CENTRO UNIVERSITÁRIO CARIOCA

JAIME MARTINS DELLA CORTE DE ARAÚJO

ROBERTA VASCONCELLOS DO NASCIMENTO

ATIVIDADES PRÁTICAS PARA O ENSINO DE SEGURANÇA DA INFORMAÇÃO EM DISPOSITIVOS PARA INTERNET DAS COISAS

RIO DE JANEIRO

2022

JAIME MARTINS DELLA CORTE DE ARAÚJO

ROBERTA VASCONCELLOS DO NASCIMENTO

Atividades práticas para o ensino de segurança da informação em dispositivos para Internet das Coisas

Trabalho de conclusão de curso apresentado ao Centro Universitário Carioca – Unicarioca como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Marcelo Perantoni

Rio de Janeiro

2022

C357a Della Corte, Jaime Martins
Atividades práticas para o ensino de segurança da informação em dispositivos para internet das coisas / Jaime Martins Della Corte e Roberta Vasconcellos do Nascimento – Rio de Janeiro, 2022.
69 f.

Orientador: Marcelo Perantoni
Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Centro Universitário UniCarioca, Rio de Janeiro, 2022.

1. IoT. 2. Segurança da informação. 3. Tecnologia –Vulnerabilidades. I. Nascimento, Roberta Vasconcellos do. II. Perantoni, Marcelo, prof. orient. III. Título.

CDD 004.60289

Dedicatória

Agradeço e dedico este trabalho de conclusão de curso primeiramente aos meus pais. Esta é a prova de que todo seu investimento e dedicação valeram a pena. A todos os professores desta minha jornada, eles são minha maior inspiração e sempre lembrarei dos meus queridos mentores. Ao meu parceiro de TCC, Jaime e aos meus amigos Nelson, Rafael e Maurício por estarem disponíveis tanto para as minhas dúvidas quanto para questões pessoais, me dando muito apoio e força. Tenho certeza de que a qualidade deste trabalho não seria a mesma sem a ajuda de todos vocês.

– Roberta Vasconcellos do Nascimento

Agradeço e dedico este Trabalho de Conclusão de Curso (TCC), em primeiro lugar, aos meus pais Ana Paula e Jaime Della Corte, que me inspiraram e sempre me motivaram em tudo, ensinando-me a trilhar no caminho que me levasse ao êxito, dando-me apoio para ultrapassar barreiras e auxílio para lidar com os momentos difíceis e enfrentamento de desafios, fazendo toda a diferença no meu sucesso e construção do meu caráter. A minha irmã Jamile por existir, tornando-me, assim como meus pais, responsável para ser exemplo para que ela colha bons frutos na vida pessoal e futuramente profissional. Ao meu Orientador, professor Perantoni, que com muita sabedoria e experiência nos conduziu e acreditou na concretização do nosso projeto, disponibilizando a mim e a minha grande parceira de TCC Roberta Nascimento, total ajuda com bastante paciência e atenção durante todo este processo. Aos meus amigos Yuri “Solid1” Miranda, Luiza “Lufeli” Velloso e Daniel “DanieiDK” Costa, que me incentivaram e me apoiaram todas as vezes que precisei. Gratidão a cada um de vocês!

– Jaime Martins Della Corte

JAIME MARTINS DELLA CORTE DE ARAÚJO

ROBERTA VASCONCELLOS DO NASCIMENTO

Atividades práticas para o ensino de segurança da informação em dispositivos para Internet das Coisas

Trabalho de conclusão de curso apresentado ao Centro Universitário Carioca – Unicarioca como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

BANCA EXAMINADORA

_____

Professor Marcelo Perantoni
Orientador
Centro Universitário Carioca (Unicarioca)

_____

André Luiz Avelino Sobral
Coordenador dos Cursos de Análise e Desenvolvimento de Sistemas, Ciência da Computação e Redes de Computadores
Centro Universitário Carioca (Unicarioca)

_____

Professor Sergio Assunção Monteiro
Integrante da Banca Examinadora
Centro Universitário Carioca (Unicarioca)

“O que levamos conosco deve nos elevar, não nos derrubar. Memórias, amor e esperança são meras necessidades.”
(Kai’Sa)

Resumo

O mundo passou por uma grande evolução da tecnologia nas últimas décadas e os humanos sentiram a necessidade de trazer esta evolução para seu dia a dia, com o surgimento da Internet das Coisas nos anos 90 houve um rápido crescimento nos setores comercial e doméstico, pois ela conecta objetos à Internet e promove a comunicação entre dispositivos. Por outro lado, também é possível constatar que há problemas com a segurança, tanto por parte das empresas quanto dos usuários. Os benefícios da IoT são inegáveis, mas as empresas devem aumentar o grau de segurança de seus dispositivos antes de chegarem ao mercado, permitindo que os usuários usem com responsabilidade e respeitem as regras de segurança. Com a falta de profissionais de TI no mercado, principalmente se dedicando à segurança da informação, este objetivo parece cada vez mais distante. Apresentamos uma alternativa para este problema, o incentivo ao estudo de segurança da informação em IoT com metodologias ativas. Demonstrando um projeto com protótipo com a verificação de algumas vulnerabilidades de rede presentes no mesmo e que podem estar presentes em outros dispositivos IoT.

Palavras-chave: IoT, Segurança, tecnologia, vulnerabilidades

Abstract

The world has gone through a great evolution of technology in the last decades and humans felt the need to bring this evolution into their daily lives, with the emergence of the Internet of Things in the 90s there was a rapid growth in the commercial and domestic sectors, as it connects objects to the Internet and promotes communication between devices. On the other hand, it is also possible to verify that there are problems with security, both on the part of companies and users. The benefits of IoT are undeniable, but companies must increase the degree of security of their devices before they hit the market, allowing users to use them responsibly and respect security rules. With the lack of IT professionals in the market, mainly dedicated to information security, this objective seems increasingly distant. We present an alternative to this problem, encouraging the study of information security in IoT with active methodologies. Demonstrating a prototype project with the verification of some network vulnerabilities present in it and that may be present in other IoT devices.

Keywords: IoT, Security, technology, vulnerabilities

Lista de Ilustrações

Lista de Siglas e Abreviaturas

3G – 3rd Generation

4G – 4th Generation

5G – 5th Generation

AC/DC – Alternating Current/Direct Current

ACK – Acknowledgement

AP – Access Point

API – Application Programming Interface

ARP – Address Resolution Protocol

ARPANET – Advanced Research Projects Agency Network

CD – Compact Disc

CLP – Controlador Lógico Programável

CPS – Cyber-Physical Systems

DoS – Denial of Service

EAPOL – Extensible Authentication Protocol over LAN

GMK – Group Master Key

GTK – Group Temporal Key

ICMP – Internet Control Message Protocol

IEEE – Institute of Electrical and Electronic Engineers

IoS – Internet of Services

IoT – Internet of Things

IP – Internet Protocol

IPv4 – Internet Protocol Version 4

IPv6 – Internet Protocol Version 6

LED – Light Emitting Diode

M2M – Machine-to-machine

MAC – Media Access Control

MIC – Message Integrity Code

MSK – Master Session Key

NFC – Near Field Communication

NTC – Negative Temperature Coefficient

PMK – Pairwise Master Key

PTK – Pairwise Transient Key

RTT – Round Trip Time

SOC – System On Chip

STA – Station

SYN – Synchronize

TCC – Trabalho de Conclusão de Curso

TCP – Transmission Control Protocol

TI – Tecnologia da Informação

TX/RX – Transmissor / Receptor

UDP – User Datagram Protocol

USB – Universal Serial Bus

VPN – Virtual Private Network

WEP – Wired Equivalent Privacy

WLAN – Wireless Local Area Network

WPA – WiFi Protected Access

SUMÁRIO

1 Introdução

Com a chegada da quarta revolução industrial, a IoT (Internet das Coisas) está se tornando cada vez mais popular mundialmente. Os sistemas inteligentes conectados nos ajudam a resolver muitos problemas. Hoje, já podemos desfrutar de dispositivos *smart* em nossas casas, indústrias e mais locais.

Segundo Neto e Araújo (2019, p.11),

> “Em um momento em que, para muitos, não importa a exposição que tais ferramentas podem ocasionar, aplicativos e softwares são instalados sem nenhum conhecimento prévio dos seus riscos e vulnerabilidades que podem provocar. A convergência tecnológica já não possibilita a proibição de alguns recursos tecnológicos dentro do ambiente organizacional, e os mesmos dispositivos e os recursos que beneficiam as organizações acabam tornando-as vulneráveis.”

Sendo assim, a segurança da informação torna-se cada vez mais fundamental tanto no âmbito pessoal e residencial quanto no organizacional.

Enquanto isso, as universidades e centros técnicos encontraram dificuldades em formar profissionais de TI, e consequentemente de segurança da informação, suficientes no Brasil. Segundo um estudo da Associação para a Promoção da Excelência do Software Brasileiro, o mercado brasileiro de Tecnologia da Informação terá um déficit de 400 mil profissionais até 2022.
(SOFTEX, 2009). Apesar do crescimento do número de cursos voltados para TI e das inscrições, mais de 80% dos alunos não concluem o curso (CARDOSO, ÉRICO; DAVID, TOBIAS, 2016).

Uma possível iniciativa para reverter essa situação é diversificar o ensino, que é muito focado em tecnologias tradicionais e aulas pouco focadas no aluno. Existem diversos trabalhos embasando os benefícios de métodos ativos em sala de aula. Inclusive no ensino de programação, ajudando a despertar interesse dos alunos.

Segundo Gil (2020, p. 95), sobre o conceito de métodos ativos,

> “É um termo utilizado para designar um amplo espectro de estratégias para facilitar a aprendizagem, que se caracterizam principalmente por serem centradas no aluno. Abrange, dentre outras, a aprendizagem baseada em projetos, a aprendizagem baseada em problemas, o método de caso, o aprendizado baseado em jogos e a sala de aula invertida.”

Com esse cenário em mente, esse TCC abordará por meio de experimentos algumas vulnerabilidades de IoT com o uso de um protótipo com microcontrolador. Propondo métodos ativos como projetos, experimentos e dinâmicas para estimular o interesse dos alunos de forma multidisciplinar.

Com o intuito de construir uma solução que estimule a aprendizagem de conceitos de segurança da informação voltados a IoT, seguimos em direção à prototipagem.

O Arduino é uma das melhores interfaces de baixo custo para a prototipação de soluções IoT, pois possui um microcontrolador compatível com linguagem de programação baseada na linguagem C.

O protótipo em si é um circuito de monitoramento de temperatura e umidade. Para obter maior fidelidade possível aos dispositivos IoT, adicionamos o módulo ESP8266 para conexão via WiFi e programamos para que envie os dados recebidos pelos sensores para a API ThingSpeak, a qual é um serviço que permite o envio, visualização e análise de fluxos de dados na nuvem.

Os experimentos ilustrados neste TCC são *pentests* (testes de intrusão) direcionados apenas ao protótipo de dispositivo IoT. Pentest é um método de avaliação de segurança de sistema ou rede, simulando um ataque malicioso. Por intermédio dele, podemos detectar vulnerabilidades em sistemas.

Decidimos explorar apenas vulnerabilidades de rede do protótipo IoT utilizando ferramentas do Sistema Operacional Kali Linux, o qual é um sistema voltado principalmente para auditoria e segurança de computadores em geral.

Com esses apontamentos feitos, entendemos que o projeto tem abordagem multidisciplinar com conceitos de Segurança da Informação, Programação, Redes, Linux e Eletrônica.

2 Fundamentação Teórica

2.1 Contexto Histórico

A IoT nos permite realizar diversas tarefas sem sair do lugar e conecta máquinas, dispositivos e pessoas para ajudar e tornar a vida mais prática. A seguir, veremos um breve histórico da Quarta Revolução Industrial e da Internet, o contexto que fundamenta a IoT e nos remete ao atual cenário.

2.1.1 Evolução da Internet

Para melhor entendimento da evolução da utilização e capacidade da Internet ao longo de seu caminho, dividiu-se a mesma em três gerações, conforme ilustrado na figura 1.

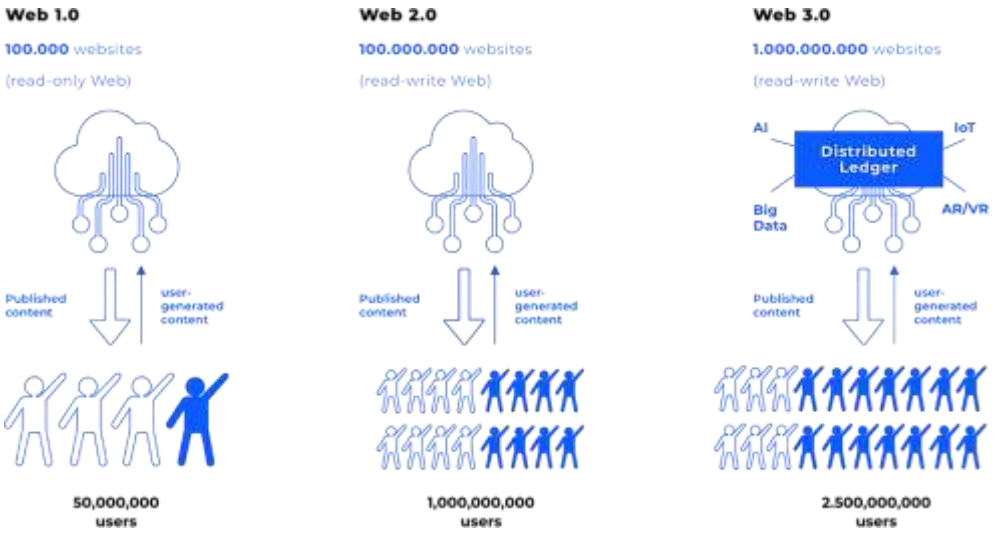

Figura 1: Ilustração sobre a divisão da evolução web. Extraído de: https://medium.com/mastering-web3-with-waves/mastering-web3-with-waves-module-2-aa98f7dfcdde

Web 1.0 (Read-Only Web, Web somente para leitura): Surgiu em meados de 1980 conectando as pessoas de forma estática e não era possível interagir com sites. Esta falta de interação é uma característica inerente da Web 1.0, porém isso não

diminui o seu impacto. A Web 1.0 ficou conhecida como “Web do Conhecimento” devido ao aumento repentino de informações que fornece aos usuários. Os primeiros sites de vendas eletrônicas disponibilizavam apenas seus catálogos em formato virtual, tornando mais fácil conhecerem seus serviços e produtos.

