Politécnica UFRJ

DESENVOLVIMENTO DE UMA FERRAMENTA PARA MONITORAMENTO DE TRÁFEGO EM REDES INDUSTRIAIS

Guilherme Lucieri Alonso Costa

Projeto de Graduação apresentado ao Curso de Engenharia Eletrônica e de Computação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientadores: Miguel Elias Mitre Campista
Antônio José Silvério

Rio de Janeiro
Agosto de 2025

DESENVOLVIMENTO DE UMA FERRAMENTA PARA MONITORAMENTO DE TRÁFEGO EM REDES INDUSTRIAIS

Guilherme Lucieri Alonso Costa

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO CURSO DE ENGENHARIA ELETRÔNICA E DE COMPUTAÇÃO DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO ELETRÔNICO E DE COMPUTAÇÃO.

Examinado por:

Prof. Miguel Elias M. Campista, D.Sc.

Prof. Antônio José Silvério, D.Sc.

Prof. Luís Henrique M. K. Costa, Dr.

Prof. Pedro Henrique Cruz Caminha, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
AGOSTO DE 2025

Declaração de Autoria e de Direitos

Eu, *Guilherme Lucieri Alonso Costa* CPF *157.368.677-88*, autor da monografia *Desenvolvimento de uma ferramenta para monitoramento de tráfego em redes industriais*, subscrevo para os devidos fins, as seguintes informações:

1. O autor declara que o trabalho apresentado na disciplina de Projeto de Graduação da Escola Politécnica da UFRJ é de sua autoria, sendo original em forma e conteúdo.
2. Executam-se do item 1. eventuais transcrições de texto, figuras, tabelas, conceitos e ideias, que identifiquem claramente a fonte original, explicitando as autorizações obtidas dos respectivos proprietários, quando necessárias.
3. O autor permite que a UFRJ, por um prazo indeterminado, esteja em qualquer mídia de divulgação, a publicação do trabalho acadêmico em sua totalidade, ou em parte. Essa autorização não envolve ônus de qualquer natureza à UFRJ, ou aos seus representantes.
4. O autor pode, excepcionalmente, encaminhar à Comissão de Projeto de Graduação, a não divulgação do material, por um prazo máximo de 01 (um) ano, improrrogável, a contar da data de defesa, desde que o pedido seja justificado, e solicitado antecipadamente, por escrito, à Congregação da Escola Politécnica.
5. O autor declara, ainda, ter a capacidade jurídica para a prática do presente ato, assim como ter conhecimento do teor da presente Declaração, estando ciente das sanções e punições legais, no que tange à cópia parcial, ou total, de obra intelectual, o que se configura como violação do direito autoral previsto no Código Penal Brasileiro no art.184 et art.299, bem como na Lei 9.610.
6. O autor é o único responsável pelo conteúdo apresentado nos trabalhos acadêmicos publicados, não cabendo à UFRJ, aos seus representantes, ou ao(s) orientador(es), qualquer responsabilização/ indenização nesse sentido.
7. Por ser verdade, firmo a presente declaração.

Guilherme L. A. Costa

Guilherme Lucieri Alonso Costa

Lucieri Alonso Costa, Guilherme

Desenvolvimento de uma ferramenta para monitoramento de tráfego em redes industriais/Guilherme Lucieri Alonso Costa. – Rio de Janeiro: UFRJ/Escola Politécnica, 2025.

XIII, 44 p.: il.; 29, 7cm.

Orientadores: Miguel Elias Mitre Campista
Antônio José Silvério

Projeto de Graduação – UFRJ/Escola Politécnica/Curso de Engenharia Eletrônica e de Computação, 2025.

Referências Bibliográficas: p. 39 – 44.

1. Segurança da Comunicação. 2. Inspeção Profunda de Pacotes. 3. Redes Industriais. I. Elias Mitre Campista, Miguel *et al.* II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia Eletrônica e de Computação. III. Desenvolvimento de uma ferramenta para monitoramento de tráfego em redes industriais.

Aos meus familiares e amigos,
por todo o apoio e incentivo ao
longo dessa jornada.

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
Escola Politécnica - Departamento de Eletrônica e de Computação
Centro de Tecnologia, bloco H, sala H-217, Cidade Universitária
Rio de Janeiro - RJ CEP 21949-900

Este exemplar é de propriedade da Universidade Federal do Rio de Janeiro, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es).

Agradecimentos

Gostaria de expressar minha gratidão a todos que tornaram possível a realização dessa grande conquista. Aos meus pais, Márcio e Andreha, por ser a base de tudo, com seu amor e incentivo incondicionais. À minha irmã, Giovanna, pela parceria e carinho ao longo de toda a minha vida. À minha namorada, Juliana, pela paciência e compreensão em todos os momentos. Aos meus amigos, com quem compartilhei essa jornada, tornando cada desafio mais fácil de superar. Agradeço também aos meus professores por todo o conhecimento compartilhado, e em particular ao meu orientador, Miguel, cuja dedicação e paciência durante a orientação dessa pesquisa foi indispensável. A cada um de vocês, meu muito obrigado.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro

DESENVOLVIMENTO DE UMA FERRAMENTA PARA MONITORAMENTO DE TRÁFEGO EM REDES INDUSTRIAIS

Guilherme Lucieri Alonso Costa

Agosto/2025

Orientadores: Miguel Elias Mitre Campista
Antônio José Silvério

Curso: Engenharia Eletrônica e de Computação

A crescente conectividade dos sistemas de controle industrial e a sua dependência de protocolos legados, como o Modbus TCP, expõem infraestruturas críticas a ameaças cibernéticas. A falta de mecanismos de segurança inerentes a este protocolo torna as redes industriais vulneráveis a ataques de reconhecimento, negação de serviço e manipulação de processo. Este trabalho propõe e implementa uma ferramenta de detecção de anomalias baseado em rede, não intrusivo e especializado ao ecossistema Modbus TCP. A proposta apresenta uma arquitetura composta por um módulo de construção de perfil, que opera para aprender o comportamento normal da rede, e um módulo de detecção em tempo real. A metodologia consiste, primeiramente, na modelagem do comportamento da rede através da análise de tráfego para estabelecer uma linha de base de parâmetros determinísticos, incluindo endereços IP autorizados, funções Modbus utilizadas, registradores acessados e taxas de comunicação. Subsequentemente, o módulo de detecção emprega uma análise multicamada que não só verifica a conformidade com o perfil de rede, mas também inspeciona a carga útil contra regras de processo definidas manualmente pelo usuário. A eficácia da ferramenta é validada através de um conjunto de simulações, capaz de gerar tanto tráfego de linha de base complexo quanto um espectro de ataques adversários controlados. Nos testes de estresse, o trabalho desenvolvido demonstrou sua resiliência e viabilidade como solução de segurança para redes industriais.

Palavras-chave: Segurança da Comunicação, Inspeção Profunda de Pacotes, Redes Industriais.

Abstract of Capstone Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Engineer

DESIGN AND IMPLEMENTATION OF A TRAFFIC MONITORING TOOL FOR INDUSTRIAL NETWORKS

Guilherme Lucieri Alonso Costa

August/2025

Advisors: Miguel Elias Mitre Campista
Antônio José Silvério

Course: Electronics and Computer Engineering

The growing connectivity of industrial control systems and their dependence on legacy protocols, such as Modbus TCP, expose critical infrastructures to cyber threats. The lack of inherent security mechanisms in this protocol makes industrial networks vulnerable to reconnaissance, denial of service, and process manipulation attacks. This work proposes and implements a non-intrusive, network-based anomaly detection tool specialized for the Modbus TCP ecosystem. The proposal features an architecture composed of a profile construction module, which operates to learn normal network behavior, and a real-time detection module. The methodology first involves modeling network behavior through traffic analysis to establish a baseline of deterministic parameters, including authorized IP addresses, utilized Modbus functions, accessed registers, and communication rates. Subsequently, the detection module employs a multi-layered analysis that not only verifies compliance with the network profile but also inspects the payload against process rules manually defined by the user. The tool's efficacy was validated through a simulation suite capable of generating both complex baseline traffic and a spectrum of controlled adversarial attacks. In stress tests, the developed project demonstrated its resilience and viability as a security solution for industrial networks.

Keywords: Communication Security, Deep Packet inspection, Industrial Networks.

Sumário

Lista de Figuras **xii**

Lista de Tabelas **xiii**

1 Introdução **1**

2 Revisão Técnica **6**

2.1 Ethernet em redes industriais 6

2.2 Protocolo Modbus TCP 7

2.2.1 Comunicação de rede entre servidor e cliente 7

2.2.2 Estrutura dos pacotes 8

2.2.3 Vulnerabilidades do protocolo 10

2.2.4 Requisitos de segurança 11

2.3 Inspeção profunda de pacotes (DPI) 11

3 Arquitetura da Ferramenta Proposta **13**

3.1 Módulo de construção de perfis 15

3.2 Módulo de detecção 15

3.3 Personalização dos limites de operação 15

3.4 Sistema de alertas 16

3.4.1 Detecção de acesso não autorizado 16

3.4.2 Detecção de comportamento anômalo de protocolo 16

3.4.3 Detecção de anomalia de carga útil 16

3.4.4 Detecção de ataques de tráfego intenso 16

4 Implementação **17**

4.1 Principais bibliotecas empregadas 19

4.1.1 Scapy 19

4.1.2 Logging 19

4.1.3 Time 19

4.1.4 Element Tree 20

4.2 Módulo de construção de perfis 20

4.2.1 Estrutura dos pacotes analisados 22
4.2.2 Arquivo `value_ranges.xml` 22
4.3 Módulo de detecção 23
4.3.1 Regra de controle de acesso 23
4.3.2 Regra de comportamento do protocolo 23
4.3.3 Regra de validação semântica 23
4.3.4 Regra de anomalia de volume 24

5 Experimentos e Resultados 25
5.1 Arquitetura da geração de perfis 25
5.2 Arquitetura da simulação de ataques 26
5.2.1 Principais bibliotecas utilizadas para as simulações 27
5.3 Métricas de avaliação do desempenho da ferramenta 27
5.3.1 Acurácia 28
5.3.2 Precisão 28
5.3.3 Sensibilidade 29
5.3.4 F1 score 29
5.4 Análise dos experimentos realizados 29
5.4.1 Ataque de IP desconhecido 29
5.4.2 Ataque de função ou endereço incomum 31
5.4.3 Ataque de injeção de carga útil 32
5.4.4 Ataque de negação de serviço distribuído (DDoS) 34

6 Conclusão e Trabalhos Futuros 37
6.1 Trabalhos futuros 37

Referências Bibliográficas 39

Lista de Figuras

2.1 Ilustração da arquitetura servidor-cliente. 8
2.2 Pilha de protocolos TCP/IP na comunicação Modbus. 9
2.3 Formato do pacote usado pelo protocolo Modbus TCP. 9

3.1 Diagrama contendo a arquitetura da ferramenta. 13

4.1 Arquitetura da ferramenta: fase de construção de perfis 17
4.2 Arquitetura da ferramenta: fase de detecção de anomalias 18

5.1 Ataque de IP desconhecido; taxa média de detecção (sensibilidade) calculada. 30
5.2 Ataque de IP desconhecido; latência média de detecção verificada. . . 31
5.3 Ataque de *payload*; crescimento na latência de detecção. 34
5.4 Configuração A: tempo para o primeiro alerta. 35
5.5 Configuração B: tempo para o primeiro alerta. 36

Lista de Tabelas

4.1 Exemplo de perfil construído pela ferramenta. 20

5.1 Dados dos cenários simulados para a construção de perfis. 26
5.2 Classificação do resultado de cada pacote analisado pela ferramenta. . 28
5.3 Ataque de função ou endereço incomum; métricas de desempenho. . . 31
5.4 Ataque de *payload*; métricas de classificação. 33
5.5 Ataque de *payload*; latência de detecção. 33

Capítulo 1

Introdução

A eclosão de soluções da Indústria 4.0, aliado ao surgimento de novas soluções de otimização da tecnologia operacional nos ambientes de produção, trouxe aos sistemas de automação fabril diversos desafios para garantir a integração entre os equipamentos e a eficiência dos processos produtivos [4]. Nesse sentido, a exigência da constante transmissão de informação entre diversos instrumentos, agregando novas tecnologias a dispositivos legados, amplifica a demanda e a complexidade das redes industriais.

Para que os sistemas ofereçam estabilidade, é necessário uma infraestrutura robusta de comunicação, que possibilite máquinas, computadores, sensores e controladores a trocar informações de forma eficaz. Isso é feito por meio de protocolos de comunicação padronizados, que garantem a confiabilidade dos dados trocados e proporcionam a interoperabilidade entre diferentes equipamentos.

Com a evolução da Indústria 4.0 e a crescente integração entre redes industriais e redes corporativas ou mesmo com a Internet, cresce também a exposição desses sistemas a ameaças cibernéticas. Essa conectividade ampliada expõe os ativos industriais a riscos imediatos e de alto custo para as operações [1, 6], como interrupções operacionais, falhas nos sistemas de controle e danos físicos, além de comprometer a segurança de pessoas e do meio ambiente. Dessa forma, a confiabilidade da comunicação entre os componentes nas redes industriais torna-se essencial. Garantir a integridade e a disponibilidade dos dados trocados entre dispositivos é fundamental para manter a continuidade da produção e assegurar a conformidade com normas e regulamentações [5].

