

FACULDADE DE TECNOLOGIA DE SÃO JOSÉ DOS CAMPOS
FATEC PROFESSOR JESSEN VIDAL

Marcos Hideki Inoue Júnior

**Análise de segurança em redes de IoTs e
investigação de suas principais vulnerabilidades**

São José dos Campos

2017

Marcos Hideki Inoue Júnior

Análise de segurança em redes de IoTs e investigação de suas principais vulnerabilidades

Trabalho de Graduação apresentado à Faculdade de Tecnologia São José dos Campos, como parte dos requisitos necessários para a obtenção do título de Tecnólogo em Banco de Dados.

FACULDADE DE TECNOLOGIA DE SÃO JOSÉ DOS CAMPOS

FATEC PROFESSOR JESSEN VIDAL

Orientador: MSc. Eduardo Sakaue

Coorientador: MSc. Diogo Branquinho Ramos

São José dos Campos

2017

Dados Internacionais de Catalogação-na-Publicação (CIP)
Divisão de Informação e Documentação

INOUE, Marcos Hideki
Análise de segurança em redes de IoTs e investigação de suas principais vulnerabilidades
São José dos Campos, 2017
[44f.](#)

Trabalho de Graduação – Curso de Tecnologia em Banco de Dados
FATEC de São José dos Campos: Professor Jessen Vidal, 2017
Orientador: MSc. Eduardo Sakaue
Coorientador: MSc. Diogo Branquinho Ramos

Áreas de Conhecimento. I. Faculdade de Tecnologia. FATEC de São José dos Campos: Professor Jessen Vidal. Divisão de Informação e Documentação. II. Análise de segurança em redes de IoTs e investigação de suas principais vulnerabilidades

REFERÊNCIA BIBLIOGRÁFICA —

INOUE, Marcos Hideki. Análise de segurança em redes de IoTs e investigação de suas principais vulnerabilidades 2017. [44f.](#) Trabalho de Graduação – FATEC de São José dos Campos: Professor Jessen Vidal.

CESSÃO DE DIREITOS —

NOME DO AUTOR: Marcos Hideki Inoue Júnior
TÍTULO DO TRABALHO: Análise de segurança em redes de IoTs e investigação de suas principais vulnerabilidades
TIPO DO TRABALHO/ANO: Trabalho de Graduação/2017

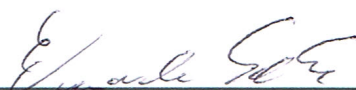
É concedida à FATEC de São José dos Campos: Professor Jessen Vidal permissão para reproduzir cópias deste Trabalho e para emprestar ou vender cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte deste Trabalho pode ser reproduzida sem a autorização do autor.

Marcos Hideki Inoue Júnior

Análise de segurança em redes IoTs e investigação de suas principais vulnerabilidades

Trabalho de Graduação apresentado à Faculdade de Tecnologia São José dos Campos, como parte dos requisitos necessários para a obtenção do título de Tecnólogo em Banco de Dados.

Composição da Banca



MSc. Eduardo Sakaue
Orientador

MSc. Diogo Branquinho Ramos
Coorientador

Fabiano Sabha Walczak
Professor Convidado

Francisco Marcelo Corrêa da Silva
Professor Convidado

São José dos Campos
2017

*A todos aqueles que de alguma forma estiveram e estão próximos de mim,
fazendo com que eu tenha DETERMINAÇÃO.*

Agradecimentos

Sou grato aos meus pais por tudo que fizeram e fazem por mim.

Agradeço a Faculdade de Tecnologia de São José dos Campos (FATEC-SJC) pela oportunidade de estudar nesta instituição de ensino maravilhosa que me proporcionou uma quantidade imensurável de conhecimento.

Agradeço também o Instituto Nacional de Pesquisas Espaciais (INPE) pela oportunidade de estágio que me proporcionou muitas oportunidades para aumentar meu conhecimento e aprender sobre assuntos variados e a utilização de linguagens diferentes.

Agradeço a microempresa Tecnologias para a Sustentabilidade (TecSUS) pela oportunidade de estágio e em conjunto com a mesma, sou grato à Microchip Technology e ao Ricardo Seiti da utilização de seus equipamentos e microcontroladores para a realização deste trabalho.

