UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Engenharias
Curso de Bacharelado em Engenharia de Controle e Automação

Trabalho de Conclusão de Curso

Análise de segurança em protocolos de comunicação IoT

Eduardo Santos da Veiga

Pelotas, 2023

Eduardo Santos da Veiga

Análise de segurança em protocolos de comunicação IoT

Trabalho de Conclusão de Curso apresentado ao Centro de Engenharias da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Marlon Soares Sigales

Pelotas, 2023

Insira AQUI a ficha catalográfica
(solicite em http://sisbi.ufpel.edu.br/?p=reqFicha)

Eduardo Santos da Veiga

Análise de segurança em protocolos de comunicação IoT

Trabalho de Conclusão de Curso aprovado, como requisito parcial, para obtenção do grau de Bacharel em Engenharia de Controle e Automação, Centro de Engenharias, Universidade Federal de Pelotas.

Data da Defesa: 30 de fevereiro de 2019

Banca Examinadora:

Prof. Dr. Marilton Sanchotene de Aguiar (orientador)
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Paulo Roberto Ferreira Jr.
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Ricardo Matsumura Araujo
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Luciano da Silva Pinto
Doutor em Biotecnologia pela Universidade Federal de Pelotas.

Dedico. . .

AGRADECIMENTOS

Agradeço ao mozy por ser um mozy

Só sei que nada sei.
— SÓCRATES

RESUMO

VEIGA, Eduardo Santos da. **Análise de segurança em protocolos de comunicação IoT**. Orientador: Marlon Soares Sigales. 2023. 42 f. Trabalho de Conclusão de Curso (Engenharia de Controle e Automação) – Centro de Engenharias, Universidade Federal de Pelotas, Pelotas, 2023.

O termo Internet das Coisas (IoT) foi concebido em 1999 por Kevin Ashton para ilustrar o poder de conectar etiquetas de RFID (Radio Frequency Identification - *Identificação por Radiofrequência*) na internet para gerenciamento de cadeia de suprimentos (ASHTON, 2009). Desde então, várias definições evoluíram baseadas nas definições das tecnologias em voga no momento, porém, todas falando de uma rede feita de máquinas, para máquinas - a rede *machine-to-machine* (M2M). Devido a heterogeneidade de dispositivos numa rede IoT, se faz necessário uma estrutura bem desenvolvida, capaz de prover a segurança exigida para os dados recolhidos de sensores.

Uma solução ampla de segurança é de difícil implementação na camada de rede e percepção definida na arquitetura descrita por (ZHANG; SUN; CHENG, 2012), uma vez que os dispositivos instalados nessa camada são de baixo poder computacional, incapazes de prover soluções de criptografia. É justamente nessa camada onde o maior risco à infraestrutura de uma solução IoT se apresenta, uma vez que essa baixa potência computacional é acompanhada de protocolos de comunicação simples e vulneráveis.

Palavras-chave: Cybersegurança. Internet-das-coisas. Protocolos-comunicação. SCADA.

ABSTRACT

VEIGA, Eduardo Santos da. **Security Analysis in IoT Communication Protocols**. Advisor: Marlon Soares Sigales. 2023. 42 f. Undergraduate Thesis (Automation and Control Engineering) – Engineering Centre, Federal University of Pelotas, Pelotas, 2023.

The Internet of Things as a term was first coined in 1999 by Kevin Ashton to better illustrate the potential in connecting several RFID tags on the internet for supply chain management (ASHTON, 2009). Since then, several definitions of the Internet of Things evolved based on the latest technology en vogue. However, all of those definitions speak of a network made by machines, for machines - the machine-to-machine network (M2M). Due to the calleidoscope of different devices one can use to create an IoT network, there is a need for a well understood framework capable of providing the security demanded by the usually sensitive nature of the data harvested by the sensors.

An all-encompassing safety solution is hard to implement in the network and perception layers as defined by the framework described by (ZHANG; SUN; CHENG, 2012), since the devices on this layer are usually weak on computing power, which hinders their ability to include cryptography and secure pipes for communications. Its exactly on this junction where the biggest risk to the infrastructure lies, since the simplicity of the end-devices usually means simplistic and vulnerable communications protocols.

Keywords: Cybersecurity. Internet-of-things. Communication-protocols. SCADA.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
SCADA	Supervisory Control and Data Acquisition - *Controle Supervisório e Aquisição de dados*
IoT	Internet of Things - *Internet das Coisas*
ICMP	Internet Control Message Protocol - *Protocolo para Mensagens de Controle via Internet*
TCP	Transmission Control Protocol - *Protocolo de controle de transmissão*
UDP	User Datagram Protocol - *Protocolo de datagramas de usuário*
VPN	Virtual Private Network - *Rede Privada Virtual*
UTM	Universal Threat Manager - *Gerenciador Universal de Ameaças*
SSH	Secure Shell - *Túnel de Comunicação Segura*
ARP	Address Resolution Protocol - *Protocolo de Resolução de Endereços*
DNS	Domain Name Server - *Servidor de Nome de Dominio*
IIoT	Industrial Internet of Things - *Internet das coisas Industriais*
IoMT	Internet of Medical Things - *Internet das coisas médicas*
IoE	Internet of Everything - *Internet de Tudo*
M2M	Machine-to-Machine - *Comunicação máquina a máquina*
LPWAN	Low-Power Wide-Area Network - *Rede de Grande Área e Baixa Potência*
LoRaWAN	Long-Range Wide Area Network - *Rede Grande Área e Longo Alcance*
NBIoT	Narrowband Internet of Things - *Internet das coisas em banda estreita*
BLE	Bluetooth Low Energy - *Protocolo Bluetooth de Baixa Energia*
RFID	Radio Frequency Identification - *Identificação por Radiofrequência*
NFC	Near Field Communication - *Comunicação em campo próximo*
WLAN	Wireless Local Area Network - *Rede Local sem-fio*
MQTT	Message Queuing Telemetry Transport - *Transporte Enfileirado de Mensagens de Telemetria*
CoAP	Constrained Application Protocol - *Protocolo de Aplicação Limitada*

LAN	Local Area Network - *Rede local*
WAN	Wide Area Network - *Rede ampla*
DHCP	Dynamic Host Configuration Protocol - *Protocolo de configuração de hóspede dinâmico*

SUMÁRIO

1 INTRODUÇÃO

Com a alvorada da era da informação, nossa existência tem sido cada vez mais coberta por um vapor de dados. A popularização dos smartphones desde o lançamento do iPhone em 2006 e a ampla adoção do equipamento em meados de 2013, somado aos avanços em miniaturização de sensores e potência computacional dos últimos 20 anos nos fizeram mergulhar num mar de informação sem precedentes. Com a ubiquidade desses dispositivos e sensores, o estado de interconexão se espalha tal qual uma teia de uma aranha em uma casa abandonada.