A passagem de Web 1.0 para Web 2.0 não aconteceu de forma clara. Nem sempre é possível marcar um site como 1.0 ou 2.0. Além disso, alguns sites mais simples podem ser tão eficazes quanto outros mais complexos. Esta mudança não se dá por nenhuma inovação tecnológica, mas por uma nova forma de usar ferramentas que estão disponíveis na web.

Web 2.0 (Read-Write Web, Web de leitura e escrita): Remete à uma natureza colaborativa e interação contínua com o usuário. Pode-se considerá-la a “Web de Comunicação” devido ao alto nível de interatividade em sua plataforma. Estas relações foram possíveis devido à expansão de plataformas como: redes sociais, blogs, wikis, etc. Com o surgimento das redes colaborativas, os internautas deixaram de ser apenas consumidores de conteúdo, e também se tornaram produtores. Há também preocupações com a estrutura da rede durante esta transição, mais preocupada com o fluxo de informações e coleta de dados, o que também acontece como barreira para a ascensão da IoT, porém a escala é diferente. Com o advento da web 2.0, os sites de vendas eletrônicas passaram a criar ferramentas de categorização de produtos e abrir espaço para avaliações de usuários.

Web 3.0 (Read-Write Web, Web de leitura e escrita): O termo Web 3.0 foi criado pelo repórter do New York Times John Markoff com base no avanço da Web 2.0 apresentado por Tim O'Reilly e Dale Dougherty em 2004. Enquanto a Web 2.0 permitiu que as pessoas interagissem, a Web 3.0 usa a Internet para mesclar dados. Os dispositivos podem ler essas informações e fornecer outras mais precisas. Por se tratar do momento atual, o conceito de Web 3.0 ainda está sendo construído, porém ele já possui características que o distinguem dos momentos anteriores. A principal característica é a relação com a IoT, onde objetos interagem com pessoas e com outros objetos (MAGRANI, 2018).

A Internet em si está se desenvolvendo e melhorando constantemente, mas não mudou muito. Desta forma, a Internet das Coisas torna-se muito importante por ser o primeiro desenvolvimento propriamente dito da Internet, uma mudança que levará a aplicativos que têm o potencial de melhorar drasticamente a maneira como as pessoas as utilizam.

Com o conceito de Web 3.0, surgiu também a Internet Semântica. O criador da World Wide Web, Tim Berners-Lee, demonstrou que a Web Semântica integra a Web 3.0. Nos primórdios da internet, tudo era gerado para a compreensão humana, o que significava que podíamos identificar facilmente as páginas da web. Os computadores não têm essa capacidade, mas isso está mudando. Com a Internet Semântica, os dispositivos podem capturar e interpretar as informações fornecidas por usuários. Ao agregar essas informações pessoais, a plataforma poderá personalizar os resultados. Por exemplo: mesmo que duas pessoas pesquisem com o mesmo termo, os resultados serão diferentes, pois a pesquisa também leva em consideração o histórico de cada pessoa (MAGRANI, 2018).

2.1.2 4ª Revolução Industrial

Figura 2: Revoluções Industriais (SACOMANO, 2018)

Ao longo da segunda metade do século XX, temos a ascensão do digital sobre o analógico e a automação vindo para reduzir tarefas repetitivas. No início do século XXI, o advento da transformação digital, caracterizada pela onipresença de computadores e smartphones, conexão à internet de amplo acesso e mídias de

comunicação mudando para o formato digital. No caso da Indústria, a base existente de automação informatizada e uma visão de negócios voltados à transformação digital fazem nascer o conceito de Indústria 4.0. É importante destacar que, enquanto outras revoluções industriais foram observadas e diagnosticadas posteriormente, na Quarta Revolução Industrial os acontecimentos estão sendo previstos e tratados como tendências (FIRJAN, 2016).

2.1.2.1 Indústria 4.0

A Indústria 4.0 promoverá transformações nas formas de produção e propõe novos desafios para o Brasil. Devido à digitalização e ao autogerenciamento das fábricas, haverá redução do quadro de funcionários, além de profissões que deixarão de existir, dando espaço para outras. A Internet das Coisas e Serviços será um pré-requisito para tal. Dessa forma, haverá uma mudança de paradigma na interação entre homem e máquina, que, nesse novo contexto, tomarão decisões conjuntas (SACOMANO; BONILLA, 2018).

O conceito de Indústria 4.0 ainda está em formação, sendo assim qualquer classificação que faz parte desse conceito não poderia passar de uma proposição.

Sacomano et al. (2018, p. 33), considera que uma proposta de classificação dos elementos formadores da Indústria 4.0 com caráter didático seria representada na Figura 3.

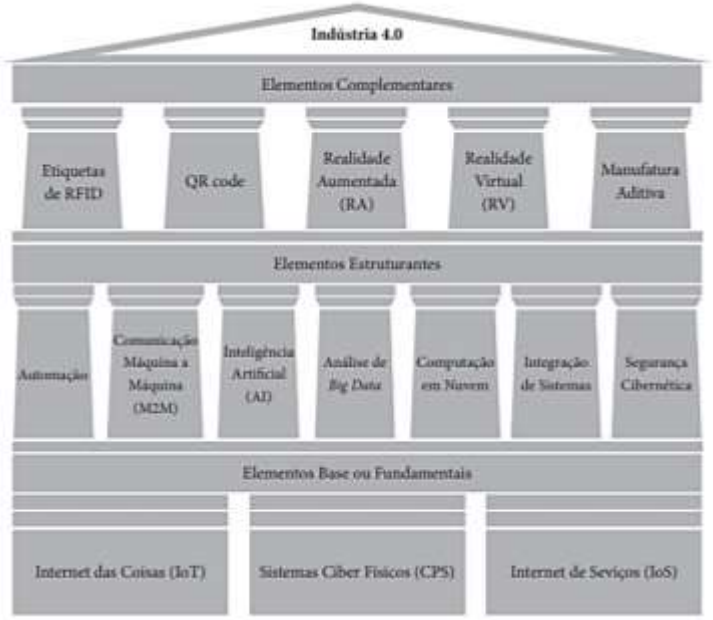

Figura 3: Elementos formadores da Indústria 4.0 (SACOMANO, 2018)

2.1.2.2 Bases fundamentais

Fundamentos, como o nome sugere, são os elementos essenciais para que o sistema se adapte à Indústria 4.0. Iremos nos aprofundar nos fundamentos da Indústria 4.0, onde nosso foco de pesquisa é a IoT.

2.1.2.2.1 Sistemas ciber-físicos (CPS)

Sistemas ciber-físicos (cyber-physical systems – CPS) são sistemas compostos por sensores e atuadores, controlados por software que monitoram uma série de dados como, por exemplo, processos industriais, mecânicos, térmicos, elétricos etc. Seus dados são enviados em tempo real permitindo que no mundo real possa-se atuar no sistema produtivo interferindo de maneira que for mais conveniente (SACOMANO, 2018).

2.1.2.2.2 Internet das coisas (IoT)

Existem diversos conceitos para definir IoT, podemos citar como sendo uma evolução da computação ligada a 4ª Revolução Industrial. Onde hardware, software, serviços e objetos físicos denominados “coisas” estão conectados à Internet (SACOMANO, 2018).

2.1.2.2.3 Internet de serviços (IoS)

O conceito de IoS complementa os demais elementos básicos da indústria 4.0. Pela IoS, novos serviços são disponibilizados. Serviços esses conectados à IoT, as quais são conectadas a CPS, ou, ainda, a IoT gerando serviços intrinsecamente ligados a ela. Em vez de comprar uma máquina, uma indústria pode comprar somente o serviço que ela oferece (SACOMANO, 2018).

2.2 A Internet das Coisas

ALVES et al. (2021, p. 100) consideram que IoT é como “um ecossistema ciber-físico de sensores e recursos interconectados, que permitem a tomada de decisões inteligentes.”

As redes IoT são compostas de grande quantidade de dispositivos, tendo essas capacidades e protocolos de comunicação normalmente diferentes entre si. Utilizando a Internet como base, os mesmos conseguem manter comunicação estando tanto próximos uns dos outros, como com grandes distâncias.

Figura 4: Esquema de uma versão simplificada de uma rede IoT (ROSA, 2021)

A Internet das Coisas (IoT) apresenta em sua estrutura diversas tecnologias para o transporte e gerenciamento de grande volume de dados. Essa comunicação é chamada de hiper conectividade entre máquinas (machine-to-machine — M2M). A M2M possibilita a comunicação totalmente autônoma entre dispositivos sem qualquer intervenção humana (MAGRANI, 2018).

O fluxo de dados que ocorre de um objeto inteligente para os consumidores finais contempla alguns passos, descritos a seguir (CARRION; QUARESMA, 2019).

Sensibilização de sensores: primeiramente, os sensores são sensíveis a determinadas grandezas físicas. E estão presentes em roteadores, smartphones, dispositivos vestíveis (wearables), termômetros, entre outros.

Centralização dos dados: após a ativação dos sensores, os dados são transportados das máquinas conectadas para serem armazenados e analisados em muitas aplicações, por meio de computação em nuvem.

Processamento lógico via software: após os dados serem guardados, aplicações são responsáveis pela análise e controle dos dados para fornecer serviços ao usuário final.

Usuário final: por fim, o usuário final tem acesso às informações processadas com recursos úteis e serviços.

2.2.1 Dispositivos e Tecnologia IoT

Os dispositivos IoT são compostos basicamente por 4 elementos: processador/memória, interface de comunicação, fonte de energia e sensores/atuadores (SANTOS et al., 2017).

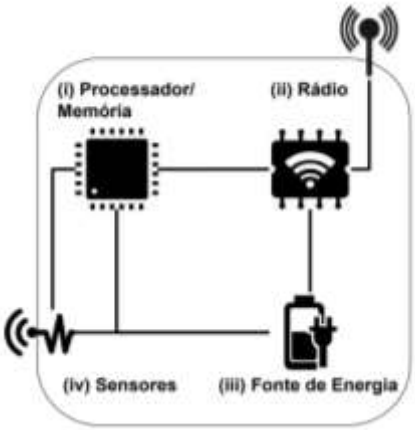

Figura 5: Elementos de um dispositivo IoT (SANTOS et al.,2017)

Processador/Memória: Esta unidade consiste em uma memória interna para dados e programas, um microcontrolador e um conversor analógico-digital que recebe os sinais dos sensores. Normalmente, a CPU usada para o dispositivo é a mesma CPU usada no sistema embarcado e geralmente não possui um poder computacional

muito alto. Geralmente, há uma memória flash externa que é usada como armazenamento secundário, como um "log" que contém dados. As características ideais dessas unidades são o baixo consumo de energia e os componentes devem ocupar o menor espaço possível (LEITE, 2019).

Interface de comunicação: A unidade consiste em canais de comunicação com ou sem fio, mais comumente sem fio. Neste caso, a maioria das plataformas usa rádios de baixo custo e baixo consumo. Como resultado, as comunicações são de curto alcance, consequentemente, recomenda-se fazer a adaptação necessária para que não ocorra a perda de dados.

Fonte de alimentação: Responsável por alimentar os componentes do dispositivo IoT. Normalmente, a fonte de alimentação consiste em uma bateria Conversores recarregáveis (ou não recarregáveis) e AC/DC, porém, existem diversas outras fontes de energia como baterias que guarnecem energia elétrica, solar, etc.

Sensores ou Atuadores: Esses elementos são responsáveis pelas interações com o ambiente em que o dispositivo está localizado. O sensor é responsável por lidar com grandezas físicas como temperatura, umidade, pressão, presença, etc. Enquanto o atuador é um dispositivo para gerar movimento, em resposta a comandos que podem ser manuais, elétricos ou mecânicos (LEITE, 2019).

2.2.2 Arquitetura de Rede IoT

Por ser uma tecnologia recente e em evolução, existem diversos modelos para definir uma rede IoT. Para fins acadêmicos, um dos principais modelos utilizados consiste em três camadas distintas: Application Layer (Camada de Aplicação), Network Layer (Camada de Rede) Perception Layer (Camada de Perceção). Cada camada representa um diferente conjunto de tecnologias, serviços e dispositivos (ROSA, 2021).

Camada de Aplicação: Esta camada se caracteriza por possuir o mais alto nível de arquitetura de rede e, portanto, nela estão presentes todas as interações com o usuário final do sistema, atuando como ponto de acesso aos serviços prestados pela rede, geralmente por meio de uma aplicação web.

Camada de Percepção: Representa os sistemas que formam o nível de arquitetura mais baixo da rede, ou seja, possui dispositivos de baixa memória que podem observar ou causar alterações no ambiente ao redor, como sensores, motores ou botões. Esta camada de segurança atua como controlador local de cada dispositivo e deve abordar questões como autenticação e análise do comportamento de sensores.

Camada de Rede: Camada responsável pela conexão e processamento entre as camadas do sistema por meio de tecnologias como Wi-Fi ou Bluetooth. Alguns autores, por exemplo, gostam de dividir essa camada em dois tópicos:

O primeiro tópico trata dos processos de rede e comunicação, principalmente pelo uso da Internet e outras tecnologias, como a de comunicação móvel, em conjunto com os níveis arquiteturais mais baixos da rede para coletar dados produzidos pela camada de percepção. O principal aspecto de segurança neste tópico é o gerenciamento da segurança da rede, com métodos para evitar congestionamentos e protocolos de autenticação de dispositivos os quais devem ser considerados (ROSA, 2021).

O segundo problema está relacionado ao próprio processamento dos dados coletados pela camada de rede, que permitirá a funcionalidade das atividades de alto nível da rede presentes na camada de aplicação. Esse serviço é normalmente fornecido por um serviço em nuvem e seu foco principal de segurança é proteger os dados processados por meio de criptografia, autenticação, detecção de intrusão e possível distorção de dados coletados e/ou processados (ROSA, 2021).

