UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ISABELA DE ALMEIDA GANTZEL

MONITORAMENTO DA CIBERSEGURANÇA EM GATEWAYS LORA

CAMPO MOURÃO

2023

ISABELA DE ALMEIDA GANTZEL

MONITORAMENTO DA CIBERSEGURANÇA EM GATEWAYS LORA

Cybersecurity Monitoring in LoRa Gateways

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação do Curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Luiz Arthur Feitosa dos Santos

CAMPO MOURÃO

2023

ISABELA DE ALMEIDA GANTZEL

MONITORAMENTO DA CIBERSEGURANÇA EM GATEWAYS LORA

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação do Curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 08/novembro/2023

Luiz Arthur Feitosa dos Santos
Doutor
Universidade Tecnológica Federal do Paraná

Paulo Henrique Sabo
Doutor
Universidade Tecnológica Federal do Paraná

Rodrigo Campiolo
Doutor
Universidade Tecnológica Federal do Paraná

CAMPO MOURÃO

2023

RESUMO

A quantidade de dispositivos IoT vem crescendo consideravelmente e tem sido comumente aplicado no cotidiano de vários setores da sociedade atual, como na área de saúde, agrícola, industrial, ambientes domésticos, entre outros. Porém, os dispositivos IoT possuem baixa capacidade computacional e, portanto, possuem processamento e memória limitada se comparado com computadores convencionais. Consequentemente, há restrições de hardware que acabam por limitar também a segurança desses dispositivos. Por esses motivos, dispositivos IoT possuem vulnerabilidades de cibersegurança, o que os torna alvos potenciais de ataques maliciosos. Este trabalho tem como objetivo analisar a segurança de ambientes que possuem dispositivos IoT, para evitar ou mitigar ataques maliciosos. Para atingir esse objetivo, foi implementado um cenário de rede que é composto de um conjunto de mecanismos de segurança: monitoramento de tráfego de rede por meio de NIDS e monitoramento de *host* com HIDS. Além disso, uma pesquisa foi conduzida para investigar os ataques maliciosos que ocorreram em redes IoT, com a finalidade de identificar a natureza, vulnerabilidades e padrões subjacentes que levaram a tais ataques. Dentre os mecanismos de segurança abordados neste trabalho, tem-se como foco o monitoramento de tráfego de rede utilizando o Snort e o monitoramento de *hosts* utilizando o OSSEC. O cenário de rede foi implementado em uma rede IoT, mais especificamente em um *gateway* LoRa buscando obter respostas a respeito da eficácia do IDS. Os resultados obtidos deste trabalho através dos *logs* demostram que as ferramentas de monitoramento de rede (Snort) e *host* (OSSEC) são capazes de auxiliar na detecção de anomalias ou de ações maliciosas que ocorrem nesses ambientes. As informações obtidas a partir do monitoramento permitem compreender melhor a segurança e integridade do sistema analisado. Em conclusão, a implementação de um IDS, utilizando o Snort e o OSSEC, se mostrou eficiente no monitoramento e detecção de potenciais ameaças e violações de segurança no *gateway* LoRa. Com isso, foi obtido uma melhor compreensão a respeito de cibersegurança em redes e dispositivos IoT, como também torná-os mais seguros.

Palavras-chave: iot; ids; snort; ossec; lorawan.

ABSTRACT

The number of IoT devices has been growing considerably and has been commonly applied in the daily life of various sectors of today's society, such as health, agriculture, industrial, domestic environments, among others. However, IoT devices have low computational capacity and therefore have limited processing and memory compared to conventional computers. Consequently, there are hardware limitations that also limit the security of these devices. For these reasons, IoT devices have cybersecurity vulnerabilities, which makes them potential targets for malicious attacks. This work aims to analyze the security of environments that have IoT devices, to avoid or mitigate malicious attacks. To achieve this goal, a network scenario was implemented that is composed of a set of security mechanisms: network traffic monitoring through NIDS, host monitoring with HIDS, and some recommendations to obtain more secure environments. In addition, a survey was conducted to investigate the malicious attacks that occur in IoT networks. The aim was to study the nature of these attacks, identify their vulnerabilities, and analyze the underlying patterns that lead to such problems. Among the security mechanisms addressed in this work, the focus is on network traffic monitoring using Snort because it is an open source technology, as well as the monitoring of hosts using OSSEC, and this network scenario was implemented in IoT networks, more specifically in a LoRa gateway, seeking to obtain answers regarding the effectiveness of the IDS. The results obtained from this work through the logs show that network monitoring tools (Snort) and host monitoring tools (OSSEC) are able to assist in the detection of anomalies or malicious actions that occur in these environments. The information obtained from the monitoring allows a better understanding of the security and integrity of the analyzed system. In conclusion, the implementation of an IDS, using Snort and OSSEC, proved to be efficient in monitoring and detecting potential threats and security breaches in the gateway LoRa. With this, a better understanding of cybersecurity in IoT networks and devices is obtained, as well as making them more secure.

Keywords: iot; ids; snort; ossec; lorawan.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

Siglas

DDoS	*Distributed Denial of Service*
DNS	*Domain Name System*
DoS	*Denial of Service*
HIDS	*Host-based Intrusion Detection System*
HTTP	*Hypertext Transfer Protocol*
IDS	*Intrusion Detection System*
IoT	*Internet of Things*
IPV4	Internet *Protocol Version 4*
IPv6	Internet *Protocol Version 6*
LAN	*Local Area Network*
LoRaWan	*Long Range Wide Area Network*
MAC	*Media Access Control*
MAN	*Metropolitan Area Network*
MIC	*Message Integrity Code*
MQTT	*Message Queuing Telemetry Transport*
NIDS	*Network Intrusion Detection System*
SCA	*Side-Channel Attacks*
SIEM	*Security Information and Event Management*
SMTP	*Simple Mail Transfer Protocolo*
SPAN	*Switch Port Analyzer*
TAP	*Test Access Point*
TCP	*Transmission Control Protocol*

UDP	*User Datagram Protocol*
UTFPR	Universidade Tecnológica Federal do Paraná
UTFPR-CM	Universidade Tecnológica Federal do Paraná de Campo Mourão
WAN	*Wide Area Network*
XSS	*Cross-Site Scripting*

SUMÁRIO

1 INTRODUÇÃO

A *Internet of Things* (IoT) é definida pela NIST (2020) como a rede de dispositivos que contém o hardware, software, *firmware*, atuadores e sensores que permitem que os dispositivos se conectem, interajam e troquem dados e informações livremente.

A presença da IoT na vida das pessoas tem se tornado cada vez mais comum, podendo estar tanto em ambientes domésticos, quanto em fábricas inteligentes. É possível encontrar esses dispositivos IoT, por exemplo, em eletrodomésticos, lâmpadas e em dispositivos como a Alexa da Amazon (LIT; KIM; SY, 2021). Segundo Evans (2011), estima-se que em 2015 havia cerca de 25 bilhões de dispositivos IoT e, em 2023, segundo levantamento realizado pela Sullivan (2023), essa estimativa aumentou para 47 bilhões de dispositivos. Portanto, de acordo com essas previsões, já existem mais dispositivos IoT do que pessoas.

No entanto, os dispositivos IoT são limitados pela configuração do hardware, o que afeta a complexidade do software. Os dispositivos IoT possuem armazenamento, memória e processamento limitados, o que resulta na falta de recursos como *firewall* e outros mecanismos de segurança, facilitando intrusões e ataques maliciosos, como pode-se observar no trabalho de Nobre *et al.* (2019).

Um exemplo marcante desses ataques ocorreu em 2016, de acordo com Scott e Spaniel (2016), a empresa Dyn[1] sofreu um ataque massivo de *Distributed Denial of Service* (DDoS) com um valor estimado de 100.000 *bots* (GEER, 2005). Isso evidencia que, à medida que o número de dispositivos IoT aumenta, também ocorre um aumento de ataques maliciosos nesses dispositivos. É importante ressaltar que a segurança dos dispositivos IoT continua sendo uma preocupação atual e que continuam acontecendo ataques maliciosos, que pode ser visto no trabalho de Perakovic, Periša e Cvitić (2015).

Como mencionado anteriormente, a empresa Dyn sofreu um ataque massivo de DDoS, porém, a empresa teve uma resposta automática, evitando grandes prejuízos. A empresa empregou técnicas de segurança cibernética, como modelagem de tráfego de entrada, manipulação de políticas *anycast* (MOURA *et al.*, 2016) para equilibrar o tráfego e a utilização de serviços internos de filtragem e depuração. No mesmo ano, ocorreu outro ataque em relação aos dispositivos IoT de DDoS ao *blog* http://www.KrebsonSecurity.com, que a empresa Akamai tentou mitigar. No entanto, devido ao volume do tráfego, a Akamai desconectou o sítio da Internet para evitar a perda de desempenho para seus outros clientes pagantes. Esse segundo ataque demonstra como a falta de uma resposta automática à um ataque acarreta em consequências mais graves.

Por esses motivos, a cibersegurança é importante para manter os dispositivos IoT mais seguros. Entretanto, a segurança cibernética continua desafiadora devido à constante evolução de novas vulnerabilidades e suas formas de exploração. O NIST (2021a) define cibersegurança como prevenção de danos, proteção e restauração de computadores, sistemas de comunica-

[1] https://www.dynstatus.com/incidents/nlr4yrr162t8

ções eletrônicas, serviços de comunicações eletrônicas, comunicações por fio e comunicações eletrônicas, incluindo as informações neles contidas, para garantir sua disponibilidade, integridade, autenticação, confidencialidade e não repúdio.

Dessa forma, é analisado neste trabalho o monitoramento de intrusão para ajudar a manter redes e dispositivos IoT livres de ameaças maliciosas ou mitiga-las quando necessário. Esse monitoramento inclui o monitoramento do tráfego da rede e *host* e, além disso, foi realizado uma pesquisa para identificar os métodos de invasão mais comuns em dispositivos IoT.

Portanto, este trabalho visa descobrir os possíveis mecanismos de segurança para obter ambientes IoT mais seguros e verificar a eficácia desse cenário em redes IoT, com foco no monitoramento do tráfego de rede e *host*, como também os arquivos de *logs* que esses monitoramentos produzem.

Como IoT pode ser uma área muito vasta, este trabalho tem como escopo, analisar a segurança do *gateway* LoRa, em uma rede de testes da Universidade Tecnológica Federal do Paraná de Campo Mourão (UTFPR-CM). Mais especificamente, foram utilizados *Network Intrusion Detection System* (NIDS) e *Host-based Intrusion Detection System* (HIDS), convencionais, para monitorar o tráfego de rede do *gateway* e indiretamente dos dispositivos LoRa, para analisar se as configurações e regras de *Intrusion Detection System* (IDS) convencionais conseguem identificar ataques. Além disso, foram propostas melhorias nas configurações e regras de IDS para tentar aprimorar o monitoramento da segurança de redes IoT.

O cenário proposto para o desenvolvimento dessa pesquisa é constituído pelo Snort e OSSEC, um NIDS e HIDS para monitorar o gateway *Long Range Wide Area Network* (LoRaWan). Este trabalho busca ainda responder as questões:

1. Qual o estado da arte em relação ao monitoramento de redes e dispositivos IoT?
2. Qual é a efetividade do IDS em redes IoT?
3. É possível propor melhorias no monitoramento de redes e dispositivos IoT?

Dessa forma, espera-se contribuir para o avanço da segurança cibernética em ambientes IoT, investigando e apresentando soluções que possam aprimorar o monitoramento e a proteção contra possíveis ameaças.

Este trabalho é apresentado em capítulos e está organizado da seguinte forma: O Capítulo 2 aborda o referencial teórico, que explica conceitos importantes para a compreensão do texto, fornecendo o embasamento teórico necessário. O Capítulo 3 apresenta os trabalhos relacionados, destacando as pesquisas e estudos já realizados desta monografia. O Capítulo 4 apresenta o método de pesquisa utilizada para a realização deste trabalho, detalhando os procedimentos adotados. O Capítulo 5 discute e apresenta os resultados obtidos com o cenário de rede estabelecido. O Capítulo 6 finaliza o trabalho com as considerações finais, resumindo as principais descobertas e conclusões.

2 REFERENCIAL TEÓRICO

Este capítulo aborda os conceitos e tecnologias para a compreensão deste trabalho. A Seção 2.1 trata sobre a IoT e as tecnologias necessárias ou relacionadas para seu uso ou funcionamento, como o TCP/IP, LoRa e Bluetooth .A Seção 2.2 apresenta a segurança cibernética, abordando o IDS, mais especificamente o NIDS e o HIDS, como também dispositivos de rede ou concentradores de rede. E, por fim, a seção 2.3 aborda os ataques cibernéticos que acontecem na IoT, como também no LoRa.

2.1 Internet of Things

O termo Internet *of Things*(IoT) foi proposto pela primeira vez em 1999 por Kevin Ashton e tem se tornado cada vez mais comum na vida das pessoas ao longo dos anos. O aumento do uso de dispositivos IoT tem ocorrido em ritmo acelerado em comparação com o crescimento da população mundial.

Os dispositivos IoT, por serem dispositivos pequenos, como sensores, possuem recursos de hardware limitados e, consequentemente, capacidades computacionais reduzidas. Portanto, tendem a concentrar em atender necessidades ou funções especificas, ao invés de garantir segurança, como mencionado em Nobre *et al.* (2019).

Os dispositivos IoT podem estar presente em redes de curto alcance, como *Local Area Network* (LAN), ou em redes de longa distância, como *Wide Area Network* (WAN) ou *Metropolitan Area Network* (MAN). Exemplos dessas redes incluem Bluetooth, TCP/IP e LoRa como mostrado na Figura 1. As próximas seções abordam sobre esses tipos de redes com mais detalhes, fornecendo uma base sólida para a compreensão dos conceitos e tecnologias relacionados à IoT, essenciais para o desenvolvimento e análise deste trabalho.

2.1.1 Rede TCP/IP

Como retratado pela NIST (2009), as comunicações TCP/IP são compostas por quatro camadas, como mostrado na Figura 2. Essas camadas trabalham em conjunto para transferir dados entre *hosts*. São elas: Acesso à rede (*i*), Internet (*ii*), Transporte (*iii*) e Aplicação (*iv*).

A camada mais baixa da comunicação TCP/IP é a de Acesso à rede (*i*), que é responsável pelo tratamento das comunicações nos componentes físicos da rede. Seu protocolo mais conhecido é o Ethernet (TANENBAUM, 2003). Em seguida, os dados são enviados para a camada de Internet (*ii*), que roteia os pacotes através das redes. Seus protocolos mais conhecidos são o Internet *Protocol Version 4* (IPV4) e o Internet *Protocol Version 6* (IPv6). A camada de Transporte (*iii*) é a responsável pela transferência de dados entre redes e pode garantir confiabilidade nas comunicações. Seus protocolos são o *Transmission Control Protocol* (TCP) e o

Figura 1 – Tipos de rede que encontram-se dispositivos IoT

Fonte: (LINKS, 2019).

User Datagram Protocol (UDP). Por fim, tem-se a camada de Aplicação (*iv*), que envia e recebe dados para aplicações específicas, como *Domain Name System* (DNS), *Hypertext Transfer Protocol* (HTTP) e *Simple Mail Transfer Protocolo* (SMTP), dentre outros (TANENBAUM, 2003).

Figura 2 – Camadas do modelo TCP/IP

Fonte: (TANENBAUM, 2003).

Visto que este trabalho visa manter redes IoT mais seguras, é por meio dos protocolos TCP/IP que torna-se possível conectar dispositivos e redes IoT entre si. Alguns dos protocolos utilizados na Internet são o IEEE 802.11 (Wi-Fi) e o IEEE 802.3 (Ethernet), que são explicados a seguir.

Segundo Torres (2015), a arquitetura IEEE 802.3 (Ethernet) é mais utilizada em redes locais cabeadas e opera na camada de Acesso à rede do modelo TCP/IP, definindo assim a parte física da rede local. O papel do Ethernet é, portanto, receber os dados entregues pelos protocolos de alto nível e inseri-los em quadros que serão enviados através da rede (TANENBAUM, 2003).

Já o protocolo IEEE 802.11 (Wi-Fi) é utilizado para a criação de redes locais sem fio, usando transmissão por ondas de rádio (radiofrequência) (TORRES, 2015). A taxa de transferência e o alcance dependem do padrão usado na camada física da rede (IEEE 802.11b, IEEE 802.11g, etc), do ambiente e do tipo de antena utilizado. Este padrão opera na camada de Acesso à rede do modelo TCP/IP e é responsável por receber os pacotes de dados transmitidos pelo protocolo de alto nível usado, dividindo-os em quadros e transmitindo-os por ondas de rádio.