Existem diversos protocolos amplamente utilizados na indústria, que atendem a diferentes necessidades e níveis de complexidade da automação industrial. Protocolos que oferecem maior facilidade de integração e escalabilidade das redes dominam um grande espaço nas soluções industriais. Por exemplo, o Profinet [58] é um padrão de rede caracterizado pela sua alta velocidade e fornecimento de recursos de diagnóstico avançados. É amplamente utilizado em aplicações de monitoramento e

supervisão, como em aplicações de controle de movimento, por permitir a comunicação em tempo real [59], de maneira síncrona, entre controladores lógicos programáveis e servomotores. Por outro lado, existem soluções mais focadas na integração entre softwares de mais alto nível. O *Open Platform Communications Unified Architecture* – OPC UA [56] é uma arquitetura de comunicação frequentemente aplicada em campos de extração de petróleo, possibilitando interconectividade entre máquinas de controle de processo, ou seja, IHMs (Interfaces Homem-Máquina) ou DCSs (Sistemas de Controle Distribuído), e máquinas de processamentos de dados [57]. Seu protocolo suporta estruturas de dados mais complexas para análises de operabilidade, além de robustos mecanismos de segurança.

Nesse contexto, o protocolo Modbus/TCP [55] ocupa um papel de destaque na comunicação em redes industriais. Amplamente utilizado por sua simplicidade, padronização e compatibilidade com uma variedade de equipamentos e sistemas, com destaque a dispositivos legados, ele permite a troca eficiente de dados entre controladores lógicos programáveis (CLPs), sensores, atuadores e supervisórios, especialmente em arquiteturas baseadas em Ethernet. Sua adoção em aplicações de monitoramento e controle em tempo real torna o Modbus uma escolha comum em ambientes industriais, inclusive naqueles em processo de modernização [17]. Entretanto, apesar de sua relevância prática, o protocolo apresenta vulnerabilidades significativas do ponto de vista da confiabilidade da comunicação. Isso se deve, principalmente, ao fato de ter sido desenvolvido há mais de quarenta anos, em um contexto onde não se previa a necessidade de proteção contra ataques externos. Afinal, as redes industriais eram isoladas e fisicamente seguras [9]. Como resultado, a sua comunicação não possui mecanismos nativos de autenticação, criptografia ou controle de acesso, o que o torna suscetível a interceptações, falsificação de comandos e ataques de negação de serviço (DoS) [33]. Por isso, embora o Modbus TCP continue sendo uma solução funcional e amplamente adotada, torna-se essencial implementar medidas complementares de segurança, para mitigar os riscos associados ao seu uso. Diversos estudos já foram desenvolvidos com o objetivo de aprimorar a confiabilidade desses sistemas, sejam eles envolvendo segmentação de rede, *firewalls* industriais, detecção de intrusão ou monitoramento comportamental.

Neste sentido, o presente projeto busca avançar na compreensão do combate de vulnerabilidades na arquitetura do protocolo Modbus TCP, com a finalidade de aprimorar a confiabilidade desses sistemas. Observa-se na literatura uma carência de trabalhos que abordem ferramentas que levam em consideração diferentes parâmetros de comunicação, de maneira integrada. N. Erez e A. Wool [7] apresentam um método de classificação de variáveis e detecção de anomalias voltado para redes Modbus TCP em ambientes SCADA, concentrando-se exclusivamente nos valores dos registradores, que são as unidades de memória, de 16 bits, que armazenam os

dados do dispositivo Modbus. Um classificador automático foi desenvolvido para identificar os tipos de registradores, e foram criados modelos de parametrização para detectar desvios no comportamento esperado. Suchorab et al. [8] propõem um sistema de monitoramento para dados Modbus, projetado para identificar estritamente mudanças não autorizadas de comandos e transmissão de dados falsos para CLPs.

Diferente dos trabalhos anteriores [7, 8], o trabalho proposto amplia a análise para múltiplas camadas da comunicação, monitorando não apenas os valores dos registradores, mas também parâmetros do cabeçalho dos pacotes, padrões de tráfego e comportamento dinâmico dos dispositivos. A importância deste trabalho está na capacidade de detectar comportamentos anômalos por meio da análise de parâmetros que vão além dos valores processados pelos registradores. Para isso, a ferramenta proposta neste projeto coleta todos os dados disponíveis nos cabeçalhos dos pacotes, permitindo a criação de perfis para cada nó da comunicação. Além de avaliar as sequências de transmissão em relação ao padrão habitual, ela monitora os padrões de uso das funções Modbus, o comportamento das máquinas identificadas pelos seus endereços IP e a taxa de transmissão. Essa abordagem supera a análise isolada de mensagens individuais, oferecendo uma visão mais abrangente do tráfego. A ferramenta opera em segundo plano no servidor Modbus, processando o tráfego em tempo real, e possibilita a configuração de parâmetros para se adaptar a diferentes arquiteturas, atendendo de forma mais precisa as necessidades e vulnerabilidades específicas de cada rede industrial.

Este projeto tem como escopo primário o desenvolvimento de uma ferramenta voltada para o monitoramento da comunicação em redes industriais baseadas no protocolo Modbus TCP. As variações e desvios às quais essas redes serão submetidas tem por finalidade emular comportamentos característicos de determinados tipos de ataques cibernéticos. O modelo computacional é voltado à identificação de anomalias, e não ao combate ativo das ameaças. A proposta busca atender a necessidade das indústrias em obter maior segurança e controle sobre o tráfego de rede, possibilitando a supervisão de comportamentos anômalos que possam comprometer a integridade ou a disponibilidade dos sistemas de automação.

Dessa forma, os objetivos específicos da ferramenta incluem: a geração de um perfil padrão para a comunicação em uma arquitetura Modbus/TCP e a identificação de anomalias com base em algoritmos de análise comportamental. A avaliação é realizada a partir da análise de cabeçalho dos pacotes, e os valores transmitidos e recebidos pelos nós da rede. Informações estruturais das mensagens são utilizadas para relatar desvios imediatos nos endereços de IP ou registro, funções atípicas, e valores ou tempos de resposta fora da faixa de aceitação. Em paralelo, os dados são processados de forma conjunta pela ferramenta, a fim de identificar também

comportamentos fora do padrão no fluxo de pacotes e taxa de resposta das mensagens. Isso possibilita que a operação industrial detecte comportamentos atípicos e adote medidas corretivas de forma ágil e eficiente, contribuindo para a segurança e a continuidade dos processos industriais.

A aplicação desenvolvida terá a capacidade de criação de perfis personalizados, permitindo que a análise seja adaptada aos padrões e exigências operacionais de cada ambiente industrial. A ferramenta é composta por dois módulos principais: o módulo de estabelecimento de perfis e o módulo de detecção. O módulo de estabelecimento de perfis tem como função analisar o tráfego de rede durante uma fase controlada e segura da operação, realizada de forma isolada em ambiente de teste, ou por meio de simulações virtuais que replicam as especificações da rede. Nesse estágio, o programa é encarregado de processar os dados e gerar um perfil de comportamento típico, que servirá de referência para o monitoramento. Ele será responsável por criar classes de comportamento para cada nó que interage com a rede, a partir de critérios determinísticos. Já o módulo de detecção atua na supervisão contínua da comunicação, utilizando o perfil previamente gerado para identificar desvios no comportamento esperado e, assim, emitir alertas em tempo real aos operadores.

A partir da metodologia de monitoramento elaborado, pretende-se evidenciar a importância de ferramentas de análise no escopo da segurança de redes industriais, atuando de forma complementar aos *firewalls* e outros mecanismos de proteção. O êxito deste trabalho está centrado na implementação prática de uma aplicação *plug and play* e adaptável, para monitoramento de comunicação em redes Modbus TCP, seguindo algumas hipóteses previamente definidas. Técnicas de *packet sniffing* [2] e inspeção profunda de pacotes [3] são empregadas na construção da ferramenta proposta.

As arquiteturas de rede usadas nos testes foram simuladas com base em cenários reais da indústria. A simulação dos cenários de teste foi realizada através de emuladores de servidores e clientes Modbus TCP, em dispositivos conectados à mesma rede e utilizando aplicações próprias para essa finalidade. Dessa forma, o ambiente de experimentação foi exposto a diferentes cenários de ataques cibernéticos, com o objetivo de validar a acurácia da ferramenta em detectar as abordagens de agentes maliciosos na rede, assim como avaliar o tempo de resposta da aplicação em identificá-los.

Este trabalho está estruturado da seguinte forma: O Capítulo 2 apresenta uma revisão dos principais conceitos técnicos relacionados a redes industriais, automação e cibersegurança, que servem de base para o desenvolvimento do projeto. No Capítulo 3, é apresentada a proposta da ferramenta, bem como a descrição de sua arquitetura. O Capítulo 4, aborda os detalhes da implementação do projeto, contemplando tanto o módulo de construção de perfis quanto o módulo de detecção de

anomalias. O Capítulo 5 descreve o ambiente de teste utilizado, os procedimentos experimentais adotados e a análise dos resultados obtidos. Por fim, o Capítulo 6 reúne as conclusões da pesquisa e propõe possíveis direcionamentos para trabalhos futuros.

Capítulo 2

Revisão Técnica

Este capítulo apresenta uma revisão técnica dos principais conceitos de rede e segurança abordados no trabalho. O objetivo é fornecer o embasamento teórico que sustenta a metodologia elaborada. Para isso, são introduzidas noções fundamentais de protocolos de comunicação, para contextualizar o problema abordado e o foco de aplicação do projeto. Na sequência, importantes conceitos sobre o protocolo de comunicação Modbus TCP, que constitui o objeto central de aplicação da ferramenta desenvolvida, são abordados.

2.1 Ethernet em redes industriais

A Ethernet é uma tecnologia de comunicação amplamente utilizada em redes locais (*Local Area Networks* – LANs) [26]. Padronizada pelo IEEE por meio da norma 802.3 [25], ela opera nas camadas física e de enlace do modelo OSI [27], definindo os métodos de formatação, transmissão e recepção de dados entre dispositivos de rede. A unidade básica de transmissão na Ethernet é o quadro, que inclui campos como endereços MAC de origem e destino, informações de controle de acesso ao meio e verificação de erros, assegurando a integridade dos dados transmitidos [25]. Na automação industrial, a Ethernet tornou-se essencial devido à sua robustez, baixa latência, alta largura de banda e compatibilidade com protocolos industriais [24].

Protocolos de camadas superiores, como a pilha TCP/IP [10], utilizam a tecnologia Ethernet como a principal alternativa para transporte de dados estruturados. O processo fundamental é o de encapsulamento, onde os datagramas IP, que por sua vez contêm segmentos TCP, são inseridos no campo de carga útil dos quadros Ethernet, para serem transportados pela rede local. Dessa forma, protocolos como Modbus TCP, Profinet e EtherNet/IP, ao operarem sobre a pilha TCP/IP, se beneficiam diretamente da infraestrutura Ethernet, com alta velocidade de transmissão [12]. Contudo, a Ethernet se responsabiliza apenas pela entrega do quadro no segmento de rede local. A garantia de confiabilidade na entrega de ponta a ponta, o controle

de fluxo e o ordenamento dos dados são funções delegadas a protocolos de camadas superiores, como o TCP [18].

2.2 Protocolo Modbus TCP

O Modbus TCP é um protocolo de comunicação baseado na pilha de protocolos TCP/IP [10], sendo operado sobre redes Ethernet. Ele permite a integração de dispositivos industriais, como CLPs (Controladores Lógicos Programáveis), IHMs (Interfaces Homem-Máquina), sistemas SCADA e sensores inteligentes, em redes locais (LANs) [28]. Ele é fundamentado no modelo de comunicação servidor-cliente, onde o cliente inicia a comunicação enviando requisições ao servidor, que responde com os dados solicitados ou confirmações [31].

A escolha do protocolo Modbus TCP como objeto central de estudo e aplicação para a ferramenta desenvolvida nesse trabalho foi motivada por sua presença expressiva no setor industrial e suas vulnerabilidades de segurança inerentes. Por ser um padrão de comunicação amplamente adotado, especialmente em sistemas com equipamentos legados, qualquer melhoria em sua segurança tem um impacto potencial abrangente. De forma análoga, sua simplicidade e ausência de mecanismos de segurança o torna um dos vetores de ataque mais explorados em redes de automação. Portanto, o desenvolvimento de uma camada de segurança complementar, como a proposta nesse trabalho, aborda uma necessidade real de proteção dos ambientes industriais.

2.2.1 Comunicação de rede entre servidor e cliente

O modelo servidor-cliente é uma arquitetura de comunicação em que um dispositivo central controla a troca de dados com um ou mais dispositivos subordinados. O servidor é responsável por interpretar todas as requisições, como leitura ou escrita de dados, enquanto os clientes apenas enviam essas solicitações [30]. Essa abordagem é comum em redes industriais pela simplicidade e previsibilidade do controle centralizado, embora limitar a comunicação entre dispositivos clientes reduza a flexibilidade da rede [29]. O diagrama da Figura 2.1 é uma ilustração desse modelo de comunicação.

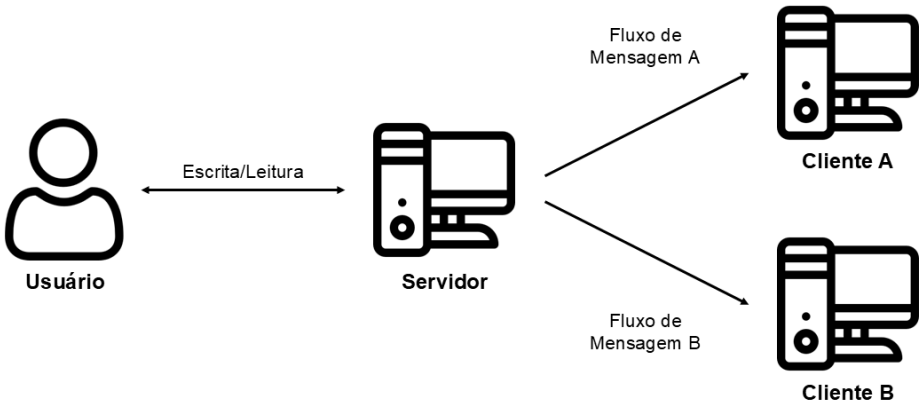

Figura 2.1: Ilustração da arquitetura servidor-cliente.