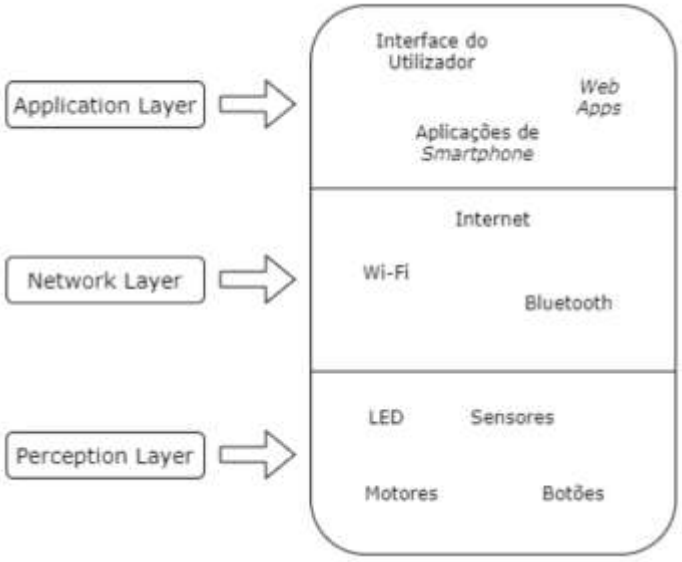

Figura 6: Camadas de uma rede IoT (ROSA, 2021)

2.3 IoT e Big Data

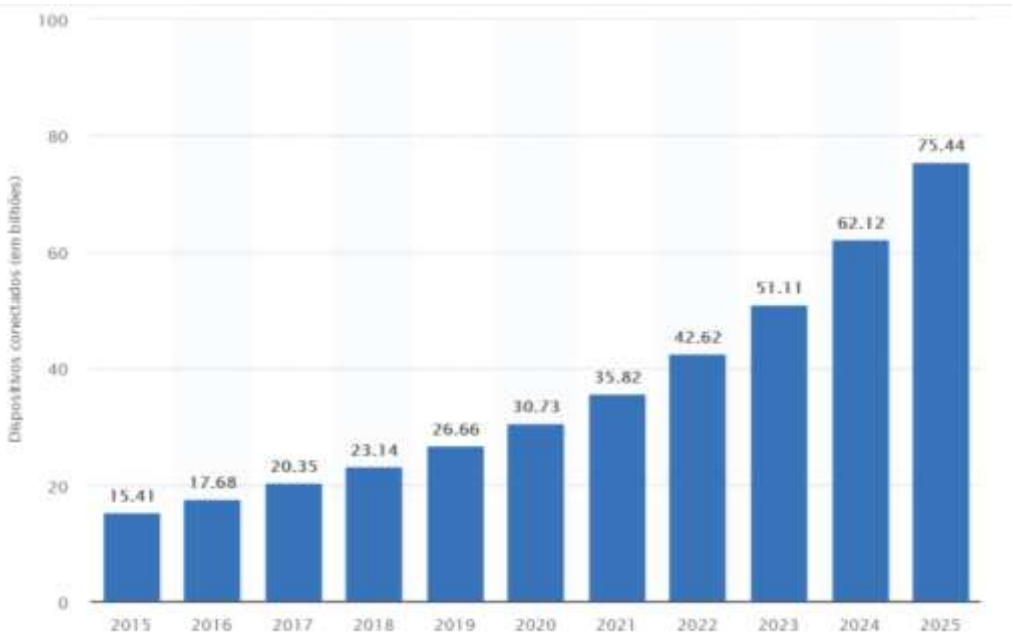

Figura 7: Dispositivos conectados de Internet das Coisas instalados em todo o mundo de 2015 a 2025 (em bilhões) (CARRION; QUARESMA, 2019)

A todo momento "coisas" estão sendo conectadas à Internet, com a capacidade de compartilhar, processar, armazenar e analisar grandes quantidades de dados entre elas. Essa prática combina conceitos de IoT com big data.

O termo Big data descreve uma grande quantidade de dados estruturados, semi estruturados ou não estruturados que podem potencialmente ser usados para obter informações. O primeiro atributo envolvido com big data é a quantidade cada vez maior de dados. Pesquisas recentes da Cisco estimam que, nos próximos anos, as medições em gigabytes serão superadas e os volumes de dados serão calculados em zettabytes ou mesmo yottabytes (CARRION; QUARESMA, 2019).

Outro atributo diz respeito à alta velocidade de geração, análise e visualização de dados. Esse recurso é aprimorado por diferentes dispositivos responsáveis por coletar e gerar dados em diferentes áreas. Por exemplo, as informações geradas por dispositivos IoT são bem diferentes daquelas obtidas nas redes sociais. Os conceitos de big data e de ciência de dados juntos podem implicar na capacidade de transformar dados brutos em gráficos para que a compreensão dos fenômenos possa ser demonstrada no contexto da tomada de decisão, a qual se torna cada vez mais baseada em dados (MAGRANI, 2018).

Figura 8: Relação entre IoT e Big Data (MORAIS,2018)

A Internet das Coisas reúne o mundo digital com o mundo físico, tornando os objetos parte de um sistema de informação. Com a IoT, podemos adicionar inteligência à infraestrutura física que molda nossa sociedade. A Internet das Coisas e seus objetos estão gerando dados o tempo todo e são um poderoso impulsionador do Big Data. Os motores a jato comerciais modernos geram cerca de 1 terabyte de dados por dia, que devem ser analisados para mantê-los funcionando o maior tempo possível (TAURION, 2013).

A tecnologia de big data permite que a informação seja processada antes de ser otimizada, racionalizada ou correlacionada. Com a análise avançada, você pode fazer e responder algumas perguntas de ciclo muito curto. O objetivo da aplicação do big data é detectar padrões nos dados e nas informações obtidas, possibilitando que as empresas desenvolvam produtos e serviços com base no perfil do público-alvo (MORAIS, 2018).

2.4 Possibilidades

Qualquer utensílio que você consiga imaginar pode entrar para o mundo da Internet das Coisas. Para nos dar mais conforto, produtividade, informação e praticidade em geral (ALMEIDA, 2019).

Gerando um cenário favorável a infraestruturas inteligentes, possibilitando serviços de melhor qualidade. Além de casas conectadas opções de implementação em diversos setores:

Hospitais e clínicas: Por exemplo, um paciente pode utilizar um aparelho capaz de medir frequência cardíaca ou pressão arterial e enviar os dados coletados em tempo real para o sistema que gerencia o exame.

Agricultura: Sensores em toda a plantação podem fornecer informações muito precisas sobre temperatura, umidade do solo, probabilidade de chuva, velocidade do vento e outros dados importantes para um bom desempenho do plantio. Além disso, os sensores acoplados aos animais podem ajudar a controlar os animais, informar seu histórico de vacinação e muito mais.

Fábrica: a IoT pode ajudar a medir a produtividade das máquinas em tempo real e também pode indicar quais partes da fábrica precisam de mais equipamentos ou suprimentos.

Lojas: As Prateleiras Inteligentes podem notificar em tempo real quando o estoque de determinado produto está começando a diminuir, quais produtos não estão

apresentando vendas satisfatórias (requer ações como reposicionamento ou criação de promoções) ou quando os produtos estão em promoção há mais tempo (ajudando nas vendas estratégia).

Transporte público: Os usuários podem encontrar a localização do ônibus que pretendem pegar por meio de um aplicativo de smartphone ou em uma tela instalada na estação. Há também sensores que podem informar as empresas sobre defeitos mecânicos em seus veículos ou como cumprir horários, indicando se a frota de ônibus precisa ser reforçada.

Logística: Dados de sensores montados em caminhões, contêineres ou até caixas individuais, combinados com informações como rotas ideais, os caminhões mais adequados para uma área específica e pedidos alocados em frotas ativas.

Serviços públicos: Sensores instalados em lixeiras podem ajudar as cidades a organizar a coleta de lixo; os veículos podem se conectar a centros que monitoram o tráfego e obtém a rota mais viável a qualquer momento ou ajudar o controle de tráfego a entender quais estradas são mais movimentadas em uma cidade.

2.5 Desafios da IoT

A Internet das Coisas aparecerá cada vez mais em nossas vidas. Embora a tecnologia esteja se espalhando rapidamente, ainda existem alguns obstáculos que podem retardar o desenvolvimento da IoT.

Os três maiores obstáculos são: implantação do IPv6, a alimentação dos sensores e a padronização (EVANS, 2011).

2.5.1 Implantação de IPv6: desafios de escalabilidade

Escalabilidade é a capacidade de um sistema, rede ou processo continuar a funcionar normalmente à medida que o tamanho do contexto ou a capacidade mudam para atender às necessidades do usuário. Portanto, temos que considerar fatores como o número de dispositivos conectados e a quantidade de dados gerados. A maioria dos dispositivos IoT usa processadores de baixa capacidade, muitos dos

quais são baseados em instruções de 32 bits e usam dispositivos IP versão 4, o que limita bastante a capacidade do dispositivo.

Em fevereiro de 2010, o mundo ficou sem endereços IPv4. Embora o impacto real não tenha sido percebido pelo público, essa situação tem o potencial de desacelerar o crescimento da IoT, pois pode haver bilhões de novos sensores exigindo endereços IP. Além disso, o IPv6 facilita o gerenciamento de rede e oferece recursos de segurança aprimorados devido aos recursos de configuração automática (EVANS, 2011).

Devemos lembrar que a mudança para o IP versão 6 implementa um modelo de endereçamento de 128 bits, o que nos permite ter vários dispositivos 2^{128} em vez dos 2^{32} do IPv4.

2.5.2 Alimentando sensores: desafios de energia, rede e comunicação

As redes utilizadas em IoT são compatíveis com as redes que utilizamos no nosso dia a dia, como WiFi, Bluetooth e NFC (Near Field Communication). No entanto, devido ao alcance limitado de tais redes, algumas aplicações contam com redes móveis como 3G e 4G ou redes similares no mercado.

A tecnologia 5G ajudará nesse sentido, fornecendo velocidades de transferência de dados muito altas, as redes 5G permitirão que cada dispositivo use apenas os recursos necessários em uma escala precisa. Evitando gargalos de rede e desperdício de energia (ALMEIDA, 2019).

Existem também soluções técnicas de hardware para problemas de potência dos sensores. Para que a IoT alcance todo o seu potencial, os sensores devem ser autossustentáveis. Em uma descoberta marcante, os cientistas anunciaram na 241ª Reunião Nacional e Exposição da Sociedade Americana de Química em março de 2011 um nanogerador comercialmente viável, uma compressão de dados, chips flexíveis que geram eletricidade (EVANS, 2011).

2.5.3 Protocolos padrão: Padronização e desafios de segurança

Vimos anteriormente que, quando se trata de arquitetura, existem vários estudos, mas não há padrões totalmente definidos aplicáveis aos dispositivos IoT, e a arquitetura não é a única área em que isso está acontecendo. As áreas de segurança, privacidade e comunicações também precisam de atenção.

2.5.4 Segurança da informação

Em IoT, precisamos de sistemas que alcancem resultados satisfatórios em um curto período de tempo. Mas quanto mais seguro é um sistema, mais degradamos seu desempenho. Sistemas com alto desempenho acabam sem todos os recursos de segurança e possuindo vulnerabilidades.

2.6 Conceitos de Segurança da Informação

Os próximos conceitos estão presentes nesse TCC para melhor entendimento dos experimentos quanto ao que se trata de Segurança da Informação.

2.6.1 Os Pilares da Segurança e Privacidade

Primeiramente deve-se alinhar três conceitos muito importantes: Segurança, Proteção e Privacidade.

A **privacidade** das informações é o direito de ter algum controle sobre como suas informações são coletadas e usadas. (ALVES et al, 2021)

Segundo Alves et al. (2021 apud William Wulf),

> “a proteção é identificada como um mecanismo, enquanto a **segurança** é identificada como uma política. [...] Entendendo assim o mecanismo como se fosse o *modus operandi*, o como fazer, de maneira que a política seria o ‘estratégico’, o que fazer.”

Percebemos que não se tem privacidade sem segurança e proteção. Com esses conceitos alinhados, partimos para os atuais seis pilares da segurança e da proteção de dados:

Confiabilidade: Proteção da informação contra o acesso não autorizado. Como forma de garantir que as informações trocadas entre os dispositivos sejam transmitidas de forma segura, garantindo que apenas pessoas autorizadas possam acessar essas informações (LEITE, 2019). Em IoT, a confidencialidade deve abranger duas áreas principais: A primeira é garantir a transferência segura de informações entre diferentes sistemas de comunicação, que é o que conhecemos como dados em movimento. A segunda é como garantir que os dados armazenados ou em repouso estejam protegidos (ALVES et al., 2021).

Integridade: Manter dados com suas características originais e protegidos contra a alteração não autorizada envolve garantir que alguma ação realizada pelo sistema ocorra de forma holística, ou seja, durante o processamento, o fluxo de dados e informações seja coerente e que não tenham sido alterados intencionalmente. Os controles de integridade garantem que a operação ou o estado do equipamento permaneça saudável durante todo o processo.

Disponibilidade: Assegura que as informações estejam à disposição quando necessárias. Usando servidores de backup em uma arquitetura hot standby, ou seja, tendo a mesma infraestrutura e aplicativos prontos para ficar online caso o sistema principal falhe. Essa arquitetura de alta disponibilidade geralmente traz custos adicionais devido à necessidade de replicar a infraestrutura, o que muitas vezes acaba sendo caro. Os sistemas mais confiáveis, como aeronaves e naves espaciais, geralmente possuem vários sistemas redundantes e caros para alcançar um alto nível de confiabilidade.

Autenticidade: Garante a verdadeira autoria da informação.

Autorização: O conceito de autorização em um plano mais amplo e incorpora também a autenticação, a maneira de garantir que o usuário ou o sistema do dispositivo seja autêntico, e autorizado a utilizar o sistema do modo correto e íntegro. Atualmente, a gestão da identidade do usuário é um dos grandes desafios da internet, uma vez que é necessária a implementação de processos que garantam a autenticação do usuário de forma constante.

Irretratabilidade ou Não Repúdio: Assegura que indivíduo ou entidade não negue autoria de uma ação. O não repúdio também é um serviço de segurança muito importante. A ideia do não repúdio é a de mecanismos que permitam ao sistema identificar quem efetivamente realizou uma determinada operação, e que este não possa negar que a tenha realizado. Existem várias tecnologias e métodos de não repúdio, mas um exemplo clássico e que podemos trazer para o nosso dia a dia é o da câmera no caixa eletrônico de um banco. Quando um cliente realiza um saque, a câmera filma. Se por algum motivo esse cliente negar que tenha realizado determinado saque, existe o registro da câmera comprovando que o saque foi realizado.

Legalidade ou Responsabilidade: Determina a obrigação de se responsabilizar por ações ocorridas no trato dos dados.

Portanto, um ataque à confiabilidade de um sistema está relacionado à obtenção de informação não autorizada. Um ataque à integridade executa uma modificação não autorizada da informação. E um ataque contra a disponibilidade do sistema ocorre quando o mesmo torna-o inacessível a outros usuários que tenham necessidade e autorização para acessá-lo. Também existem ataques que focam no controle das políticas de segurança, e assim, põem em risco todos os pilares citados, por dar privilégios não legítimos a usuários mal intencionados (COUTINHO; VASQUEZ; MACHADO, 2006).