Essa tecnologia é amplamente explorada na indústria muito antes de estar disponível para o usuário final, sob o conceito de SCADA (traduzindo livremente, controle supervisório e aquisição de dados), que é pesquisado e melhorado desde a década de 1970, quando os primeiros sistemas de controle computadorizados eram usados para monitorar processos industriais. A coalescência concomitante da computação industrial e doméstica permite amplos modelos de sistemas, independente do contexto.

E é exatamente nesse cenário de informação ubíqua e interconectividade extensiva que se oriunda a Internet das Coisas (*IoT - Internet of Things*), junto com a necessidade de proteger esses dispositivos e seus dados, seja com a miríade de informações disponíveis em um *smartphone* ou em um sistema de automação doméstica. E de maneira mais crítica, a produção ou segurança nacional implicada em uma planta industrial é de extrema prioridade de proteção.

1.1 Stuxnet

Para ilustrar a importância da proteção de plantas industriais e sistemas SCADA, pode-se analisar a história de um dos vírus do tipo *worm* mais potentes da história recente - o Stuxnet.

O stuxnet era um *worm* segmentado (KASPERSKY, 2014), que era espalhado em dispositivos USB (pen-drives) e computadores Windows. O vírus buscava em cada computador infectado por sinal de software do Siemens Step 7, utilizado por computadores industriais agindo como CLPs (Computador Lógico-Programável) controlando

e monitorando equipamentos eletromecânicos. Depois de encontrar um CLP, o **worm** atualizava seu código pela internet e começava a enviar instruções mal-formadas para o equipamento eletromecânico (centrífugas nucleares para enriquecimento de urânio) e ao mesmo tempo, respostas falsas ao controlador principal. Qualquer pessoa monitorando o equipamento não teria ideia de que algo de errado estaria acontecendo até que o equipamento começasse a se auto-destruir.

> Antes do Stuxnet, ninguém pensava na cibersegurança dos sistemas industriais. Se presumia que o isolamento das redes industriais era uma medida suficiente para preservar a segurança empresarial. Através do ataque à máquinas desconectadas, os criadores do *worm* inauguraram uma nova era na segurança da informação. A importância do Stuxnet pode ser comparado apenas com o Morris Worm , criado em 1988.
>
> - Kaspersky, 2014

1.2 A ponta da lança

Os estudos desenvolvidos por Schaffers et al. (2011) sugerem várias aplicações da internet das coisas, como redes inteligentes, aprimoramentos na gestão de energia, telemedicina, veículos inteligentes, etc. Esses campos se dispõe num paradigma urbano de IoT para melhorar os serviços públicos de manutenção, vigilância e segurança. De acordo com os autores, o emprego de IoT urbano tem o potencial de otimizar o gerenciamento de serviço público, principalmente através da análise da quantidade de dados para promover governança transparente e eficiente.

A evolução sistemática do conceito de IoT e o acompanhamento em vários domínios desenvolvido por Ibarra-esquer et al. (2017) nos mostra como idealmente, é essencial ter uma infraestrutura composta de dispositivos de sensoriamento, enlaces de comunicação e aplicações destinadas ao usuário final para ter um ambiente inteligente. No entanto, não é de forma alguma essencial dispor de toda a infraestrutura no começo (LAYA; BRATU; MARKENDAHL, 2013). Uma solução única não pode acomodar as aplicações de um campo heterogêneo como a IoT, não obstante a falta de serviços de segurança e ampla gama de dispositivos, bem como as preocupações com privacidade do usuário final.

Com a iminente ubiquidade de uma rede de máquinas construída com os conceitos de IoT, faz-se necessário um olhar com cuidado sobre a segurança empregada, devido a natureza sensível das informações que trafegam nessa rede. Assim, será feito uma extensiva revisão bibliográfica de todos os conceitos importantes para essa análise, seguido de um mapeamento dos protocolos mais utilizados e uma análise de segurança através de várias técnicas de invasão. Por fim, os resultados dessas in-

formações serão discutidos na tentativa de encontrar um paradigma seguro para uma rede IoT e SCADA.

2 REVISÃO BIBLIOGRÁFICA

2.1 Redes de Computador

2.1.1 Modelo OSI

Segundo Kurose; Ross (2022), o modelo OSI é um modelo conceitual, desenvolvido pela Organização de Padronização Internacional (ISO) que permite sistemas diversos se comunicarem usando protocolos. Foi o primeiro modelo padrão para comunicação em rede, adotado por todas as grandes companhias de computação e telecomunicação desde a década de 1980.

Apesar da internet moderna não ser estritamente baseada no modelo OSI, e sim no modelo TCP/IP mais simples, Kurose; Ross (2022) mostra que o modelo OSI ainda é amplamente usado, uma vez que permite visualizar como redes de computador operam, e serve como um guia para auxiliar isolar e diagnosticar condições anormais.

De acordo com Imperva (2016), as vantagens do modelo OSI são, entre outras:

- Determinar o hardware e software necessários para construir a rede;
- Entender e comunicar o processo seguido por componentes comunicando através de uma rede;
- Realizar diagnósticos, identificando qual camada de rede está causando problemas;
- Definir qual partes da rede um produto vai agir.

2.1.1.1 Camada 1: Física

Essa camada inclui o equipamento físico envolvido na transferência de dados, como cabos, *switches* de rede e fibras óticas (CLOUDFLARE, 2018). É aqui que os dados são convertidos em uma transmissão de dígitos binários.

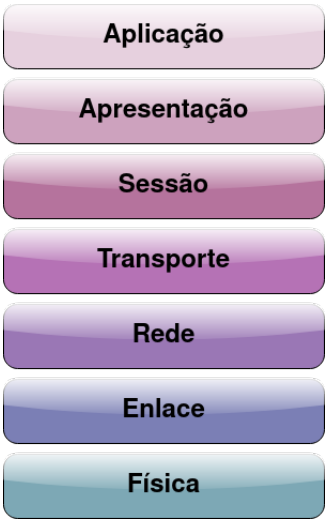


Figura 1 – Visualização do modelo OSI, modificado de (KUROSE; ROSS, 2022)

2.1.1.2 Camada 2: Enlace

A camada de enlace estabelece e termina a conexão entre dois nós fisicamente conectados numa rede. A camada de rede é dividida em duas partes (IMPERVA, 2016):

- Controle Lógico de Enlace (LLC), responsável pela sincronia de quadros, realiza correção de erros e identifica protocolos de rede;
- Controle de Acesso de Meio (MAC), responsável por conectar e endereçar dispositivos e definir permissões para transmitir e receber dados

2.1.1.3 Camada 3: Rede

De acordo com Cloudflare (2018), a camada de rede é responsável por facilitar a comunicação entre duas redes diferentes. Se dois dispositivos em comunicação estiverem na mesma rede, a camada então é redundante. A facilitação de comunicação ocorre Cloudflare (2018) pela fragmentação de segmentos da camada de transporte em unidades menores, chamadas de pacotes, no dispositivo remetente, e remontando esses pacotes no dispositivo receptor. A camada de rede também é responsável pelo roteamento de dados da origem, utilizando endereços de rede (tipicamente, endereços IP), até o destino.