2.2.2 Estrutura dos pacotes

Na figura 2.2 é possível identificar como, no protocolo Modbus TCP, a mensagem Modbus é encapsulada pelo protocolo TCP, diante da estrutura da pilha de protocolos TCP/IP. Um cabeçalho específico chamado MBAP (*Modbus Application Protocol*) é adicionado, que possui 7 bytes distribuídos nos seguintes campos principais: (1) o identificador de transação (2 bytes), usado para associar respostas a requisições específicas, garantindo o correto pareamento em comunicações simultâneas; (2) o identificador de protocolo (2 bytes), normalmente definido como 0x0000 para indicar o protocolo Modbus; (3) o campo de comprimento (2 bytes), que indica o número de bytes restantes na mensagem, incluindo o identificador de unidade, código de função e dados; (4) o identificador de unidade (1 byte), que equivale ao endereço do dispositivo na rede Modbus TCP e permite distinguir dispositivos em arquiteturas que podem conter múltiplos nós.

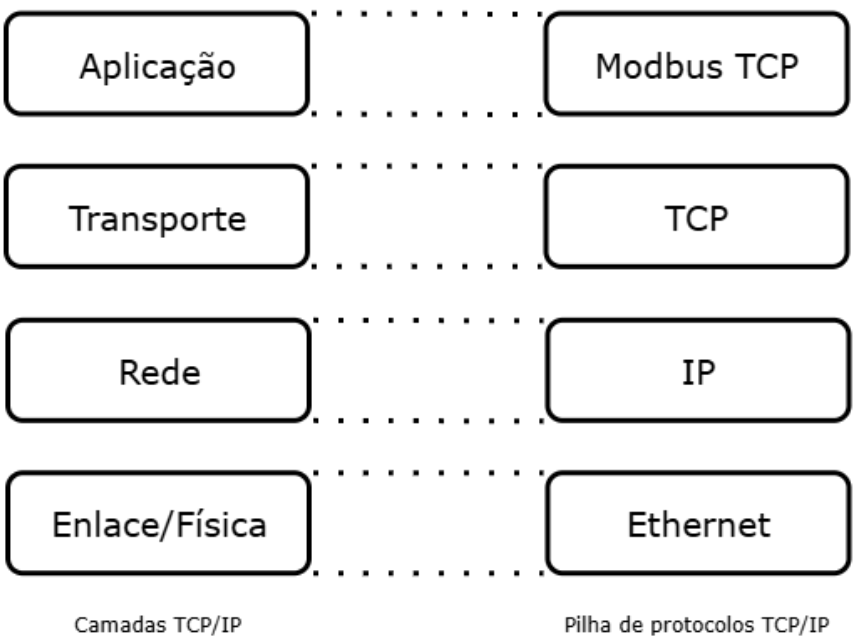

Figura 2.2: Pilha de protocolos TCP/IP na comunicação Modbus.

A Figura 2.3 representa a estrutura detalhada do pacote Modbus TCP. Formalmente denominada Unidade de Dados da Aplicação (*Application Data Unit* – ADU), sua estrutura é composta por duas seções principais: o cabeçalho MBAP e a PDU Modbus. O cabeçalho MBAP (*Modbus Application Protocol*) é um prefixo de 7 bytes específico para a implementação sobre TCP/IP, cuja função é gerenciar as transações e identificar os dispositivos clientes. Anexada ao MBAP está a PDU (*Protocol Data Unit*), que constitui o núcleo do comando e é subdividida em dois campos: (1) o código de função, que indica a operação a ser executada, como leitura ou escrita; e (2) o campo de dados, de tamanho variável, que contém as informações da operação, como endereços e valores. A comunicação é sempre iniciada pelo cliente, que estabelece uma conexão com o servidor utilizando a porta TCP 502, que é a porta padrão reservada para a aplicação Modbus [28].

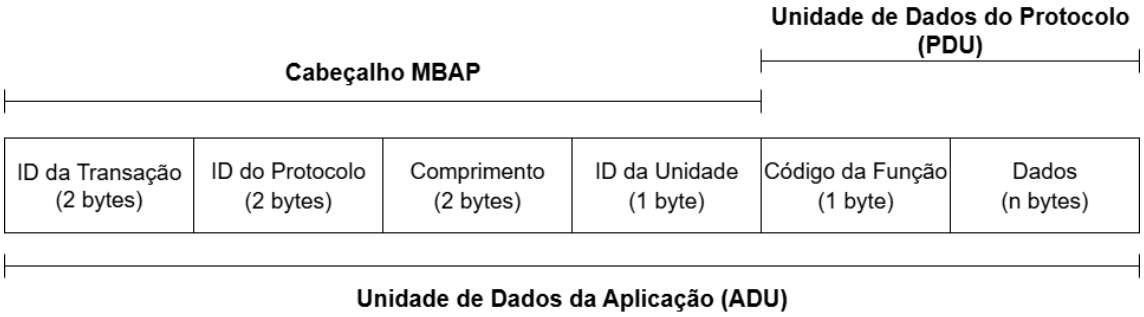

Figura 2.3: Formato do pacote usado pelo protocolo Modbus TCP.

2.2.3 Vulnerabilidades do protocolo

As redes Modbus TCP apresentam vulnerabilidades significativas devido à sua concepção original, que não contemplava requisitos de segurança cibernética. Projetado para ambientes isolados, o protocolo opera sem mecanismos nativos de autenticação, criptografia ou controle de acesso, o que o torna suscetível a diferentes tipos de ataques quando exposto a redes corporativas ou à Internet [19, 20].

Entre os principais vetores de ataque, destacam-se: o *spoofing*, em que o invasor falsifica a identidade de um dispositivo ou usuário legítimo na rede, geralmente manipulando endereços IP, endereços MAC ou identificadores de protocolo, para obter acesso não autorizado, enganar mecanismos de proteção ou interceptar comunicações [35]; o *DoS (Denial of Service)*, que visa sobrecarregar o servidor com requisições contínuas, interrompendo a operação normal do sistema [33]; e o *DDoS (Distributed Denial of Service)*, que funciona da mesma forma do ataque de negação de serviço tradicional, mas com o agravante de ser gerado por múltiplos agentes maliciosos distribuídos. Em adição, a ausência de validação no conteúdo das mensagens pode permitir a modificação de registradores críticos, comprometendo a integridade dos dados e o controle de processos. Esses ataques serão usados como base para os experimentos que avaliam o desempenho da proposta deste projeto. A arquitetura e as funcionalidades da ferramenta desenvolvida serão apresentados em detalhe no Capítulo 3.

É importante reconhecer que a segurança de redes Modbus/TCP em ambientes industriais não depende de uma solução única, mas sim de uma estratégia de defesa em profundidade, na qual múltiplas camadas de proteção atuam em conjunto. Nesse contexto, existem outras abordagens consolidadas, como o uso de Redes Privadas Virtuais (*Virtual Private Connections* – VPNs) com IPsec [60] para criptografar a comunicação, e a implementação de TLS [61] nos pacotes Modbus. A ferramenta desenvolvida nesse trabalho não se posiciona como uma substituta para essas tecnologias, mas sim como uma camada de segurança complementar e contextual. Suas vantagens residem em áreas onde as abordagens tradicionais apresentam limitações, além do custo intrínseco à qualquer uma dessas qualquer abordagem de segurança. Enquanto uma VPN protege o tráfego contra interceptação externa, ela não oferece proteção contra ameaças internas ou dispositivos já comprometidos que operam dentro do túnel seguro. Similarmente, um *firewall* pode permitir o tráfego de um CLP na porta assignada a ele, mas não é capaz de discernir se o comportamento daquele CLP, as funções que ele executa, os endereços que acessa e a frequência com que o faz, desviou de seu padrão normal.

2.2.4 Requisitos de segurança

No contexto de sistemas de automação industriais abordado nesse trabalho, o conceito de segurança está intrinsecamente ligado à proteção do processo físico e à garantia da continuidade operacional. Dessa forma, os pilares da segurança em Tecnologia de Automação são priorizados para refletir esses objetivos críticos, com ênfase na disponibilidade, na integridade e, subsequentemente, na confidencialidade. A disponibilidade assegura que os sistemas de controle estejam sempre operantes para evitar paradas na produção; a integridade garante que os dados e comandos não sejam corrompidos ou falsificados, protegendo o processo de manipulações perigosas; e a confidencialidade visa proteger informações operacionais sensíveis contra espionagem.

Para atender a esses requisitos, a ferramenta emprega uma metodologia de detecção de anomalias. Diferente de sistemas baseados em assinaturas, que procuram por padrões de ataques já conhecidos, a detecção de anomalias opera estabelecendo primeiro um perfil detalhado do comportamento normal e legítimo da rede, ou seja, uma linha de base. Uma anomalia é, então, definida como qualquer evento ou padrão de comunicação que se desvie significativamente desta linha de base estabelecida.

2.3 Inspeção profunda de pacotes (DPI)

Conforme descrito na Seção 2.1, a tecnologia Ethernet é responsável por gerenciar a transmissão de dados em uma rede local por meio de quadros. Contudo, para que a comunicação possa ocorrer para além do segmento de rede local e ser direcionada através de múltiplos roteadores, é necessário um protocolo para gerenciar o endereçamento lógico. É neste contexto que atua o Protocolo de Internet (IP), cuja unidade fundamental de comunicação é o pacote. Este pacote IP é, por sua vez, encapsulado dentro da carga útil de um quadro Ethernet para ser transportado. A inspeção de pacotes, portanto, é a técnica utilizada para monitorar e analisar o tráfego em uma rede, examinando as informações contidas nessas unidades de comunicação, que são organizadas em cabeçalhos e carga útil (*payload*), a fim de viabilizar a segurança e a gestão da rede [13].

A forma mais básica de inspeção de pacotes, frequentemente denominada inspeção de cabeçalhos, consiste na análise dos campos presentes nos cabeçalhos dos pacotes, como endereços IP de origem e destino, números de porta, protocolos utilizados e flags de controle. Essa abordagem é eficaz para operações de filtragem e roteamento simples, porém apresenta limitações diante de técnicas modernas de evasão, que são métodos utilizados por um atacante para disfarçar ou ofuscar tráfego malicioso, fazendo-o parecer legítimo para os mecanismos de segurança que

analisam apenas informações superficiais. No tunelamento de protocolos, por exemplo, um protocolo não autorizado é encapsulado dentro do tráfego de um protocolo permitido (como DNS ou HTTP), passando despercebido. Outra técnica de evasão muito comum é a utilização indevida de portas padrão, que consiste em executar um serviço malicioso em uma porta normalmente confiável e liberada, para burlar regras de firewall baseadas unicamente em portas [14].

Já a inspeção profunda de pacotes (*Deep Packet Inspection* – DPI) melhora a detecção de tráfego malicioso, ao examinar não apenas os cabeçalhos dos pacotes, mas também o conteúdo completo da carga útil, permitindo identificar ameaças que métodos tradicionais de inspeção superficial não detectam. Ao analisar toda a informação contida nos pacotes, a DPI consegue reconhecer padrões suspeitos, desvios de protocolo, tentativas de exfiltração de dados, *malwares*, ataques de negação de serviço e outras atividades maliciosas ocultas no tráfego [14, 15].

Ferramentas que combinam inspeção profunda de pacotes com sistemas de detecção de intrusão ampliam ainda mais a capacidade de identificar tráfego malicioso, ao correlacionar informações extraídas dos pacotes com análises de dados e padrões conhecidos de ataque [16]. Por fim, a DPI também monitora o tráfego de saída, ajudando a prevenir vazamentos de dados e a identificar destinos suspeitos.

Capítulo 3

Arquitetura da Ferramenta Proposta

O protocolo Modbus TCP adota uma arquitetura clássica servidor-cliente, na qual o servidor atua como unidade central de controle, responsável por interpretar requisições de leitura e escrita, enquanto os clientes iniciam ações fornecendo dados ou executando comandos. Esses clientes podem ser dispositivos de I/O, unidades de rede ou sensores e atuadores responsáveis pela aquisição de variáveis de processo. Conforme ilustrado na Figura 3.1, a ferramenta proposta é estrategicamente implementada para operar de forma passiva e não intrusiva na rede, inspecionando o tráfego. O ciclo operacional envolve a coleta de dados pelo servidor e o envio das respostas aos clientes, consolidando o fluxo completo das informações, e o endereçamento IP garante a identificação única de cada nó Modbus na rede [32].

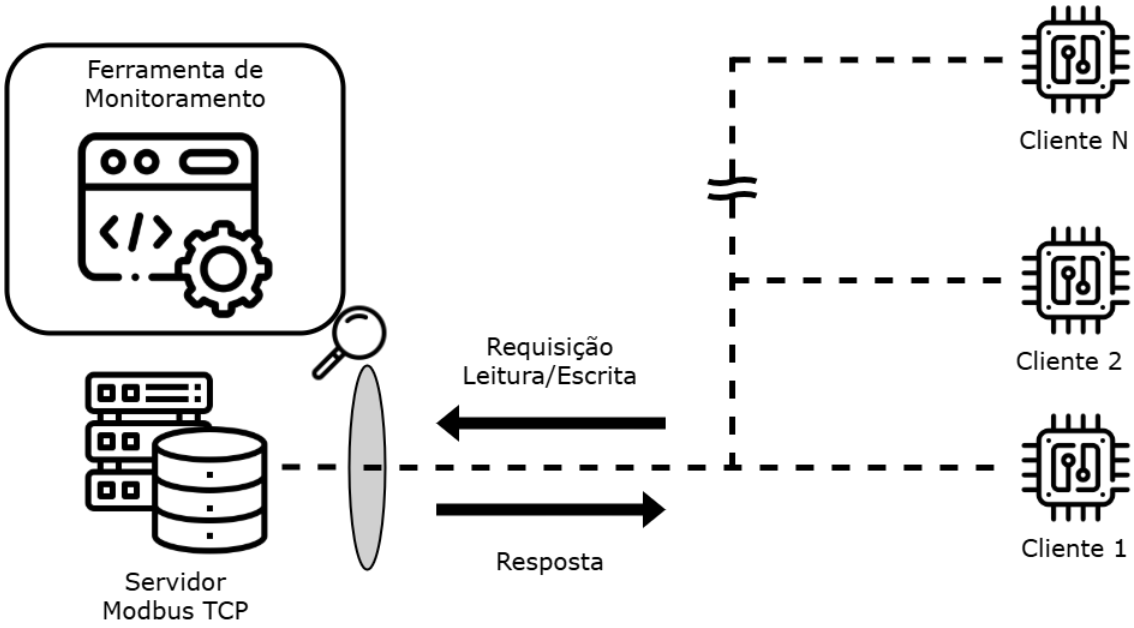

Figura 3.1: Diagrama contendo a arquitetura da ferramenta.