Iremos explorar vulnerabilidades que comprometem a disponibilidade do protótipo IoT.

2.7 Riscos relacionados a IoT

Existem vários tipos e modelos de dispositivos IoT e, com isso, muitos riscos a serem levados em consideração. Exploraremos aqui os riscos inerentes ao protótipo aqui apresentado.

2.7.1 Dispositivos que não podem ser acessados e conectados por um curto período de tempo

Existem vários tipos de dispositivos IoT que nem sempre estão conectados. Isso pode representar um benefício contra ameaças externas, no entanto, introduz uma vulnerabilidade muito grande. Esses dispositivos, por não serem de fácil acesso, normalmente possuem sistemas de controle de acesso fracos, muitas vezes usam uma senha de acesso padrão ou até mesmo nenhuma senha. Estes dispositivos normalmente são mais antigos e devido ao seu baixo poder de processamento, se torna difícil usar criptografia para proteger o canal de comunicação, o que representa mais um risco, dessa vez relacionado à privacidade dos dados trocados com o dispositivos (MORAES; HAYASHI, 2021).

2.7.2 Sem perímetro

Quando falamos sobre o uso de dispositivos IoT, temos que levar em conta que grande parte é composta por dispositivos móveis. Estes, por sua vez, podem ser conectados por meio de uma conexão de dados (3G, 4G), ou até mesmo uma rede Wi-Fi diferente, dificultando o estabelecimento de limites de rede para controle. Além disso, muitos dispositivos estão diretamente conectados à nuvem. Quando há um perímetro, há um firewall, que funciona como um dispositivo de controle de acesso à Internet. Toda a comunicação precisa passar por ele. Dada a arquitetura de segurança, garantir o controle de acesso a dispositivos que não são implementados localmente é bastante complexo. Eles são limitados em sua capacidade de criptografar, o que também nos impede de criar VPNs (Redes Privadas Virtuais) para restringir e controlar o tráfego através do dispositivo (MORAES; HAYASHI, 2021).

2.7.3 Dispositivos sem atualização

Muitos desses dispositivos não são atualizados ao longo de sua vida útil, sendo assim, podem criar vulnerabilidades no sistema operacional, no gerenciamento da rede, nos protocolos de comunicação e, por fim, deixar brechas exploráveis. Muitos fabricantes não geram atualizações por medo de quebrar certos recursos. Dispositivos que não estão atualizados são vulneráveis a ataques, como estouros de buffer, que exploram suas vulnerabilidades e permitem o acesso a elas (MORAES; HAYASHI, 2021).

2.7.4 Ataques em portas de comunicação

Muitos dispositivos têm portas de comunicação inerentemente abertas, além disso não ser sempre necessário, é um risco significativo e pode ser uma maneira de entrar e explorar a vulnerabilidade (MORAES; HAYASHI, 2021).

2.7.5 Ataque de negação de serviço

Como mencionado, muitos desses dispositivos usam senhas de texto simples, que podem ser facilmente comprometidas. Alguns anos atrás, grupos de hackers tomaram conhecimento dessas vulnerabilidades e começaram a invadir milhões de dispositivos IoT em todo o mundo para ataques DDoS (Distributed Denial of Service, ou em português, Negação de Serviço Distribuída) (MORAES; HAYASHI, 2021).

2.8 Ataques de negação de serviço (DoS)

Um ataque de negação de serviço (DoS – Denial of Service) é uma técnica de acessibilidade de negação de serviço muito simples. Este ataque envolve sobrecarregar o alvo com pacotes grandes. Este ataque não compromete as informações ou privacidade do alvo, apenas impede o acesso a ele e, ao enviar vários pacotes, o alvo não consegue combater o invasor, impedindo que o servidor atenda a usuários legítimos.

2.9. Conceitos de Redes Orientadas à Conexão

Em uma rede orientada a conexão, para que haja comunicação entre um computador e um terminal, deve ocorrer primeiro um estabelecimento de conexão, que é chamado de handshake (negociação). Uma vez que a conexão é estabelecida, o estado de transferência de dados é alcançado. Os dados do usuário são trocados com base em protocolos pré-estabelecidos. Após a transferência de dados, a conexão é encerrada (FEDELI; POLLONI; PERES, 2010).

Os conceitos a seguir estão presentes nesse TCC para melhor entendimento dos experimentos quanto ao que se trata de redes orientadas à conexão.

2.9.1 Three-way Handshake

É um mecanismo especificado na documentação do protocolo TCP para se estabelecer uma conexão. O processo de estabelecimento de conexões é realizado por ele da seguinte forma:

1. O cliente envia um pacote de estabelecimento de conexão denominado SYN (Synchronize flag ou Flag de sincronização);
2. Caso aceite a conexão, o servidor responde com um pacote de reconhecimento SYN/ACK;
3. O cliente responde com uma confirmação ACK (Acknowledgement flag ou Flag de reconhecimento) (SCHMITT; PERES; LOUREIRO, 2013).

2.9.1.1 Round Trip Time

Segundo Peres, Loureiro e Schmitt (2014, p. 25),

> “O RTT (Round Trip Time ou “tempo médio de viagem”) é o tempo necessário para enviar uma mensagem entre as estações. A medida do RTT é feita pela primeira vez durante o estabelecimento da conexão e é atualizada durante toda a duração da conexão, tendo em vista que as condições (e, consequentemente, o próprio RTT) da rede variam neste período.”

Como o primeiro RTT é medido durante a primeira vez que a conexão é estabelecida, caso o servidor não responda ao primeiro SYN, o cliente deverá aguardar por um tempo limite relativamente grande já que o RTT ainda não é conhecido. Da mesma forma, se o servidor envia o SYN/ACK e não recebe o ACK como resposta, ficará aguardando (em alguns casos, por até 4 minutos). Isso possibilita um ataque conhecido como **SYN Flooding** (PERES; LOUREIRO; SCHMITT, 2014), o qual será exemplificado em um de nossos experimentos.

2.9.2 Four-way-Handshake

O handshake de quatro vias é um protocolo de autenticação de rede estabelecido pelo IEEE-802.11i que aborda padrões de configuração para construção e uso de WLANs (Wireless Local Area Networks).

Para entendê-lo melhor, primeiramente precisamos entender algumas chaves essenciais na Figura 9:

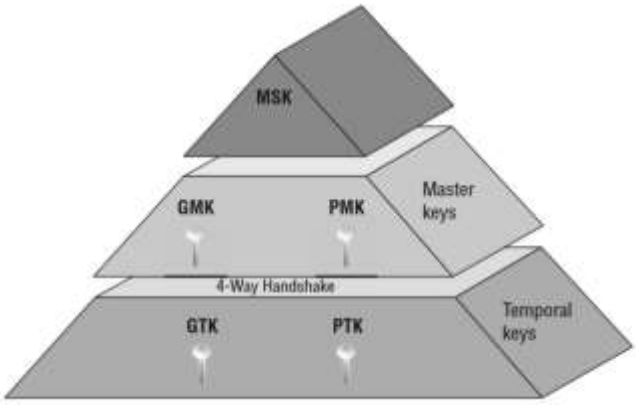

Figura 9: Chaves utilizadas no 4-way handshake (HAIDER, 2019)

A chave de primeiro nível , MSK (Chave Mestra de Sessão) é gerada durante o processo de autenticação 802.1X/EAP. As chaves de segundo nível são geradas a partir do MSK: GMK (Chave Mestra de Grupo) e PMK (Chave Mestra em Pares). PMK é usada para gerar PTK (Chave Transitória em Pares) e GMK é usada para criar GTK (Chave Temporal de Grupo). As chaves de terceiro nível são as chaves reais usadas para criptografia de dados (HAIDER, 2019).

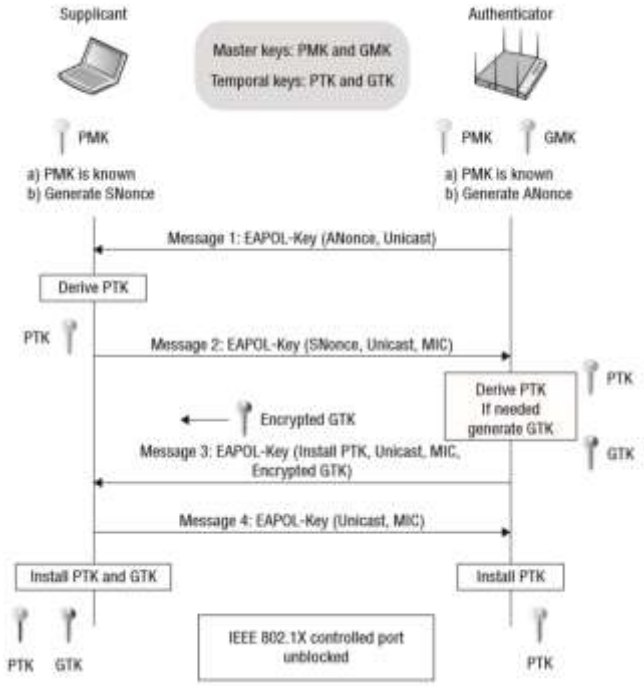

Figura 10: 4-way handshake (HAIDER, 2019)

Mensagem 1: O ponto de acesso envia mensagens do tipo EAPOL (Extensible Authentication Protocol) com ANonce (número aleatório) para o dispositivo para gerar PTK. O Cliente possui a PMK, o SNonce e seu próprio endereço MAC. Uma vez que recebe o ANonce do autenticador, ele tem todas as entradas para criar o PTK.

PTK = PRF (PMK + Anonce + SNonce + Mac (Autenticador) + Mac (Cliente))

Mensagem 2: Uma vez que o dispositivo tenha criado seu PTK, ele envia o SNonce que é necessário para o autenticador gerar sua PTK. O dispositivo envia esta EAPOL com SNonce e MIC. O MIC é uma verificação de integridade da mensagem e serve para o ponto de acesso verificar se esta mensagem foi corrompida ou modificada. Uma vez recebido pelo autenticador, o SNonce pode gerar PTK também para criptografia de tráfego unicast (ponto-a-ponto).

Mensagem 3: Esta mensagem é enviada do autenticador para o dispositivo cliente contendo GTK. O ponto de acesso cria o GTK sem o envolvimento do cliente.

Mensagem 4: A quarta e última mensagem EAPOL será enviada do cliente para o autenticador apenas para confirmar que as chaves foram instaladas.

Para hackear Wi-Fi sem senha (PSK), precisamos capturar o handshake entre o cliente e o ponto de acesso ao qual queremos nos conectar. Podemos forçar o cliente a se reconectar simplesmente atacando o cliente com uma mensagem de desautenticação e capturando o handshake (HAIDER, 2019).

3 Projeto Prático

3.1 Descrição do cenário

Composto por:

Roteador TP-LINK TL-WR740N;

Protótipo de IoT;

Máquina virtual com Kali Linux;

Adaptador USB wireless.

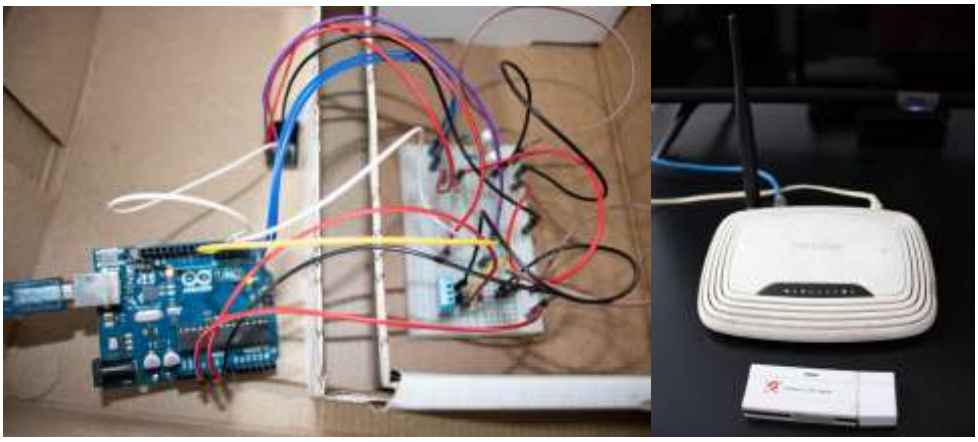

Figura 11: Fotos do protótipo IoT, roteador e adaptador wireless
Fonte: Elaboração própria

Montamos este cenário em um local onde roteador, protótipo e máquina (rodando o Kali Linux virtualmente) estavam próximos. O protótipo IoT é o gerador de tráfego na rede, conectado ao ponto de acesso no roteador, tráfego necessário para a captura durante a execução dos experimentos. Escolhemos o roteador TL-WR740N por dar suporte aos protocolos utilizados nos experimentos (TCP e 802.11).

Utilizamos o sistema Kali Linux, rodando em máquina virtual, já que o mesmo possui as ferramentas necessárias para a realização dos experimentos. E o adaptador

USB Wireless foi utilizado para complementar funcionalidades necessárias para os experimentos e que a máquina não possuía.

3.2 Ferramentas utilizadas

3.2.1 ThingSpeak

O ThingSpeak é um serviço de plataforma de análise de IoT que permite agregar, visualizar e analisar fluxos de dados ao vivo na nuvem. É possível enviar dados para o ThingSpeak de seus dispositivos, criar visualização instantânea de dados ao vivo e enviar alertas (THINGSPEAK, 2022).

Figura 12: Página do projeto no ThingSpeak (medições de 14 de maio)
https://thingspeak.com/channels/1692858

3.2.2 Kali Linux

Kali Linux é um projeto de código aberto baseado na distribuição Debian Wheezy Linux, mantido pela Offensive Security, e vem pré-instalado com diversos programas para testes de invasão e análise forense. O Kali Linux pode ser executado localmente, inicializar a partir de um CD ou USB ou executado em uma máquina virtual (KALI LINUX, 2022).

3.2.3 AIRCRACK-NG

É um software de código aberto que possui diversas ferramentas de linha de comando para auditoria de redes 802.11 (AIRCRACK-NG, 2022). Aircrack-ng trata-se de um programa para quebrar as cifras dos protocolos de segurança utilizados no padrão 802.11. Ele implementa vários tipos de ataques, incluindo aqueles que usam

dicionários. O kit apresenta um sniffer de pacotes, ferramentas de análise e funciona com qualquer placa de rede que suporte monitoramento (LÜDTKE, 2015).