2.1.1.4 Camada 4: Transporte

A camada de transporte é responsável pela comunicação ponta a ponta entre os dois dispositivos. A explicação apresentada em Kurose; Ross (2022) nos mostra que os protocolos de camada de transporte provém *comunicação lógica* entre processos de aplicações rodando em computadores diferentes numa mesma rede. Ou seja, como se os diferentes computadores rodando os processos estivessem diretamente

conectados, independente da distância física ou tipo de conexão intermediária, que é tratada pelas camadas inferiores

2.1.1.5 Camada 5: Sessão

A camada de sessão cria canais de comunicação (sessões) entre dispositivos. Conforme mostrado por Imperva (2016), é necessário que as sessões se mantenham abertas enquanto dados estão sendo transferidos. A camada de sessão também é responsável por sincronizar os dados por pedaços. O exemplo mostrado por Cloudflare (2018) nos ajuda a ilustrar a utilidade: se um arquivo de 100 megabytes está sendo transferido, a camada de sessão pode criar pedaços de 5 megabytes. Assim, se a conexão for interrompida durante a transferência, ela pode ser reiniciada no último pedaço.

2.1.1.6 Camada 6: Apresentação

Como descrito em Cloudflare (2018), dois dispositivos em comunicação devem estar usando métodos de codificação diferentes, portanto camada de apresentação, de acordo com Imperva (2016), prepara os dados para a camada de aplicação e define como dois dispositivos devem encodificar, criptografar e comprimir dados de maneira a ser recebido corretamente na outra ponta.

2.1.1.7 Camada 7: Aplicação

De acordo com Cloudflare (2018), a camada de aplicação é a única camada que interage diretamente com os dados do usuário. Aplicações como os navegadores dependem da camada de aplicação pra iniciar a comunicação. No entanto, é necessário lembrar que essas aplicações não fazem parte da camada de aplicação, mas sim, Cloudflare (2018) mostra que a camada de aplicação é responsável pelo protocolo e manipulação de dados que o software depende para apresentar dados significativos para o usuário. Protocolos da camada de aplicação incluem, por exemplo, de acordo com Imperva (2016), o HTTP, FTP, POP, SMTP e DNS.

2.1.2 Modelo OSI vs Modelo Internet (TCP/IP)

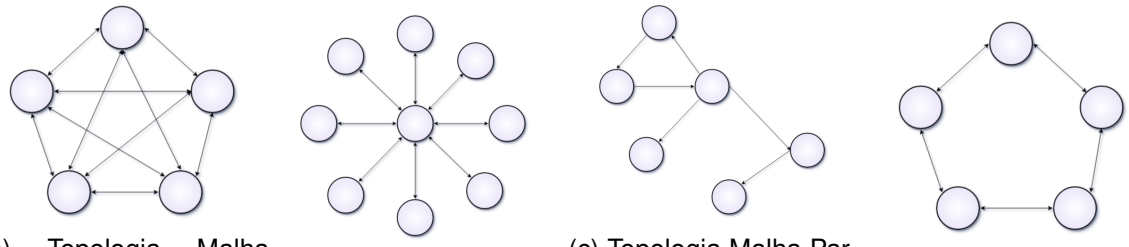
O modelo TCP/IP ajuda a determinar como um computador deve se conectar a internet e como dados podem ser transmitidos. Esse modelo ajuda a criar uma rede virtual quando múltiplos computadores estão conectados mutualmente (CLOUDFLARE, 2018). A diferença chave é que o modelo TCP/IP é mais simples, simplificando várias camadas em uma, a saber:

- Camadas 7, 6 e 5 da OSI são combinadas na camada de Aplicação TCP/IP;
- Camadas 1 e 2 são combinadas na camada de acesso de rede TCP/IP;

Outra diferença importante é a especificação dos modelos (IMPERVA, 2016). O modelo TCP/IP é funcional e feito pra resolver problemas específicos de comunicação, e é baseado em protocolos padronizados e específicos. Em contrapartida, o modelo OSI é genérico e independente de protocolos, feito para descrever todas as formas de comunicação em rede. No mesmo tom, no modelo TCP/IP, a maioria das aplicações usam todas as camadas, enquanto no modelo OSI, aplicações simples não usam todas as 7 camadas. No modelo OSI, apenas as camadas 1, 2 e 3 são mandatórias para permitir comunicação de dados

2.1.3 Topologias

Redes de comunicação existem em diferentes sabores, que podem ser divididos em redes físicas e lógicas. Atualmente, robustez e escalabilidade são de caráter essencial, portanto, a ênfase é posta no design de topologia lógico (SCHILLER et al., 2022). As topologias de rede podem ser resumidas em quatro grandes grupos, incluindo o papel específico de nós do tipo *gateway* (portal), uma vez que esses nós mostram impactos diferentes em vários cenários em IoT.



(a) Topologia Malha (mesh) (b) Topologia Estrela (c) Topologia Malha Parcial (d) Topologia Ciranda

Figura 2 – Topologias de rede, adaptado de (SCHILLER et al., 2022)

2.1.3.1 Conexão ponto-a-ponto

Nós ligados com uma conexão dedicada formam uma topologia ponto a ponto (SCHILLER et al., 2022), onde nós agem como pontos finais. De maneira a funcionar corretamente, nem um dos pontos nem o enlace pode falhar

2.1.3.2 Ciranda

Quando nós são ligados em série, a rede resultante é conhecida como ciranda (ou, cadeia de margaridas). Uma rede linear conecta cada nó ponto-a-ponto, onde um nó pode funcionar como um monitor IoT (SCHILLER et al., 2022). Geralmente é utilizada em redes menores devido a baixo custo de instalação. No entanto, é uma topologia complexa de diagnosticar, uma vez que uma perturbação em um nó pode se propagar para toda a rede.

2.1.3.3 Estrela

Quando todos os nós estão conectados a um *gateway* central, a topologia resultante é chamada de topologia estrela. As vantagens principais são a efetividade de custo, facilidade de instalação e resiliência (SCHILLER et al., 2022). Falha ou vulnerabilidade de um nó não compromete toda a rede.

2.1.3.4 Malha

Uma topologia de malha (*mesh*) é caracterizada por pelo menos três nós distintos, onde cada um desses pontos é vizinho de um conjunto de outros nós, conforme ilustrado na figura 2a. Essas conexões podem ser geradas de maneira dinâmica ou hierárquica.

2.1.3.5 Nós Gateway

De acordo com Glória; Cercas; Souto (2017), um nó do tipo Gateway (*Portal*) é construido por um hardware pronto para suportar diferentes tipos de nós sensores e protocolos de comunicação, cabeados ou sem-fio. Também é responsável por prover um conjunto de de informação unificado para a aplicação. O grande desafio de criar um gateway IoT é a falta de padrão industrial, dado que cada sensor pode comunicar com um protocolo diferente. A figura 3 ilustra a posição do gateway numa arquitetura IoT.