Alguns exemplos de aplicação em LAN podem ser vistos na Figura 1. Essas aplicações incluem redes internas, como e-mail, telefones, segurança, gerenciamento de energia e monitoramento residencial. Assim, redes TCP/IP são base para o acesso a internet devido aos seus protocolos.

2.1.2 Rede LoRa

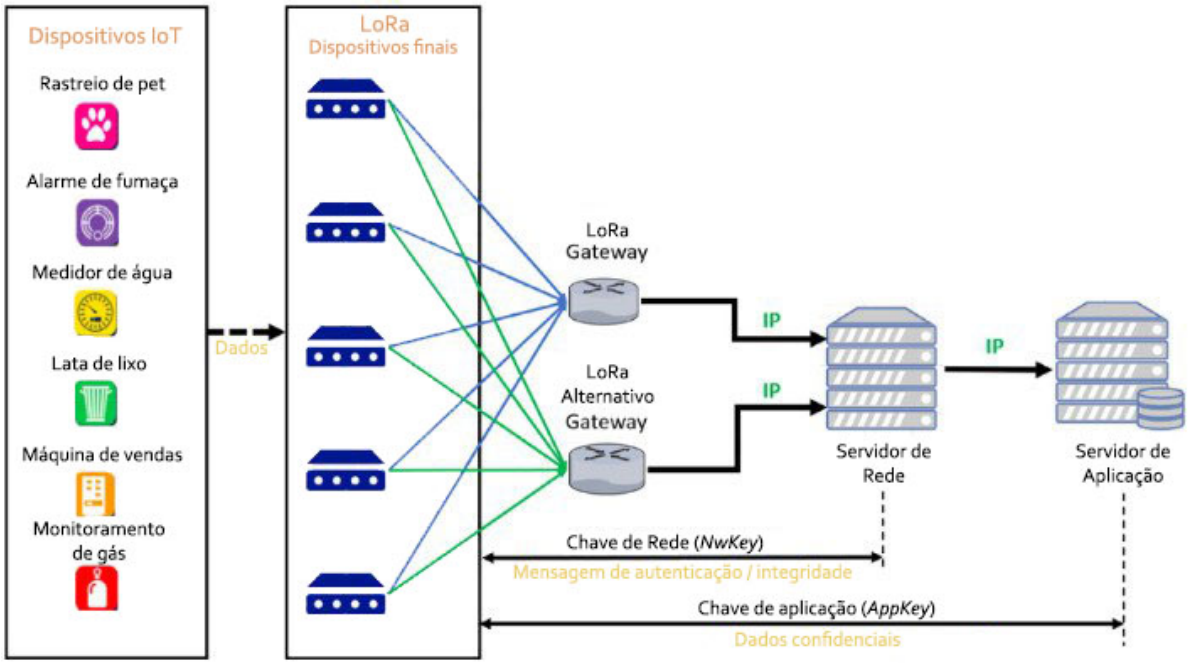
O NIST (2018) define que redes LoRa (*Long Range*) pertencem à camada de Acesso à rede do modelo TCP/IP. LoRaWan é um protocolo para redes sem fio e aplicações IoT. A especificação do protocolo é construída com base na tecnologia LoRa, desenvolvida pela LoRa Alliance, que utiliza espectro de rádio não licenciado nas bandas Industrial, Científica e Médica (ISM) para permitir baixo consumo de energia. Além de cobrir áreas amplas, oferece também comunicação bidirecional segura entre sensores remotos e *gateways* conectados à rede.

Segundo Aras *et al.* (2017) a tecnologia LoRa é descrita como uma solução de comunicação sem fio projetada para conectar dispositivos de baixa potência em longas distâncias, com baixo consumo de energia e custo reduzido. Ela utiliza a modulação de espectro espalhado (SEMTECH, 2015) para permitir a transmissão de dados em áreas urbanas, suburbanas e rurais, proporcionando ampla cobertura. Baseada em protocolo de comunicação de longo alcance e baixa taxa de transferência de dados, o que a torna adequada para aplicações que exigem comunicação de dados simples, como monitoramento remoto, rastreamento de ativos, medição de energia e agricultura inteligente.

Uma das principais vantagens da tecnologia LoRa é sua eficiência energética, permitindo que dispositivos conectados operem por longos períodos de tempo usando baterias de pequena capacidade (ARAS *et al.*, 2017). Além disso, a LoRa suporta uma grande quantidade de dispositivos conectados simultaneamente, o que é fundamental para redes de Internet das Coisas (IoT) em grande escala.

A arquitetura da rede LoRa é descrita pelo autor Aras *et al.* (2017) como uma composição de três elementos principais: dispositivos finais, *gateways* e uma rede de *back-end*. Os dispositivos finais são os nós sensores ou atuadores que coletam ou enviam dados para a rede. Os *gateways* são responsáveis por receber e encaminhar os dados entre os dispositivos finais e a rede de *back-end*. A rede de *back-end* gerencia e processa os dados recebidos pelos *gateways*, fornecendo serviços de conectividade e aplicativos. A Figura 3 demostra a arquitetura LoRa.

Figura 3 – Arquitetura LoRa



Fonte: (NOURA *et al.*, 2020).

A LoRa é uma tecnologia de código aberto, o que significa que seu protocolo de comunicação é público e permite a interoperabilidade entre diferentes fabricantes. Isso facilita a adoção e o desenvolvimento de soluções baseadas em LoRa por diferentes empresas e setores.

Alguns exemplos de aplicação em redes LoRa são: redes externas como cidades inteligentes (LEMOS, 2013), fábricas 4.0 (SANTOS *et al.*, 2018), agricultura e entre outros.

Segundo Aras *et al.* (2017), LoRaWan é o protocolo de rede para dispositivos baseados em LoRa e oferece segurança por meio de criptografia de chave simétrica. No entanto, mesmo com as medidas de segurança do LoRaWan, os dispositivos LoRa apresentam vulnerabilidades de segurança. Por exemplo, devido ao tempo de transmissão longo do LoRa, existe o risco de interceptação ou corrupção de pacotes antes que eles cheguem ao *gateway*. Além disso, o LoRa é vulnerável a ataques de interferência de sinal ou *jamming*, nos quais os sinais de comunicação são intencionalmente interrompidos. Para mitigar essas vulnerabilidades, são necessárias medidas de segurança adicionais e soluções para proteger os dispositivos LoRa contra ataques.

2.1.3 Rede Bluetooth

O Bluetooth é um protocolo sem fio que permite a comunicação entre dispositivos em curtas distâncias (NIST, 2014). A principal diferença entre o protocolo 802.11(Wi-Fi) e o Bluetooth é que o Bluetooth possui alcance e largura de banda menores, além de consumir menos energia em comparação com o 802.11(Wi-Fi), como é explicado por Torres (2015). O protocolo 802.11(Wi-Fi) opera na faixa de frequência de 2,4 e 5,0 GHz, enquanto o Bluetooth, utiliza a faixa de frequência de 2,4 GHz. É importante ressaltar que o Bluetooth e LoRa consomem menos energia em comparação com o Ethernet e Wi-FI.

Alguns exemplos de aplicação da rede Bluetooth incluem dispositivos pessoais, como pulseiras inteligentes, relógios inteligentes, contador de passos, teclados e mouses.

Na próxima seção, é abordado o tema segurança cibernética, uma vez que este trabalho visa estabelecer um cenário de rede utilizando métodos de segurança para manter redes IoT mais seguras.

2.2 Segurança Cibernética

De acordo com a NIST (2021a), a segurança cibernética é a prevenção de danos, proteção e restauração de computadores, sistemas de comunicações eletrônicas, serviços de comunicações eletrônicas, comunicações por fio e comunicações eletrônicas, incluindo as informações neles contidas, para garantir disponibilidade, integridade, autenticação, confidencialidade e não repúdio. A segurança cibernética também pode ser entendida como o processo da proteção de dados através da prevenção, detecção e resposta a ataques.

Segundo Tanenbaum (2003), a segurança visa garantir que pessoas mal-intencionadas não leiam ou modifiquem secretamente mensagens enviadas a outros destinatários. Também trata de situações em que mensagens legítimas são capturadas e reproduzidas, além de lidar com pessoas que tentam negar o fato de terem enviado determinadas mensagens.

O NIST (2015b) também define segurança cibernética como a proteção de informações e sistemas de informações de acesso não autorizado, uso, divulgação, interrupção, modificação ou destruição para fornecer confidencialidade, integridade e disponibilidade.

A segurança cibernética emprega várias técnicas para auxiliar na manutenção da segurança e na mitigação de ataques maliciosos quando necessário. Nesse contexto, o IDS, um sistema de detecção de intrusão, é apresentado a seguir como possível solução para aumentar a segurança dos dispositivos IoT.

2.2.1 IDS

IDS é um mecanismo de segurança utilizado para monitoramento capaz de identificar ou detectar a presença de atividades intrusivas em dispositivos de rede ou de computadores, que foi aplicado neste trabalho com o objetivo de monitorar redes IoT. A NIST (2015a) define IDS como um serviço de segurança que monitora e analisa eventos de rede ou sistema, com a finalidade de detectar e fornecer alertas em tempo real ou quase em tempo real quando ocorrem tentativas não autorizadas de acesso a recursos do sistema.

De acordo com Kizza (2013), o IDS é classificado com base em seu escopo de monitoramento. Aqueles que monitoram pacotes, analisam o tráfego são conhecidos como detecção de intrusão baseada em rede, ou NIDS. Em geral, localizados em pontos estratégico da topologia da rede, em nós configurados para isto, e possuem ampla visão do fluxo. Enquanto aqueles que atuam sob sistemas operacionais em *hosts*, analisando seus processos, programas, conexão sem ter a visão geral da rede são chamados de detecções baseadas em *host*, ou HIDS.

2.2.1.1 NIDS

O NIDS é definido pelo NIST (2006) como software que realiza captura de pacotes e análise de tráfego de rede para identificar atividades suspeitas e registrar informações relevantes. Os NIDS, além de monitorar, podem oferecer ferramentas de resposta a ataques maliciosos, auxiliando na prevenção de danos. Essas ferramentas podem incluir ações como bloqueio de tráfego indesejado ou notificação de incidentes de segurança.

Segundo a NIST (2006), os *sniffers* de pacotes são utilizados para capturar e analisar o tráfego de rede. Eles podem registrar todos os pacotes que passam entre *hosts*, fornecendo informações adicionais para análise e investigação. Além disso, os *sniffers* de pacotes também atuam como analisadores de tráfego, podendo reconstruir fluxos de pacotes e decodificar comunicações em diferentes protocolos. Eles são capazes de processar tanto o tráfego de rede ao vivo quanto pacotes que foram gravados anteriormente em arquivos de captura. Eles são extremamente valiosos na exibição de dados de pacotes brutos de forma compreensível.

A NIST (2022a) cita os *logs*, cujo estão associados ao NIDS, eles registram eventos ocorridos nos sistemas e redes de organizações. Esses *logs* podem ser analisados para identificar ações suspeitas ou depurar possíveis problema na rede ou dispositivos.

Ademais, Kizza (2013) menciona que o NIDS captura e inspeciona todo os pacotes destinados à rede, independentemente de serem permitidos ou não. O NIDS utiliza assinaturas de pacotes baseadas no conteúdo para identificar atividades maliciosas. Quando um pacote combina com uma assinatura, um alerta é gerado.

O uso do NIDS oferece várias vantagens no monitoramento e detecção de intrusões em redes:

- Capacidade de detectar ataques que um sistema baseado em *host* perderia: Como o NIDS está em uma máquina dedicada e protegida, ele é capaz de detectar ataques que podem passar despercebidos por um sistema baseado em *host* . Além disso, como os NIDS analisam o tráfego de rede em tempo real, é difícil para um invasor remover evidências de ataques.

- Detecção e resposta em tempo real: Os NIDS são colocados em pontos estratégicos da rede, permitindo detecção em tempo real de invasões e a geração de alertas imediatos para os administradores. Permitindo respostas rápidas aos incidentes de segurança.

- Capacidade de detectar ataques mal sucedidos e intenções maliciosas: Os NIDS, principalmente os de DMZ (zona desmilitarizada)(DART *et al.*, 2013), são capazes de detectar ataques que podem passar pelo *firewall* externo, mesmo se depois for barrado pelo *firewall* interno. O NIDS registra essas tentativas, permitindo uma análise posterior e o monitoramento da frequência desses ataques.

O autor Kizza (2013) menciona que o funcionamento do NIDS consiste de várias partes que trabalham em conjunto para gerar alertas. Isto é, uma arquitetura, ela consiste de dispositivo de rede, onde posicionar o NIDS na rede, o analisador de tráfego, o notificador do ataque e o gerenciador.

O dispositivo de rede, como um *network Test Access Point* (TAP), que coleta dados da rede e os distribui para os sensores do NIDS, é importante porque todo o tráfego na rede passa por ele, evitando perda de pacotes em redes de alto fluxo. O dispositivo de rede pode ser implementado como um agente de software em um sensor ou como hardware, como um roteador ou até uma placa de rede adicional.

O NIDS pode ser implementado em diferentes posições na rede, como pode-se observar na Figura 4, dependendo das necessidades de segurança e das características da infraestrutura. Algumas posições onde o sensor NIDS pode ser alocados são:

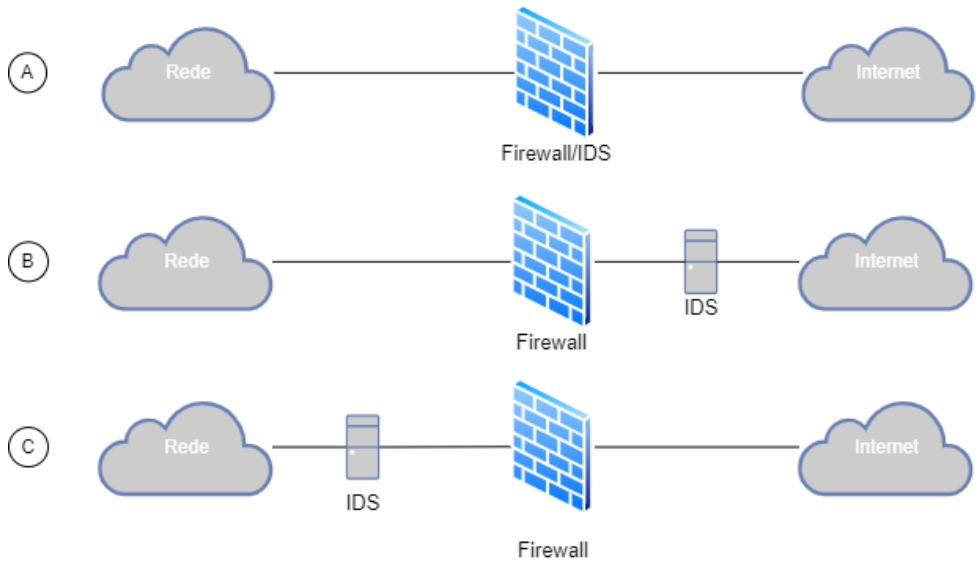
a) Dentro de *firewalls*: Permite a detecção de ataques que conseguiram passar pelas camadas de segurança externas.

b) Entre o *firewall* e a Internet: Esta posição permite que o NIDS monitore todo o tráfego que entra na rede a partir da Internet. Na parte posterior do *firewall* da rede: Essa posição lida com todo o tráfego que conseguiu passar pelo *firewall*.

c) Dentro da rede: Nessa posição o NIDS monitora o tráfego entre os diferentes dispositivos e identifica atividades suspeitas. Na DMZ de um roteador: a DMZ é uma rede de perímetro, uma zona intermediaria entre a zona externa e a rede interna (GOODRICH; TAMASSIA, 2013).

Após a coleta e análise dos dados de tráfego, o autor Kizza (2013) menciona o analisador de tráfego presente na arquitetura. O analisador classifica o tráfego suspeito com base

Figura 4 – Pontos para alocar o NIDS



Fonte: Autoria própia.

na ameaça e natureza detectadas, determinando a gravidade do ataque. Várias camadas de monitoramento podem ser feitas, na qual a camada primária determina a gravidade da ameaça e as camadas secundárias determinam o escopo, intenção e frequência da ameaça.

A arquitetura também inclui um notificador de alertas, responsável por entrar em contato com o oficial de segurança responsável pelo tratamento de incidentes quando uma ameaça é considerada grave de acordo com as políticas de segurança da organização.