Sou extremamente grato também ao meu orientador Eduardo Sakaue por sua orientação, atenção e ajuda como orientador e como amigo. Também ao meu coorientador Diogo Branquinho Ramos por sua coorientação e ajuda seja como coorientador ou amigo.

Sou muito grato pelos meus professores que tive, em especial o professor Giuliano Bertoti que me motivou a realizar o modelo de \LaTeX , ao professor Emanuel Mineda por sua ajuda e amizade, ao professor Fabiano Sabha pela sua companhia e amizade e Jean Carlos pela sua amizade e companheirismo.

Sou muito agradecido pela companhia e amizade de minha melhor amiga Deborah Susan e de meus amigos e amigas: Safire Lauene, Samantha Marques, Camilo Damaso, William Siqueira, Erivan Lima, Luana Câmara, Rafael Viana, Leonardo Neves, Pedro Valentim, Matheus Monteiro, Vanilson Leite, Reginaldo Moreira, Antônio Siqueira e Clélio Henrique.

Sou grato aos meus colegas de trabalho da TecSUS: Thiago Gomes, Wagner Fukuoka e Ariadne Mioni que me oferecem um ótimo ambiente de trabalho. Também agradeço meus colegas de trabalho do INPE: Verônica Maria, Rodrigo Takeshi, Anna Karina, Müller Lopes e José Marchezi que me proporcionaram um ambiente de trabalho descontraído e divertido, também me incentivaram e me ajudaram com o uso do \LaTeX para a realização deste trabalho e a criação do modelo de Trabalho de Graduação para a FATEC-SJC utilizando \LaTeX .

*“Companies spend millions of dollars on firewalls,
encryption and secure access devices,
and it’s money wasted,
because none of these measures address the weakest link in the security chain.”
(Kevin Mitnik)*

Resumo

A *Internet of Things* (IoT) está diretamente conectada às inovações tecnológicas em diversas áreas, logo, a área de *IoT* irá crescer significativamente e isso gera a necessidade de uma atenção especial com a segurança nessas redes. As redes *IoTs* são vulneráveis e suscetíveis a ataques, logo se não forem tomadas medidas a fim de garantir a segurança destas redes, as mesmas poderão ficar indisponíveis e conseqüentemente deixar prejuízos para empresas que utilizam sistemas IoT. As vulnerabilidades de maior impacto encontradas foram a escuta passiva, *man in the middle* e o *denial of service* como o *flooding* e o *jamming*. Este trabalho apresenta um estudo da segurança em redes IoT utilizando mecanismos confiáveis para garantir a segurança nestas redes e, com o objetivo de realizar isso, mecanismos como um sistema para entrega de endereços, padrão de *whitelist* e criptografia são implementados. Os resultados revelam que os mecanismos aplicados são efetivamente funcionais e garantem a segurança básica da rede, deste modo podem ser utilizados para assegurar redes IoTs.

Palavras-chave: IoT, Internet of Things, Security, DoS, Flooding, Jamming, Sniffing, Man in the Middle, Whitelist, Wireless Mesh Networks.

Abstract

Internet of Things (IoT) is directly connected to technological innovations in several areas, thus this area will grow significantly and this generates the need for special attention on security. IoT networks are vulnerable and is an easy target for attacks, so if no action is taken in order to avoid that, these networks can become unavailable and consequently leaving a disadvantage to companies who uses IoT systems. The vulnerabilities of greatest impact found were eavesdropping, man in the middle and denial of service like flooding and jamming. This work presents a study of security in an IoT network using trusted mechanisms to grant defense in these networks, in order to do that, mechanisms like a system to deliver addresses, whitelist pattern and criptography are implemented. Results reveal that mechanisms applyied are effectively worth and grant a basic security to the network. Thus can be used to secure IoTs networks.

Keywords: IoT, Internet of Things, Security, DoS, Flooding, Jamming, Sniffing, Man in the Middle, Whitelist, Wireless Mesh Networks.