3.2.4 AIRMON-NG

É um script que pode ser usado para ativar o modo de monitor em interfaces sem fio. Esta ferramenta responde com algumas informações sobre o adaptador sem fio, incluindo o chipset, driver e eventuais processos que podem entrar em conflito, ou serem prejudiciais durante o uso da ferramenta (AIRCRACK-NG, 2022).

3.2.5 AIREPLAY-NG

É uma ferramenta que pode ser usada para gerar ou acelerar o tráfego no AP. Existem diferentes ataques disponíveis: desautenticação do cliente para recolher dados de handshake WPA, autenticação forjada, repetição interativa de pacotes, criação manual de pacotes de solicitação ARP e reinjeção de solicitação ARP. O Aireplay-ng pode adquirir pacotes de duas fontes: uma transmissão ao vivo de pacotes ou um arquivo PCAP pré-capturado. Os arquivos PCAP são um tipo de arquivo padrão associado a ferramentas de captura de pacotes, como libpcap e winpcap. Wireshark e TCPdump também usam arquivos PCAP (AIRCRACK-NG, 2022).

3.2.6 HPING3

A ferramenta hping é um montador/parser de pacotes TCP/IP orientado por linha de comando. Ele suporta os protocolos TCP, UDP, ICMP e RAW-IP. Possui modo traceroute, capacidade de enviar arquivos entre canais e muitos outros recursos. Para tarefas de verificação de intrusão e auditoria de segurança, a ferramenta hping é uma das melhores ferramentas da web. Agora em sua terceira geração, o hping tornou-se o programa de escolha para a criação de pacotes IP, frequentemente usado para testar firewalls e sistemas de detecção de intrusão. Também pode ser usado para testar a segurança da rede e do host. Por exemplo: varredura de portas, auditoria de implementação TCP/IP e geração de tráfego de dados intensivo. Como o hping pode ser usado para manipular todos os campos, propriedades e tipos de protocolo presentes no conjunto de protocolos baseado em TCP/IP, alguns usuários se referem a ele como um "modelador de pacotes". Um dos principais usos para ele pelos invasores é realizar testes de limite. O aplicativo pode

ser usado para gerar tráfego para testar se um firewall pode bloquear pacotes manipulados e se a estrutura de detecção de intrusão pode identificar anomalias e problemas. Programas como o hping3 são ótimos para gerar esse tráfego "incomum" (FIRMINO, 2019).

3.2.7 Wireshark

Wireshark é um analisador de pacotes de rede. O qual apresenta dados de pacotes capturados com o máximo de detalhes possível. O Wireshark está disponível gratuitamente, é de código aberto e é um dos melhores analisadores de pacotes disponíveis atualmente (WIRESHARK, 2022).

3.3 Protótipo IoT

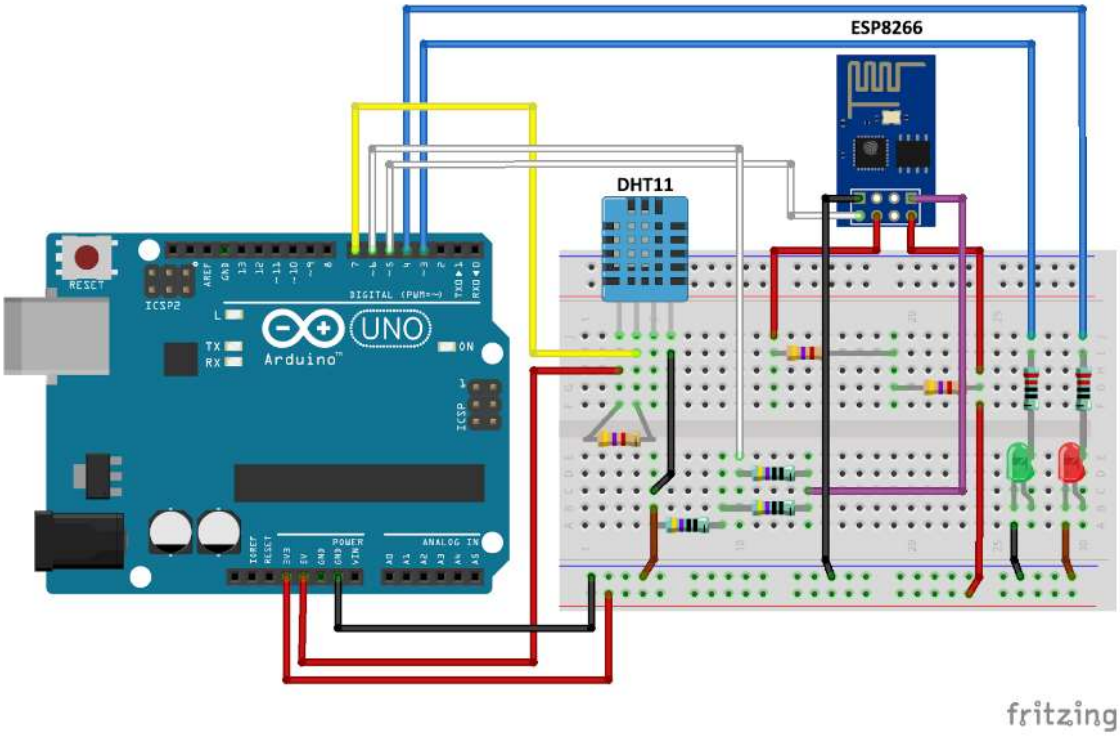

Figura 13: Esquema de montagem do Circuito IoT para Monitoramento de Temperatura e Umidade

Fonte: Elaborado utilizando o site https://fritzing.org/

3.3.1. Circuito IoT para Monitoramento de Temperatura e Umidade

Componentes:
Arduino UNO;
Sensor DHT11 (Temperatura e Umidade);
Modulo ESP8266 (WiFi);
LEDs Vermelho e Verde;
Resistores: 220Ω, 470Ω, 4.7kΩ.

Conexões:
Fios pretos e marrons: GND - Terra;
Fios Vermelhos: Alimentação 3.3V e 5V;
Fio amarelo: DHT11 - Comunicação com a porta 7 do Arduino;
Fios Azuis: LEDs - Comunicação com portas 3 e 4 do Arduino;
Fios Brancos: Comunicação TX e RX do Arduino com o RX e TX do ESP;
Fio Roxo: Divisor de tensão.

3.3.2 Sobre os componentes

3.3.2.1 Arduino UNO

O Arduino é uma plataforma flexível de prototipagem eletrônica amplamente utilizada. Seu principal objetivo é tornar o acesso a protótipos eletrônicos mais fácil, barato e flexível. A versão simples utiliza um microcontrolador Atmel AVR com suporte de entrada/saída embutido e uma linguagem de programação baseada em C/C++. Na área de eletrônica é possível criar uma grande variedade de projetos, desde as mais simples até aplicações IoT, robótica, sistemas de automação residencial ou industrial, alarmes e muito mais.

3.3.2.2 Sensor DHT11

Este sensor inclui um componente medidor de umidade e um componente NTC (Negative Temperature Coefficient ou Coeficiente Negativo de Temperatura) para temperatura, ambos conectados a um controlador de 8-bits. As leituras do sensor são

enviadas usando apenas um único fio de barramento (MOUSER ELECTRONICS, 2022)

3.3.2.3 Módulo ESP8266

O módulo WiFi ESP8266 é um SOC (System On Chip ou, em português, Sistema em Chip) com protocolo TCP/IP integrado que consegue dar a um microcontrolador acesso a sua rede WiFi. O ESP8266 é capaz tanto de hospedar uma aplicação quanto realizar funções de redes WiFi a partir de outro processador de aplicação (AI-THINKER, 2015).

– Tensão de operação: 3,3V;
– Suporte à redes: 802.11 b/g/n
– Alcance: 90m aprox.
– Comunicação: Serial (TX/RX)
– Suporta comunicação TCP e UDP
– Modo de segurança: OPEN/WEP/WPA_PSK/WPA2_PSK/WPA_WPA2_PSK

3.3.2.4 Comandos AT

Os comandos AT apresentados na figura 13, são descendentes direto do chamado "padrão Hayes" de 1981, para permitir que computadores pessoais interajam com conexões telefônicas controlando diretamente o modem.

A definição básica de Hayes (incluindo o prefixo AT, que significa "atenção") existe nas linguagens de comando de muitos dispositivos modernos, incluindo muitos periféricos de computador, dificilmente alguém que não seja um programador de driver ou firmware precisa vir conhecê-los.

No entanto, quando estamos programando o Arduino para enviar comandos ao módulo ESP8266 WiFi, estamos fazendo o papel de um programador de firmware, por isso é necessário dominar a sintaxe dos comandos AT aceitos pela

Série ESP (BR-ARDUINO, 2015).

Function	AT Command	Response
Working	AT	OK
Restart	AT+RST	OK [System Ready, Vendor:www.ai-thinker.com]
Firmware version	AT+GMR	AT+GMR 0018000902 OK
List Access Points	AT+CWLAP	AT+CWLAP +CWLAP:(4,"RochefortSurLac",-38,"70:62:b8:6f:6d:58",1) +CWLAP:(4,"LiliPad2.4",-83,"f8:7b:8c:1e:7c:6d",1) OK
Join Access Point	AT+CWJAP? AT+CWJAP="SSID","Password"	Query AT+CWJAP? +CWJAP:"RochefortSurLac" OK
Quit Access Point	AT+CWQAP=? AT+CWQAP	Query OK
Get IP Address	AT+CIFSR	AT+CIFSR 192.168.0.105 OK
Set Parameters of Access Point	AT+ CWSAP? AT+ CWSAP= <ssid>,<pwd>,<chl>, <ecn>	Query ssid, pwd chl = channel, ecn = encryption
WiFi Mode	AT+CWMODE? AT+CWMODE=1 AT+CWMODE=2 AT+CWMODE=3	Query STA AP BOTH
Set up TCP or UDP connection	AT+CIPSTART=? (CIPMUX=0) AT+CIPSTART = <type>,<addr>,<port> (CIPMUX=1) AT+CIPSTART= <id><type>,<addr>, <port>	Query id = 0-4, type = TCP/UDP, addr = IP address, port= port
TCP/UDP Connections	AT+ CIPMUX? AT+ CIPMUX=0 AT+ CIPMUX=1	Query Single Multiple
Check join devices' IP	AT+CWLIF	
TCP/IP Connection Status	AT+CIPSTATUS	AT+CIPSTATUS? no this fun
Send TCP/IP data	(CIPMUX=0) AT+CIPSEND=<length>; (CIPMUX=1) AT+CIPSEND= <id>,<length>	
Close TCP / UDP connection	AT+CIPCLOSE=<id> or AT+CIPCLOSE	
Set as server	AT+ CIPSERVER= <mode>[,<port>]	mode 0 to close server mode; mode 1 to open; port = port
Set the server timeout	AT+CIPSTO? AT+CIPSTO=<time>	Query <time>0~28800 in seconds
Baud Rate*	AT+CIOBAUD? Supported: 9600, 19200, 38400, 74880, 115200, 230400, 460800, 921600	Query AT+CIOBAUD? +CIOBAUD:9600 OK
Check IP address	AT+CIFSR	AT+CIFSR 192.168.0.106 OK
Firmware Upgrade (from Cloud)	AT+CIUPDATE	1. +CIPUPDATE:1 found server 2. +CIPUPDATE:2 connect server 3. +CIPUPDATE:3 got edition 4. +CIPUPDATE:4 start update
Received data	+IPD	(CIPMUX=0): + IPD, <len>: (CIPMUX=1): + IPD, <id>, <len>: <data>
Watchdog Enable*	AT+CSYSWDTENABLE	Watchdog, auto restart when program errors occur: enable
Watchdog Disable*	AT+CSYSWDTDISABLE	Watchdog, auto restart when program errors occur: disable

Figura 14: Tabela de comandos AT(BR-ARDUINO, 2015).

3.3.2.5. LEDs

Utilizados neste circuito apenas para reagir ao sucesso (led verde) ou falha (led vermelho) do envio de dados por comando AT.

3.4 Código

```
#include "DHT.h" // Biblioteca do sensor
#include <SoftwareSerial.h> // Biblioteca para emular conexão serial

#define DHTPIN 7      // Pino conectado ao Sensor DHT
#define DHTTYPE DHT11 // Modelo do sensor DHT

#define RX 6
#define TX 5

#define GREEN_LED 3
#define RED_LED 4

String AP = "TESTE"; // Nome da rede onde o protótipo se conectará
String PASS = "0123456789"; // Senha da rede
String API = "R5W7L8Y4T95UR75Y"; // Chave do API Thingspeak
String HOST = "api.thingspeak.com";
String PORT = "80";   //Porta de comunicação com o Thingspeak

int countTrueCommand;
int countTimeCommand;

boolean found = false;
float valSensorT = 1;
float valSensorH = 1;
```

Figura 15: Trecho do código implementado no protótipo IoT: Declaração de variáveis

Fonte: Elaboração própria

Declaração de variáveis e importação das bibliotecas DHT.h (Sensor de temperatura e Umidade) e SoftwareSerial. A biblioteca SoftwareSerial é usada para emular portas seriais RX e TX pois o Arduino UNO só possui os pinos 0 e 1 para comunicação serial, as quais são utilizadas para comunicação com o computador. Assim, precisamos emular portas seriais via software para fazer as conexões com o módulo ESP.