A ferramenta proposta neste trabalho é composta por dois módulos principais: o módulo de estabelecimento de perfis e o módulo de detecção. O módulo de estabelecimento de perfis analisa o tráfego durante uma janela de operação controlada, em ambiente isolado ou por meio de simulações virtuais que replicam as especificações da rede da planta industrial. Nessa fase, o programa gera um perfil de comportamento típico, que serve como referência para o monitoramento posterior. Na segunda etapa, o módulo de detecção opera de forma contínua no ambiente real de operação, identificando desvios em relação ao comportamento esperado e emitindo alertas em tempo real aos operadores. As condições de alerta geradas incluem: IP desconhecido, código de função incomum, endereço de registro incomum, valor de escrita fora da faixa e taxa de pacotes anormal.

Na literatura, trabalhos como o de N. Erez e A. Wool [7] demonstram a eficácia da inspeção da carga útil para detecção de anomalias em redes Modbus, conforme discutido no Capítulo 1. Contudo, a solução desenvolvida neste projeto diferencia-se por realizar uma análise mais abrangente, que inspeciona um conjunto de atributos extraídos de todo o encapsulamento do pacote, desde os cabeçalhos de rede e aplicação, até a própria carga útil. Essa abordagem permite a identificação de padrões anômalos alternativos ou mais sutis. Adicionalmente, a ferramenta oferece parâmetros ajustáveis, viabilizando a customização das análises conforme o perfil operacional da planta, o que confere maior flexibilidade e adaptabilidade frente a diferentes cenários industriais. Em comparação com soluções existentes, essa diversidade e ajustabilidade conferem maior versatilidade à solução desenvolvida.

Além disso, é fundamental distinguir a função da ferramenta proposta daquela de um *firewall* tradicional, pois ambos operam sob paradigmas de segurança distintos, embora complementares. Um *firewall* atua primariamente como um mecanismo de controle de acesso no perímetro da rede, aplicando um conjunto de regras estáticas que permitem ou bloqueiam o tráfego com base em parâmetros de cabeçalho, como endereços IP e portas de comunicação. Em contraste, a ferramenta desenvolvida opera como um sistema de detecção de intrusão comportamental, que vai além do perímetro para analisar o tráfego já permitido. Sua análise é contextual, avaliando não apenas a identidade da fonte, mas também a legitimidade de suas ações, a semântica de sua carga útil e seu padrão temporal de tráfego. Além disso, um dos principais ganhos da ferramenta, do ponto de vista do usuário, é sua capacidade de traçar os perfis de forma autônoma, em oposição aos *firewalls*, em que a configuração deve ser feita manualmente, prevendo todas as variáveis da arquitetura da planta industrial.

A seguir, são detalhadas as funcionalidades específicas da ferramenta proposta, evidenciando como cada uma contribui para a detecção de anomalias:

3.1 Módulo de construção de perfis

A funcionalidade primordial da ferramenta é sua capacidade de mapear detalhadamente o comportamento de uma rede em cenário industrial. Para isso, o usuário inicia um período de construção de perfis, em um ambiente de rede controlado e seguro. Durante esta fase, a ferramenta captura e analisa todo o tráfego Modbus TCP para mapear as interações entre os dispositivos. Ao final do período, ela gera e salva automaticamente um perfil de comportamento para cada nó da comunicação. Esse perfil serve como uma fotografia detalhada da operação normal, estabelecendo uma linha de base que inclui todos os clientes autorizados, as funções Modbus que eles tipicamente executam e os endereços de registradores que costumam acessar. A garantia de que o perfil de normalidade não esteja contaminado por atividades maliciosas preexistentes é um requisito metodológico fundamental. Isso é tipicamente alcançado executando-se a fase de construção de perfil durante o comissionamento de novos sistemas, em um ambiente de bancada de testes replicado, ou durante uma janela de manutenção em redes existentes, na qual a rede de automação é temporariamente isolada de conexões externas.

3.2 Módulo de detecção

Uma vez que o perfil de normalidade é estabelecido, o módulo de detecção pode ser utilizado para monitorar continuamente a comunicação da rede em tempo real. Cada pacote Modbus TCP que trafega na rede é interceptado pelo *sniffer* [23] da ferramenta, e submetido a uma análise multicamada, ou seja, aplica diferentes tipos de regras de segurança com base nas informações extraídas dos segmentos do pacote. Essas regras são formadas a partir do perfil de normalidade previamente gerado, e qualquer violação de uma dessas normas aciona um alerta imediato, indicando um desvio do comportamento esperado.

3.3 Personalização dos limites de operação

É possível que o usuário defina manualmente os limites operacionais seguros para seus registradores. Através de um arquivo de configuração, o operador pode, de forma opcional, especificar valores mínimos e máximos para qualquer endereço, incorporando as especificações técnicas da planta industrial diretamente na lógica de detecção.

3.4 Sistema de alertas

A principal contribuição deste trabalho reside em sua capacidade de detectar diferentes classes de anomalias. Toda anomalia detectada resulta em uma notificação imediata no console de operação da ferramenta, e é registrada com detalhes no arquivo de *logs* do sistema, criando um registro histórico auditável. Os diferentes tipos de alertas são detalhados a seguir:

3.4.1 Detecção de acesso não autorizado

A ferramenta identifica qualquer tentativa de comunicação de um endereço IP que não pertença ao perfil de dispositivos conhecidos. Isso serve como uma primeira linha de defesa contra invasores externos ou dispositivos não autorizados conectados à rede.

3.4.2 Detecção de comportamento anômalo de protocolo

Para os dispositivos já conhecidos e autorizados, a ferramenta monitora se suas ações estão de acordo com seu comportamento histórico individual, que foi mapeado durante a construção dos perfis. Um alerta é gerado se um dispositivo legítimo tentar executar uma função Modbus ou acessar um endereço de registrador que não faz parte de seu perfil particular, o que pode indicar o comprometimento de um dispositivo interno ou uma falha operacional.

3.4.3 Detecção de anomalia de carga útil

A ferramenta inspeciona o conteúdo semântico dos pacotes. Ela gera um alerta se uma operação de escrita tentar inserir um valor fora dos limites seguros e pré-estabelecidos para os registradores. Essa funcionalidade é essencial para prevenir a manipulação de processos cruciais da operação industrial.

3.4.4 Detecção de ataques de tráfego intenso

A ferramenta monitora a taxa de pacotes enviadas e recebidas por minuto, para o servidor e para cada cliente. Se o volume de tráfego de um ou mais IPs excederem o limite de normalidade estabelecido no perfil, um alerta de negação de serviço é gerado. Isso protege o servidor contra ataques de inundação que visam esgotar seus recursos de processamento.

Capítulo 4

Implementação

A solução desenvolvida consiste em uma ferramenta de detecção de anomalias, com foco específico no protocolo Modbus TCP. A sua arquitetura, representada na Figura 4.1 e na Figura 4.2, foi projetada para ser passiva, não interferindo no tráfego da rede de automação, e opera com base em um modelo de regras e perfis de comportamento para identificar atividades anômalas que possam indicar uma falha de sistema ou uma ação maliciosa. O código referente à implementação da ferramenta está armazenado no GitHub [39].

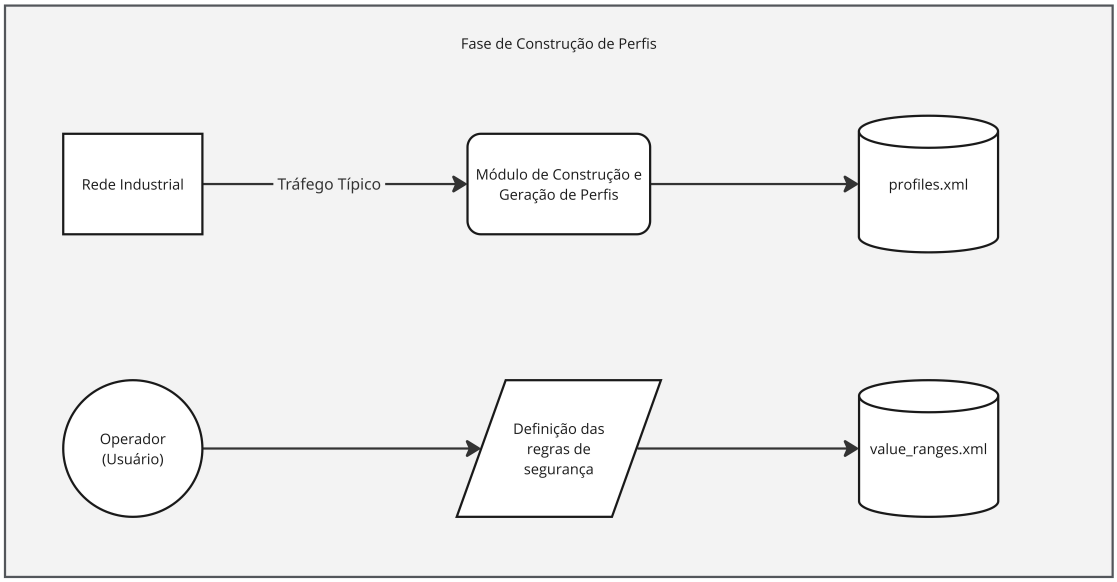

Figura 4.1: Arquitetura da ferramenta: fase de construção de perfis

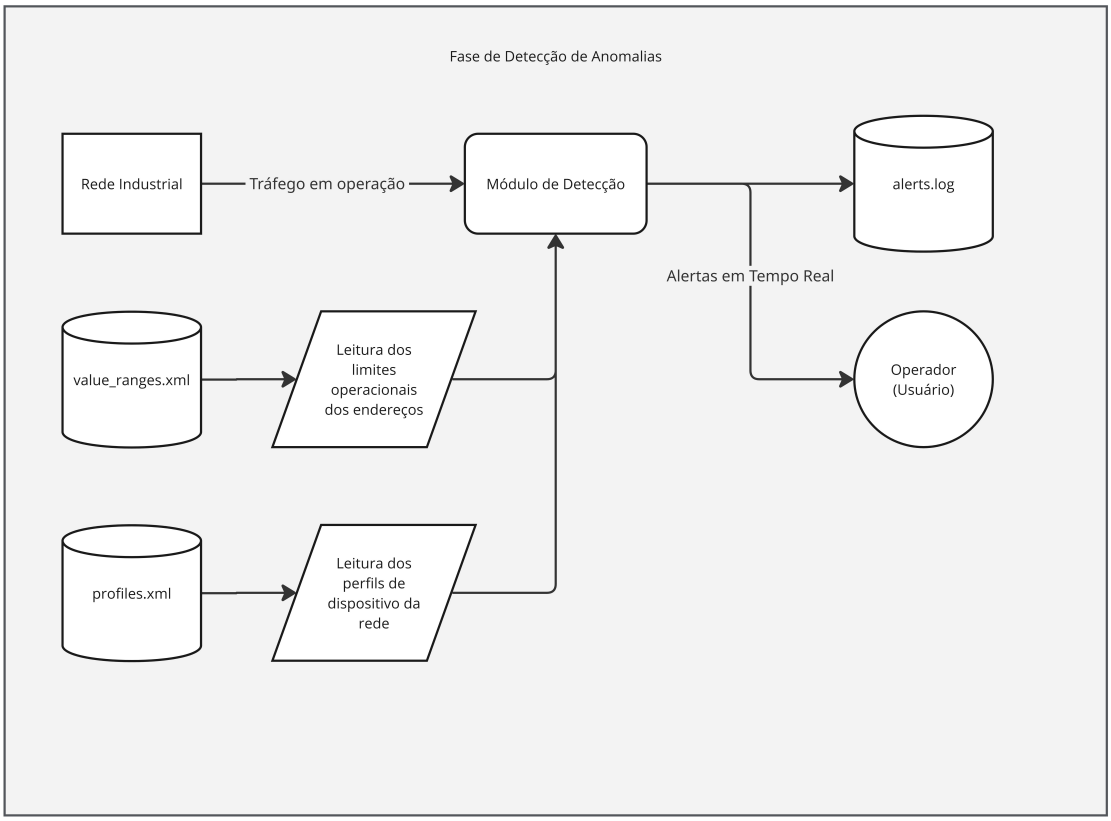

Figura 4.2: Arquitetura da ferramenta: fase de detecção de anomalias

A arquitetura da ferramenta é modular, dividida em dois módulos principais e independentes, cada um implementado como uma execução distinta. Essa separação garante que o processo de construção dos perfis, que pode consumir mais recursos, não interfira na operação de detecção em tempo real, que é otimizada para desempenho e baixa latência. Além disso, a independência entre os modos concede a capacidade do comportamento típico da rede ser traçado a partir de um programa de terceiros, com metodologias distintas deste trabalho, contanto que o arquivo de perfil estático, gerado pelo mapeamento em formato xml, seja compatível com o formato adotado neste projeto.

Utilizando a biblioteca scapy [40], detalhada neste capítulo, a ferramenta opera monitorando a rede sob o paradigma de *passive sniffing* [22]. Para garantir a visibilidade do tráfego relevante, a ferramenta foi projetada para ser executada no *host* que atua como o servidor da comunicação Modbus TCP. Ao operar nesse host, a ferramenta consegue inspecionar todas as requisições que os clientes enviam para o servidor, e todas as respostas que ele devolve para esses clientes.