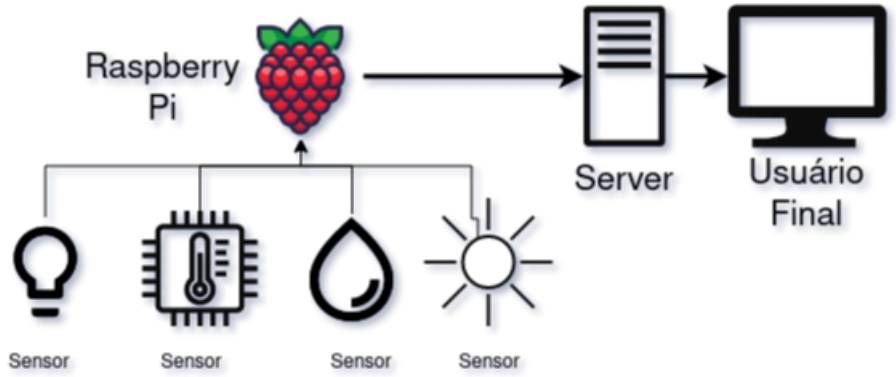


Figura 3 – Visualização de uma arquitetura exemplo envolvendo gateway, modificado de (GLÓRIA; CERCAS; SOUTO, 2017)

2.2 Preocupações com Segurança

O termo *Segurança da Informação* não tem uma definição estática e evolui conforme a tecnologia existente também o faz. Porém, desde a década de 80, a segurança da informação engloba as metas de garantir a disponibilidade, confidencialidade e integridade da informação (SCHILLER et al., 2022). Assim, vamos definir os seguin-

tes axiomas:

- Uma mensagem é confidencial se, e somente se, apenas o remetente e o destinatário sabem da sua existência
- Uma mensagem é integra apenas se o conteúdo é idêntico entre o remetente e o destinatário.
- Uma mensagem tem disponibilidade se é legível pelo remetente e o destinatário a qualquer momento
- Uma mensagem deve ser capaz de demonstrar a sua origem e vice versa
- Uma mensagem não pode ser enviada com o remetente adulterado

2.2.1 Terminologia de segurança

A terminologia de segurança descrita por Schiller et al. (2022) cria um terreno comum para análise de segurança, definindo o conceito de *adversário*, *malware*, *risco*, *ameaça* e *vulnerabilidade*:

- Um *Adversário* é definido como uma entidade acessando os recursos do sistemas de maneira ilegítima;
- Um *Malware* é a ferramenta em software que atinge o objetivo do adversário
- Uma *Ameaça* é qualquer evento tendo o potencial de violar a segurança e causar dano. Sua existência requer capacidade de execução ou circunstâncias favoráveis. Por exemplo, um *malware* instalado por um *adversário* sempre configura uma ameaça, mas nem sempre se torna um *Risco*
- Uma *Vulnerabilidade* existe quando uma falha no ponto de design, implementação, operação ou gerenciamento acontece.
- *Risco* ocorre quando uma ameaça e uma vulnerabilidade se encontram.

Quando todas as pré-condições supracitadas são executadas, esse *Risco* vira um *Ataque*. Um ataque pode ter como objetivo alterar recursos do sistema ou adquirir informação.

2.3 Internet das Coisas

O avanço tecnológico, no passado recente, transformou a internet para uma rede onde tudo é ligado e objetos de uso cotidiano podem ser reconhecidos e controlados. A internet das coisas é uma rede de objetos físicos que podem fazer sensoriamento, comunicação e serem acessados através da internet, tornando-se parte integral da mesma (SHARMA; SHAMKUWAR; SINGH, 2019). Esses objetos são embarcados com eletrônicos, *software*, sensores, atuadores e a conectividade de rede que habilita-os a coletar e trocar seus dados usando vários protocolos. Portanto, (SHARMA; SHAMKUWAR; SINGH, 2019) nos diz que a IoT oferece conectividade de serviços, dispositivos e sistemas que vão além da comunicação Máquina-Máquina (M2M) e está disposto a receber aplicações em domínios diferentes.

2.3.1 Uma breve contextualização histórica

O termo Internet das Coisas foi concebido em 1999 por Kevin Ashton para ilustrar o poder de conectar etiquetas de RFID na internet para gerenciamento de cadeia de suprimentos (ASHTON, 2009). Desde então, várias definições evoluíram baseadas nas definições das tecnologias em voga no momento e na gama de aplicações possíveis. Diferentes pesquisadores e cientistas definem o termo de maneiras diferentes, porém, a definição utilizada aqui é que a IoT é uma constelação de objetos, tecnologias, dispositivos e protocolos conectados por uma estrutura unificada que inclui computação ubíqua, computação em nuvem, análise de dados e visualização.

Com a taxa de crescimento exponencial da IoT, o campo industrial está amplamente motivado para investir com o objetivo de melhorar processos, minimizar riscos e melhorar experiência do usuário (SHARMA; SHAMKUWAR; SINGH, 2019). No entanto, IoT não é só colocar um sensor num objeto e chamá-lo de "inteligente". Uma solução compreensiva de IoT precisa de uma infraestrutura apropriada e um ambiente de apoio para a coleção e análise de dados. Se não existir um estudo intenso do ponto de vista de sistemas, se torna extremamente complexo entender as tecnologias chave de IoT (ZHANG; SUN; CHENG, 2012). Sendo assim, é necessário primeiro estruturar essas tecnologias.

2.3.2 O modelo de seis camadas da IoT

A arquitetura existente da internet foi adotada quatro décadas atrás, na forma de protocolos TCP/IP, mas hoje é incompatível para servir a rede imensa que a IoT necessita. Portanto, é necessário existir uma arquitetura nova que consiga trabalhar com uma rede de mais de 25 bilhões de dispositivos conectados (SHARMA; SHAMKUWAR; SINGH, 2019). Essa nova arquitetura deve usar protocolo de código aberto para suportar aplicações de redes já existente e prover segurança e qualidade de

serviço. Portanto, várias arquiteturas de segurança para IoT são propostas. Zhang; Sun; Cheng (2012) propôs uma arquitetura em seis camadas baseada numa estrutura hierárquica e desenvolvidas no contexto de uma rede RFiD, ilustradas na Figura 4. Para este trabalho, serão consideradas para análise apenas as camadas de rede, percepção e aplicação.

2.3.2.1 Primeira camada: Codificação

Essa é a camada base da arquitetura IoT, onde cada objeto de interesse é fornecido com um código para identificação única.

2.3.2.2 Segunda camada: Percepção

O objeto de interesse com código único é atribuído com significado físico, através da conexão de dispositivos IoT. Os dispositivos geralmente são sensores como etiquetas RFID, sensores infravermelhos ou outros sensores. Nessa camada, os sensores recolhem a informação do objeto de interesse, convertendo para sinal digital e passados adiante para a camada de rede [1].