Como é possível notar, temos a senha da rede declarada em texto claro dentro do código, uma grande vulnerabilidade de alguns dispositivos IoT e que foi citada anteriormente no tópico sobre riscos relacionados à IoT.

```
void setup() {
  dht.begin(); //Iniciando sensor
  Serial.begin(9600); //baud-rate do monitor serial
  esp8266.begin(115200); //baud-rate do esp8266
```

Figura 16: Trecho do código implementado no protótipo IoT: Definição do baud-rate

Fonte: Elaboração própria

Baud Rate designa uma medida de velocidade de tráfego eletrônico de dados que mede o número de sinais elétricos transmitidos por unidade de tempo. No caso, temos por padrão 115200 baud rate para o ESP8266 e utilizamos 9600 baud rate para envio ao monitor serial.

```
void sendCommand(String command, int maxTime, char readReplay[]) {
  Serial.print(countTrueCommand);
  Serial.print(". COMANDO AT => ");
  Serial.print(command);
  Serial.print(" ");
  while(countTimeCommand < (maxTime*1))
  {
    esp8266.println(command);//at+cipsend
    if(esp8266.find(readReplay))//ok
    {
      found = true;
      break;
    }

    countTimeCommand++;
  }

  if(found == true)
  {
    Serial.println("[OK]");
    countTrueCommand++;
    countTimeCommand = 0;
    Okay();
  }

  if(found == false)
  {
    Serial.println("[FALHA]");
    countTrueCommand = 0;
    countTimeCommand = 0;
    Erro();
  }

  found = false;
 }
```

Figura 17: Trecho do código implementado no protótipo IoT: método para comandos AT

Fonte: Elaboração própria

Método padrão utilizado para enviar comandos AT e reconhecer erros.

```
sendCommand("AT",5,"OK");
sendCommand("AT+CWMODE=1",5,"OK");
sendCommand("AT+CWJAP=\""+ AP +"\",\""+ PASS +"\"",20,"OK");
```

Figura 18: Trecho do código implementado no protótipo IoT: Comandos AT

Fonte: Elaboração própria

Comandos AT para iniciar a comunicação, ativar modo WiFi STA e enviar login e senha de rede para conectar o protótipo. O **Modo STA** ou Modo Station é ativado para informar ao módulo ESP que ele será usado para conectar-se a uma rede WiFi.

```
float getSensorDataT(){
  delay(2500); // Aguarda 2.5 segundos entre as medições do sensor
  float t = dht.readTemperature(); // lendo temperatura

  // Caso haja erro de leitura do sensor de temperatura
  if (isnan(t)) {
    Erro();
    Serial.println(F("Falha ao ler o sensor de temperatura!"));
    return;
  }
  Serial.println("");
  Serial.print(F("===== Temperatura: "));
  Serial.print(t);
  Serial.println(F("°C ====="));
  Serial.println("");
  return t;
}
```

Figura 19: Trecho do código implementado no protótipo IoT: Valor de temperatura

Fonte: Elaboração própria

Método que lê e retorna o valor de temperatura do sensor DHT11. Em caso de falha de leitura (isnan = is not a number, não é um número) retorna um aviso para o monitor serial e acende o LED vermelho.

```
float getSensorDataH(){

  delay(2500); // Aguarda 2.5 segundos entre as medições do sensor
  float h = dht.readHumidity(); // lendo umidade

  // Caso haja erro de leitura do sensor de temperatura
  if (isnan(h)) {
  //  Erro();
    Serial.println(F("Falha ao ler o sensor de umidade!"));
    return;
  }
  Serial.println("");
  Serial.print(F("===== Umidade: "));
  Serial.print(h);
  Serial.println(F("% ====="));
  Serial.println("");
  return h;
}
```

Figura 20: Trecho do código implementado no protótipo IoT: Valor de Umidade

Fonte: Elaboração própria

Método que lê e retorna o valor de umidade do sensor DHT11. Em caso de falha de leitura (isnan = is not a number, não é um número) retorna um aviso para o monitor serial e acende o LED vermelho.

```
// LED em caso de erro
void Erro(){
    digitalWrite(RED_LED, HIGH);
    delay(1000);
    digitalWrite(RED_LED, LOW);

}
// LED em caso de sucesso
void Okay(){
    digitalWrite(GREEN_LED, HIGH);
    delay(1000);
    digitalWrite(GREEN_LED, LOW);
```

Figura 21: Trecho do código implementado no protótipo IoT: Atividade dos LEDs

Fonte: Elaboração própria

Atividade dos LEDs em caso de erro e de sucesso.

```
void loop() {
 valSensorT = getSensorDataT();
 String getDataT = "GET /update?api_key="+ API +"&"+ "field1" +"="+String(valSensorT);
 delay(1000);
 sendCommand("AT+CIPMUX=1",5,"OK");
 sendCommand("AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+ PORT,15,"OK");
 sendCommand("AT+CIPSEND=0," +String(getDataT.length()+4),5,">");
 esp8266.println(getDataT);
 delay(2000);
 sendCommand("AT+CIPCLOSE=0",5,"OK");

 valSensorH = getSensorDataH();
 String getDataH = "GET /update?api_key="+ API +"&"+ "field2" +"="+String(valSensorH);
 delay(1000);
 sendCommand("AT+CIPMUX=1",5,"OK");
 sendCommand("AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+ PORT,15,"OK");
 sendCommand("AT+CIPSEND=0," +String(getDataH.length()+4),5,">");
 esp8266.println(getDataH);
 delay(2000);
 sendCommand("AT+CIPCLOSE=0",5,"OK");
 countTrueCommand++;
}
```

Figura 22: Trecho do código implementado no protótipo IoT: Método loop

Fonte: Elaboração própria

Método em loop para receber os dados do sensor de temperatura, enviar esses dados para a API ThingSpeak no campo 1, comandos AT para iniciar conexão TCP com site da API ThingSpeak na porta 80, enviar valor do sensor de temperatura e finalizar a conexão. Em seguida faz o mesmo para o sensor de umidade mudando apenas o campo da API ThingSpeak para 2.

Parâmetros necessários para cada comando AT:

sendCommand("AT+CIPMUX=1",5,"OK");

Comando, Tempo máximo de resposta e Resposta

sendCommand("AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+ PORT,15,"OK");

Comando = ID, Tipo (TCP ou UDP), Endereço Host, Porta e Resposta

sendCommand("AT+CIPSEND=0," +String(getDataT.length()+4),5,">");

Comando=ID, Tamanho da string, Tempo máximo de resposta e Resposta

sendCommand("AT+CIPCLOSE=0",5,"OK");

Comando = ID, Tempo de resposta, Resposta

```
0. COMANDO AT => AT [OK]
1. COMANDO AT => AT+CWMODE=1 [OK]
2. COMANDO AT => AT+CWJAP="TESTE","0123456789" [OK]

===== Temperatura: 27.30°C =====

3. COMANDO AT => AT+CIPMUX=1 [OK]
4. COMANDO AT => AT+CIPSTART=0,"TCP","api.thingspeak.com",80 [OK]
5. COMANDO AT => AT+CIPSEND=0,53 [OK]
6. COMANDO AT => AT+CIPCLOSE=0 [OK]

===== Umidade: 64.00% =====

7. COMANDO AT => AT+CIPMUX=1 [OK]
8. COMANDO AT => AT+CIPSTART=0,"TCP","api.thingspeak.com",80 [OK]
9. COMANDO AT => AT+CIPSEND=0,53 [OK]
10. COMANDO AT => AT+CIPCLOSE=0 [OK]
```

Figura 23: Monitor serial do Arduino executando o programa

Fonte: Elaboração própria

3.5 Experimentos

3.5.1 Preparação

Devido à sua natureza complexa e heterogénea, uma rede IoT pode ser vulnerável a um grande número de diferentes ataques, é importante definir o que os definem e em que consistem (ROSA, 2021).

De acordo com Rosa (2021, p.13, apud ZHANG, 2017), “um ataque que incide numa rede IoT pode ser classificado através de 6 atributos distintos: Camada alvo, dispositivo alvo, canal de transmissão, consequências, tamanho e furtividade.”

1. Camada Alvo: o primeiro atributo de um ataque que pode ser identificado é a camada alvo. Como visto anteriormente, a rede IoT consiste em três camadas distintas. No nosso exemplo, o alvo é a Camada de Rede.

2. Dispositivo Alvo: Um ataque pode ter como alvo um único sensor ou um grupo inteiro de dispositivos na mesma rede, por isso além da camada a que pertence, é importante definir qual dispositivo é vulnerável. Em nossos experimentos o alvo sempre será o protótipo, mais especificamente o módulo ESP.

3. Canal de Transmissão: O meio de comunicação da intrusão é um atributo

importante no ataque. Esse canal pode assumir várias formas, seja um canal remoto, como WiFi ou Bluetooth, ou um canal mais próximo do dispositivo, como um toque físico, gesto ou sinal sonoro. Em nosso exemplo nos comunicaremos por WiFi.

4. Consequências: Uma invasão de um sistema pode provocar vários resultados. Devemos analisar qual será a consequência direta ao ataque. Em nossos experimentos optamos por consequência diferentes. No primeiro, retirar o módulo ESP da rede. E no segundo, retardar o funcionamento do mesmo, provocando falhas de envio.

5. Tamanho: Outra propriedade possível de identificar é o número de dispositivos ameaçados. Podendo limitar o ataque a um único dispositivo ou utilizar esse dispositivo para influenciar outros na rede a que estes estão ligados. Ambos os ataques prejudicam a rede, porém o segundo tem uma escala maior de invasão. Em nossos experimentos o alvo será apenas o protótipo.

6. Furtividade: A última característica de uma invasão é a sua capacidade de permanecer em segredo antes, durante e depois da intervenção na rede. O usuário da rede pode ficar totalmente alheio ao ataque, ter conhecimento parcial ou até mesmo total sobre a intrusão e as possíveis consequências, geralmente devido às mudanças, ou falta destas, no ambiente regular da rede causadas pela intrusão (ROSA, 2021). Discutiremos sobre esse tópico junto ao término de cada experimento.

3.5.2 Experimento 1: Ataque DoS de desautenticação

Um ataque de desautenticação de rede sem fio consiste em um ataque que nega serviços que visam as comunicações de um ponto de acesso sem fio Wi-Fi e um usuário.

A desautenticação funciona de uma maneira única comparado com a maioria dos bloqueadores de rádio. O protocolo IEEE 802.11 (Wi-Fi) contém provisões para quadros de desautenticação. O envio de quadros de um ponto de acesso para uma estação é conhecido como uma "técnica de autorização para notificar estações não autorizadas de que foi desconectada da rede".

Um invasor pode enviar um quadro de desautenticação para o AP a qualquer momento, contendo o endereço falsificado da vítima. O protocolo não requer nenhuma criptografia do quadro, embora a sessão tenha estabelecido privacidade de dados usando Wired Equivalent Privacy (WEP), e o invasor só precisa saber o endereço MAC da vítima, que está disponível em texto não criptografado via sniffing sem fio na rede (SILVA, 2005).

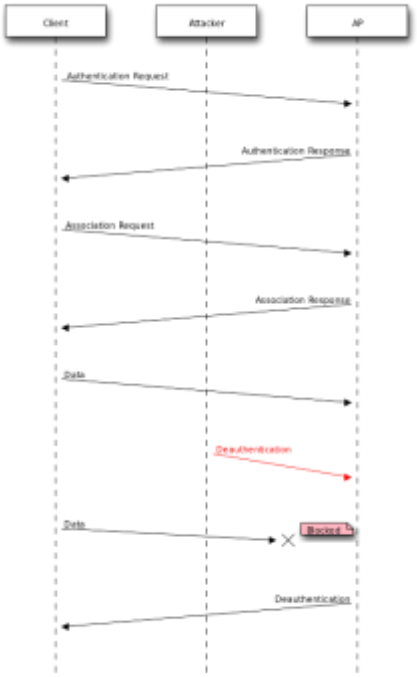

Figura 24: Ataque de Desautenticação

Extraído de: https://en.wikipedia.org/wiki/Wi-Fi_deauthentication_attack

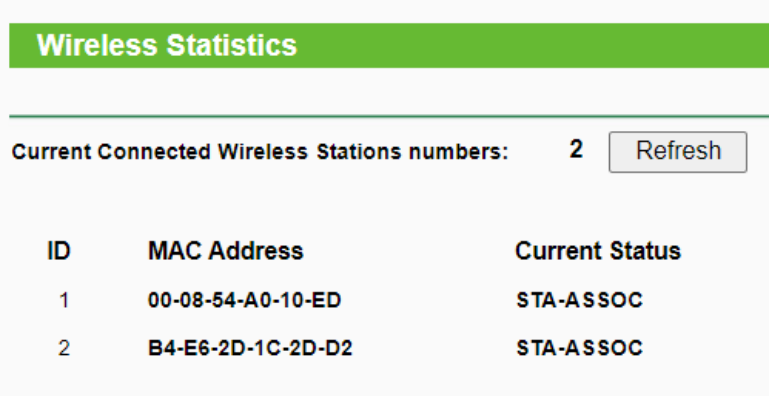

Figura 25: Endereços MAC conectados ao roteador: Máquina virtual e Protótipo.

Fonte: Roteador TP-LINK.

No Kali Linux, acessamos o terminal de comandos e utilizamos o comando NMAP como é mostrado na figura a seguir.

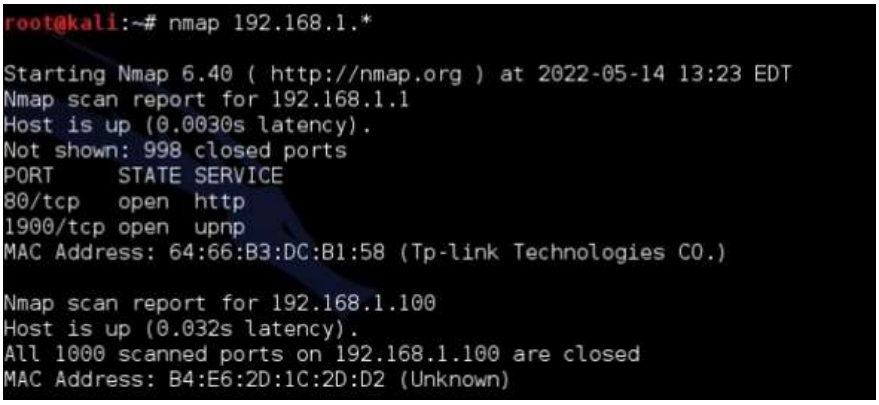

Figura 26: Identificando endereços MAC com NMAP.

Fonte: Elaboração própria no Kali Linux

Como visto na figura 26, podemos identificar os endereços MAC, IP e algumas outras informações sobre cada IP iniciado com “192.168.1.”. Utilizaremos os endereços MAC do roteador, 64:66:B3:DC:B1:58, e do protótipo IoT, B4:E6:2D:1C:2D:D2.

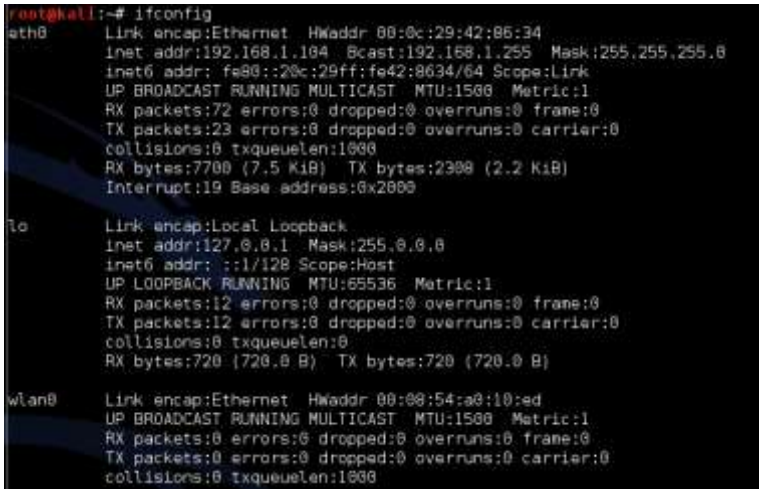

Figura 27: Identificando interface a ser monitorada

Fonte: Elaboração própria no Kali Linux

Utilizamos o comando ifconfig para verificar o nome dado para a interface de dispositivos wireless. Neste experimento monitoramos a interface wlan0, pois precisamos identificar o que é enviado e recebido via protocolo 802.11, como podemos ver na figura a seguir.