Além desses componentes, a arquitetura possui um gerenciador que atua como a autoridade para controlar todo o sistema. Ele define políticas, processa os alarmes coletados e pode encaminhar as ameaças para ações apropriadas, como o roteamento de dados para um *firewall*.

Além do gerenciador, existe o subsistema de resposta, que fornece os recursos para agir com base nas ameaças dos sistemas alvo. Essas respostas podem ser geradas ou iniciadas automaticamente pelo operador do sistema. As respostas comuns incluem reconfigurar um roteador ou um *firewall* e desligar uma conexão.

Por fim, há o banco de dados, que é necessário para modelar padrões de comportamento históricos que podem ser úteis durante a avaliação de danos ou outras tarefas investigativas. Ele auxilia no desenvolvimento de padrões para indivíduos e ajuda a detectar tentativas de intrusão.

O próximo tópico aborda sobre o Snort, um NIDS, que é o responsável por monitorar rede de computadores TCP/IP, e também redes IoT. Neste trabalho, o Snort vai monitorar o tráfego de rede que passa pelo *gateway* LoRa.

2.2.1.1.1 Snort

O autor Roesch (1999) define o Snort como um NIDS com capacidade de coletar e analisar o tráfego de rede em busca de atividades suspeitas ou ataques. Possui facilidade de implantação e configuração, sendo útil como parte de uma infraestrutura de segurança de rede integrada. É descrito como alternativa de baixo custo aos sistemas comerciais de detecção de intrusões, sendo disponibilizado gratuitamente sob a Licença Pública Geral GNU.

O autor destaca que o Snort é considerado um sistema de detecção de intrusões leve por sua capacidade de ser implantado em redes pequenas. Ele é projetado para coletar e analisar pacotes de rede em tempo real, usando regras de correspondência de padrões de conteúdo para detectar uma variedade de ataques e sondagens maliciosas.

Além disso, o Snort possui um subsistema de registro e alerta. O subsistema de alerta e registro é selecionado em tempo de execução com opções de linha de comando. As opções de registro podem ser definidas para registrar pacotes em seus formatos decodificados, formato legível por humanos, ou no formato binário para um único arquivo de *log*. Existem duas opções para enviar os alertas para um simples arquivo de texto; *full* e *fast alert*. O *full* grava a mensagem de alerta e as informações do cabeçalho do pacote através do protocolo da camada de transporte. O *fast alert* grava um subconjunto condensado das informações do cabeçalho ao arquivo de alerta, permitindo maior desempenho sob carga do que o modo completo.

A seguir, é abordado o IDS de detecções baseadas em *host*, o HIDS.

2.2.1.2 HIDS

Kizza (2013) define HIDS como mecanismo de detecção de atividades maliciosas em um único computador, que utiliza software para monitorar *logs* específicos do sistema para buscas por mudanças suspeitas. Quando alterações são detectadas, o HIDS compara o novo *log* com suas assinaturas de ataque configuradas para ver se há correspondência. Caso haja, isso sinaliza a presença de atividades maliciosas.

O autor Bray, Cid e Hay (2008) define HIDS como um sistema que detecta eventos em servidores ou estações de trabalho e que podem gerar alertas semelhantes a um NIDS. No entanto, os HIDS são capazes de inspecionar todo fluxo de comunicação destinado ao *host*. Além disso, comunicações criptografadas podem ser monitoradas, pois a inspeção do HIDS pode examinar o tráfego antes que seja criptografado. Isso significa que as assinaturas do HIDS ainda serão capazes de corresponder a ataques comuns e não serão prejudicadas pela criptografia.

HIDS também são capazes de realizar verificações adicionais de nível do sistema que apenas o software IDS instalado em uma máquina hospedeira pode fazer, como verificação de integridade de arquivos, monitoramento de registro, análise de *logs*, detecção de *rootkit* e resposta ativa.

A verificação de integridade de arquivos é realizada por meio da geração de uma impressão digital única para cada arquivo no sistema operacional, conhecida como *hash* criptográfico. Essa impressão digital é gerada com base no nome e conteúdo do arquivo. HIDS podem monitorar arquivos importantes para detectar alterações nessa impressão digital quando alguém ou algo modifica o conteúdo do arquivo ou substitui o arquivo por uma versão completamente diferente.

O monitoramento do registro do sistema permite que um HIDS detecte alterações nas chaves de registro importantes. Isso ajuda a garantir que usuários ou aplicativos não estejam instalando programas novos ou modificando programas existentes com intenções maliciosas.

A detecção de *rootkit* é realizada para identificar programas desenvolvidos para obter controle encoberto sobre um sistema operacional, enquanto se escondem e interagem com o sistema no qual estão instalados. Um *rootkit* instalado pode ocultar serviços, processos, portas, arquivos, diretórios e chaves de registro do restante do sistema operacional e do usuário.

A resposta ativa permite executar automaticamente comandos ou respostas quando um evento específico ou conjunto de eventos é acionado. No entanto, isso também apresenta riscos, pois pode bloquear erroneamente tráfego legítimo ou ser explorado por um atacante para negar o acesso ao sistema.

As vantagens de utilizar HIDS são:

- Capacidade de verificar rapidamente o sucesso ou a falha de ataques: como eles registram eventos contínuos que realmente ocorreram, os HIDS têm informações mais precisas e menos propensas a falsos positivos do que o NIDS. Nesse sentido, ele complementa o NIDS não como alertas precoces, mas como sistemas de verificações.

- Monitoramento de baixo nível: como o HIDS monitora em *hosts* locais, ele pode detectar atividades de baixo nível, como acessos a arquivos, alterações nas permissões de arquivos, tentativa de instalar novos executáveis ou acessos a serviços privilegiados, alterações nos principais arquivos e executáveis do sistema e tentativas de sobrescrever arquivos vitais do sistema ou para instalar cavalos de Troia ou *backdoors* (ZEIDANLOO *et al.*, 2010). Essas atividades de baixo nível podem ser detectadas rapidamente, e os relatórios são rápidos e oportunos para permitir que o administrador tome as devidas medidas. Alguns desses ataques de baixo nível são tão sutis que os NIDS não consegue detectá-los.

- Detecção e resposta quase em tempo real: os HIDS têm a capacidade de detectar atividades nos *host* de destino e reportá-los ao administrador rapidamente em uma taxa quase em tempo real.

- Capacidade de lidar com ambientes criptografados e comutados: redes grandes são frequentemente divididas em vários segmentos menores. Cada um desses segmentos são monitorados pelo NIDS. Em redes fortemente comutadas, pode ser desafiador

determinar onde implantar o NIDS para obter cobertura de rede adequada. Esse problema pode ser resolvido usando técnicas como espelhamento de tráfego e portas administrativas em *switches*, mas nem sempre são eficazes. O HIDS por sua vez, permite a implantação em quantos *hosts* forem necessários, proporcionando maior visibilidade necessária em ambientes comutados. Além disso, como opera no próprio sistema, ele consegue visualizar o tráfego de entrada sem criptografia, diferente do NIDS.

- Custo-benefício: como não são necessários dispositivos adicionais para instalar o HIDS, isso pode resultar em grandes economias. Isso se compara favoravelmente com os grandes custos de instalação de NIDS, que requerem servidores dedicados e caros.

No entanto, o HIDS também apresenta desvantagens, que incluem:

- Visão limitada da rede: por ser implantados em *hosts*, os HIDS têm visualização da rede limitada.
- Vulnerabilidade a adulterações ilegais: por estar próximos aos usuários, são mais suscetíveis a adulterações ilegais, como alterações nos arquivos de *log*.
- Sobrecarga de análise de dados: devido a grande quantidade de *logs* gerados, a análise desses dados brutos podem colocar sobrecargas significativas tanto no poder de processamento necessário para analisá-los quanto na equipe de segurança necessária para revisá-los.

O próximo tópico a ser abordado é o Ossec, um HIDS, que é o responsável por monitorar os *hosts* IoT, neste trabalho o *host* é o *gateway* LoRa.

2.2.1.2.1 OSSEC

Vukalović e Delija (2015) definem o OSSEC, como um sistema de detecção de intrusão baseado em *host* (HIDS). O OSSEC é um sistema de detecção de intrusão de código aberto que examina *logs* gerados por diversas aplicações para detectar invasões. Possui dois modos de operação: local, que monitora um único sistema, e agente/servidor, que coleta e monitora *logs* de várias fontes em toda a rede. Com a capacidade de ler e analisar arquivos de *log* de mais de 40 programas e dispositivos diferentes, a OSSEC possui componentes separados para coletar, ler, analisar *logs* e enviar alertas por e-mail. Também é possível instalar agentes OSSEC em dispositivos de rede, que coletam *logs* e os enviam para um servidor central para análise usando o protocolo UDP. Ao receber os *logs*, a OSSEC extrai os valores dos campos, identifica informações cruciais e verifica-os em relação a regras predefinidas ou criadas manualmente. Caso necessário, envia alertas por e-mail. A OSSEC é compatível com a maioria dos sistemas operacionais.

Considerando os dois tipos de IDS, o NIDS e o HIDS, conclui-se que ambos trazem para a segurança suas próprias vantagens e fraquezas que se complementam e podem aumentar a segurança das redes. Portanto, uma boa opção para manter ambientes melhor monitorados e possivelmente mais seguros é utilizar os dois tipos de IDS. Por exemplo, pode ser estabelecido NIDS para monitorar redes onde há um *host* com HIDS instalado.

Para aproveitar ao máximo o NIDS, é necessário utilizar dispositivos de rede, como *hubs*, *switches* e *network* TAP para a realização do monitoramento da rede. O próximo tópico aborda o tema de dispositivos de rede.

2.2.2 Dispositivos de rede

Os dispositivos de rede para auxiliar o monitoramento da rede no contexto do NIDS podem ser *hubs*, *switches* com portas *Switch Port Analyzer* (SPAN) ou *network* TAP, além de outros. Esses dispositivos espelham o tráfego de rede ou realizam cópias desse tráfego, funcionando como um concentrador.

O *hub* é definido pela NIST (2021b) como um ponto de conexão comum para dispositivos em rede. Geralmente, os *hubs* são usados para transmitir dados de um dispositivo (ou segmento) para outro. Segundo Tanenbaum (2003), *hub* possui várias linhas de entrada que conecta eletricamente. Os quadros que chegam em qualquer dessas linhas são enviados para todas as outras. Se dois quadros chegarem ao mesmo tempo, ocorrerá uma colisão, como ocorre em cabos coaxiais. Em outras palavras, o *hub* forma um único domínio de colisão. Todas as linhas que se conectam ao *hub* devem operar na mesma velocidade.

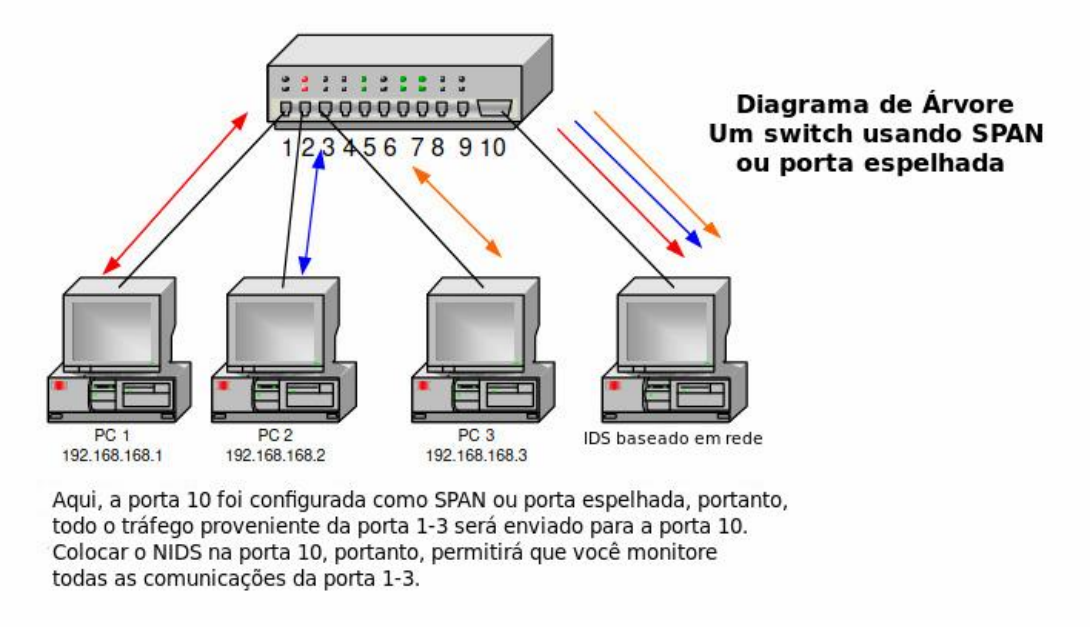
Tanenbaum (2003) também define os *switches*, que se baseiam no roteamento em endereços de quadro. O *switch* ativamente encaminha o quadro de A até B, pois não há outro caminho que o quadro possa seguir. Cada porta do *switch* normalmente se conecta a um único computador, e os *switches* precisam ter espaço para várias placas de linha. Cada placa de linha fornece espaço de *buffer* para os quadros que chegam às suas portas. Como cada porta representa seu próprio domínio de colisão, os *switches* não perdem quadros devido a colisões, diferente do *hub*. Todavia, se os quadros chegarem com velocidade maior do que a que podem ser retransmitidos, o *switch* poderá ficar sem espaço de *buffer* e começará a descartar quadros.

De acordo com a NIST (2022a), a porta SPAN do *switch* oferece monitoramento passivo para anomalias baseadas em rede e recupera informações sobre terminais dentro da rede. Na Figura 5, é mostrado o funcionamento da porta SPAN.

Os TAP são dispositivos que se conectam passivamente à Ethernet ou Fibra e fazem cópias de todos os dados que passam por eles (EDWARDS, 2002). Normalmente, são projetados para serem tolerantes a falhas com conexões principais cabeadas, garantindo que, em caso de perda de energia da unidade, a conexão principal permanece aberta.

Edwards (2002) também destaca as vantagens que os TAP oferecem sobre as portas SPAN. Primeiro, os TAP têm zero impacto sobre a rede ou sua infraestrutura. Isso significa que

Figura 5 – Funcionamento da porta SPAN do switch



Fonte: (EDWARDS, 2002).

não há necessidade de alterar a configuração dos roteadores ou *switches*, evitando impactos no desempenho. Em segundo lugar, os TAP permitem que os agentes de segurança obtenham suas próprias cópias do tráfego de rede, mantendo-o separado da infraestrutura principal. Em terceiro lugar, os TAP protegem o dispositivo no final da conexão em que estão instalados, pois apenas o tráfego de transmissão (TX) é copiado conforme ilustrado na Figura 6. Isso impede que possíveis invasores se conectem aos dispositivos no final da conexão monitorada, mesmo que conheçam o endereço IP, pois nenhum dado será enviado de volta à conexão.

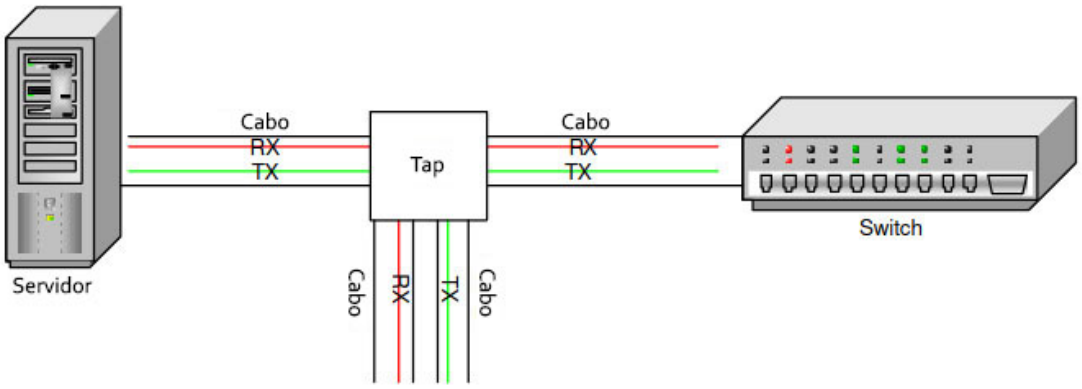
Ele menciona que cada cabo de transmissão (TX) é emendado (e, portanto, requer dois cabos TAP), de modo que não se pode simplesmente conectar NIDS a cada emenda TX, pois haverá apenas metade da conversação. Portanto, é necessário reagregar essas duas conexões antes de enviá-las para o NIDS, de modo que, será necessário um IDS *Balancer* para que a solução funcione. O IDS *Balancer* entende e acompanha as conversas e, independentemente da porta que os pacotes chegam, o IDS reagrega o tráfego antes de enviá-lo ao NIDS, garantindo que veja toda a conversa.