Lista de ilustrações

Figura 1 – Cenário da <i>Internet of Things</i> com a divergência de visões	16
Figura 2 – <i>Jammer</i> Constante	21
Figura 3 – <i>Man in The Middle</i>	22
Figura 4 – <i>Wormhole Attack</i>	23
Figura 5 – <i>Sinkhole Attack</i> ; (a) nó comprometido; (b) Estação principal.	24
Figura 6 – Adaptador USB da Atmel para baixa frequência.	25
Figura 7 – <i>Pacote capturado pelo sniffer</i>	26
Figura 8 – Nó estrategicamente posicionado para realizar o ataque.	28
Figura 9 – Nó malicioso acoplado na rede.	28
Figura 10 – Nó malicioso realizando <i>flooding</i>	30
Figura 11 – Nó malicioso realizando <i>jamming</i> com sua área de abrangência.	32
Figura 12 – <i>SMART SAM R21 Xplained Pro</i>	33
Figura 13 – <i>Routers</i> simulando postes	34
Figura 14 – Pacote capturando utilizando <i>sniffer</i> na rede <i>Streetlight</i>	35
Figura 15 – Coordenador inválido enviando um pacote malicioso.	36
Figura 16 – Nó alvo recebendo o pacote e checando o endereço <i>MAC</i> de quem enviou.	37
Figura 17 – Pacotes do <i>flooding</i> capturados pelo <i>sniffer</i>	38
Figura 18 – Comportamento do coordenador no campo do <i>jamming</i>	39
Figura 19 – Comportamento do <i>router</i> fora do campo do <i>jamming</i>	39
Figura 20 – Análise da potência do ruído emitido utilizando um analisador de espectro.	40

Lista de tabelas

Tabela 1 – Tabela de Ataques	20
Tabela 2 – Exemplo da <i>whitelist</i>	29
Tabela 3 – Especificações técnicas do microcontrolador SMART SAM R21	33

Lista de abreviaturas e siglas

DDoS	<i>Distributed Denial of Service</i>
DoS	<i>Denial of Service</i>
DHCP	<i>Dynamic Host Configuration Protocol</i>
IoT	<i>Internet of Things</i>
LWMesh	<i>Lightweight Mesh</i>
MAC	<i>Media Access Control</i>
MITM	<i>Man In The Middle</i>
RFID	<i>Radio-Frequency IDentification</i>
SSID	<i>Service Set Identifier</i>
WSN	<i>Wireless Sensor Network</i>
WMN	<i>Wireless Mesh Network</i>

Lista de símbolos

m^2 Metro quadrado

GHz Giga-hertz

MHz Mega-hertz

Sumário

1	INTRODUÇÃO	15
1.1	Problema	18
1.2	Objetivo Geral	18
1.3	Objetivo Específico	18
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	Redes de IoT	19
2.1.1	Standard IEEE 802.15.4	19
2.2	Conceitos de Segurança	20
2.3	Ataque de Negação de Serviço	20
2.4	<i>Jamming</i>	20
2.5	Escuta Passiva	21
2.6	<i>Man in The Middle</i>	22
2.6.1	<i>Wormhole Attack</i>	23
2.6.2	<i>Sinkhole Attack</i>	24
3	DESENVOLVIMENTO	25
3.1	Escuta Passiva	25
3.1.1	Metodologia do ataque	25
3.1.2	Implementação da mitigação	27
3.2	<i>Man In The Middle</i>	27
3.2.1	Metodologia do ataque	27
3.2.2	Implementações da mitigação	28
3.3	<i>Flooding</i>	29
3.3.1	Metodologia do ataque	30
3.3.2	Implementações da mitigação	30
3.4	<i>Jamming</i>	31
3.4.1	Metodologia do ataque	31
3.4.2	Implementações da mitigação	31
4	CASOS DE TESTES	33
4.1	Projeto de Iluminação Pública (<i>StreetLight</i>)	34
4.1.1	Topologia da Rede	34
4.1.2	Comportamento dos componentes da Rede	35
4.1.3	Teste de <i>Sniffing</i>	35
4.1.4	Teste de <i>Man In The Middle</i>	36

4.1.5	Teste de <i>flooding</i>	37
4.1.6	Teste de <i>Jamming</i>	39
4.2	Considerações	40
5	CONCLUSÃO	41
	REFERÊNCIAS	43

1 Introdução

A maior parte da vida moderna depende dos computadores, das redes de computadores e atualmente a mais notável interação do ser humano é o deslizar do dedo em uma *smartphone*. Essas tecnologias tendem a se integrar ainda mais com a nossa realidade com o avanço tecnológico dos microcomputadores e microcontroladores, tornando assim o termo *Internet of Things (IoT)* cada vez mais conhecido pela sua peculiaridade de objetos se conectando à internet.