O mecanismo de detecção da ferramenta é fundamentado na utilização de perfis de comportamento, que servem como uma linha de base para definir a normalidade da rede. O perfil, neste contexto, é um modelo de dados determinístico que descreve o padrão de comunicação esperado e é contextual para cada dispositivo individu-

almente. Esses modelos de comportamento são constituídos por um conjunto de parâmetros específicos que caracterizam a operação de cada cliente, incluindo seu identificador de rede (endereço IP), o repertório de funções de protocolo que utiliza, o escopo de endereços de memória que acessa, e seu padrão temporal de comunicação, como a taxa de pacotes. O cálculo e a construção desses perfis são realizados de forma automatizada durante a fase de construção, na qual a ferramenta monitora o tráfego e agrega os referidos parâmetros para cada dispositivo ativo. Uma vez estabelecido, esse modelo de normalidade é então empregado pelo módulo de detecção, que realiza majoritariamente uma análise *stateless*, onde cada pacote é avaliado individualmente contra as regras estáticas do perfil. Contudo, para anomalias de natureza temporal, como ataques de negação de serviço, a ferramenta implementa uma análise *stateful*, mantendo um histórico recente dos pacotes para identificar desvios no volume de comunicação que violam o comportamento aprendido.

4.1 Principais bibliotecas empregadas

A ferramenta foi inteiramente desenvolvida em Python, aproveitando seu ecossistema de bibliotecas para manipulação de rede e dados. As bibliotecas fundamentais para o software são:

4.1.1 Scapy

A biblioteca `Scapy` é o componente crucial da ferramenta para toda a interação com a rede e grande parte das funcionalidades específicas do programa. A biblioteca é utilizada para a captura de pacotes, filtragem de tráfego por porta e, principalmente, para a dissecção camada por camada dos pacotes Modbus TCP. A extensão `scapy.contrib.modbus` fornece as classes necessárias para interpretar a estrutura dos conjuntos ADU (*Application Data Unit*) e PDU (*Protocol Data Unit*) do Modbus [40].

4.1.2 Logging

O `Logging` é uma biblioteca padrão do Python, útil para a criação de registros de eventos. É empregada no módulo de detecção para gerar um log persistente e formatado de todas as anomalias detectadas, para auditoria [48].

4.1.3 Time

A biblioteca `time` foi utilizada para todo o gerenciamento temporal dos mecanismos da ferramenta. Ela fornece funcionalidades para medição de intervalos de tempo,

medição de atraso e formatação de estampas de tempo [45].

4.1.4 Element Tree

O módulo `xml.etree.ElementTree` é utilizado para a construção e persistência dos perfis de comportamento. A escolha do formato XML para os perfis, armazenados no arquivo `profiles.xml` visa garantir que o perfil gerado seja estruturado e manipulável de forma eficiente pelo programa, mas também seja legível por humanos e facilmente extensível [47]. Além disso, o módulo `xml.dom.minidom` [50] foi utilizado de maneira complementar ao `Element Tree`, para a depuração e formatação correta dos arquivos.

4.2 Módulo de construção de perfis

Este módulo é responsável por aferir o comportamento normal da rede. Ele é executado em um ambiente controlado ou durante uma janela de operação segura para analisar o tráfego e gerar um modelo de normalidade. Os perfis de comportamento são agregados em um arquivo formatado em XML, nomeado `profiles.xml`, que serve de base para o módulo de detecção. A estrutura do arquivo é centrada no elemento `ip_profile`, que encapsula o modelo de comportamento para um único cliente da rede, identificado pelo seu atributo de endereço IP. Dentro de cada perfil, o modelo é detalhado em três seções distintas: (1) um conjunto de códigos de função permitidos, que define o repertório de operações legítimas para o dispositivo; (2) um conjunto de endereços de registradores acessados, que delimita o escopo de dados com o qual o dispositivo interage; e (3) um limite de taxa de comunicação, que estabelece o limiar máximo de pacotes por minuto para o comportamento temporal normal. Na Tabela 4.1 é descrito um exemplo de perfil de um dispositivo, traçado pela ferramenta.

Tabela 4.1: Exemplo de perfil construído pela ferramenta.

Descrição	**Detalhes**
Identificador do dispositivo	192.168.1.1
Códigos de função permitidos	0x03, 0x04, 0x06.
Endereços de registradores permitidos	40001, 40005, 40009, 40013.
Taxa máxima de pacotes por minuto	100 pacotes por minuto.

A captura e processamento dos pacotes é realizado a partir da função `process_packet()`. É aplicado um filtro durante o processo de *sniffing*, para que apenas pacotes Modbus TCP sejam processados. O código utiliza o método `scapy.sniff`,

com a função `process_packet` designada como um *callback*, que é executado imediatamente para cada pacote que responde ao filtro aplicado. Esse *callback* com filtro evita o armazenamento de grandes volumes de pacotes, tornando assim a captura mais eficiente em termos de memória.

A função `process_packet` recebe cada pacote capturado. Primeiramente, ela verifica a presença da estrutura `ModbusADURequest` e em caso positivo, disseca o pacote para extrair seus atributos chave. O acesso aos campos é feito camada por camada: `packet[IP].src` para o IP de origem, `packet[ModbusADURequest].funcCode` para o código de função, e `packet[ModbusADURequest].payload.startAddr` para o endereço do registrador alvo da operação, acessando o payload da ADU para chegar à estrutura PDU do cabeçalho MBAP dos pacotes.

Os dados extraídos são agregados em uma estrutura de dados em memória, usufruindo da biblioteca `defaultdict`. Para cada dispositivo, seja algum dos clientes ou o servidor Modbus, a ferramenta armazena um conjunto de códigos de função e um conjunto de endereços que aquele endereço IP acessou, e o uso de `sets` garante o armazenamento de valores únicos. Adicionalmente, a estampa de tempo de cada pacote é registrado para o cálculo da taxa de comunicação.

Ao final do período de construção dos perfis, a ferramenta executa a etapa final de processamento e consolidação dos dados. Utilizando métodos da biblioteca `xml.etree.ElementTree`, o sistema itera sobre a estrutura de dados que foi agregada em memória. Nesta estrutura, cada dispositivo é unicamente identificado e indexado pelo seu endereço IP de origem, uma abordagem fundamentada na prática comum de endereçamento estático para dispositivos em redes de automação. Uma das análises mais importantes nessa fase é o cálculo do perfil temporal de cada cliente e do servidor. A função `calculate_rate_limits` é invocada para determinar a taxa de tráfego normal para cada endereço IP, incluindo para o servidor modbus. Esse cálculo não se baseia em uma simples média, que poderia mascarar picos de tráfego, mas sim na identificação do pico máximo de pacotes observados em qualquer janela de tempo deslizante de 60 segundos durante todo o período de aprendizado. A este valor máximo é então adicionada uma margem de tolerância pré-definida de 20% por padrão, mas que é configurável pelo usuário. Isso possibilita criar um limiar de segurança, que acomoda pequenas variações operacionais e minimiza a ocorrência de falsos positivos. Por fim, constrói-se um nó XML para cada dispositivo analisado na rede, que consolida todas as informações aprendidas: funções permitidas, endereços acessados e o limite de taxa. Esta árvore é então serializada e salva no arquivo `profiles.xml`, criando um modelo de normalidade persistente.

4.2.1 Estrutura dos pacotes analisados

A Seção 2.2.2 introduz a estrutura dos pacotes lidos pela ferramenta, sob o ponto de vista da estrutura do TCP/IP [38]. O software inspeciona campos em múltiplas camadas:

Camada de rede (IP)

O campo do IP de origem é o primeiro ponto de análise, utilizado para identificar o remetente do pacote e associá-lo a um perfil de comportamento. O uso deste campo como identificador de dispositivos permite que a ferramenta detecte requisições com IP desconhecido.

Camada de transporte (IP)

A Porta de Destino é usada para filtrar e capturar exclusivamente o tráfego Modbus.

Camada de aplicação (Modbus TCP)

Esta é a camada mais crítica para a análise. Um pacote Modbus TCP consiste em duas partes principais. Primeiramente, o cabeçalho MBAP, que a ferramenta interpreta como a camada `ModbusADURequest`. Ele contém os campos de `transactionId`, para parear requisições e respostas, `protoId` (sempre 0 para Modbus), `e unitId`, que é o endereço do cliente na rede serial, usado para gateways. Em segundo lugar, é interpretado o *Protocol Data Unit* (PDU), que consiste na carga útil do cabeçalho MBAP, e representa o comando Modbus em si. A ferramenta extrai daqui os campos relevantes para a detecção, que são o `funcCode`, que é o código da função, definindo a ação a ser executada, e os parâmetros da função, que são efetivamente os dados. Dentro dos parâmetros, são armazenados o endereço de inicio da operação, o endereço específico do registrador e os valores a serem escritos.

4.2.2 Arquivo `value_ranges.xml`

O arquivo `value_ranges.xml` é um componente de segurança manual e opcional, que deve ser preenchido previamente à execução do módulo de detecção, por um operador ou engenheiro da planta industrial, com base nos limites operacionais dos equipamentos. Sua função é adicionar uma camada de verificação na segurança do processo, indo além da simples análise do comportamento da rede, e seu uso é detalhado na Seção 4.3.

4.3 Módulo de detecção

Este módulo da ferramenta é responsável por realizar o monitoramento contínuo, e implementa regras de detecção em tempo real. Na sua inicialização, os arquivos `profiles.xml` e `value_ranges.xml` são lidos e seus dados são carregados em dicionários. Os endereços e funções de cada perfil são armazenados como `sets`, garantindo que as verificações de pertencimento sejam operações de alto desempenho, o que é importante para a análise em tempo real.

Após a interpretação dos perfis, o método `scapy.sniff` é novamente utilizado para capturar o tráfego em tempo real. Cada pacote capturado é passado para a função `analyze_packet`, que executa uma análise multicamada sequencial para verificar a conformidade do pacote. A função `trigger_alert` é o ponto de saída centralizado para todas as detecções. Ela utiliza a biblioteca `logging` para escrever um registro detalhado e com *timestamp* de cada alerta, no arquivo `alerts.log`. A seguir, são detalhadas as condições que provocam alarmes pela ferramenta.

4.3.1 Regra de controle de acesso

O endereço IP de origem do pacote `packet[IP].src` é verificado contra as chaves do dicionário `loaded_profiles`. Se o IP não existir, um alerta de IP desconhecido é imediatamente gerado. Isso significa também que, para adicionar novos equipamentos na rede sem que eles provoquem alertas de endereço IP desconhecido, um novo processo de construção de perfil deve ser executado.

4.3.2 Regra de comportamento do protocolo

Se o endereço IP é conhecido, a análise prossegue. O `funcCode` e o endereço do pacote são comparados com os conjuntos de funções e endereços permitidos para aquele IP específico no perfil. Uma divergência em algum desses parâmetros gera um alerta para uso de um código de função incomum ou de um endereço incomum.

4.3.3 Regra de validação semântica

Se o pacote é uma operação de escrita, a análise inspeciona a carga útil. O endereço de escrita e o valor numérico sendo escrito são extraídos. Se aquele endereço possuir uma regra definida no arquivo `value_ranges.xml`, o valor é comparado com os limites mínimos e máximos definidos. Uma violação desses limites dispara um alerta para identificar um valor de escrita fora da faixa.

4.3.4 Regra de anomalia de volume

Independentemente das outras checagens, o `timestamp` de cada pacote é adicionado a uma lista correspondente ao seu endereço IP. A ferramenta mantém uma janela de tempo deslizante de 60 (sessenta) segundos para contar o número de pacotes recentes. Se essa contagem ultrapassar o limite definido no perfil, um alerta de taxa de pacotes anormal é acionado pela ferramenta.

Capítulo 5

Experimentos e Resultados

A ferramenta proposta neste trabalho é avaliada em um ambiente virtual, utilizando a geração de tráfego sintético para criar cenários de rede. Essa metodologia foi escolhida para permitir a criação de conjuntos de dados de linha de base, ou seja, tráfego normal e bem definido, e subsequentemente, a injeção de pacotes anômalos específicos para validar a eficácia de cada método de detecção. O processo é dividido em duas fases: a primeira de construção de perfil com tráfego normal simulado e a segunda de testes com tráfego de ataque sinteticamente construído.

5.1 Arquitetura da geração de perfis

Para construir os perfis base de comunicação dos testes, que configuram os cenários de tráfego típico da rede industrial, foi desenvolvida uma suíte de emulação para gerar tráfego Modbus TCP sintético, em ambiente virtual. A simulação é estruturada como um sistema multiagente, baseado no conceito de *multithreads* [37], em que cada agente, representando um dispositivo cliente na rede industrial, é instanciado e executado em uma *thread* de sistema operacional independente. A partir dessa abordagem são emulados diferentes cenários de comunicação real de operação, na qual múltiplos dispositivos operam de forma assíncrona, com cadências, padrões de comunicação e propósitos distintos, gerando um fluxo de tráfego agregado. Foi gerado um total de 30 (trinta) cenários de rede.

Tabela 5.1: Dados dos cenários simulados para a construção de perfis.

Descrição	**Detalhes**
Cenários típicos	30 conjuntos de perfis simulados.
Clientes simulados	Número variado de clientes, incluindo: 50, 100, 200 e 500.
Funções Modbus simuladas	Leitura: 1 (*Read Coils*), 3 (*Read Holding Registers*), 4 (*Read Input Registers*); e escrita: 6 (*Write Single Register*).
Faixa de endereços utilizada	Endereços discretos e blocos, incluindo os endereços iniciais: 10, 20, 50, 100, 205, 810, 1000 e 2000.
Taxa máxima de pacotes por minuto	Taxas máximas de 10, 50, 100 e 500 pacotes por minuto.

Para garantir a fidelidade da simulação, cada agente é associado a uma persona com um perfil de comportamento. Esses perfis ditam os tipos de requisições Modbus que cada cliente está autorizado a fazer, bem como a frequência de suas operações, levando em consideração os códigos de função, endereços alvo, quantidade de registros e carga útil.