2.3.2.3 Terceira camada: Rede

A camada de rede é responsável pela transmissão segura de dados entre a camada de percepção e a camada de *middleware*, onde ocorre o processamento. Essa camada pode utilizar vários meios de transmissão como Bluetooth, WiMaX, Zigbee, GSM, 3G, com protocolos como IPv4, IPv6, MQTT, AMQP, CoAP, XMPP, DDS e etc, e serve como um ponto de inflexão entre a internet e a rede baseada em comunicação.

2.3.2.4 Quarta camada: Middleware

Essa camada usa tecnologias avançadas como computação ubíqua, computação em nuvem para acessar uma base de dados diretamente e armazenar e prover as informações necessárias. Essa camada principalmente processa os dados de sensores recebidos pela camada de rede e executa uma ação automatizada baseada no resultado.

2.3.2.5 Quinta camada: Aplicação

Essa camada entrega serviços personalizados baseado nas necessidades do usuário, usando o resultado dos dados processados.

2.3.2.6 Sexta camada: Negócios

A camada de negócios é a camada superior da arquitetura IoT, onde várias regras de negócio são geradas para estratégias efetivas. As aplicações e serviços disponibi-

[1] não confundir com a camada OSI homônima

lizados pela IoT são gerenciados nessa camada.

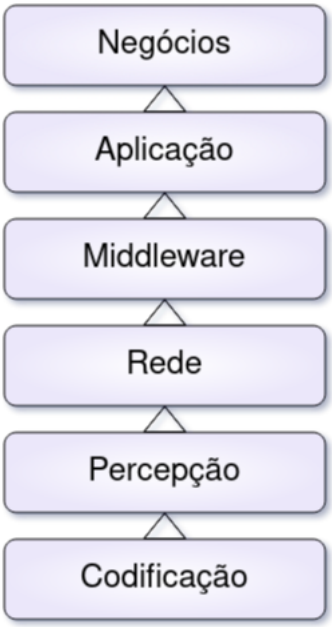


Figura 4 – Arquitetura de 6 camadas proposta por Zhang; Sun; Cheng (2012)

2.3.3 Protocolos

2.3.3.1 TCP/IP

O Protocolo de Controle de Transmissão E Protocolo de Internet (TCP/IP) é o protocolo mais amplamente utilizado na internet e quebra uma mensagem em pacotes para transmissão.

Vários problemas na arquitetura TCP/IP devem ser considerados no contexto de IoT (SCHILLER et al., 2022). Entre eles, a unidade máxima de transmissão (MTU) de 1280 bytes, que pode ser grande demais para dispositivos de baixa potência. De maneira semelhante, o TCP oferece várias utilidades como confiança de transmissão, controle de fluxo e capacidade de evitar congestionamentos, que podem ser pesadas demais para implementação em dispositivos IoT. Ademais, vias de comunicação implementadas em IoT podem ter perdas na comunicação que não são compensadas com os controles de fluxo previstos no TCP, uma vez que o TCP sempre trata perdas como sendo causadas por congestionamento na rede. Esses listados são apenas alguns disponíveis no mercado, sendo o MQTT e o CoAP mais comuns.

2.3.3.2 6LoWPAN

A Força Tarefa de Engenharia de Internet (IETF - *Internet Engineering Task Force*) criou um grupo de trabalho denominado *IPv6 over Low-Power Wireless Personal Area Networks* (IPv6 sobre redes sem fio pessoais de baixa potência), abreviado como 6LoWPAN. Esse protocolo implementa funcionalidade IPv6 sobre protocolo UDP para

garantir que nós sensores sejam compatíveis com diversas camadas físicas e MAC (SCHILLER et al., 2022)

2.3.3.3 Thread

Dispositivos prontos para protocolo Thread suportam um conjunto do padrão IEEE 802.15.4 e fazem uma implementação da 6LoWPAN (SCHILLER et al., 2022). Devido ao baixo consumo de energia, dispositivos IoT tem um sinal fraco de transmissão, fazendo a comunicação ser mais difícil Portanto, o Datagram Transport Layer Security (DTLS - *Segurança da Camada de Transporte por Datagramas*) é utilizado no Thread para garantir confidencialidade da mensagem.

2.3.3.4 MQTT / CoAP

o Protocolo Limitado de Aplicação (*CoAP - Constrained Application Protocol*) e o Transporte de Mensagens de Telemetria Enfileirada (*MQTT - Message Queuing Telemetry Transport*) são os dois principais protocolos de mensagem na camada de transporte usado em IoT (SCHILLER et al., 2022). Ambos protocolos foram desenvolvidos para trabalhar com dispositivos IoT de baixa potência. O CoAP é um protocolo na camada de serviço feito para dispositivos de internet com recursos limitados, como redes de sensores sem fio, com a intenção de simplificar a funcionalidade do HTTP *(Hypertext Transfer Protocol - Protocolo de Transferência de Hipertexto*) e adaptar pra aplicações de IoT ao aumentar a simplicidade. Já o MQTT é um protocolo de mensagem do tipo assinatura/publicação para conectar dispositivos embarcados através de um servidor (chamado *broker*), onde as mensagens são endereçadas para tópicos aos quais os demais clientes assinados conseguem receber.

2.3.3.5 Grafana

O Grafana é uma das arquiteturas mais populares para receber dados de dispositivos IoT (SCHILLER et al., 2022), ela utiliza o protocolo MQTT atrelado à plataforma de código aberto Graphite e um sistema de visualização, que pode pegar dados em série temporal e exibi-los de forma amigável.

2.3.3.6 AMQP

o Protocolo Avançado de Fila de Mensagens *(AMQP - Advanced Message Queue Protocol)* foi feito para prover mensagens de alta performance e uso geral, com disponibilidade de implementação de filas e roteamento complexo (O'HARA, 2007), resultando num protocolo capaz de fazer rotinas de entrega mais complexa como multicast.

2.3.3.7 Modbus

O protocolo Modbus existe na camada de aplicação e cria comunicação modo cliente/servidor entre dispositivos conectados em barramentos diferentes ou redes distintas. Comunicações Modbus são de dois tipos (FOVINO et al., 2009).

- Pergunta/resposta, quando a comunicação é entre um mestre e um trabalhador
- Broadcast (*transmissão*), quando a comunicação é entre o mestre para todos os trabalhadores

Uma transação Modbus existe em um único frame (*quadro*) para o modo pergunta/resposta ou um único frame para o modo broadcast. O frame Modbus contem o endereço do receptor de destino, o comando que ele deve executar e os dados necessários pra executar o comando. A implementação Modbus TCP embarca o frame Modbus num frame TCP.