Visto os três dispositivos de rede. A seguir, é apresentado os ataques cibernéticos que mais acontecem em redes IoT, como também em redes LoRa.

2.3 Ataques Cibernéticos

A NIST (2022b) define ataques cibernéticos como aqueles que ocorrem no ciberespaço e que tem o objetivo de interromper, desabilitar, destruir ou controlar maliciosamente ambientes

Figura 6 – Dispositivo de rede: TAP



Fonte: (EDWARDS, 2002).

de computação, além de comprometer a integridade de dados ou roubar informações sensíveis. Esses ataques podem ter consequências catastróficas.

Hassija *et al.* (2019), analisam os desafios de segurança e das fontes de ameaça nos dispositivos IoT. Os dispositivos IoT apresentam desafios segurança específicos, como problemas de privacidade, autenticação, gerenciamento e armazenamento de informações. Esses desafios podem se tornar vulnerabilidades, que são discutidas a seguir.

Em qualquer ambiente IoT, existem quatro camadas importantes. A primeira camada consiste no uso de sensores e atuadores para coletar os dados e executar várias funcionalidades. Com base nisso, na segunda camada, uma rede de comunicação é utilizada para transmitir os dados coletados. A maioria dos dispositivos IoT em desenvolvimento utiliza a terceira camada chamada *middleware*, que atua como uma ponte entre a rede e a camada de aplicação. Por fim, na quarta camada, existem diversos dispositivos IoT de ponta a ponta, como redes inteligentes, transporte inteligente e fábricas inteligentes. Além dessas camadas, existem vários *gateways* que conectam essas camadas e auxiliam no fluxo de dados.

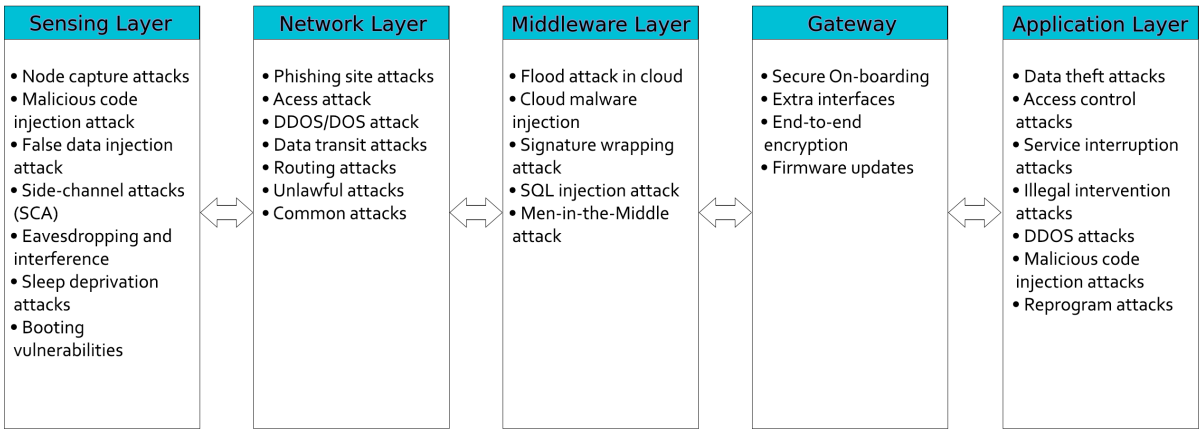
A Figura 7 ilustra as fontes de ameaças em ambientes IoT, mostrando os possíveis ataques que ocorrem em cada camada e no *gateway*.

2.3.1 *Sensing Layer*

A camada de detecção é responsável principalmente pelos sensores e atuadores físicos na IoT. Os sensores detectam fenômenos físicos que acontecem ao seu redor, enquanto os atuadores realizam ações com base nos dados detectados. As principais ameaças de segurança encontradas na camada de detecção são:

1. Captura de Nós (*Node Capturing*): as aplicações IoT envolvem diversos nós de baixa potência, como sensores e atuadores. Esses nós são vulneráveis a uma variedade de

Figura 7 – Ataques por camadas / gateway



Fonte: (HASSIJA *et al.*, 2019).

ataques. Invasores podem tentar capturar ou substituir um nó no sistema IoT por um nó malicioso controlado pelo invasor. Isso compromete a segurança do IoT.

2. Ataque de Injeção de Código Malicioso (*Malicious Code Injection Attack*): o *firmware* ou *software* dos nós IoT são frequentemente atualizados por meio de redes sem fio, proporcionando uma porta de entrada para invasores injetarem código malicioso na memória do nó. Usando esse código malicioso, os invasores podem forçar os nós a executarem funções não intencionais ou tentar acessar o sistema IoT.

3. Ataque de Injeção de Dados Falsos (*False Data Injection Attack*): uma vez que um nó é capturado, o invasor pode usá-lo para injetar dados falsos no sistema IoT. Isso pode levar a resultados falsos e mau funcionamento do ambiente IoT. O invasor também pode usar esse método para causar ataques DDoS.

4. Ataques de Canal Lateral (*Side-Channel Attacks* (SCA)): além de ataques diretos aos nós, diversos ataques de canal lateral podem resultar no vazamento de dados confidenciais. Esses ataques podem ser baseados no consumo de energia, temporização ou eletromagnéticos.

5. Espionagem e Interferência (*Eavesdropping and Interference*): as aplicações IoT consistem em vários nós implantados em ambientes abertos. Como resultado, esses dispositivos IoT estão expostos a invasores que podem espionar e capturar dados durante diferentes etapas, como transmissão de dados ou autenticação.

6. Ataques de Privação de Sono (*Sleep Deprivation Attacks*): invasores tentam esgotar a bateria dos dispositivos IoT de baixa potência, resultando em negação de serviço dos nós no IoT devido à descarga da bateria. Esse tipo de ataque pode ser realizando executando laços infinitos nos dispositivos usando código malicioso ou aumentando artificialmente o consumo de energia desses dispositivos.

7. Ataques durante a Inicialização (*Booting Attacks*): dispositivos como roteadores são vulneráveis a vários ataques durante o processo de inicialização, uma vez que os mecanismos de segurança integrados podem não estar habilitados nesse momento. Invasores podem aproveitar essa vulnerabilidade e tentar atacar os dispositivos quando estão sendo reiniciados. É essencial proteger o processo de inicialização, especialmente considerando que os dispositivos de borda geralmente são de baixa potência e passam por ciclos de sono-vigília.

2.3.2 *Network Layer*

A camada de rede é responsável por transmitir as informações da camada de detecção para a unidade de processamento para que essas informações sejam processadas. Os principais problemas de segurança encontrados na camada de rede são:

1. Ataque de Phishing (*Phishing Site Attack*): existe a possibilidade de encontrar sítios de *phishing* quando usuários visitam páginas da Web na Internet. Depois que a conta e a senha do usuário são comprometidas, o ambiente IoT usado pelo usuário fica vulnerável a ataques cibernéticos.

2. Ataque de Acesso (*Access Attack*): um indivíduo não autorizado ou invasor obtém acesso à rede IoT. O invasor pode permanecer na rede sem ser detectado por um longo período. O objetivo desse tipo de ataque é roubar dados ou informações valiosas, em vez de causar danos à rede.

3. Ataque de Negação de Serviço (*Denial of Service* (DoS)/DDoS): o invasor sobrecarrega os servidores de destino com um grande número de solicitações indesejadas. Isso incapacita o servidor de destino, interrompendo assim os serviços para usuários legítimos. Se várias fontes forem usadas pelo invasor para sobrecarregar o servidor de destino, esse ataque será chamado DDoS. Muitos dispositivos IoT não são fortemente configurados e, portanto, tornam-se alvos fáceis para os invasores lançarem ataques DDoS.

4. Ataques aos Dados em Trânsito (*Data Transit Attacks*): Os dispositivos IoT lidam com grande quantidade de armazenamento e troca de dados, que são valiosos e, portanto, são alvo de invasores. Os dados em trânsito de um local para outro são altamente vulneráveis a ataques cibernéticos. Em dispositivos IoT, há grande movimentação de dados entre sensores, atuadores e nuvem. Portanto, os dispositivos IoT são suscetíveis a violações de dados.

5. Ataques de Roteamento (*Routing Attacks*): nós maliciosos em dispositivos IoT podem tentar redirecionar os caminhos de roteamento durante o trânsito de dados. Os ataques

Sinkhole são um tipo específico de ataque de roteamento em que o invasor anuncia um caminho de roteamento artificialmente mais curto e atrai nós para rotear o tráfego por meio dele. Um ataque de *worm-hole* é outro tipo de ataque que pode representar ameaça à segurança se combinado com outros ataques, pois é uma conexão fora de banda entre dois nós para transferência rápida de pacotes, permitindo que o invasor crie o *worm-hole* entre um nó comprometido e o dispositivo conectado à Internet, tentando contornar os protocolos básicos de segurança em dispositivos IoT.

2.3.3 *Middleware Layer*

A camada de *middleware*, que atua como uma camada de abstração entre a camada de rede e a camada de aplicação, fornece recursos poderosos de computação e armazenamento. Essa camada inclui armazenamentos de dados persistentes, sistemas de enfileiramento, aprendizado de máquina e o uso de *brokers*, que podem validar, armazenar, rotear e entregar mensagens aos destinos apropriados. No entanto, a camada de *middleware* também pode ser alvo de diversos ataques, incluindo:

1. Ataque Man-in-the-Middle (*Man-in-the-Middle Attack*): o protocolo *Message Queuing Telemetry Transport* (MQTT) utiliza o modelo de comunicação de publicação-assinatura entre clientes e assinantes usando o *broker* MQTT, que atua como intermediador. Isso ajuda a separar a publicação e os clientes assinantes um do outro, permitindo que as mensagens sejam enviadas sem o conhecimento do destino. Se o invasor conseguir controlar o protocolo e se tornar um *man-in-the-middle*, ele poderá obter controle total de toda a comunicação sem que os clientes percebam.

2. Ataque de Injeção de SQL (*SQL Injection Attack*): o invasor pode injetar instruções SQL maliciosas em programas. Dessa forma, os invasores podem obter dados privados de qualquer usuário e até mesmo alterar registros no banco de dados.

3. Ataque de Quebra de Assinatura (*Signature Wrapping Attack*): o invasor compromete o algoritmo de assinatura, que é responsável por garantir integridade e autenticação dos dados. Isso permite que o invasor execute operações ou modifique as mensagens espionadas explorando vulnerabilidades.

4. Injeção de Malware na Nuvem (*Cloud Malware Injection*): o invasor pode obter controle e injetar código malicioso ou até mesmo injetar uma máquina virtual na nuvem. O invasor finge ser um serviço válido, tentando criar uma instância de máquina virtual ou um módulo de serviço mal-intencionado. Dessa forma, o invasor pode obter acesso às solicitações de serviço da vítima e capturar dados confidenciais, os quais podem ser modificados conforme a instância.

5. Ataque de Sobrecarga na Nuvem (*Flooding Attack in Cloud*): este ataque funciona de forma semelhante ao ataque DoS na nuvem e afeta a qualidade do serviço. Os invasores enviam continuamente várias solicitações a um serviço para esgotar os recursos da nuvem. Esses ataques podem ter um grande impacto nos sistemas em nuvem, aumentando a carga nos servidores em nuvem.

2.3.4 *Gateway*

O *Gateway* desempenha um papel importante na conexão de dispositivos, pessoas e serviços em ambiente IoT. Eles fornecem soluções de hardware e software para dispositivos IoT, além de serem responsáveis por cifrar e decifrar dados IoT, bem como traduzir protocolos para permitir a comunicação entre diferentes camadas (HASSIJA *et al.*, 2019). No entanto, a segurança é um desafio significativo para os *gateways* IoT, e alguns dos desafios comuns incluem:

1. Integração Segura (*Secure On-boarding*): quando um novo dispositivo ou sensor é instalado em um sistema IoT, é importante proteger as chaves de criptografia. Os *gateways* atuam como intermediários entre os novos dispositivos e os serviços de gerenciamento, e todas as chaves passam pelos *gateways*. Portanto, são suscetíveis a ataques *man-in-the-middle* e espionagem, que visam capturar as chaves de criptografia, especialmente durante o processo de integração.

2. Interfaces Adicionais (*Extra Interfaces*): uma estratégia importante para a segurança é minimizar o número de interfaces disponíveis nos dispositivos IoT. Os fabricantes de *gateway* IoT devem implementar apenas as interfaces e os protocolos necessários. Além disso, serviços e funcionalidades devem ser restritos aos usuários finais para evitar autenticação de *backdoor* ou violação de informações.

3. Criptografia de Ponta a Ponta (*End-to-End Encryption*): geralmente, os *gateways* são usados para obter e aplicar atualizações de *firmware* nos dispositivos IoT. A versão atual e a nova do *firmware* devem ser registradas, e a validade das assinaturas deve ser verificada para atualizações seguras. Além disso, é essencial que apenas o destinatário exclusivo possa descriptografar as mensagens criptografadas. A operação de decifrar realizada no nível do *gateway* pode tornar os dados suscetíveis a violações de dados, pois não é criptografia de ponta a ponta. Os *gateways* precisam tratar as mensagens criptografadas ao traduzir as informações de um protocolo para outro.

4. Atualizações de Firmware (*Firmware updates*): os *gateways* desempenham um papel na obtenção e aplicação de atualizações de *firmware* nos dispositivos IoT. É importante registrar a versão atual e a nova do *firmware* e verificar a validade das assinaturas para

garantir atualizações seguras. As atualizações de *firmware* podem trazer melhorias de segurança, correções de falhas e novos recursos para os dispositivos IoT.

2.3.5 *Appliation Layer*

A camada de aplicação é responsável por fornecer diretamente serviços aos usuários finais. No entanto, essa camada enfrenta problemas de segurança específicos que são distintos das outras camadas, como roubo de dados e problemas de privacidade. Os principais problemas de segurança encontrados na camada de aplicação são:

1. Roubo de dados (*Data Thefts*): os aplicativos IoT lidam com grandes quantidades de dados críticos e privados. Os dados em trânsito são vulneráveis à ataques, e nos dispositivos IoT há grande movimentação de dados. Para proteger aplicativos IoT contra roubos de dados, são utilizados técnicas como criptografia de dados, isolamento de dados, autenticação de usuários e redes, e gerenciamento de privacidade.

2. Ataques de controle de acesso (*Access Control Attacks*): o controle de acesso é um mecanismo de autorização que permite que apenas usuários ou processos legítimos acessem os dados ou contas. Os ataques de controle de acesso são críticos em dispositivos IoT, pois comprometer o acesso pode tornar o sistema vulnerável a ataques.

3. Ataques de interrupção de serviço (*Service Interruption Attacks*): esses ataques, conhecidos também como ataques de interrupção ilegal ou ataques DDoS, têm como objetivo privar usuários legítimos de utilzar serviços IoT, sobrecarregando artificialmente os servidores ou a rede, tornando-os incapazes de responder adequadamente.

4. Ataques de injeção de código malicioso (*Malicious Code Injection Attacks*): se o sistema for vulnerável a *scripts* maliciosos e direcionamentos incorretos devido a verificações de código insuficientes, esse poderá ser o ponto de entrada que invasores escolheriam. Os invasores exploram técnicas como *Cross-Site Scripting* (XSS) para injetar *scripts* malicioso em sítios confiáveis. Um ataque XSS bem-sucedido pode resultar no sequestro de dispositivos IoT e paralisar sistemas IoT.

5. Ataques de sniffing (*Sniffing Attacks*): os invasores podem usar *sniffers* para monitorar o tráfego de rede em aplicativos IoT. Isso pode permitir que obtenham acesso a dados confidenciais do usuário, caso não haja protocolos de segurança adequados implementados para evitar esses ataques.

6. Ataques de reprogramação (*Reprogram Attacks*): se o processo de configuração não estiver protegido, os invasores podem tentar reprogramar os objetos IoT remotamente. Isso pode levar ao sequestro da rede IoT, permitindo que os invasores assumam o controle dos dispositivos e executem ações indesejadas.