IoT não é uma tecnologia nova, entretanto tem ganho maior espaço com a necessidade das pessoas de receberem informações sobre diversas coisas do seu dia a dia em tempo real, logo o termo significa nada mais do que objetos que realizam ações ou geram informações conectados à internet. A principal ideia de *IoT* vem da presença de coisas ou objetos de nosso dia a dia que são capazes de interagir e cooperar entre eles a fim de alcançar um objetivo em comum. A principal força da *Internet of Things* vem da ideia na qual a mesma terá um grande impacto em diversos aspectos de nosso dia a dia e no comportamento de seus usuários, já que diversos cenários serão afetados por ela, como por exemplo o recebimento de informações em tempo real de hidrômetros, monitoramento de vazamentos e entre diversos outros, com todas essas informações disponíveis na nuvem é possível acessá-las de diversas maneiras, seja ela por um computador ou um *smartphone*. Há diversas definições parecidas de *IoT*, e atualmente muitas pessoas tem dificuldades para entender realmente o que significa seu conceito, suas ideias básicas e suas implicações sociais, econômicas e técnicas ([STRATEGY; UNIT, 2005](#)).

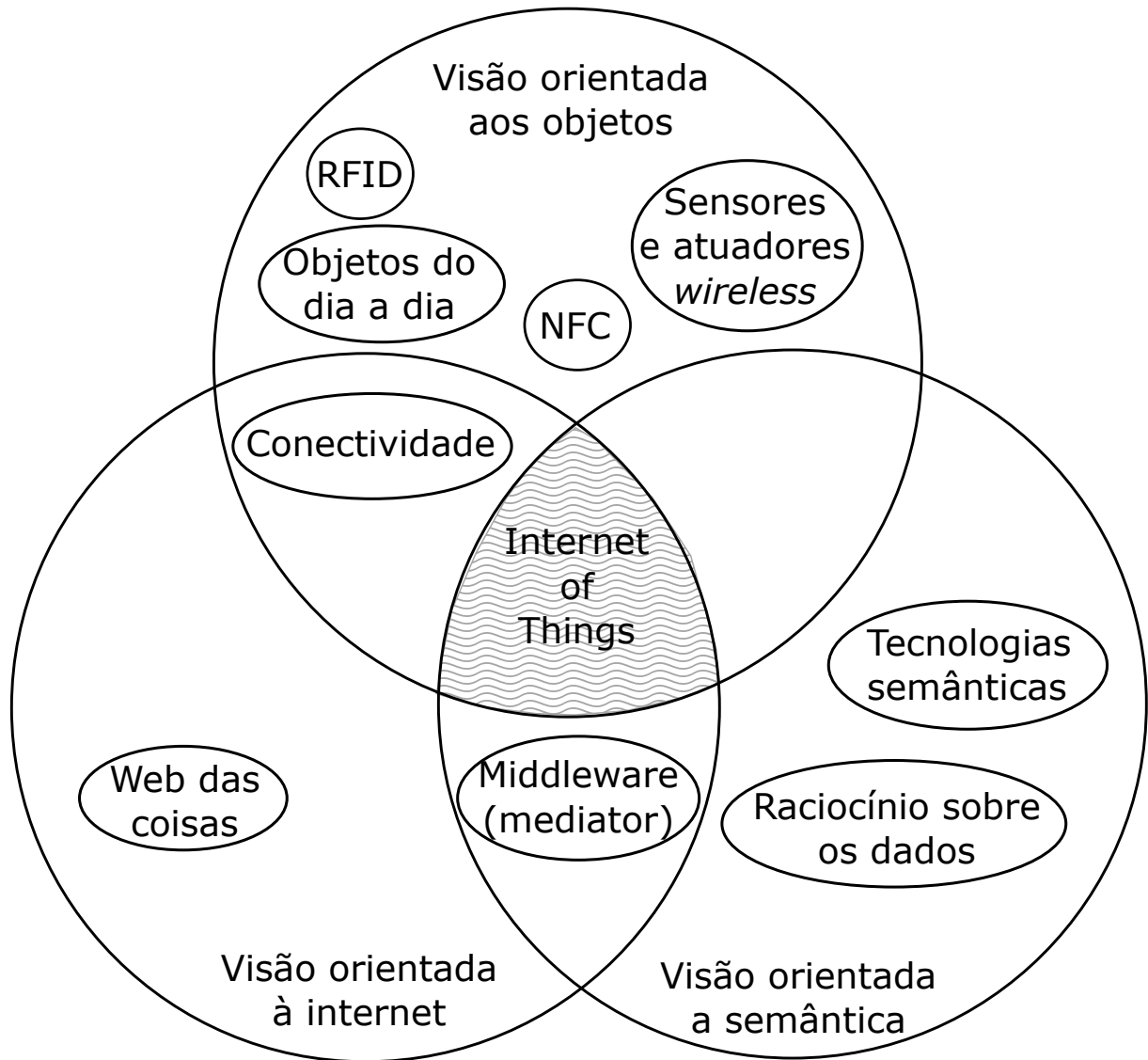
Segundo [Atzori, Iera e Morabito \(2010\)](#), o que gera confusão com o termo *IoT* é o termo em si, que une *Internet* e *Things*, dando uma ideia de conectividade que une coisas ou objetos. Expressa uma ideia genérica de objetos onde todos eles estão unidos por um meio em comum, todavia há pequenas diferenças nos pontos de vista dos componentes da *IoT*. O cenário como pode ser visto na [Figura 1](#) é dividido em 3 visões:

- Visão orientada aos objetos, que são objetos do nosso dia a dia, como sensores, cartões de *Radio-Frequency Identification (RFID)* e entre outros.
- Visão orientada a Internet, trata da visão das redes ou da internet em si no contexto de *IoT*, pode ser chamada de *web* das coisas.
- Visão orientada a Semântica, visa os conceitos tecnológicos e teóricos em si.

Todas essas visões tornam as aplicações de *IoT* muito abrangentes possibilitando uma variedade enorme de sistemas que poderão ser desenvolvidos. As principais atuações em

nossas vidas estão em casa (*Smart Home*) e em cidades (*Smart Cities*). São diversos domínios nas aplicações dessa tecnologia, e elas são agrupadas em: Transporte, Saúde, Ambientes Inteligentes (Escritórios, Casas) e Pessoal ou social.

Figura 1 – Cenário da *Internet of Things* com a divergência de visões



Fonte – Adaptada pelo autor, de [Atzori, Iera e Morabito \(2010\)](#)

A criação da *Internet of Things* de acordo com [Strategy e Unit \(2005\)](#) está diretamente relacionada as inovações tecnológicas em diversos campos. As principais tecnologias vinculadas com *IoT* são as de identificação, sensores, tecnologias inteligentes e nanotecnologia. As tecnologias de identificação vem das *tags* de *RFID*, onde é possível gerar uma identificação por radiofrequência. As tecnologias de sensores são dispositivos eletrônicos que conseguem sensorear o meio e responder à certos estímulos, seja ele mecânico, térmico, eletroestático, eletromagnético, radiação, químico, biológico e entre outros. As tecnologias

inteligentes estão muito presentes no mercado, sejam eles *smartphones*, leitores biométricos ou até mesmo carros inteligentes.

O segmento de *IoT* de acordo com [Atzori, Iera e Morabito \(2010\)](#) está em alta e muitas empresas estão competindo entre si para alcançar a liderança e crescer no mercado. Esse segmento é algo que pode ser utilizado desde equipamentos domésticos até em hospitais sendo composta de três componentes principais: as coisas, as redes de comunicação que as conectam e os sistemas de computação que usam os dados que fluem das coisas. A maior parte desses dispositivos comunicam-se através de um tipo de rede sem fio denominada *Wireless Mesh Networks* (WMNs) ou redes *mesh* sem fio.