5.2 Arquitetura da simulação de ataques

Após o estabelecimento de um perfil de normalidade, a fase subsequente da avaliação consiste em submeter o módulo de detecção a um conjunto de ataques simulados. O objetivo desta fase é validar a capacidade de cada mecanismo de detecção em identificar desvios específicos do perfil estabelecido, permitindo assim a medição quantitativa da eficácia da ferramenta.

A metodologia para a geração de tráfego de ataque se baseia na criação deliberada de pacotes adversários, conhecido na literatura como *adversarial packet crafting* [21]. Cada cenário de ataque é projetado para violar uma única regra do perfil de segurança, permitindo a validação isolada de uma capacidade de detecção específica. Essa abordagem garante que um alerta gerado possa ser diretamente correlacionado a uma anomalia injetada, permitindo o cálculo de métricas de desempenho (detalhadas na Seção 5.3).

Os ataques simulados incluem: requisição de endereço IP desconhecido, função ou endereço incomum para o cliente, valor de escrita fora da faixa e ataque de negação de serviço distribuído[34]. Para todos os casos mencionados, são usadas técnicas de *spoofing* [35], emulando a ação de atacantes tentando se passar por

clientes legítimos, escondendo sua identidade real, ou sobrecarregando o servidor com tráfego de múltiplas fontes falsas.

5.2.1 Principais bibliotecas utilizadas para as simulações

A construção dos cenários de tráfego habitual e de ataques no ambiente de simulação foi possível a partir de uma série de simulações, fundamentadas em um conjunto de bibliotecas Python.

A biblioteca `scapy` [40] é utilizada para a manipulação e geração de pacotes de rede em ambas as fases da emulação, e permite a construção, dissecção e injeção de tráfego de rede com controle sobre cada camada e campo do protocolo. As bibliotecas `threading` [41] e `subprocessing` [42] são empregadas para instanciar e gerenciar as *threads* de cada cliente simulado, de modo a encapsular o comportamento de cada agente e permitir sua execução paralela e assíncrona. As bibliotecas `random` [44] e `time` [45] são utilizadas para introduzir comportamento estocástico e variável na geração de tráfego, dentro dos parâmetros definidos na Seção 5.1.

A escala de tráfego utilizada nos experimentos é consistente com a observada em trabalhos acadêmicos de referência, utilizados para treinar e validar soluções de segurança para ambientes de controle industrial. Conjuntos de dados como o `SWaT` [62], da Universidade de Tecnologia e Design de Singapura, para uma planta industrial de tratamento de água, ou os gerados por Morris e Gao [63], para estudos de detecção de intrusão em sistemas de controle, frequentemente apresentam taxas de tráfego que, durante eventos normais de pico ou anomalias simuladas, operam em ordem de magnitude análoga aos conjuntos aplicados na validação da ferramenta.

5.3 Métricas de avaliação do desempenho da ferramenta

Esta seção apresenta a fundamentação matemática e a descrição das métricas utilizadas para quantificar a eficiência e a capacidade de detecção da ferramenta. O trabalho desenvolvido consiste intrinsecamente em um classificador de tráfego, que examina o fluxo de pacotes da rede. Dada essa funcionalidade, seu desempenho é mensurado por meio de um conjunto de métricas de desempenho, que são calculadas a partir da tipificação do resultado de cada predição, estruturado conforme a Tabela 5.2.

Tabela 5.2: Classificação do resultado de cada pacote analisado pela ferramenta.

Classificação do resultado	**Descrição**
Verdadeiro Positivo (VP)	Um ataque real é lançado, e a ferramenta gera o alerta correto.
Verdadeiro Negativo (VN)	Tráfego normal é enviado, e a ferramenta corretamente permanece em silêncio.
Falso Positivo (FP)	Tráfego normal é enviado, mas a ferramenta incorretamente gera um alerta.
Falso Negativo (FN)	Um ataque real é lançado, mas a ferramenta falha em detectá-lo.

5.3.1 Acurácia

A acurácia representa a proporção de classificações corretas (anomalias e tráfego normal) em relação ao volume total de eventos analisados.

$$\text{Acurácia} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{VN} + \text{FP} + \text{FN}}$$

Em contextos de segurança de redes, onde o tráfego anômalo é tipicamente raro, a acurácia pode ser uma métrica enganosa, pelo desbalanceamento no volume de anomalias em relação ao volume de pacotes normais. Um sistema que classifica todo o tráfego como normal alcançaria uma acurácia elevada, mas não seria funcionalmente útil, pois falharia em detectar qualquer ameaça. Por essa razão, métricas mais específicas são necessárias para uma avaliação completa.

5.3.2 Precisão

A precisão quantifica a confiabilidade dos alertas gerados. Ela mede a proporção de anomalias corretamente identificadas em relação ao total de eventos que a ferramenta classificou como anômalos.

$$\text{Precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}}$$

Um valor de precisão elevado indica uma baixa incidência de falsos alarmes, ou seja, falsos positivos. Essa métrica é relevante para avaliar a eficiência operacional da ferramenta, pois uma baixa precisão sobrecarregaria os operadores com a investigação de alertas irrelevantes, e diminuiria a confiabilidade do monitoramento.

5.3.3 Sensibilidade

A sensibilidade avalia a capacidade de identificar todas as anomalias que de fato ocorreram. Ela representa a proporção de anomalias reais que foram efetivamente detectadas em relação ao total de anomalias existentes no conjunto de dados.

$$\text{Sensibilidade} = \frac{\text{VP}}{\text{VP} + \text{FN}}$$

Esta métrica corresponde diretamente à uma taxa de detecção de ameaças da ferramenta. Um valor de sensibilidade igual a 1,0 (100%) significaria que nenhuma ameaça real passou despercebida pelo módulo de detecção, o que representa o cenário ideal em termos de cobertura de segurança.

5.3.4 F1 score

O F1 score é a média harmônica entre a precisão e a sensibilidade, servindo como uma métrica que equilibra ambas.

$$F1 = 2 \times \frac{\text{Precisão} \times \text{Sensibilidade}}{\text{Precisão} + \text{Sensibilidade}}$$

Essa medição penaliza sistemas que otimizam excessivamente uma métrica em detrimento da outra. Um F1-Score elevado indica que a ferramenta mantém um balanço saudável entre a confiabilidade de seus alertas (precisão) e sua capacidade de não perder ameaças reais (sensibilidade).

5.4 Análise dos experimentos realizados

Dadas as métricas apresentadas para quantificar a eficiência e a capacidade de detecção nas classificações, em seguida são apresentados os resultados dos experimentos realizados para a avaliação do desempenho da ferramenta.

5.4.1 Ataque de IP desconhecido

Nesse tipo de ataque, um pacote Modbus com conteúdo válido é encapsulado, e definido um endereço de origem que não consta na lista de IPs autorizados. O objetivo é validar a capacidade de identificar tentativas de acesso por fontes não autorizadas em diferentes escalas de volume, baseado nos IPs aprendidos durante a fase de construção de perfis.

A metodologia consistiu na execução de baterias de teste sequenciais para cada um dos cenários, onde cada teste simula um ataque de reconhecimento, variando a quantidade de endereços IP de origem únicos. Foi simulado um número pré-definido

de pacotes, variando entre 1, 100, 1.000, 10.000, 100.000 e 500.000 por conjunto de perfis, em uma janela de tempo de 15 minutos, onde 10% dos pacotes foram forjados com um endereço IP de fora do perfil da rede.

Para a avaliação dos resultados dessa categoria de ataque, a métrica de sensibilidade foi priorizada. Nesse contexto, ela representa a taxa de detecção real dos ataques. A ferramenta alcançou uma sensibilidade de 100% para as baterias com 1 a 10.000 endereços IP, detectando todos os pacotes anômalos. Para os testes na faixa dos 100.000 a 500.000 pacotes, a taxa de detecção permaneceu em aproximadamente 99.9%. Essa pequena quantidade de falhas de identificação de IPs intrusos se deu por perdas de pacotes na rede, ao forçar a geração muito intensa de tráfego durante essas simulações. Pelo critério determinístico, todos os pacotes que chegam com integridade até a ferramenta, serão processados conforme sua presença nos perfis de comunicação habitual.

Destaca-se também que nenhum caso de falso negativo foi levantado durante o monitoramento. Consequentemente, o F1-Score tendeu a 1.00 para os testes em todas as escalas. A Figura 5.1 expõe a sensibilidade encontrada para cada bateria de testes.

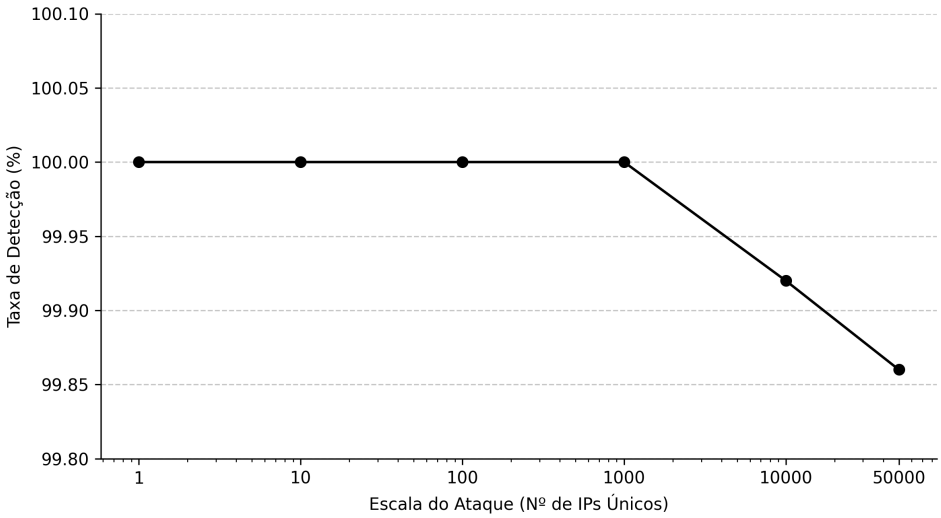

Figura 5.1: Ataque de IP desconhecido; taxa média de detecção (sensibilidade) calculada.

Com a intenção de analisar o desempenho da ferramenta, foi extraída a latência média entre as inserções maliciosas e a geração do alerta, como evidenciado no gráfico da Figura 5.2. O aumento progressivo observado é esperado e revela o comportamento do módulo de detecção sob tráfego intenso. Conforme mais pacotes chegam em um curto período, essa fila aumenta, e, consequentemente, o tempo médio para um pacote ser processado e gerar um alerta também aumenta. Mesmo na situação de maior sobrecarga, com 500.000 pacotes trafegados, os resultados evidenciaram

que o detector é capaz de notificar um operador sobre um evento de segurança em larga escala em menos de meio segundo.

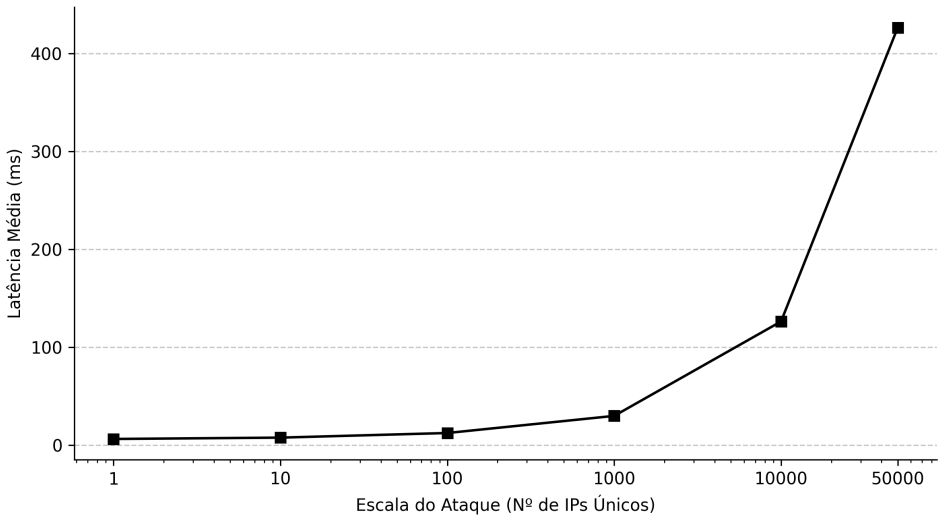

Figura 5.2: Ataque de IP desconhecido; latência média de detecção verificada.

5.4.2 Ataque de função ou endereço incomum

Um IP de um cliente legítimo é usado como a origem da requisição, e a anomalia é injetada na camada de aplicação, onde uma instância é criada com uma função ou endereço que deliberadamente não pertence ao conjunto de operações normais daquele cliente específico, conforme definido no perfil. A intenção desse ataque é aferir a capacidade da ferramenta de detectar um desvio de comportamento de um cliente supostamente confiável, simulando um ataque de abuso de privilégio ou reconhecimento de rede.

Nessa fase de testes, é avaliado principalmente o desempenho da ferramenta em um cenário que intercala um alto volume de tráfego benigno com atividades anômalas específicas. Foram conduzidas simulações na qual o comportamento de um dispositivo conhecido e confiável, em meio às suas operações normais, realiza uma série de ações não autorizadas. Para esse fim, foi gerado um fluxo total de 50.000 pacotes, onde 10% fugiam do perfil construído para determinado dispositivo. Os resultados medidos podem ser observados na Tabela 5.3.

Tabela 5.3: Ataque de função ou endereço incomum; métricas de desempenho.

Métrica	**Resultado aproximado**
Acurácia	100%
Precisão	100%
Sensibilidade	100%
F1 score	1,0

Todos os 5.000 alertas gerados por cenário corresponderam a uma anomalia real injetada. A ausência de falsos positivos demonstra que a ferramenta é altamente confiável e não sobrecarrega o operador com alarmes falsos, um requisito fundamental para ambientes operacionais. O F1 score atingido de 1,0 reforça a excelência do resultado global, visto que a lógica determinística valoriza a confiabilidade dos alarmes que foram de fato gerados.