Como vimos, o artigo de Hassija *et al.* (2019) discute ameaças de segurança em diferentes camadas de redes IoT, incluindo camada de detecção, rede, *middleware*, *gateways* e aplicação. Esses ataques e vulnerabilidades servirão para estudos futuros, serão simulados na rede LoRa com o intuito de verificar se o IDS conseguirá identificar o ataque. A seguir, é apresentados mais ataques em IoT, no entanto, baseados em LoRaWan, pois esse trabalho realiza em prática o monitoramento de redes IoT em rede LoRa. Esses ataques baseados em LoRaWan serão considerados trabalhos futuros para avaliar a validade do monitoramento nesse contexto.

2.3.6 Ataques específicos na rede LoRa

Noura *et al.* (2020) descreve diferentes tipos de ataques e vulnerabilidades em redes LoRaWan e propõe contramedidas para prevenir algumas das vulnerabilidades existentes. Os ataques são:

1. Ataques de autenticação:
    - Ataque Man-in-the-Middle: O atacante intercepta as mensagens entre o dispositivo final e o *gateway*, modifica o *payload* e utiliza chaves comprometidas para assinar a mensagem modificada e enviá-la para o *gateway*. Para prevenir esse ataque, propõe-se o uso de uma função não linear e não invertível para relacionar a integridade da mensagem (*Message Integrity Code* (MIC)) com a chave de criptografia *h(AppKey)*.
2. Ataques de disponibilidade:
    - Ataques de *Sinkhole*: Um nó malicioso direciona o tráfego da rede para um nó específico, comprometendo a disponibilidade do sistema. O uso de IDS pode ajudar a detectar e impedir esses ataques.
    - Ataque de repetição: O atacante utiliza técnicas de *jamming* para bloquear a comunicação entre o dispositivo final e a rede, impedindo a transmissão de pacotes. Esse tipo de ataque pode levar a ataques DoS se o atacante inundar o dispositivo com um grande número de mensagens. Propõe-se que o dispositivo final passe pelo procedimento de ativação periodicamente para obter novas chaves de sessão e a adição de carimbos de tempo ou contadores nos cabeçalhos das mensagens.
    - Ataque de roteamento *down-link*: O atacante intercepta o tráfego entre o dispositivo e o *gateway*, retransmitindo-o para uma rede comprometida. Isso pode resultar em pacotes duplicados, afetando a disponibilidade do sistema. Para prevenir esse ataque, pode-se usar autenticação e verificação dos pacotes recebidos do *gateway*.

- Ataque de repetição de *join-accept*: O atacante envia uma mensagem de *join-accept* falsa para um dispositivo antes que ele receba a mensagem autenticada do servidor de rede. Isso impede que o dispositivo envie ou receba pacotes e interrompe a comunicação com o servidor de rede. Uma proposta de contramedida é o uso de um número pseudoaleatório para autenticar a mensagem de *join-accept*.
- Ataque de sincronização de *beacon*: O atacante compromete o *gateway* e envia sinais (*beacons*) falsos para os dispositivos finais. Isso leva os dispositivos a abrir janelas de recepção não confirmadas, aumentando as colisões entre os pacotes transmitidos. Uma contramedida proposta é o uso de uma chave no *gateway* para autenticar a transmissão.
- Ataque de *spoofing* de ACK: O atacante usa mensagens ACK de *down-link* para confirmar mensagens de *up-link* de dispositivos. O atacante pode bloquear a recepção de ACKs legítimos, enganando dispositivos para pensar que suas mensagens foram transmitidas com sucesso. Uma contramedida proposta é o uso de MIC para verificar a integridade da mensagem.
- Ataque de inundação de rede: O atacante usa um ou mais dispositivos finais para inundar a rede LoRaWan com pacotes, comprometendo a disponibilidade da rede. Restrições de tempo de transmissão podem ser adicionadas para resistir a esse tipo de ataque.
- Ataque de encaminhamento seletivo: O invasor encaminha seletivamente pacotes na rede, comprometendo a disponibilidade. O uso de IDS pode ajudar a detectar e prevenir esse tipo de ataque.
- Ataque de *jamming*: O atacante transmite sinais de rádio na mesma frequência de transmissão para interrompê-la.Esse tipo de ataque, pode ser evitado detectando os dispositivos que apresentam comportamentos anormais. Isso pode ser feito enquanto o ataque está em execução, pois todos os dispositivos de comunicação maliciosos serão detectados e descartados da rede. Além disso, para preservar a disponibilidade do dispositivo, o administrador da rede pode mudar a transmissão para outra banda de frequência.

3. Ataques de confidencialidade:

    - Espionagem: O LoRaWan usa AES-128 no modo de contagem para garantir a confidencialidade das mensagens. No entanto, se o contador transbordar, o mesmo fluxo de chaves será produzido novamente. Isso pode levar à quebra da confidencialidade das comunicações. Atualizar as chaves de sessão da rede e da aplicação podem prevenir esse tipo de ataque.

- Análise de tráfego de rede: Envolve a captura e análise dos pacotes transmitidos através do *gateway* para obter informações a respeito do tráfego e comprometer a confidencialidade e a privacidade do sistema. Para prevenir esse ataque, mecanismos de criptografia podem ser implementado para proteger os pacotes transmitidos. Além disso, o uso de identificações variáveis e diferentes para cada sessão torna os ataques de análise de tráfego mais difíceis.

4. Ataques de integridade:

    - Ataque de inversão de bits: Envolve a modificação de bits específicos no texto cifrado. Uma contramedida proposta é executar o cálculo do MIC no servidor de aplicativos em vez de verificar no servidor de rede.

Esses são alguns dos tipos de ataques mais comuns em redes LoRaWan, classificados de acordo com seus objetivos de autenticação, disponibilidade, confidencialidade e integridade. Para garantir a segurança da rede e dos dispositivos LoRaWan, é importante implementar medidas de segurança adequados, como criptografia robusta, autenticação forte, gerenciamento adequado de chaves e monitoramento constante para detectar e mitigar possíveis ataques. No entanto, é importante observar que existem outras ameaças, bem como contramedidas que podem ser aplicadas para garantir a segurança e a confiabilidade das redes LoRaWan.

Além disso, essa análise de ataques serve como referência para trabalhos futuros que visam testar a validade do monitoramento de redes LoRaWan para detectar e mitigar esses tipos de ataques em ambientes comprometidos.

2.4 Considerações do Capítulo

No presente capítulo, foi explorado tópicos fundamentais para a compreensão do trabalho. A partir da seção a respeito de IoT, é abordado a importância de garantir a segurança dos dispositivos IoT cada vez mais conectados. Ao abordar os conceitos de NIDS e HIDS, é possível compreender as diferentes abordagens de monitoramento e detecção de intrusões em redes e sistemas hospedeiros. A análise dos diversos tipos de ataques, como *malware*, *phishing* e DoS, ressaltou a necessidade de estar ciente das ameaças em constante evolução e tomar medidas proativas para se proteger. Esses conceitos servem como base sólida para o desenvolvimento deste trabalho.

Portanto, discutido os conceitos básicos necessários para o monitoramento de redes IoT, o capítulo a seguir apresenta os trabalhos relacionados ao monitoramento em redes IoT, com um foco específico em redes LoRaWan. Esse enfoque se deve à disponibilidade e viabilidade de realização de testes em redes LoRaWan. Foi explorado o desenvolvimento de soluções de monitoramento de tráfego, com o objetivo de identificar padrões de atividade suspeitos, detectar

possíveis ataques e assegurar a integridade e a confidencialidade das comunicações nesse contexto específico.

3 TRABALHOS RELACIONADOS

Neste capitulo, são apresentados trabalhos relacionados que abordam o tema de segurança em dispositivos IoT e propõem soluções para mitigar os riscos associados a esses dispositivos. Os estudos têm em comum a preocupação em oferecer proteção por meio de gerenciadores, arquiteturas ou IDS.

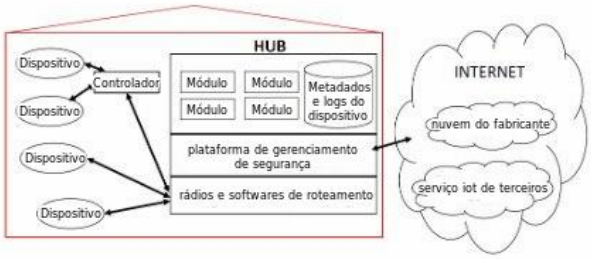
3.1 Gerenciador de Segurança utilizando dispositivos de rede para dispositivos IoT

Simpson, Roesner e Kohno (2017) propõem um gerenciador de segurança construído sob o *hub* ou roteador de *gateway* de casas inteligentes. O objetivo desse gerenciador é interceptar todo o tráfego direcionado aos dispositivos IoT, monitorar constantemente seu estado e vulnerabilidades conhecidas, e intervir quando necessário para mitigar os riscos de segurança. Além disso, o gerenciador realiza filtragem de tráfego, fortalece as autenticações dos dispositivos e facilita a atualização de software.

Os autores destacam que a atualização de dispositivos vulneráveis é uma solução eficaz para mitigar riscos, porém, enfrenta desafios práticos. Alguns dispositivos não possuem a capacidade de receber atualizações, outros são abandonados pelas empresas fabricantes, mesmo que ainda estejam funcionais, e alguns dependem da ação do usuário, o que nem sempre ocorre.

O ciclo de vida da vulnerabilidade e suas soluções são discutidos no trabalho. O *hub*, como pode-se ver na Figura 8, age em diferentes etapas desse ciclo, identificando dispositivos afetados, filtrando o fluxo de ataques, aumentando a autenticação, colocando em quarentena dispositivos comprometidos e informando o usuário sobre o estado da rede. Além disso, há uma lista de vulnerabilidades comuns e alerta o usuário sobre a disponibilidade de atualizações para dispositivos afetados.

Figura 8 – Hub



Fonte: (SIMPSON; ROESNER; KOHNO, 2017).

Em relação ao *design* do gerenciador de segurança, o *hub* se posiciona como um intermediário entre os dispositivos IoT e outras partes, interceptando a comunicação. Ele é projetado para ser consciente de todos os dispositivos IoT, determinar os momentos adequados para reiniciar os dispositivos e buscar e instalar atualizações.

Em suma, o trabalho propõe o desenvolvimento de gerenciadores de segurança baseados em *hubs* ou roteadores para dispositivos IoT. Essas soluções oferecem proteção adicional, identificam vulnerabilidades, filtram o tráfego, fortalecem a autenticação, facilitam atualizações de software e notificam os usuários sobre a segurança da rede. Portanto, este trabalho relacionado serviu como inspiração para a concepção do presente trabalho, que tem como objetivo desenvolver um cenário de rede com o monitoramento, paralelizando a ideia de utilizar os IDS como um gerenciador de segurança para oferecer proteção às redes IoT.

3.2 IDS utilizados em redes LoRa

No artigo do autor Danish *et al.* (2018), descrevem o protocolo LoRaWan, que é um protocolo de rede de longa distância e baixa potência projetado para permitir que nós operados por bateria se comuniquem entre si. No entanto, existem vulnerabilidades no mecanismo de segurança do procedimento de adesão do LoRaWan. Um atacante pode lançar um ataque de negação de serviço (DoS) usando um *jammer* para desconectar permanentemente os nós finais do LoRa da rede LoRaWan. Neste artigo, é proposto um IDS baseado em LoRaWan para detectar ataques de *jamming*. O IDS é treinado com dados reais de solicitação de adesão e utiliza os algoritmos KLD e HD (GAO *et al.*, 2020) para detectar ataques de *jamming*. O KLD compara a similaridade estatística entre as distribuições de números aleatórios gerados durante o procedimento de adesão, enquanto o HD compara a distância de Hamming (NOROUZI; FLEET; SALAKHUTDINOV, 2012) entre os números aleatórios consecutivos. O IDS é implantado em um ambiente de teste experimental no *gateway* com nós finais LoRa e servidor de rede para monitorar os padrões de tráfego em tempo real. As avaliações de desempenho mostram altas taxas de detecção para ambos os algoritmos, com taxas de falsos positivos baixas. A proposta contribui para melhorar a segurança do LoRaWan contra ataques de *jamming*.

Esse artigo serve como trabalho relacionado, pois este trabalho tem como objetivo monitorar redes IoT. No entanto, o trabalho relacionado propõe criar um IDS para detectar ataques de *jamming* utilizando os algoritmos KLD e HD em redes LoRa. Diferente da proposta deste trabalho, que consiste em utilizar o Snort, um IDS *open source*, para monitorar *gateways* em redes IoT e detectar qualquer atividade suspeita, independentemente do tipo de ataque. Essa abordagem permite uma análise abrangente do tráfego de rede, fornecendo detecção mais ampla e flexível de possíveis ameaças.

3.3 Cidade inteligente utilizando LoRaWan e IDS

O autor Elsaeidy *et al.* (2017) apresentam uma arquitetura de plataforma para proteger cidades inteligentes contra ataques cibernéticos. As tecnologias de banda estreita são essenciais para cidades inteligentes, mas também têm limitações de segurança. É proposto um modelo

baseado em aprendizado profundo para detectar e bloquear ataques de negação de serviço (DoS) em tempo real para aplicações de cidades inteligentes e extrair características de rede de alto nível e padrões dos dados históricos de atacantes. Essas características são usadas para identificar se um novo usuário é normal ou atacante com base em seus dados comportamentais.

O texto destaca a importância das tecnologias de banda estreita, como LoRaWan, para atender aos requisitos das cidades inteligentes. Essas tecnologias oferecem vantagens como baixa latência, alta taxa de transferência de dados, grande área de cobertura e confiabilidade. No entanto, também apresentam vulnerabilidades de segurança, tornando-se alvos atraentes para ataques cibernéticos. Os IDS são mencionados como ferramentas utilizadas para identificar e detectar ataques maliciosos. Eles desempenham um papel crucial na proteção do sistema contra ataques prejudiciais, detectando e mitigando as consequências dos ataques.

Portanto, o artigo correlaciona IDS e LoRaWan, destacando a importância dos sistemas de detecção de intrusões para garantir a segurança das infraestruturas de cidades inteligentes e mencionando o uso do LoRaWan como tecnologia de comunicação de banda estreita nessas aplicações.

3.4 Arquitetura de detecção de intrusão utilizando o Snort com o IoT Raspberry Pi

Sforzin *et al.* (2016), discutem a correlação entre o Snort e a IoT. O texto propõe uma arquitetura de detecção de intrusão para a IoT, utilizando dispositivos de baixo custo, como o Raspberry Pi, executando o Snort. Foram realizados experimentos para avaliar o desempenho do Raspberry Pi (RICHARDSON; WALLACE, 2012) ao executar o Snort como um sistema de detecção de intrusão em um ambiente distribuído, como a IoT.

A principal motivação do trabalho é encontrar uma solução robusta e escalável de segurança para proteger os dispositivos IoT contra ataques cibernéticos. A arquitetura proposta utiliza o Raspberry Pi como o dispositivo central de detecção de intrusão, que pode ser implantado em uma variedade de ambientes IoT. Os dispositivos Raspberry Pi podem ser usados individualmente ou em conjunto para realizar a detecção de intrusão colaborativa.

A proposta de arquitetura visa fornecer segurança e privacidade em ambientes IoT de forma portável, de fácil configuração e uso versátil. O Raspberry Pi é escolhido como o dispositivo central devido à sua portabilidade, facilidade de configuração e capacidade de executar o Snort. A arquitetura permite a detecção de ataques em tempo real, notificando os usuários ou administradores da rede quando atividades suspeitas são detectadas.

Além disso, os dispositivos Raspberry Pi podem ser usados de forma colaborativa, trocando informações entre si para melhorar a detecção de ataques e reduzir falsos positivos. Os dados coletados pelos dispositivos podem ser enviados para um servidor remoto executando software de gerenciamento de informações e eventos de segurança *Security Information and*

Event Management (SIEM), permitindo que os administradores da rede realizem operações de manutenção ou emergência.

Em resumo, o texto propõe uma arquitetura de detecção de intrusão baseada no Raspberry Pi e no Snort para ambientes IoT, com ênfase na portabilidade, facilidade de uso e detecção colaborativa de ataques. Os resultados dos experimentos mostraram que o Raspberry Pi é capaz de lidar com as operações do Snort de forma eficaz, abrindo possibilidades para sua aplicação como um sistema de detecção de intrusão em dispositivos IoT.

É o primeiro a estabelecer uma conexão com o IoT e o Snort, sendo relevante para o presente estudo, cujo o objetivo é utilizar o LoRa para monitorar redes IoT, com foco em redes LoRa.