3 METODOLOGIA

Devido a natureza polimórfica de sistemas de Internet das Coisas, uma análise de segurança requer como primeiro passo uma análise profunda de *hardware*, *software* e de rede. Para cobrir essas camadas, é necessário conduzir uma análise de vulnerabilidade em potencial no sistema como, por exemplo, *software* desatualizado, comunicação insegura, ou senhas fracas que podem ser exploradas.

De maneira semelhante, os sistemas de autenticação (LDAP, TACACS+, e semelhantes) devem ser analisados para garantir que mecanismos de autorização e controle de acesso como autenticação em múltiplos fatores estejam íntegros.

A parte mais vulnerável na maioria dos sistemas de IoT é a segurança da comunicação dos dispositivos de onde os dados são coletados, uma vez que nem sempre esses dispositivos possuem a capacidade computacional para implementar modos seguros de comunicação. Por exemplo, um sensor que transmite seus dados via wi-fi em texto aberto, ou com uma criptografia fraca. Por exemplo, a aplicação de uma criptografia TLS em uma plataforma ESP32 consome cerca de 17 vezes mais energia do que uma transmissão sem criptografia (FISCHER et al., 2019).

Uma solução para essa análise é a aplicação de um *pentest* (teste de penetração), onde um ataque simulado é executado em um sistema computacional para avaliar sua segurança. Durante o pentest, todos os aspectos do sistema são postos a prova, uma vez que o agente aplicando esse teste dispõe das mesmas ferramentas, processos e técnicas de um adversário. A abordagem mais extensiva de um *pentest* envolve não só todos os passos supracitados como o desenvolvimento de uma descrição completa do sistema, a fim de compreender a área de superfície da penetração a ser executada e onde aplicar cada técnica. As técnicas e ferramentas variam, abaixo listam-se algumas:

- Ferramentas de mapeamento de rede e portas, como o nmap
- Analisadores de vulnerabilidade, como Nessus ou OpenVAS;
- Formalismos de exploração, para automatizar a exploração de vulnerabilidades descobertas como o Metasploit ou CANVAS

- Engenharia reversa em *firmwares* e executáveis, procurando vulnerabilidades como um *buffer overflow*, usando ferramentas como o IDA Pro, radare2 ou Binary Ninja
- Fuzzing, que é a técnica baseada em procurar vulnerabilidades através do envio de quantidades grandes de dados inesperados, aleatórios ou mal-formatados para o sistema, a fim de testar as rotinas de validação de entrada do sistema sob teste.

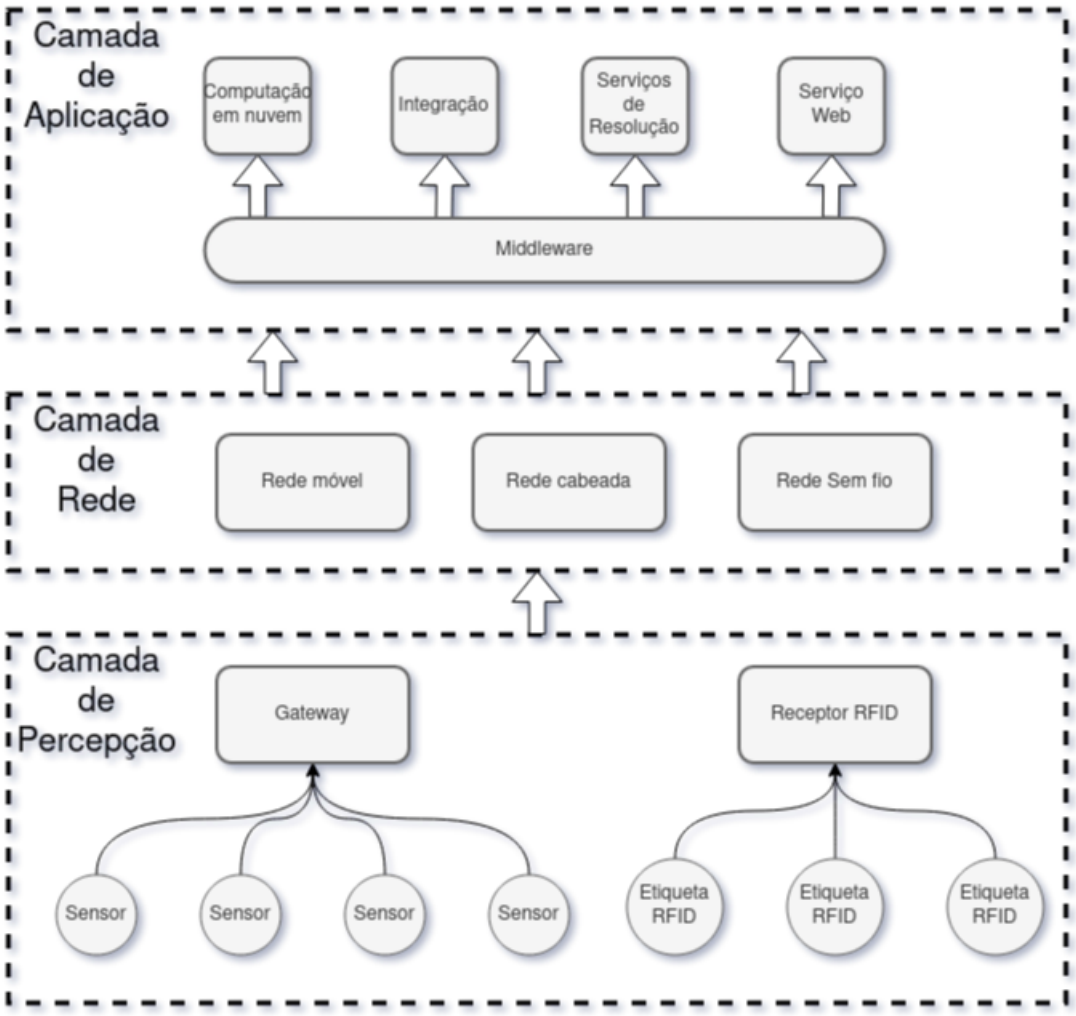


Figura 5 – Uma aplicação teórica, ilustrando as camadas sob análise. Adaptado de (SCHILLER et al., 2022)

Assim, serão criados dois ambientes: O ambiente SCADA, usando Modbus-TCP e o ambiente IoT, usando MQTT e CoAP. Nesses ambientes, essas técnicas de análise de segurança serão empregadas a fim de encontrar vulnerabilidades nessas implementações.

3.1 Descrição do Ensaio - SCADA

A maioria dos protocolos SCADA em uso hoje em dia foram desenvolvidos décadas atrás, quando a infraestrutura tecnológica e as ameaças eram bem diferentes das atuais. Por exemplo, o protocolo Modbus foi originalmente proposto num artigo de 1979.

Como as redes Modbus eram originalmente isoladas e livres de adversários externos, aspectos chave como integridade e autenticação não foram levadas em consideração na proposta original do protocolo.