5.4.3 Ataque de injeção de carga útil

Após validar as defesas de dispositivos desconhecidos e funções ou endereços incomuns, esta seção tem como objetivo aferir a qualidade da ferramenta em proteger a qualidade dos valores numéricos que são propagados. O ataque de injeção de carga útil testa a camada semântica da detecção, que protege o processo industrial contra a inserção de parâmetros operacionais perigosos. Embora toda a detecção da ferramenta seja, em sua essência, baseada em regras, a origem das métricas que determinam o comportamento típico da carga útil vêm de valores estipulados pelo usuário.

Um pacote de escrita é construído, sendo esse uma operação permitida para o IP de origem e o endereço alvo. A anomalia reside exclusivamente na carga útil do pacote, que é preenchida com um valor numericamente fora dos limites de segurança definidos pelo operador. Os resultados obtidos nos testes se mostraram condizentes com os da Seção 5.4.2, o que era esperado pela similaridade no método de detecção do programa para ambos os tipos de alerta. Em comparação, a detecção semântica apresentou uma latência de resposta significativamente superior, que se deve à lógica de processamento do programa, que necessita da leitura iterativa do arquivo de limites operacionais do sistema. O processo da extração e interpretação da carga útil do pacote, acumulada num fluxo de comunicação muito intenso, demanda maior poder computacional da máquina em que está rodando a ferramenta.

Foi conduzida uma bateria de testes escalonados para cada um dos 30 cenários propostos, simulando um tráfego misto direcionado a um registrador, protegido por uma regra de valor pré-definida no arquivo de limites operacionais do sistema. Para cada etapa do teste, com 100, 1.000, 10.000, 100.000 e 500.000 pacotes totais, foi gerado um fluxo de comunicação contínuo composto por 40% de pacotes de ataque e 60% de pacotes seguros.

Os resultados dos testes foram positivos, como evidenciado nas métricas de classificação da Tabela 5.4. No entanto, apesar do desempenho classificatório sem erros, estima-se que a ferramenta tenha uma eficiência média de identificação similar ao dos ataques de IP desconhecido. Pela estrutura da biblioteca scapy, as simulações com manipulações semânticas exigiram maior poder computacional de processamento,

o que causou intervalos ocasionais com interrupções de até cerca de 500 milissegundos entre o envio de cada pacote. Portanto, as limitações computacionais dos simuladores foram um obstáculo para gerar baterias igualmente rápidas, como na Seção 5.4.2.

Tabela 5.4: Ataque de *payload*; métricas de classificação.

Métrica	**Resultado aproximado**
Acurácia	100%
Precisão	100%
Sensibilidade	100%
F1 score	1,0

Já em relação ao tempo médio de detecção, observou-se um aumento quase exponencial na latência média na Tabela 5.5, à medida que o volume total de pacotes na bateria de teste aumentou. A latência medida representa o tempo total desde a geração do pacote no emulador do atacante até a geração do alerta. Este comportamento indica o efeito do enfileiramento de pacotes na parte da inspeção, visto que sua interpretação é realizada sequencialmente. A análise do gráfico da Figura 5.3 corrobora essa justificativa, dado o crescimento não linear dos tempos de detecção. Ainda assim, em um cenário real da indústria, o *delay* de apenas alguns segundos pode ser considerado satisfatório para o operador ou um sistema de segurança ativa agir em resposta ao alerta levantado pela ferramenta.

Tabela 5.5: Ataque de *payload*; latência de detecção.

Total de Pacotes na Bateria	**Latência Média (ms)**
100	26 ms
1.000	31 ms
10.000	144 ms
100.000	891 ms
500.000	3.172 ms

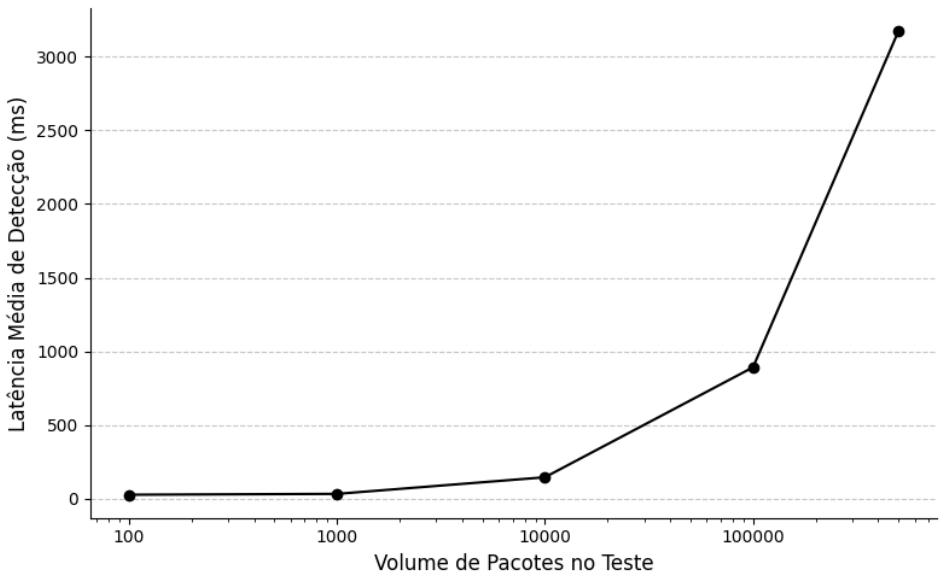

Figura 5.3: Ataque de *payload*; crescimento na latência de detecção.

5.4.4 Ataque de negação de serviço distribuído (DDoS)

Para avaliar a estabilidade e o desempenho do detector sob uma inundação de tráfego, são instanciados múltiplos fluxos de pacotes concorrentes. Com isso, uma sobrecarga de pacotes é transmitida, simulando um ataque de múltiplas fontes autorizadas. Assim, é possível analisar a capacidade de processamento do detector e a eficácia da detecção de uma negação de serviço originada por dispositivos legítimos e comprometidos dentro da própria rede. Nessa situação, todas as outras regras de detecção são respeitadas, e a única anomalia é o padrão temporal.

A métrica principal de sucesso para este teste é o tempo levado para a ferramenta emitir o primeiro alerta, ou seja, quanto tempo o detector leva para perceber que a taxa de comunicação normal foi violada e gerar o alerta de DDoS. Essa etapa da validação consistiu em submeter o programa a um teste de estresse, dentro dos limites dos mecanismos de emulação disponíveis.

Para analisar o comportamento da ferramenta sob diferentes formas de ataque, foram definidos duas configurações distintas, ambas com uma carga total de tráfego idêntica, de 500 pacotes por segundo, e resultando em um total de 60.000 pacotes enviados durante o período de teste de cada cenário. A configuração A utilizou uma configuração de 50 *threads* enviando pacotes a uma taxa de 10 pacotes por segundo cada, de modo que o *flood* de ataques seja mais distribuído entre os atacantes. Já a configuração B emulou, para cada cenário industrial, 5 *threads* enviando pacotes a uma taxa de 100 pacotes por segundo cada, sendo assim uma distribuição mais concentrada de ameaças feitas por atacante.

O primeiro ponto a ser destacado pelas baterias de teste, foi de que a ferramenta

identificou 100% dos casos de taxa de pacotes do perfil excedida, emitindo alertas de ataque. Além disso, as configurações alcançaram resultados bem similares no aspecto do desempenho, evidenciando que a distribuição maior dos atacantes em *threads* não influenciou na latência das respostas. Em média, o tempo de resposta ao primeiro alerta, na configuração A, foi de 61,69 segundos (evidenciado no gráfico da Figura 5.4), ou seja, 1,69 segundos após os 60 segundos da janela de análise da ferramenta. Já na configuração B, o tempo de resposta ao primeiro alerta foi de aproximadamente 61,31 segundos (evidenciado no gráfico da Figura 5.5), que significa um *delay* de 1,31 segundos da janela de supervisão. O resultado comprova que a ferramenta não se baseia apenas em assinaturas estáticas, mas é capaz de policiar o ritmo da rede e protegê-la contra a instrumentalização de seus próprios ativos legítimos.

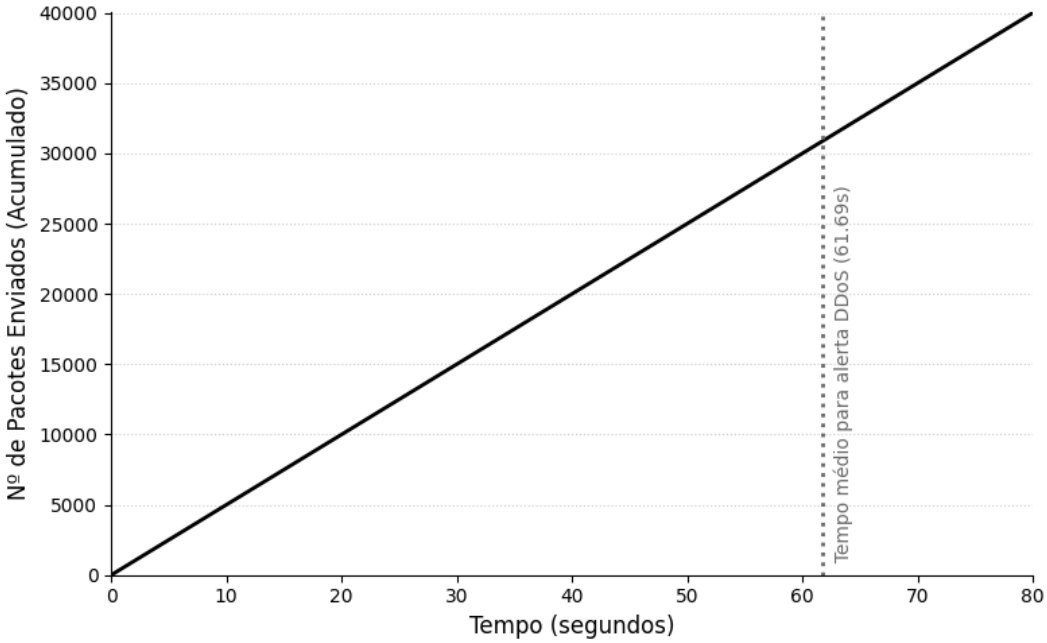

Figura 5.4: Configuração A: tempo para o primeiro alerta.

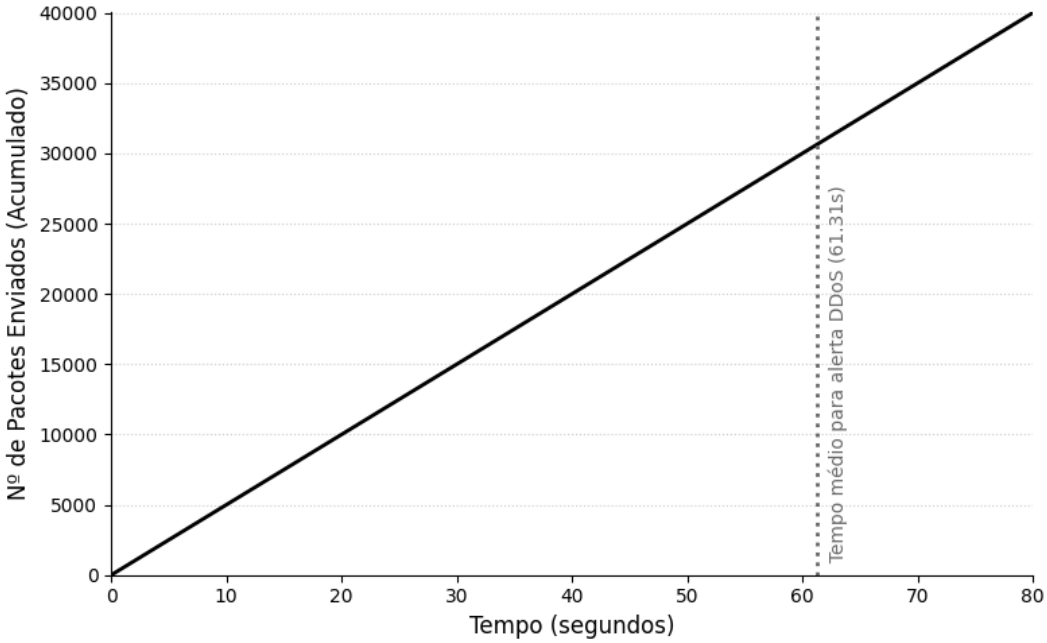

Figura 5.5: Configuração B: tempo para o primeiro alerta.

Capítulo 6

Conclusão e Trabalhos Futuros

O trabalho desenvolvido abordou uma lacuna de segurança relacionada ao protocolo Modbus TCP, uma tecnologia fundamental para a comunicação em redes de automação industrial, mas que, por sua concepção, não foi projetada para resistir às ameaças cibernéticas contemporâneas. Em resposta à carência de soluções abrangentes na literatura, foi proposta, desenvolvida e validada uma ferramenta de detecção de anomalias com uma abordagem multicamada.

A principal contribuição do projeto reside na criação de uma ferramenta de monitoramento que integra, de forma coesa, a análise de múltiplos parâmetros da comunicação para construir uma visão ampla da segurança da rede. O projeto demonstrou a capacidade de, primeiramente, construir perfis de comportamento para cada nó da rede através de um módulo de construção de perfis e, posteriormente, detectar em tempo real um espectro diversificado de anomalias. As camadas de detecção implementadas foram capazes de identificar acessos por fontes não autorizadas, desvios do padrão de comunicação de dispositivos legítimos, manipulações de dados de processo e ataques de sobrecarga de tráfego.