3.5 Arquitetura de rede com o IDS Snort em redes LoRa

Oniga *et al.* (2017) abordam preocupações de segurança relacionadas à proteção de dados e privacidade em redes de sensores que utilizam a tecnologia LoRaWan no contexto da IoT. O artigo realiza uma análise aprofundada dos aspectos de segurança em redes de sensores LoRaWan e propõe uma nova arquitetura de rede segura. A arquitetura proposta garante a transmissão protegida de dados e previne acesso não autorizado e perda de dados.

A arquitetura básica de redes LoRaWan segue topologias em estrela, em que os nós finais se comunicam com os *gateways*, que por sua vez se comunicam com o Servidor de Rede. O Servidor de Rede é responsável por fornecer comandos *Media Access Control* (MAC) e controle de rede, enquanto o Servidor de Aplicação gerencia as chaves dos nós finais e os *payloads* enviados ou recebidos por eles.

O LoRaWan oferece proteção contra ataques de falsificação e ataques de reprodução, usando contadores de quadros para verificar a ordem e a autenticidade das mensagens. Além disso, a gestão de sessões e chaves é importante para garantir a segurança dos nós finais. Recomendações de segurança incluem monitoramento de comportamento suspeito dos nós finais com base em mensagens rejeitadas, substituição de sessões antigas por novas para evitar falhas de memória e implementar um tempo de expiração para remover sessões inativas.

Em redes LoRaWan, existem vários pontos sensíveis que exigem controles de segurança adicionais, como um IDS. Um ponto sensível da rede é a conexão dos *gateways* com a rede interna. Uma boa prática é implementar uma técnica de detecção de intrusão baseada em monitoramento de tráfego de rede, na qual o sistema gera alertas quando o tráfego detectado é suspeito.

Para a implementação da arquitetura proposta do artigo, são necessários os seguintes componentes:

1. Rede de sensores LoRaWan: O ambiente de teste consiste em um LoRaWan implementado usando a solução *open source* LoRa Server para o servidor *back-end*. Os

gateways são implementados utilizando o *open source* LoRa-net fornecido pela LoRa Alliance e os nós finais são implementados utilizando projetos *open source* do Github de hallard[1] e de jeroennijhof[2].

2. VPN: O software utilizado para implementar a VPN (FERGUSON; HUSTON, 1998) é a OpenVPN (FEILNER, 2006) e serve para fornecer transmissão segura de dados e reduzir a potencial exposição a ataques originários do *gateway*. Cada *gateway* possui um certificado digital gerado pela Autoridade Certificadora e assinado com o certificado do servidor.

3. IDS: O software utilizado na implementação do IDS é o Snort, um IDS de código aberto usado para realizar análise de tráfego em tempo real de pacotes e *logs* em redes IP.

4. Controle de tráfego de rede: Cada entidade implementa regras do *Iptables* (PURDY, 2004) que permitem apenas o tráfego de rede útil para dispositivos LoRaWan e para manutenção do sistema, negando qualquer outra conexão desnecessária.

5. PKI: A arquitetura proposta implementa uma infraestrutura de chave pública (PKI) usando solução de software de código aberto fornecida pela Cloudflare. Cada entidade da arquitetura possui um certificado assinado pela Autoridade Certificadora e o utiliza para iniciar comunicações seguras com outras entidades.

Em resumo, o artigo fornece uma análise abrangente dos aspectos de segurança em redes de sensores LoRaWan e propõe uma arquitetura segura para garantir a proteção de dados e a privacidade em aplicações IoT construídas com o protocolo LoRaWan. Portanto, esse trabalho relacionado foi o mais completo contendo o monitoramento utilizando o Snort em redes LoRa, porém utiliza outras técnicas de segurança, não focando apenas no monitoramento, diferente deste presente trabalho.

3.6 HIDS utilizado em redes LoRa

Bouazzati *et al.* (2023) abordam o uso de contadores de desempenho de hardware em um sistema de detecção de intrusão baseado em *host* (HIDS) para dispositivos IoT que utilizam protocolos de baixa taxa de dados como LoRa. O HIDS é implementado em hardware e monitora os contadores disponíveis no processador de conectividade sem fio do dispositivo IoT para detectar ataques remotos em tempo real. O artigo demonstra a eficácia do sistema ao detectar *exploits* de injeção de pacotes. Além disso, o artigo discute as vulnerabilidades e ataques comuns contra dispositivos IoT e apresenta abordagens de detecção de intrusão. Ele descreve os

[1] https://github.com/hallard/arduino-lmic/tree/rpi
[2] https://github.com/jeroennijhof/LoRaWAN

procedimentos para implementar e avaliar o HIDS, incluindo a configuração experimental com simulação e teste de laboratório.

O artigo apresenta a implementação de um HIDS, que monitora os contadores disponíveis no processador de conectividade sem fio do dispositivo IoT. Essa abordagem visa detectar ataques remotos em tempo real. Portanto, o trabalho relacionado é útil, pois é criado um HIDS para assegurar o LoRa, e este trabalho tem como objetivo utilizar o OSSEC, um HIDS, para monitorar um *gateway* LoRa.

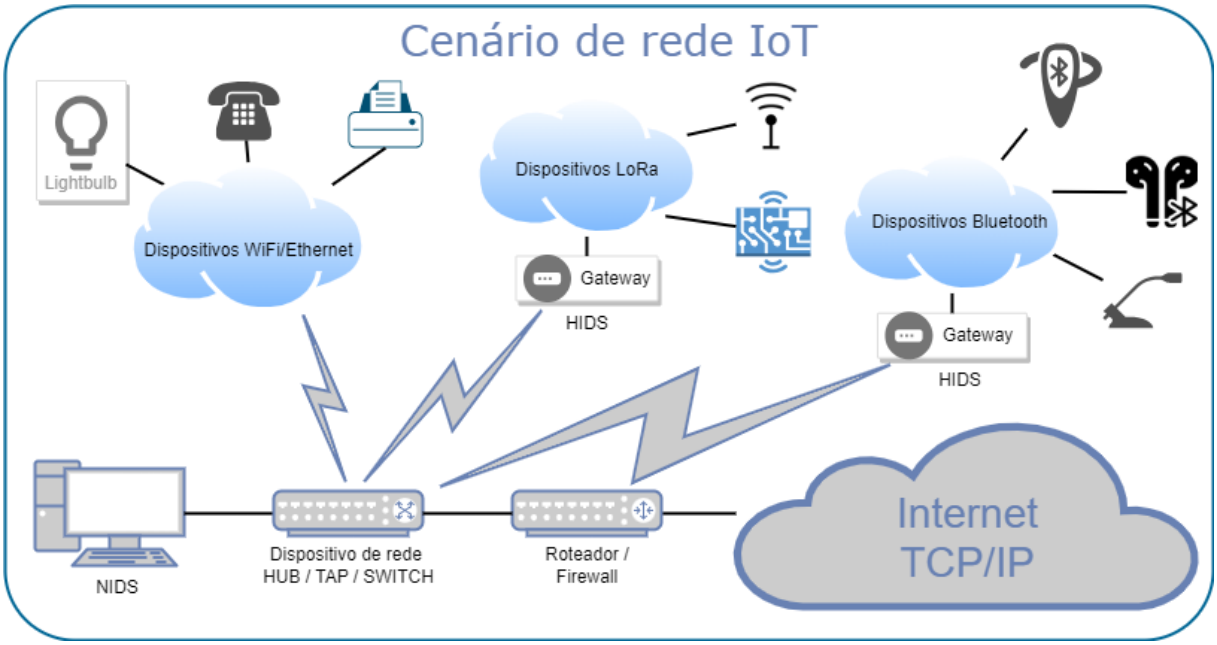
No Capítulo a seguir são apresentados os materiais e método de pesquisa.

4 METODOLOGIA

Objetivou-se nesta pesquisa implementar um cenário de rede com monitoramento do *gateway* LoRa por meio de dois IDS em ambientes IoT, visando manter a segurança dos ambientes, como também verificar a efetividade dos NIDS e HIDS para o monitoramento de redes e dispositivos IoT.

A Figura 9 ilustra a organização generalizada do cenário proposto para o desenvolvimento do presente trabalho. No cenário ilustrado na Figura 9, há um cenário que pode ser composto de: NIDS, HIDS, dispositivo concentrador de rede como *hub*, *switch* ou TAP, e dispositivos IoT como dispositivos Wi-Fi/Ethernet, LoRa e Bluetooth. O NIDS monitorará o tráfego de rede recebido do concentrador de rede, enquanto o HIDS monitorará o *host*. Note, que no cenário apresentado na Figura 9, há os tipos de tecnologia, mais comuns em redes IoT (Bluetooth, WiFi, Ethernet e LoRa), já que uma solução com IDS, pode monitorar a princípio essas redes. Todavia neste trabalho foi monitorado apenas a rede LoRa, do lado do *gateway*.

Figura 9 – Cenário de rede



Fonte: Autoria própia.

Com base nos trabalhos de Edwards (2002) e de Kizza (2013), verificou-se que o NIDS precisa receber o tráfego de rede, seja espelhando os pacotes ou copiando-os. Os dispositivos de rede capazes disso, abordados ao longo o texto, são: *hub*, *switch* e *network* TAP. Outro ponto importante foi o posicionamento do NIDS. As posições possíveis, conforme mostrado na Figura 4, são:

a) Dentro do *firewall*.

b) Entre o *firewall* e a Internet. Fora do *firewall*.

c) Dentro da rede em que estará os dispositivos a serem monitorados. Em um roteador conectado à DMZ.

Dentre essas opções, as duas mais interessantes são:

1. Entre o *firewall* e a Internet.
2. Fora do *firewall*.

No primeiro ponto, o NIDS pode visualizar todo o tráfego da Internet à medida que ele entra na rede. Já no segundo ponto, o NIDS lida com os tráfegos maliciosos que conseguem passar pelo *firewall*. No segundo caso, o NIDS fica menos sobrecarregado de pacotes para analisar, podendo ser uma vantagem para o NIDS.

A partir do ponto em que o tráfego chega com sucesso ao analisador do NIDS, cabe ao analisador determinar o nível de ameaça com base na natureza e ameaça do tráfego suspeito. O tráfego é então classificado como seguro ou suspeito. O analisador indica a gravidade da ameaça, bem como o escopo, a intenção e a frequência da ameaça. Depois, o incidente é notificado, através de *logs*, ao administrador de rede responsável pelo tratamento.

Além de utilizar o NIDS no cenário de rede, também é possível utilizar o HIDS, que se baseia no monitoramento de *host*. O uso do NIDS e HIDS em conjunto proporciona mais segurança devido à ampla cobertura de monitoramento.

Dada as possibilidades apresentadas anteriormente, as utilizadas no cenário real de testes são: *switch*, o NIDS, que foi alocado dentro da rede em que esta o *gateway* LoRa a ser monitorado e o HIDS.

O cenário proposto neste trabalho foi implementado e testado na rede da Universidade Tecnológica Federal do Paraná (UTFPR) em Campo Mourão, que possui uma rede LoRaWan com *gateway* e servidor. Essa rede foi utilizada para verificar, na prática, a efetividade do monitoramento de IoT através do uso de IDS.

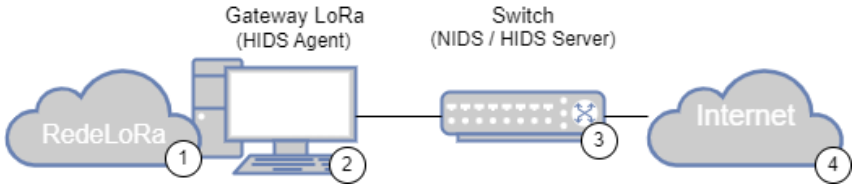
A Figura 10 mostra como está implementado esse cenário de rede na UTFPR. O cenário na prática é composto de um *gateway* LoRa (2) conectado a uma máquina com o NIDS (3), mais especificamente com o Snort (3) dentro da rede e com o OSSEC server (3), que monitora o *gateway* LoRa (2), o qual tem instalado o OSSEC agent (2). Essa máquina foi configurada com duas placas de rede, sendo elas do NIDS e do HIDS Server, elas funcionam como um *switch* (3) entre o dispositivo de rede e o *gateway* LoRa encaminhando os pacotes para o NIDS.

Para a realização do cenário foram avaliadas algumas tecnologias, as quais são apresentadas na seção a seguir.

4.1 Materiais e Métodos

Para o NIDS, foi escolhido o Snort, pois esse pode ser utilizado como *sniffer* de pacotes e *logger*. O Linux no modo *bridge* funciona como um dispositivo concentrador de rede que

Figura 10 – Cenário de rede LoRaWan



Fonte: Autoria própia.

transmite o tráfego de rede para o Snort. O dispositivo concentrador de rede utilizado no cenário são, especificamente, duas placa de rede em um computador com Linux, que funciona como um *switch*.

Já para o HIDS, foi escolhido o Ossec, pois esse pode ser usado para detecção de *rootkits* e *malwares*, respostas a ataques e alterações no sistema em tempo real, por meio de vários mecanismos, incluindo políticas de *firewall*, integração com terceiros, como portais de suporte, bem como ações de autocorreção, auditoria em nível de aplicativo e sistema para conformidade com muitos padrões comuns, monitoramento de integridade de arquivos, e inventário do sistema.

Para a implementação desse cenário de rede, foi instalado o Snort na máquina apresentada na Figura 10. A sequência de passos empregados para a instalação e configuração do Snort, para este cenário, pode ser vista no Anexo A, ele foi desenvolvido para auxiliar e explicar com mais detalhes a instalação do Snort na máquina com o Linux Mint 21.1 Cinnamon Edition.

Ao instalar o Snort, definiu-se a faixa de rede local. Após a instalação, o Snort foi configurado para realizar o monitoramento. Para a configuração, o Snort foi configurado através do arquivo de configuração do Snort, o arquivo *snort.conf* (GUPTA *et al.*, 2017), para estabelecer a rede que o Snort monitorará. Além disso, para a configuração dos *logs*, foi ativado a opção do *fast alert*, fazendo com que os alertas sejam armazenados em um arquivo de *log* com texto simples. Quando o *fast alert* é ativado, o arquivo de *log* armazena informações como o carimbo de data/hora, mensagem de alerta, endereço IP, porta de origem, endereço IP e a porta de destino. Caso o *fast alert* não seja ativado, o Snort armazena os arquivos de *logs* em formato binário, dificultando o entendimento do arquivo de *log*, consequentemente, o monitoramento pelo administrador de rede.

Para evitar que o Snort pare de funcionar em quedas de luz ou quando o computador que hospeda o IDS é desligado, o Snort foi configurado para iniciar o monitoramento do *gateway* LoRa sempre que o computador for inicializado. Com o Snort configurado, é possível analisar o arquivo de texto contendo os alertas gerados pelo Snort, para verificar a eficácia do monitoramento do NIDS em *gateways* LoRa.

Além disso, para o cenário de rede, também foi instalado o OSSEC, que possui tanto o OSSEC *server* quanto o OSSEC *agent*. O *server* atua como servidor, que vai concentrar todo o gerenciamento e correlacionamento de eventos, envio de alertas e resposta ativa quando configurada. Já o agente funciona como um encaminhador de eventos para o Ossec *server*, onde

os eventos serão analisados e correlacionados. Para a instalação e configuração do OSSEC *server* foi desenvolvido o manual de instalação, que pode ser visto no Anexo A, para auxiliar e explicar com mais detalhes a instalação do OSSEC *server*.

Visto o Anexo, foram necessários alguns comandos de instalação e configuração. Para a configuração, adicionou-se o IP do dispositivo que terá o OSSEC *agent* instalado. Além da instalação do OSSEC *server*, foi instalado o OSSEC *agent* na máquina a ser monitorada. No caso do cenário proposto, o OSSEC *agent* foi instalado no *gateway* LoRa. Portanto, com o cenário implementado e em funcionamento, a próxima etapa foi analisar os arquivos de *log* para verificar a eficácia do monitoramento do HIDS em *gateways* LoRa.

O capítulo seguinte apresentará os resultados do monitoramento do Snort e do OSSEC no *gateway* LoRa, permitindo concluir se a abordagem é eficaz ou não.

5 RESULTADOS

Neste capítulo, são apresentados os resultados obtidos do monitoramento de uma rede LoRa disponível na UTFPR. O monitoramento foi feito com os IDS Snort e OSSEC com configurações básicas, com o intuito de verificar se essas configurações conseguem identificar ciberameaças relacionadas com elementos LoRa. Além disso, são discutidos os *logs* obtidos durante o período de monitoramento, analisando as informações registradas e identificando possíveis ameaças ou atividades maliciosas. A análise dos resultados possibilitou avaliar a eficácia das ferramentas utilizadas e a capacidade de detecção de intrusões no contexto específico dos *gateways* LoRa. A partir desses resultados, é realizada uma discussão se é possível verificar se a abordagem adotada é efetiva para o monitoramento e se contribui para maior segurança dos ambientes analisados.