O transporte de mensagens Modbus usando TCP introduz novos níveis de complexidade no que tange ao gerenciamento da entrega confiável de pacotes de controle em um ambiente com limitações estritas de tempo-real. Não obstante, abre novas avenidas para adversários com alvo em sistemas industriais. A falta de mecanismos para proteger a confidencialidade e verificar a integridade de mensagens transmitidas entre um mestre e seus trabalhadores. Por exemplo, não é possivel identificar se a mensagem original foi adulterada por um adversário. De maneira semelhante, não existe autenticação entre mestre e trabalhadores, ou seja, um dispositivo comprometido pode agir como mestre e mandar comandos para os trabalhadores.

Alguns ataques chave a ser conduzido contra Modbus TCP são:

- Execução de Comandos Não-autorizados, devido a falta de autenticação entre mestre e trabalhador significa que um adversário pode mandar mensagens forjadas da Modbus para um grupo de trabalhadores. Para executar este ataque, o adversário deve ser capaz de acessar a rede onde reside os servidores SCADA ou o barramento onde os trabalhadores estão conectados.

- Ataque de Negação de Serviço (*Denial of Service*, ou DoS), onde, por exemplo, é possível mandar mensagens confusas aos terminais de um dispositivo que finge ser o mestre. Essa sobrecarga de mensagens faz com que os terminais gastem seus recursos de processamento

- Ataque *Man-in-the-middle*), devido a falta de testes de integridade possibilita um adversário a se posicionar entre o controlador e o mestre de maneira a alterar mensagens legítimas para dispositivos trabalhadores

3.1.1 Criação de Ambiente

Para a criação desse laboratório de pentest, será usado uma máquina Linux Ubuntu 20.04 com kernel 5.19 usando o programa VirtualBox da Oracle. O VirtualBox por sua vez usará a pilha KVM para efetuar a virtualização de 5 máquinas:

- Um firewall de código aberto pfsense;

- Uma máquina debian agindo como estação de trabalho usando OpenPLC;
- Uma máquina para execução dos comandos da estação de trabalho, rodando o OpenPLC Runtime;
- Uma máquina para a aquisição de dados SCADA, usando uma imagem do SCADABR;
- Uma máquina para a interpretação dos comandos, usando Factory IO;
- uma máquina de um adversário, invadindo a rede.

O diagrama de rede desse ensaio está ilustrado na figura 6.

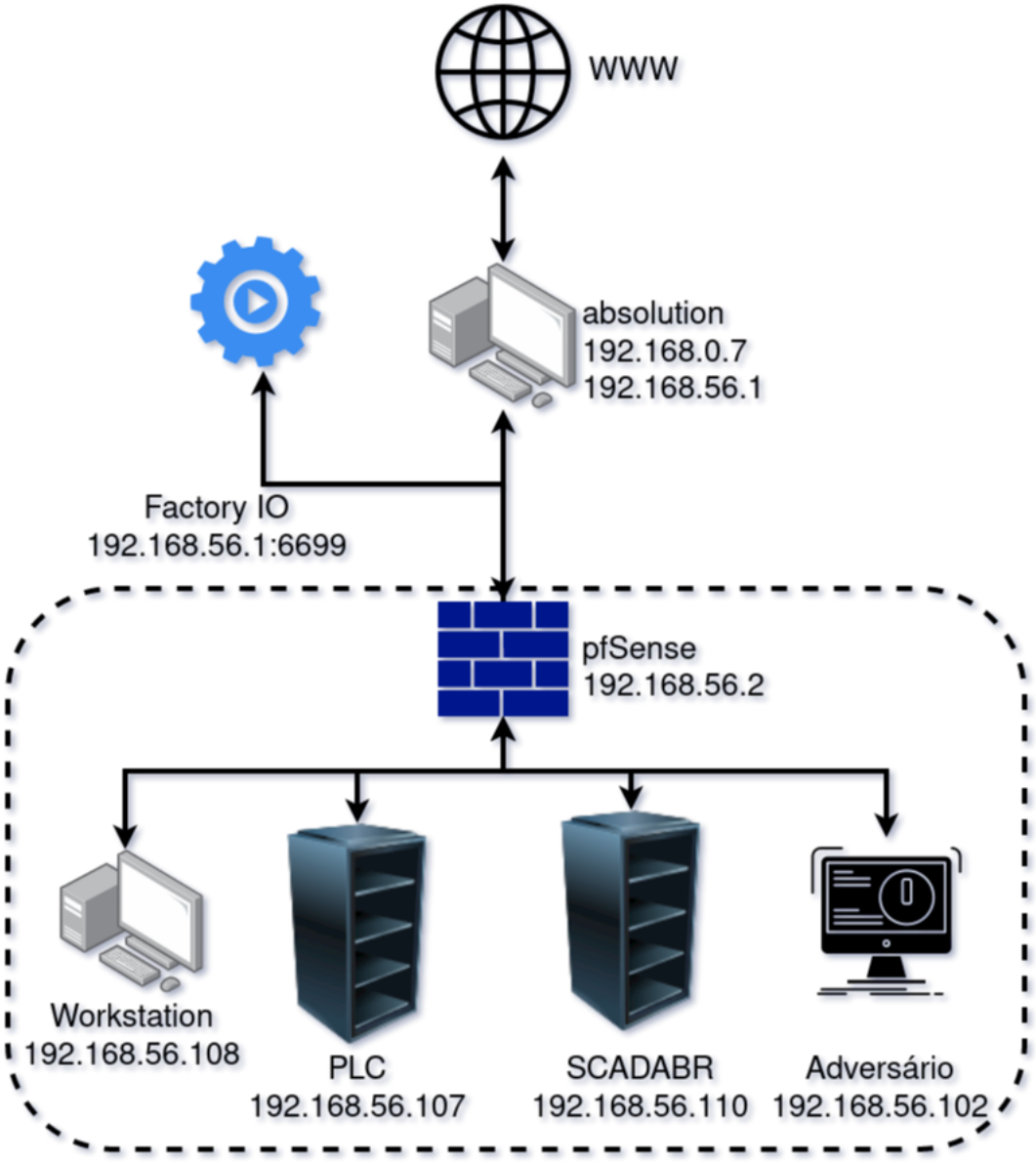


Figura 6 – Ilustração da rede para o laboratório SCADA

A partir dessa construção, será desenvolvido um programa simbólico em Ladder para ser executado no ambiente FactoryIO e gerenciado pelas máquinas PLC e SCADABR. Assim, então, será conduzido uma análise de exploração dessa configuração como prova de conceito.

3.2 Descrição do Ensaio - IoT

Primeiramente, é necessário assumir que nada é conhecido sobre o sistema sujeito ao ataque. Ou seja, nenhuma informação sobre a infraestrutura, mecanismos de defesa e canais de comunicação. O ataque então é iniciado através de um mapeamento básico usando uma ferramenta como o *nmap* para coleção de informação. A porta padrão do MQTT é a 1883, no entanto, o *nmap* nos dá uma relação completa das portas abertas na rede.