Figura 28: Iniciando modo de monitoramento mon0.

Fonte: Elaboração própria no Kali Linux

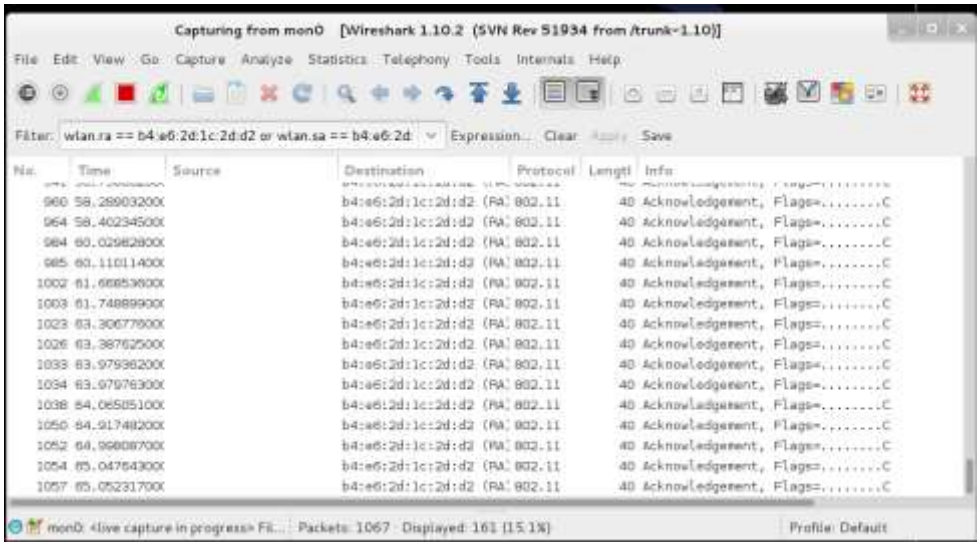

Figura 29: Wireshark capturando pacotes enviados e recebidos pelo protótipo em mon0.

Fonte: Elaboração própria no Wireshark

No Wireshark, com o filtro “wlan.ra == b4:e6:2d:1c:2d:d2 or wlan.sa == b4:e6:2d:1c:2d:d2” podemos verificar os pacotes enviados e recebidos pelo protótipo.

```
root@kali:~# aireplay-ng --deauth 0 -c B4:E6:2D:1C:2D:D2 -a 64:66:B3:DC:B1:58 mo
n1 --ignore-negative-one
13:30:56  Waiting for beacon frame (BSSID: 64:66:B3:DC:B1:58) on channel -1
13:30:56  Sending 64 directed DeAuth. STMAC: [B4:E6:2D:1C:2D:D2] [121|128 ACKs]
13:30:57  Sending 64 directed DeAuth. STMAC: [B4:E6:2D:1C:2D:D2] [128|128 ACKs]
13:30:58  Sending 64 directed DeAuth. STMAC: [B4:E6:2D:1C:2D:D2] [136|138 ACKs]
13:30:58  Sending 64 directed DeAuth. STMAC: [B4:E6:2D:1C:2D:D2] [129|128 ACKs]
13:30:59  Sending 64 directed DeAuth. STMAC: [B4:E6:2D:1C:2D:D2] [136|137 ACKs]
13:30:59  Sending 64 directed DeAuth. STMAC: [B4:E6:2D:1C:2D:D2] [128|128 ACKs]
```

Figura 30: Ataque de desautenticação

Fonte: Elaboração própria no Kali Linux

Na Figura 30, --deauth é utilizado para ataques de desautenticação, o 0 representa um número infinito de ataques, -c é o cliente a ser atacado, seguido pelo endereço MAC do dispositivo, -a é o roteador em que a vítima está conectada e mon0 é o nome da interface em modo monitor. O comando --ignore-negative-one é utilizado justamente por conta de estarmos em modo monitor.

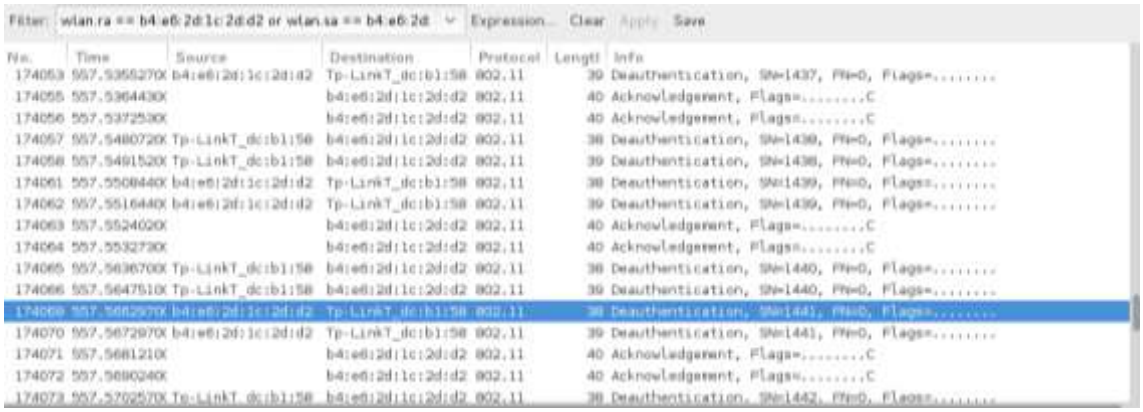

Figura 31: Pacotes sendo recebidos pelo Wireshark.

Fonte: Elaboração própria no Wireshark

Como podemos conferir na Figura 31, há diversos pedidos de desautenticação sendo enviados pelo protótipo.

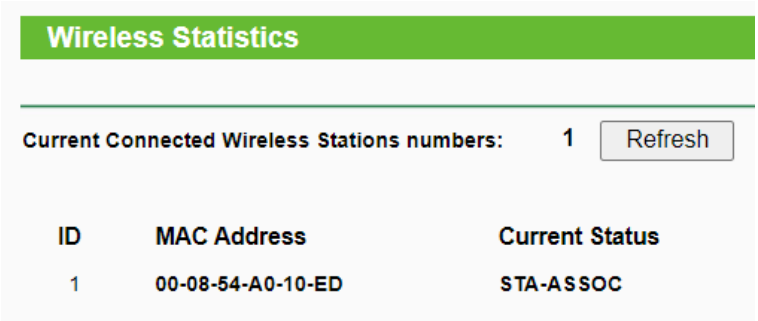

Figura 32: Protótipo encontra-se fora da rede.

Fonte: Roteador TP-LINK.

Como demonstrado na Figura 32, o protótipo está fora da rede, porém, assim que encerramos o processo, o protótipo conecta novamente.

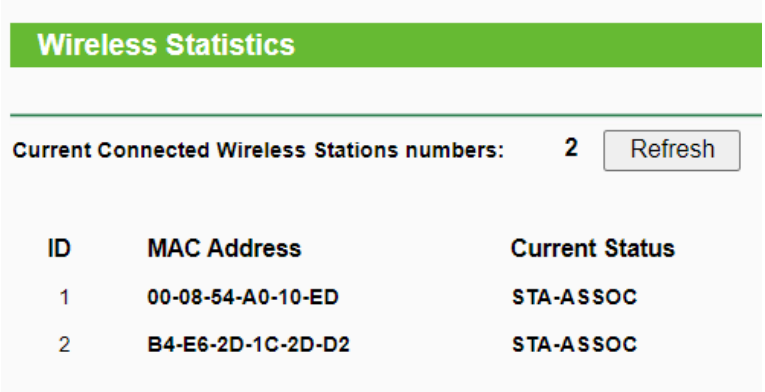

Figura 33: Protótipo voltando à rede.

Fonte: Roteador TP-LINK.

Isto ocorre por conta dos comandos AT e o loop utilizados no código em execução no protótipo. Para cada valor de temperatura e umidade enviado pelo sensor há um pedido de conexão e um de desconexão, dessa forma o programa continua rodando no protótipo, mesmo com os erros seguidos de tentativa de envio devido a desautenticação, e assim que pode autenticar na rede voltará a enviar os dados registrados pelos sensores.

Figura 34: Mensagens 1 e 3 do 4-way handshake.

Fonte: Elaboração própria no Wireshark

Após o ataque é possível notar parte do processo do 4-way handshake. Que ocorre com a tentativa de reconexão (Figura 35).

O ataque de desautenticação força a vítima a reautenticar. O atacante pode farejar o 4-way handshake e executar um ataque WPA de força bruta para descobrir a senha de rede.

Outro ataque que depende muito deste ataque de desautenticação é o que força o usuário a se conectar a um ponto de acesso falso. Clonando o roteador do usuário e então desautenticando o mesmo do roteador original. É necessário garantir

que seu roteador tenha um sinal mais alto que o roteador original. O dispositivo do usuário se conectará automaticamente ao seu roteador, pois está “mais próximo”.

Uma vez que o dispositivo do usuário esteja conectado ao seu ponto de acesso falso, você pode facilmente farejar todas as suas conexões de saída e entrada.

3.5.3 Experimento 2: Ataque SYN Flooding

Um ataque SYN (chamado também de SYN flooding) explora uma vulnerabilidade que implementa a fase de handshake de três vias na maioria dos hosts. Quando o host B recebe uma solicitação SYN do host A, essa conexão deve se manter parcialmente aberta na fila de escuta por pelo menos 75 segundos. Sendo essencial para permitir que conexões sejam abertas em redes de alta latência, como satélite. Essa técnica possui um problema que a maioria das implementações só pode lidar com um número muito limitado de conexões (a maioria, por padrão, lida apenas com 5 conexões, embora outras implementações possam lidar com até 1024). Um host A de ação maliciosa pode aproveitar esse tamanho reduzido de "fila de escuta" e enviar várias solicitações SYN para o host B sem responder ao SYN&ACK enviado pelo host B. O host de destino se enche rapidamente, impedindo-o de aceitar novas conexões até que a fila processe conexões parcialmente atendidas ou elimine os tempos limite. Essa capacidade de remover um host da rede por pelo menos 75 segundos é característica de ataques DoS (TAROUCO, 1999).

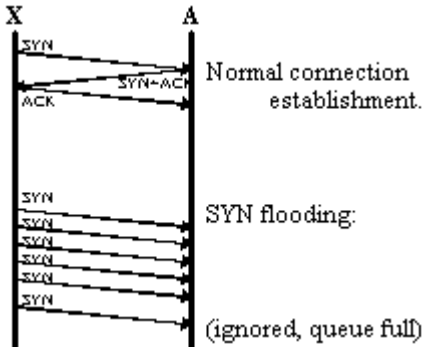

Figura 35: Ataque SYN FLOOD (TAROUCO, 1999)

Começaremos como no experimento anterior, identificando os endereços com NMAP, porém, desta vez utilizaremos os endereços IP.

Figura 36: Identificando IPs com NMAP.

Fonte: Elaboração própria no Kali Linux

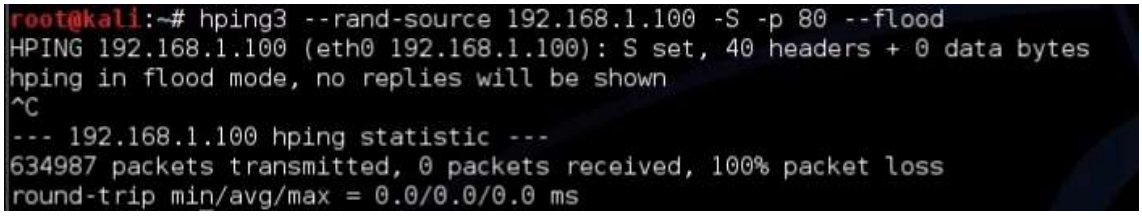

Figura 37: Utilizando HPING3.

Fonte: Elaboração própria no Kali Linux

Na Figura 37, --rand-source é utilizado para modo aleatório de IPs de envio, seguido do IP de destino (vítima do ataque), -S significa que serão enviadas SYN flags, -p 80 representa a porta 80 do dispositivo e --flood é utilizado para que o processo continue ocorrendo até que seja dado um comando de parada.

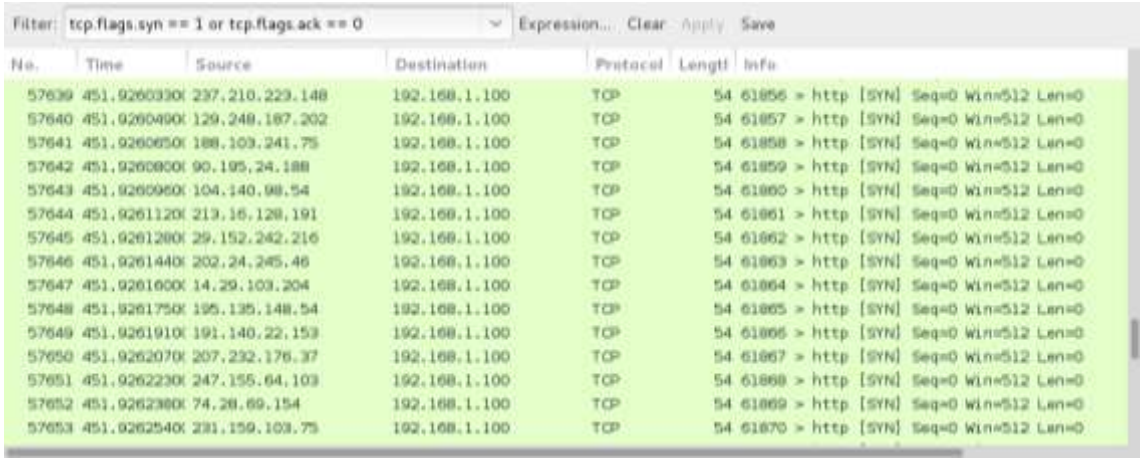

Figura 38: IP recebendo diversos pacotes SYN durante o ataque

Fonte: Elaboração própria no Wireshark

No Wireshark, com o filtro “tcp.flags.syn == 1 or tcp.flags.ack == 0” podemos visualizar diversos pacotes SYN sendo enviados sem resposta.