5.1 Logs obtidos do NIDS

No decorrer do período de monitoramento do Snort, entre 20 de maio e 8 de junho de 2023, foram registradas mais de 3500 notificações de *log*, totalizando 3822 registros. Devido à extensão desse conjunto de dados, é inviável discutir cada notificação individualmente. Para uma análise mais concisa, foi criado um repositório no GitHub (disponível em https://github.com/IsabelaGantzel/TCC—LoRa-IDS/blob/main/snort.alert.fast), onde estão disponíveis todos os registros detalhados. No entanto, para o propósito deste estudo, foram selecionados trechos desses *logs* para avaliar a eficácia do monitoramento realizado. A partir desse trecho representativo, foram destacadas as principais ameaças identificadas, permitindo avaliar a capacidade do Snort como ferramenta de detecção de intrusões em IoT, mais especificamente em *gateways* LoRa.

Durante o período de monitoramento do Snort realizado em uma rede de produção, isto é, os alertas não foram criados para testar o monitoramento, foram identificadas nove registros distintos e significativos, abrangendo diferentes tipos de ameaças ou vulnerabilidades. Essas notificações incluem consultas de DNS *spoofing*, tentativas de vazamento de informações e varreduras de rede. Cada uma dessas notificações representa uma potencial violação de segurança detectada pelo Snort. A seguir, é possível visualizar esses registros na Listagem 1.

Os registros fornecem informações sobre as atividades e eventos detectados pelo Snort, incluindo o tipo de tráfego, endereços IP de origem e destino, classificação da ameaça e prioridade. A classificação das ameaças com base na gravidade e prioridade permite que a equipe de segurança possa priorizar ações de resposta e mitigação, garantindo a proteção do ambiente monitorado.

Entre os registros observados, destacam-se consultas de DNS suspeitas, tentativas de acesso SNMP público e privado, além de varreduras de rede utilizando ICMP PING e o *scanner* de portas Nmap com a técnica XMAS. Ao verificar os endereços de IP de origem, notou-se dois

Listagem 1 – Logs obtidos do Snort

```
1: 05/20-00:32:42.828652 [**] [1:254:4] DNS SPOOF query response
   with TTL of 1 min. and no authority [**] [Classification:
   Potentially Bad Traffic] [Priority: 2] UDP 172.16.255.202:53 ->
   192.168.2.83:45632
2: 05/20-21:10:41.551750 [**] [1:1411:10] SNMP public access udp [**]
   [Classification: Attempted Information Leak] [Priority: 2] UDP
   172.16.255.210:37080 -> 192.168.2.83:161
3: 05/20-21:10:41.551750 [**] [1:1417:9] SNMP request udp [**]
   [Classification: Attempted Information Leak] [Priority: 2] UDP
   172.16.255.210:37080 -> 192.168.2.83:161
4: 05/25-14:44:51.212162 [**] [1:469:3] ICMP PING NMAP [**]
   [Classification: Attempted Information Leak] [Priority: 2] ICMP
   172.16.255.162 -> 192.168.2.83
5: 05/25-14:44:51.324350 [**] [1:1421:11] SNMP AgentX/tcp request
   [**] [Classification: Attempted Information Leak] [Priority: 2]
   TCP 172.16.255.162:44129 -> 192.168.2.83:705
6: 05/25-14:44:51.435230 [**] [1:1418:11] SNMP request tcp [**]
   [Classification: Attempted Information Leak] [Priority: 2] TCP
   172.16.255.162:44129 -> 192.168.2.83:161
7: 05/25-14:44:52.510071 [**] [1:1228:7] SCAN nmap XMAS [**]
   [Classification: Attempted Information Leak] [Priority: 2] TCP
   172.16.255.162:43954 -> 192.168.2.83:1
8: 06/01-21:10:46.240240 [**] [1:1413:10] SNMP private access udp
   [**] [Classification: Attempted Information Leak] [Priority: 2]
   UDP 172.16.255.210:59701 -> 192.168.2.83:161
9: 06/01-21:10:46.240240 [**] [1:1417:9] SNMP request udp [**]
   [Classification: Attempted Information Leak] [Priority: 2] UDP
   172.16.255.210:59701 -> 192.168.2.83:161
```

Fonte: Autoria própria (2023).

IPs padrões sendo eles o roteador e o gateway LoRa (172.16.255.202 e 172.16.255.210) que repetem constantemente no arquivo de *log*. Conclui-se, portanto, que é apropriado adicionar uma regra adicional ao Snort para notificar sempre que o *gateway* LoRa (192.168.2.83) for acessado por um IP diferente dos IPs padrões, esses IPs padrões são IPs provenientes da rede.

Com o objetivo de aprimorar a detecção desses acessos, foi implementada duas novas regras, também chamadas de assinaturas no Snort. Essas assinaturas irão gerar um alerta sempre que ocorrer uma comunicação com o *gateway* LoRa em que o IP de origem ou destino seja diferente dos IPs padrões esperados. Foi estudado a opção de criar regras com assinatura para especificar os IP esperados e a exceção gerar o alerta, porém o Snort não suporta esse tipo de regra. A seguir, na Listagem 2 são apresentadas as assinaturas:

Essas novas assinaturas irão gerar alertas toda vez que ocorrer comunicação IP que não seja com os IPs: 172.16.255.202 ou o IP 172.16.255.210. Como pode ser observado na Listagem 3. Assim, garantindo uma notificação imediata sempre que o *gateway* LoRa for acessado por um IP diferente dos IPs esperados. Dessa forma, implementando a nova assinatura

Listagem 2 – Assinaturas implementadas no Snort

```
1: alert ip !172.16.255.202/32 !172.16.255.210/32 -> 192.168.2.83
   any (msg:"Acesso ao gateway LoRa por um IP fora do padrão";
   sid:1000002;)
2: alert ip 192.168.2.83 any -> !172.16.255.202/32 !172.16.255.210/32
   (msg:"Acesso ao gateway LoRa por um IP fora do padrão";
   sid:1000001;)
```

Fonte: Autoria própria (2023).

no Snort, o monitoramento será mais eficaz, permitindo a detecção precoce de potenciais anomalias ou comportamentos indesejados ao acessar o *gateway* LoRa.

Ao observar o *log* obtido após a implementação das novas assinaturas, foi possível notar que o objetivo de gerar alertas toda vez que ocorre uma comunicação IP diferente dos IPs esperados foi atingido, como também foi observado que as novas assinaturas podem ajudar o administrador a monitorar atividades suspeitas, porque se o tráfego não corresponder às condições específicas dessas assinaturas, as notificações serão geradas, ou seja, elimina as notificações dos IPs padrões, consequentemente diminuindo as notificações que o administrador precisa analisar em busca de atividades suspeitas, ou seja, diminuindo os falsos positivos.

Contudo a mensagem padrão "Acesso ao gateway LoRa por um IP fora do padrão" das novas assinaturas, acaba por substituindo as mensagens anteriores, como "Classification: Attempted Information Leak", que detalhavam o *log*. Visto isso, para trabalhos futuros, será estudada a possibilidade da implementação de filtragem de pacotes através do Iptables, como também será avaliada a sua efetividade e se irá bloquear a comunicação com os IPs que serão filtrados. Essa filtragem filtraria os pacotes com IPs diferente dos IPs padrões, consequentemente, o Snort poderia desativar essas novas assinaturas implementadas, voltando a ter mensagens dos *logs* mais detalhadas com as classificações.

Listagem 3 – Log com a nova regra implementada

```
1: 07/15-18:48:15.798098 [**] [1:1000001:0] Acesso ao gateway LoRa
   por um IP fora do padrão [**] [Priority: 0] ICMP 192.168.2.83 ->
   142.250.219.238
2: 07/15-18:48:15.817871 [**] [1:1000002:0] Acesso ao gateway LoRa
   por um IP fora do padrão [**] [Priority: 0] ICMP 142.250.219.238
   -> 192.168.2.83
3: 007/15-18:48:21.983746 [**] [1:1000001:0] Acesso ao gateway LoRa
   por um IP fora do padrão [**] [Priority: 0] UDP 192.168.2.83:53589
   -> 13.238.174.71:1700
4: 07/15-18:48:22.319884 [**] [1:1000002:0] Acesso ao gateway LoRa
   por um IP fora do padrão [**] [Priority: 0] UDP 13.238.174.71:1700
   -> 192.168.2.83:53589
```

Fonte: Autoria própria (2023).

Portanto, as detecções do Snort demonstram eficácia no monitoramento realizado, identificando atividades suspeitas e potencialmente maliciosas. É importante ressaltar a importância de uma análise detalhada desses eventos para entender o contexto em que ocorreram e

determinar as medidas necessárias para prevenir futuras ocorrências, como a nova regra estabelecida. Essas informações são essenciais para fortalecer a segurança dos ambientes IoT e garantir a integridade e confidencialidade dos dados transmitidos. A seguir, é apresentado e discutido os *logs* obtidos do monitoramento do HIDS.

5.2 Logs obtidos do HIDS

O monitoramento do OSSEC foi realizado entre 07 de junho e 12 de junho de 2023, nesse período foram obtidos arquivos de *log*. Para uma análise mais concisa, foi criado um repositório no GitHub (disponível em https://github.com/IsabelaGantzel/TCC—LoRa-IDS/blob/main/ossec-alerts.log), onde estão disponíveis todos os registros detalhados. No entanto, para o propósito deste estudo, foram selecionados trechos significativo desse *log* para avaliar a eficácia do monitoramento realizado. A partir desse trecho representativo, foram destacados os principais registros identificadas, permitindo avaliar a capacidade do OSSEC como ferramenta de detecção de intrusões em IoT, mais especificamente em *gateways* LoRa.

Durante o período de monitoramento do OSSEC, foram identificadas registros significativos, abrangendo diferentes tipos de informações. Esses registros fornecem informações como atividades de autenticação no SSH, abertura e encerramento de sessões de *login*, alterações em arquivos do sistema, problemas desconhecidos no sistema, alterações de *checksum* e negação de acesso pelo AppArmor[1]. Esses registros são relevantes para o monitoramento e a detecção de atividades suspeitas ou intrusões em um sistema. A análise completa dos *logs* pode ajudar a identificar potenciais ameaças ou problemas de segurança no ambiente.

Os registros na Listagem 4 seguem um formato estruturado que fornece informações a respeito dos eventos detectados pelo OSSEC que ocorreram no *gateway* LoRa. Essas informações incluem: identificador único do evento, consistindo em um número sequencial seguido de carimbo de data/hora, o nome do dispositivo ou endereço IP, o caminho do arquivo de *log* onde o evento foi registrado, a regra de detecção acionada pelo evento, fornecendo um número e nível de severidade associado. Por fim, informações mais específicas sobre o evento em questão, como uma descrição detalhada do evento ou da ação executada. A classificação das ameaças com base na severidade permite que a equipe de segurança possa priorizar ações de resposta e mitigação, garantindo a proteção do ambiente monitorado.

Em conclusão, a análise dos registros na Listagem 4 de *log* do OSSEC durante o período de monitoramento do *gateway* LoRa revelou uma série de eventos e alertas relevantes. Através desses registros, foi possível identificar a conexão de novos agentes do OSSEC, eventos de auditoria do sistema, alterações na integridade de arquivos e rotação de arquivos de *log*. Esses resultados fornecem compreensão valiosos sobre a segurança e integridade do sistema monitorado. Além disso, a disponibilização dos registros de *log* no repositório do GitHub ofe-

[1] https://apparmor.net/

rece oportunidade para análises futuras e contribuições para a comunidade acadêmica e de segurança cibernética.

Com base nos resultados apresentados, acredita-se que a implementação de um IDS, utilizando o Snort e o OSSEC, se mostrou eficiente no monitoramento e detecção de potenciais ameaças e violações de segurança no *gateway* LoRa. Inclusive, mesmo utilizando as configurações padrões do Snort e do OSSEC, o monitoramento do LoRa é efetivo. Isto é, foi efetivo no monitoramento dos pacotes que saem da rede LoRa e vão do *gateway* LoRa até o *switch*. A seguir, é apresentado o capítulo final, abordando a conclusão do trabalho.

Listagem 4 – Trecho do log do OSSEC

```
1: ** Alert 1686307969.3232: mail - ossec, 2023 Jun 09 07:52:49
   (gwLora) 192.168.2.83->ossec Rule: 501 (level 3) -> 'New ossec
   agent connected.' ossec: Agent started: 'gwLora->192.168.2.83'.
2: ** Alert 1686308167.4384: - pam,syslog,authentication_failed, 2023
   Jun 09 07:56:07 (gwLora) 192.168.2.83->/var/log/auth.log Rule:
   5503 (level 5) -> 'User login failed.' Src IP: 172.16.255.198
   User: luiz Jun 9 07:56:06 utfpr-cm sshd[28918]: pam_-
   unix(sshd:auth): authentication failure; logname= uid=0 euid=0
   tty=ssh ruser= rhost=172.16.255.198 user=luiz
3: ** Alert 1686308169.4740: - syslog,sshd,authentication_failed,
   2023 Jun 09 07:56:09 (gwLora) 192.168.2.83->/var/log/auth.log
   Rule: 5716 (level 5) -> 'SSHD authentication failed.' Src IP:
   172.16.255.198 User: luiz Jun 9 07:56:08 utfpr-cm sshd[28918]:
   Failed password for luiz from 172.16.255.198 port 49532 ssh2
4: ** Alert 1686308183.5679: mail - syslog,access_-
   control,authentication_failed, 2023 Jun 09 07:56:23 (gwLora)
   192.168.2.83->/var/log/auth.log Rule: 2502 (level 10) -> 'User
   missed the password more than one time' Src IP: 172.16.255.198
   User: luiz Jun 9 07:56:22 utfpr-cm sshd[28918]: PAM 2 more
   authentication failures; logname= uid=0 euid=0 tty=ssh ruser=
   rhost=172.16.255.198 user=luiz
5: ** Alert 1686308191.6069: mail - syslog,fts,authentication_success
   2023 Jun 09 07:56:31 (gwLora) 192.168.2.83->/var/log/auth.log
   Rule: 10100 (level 4) -> 'First time user logged in.' Src IP:
   172.16.255.198 User: luiz Jun 9 07:56:31 utfpr-cm sshd[28932]:
   Accepted password for luiz from 172.16.255.198 port 52572 ssh2
6: ** Alert 1686354568.55162: mail - syslog,errors, 2023 Jun 09
   20:49:28 (gwLora) 192.168.2.83->/var/log/syslog Rule: 1002 (level
   2) -> 'Unknown problem somewhere in the system.' Jun 9 20:49:27
   utfpr-cm ttn-gateway[6272]: INFO: packets received with a CRC
   error will NOT be forwarded
7: ** Alert 1686387985.400: mail - ossec,syscheck, 2023 Jun
   10 06:06:25 (gwLora) 192.168.2.83->syscheck Rule: 550
   (level 7) -> 'Integrity checksum changed.' Integrity
   checksum changed for: '/etc/fake-hwclock.data' Old md5sum
   was: 'd08679532f4c48a1b5dd51c5826ab795' New md5sum is
   : 'cb8d21f7d1b03c3f02396964aedd9183' Old sha1sum was:
   'ad82445e1a29bdc781b147e4faf4273b3d70032b' New sha1sum is :
   'c4a301fcb0da207d3060dfc0103d05e9307de0be'
8: ** Alert 1686389156.2701: - ossec, 2023 Jun 10 06:25:56 (gwLora)
   192.168.2.83->ossec-logcollector Rule: 591 (level 3) -> 'Log file
   rotated.' ossec: File rotated (inode changed): '/var/log/syslog'.
```

Fonte: Autoria própria (2023).

6 CONCLUSÃO

Nesta monografia, foi abordada a importância de manter a segurança dos dispositivos IoT, considerando suas limitações e vulnerabilidades. Com o aumento significativo da utilização desses dispositivos, torna-se imprescindível adotar medidas para garantir a proteção contra ataques maliciosos e mitigar os possíveis impactos dessas ameaças. Além disso, este trabalho tem como objetivo estudar cibersegurança em redes e dispositivos IoT, e além de pesquisar e propor mecanismos de segurança cibernética formando um cenário de rede que envolve o monitoramento por meio de NIDS e HIDS, com o intuito de obter maior segurança contra ataques maliciosos em redes IoT. Assim como responder as perguntas:

1. Qual o estado da arte em relação ao monitoramento de redes e dispositivos IoT?
2. Qual é a efetividade do IDS em redes IoT?
3. É possível propor melhorias no monitoramento de redes e dispositivos IoT?