As WMNs não dependem diretamente de um *Access Point* (AP) que envia e recebe as informações para se manter na rede similar a topologia estrela. Essas redes suportam saltos e utilizam a topologia malha. Este tipo de rede é descentralizada, relativamente barata, muito confiável e resiliente, desde que cada nó apenas transmita para outro nó de sua rede. Esses nós agem como repetidores para transmitir as informações até o seu destino, muito útil para longas distâncias ([SIDDQUI; HONG, 2007](#)).

Atualmente com a tecnologia avançada, as empresas e organizações estão mais dependentes de seus computadores e dispositivos, logo atrai a atenção de pessoas mal intencionadas. As ameaças de criminosos e terroristas para os sistemas da informação estão aumentando e empresas de diversos portes sentem a necessidade de proteger seus dados e seus sistemas. Esta necessidade é vital para o negócio, pois grande valor de uma empresa pode estar em seus dados ou toda uma operação pode depender do bom funcionamento de seus equipamentos. As ameaças cibernéticas estão sempre em constante avanço, buscando serem efetivos e cada vez mais sofisticados, em contra partida são utilizadas tecnologias atualizadas, políticas e práticas com o objetivo de evitar possíveis ameaças a integridade e disponibilidade das informações ([MALGERI, 2009](#)).

Apenas 40% das empresas que utilizam *IoT* já implementaram alguma medida de segurança, contudo a pesquisa revela que 92% dos usuários de *IoT* estão preocupados com a segurança. Isso abrirá oportunidades para abordagens de segurança mais robustas, não visando apenas empresas, mas também dispositivos voltados para o ambiente doméstico ([MATUSZAK; BELL; LE, 2015](#)).

Há casos como o da *botnet* Mirai, na qual diversos dispositivos *IoT* foram comprometidos fazendo-o com que realizassem um grande ataque *DDoS* (*Distributed Denial Of Service*) contra a empresa Akamai Technologies que trabalha com *cloud computing*. De acordo com a empresa, foi o maior *DDoS* registrado por eles, sendo ele de 667 Gigabits de tráfego por segundo ([KREBS, 2016](#)). Não foi citado muito dos dispositivos *IoT* utilizados no ataque, entretanto, é subjetivamente compreendido que os dispositivos não tinham a segurança devida implementada, ou não foi efetivamente testada, ou a segurança dos

dispositivos foram burladas.

1.1 Problema

Vulnerabilidades e falhas de segurança em redes de *IoT*.

1.2 Objetivo Geral

Este trabalho tem como objetivo analisar e levantar as possíveis ameaças e vulnerabilidades de redes de *IoT* que utilizam a topologia *mesh* a fim de buscar e tentar aplicar medidas e tecnologias para evitar possíveis falhas de segurança, de privacidade e perda de informações, para tornar o uso de redes de *IoT* mais seguros.

1.3 Objetivo Específico

Levantar possíveis ameaças e vulnerabilidades no protocolo *Lightweight Mesh* que tem como base IEEE 802.15.4, buscar medidas e tecnologias para levar possíveis maneiras de mitigação das mesmas.

2 Fundamentação Teórica

Neste capítulo serão fundamentados os conhecimentos básicos para o entendimento do trabalho e as vulnerabilidades e possíveis ataques que podem ser considerados ameaças comuns em redes de *IoT* que utilizam a topologia *mesh* com a comunicação sem fio. Para identificar as vulnerabilidades, analisaremos uma rede que tem como código fonte dos microcontroladores um exemplo de *LWMesh* e por meio de um julgamento de custo e impactos, será atribuído um nível para a vulnerabilidade. A seguir serão apresentadas vulnerabilidades e falhas que são importantes para o desenvolvimento do trabalho.

2.1 Redes de IoT

A IoT tem dois meios para transmitir as informações, o meio com fio e sem fio, portanto neste trabalho será abordado as transmissões sem fio (*wireless*).

2.1.1 *Standard* IEEE 802.15.4

Este é um *standard* criado por [Committee et al. \(2006\)](#) que define os protocolos de comunicação e a interconexão dos dispositivos via rádio. Alguns protocolos que utilizam este *standard* por exemplo são: ZigBee, MiWi e LWMesh.