Durante o ataque o protótipo recebeu diversos pacotes e não conseguiu lidar com o fluxo, tentando responder às diversas solicitações enviadas sem sucesso. Essa forma de ataque é simples, porém não furtiva. Os Firewalls identificariam os IP gerados pela máquina atacante e os bloqueariam com facilidade. Realizaremos a seguir o mesmo experimento, porém com furtividade.

```
root@kali:~# hping3 -a 192.168.1.1 192.168.1.100 -S -q -p 80 --flood
HPING 192.168.1.100 (eth0 192.168.1.100): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.100 hping statistic ---
1313562 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Figura 39: SYN Flooding com clonagem de IP

Fonte: Elaboração própria no Kali Linux

Na Figura 39, utilizamos -a seguido do ip do roteador para clonarmos o mesmo, este será nosso endereço de envio. Seguido do IP de destino (vítima do ataque), -S significa que serão enviadas SYN flags, -q indica modo furtivo, -p 80 representa a porta 80 do dispositivo e --flood é utilizado para que o processo continue ocorrendo até que seja dado um comando de parada.

Figura 40: IP recebendo diversos pacotes de reconhecimento SYN durante o ataque

Fonte: Elaboração própria no Wireshark

3.5.4 Contramedidas

No Linux, um ataque SYN Flood pode ser evitado habilitando os SYN Cookies, um recurso fornecido diretamente pelo kernel, que pode ser feito com o seguinte comando, que pode ser incluído no seu script de firewall:

echo 1 > /proc/sys/net/ipv4/tcp_syncookies

Quando o recurso é ativado, o sistema começa a responder ao pacote SYN inicial com um cookie que identifica o cliente. Dessa forma, o sistema aloca espaço para a conexão somente após receber o pacote de resposta ACK, tornando o ataque ineficaz. O invasor ainda pode consumir alguma largura de banda, forçando o servidor a enviar um grande número de cookies SYN de resposta, mas o impacto no servidor será mínimo (MONQUEIRO, 2010).

4 Conclusão

Neste TCC foram propostas práticas que englobam a multidisciplinaridade por trás do conceito de Internet das Coisas. Com seu detalhamento, é possível, posteriormente, criar um material de ensino contendo conceitos de segurança da informação, redes de computadores, programação, Linux e eletrônica.

Abordamos dois tipos de ataques de negação de serviço explicando suas propriedades, como exploram falhas no processo de handshake e demonstrando a importância de ser criado um perímetro com Firewall e/ou VPN para proteger a rede

WiFi de dispositivos IoT. E também, no caso de ataque SYN Flooding, a importância do uso de SYN Cookies (no Linux) para a proteção da rede deste tipo de ataque. Demonstrou-se também a vulnerabilidade de credenciais de login e senha presentes em texto claro no código de um dispositivo.

Dessa forma, ao longo do trabalho verificamos que existem um conjunto de problemas de segurança que são pouco ou simplesmente não considerados ao se implementar uma rede IoT. Sabendo da importância e evolução tecnológica inerente a esses dispositivos, se faz necessário um material voltado à segurança em IoT para estudantes na graduação.

Esperamos que os resultados obtidos neste trabalho possam ser postos em prática nas instituições de ensino, agregar conhecimento aos estudantes e incentivá-los a explorar as áreas de segurança da informação, redes e programação voltadas à Internet das Coisas.

5 Trabalhos Futuros

O foco principal deste trabalho é incentivar o estudo de segurança da informação em IoT. Para que o mesmo seja totalmente alcançado é necessária a sua implementação. Porém como complemento do mesmo, é possível a criação de um material de aula com mais tópicos pertinentes à Segurança da Informação e outras disciplinas. Como, por exemplo:

- A continuação do ataque de desautenticação explorando as mensagens interceptadas do Four-Way Handshake e descriptografando a senha contida nelas;
- Tratar sobre outras formas de transmissão de dados;
- Ataques com outras consequências, por exemplo, um que altere os dados enviados pelo protótipo, atacando assim sua integridade;
- Alterar o funcionamento do protótipo, para que o mesmo envie dados a uma página com banco de dados criada pelo aluno.

Link do Projeto no GitHub

https://github.com/betavasconcellos/TCC

Referências

AIRCRACK-NG. **Aircrack-ng.** Aircrack-ng, 2022. Disponível em: http://www.aircrack-ng.org/doku.php?id=aircrack-ng. Acesso: 15 mai. 2022.

AIRCRACK-NG. **Aireplay-ng.** Aircrack-ng, 2022. Disponível em: http://www.aircrack-ng.org/doku.php?id=aireplay-ng. Acesso: 15 mai. 2022.

AIRCRACK-NG. **Airmon-ng.** Aircrack-ng, 2022. Disponível em: http://www.aircrack-ng.org/doku.php?id=airmon-ng. Acesso: 15 mai. 2022.

AI-THINKER. **ESP-01 WiFi Module**. AI-Thinker, 2015. Disponível em: https://www.microchip.ua/wireless/esp01.pdf. Acesso: 08 mai 2022.

ALMEIDA, Paulo Samuel. **Indústria 4.0: Princípios básicos, aplicabilidade e implantação na área Industrial**. São Paulo: Érica, 2019.

ALVES, David; PEIXOTO, Mário; ROSA, Thiago. **Internet das coisas (IoT): Segurança e privacidade dos dados pessoais**. Rio de Janeiro: Alta Books, 2021.

ARDUINO. **Arduino Playground.** Arduino, 2022. Disponível em: https://playground.arduino.cc/Portugues/HomePage/. Acesso: 15 mai. 2022.

BR-ARDUINO. **ESP8266: Comandos AT**. BR-Arduino, 2015. Disponível em: https://br-arduino.org/2015/11/esp8266-comandos-at.html. Acesso: 15 mai. 2022.

CARDOSO, Érico Edú Corrêa Cardoso; DAVID, Tobias. **A falta de profissionais de tecnologia de informação no mercado de trabalho**. In: CONGRESSO INTERNACIONAL UMA NOVA PEDAGOGIA PARA A SOCIEDADE FUTURA: PROTAGONISMO RESPONSÁVEL ,2.,2016, Recanto Maestro. Anais... Restinga Sêca: Fundação Antonio Meneghetti, 2016. p. 697-700. Disponível em: https://reciprocidade.emnuvens.com.br/novapedagogia/article/view/216. Acesso: 15 abr. 2022.

CARRION, Patrícia; QUARESMA, Manuela. **Internet das coisas (IoT): definições e aplicabilidade aos usuários finais**. Human Factors in Design, v. 8, no. 15, p. 49–66, 2019. Disponível em: https://www.revistas.udesc.br/index.php/hfd/article/view/231679630815201904. Acesso: 15 abr. 2022.

CODD, Terrance. **Wireless Temperature and Humidity Monitor With ESP8266**. INSTRUCTABLES, 2015. Disponível em: https://www.instructables.com/Wireless-Temperature-and-Humidity-Monitor-With-ESP/. Acesso: 15 mai. 2022.

COUTINHO, Rômulo; VASQUEZ, Yuri; MACHADO, Leonardo. **Sistemas de Detecção de Intrusão**. UFRJ, 2006. Disponível em: https://www.gta.ufrj.br/grad/10_1/sdi/ataques.html . Acesso: 19 mai. 2022.

EVANS, Dave. **A Internet das Coisas - Como a próxima evolução da Internet está mudando tudo**. [s.l.]: CISCO, 2011. 13 p. Disponível em: https://www.cisco.com/c/dam/global/pt_br/assets/executives/pdf/internet_of_things_iot_ibsg_0411final.pdf . Acesso em: 15 abr. 2022.

FIRMINO, Macêdo. **Simulando ataques DoS com Hping**. IFRN – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, 2019. Disponível em: http://docente.ifrn.edu.br/josemacedo/disciplinas/2019/2019.1/praticas-de-laboratorio/03-simulando-ataques-dos-com-hping. Acesso: 08 mai. 2022.

GIL, Antonio Carlos. **Metodologia do Ensino Superior**. 5. ed. São Paulo: ATLAS, 2020. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788597023954/. Acesso em: 22 mar. 2022.

HAIDER, Zeeshan. **4-Way Handshake**. WiFi Professionals, 2019. Disponível em: https://www.wifi-professionals.com/2019/01/4-way-handshake. Acesso em: 08 mai. 2022.

KALI LINUX. **What is Kali Linux?.** Kali Linux, 2022. Disponível em: https://www.kali.org/docs/introduction/what-is-kali-linux/. Acesso: 15 mai. 2022.

LEITE, Leandro Rogério. **Internet das Coisas (IoT): Vulnerabilidades de Segurança e Desafios**. Trabalho de Conclusão de Curso (Bacharelado em

Tecnologia em Segurança da Informação) – FATEC, Americana, 2019. Disponível em: http://ric.cps.sp.gov.br/handle/123456789/3978. Acesso: 15 mai. 2022.

LÜDTKE, Rudolfo Kunde. **Teste de Invasão em Redes Sem Fio 802.11**. Trabalho de Conclusão de Curso (Tecnólogo em Redes de Computadores) – Universidade Federal de Santa Maria, Santa Maria, 2015. Disponível em: http://www.redes.ufsm.br/docs/tccs/Rudolfo-Kunde.pdf. Acesso em: 15 mai. 2022.

MAGRANI, Eduardo. **A internet das coisas**. Rio de Janeiro: FGV Editora, 2018. Disponível em: https://bibliotecadigital.fgv.br/dspace/bitstream/handle/10438/23898/A%20internet%20das%20coisas.pdf. Acesso em: 15 abr. 2022.

MASCHIETTO, Luís G.; VIEIRA, Anderson Luiz N.; TORRES, Fernando E.; et al. **Arquitetura e Infraestrutura de IoT**. Porto Alegre : SAGAH, 2021. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9786556901947/. Acesso em: 22 mar. 2022.

MONQUEIRO, Julio. **Bloqueando ataques de SYN Flood**. HARDWARE, 2010. Disponível em: https://www.hardware.com.br/dicas/syncookies.html. Acesso: 15 mai. 2022.

MORAES, Alexandre D.; HAYASHI, Victor T. **Segurança em IoT**. Rio de Janeiro: Editora Alta Books, 2021. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788550816548/. Acesso em: 22 mar. 2022.

MORAIS, Izabelly; et al. **Introdução a Big Data e Internet das Coisas (IoT)**. Porto Alegre: SAGAH, 2018. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788595027640/. Acesso em: 18 mai. 2022.

MOUSER ELECTRONICS. **DHT11 Humidity & Temperature Sensor**. Mouser Electronics, 2022. Disponível em: https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf . Acesso: 08 mai. 2022

NETO, Pedro; ARAÚJO, Wagner. **Segurança da Informação: Uma visão sistêmica para implantação em organizações**. João Pessoa: Editora UFPB, 2019. Disponível em: http://www.editora.ufpb.br/sistema/press5/index.php/UFPB/catalog/download/209/75/905-1?inline=1. Acesso em: 15 mai. 2022.

Observatório SOFTEX – SOFTWARE e Serviços de TI: A indústria brasileira em perspectiva. Campinas: [s.n.], 2012. Disponível em: https://geein.fclar.unesp.br/admin/dbo/core/classes/download.php?name=Software%20e%20Servi%C3%A7os%20de%20TI:%20A%20Ind%C3%BAstria%20Brasileira%20em%20Perspectiva.pdf&file=1459358127-0964.pdf. Acesso: 15 abr. 2022.

PERES, André; LOUREIRO, César Augusto; SCHMITT, Marcelo Augusto. **Redes de Computadores II**. Porto Alegre: Bookman, 2014. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582601488/. Acesso em: 18 mai. 2022.

FEDELI, Ricardo Daniel; POLLONI, Enrico Giulio; PERES, Fernando Eduardo. **Introdução à Ciência da Computação - 2ª edição atualizada**. São Paulo: Cengage Learning Brasil, 2010. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788522110001/. Acesso em: 08 mai. 2022.

ROSA, João Pedro G.. **Mecanismos de Segurança IoT.** Dissertação (Mestrado em Engenharia Eletrotécnica e de Computadores) – Universidade Nova de Lisboa, Lisboa, 2021. Disponível em: https://run.unl.pt/handle/10362/120558. Acesso em 15 mai. 2022.

SACOMANO, José et al. **Indústria 4.0: conceitos e fundamentos**. São Paulo: Editora Blucher, 2018. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788521213710/. Acesso em: 15 mai. 2022.

SANTOS, Bruno et al. **Internet das Coisas: da Teoria à Prática.** Universidade Federal de Minas Gerais (UFMG), 2017. Disponível em:

https://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf. Acesso em 18 mai. 2022.

SCHMITT, Marcelo Augusto R.; PERES, André; LOUREIRO, César Augusto H. **Redes de Computadores: nível de aplicação e instalação de serviços**. Porto Alegre: Bookman, 2013. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582600948/. Acesso em: 08 mai. 2022.

SILVA, Gilson Marques. **Segurança em Redes Locais sem Fio**. Dissertação (Pós-Graduação em Ciência da Computação) – Universidade Federal de Uberlândia, Uberlândia, 2005. Disponível em: https://repositorio.ufu.br/bitstream/123456789/12491/1/MarquesDISSPRT.pdf. Acesso em 18 mai. 2022.

TAROUCO, Liane. **Vulnerabilidades do TCP.** UFRGS, 1999. Disponível em: http://penta2.ufrgs.br/gere97/upload/files/gere97/redes9/aula9.htm. Acesso: 15 mai. 2022.

TAURION, Cezar. **Big Data**. Rio de Janeiro: Brasport, 2013.

THINGSPEAK. **IoT Analytics - ThingSpeak Internet of Things**. The MathWorks, 2022. Disponivel em: https://thingspeak.com/. Acesso em: 08 mai. 2022.

VIEIRA, Larissa Benevides; GRADVOH, André Leon. **Atividades dinâmicas para o ensino de segurança da informação em dispositivos para Internet das Coisas.** In: CONGRESSO DE INICIAÇÃO CIENTÍFICA DA UNICAMP, 28., 2020, Limeira. Anais eletrônicos … Limeira: [s.n.], 2020. Disponível em: https://www.prp.unicamp.br/inscricao-congresso/resumos/2020P16383A34268O479.pdf. Acesso em: 15 abr. 2022.

WIRESHARK. **About Wireshark: Introduction.** Wireshark, 2022. Disponível em: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html. Acesso: 08 mai. 2022.