. Captura de Tráfego em Cenário Simulado

No intuito de avaliação e validação, foi concebido um cenário de simulação destinado a testes, no qual ataques DDoS de reflexão, especificamente do tipo SYN, foram deliberadamente conduzidos em direção a um endereço IP simulado, representando a entidade alvo. Para registrar e analisar o tráfego resultante desses ataques, empregou-se a ferramenta CICFlowMeter. Esta aplicação possibilitou a captação precisa

das características do tráfego, permitindo uma análise detalhada dos padrões associados aos ataques SYN. Os dados provenientes da presente coleta foram apropriadamente armazenados em um arquivo no formato .csv. No interior deste arquivo, constam 107 registros de tráfego, dos quais somente 4 registros são identificados como tráfego benigno. Os restantes 103 registros, por sua vez, caracterizam-se como registros de ataques DDoS do tipo Syn. Este conjunto de dados, assim consolidado, revela-se representativo e propício à exploração durante as fases subsequentes do processo de análise e classificação de ameaças DDoS.

5.6. Testes das AI com Tráfego Capturado

Empregando a Inteligência Artificial previamente treinada com o propósito de identificar ataques DDoS de reflexão, especialmente do tipo SYN, procedeu-se à avaliação de sua acurácia por meio de um teste específico. Esse teste foi conduzido utilizando o tráfego previamente capturado na fase anterior do experimento, onde ataques simulados foram direcionados ao sistema. Ao realizar a comparação entre as predições geradas pela AI e os eventos reais contidos no tráfego capturado, pôde-se mensurar a eficácia do modelo na identificação precisa de ataques DDoS de reflexão SYN. Esta avaliação robusta do desempenho do sistema em condições simuladas revelou que a IA identificou corretamente os 4 registros de tráfego benigno, enquanto os demais 103 registros foram categorizados com precisão como ataques DDoS SYN.

```
Ação tomada: Possível anomalia detectada
Ação tomada: Possível anomalia detectada
Tudo parece normal
Tudo parece normal
Ação tomada: Possível anomalia detectada
Ação tomada: Possível anomalia detectada
Ação tomada: Possível anomalia detectada
Ação tomada: Possível anomalia detectada
Ação tomada: Possível anomalia detectada
Ação tomada: Possível anomalia detectada
Ação tomada: Possível anomalia detectada
Ação tomada: Possível anomalia detectada
Ação tomada: Possível anomalia detectada
Ação tomada: Possível anomalia detectada
Ação tomada: Possível anomalia detectada
```

Figura 6. A figura em questão ilustra o alerta gerado pelo sistema durante o processo de classificação do registro analisado.

Conforme representado na Figura 6, é possível identificar que a IA, após analisar o registro de tráfego, classifica-o como um ataque DDoS, designando-o como anomalia. Em contrapartida, quando se trata de um registro benigno, a IA emite apenas um alerta de normalidade ao sistema.

Essa análise crítica da acurácia constitui um passo crucial no aprimoramento contínuo da capacidade preditiva da IA diante de ameaças DDoS específicas.

6. Considerações Finais

Os resultados alcançados até o presente momento confirmam a eficácia das técnicas de machine learning, em particular a árvore de decisão e a floresta aleatória, no que tange à detecção de ataques DDoS do tipo SYN de reflexão. A utilização do conjunto de ferramentas, tanto durante a fase de treinamento quanto na de

testes, reforça a viabilidade do modelo adotado em ambientes mais complexos e com um maior número de variáveis.

Adicionalmente, este estudo aponta para a possibilidade de expansão do modelo, visando à implementação de medidas de mitigação e alerta para a detecção eficaz de ataques DDoS. Estas descobertas sublinham a relevância da aplicação de técnicas de machine learning no contexto da segurança cibernética, oferecendo uma promissora abordagem para a salvaguarda contra ameaças online.

Entretanto, é importante ressaltar que a pesquisa está sujeita a certas limitações, e oportunidades para futuras investigações que aprofundem e ampliem esses resultados são claramente identificadas. Os achados apresentados neste estudo representam um passo inicial e promissor no âmbito da defesa cibernética, incentivando investigações adicionais para aprimorar a detecção e a resposta a ataques DDoS.

Referências

ALI, Jehad et al. Random forests and decision trees. **International Journal of Computer Science Issues (IJCSI)**, v. 9, n. 5, p. 272, 2012.

BI, Qifang et al. What is machine learning? A primer for the epidemiologist. **American journal of epidemiology**, v. 188, n. 12, p. 2222-2239, 2019.

CARVALHO, Ranyelson Neres. DoSSEC: proposta de detecção e mitigação de ataques SYN Flood em redes SDN. 2020.

CERT.BR. Recomendações para Melhorar o Cenário de Ataques Distribuídos de Negação de Serviço (DDoS). 2016. Disponível em: https://www.cert.br/docs/whitepapers/ddos/. Acesso em: 21 jun. 2023.

CLARINDO, Anderson Beloni; SILVA, Keterly Geovana Gouveia. RELATÓRIO FINAL: detectando ataques ddos com inteligência artificial. Bauru: Centro Universitário Sagrado Coração - Unisagrado, 2022. 23 p. Disponível em: https://repositorio.unisagrado.edu.br/handle/handle/1387. Acesso em: 12 jul. 2023.

FIGUEIREDO, Bruno Ianoni; FERREIRA, Frederico Reid Sulahian; SIMANTOB, Marco Gabriel de Melo. Estudo e Investigação de Técnicas de IA para Detecção de Ataques DDOS. 2022.

FOX, Eduardo Farias Brinds-Ley et al. Detecção de ataques syn-flooding em redes definidas por software. 2019.

FRATELLO, Michele; TAGLIAFERRI, Roberto. Decision trees and random forests. **Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics**, v. 374, 2018.

GARCIA, Caio Cruz Alfonso; DE CARVALHO GUTIERREZ, Carolina; DA SILVA, Nathan Brito. Inteligência artificial aplicada a reconhecimento de detecção de ataque cibernético. 2022.

JUNIOR, Guanis B. Vilela et al. Métricas utilizadas para avaliar a eficiência de classificadores em algoritmos inteligentes. **Revista CPAQV–Centro de Pesquisas Avançadas em Qualidade de Vida| Vol**, v. 14, n. 2, p. 2, 2022.

KOTSIANTIS, Sotiris B. et al. Supervised machine learning: A review of classification techniques. **Emerging artificial intelligence applications in computer engineering**, v. 160, n. 1, p. 3-24, 2007.

LAU, Felix et al. Distributed denial of service attacks. In: **Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics**

evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0. IEEE, 2000. p. 2275-2280.

LAUFER, Rafael P. et al. Negação de serviço: Ataques e contramedidas. **Sociedade Brasileira de Computação**, 2005.

MELLO, FÁBIO PINHÃO; JUNIOR, RICARDO DA CRUZ MENDES; ROCHA, DANIEL GUIMARÃES. ANÁLISE DE ATAQUES DDOS.

MIRKOVIC, Jelena et al. **Internet denial of service: attack and defense mechanisms (Radia Perlman Computer Networking and Security)**. Prentice Hall PTR, 2004.

NASTESKI, Vladimir. An overview of the supervised machine learning methods. **Horizons. b**, v. 4, p. 51-62, 2017.

NAZARIO, Jose. DDoS attack evolution. **Network Security**, v. 2008, n. 7, p. 7-10, 2008.

UNB, University of New Brunswick. DDoS Evaluation Dataset (CIC-DDoS2019). 2019. Disponível em: https://www.unb.ca/cic/datasets/ddos-2019.html. Acesso em: 16 jul. 2023.