Em relação ao estado da arte do monitoramento de redes em dispositivos IoT foi observado nos trabalhos relacionados, que o monitoramento de redes em dispositivos IoT é utilizado como opção de mecanismo de segurança para manter as redes IoT seguras, e além disso, demonstrou efetividade. Contudo existem poucos estudos realizados na área, especialmente quando se trata de LoRaWan.

Visto isso, foi proposto fazer um cenário de rede que busca assegurar os dispositivos IoT por meio de diversas técnicas de segurança. Essas técnicas incluem o monitoramento do tráfego de rede por meio de Snort, um NIDS, o monitoramento do *host* por meio do OSSEC, um HIDS e o uso de dispositivos concentradores de rede.

Além disso, realizou-se uma pesquisa dos ataques comumente direcionados aos dispositivos IoT e LoRa, identificando suas características e impactos. Com base nessa análise, foram propostas recomendações de segurança para auxiliar na proteção efetiva desses dispositivos.

O cenário de rede proposto foi implementado e testado na rede da UTFPR em Campo Mourão, pois a infraestrutura da rede LoRaWan, *gateway* e servidor já estavam implementadas. Assim, permitindo a obtenção de resultados por meio dos *logs* gerados pelos NIDS e HIDS. Inicialmente, o Snort e o OSSEC foram utilizados com configurações padrões, sem adições de assinaturas, mesmo em suas configurações padrões, o monitoramento foi promissor.

Os registros de *logs* obtidos dos pacotes do *gateway* LoRa destacaram atividades que podem representar riscos à segurança do ambiente IoT, ou seja, foram significativos para identificar ameaças em potencial e para aprimorar a segurança da rede. Em resposta a essas detecções, é recomendado tomar ações como investigação detalhada, contenção imediata, remediação, análise forense e aprimoramento da segurança. Cada caso requer uma análise cuidadosa para determinar a extensão do risco e as medidas adequadas a serem tomadas. Visto isso, respondendo a pergunta sobre a efetividade do IDS em redes IoT, é possível dizer que os IDS

conseguem detectar as anomalias nas redes IoT e nos *hosts* com esses registros de *logs*, ou seja, é efetivo.

Como trabalhos futuros, respondendo se é possível propor melhorias no monitoramento de redes e dispositivos IoT, sugere-se a utilização do Iptables em conjunto com os IDS para filtrar pacotes e aliviar os *logs* com os IPs padrões, que repetem constantemente no tráfego de rede. Além disso, para uma melhor avaliação da efetividade dos IDS em relação com a rede LoRa monitorada, serão implementados dispositivos LoRa na rede, para monitorar não só o *gateway*, mas também os dispositivos LoRa da rede. Visto isso, serão realizados também simulações de ataques específicos direcionados ao protocolo LoRa, visando aprofundar a compreensão dos possíveis cenários de ameaças e desenvolver estratégias de defesa mais robustas com o Snort e OSSEC.

Em suma, a proteção dos dispositivos IoT é uma preocupação crescente e demanda a aplicação de abordagens abrangentes de segurança. O cenário de rede proposto e as recomendações apresentadas neste trabalho visam contribuir para a manutenção da segurança dos dispositivos *IoT*, permitindo que eles continuem a desempenhar um papel fundamental em nosso mundo cada vez mais conectado.

REFERÊNCIAS

ARAS, E. *et al.* Exploring the security vulnerabilities of lora. *In*: **2017 3rd IEEE International Conference on Cybernetics (CYBCONF)**. [*S.l.*: *s.n.*], 2017. p. 1–6.

BOUAZZATI, M. E. *et al.* A lightweight intrusion detection system against iot memory corruption attacks. *In*: **2023 26th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)**. [*S.l.*: *s.n.*], 2023. p. 118–123.

BRAY, R.; CID, D.; HAY, A. **OSSEC host-based intrusion detection guide**. [*S.l.*]: Syngress, 2008.

DANISH, S. M. *et al.* Network intrusion detection system for jamming attack in lorawan join procedure. *In*: **2018 IEEE International Conference on Communications (ICC)**. [*S.l.*: *s.n.*], 2018. p. 1–6.

DART, E. *et al.* The science dmz: A network design pattern for data-intensive science. *In*: . New York, NY, USA: Association for Computing Machinery, 2013. (SC '13). ISBN 9781450323789. Disponível em: https://doi.org/10.1145/2503210.2503245.

EDWARDS, S. Network intrusion detection systems: Important ids network security vulnerabilities. **White Paper Top Layer Networks, Inc.**, 2002.

ELSAEIDY, A. *et al.* A smart city cyber security platform for narrowband networks. *In*: **2017 27th International Telecommunication Networks and Applications Conference (ITNAC)**. [*S.l.*: *s.n.*], 2017. p. 1–6.

EVANS, D. A internet das coisas: como a próxima evolução da internet está mudando tudo. **CISCO IBSG**, 2011.

FEILNER, M. **OpenVPN: Building and integrating virtual private networks**. [*S.l.*]: Packt Publishing Ltd, 2006.

FERGUSON, P.; HUSTON, G. What is a vpn? Revision, 1998.

GAO, H. *et al.* An improved two-dimensional variational mode decomposition algorithm and its application in oil pipeline image. **Systems Science & Control Engineering**, Taylor & Francis, v. 8, n. 1, p. 297–307, 2020.

GEER, D. Malicious bots threaten network security. **Computer**, v. 38, n. 1, p. 18–20, 2005.

GOODRICH, M. T.; TAMASSIA, R. **Introdução à segurança de computadores**. [*S.l.*]: Bookman, 2013.

GUPTA, R. *et al.* Intrusion detection system using snort. **International Research Journal of Engineering and Technology (IRJET)**, v. 4, n. 04, p. 2100–2104, 2017.

HASSIJA, V. *et al.* A survey on iot security: Application areas, security threats, and solution architectures. **IEEE Access**, v. 7, p. 82721–82743, 2019.

KIZZA, J. M. **Guide to computer network security**. [*S.l.*]: Springer, 2013. ISBN 9783319556055.

LEMOS, A. Cidades inteligentes. **GV-executivo**, v. 12, n. 2, p. 46–49, 2013.

LINKS, C. **IoT Standards: The End Game**. 2019. Qorvo Blog team. Disponível em: https://www.qorvo.com/design-hub/blog/iot-standards-the-end-game. Acesso em: 13 jun. 2022.

LIT, Y.; KIM, S.; SY, E. A survey on amazon alexa attack surfaces. *In*: **2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)**. [*S.l.*: *s.n.*], 2021. p. 1–7.

MOURA, G. C. *et al.* Anycast vs. ddos: Evaluating the november 2015 root dns event. *In*: **Proceedings of the 2016 Internet Measurement Conference**. New York, NY, USA: Association for Computing Machinery, 2016. (IMC '16), p. 255–270. ISBN 9781450345262. Disponível em: https://doi.org/10.1145/2987443.2987446.

NIST. **Guide to Integrating Forensic Techniques into Incident Response**. [*S.l.*], 2006.

NIST. **Guidelines on Firewalls and Firewall Policy**. [*S.l.*], 2009.

NIST. **Guidelines on Mobile Device Forensics**. [*S.l.*], 2014.

NIST. **Guide to Industrial Control Systems (ICS) Security**. [*S.l.*], 2015.

NIST. **Supplemental Information for the Interagency Report on Strategic U.S. Government Engagement in International Standardization to Achieve U.S. Objectives for Cybersecurity**. [*S.l.*], 2015.

NIST. **Interagency Report on the Status of International Cybersecurity Standardization for the Internet of Things (IoT)**. [*S.l.*], 2018.

NIST. **Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations**. [*S.l.*], 2020.

NIST. **Developing Cyber-Resilient Systems: A Systems Security Engineering Approach**. [*S.l.*], 2021.

NIST. **Managing the Security of Information Exchanges**. [*S.l.*], 2021.

NIST. **Protecting Information and System Integrity in Industrial Control System Environments: Cybersecurity for the Manufacturing Sector**. [*S.l.*], 2022.

NIST. **Protecting Information and System Integrity in Industrial Control System Environments: Cybersecurity for the Manufacturing Sector**. [*S.l.*], 2022.

NOBRE, J. *et al.* Segurança da informação para internet das coisas (iot): uma abordagem sobre a lei geral de proteção de dados (lgpd). **Revista Eletrônica de Iniciação Científica em Computação**, v. 17, 11 2019.

NOROUZI, M.; FLEET, D. J.; SALAKHUTDINOV, R. R. Hamming distance metric learning. **Advances in neural information processing systems**, v. 25, 2012.

NOURA, H. *et al.* Lorawan security survey: Issues, threats and possible mitigation techniques. **Internet of Things**, v. 12, p. 100303, 2020. ISSN 2542-6605. Disponível em: https://www.sciencedirect.com/science/article/pii/S2542660520301359.

ONIGA, B. *et al.* Analysis, design and implementation of secure lorawan sensor networks. *In*: **2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)**. [*S.l.*: *s.n.*], 2017. p. 421–428.

PERAKOVIC, D.; PERIšA, M.; CVITIć, I. Analysis of the iot impact on volume of ddos attacks. *In*: **33rd Symposium on New Technologies in Postal and Telecommunication Traffic (PosTel 2015)**. [*S.l.*: *s.n.*], 2015. p. 295—-304.

PURDY, G. N. **Linux iptables Pocket Reference: Firewalls, NAT & Accounting**. [*S.l.*]: "O'Reilly Media, Inc.", 2004.

RICHARDSON, M.; WALLACE, S. **Getting started with raspberry PI**. [*S.l.*]: "O'Reilly Media, Inc.", 2012.

ROESCH, M. Snort - lightweight intrusion detection for networks. *In*: **Proceedings of the 13th USENIX Conference on System Administration**. USA: USENIX Association, 1999. (LISA '99), p. 229–238.

SANTOS, B. P. *et al.* Indústria 4.0: desafios e oportunidades. **Revista Produção e Desenvolvimento**, v. 4, n. 1, p. 111–124, 2018.

SCOTT; SPANIEL. Rise of the machines: The dyn attack was just a practice run december 2016. **Institute for Critical Infrastructure Technology, Washington, DC, USA**, 2016.

SEMTECH, A. Lora modulation basics. **Semtech Corporation, Tech. Rep.**, 2015.

SFORZIN, A. *et al.* Rpids: Raspberry pi ids — a fruitful intrusion detection system for iot. *In*: **2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)**. [*S.l.*: *s.n.*], 2016. p. 440–448.

SIMPSON, A. K.; ROESNER, F.; KOHNO, T. Securing vulnerable home iot devices with an in-hub security manager. *In*: **2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)**. [*S.l.*: *s.n.*], 2017. p. 551–556.

SULLIVAN, F. . **The Top Growth Opportunities for IoT in 2023**. 2023. Frost & Sullivan. Disponível em: https://www.frost.com/frost-perspectives/the-top-growth-opportunities-for-iot-in-2023/. Acesso em: 06 set. 2023.

TANENBAUM, A. S. **Redes de Computadores**. trad. 4 ed. Rio de Janeiro: Elsevier, 2003.

TORRES, G. **Redes de computadores**. [*S.l.*]: Novaterra Editora e Distribuidora LTDA, 2015.

VUKALOVIć, J.; DELIJA, D. Advanced persistent threats - detection and defense. *In*: **2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)**. [*S.l.*: *s.n.*], 2015. p. 1324–1330.

ZEIDANLOO, H. R. *et al.* All about malwares (malicious codes). *In*: **Security and Management**. [*S.l.*: *s.n.*], 2010. p. 342–348.

ANEXO A – Manual de instalação do Snort e OSSEC

Manual de Instalação do Snort e do OSSEC

Este manual fornecerá um passo a passo para a instalação e configuração do Snort e do OSSEC no Linux Mint 21.1 Cinnamon Edition. Siga as instruções abaixo:

Instalação do Snort:

1. Instale e inicie o Linux Mint 21.1 Cinnamon Edition em sua máquina. Para o download foi necessário entrar no site https://www.linuxmint.com/edition.php?id=302 e baixar da localização Brazil, em seguida colocar a iso em um pendrive que realiza boot para realizar a instalação do Mint na máquina.
2. Abra o terminal.
3. Execute o seguinte comando para instalar o Snort: sudo apt-get install snort.
4. Durante a instalação, uma janela de configuração do Snort será aberta. Insira o endereço IP da sua rede, por exemplo "192.168.2.0/24". Isso define a rede que o Snort irá proteger.
5. Para confirmar se a instalação foi bem-sucedida, digite o seguinte comando no terminal: snort –version.

Configuração do Snort:

1. Abra o arquivo de configuração do Snort, chamado "snort.conf". No terminal, digite: sudo nano /etc/snort/snort.conf.
2. Localize a linha que contém "ipvar HOME_NET" e defina o valor como "192.168.2.0/24", ficando assim ‘ipvar HOME_NET 192.168.2.0/24’. Isso define a rede que o Snort irá proteger. Salve o arquivo após fazer essa alteração.
3. Adicione a seguinte linha no arquivo "snort.conf" para ativar o fast alert: output alert_fast: snort.alert.fast. O fast alert envia alertas para um arquivo de texto simples. Caso contrário, o snort faz apenas logs em formato binário, dificultando o monitoramento pelo administrador de rede. Salve o arquivo após adicionar essa linha.
4. Para baixar as regras padrão do Snort, digite o seguinte comando no terminal: sudo wget https://www.snort.org/rules/snortrules-snapshot-2983.tar.gz?oinkcode=<your-oink-code-goes-here> -O snortrules-snapshot-2983.tar.gz
5. Agora, extraia as regras baixadas para o diretório de regras do Snort. Digite o seguinte comando no terminal: sudo tar -xvzf snortrules-snapshot-2983.tar.gz -C /etc/snort/rules.
6. Por fim, foi configurado no init.d para o snort iniciar toda vez que a máquina for iniciada.

Pré-instalação do OSSEC:

1. Abra o terminal.
2. Execute os seguintes comandos para instalar os requisitos:

   sudo apt-get update && \

   sudo apt-get -y upgrade && \

   sudo apt-get install -y build-essential && \

sudo apt-get install -y zlib1g-dev libpcre2-dev libevent-dev libssl-dev libsystemd-dev jq

Importação do certificado e chave:

1. No terminal, execute os seguintes comandos para importar o certificado correspondente e o arquivo de chave (.asc):

   wget http://www.ossec.net/files/OSSEC-ARCHIVE-KEY.asc && \
   wget https://github.com/ossec/ossec-hids/releases/download/3.7.0/ossec-hids-3.7.0.tar.gz.asc && \

   gpg --import OSSEC-ARCHIVE-KEY.asc

Download e verificação do OSSEC:

1. No terminal, execute os seguintes comandos para baixar e verificar o pacote do OSSEC:

   wget https://github.com/ossec/ossec-hids/archive/3.7.0.tar.gz && \

   gpg --verify ossec-hids-3.7.0.tar.gz.asc 3.7.0.tar.gz

2. Extraia e instale executando o comando:

   tar -zxvf 3.7.0.tar.gz && cd ossec-hids-3.7.0/ && \

   wget https://github.com/PCRE2Project/pcre2/releases/download/pcre2-10.40/pcre2-10.40.tar.gz && \

   tar -zxvf pcre2-10.40.tar.gz -C src/external/ && \

   sudo PCRE2_SYSTEM=yes ./install.sh

Configuração do OSSEC:

1. No terminal, execute o seguinte comando para abrir o arquivo de configuração do OSSEC: sudo nano /var/ossec/etc/ossec.conf
2. Adicione a seguinte linha no arquivo "ossec.conf" para permitir o IP do agente: <allow_list>192.168.2.83</allow_list>

   Certifique-se de substituir "192.168.2.83" pelo IP do seu agente.

3. Adicione a seguinte linha para permitir o syslog remoto do agente: <allowed-ips>192.168.2.83</allowed-ips>

   Novamente, substitua "192.168.2.83" pelo IP do seu agente.

4. Salve o arquivo após fazer essas alterações.
5. Reinicie o OSSEC, executando o seguinte comando: sudo /var/ossec/bin/ossec-control restart

Após seguir essas etapas, você terá o Mint, o Snort e o OSSEC instalados e configurados. Certifique-se de revisar e ajustar as configurações de acordo com suas necessidades específicas de segurança.