Para um cenário inicial, é possível assinar a todos os tópicos negociados pelo *broker*, através da assinatura ao tópico $\#$, que pode nos dar dados confidenciais para ser analisado em um outro momento. Um outro cenário pode ser iniciado ao publicar dados num broker que não tem um mecanismo de autenticação, efetivamente assumindo o controle do dispositivo na outra ponta da rede. Esse último pode ser utilizado para publicar quantidades imensas de dados de maneira a causar uma sobrecarga no broker e no dispositivo controlado (ataque de negação de serviço - DoS).

De acordo com Andy; Rahardjo; Hanindhito (2017), ambos os cenários supracitados são genéricos o suficiente para serem aplicados em uma rede local ou pública.

3.2.1 Infraestrutura

4 ANÁLISE DE RESULTADOS

4.1 Análise do ensaio IoT

4.1.1 Privacidade

A questão da privacidade é primordial no protocolo MQTT, uma vez que, por padrão, o protocolo não provê nenhuma criptografia sobre os dados trafegados. Apesar da utilização de mecanismos de autenticação, qualquer adversário ainda consegue farejar e analizar os dados com o uso de uma ferramenta como o *Wireshark*, como demonstrado por Andy; Rahardjo; Hanindhito (2017).

4.1.2 Autenticação

Se o broker usa autenticação do cliente através de uma combinação usuário/senha, idealmente o adversário não pode agir como um publicador ou assinante sem o conhecimento dessas informações. Em um cenário onde um adversário está na mesma rede que o nó publicador, o adversário consegue farejar o tráfego na rede enquanto espera por um pacote de solicitação de conexão do publicador entrar em trânsito. Assim, é possível revelar o usuário e senha usado por esse usuário legítimo para assumir o controle dos dispositivos assinantes.<F9><F9>

5 CONCLUSÃO

Apesar dessas considerações, existem varias soluções para aumento de segurança. Uma delas é a reimplementação de protocolos como MQTT e Modbus TCP levando em consideração as ferramentas necessárias desses protocolos para garantir a segurança num ambiente de rede moderno, como proposto por Fovino et al. (2009), onde essa reimplementação é construida com os seguintes pressupostos:

- Nenhuma entitade deve poder acessar o conteúdo de uma mensagem sem autorização;
- Integridade: nenhuma entidade deve poder modificar o conteúdo de uma mensagem sem autorização. Isso é garantido usando uma chave de criptografia (SHA2) calculada a partir do conteúdo da mensagem. Essa chave transmitida junto com o pacote original, para ser comparada pelo cálculo da chave efetuado pelo receptor.
- Autenticação: nenhuma entidade deve poder se passar por outra entidade. Apesar da garantia de integridade, o cálculo dessa chave não basta para garantir que a mensagem veio de um remetente autêntico. Assim, a autenticação é garantida através de uma assinatura RSA por parte do remetente que pode ser examinada pelo receptor.
- Nenhuma entidade deve poder negar uma ação em execução
- Nenhuma entidade deve poder reutilizar uma mensagem capturada como meio de acesso não autorizado. Isso é implementado usando um carimbo datado para discriminar entre um pacote novo e um usado.

REFERÊNCIAS

ANDY, S.; RAHARDJO, B.; HANINDHITO, B. Attack scenarios and security analysis of MQTT communication protocol in IoT system. In: INTERNATIONAL CONFERENCE ON ELECTRICAL ENGINEERING, COMPUTER SCIENCE AND INFORMATICS (EECSI), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.1–6.

ASHTON, K. That 'Internet of Things' thing. **RFiD Journal**, [S.l.], 2009.

CLOUDFLARE. **What is the OSI Model?** Disponível em: <https://www.cloudflare.com/learning/ddos/glossary/open-systems-interconnection-model-osi>. Acesso em: 2023-4-4.

FISCHER, T. et al. Analyzing power consumption of TLS ciphers on an ESP32. **crypto day matters 30**, [S.l.], 2019.

FOVINO, I. N.; CARCANO, A.; MASERA, M.; TROMBETTA, A. Design and implementation of a secure modbus protocol. In: CRITICAL INFRASTRUCTURE PROTECTION III: THIRD ANNUAL IFIP WG 11.10 INTERNATIONAL CONFERENCE ON CRITICAL INFRASTRUCTURE PROTECTION, HANOVER, NEW HAMPSHIRE, USA, MARCH 23-25, 2009, REVISED SELECTED PAPERS 3, 2009. **Anais...** [S.l.: s.n.], 2009. p.83–96.

GLÓRIA, A.; CERCAS, F.; SOUTO, N. Design and implementation of an IoT gateway to create smart environments. **Procedia Computer Science**, [S.l.], v.109, p.568–575, 2017.

IBARRA-ESQUER, J. et al. Tracking the Evolution of the Internet of Things Concept Across Different Application Domains. **Sensors**, [S.l.], v.17, n.6, p.1379, Jun 2017.

IMPERVA. **OSI Model Explained**: The OSI 7 Layers. Disponível em: <https://www.imperva.com/learn/application-security/osi-model/>. Acesso em: 2023-4-4.

KASPERSKY. **Stuxnet**: As Origens. Disponível em: <https://www.kaspersky.com.br/blog/stuxnet-as-origens/4391/>. Acesso em: 2023-4-4.

KUROSE, J. F.; ROSS, K. W. **Computer Networking - A Top-down Approach**. Harlow, Reino Unido: Pearson, 2022.

LAYA, A.; BRATU, V.-I.; MARKENDAHL, J. **Who is investing in machine-to-machine communications?** [S.l.]: International Telecommunications Society (ITS), 2013. 24th European Regional ITS Conference, Florence 2013.

O'HARA, J. Toward a commodity enterprise middleware: Can AMQP enable a new era in messaging middleware? A look inside standards-based messaging with AMQP. **Queue**, [S.l.], v.5, n.4, p.48–55, 2007.

SCHAFFERS, H. et al. Smart Cities and the Future Internet: Towards Cooperation Frameworks for Open Innovation. In: THE FUTURE INTERNET, 2011, Berlin, Heidelberg. **Anais...** Springer Berlin Heidelberg, 2011. p.431–446.

SCHILLER, E. et al. Landscape of IoT security. **Computer Science Review**, [S.l.], v.44, p.100467, 2022.

SHARMA, N.; SHAMKUWAR, M.; SINGH, I. **The History, Present and Future with IoT**. [S.l.]: Springer International Publishing, 2019. p.27–51.

ZHANG, M.; SUN, F.; CHENG, X. "Architecture of Internet of Things and Its Key Technology Integration Based-On RFID". In: FIFTH INTERNATIONAL SYMPOSIUM ON COMPUTATIONAL INTELLIGENCE AND DESIGN, 2012., 2012. **Anais...** [S.l.: s.n.], 2012. v.1, p.294–297.

Apêndices

APÊNDICE A – Um Apêndice

Anexos

ANEXO A – Um Anexo

ANEXO B – Outro Anexo