A validação realizada por meio de simulações controladas confirmou a eficácia da ferramenta, baseado em critérios gerais utilizados para avaliar o desempenho de classificadores de dados, com destaque à sensibilidade e ao F1 score. O projeto alcançou uma taxa de detecção satisfatória mesmo em testes de estresse com milhares de eventos anômalos, provando a validade da utilização de critérios determinísticos e regras estáticas no que tange a supervisão de redes industriais. Conclui-se, portanto, que o trabalho proposto apresenta uma solução viável para aumentar a segurança e a confiabilidade de redes industriais que dependem do protocolo Modbus TCP.

6.1 Trabalhos futuros

Como uma alternativa para dar continuidade à pesquisa e ao desenvolvimento da ferramenta, a diretriz de regras determinísticas poderia ser combinada a um

modelo de aprendizado de máquina. Enquanto a metodologia atual detecta violações de regras conhecidas, um modelo não supervisionado poderia ser treinado com os dados de carga útil para detectar desvios estatísticos sutis nos valores do processo, identificando anomalias novas e desconhecidas que não violam as regras de valor mínimo ou máximo, mas que fogem do padrão operacional aprendido. Além disso, a arquitetura da ferramenta serve como um *framework* que pode ser estendido para suportar outros protocolos de rede mais modernos, como OPC UA ou profinet, ampliando o escopo de proteção da solução. Para uma aplicação em ambientes corporativos, a ferramenta poderia ser integrada a um sistema de gerenciamento de eventos, através da exportação de alertas via syslog, ou por meio de bases de dados SQL.

Referências Bibliográficas

[1] BRANQUINHO, Marcelo Ayres; SEIDL, Jan; MORAES, Leonardo Cardoso de; BRANQUINHO, Thiago Braga; JUNIOR, Jarcy de Azevedo. Segurança de Automação Industrial e SCADA. 1. ed. Brasil: Elsevier, 2014.

[2] SANDERS, Chris. Análise de pacotes na prática. Brasil: Novatec, 2017.

[3] NORTHCUTT, S. Network intrusion detection: an analyst's handbook. Indianapolis: New Riders, 1999. (New Riders professional library). ISBN 9780735708686. Disponível em: `https://books.google.com.br/books?id=kAS9QgAACAAJ`.

[4] CHIAÇA, Matheus Ruiz Munhoz; ABRANTES, Maria Luiza Marques de. *Agilidade dos processos na indústria 4.0.* Centro de Pós-Graduação Oswaldo Cruz. Disponível em: `https://www.oswaldocruz.br/revista_academica/content/pdf/Edicao26_Matheus_Ruiz_Munhoz_Chiaca.pdf`.

[5] X. Fan, K. Fan, Y. Wang and R. Zhou, "Overview of cyber-security of industrial control system,"2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC), Shanghai, China, 2015, pp. 1-7. Disponível em: `https://doi.org/10.1109/SSIC.2015.7245324`.

[6] Z. Drias, A. Serhrouchni and O. Vogel, "Analysis of cyber security for industrial control systems,"2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC), Shanghai, China, 2015, pp. 1-8. Disponível em: `https://doi.org/10.1109/SSIC.2015.7245330`.

[7] EREZ, Noam; WOOL, Avishai. Control variable classification, modeling and anomaly detection in Modbus/TCP SCADA systems. International Journal of Critical Infrastructure Protection, v. 10, p. 59-70, 2015. Disponível em: `https://doi.org/10.1016/j.ijcip.2015.05.001`.

[8] SUCHORAB, J.; PLAMOWSKI, S.; AWRYCZUK, M. Anomaly detection system for Modbus data based on an open-source tool. Computers and Security, v. 157, p. 104572, 2025. Disponível em: https://doi.org/10.1016/j.cose.2025.104572.

[9] ALANAZI, M.; MAHMOOD, A.; CHOWDHURY, M. J. M. SCADA vulnerabilities and attacks: A review of the state-of-the-art and open issues. Computers and Security, v. 125, p. 103028, 2023. Disponível em: https://doi.org/10.1016/j.cose.2022.103028.

[10] Behrouz A. Forouzan and Sophia Chung Fegan. 2002. TCP/IP Protocol Suite (2nd. ed.). McGraw-Hill Higher Education.

[11] SHOCH, John F.; HUPP, Jon A. Measured performance of an Ethernet local network. Communications of the ACM, New York, v. 23, n. 12, p. 711-721, Dec. 1980. DOI: 10.1145/359038.359044. Disponível em: https://doi.org/10.1145/359038.359044.

[12] WALRAND, Jean; VARAIYA, Pravin. Packet-Switched Networks. In: WALRAND, Jean; VARAIYA, Pravin. High-Performance Communication Networks. 2. ed. San Francisco: Morgan Kaufmann, 2000. p. 103-154. ISBN 978-1-55860-574-9. Disponível em: https://www.sciencedirect.com/science/article/pii/B9780080508030500083.

[13] KUROSE, J. F.; ROSS, K. W. Redes de computadores e a Internet. 5. ed. São Paulo: Pearson, 2012.

[14] COSTA, D. P. M. Análise e modelagem de sistemas de inspeção profunda de pacotes em redes de alta velocidade. 2020. Dissertação de Mestrado, Universidade Federal de Pernambuco, Recife, 2020. Disponível em: https://repositorio.ufpe.br/bitstream/123456789/25608/1/DISSERTA%C3%87%C3%83O%20Diego%20Pereira%20Martins%20da%20Costa.pdf.

[15] FORTINET. O que é inspeção profunda de pacotes (DPI)? 2025. Disponível em: https://www.fortinet.com/br/resources/cyberglossary/dpi-deep-packet-inspection.

[16] SANTOS, M. R. P. Classificação de tráfego e dispositivos de IoT através do fluxo de rede e inspeção profunda de pacotes. 2018. Dissertação de mestrado, Universidade Federal do Ceará, Fortaleza, 2018. Disponível em: https://repositorio.ufc.br/bitstream/riufc/37849/3/2018_dis_mrpsantos.pdf.

[17] Cordeiro, J. A. F. (2019). Estudo dos principais protocolos de redes industriais utilizadas no Brasil: AS-I, MODBUS e PROFIBUS (Monografia). Universidade Federal de Ouro Preto. Disponível em: http://www.monografias.ufop.br/handle/35400000/2412.

[18] LIN, Zhihong; PEARSON, Stephanie et al. An inside look at industrial Ethernet communication protocols. Texas Instruments, White Paper, 2013. Disponível em: https://www.ti.com/lit/wp/spry254b/spry254b.pdf.

[19] H. -C. Chang, C. -Y. Lin, D. -J. Liao and T. -M. Koo, "The Modbus Protocol Vulnerability Test in Industrial Control Systems,"2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Chongqing, China, 2020, pp. 375-378, doi: https://doi.org/10.1109/CyberC49757.2020.00070.

[20] RASHID, Marifat; SINGH, Yashwant; MANZOOR, Sheikh Imroza. Methodology for Assessing Existing Vulnerabilities in Modbus Protocol. Procedia Computer Science, v. 259, p. 1983-1993, 2025. ISSN 1877-0509. DOI: 10.1016/j.procs.2025.04.154. Disponível em: https://www.sciencedirect.com/science/article/pii/S1877050925012566.

[21] PERALES GÓMEZ, Ángel Luis et al. Crafting adversarial samples for anomaly detectors in industrial control systems. Procedia Computer Science, v. 184, p. 573-580, 2021. ISSN 1877-0509. Disponível em: https://doi.org/10.1016/j.procs.2021.03.072.

[22] SADDAM, R. A. et al. Sniffing and spoofing in computer security. International Journal of Integrative Sciences (IJIS), v. 2, n. 6, p. 869-874, 2023. Disponível em: https://doi.org/10.55927/ijis.v2i6.4528.

[23] ANDRESS, Jason. Network security. In: ANDRESS, Jason. The basics of information security. Syngress, 2011. p. 115-130. Disponível em: https://doi.org/10.1016/B978-1-59749-653-7.00008-6.

[24] GTA UFRJ. Ethernet Industrial. Disponível em: https://www.gta.ufrj.br/ensino/eel878/redes1-2023-1/trabalhos/Grupo09/.

[25] IEEE. IEEE Standard for Ethernet. *IEEE Std 802.3-2022 (Revision of IEEE Std 802.3-2018)*, p. 1-7025, 29 jul. 2022. Disponível em: https://doi.org/10.1109/IEEESTD.2022.9844436.

[26] CISCO. Understanding Local Area Netowrks (LAN). Disponível em: https://www.cisco.com/c/en/us/products/switches/what-is-a-lan-local-area-network.html.

[27] AMAZON WEB SERVICES. O que é o modelo OSI. Disponível em: https://aws.amazon.com/pt/what-is/osi-model/.

[28] PROSOFT TECHNOLOGY. Introduction to Modbus TCP/IP. Disponível em: https://www.prosoft-technology.com/kb/assets/intro_modbustcp.pdf.

[29] HILSCHER. Master-slave model. Disponível em: https://www.hilscher.com/de/service-support/glossary/master-slave-model.

[30] IBM. Master/slave model. Disponível em: https://www.ibm.com/docs/en/aix/7.1.0?topic=models-masterslave-model.

[31] W AND T. Modbus TCP: standard protocol for automation technology. Disponível em: https://www.wut.de/e-57www-04-apus-000.php.

[32] MODBUS ORGANIZATION. Modbus messaging implementation guide. Version 1.0b, 2012. Disponível em: https://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf.

[33] PALO ALTO NETWORKS. Understanding Denial of Service (DoS) Attack. Disponível em: https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos.

[34] IBM. DDoS (Distributed Denial of Service). Disponível em: https://www.ibm.com/think/topics/ddos.

[35] CYBLE. Spoofing in cyber security. Disponível em: https://cyble.com/knowledge-hub/what-is-spoofing-in-cyber-security/.

[36] CHAINLINK. Replay Attacks Explained. Disponível em: https://chain.link/education-hub/replay-attack.

[37] ORACLE. Multithreaded Programming Guide. 2011. Disponível em: https://docs.oracle.com/cd/E18752_01/html/816-5137/mtintro-25092.html.

[38] IBM. TCP/IP concepts=. Disponível em: https://www.ibm.com/docs/en/zos/3.1.0?topic=concepts-tcpip.

[39] COSTA, Guilherme. modbus-anomaly-detector. 2025. Disponível em: https://github.com/guiga07/modbus-anomaly-detector.

[40] BIONDI, P. Scapy: Scapy Community. Disponível em: https://scapy.net.

[41] PYTHON SOFTWARE FOUNDATION. threading Thread-based parallelism. Disponível em: https://docs.python.org/3/library/threading.html.

[42] PYTHON SOFTWARE FOUNDATION. subprocess Subprocess management. Disponível em: https://docs.python.org/3/library/subprocess.html.

[43] NUMPY. NumPy: The fundamental package for scientific computing with Python. Disponível em: https://numpy.org/.

[44] PYTHON SOFTWARE FOUNDATION. random Generate pseudo-random numbers. Disponível em: https://docs.python.org/3/library/random.html.

[45] PYTHON SOFTWARE FOUNDATION. time Time access and conversions. Disponível em: https://docs.python.org/3/library/time.html.

[46] PYTHON SOFTWARE FOUNDATION. os. Disponível em: https://docs.python.org/3/library/os.html.

[47] PYTHON SOFTWARE FOUNDATION. xml.etree.ElementTree. Disponível em: https://docs.python.org/3/library/xml.etree.elementtree.html.

[48] PYTHON SOFTWARE FOUNDATION. logging. Disponível em: https://docs.python.org/3/library/logging.html.

[49] PYTHON SOFTWARE FOUNDATION. collections.defaultdict. Disponível em: https://docs.python.org/3/library/collections.html#collections.defaultdict.

[50] PYTHON SOFTWARE FOUNDATION. xml.dom.minidom. Disponível em: https://docs.python.org/3/library/xml.dom.minidom.html.

[51] PANDAS. Pandas Documentation. Disponível em: https://pandas.pydata.org/docs/.

[52] PYTHON SOFTWARE FOUNDATION. datetime Basic date and time types. Disponível em: https://docs.python.org/3/library/datetime.html.

[53] PYTHON SOFTWARE FOUNDATION. re Regular expression operations. Disponível em: https://docs.python.org/3/library/re.html.

[54] SCIKIT-LEARN. F1 score. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html.

[55] MODICON. Modbus Protocol Reference Guide. North Andover, MA: Schneider Electric, 2012. Disponível em: https://www.modbus.org/docs/PI_MBUS_300.pdf.

[56] OPC UA. Arquitetura Unificada. Disponível em: https://opcfoundation.org/about/opc-technologies/opc-ua/.

[57] MDIS. Offshore Oil and Gas OPC UA information model for MCS-DCS. Disponível em: https://opcfoundation.org/markets-collaboration/mdis/.

[58] PROFINET. Padrão industrial Siemens. Disponível em: https://www.siemens.com/br/pt/produtos/automacao/comunicacao-industrial/profinet.html.

[59] PROFINET. Ethernet standard for the process industry. Disponível em: https://www.siemens.com/global/en/products/automation/industrial-communication/profinet/process-industry.html.

[60] AMAZON WEB SERVICES. O que é IPSec? [S.l.], 2024. Disponível em: https://aws.amazon.com/pt/what-is/ipsec/.

[61] CLOUDFLARE. O que é TLS (Transport Layer Security)? [S.l.], 2024. Disponível em: https://www.cloudflare.com/pt-br/learning/ssl/transport-layer-security-tls/.

[62] MATHUR, Aditya; TIPPENHAUER, Nils Ole. SWaT: A Water Treatment Testbed for Research and Training on ICS Security. In: WORKSHOP ON CYBER-PHYSICAL SYSTEMS FOR SMART WATER NETWORKS (CYSWATER), 2016. Disponível em: https://doi.org/10.1109/CySWater.2016.7469060.

[63] MORRIS, Thomas; GAO, Wei. Industrial control system traffic data sets for intrusion detection research. In: INTERNATIONAL CONFERENCE ON CRITICAL INFRASTRUCTURE PROTECTION (ICCIP), 2014, p. 65-78. Disponível em: https://doi.org/10.1007/978-3-662-45355-1_5.