Os protocolos criados em cima deste *standard* possuem as seguintes características:

- Topologias: Estrela ou malha;
- Endereçamento de 16 bits a 64 bits;
- Baixo consumo de energia;
- Indicador de Qualidade de *Link* (LQI);
- Podem operar em 16 canais na frequência de 2.4GHz, 30 canais na frequência de 915MHz e 3 canais na frequência de 868MHz;

Divide-se em camadas (*layers*) baseado no modelo OSI para facilitar a compreensão, contém a camada física (PHY) que conta com o transceptor de rádio frequência, um mecanismo de controle e uma camada de meio de controle de acesso (MAC) que é responsável pelo serviço de transmissão e recepção de dados, e fornece meios de implementação de mecanismos de segurança apropriados na aplicação.

2.2 Conceitos de Segurança

Segundo [Pfleeger e Pfleeger \(2002\)](#) o termo “segurança” é utilizado de diversas maneiras no dia a dia, entretanto todos esses termos tem seus significados de acordo com o contexto utilizado. Na computação o termo segurança visa tratar três aspectos importantes: confidencialidade, integridade e disponibilidade.

- **Confidencialidade:** Assegura que as informações serão acessadas apenas por pessoas autorizadas. Também pode ser chamado de privacidade.
- **Integridade:** Significa que os dados podem ser apenas modificadas por pessoas ou por meios autenticados. Neste contexto, a modificação inclui escrita, alteração, exclusão e criação.
- **Disponibilidade:** Significa que as informações podem ser apenas acessadas por pessoas autorizadas ou em horários permitidos. O oposto de disponibilidade que será citado ao decorrer do trabalho é negação de serviço.

A seguir será conceituado os ataques que foram citados no desenvolvimento deste trabalho.

Tabela 1 – Tabela de Ataques

Camada	Ataques	Defesas Conhecidas
Física	<i>Jamming</i>	Não Aplicável
	Escuta Passiva	Criptografia
Link	Colisão de Pacotes	Implementações no código
	Exaustão	Limitação de Envios
Rede e Roteamento	<i>Man in the Middle</i>	Autorização e Monitoramento
Transporte	<i>Flooding</i>	Autenticação

2.3 Ataque de Negação de Serviço

Ataque de negação de serviço, ou *Denial Of Service (DoS)* segundo [Wood e Stankovic \(2002\)](#) são tipos de ataques que podem comprometer a rede de forma crítica por meio de perturbação na rede ou até mesmo na invalidação da mesma. Esse tipo de vulnerabilidade pode ser causada por exploração de falhas de hardware, bugs no software, exaustão dos recursos dos nós ou até mesmo condições do ambiente.

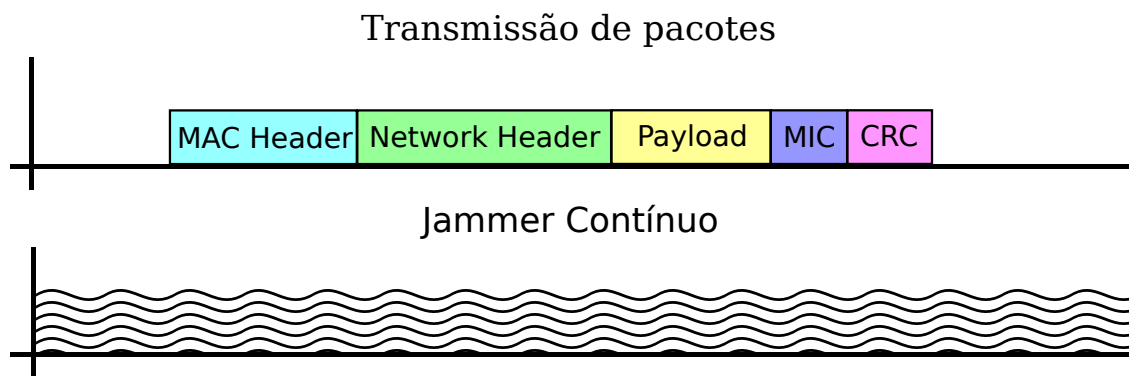
2.4 *Jamming*

Muitos dispositivos são dependentes de redes de comunicação sem fio, porém o bloqueio desses sinais tornando-os indisponíveis é denominado *Jamming*. É uma grande

ameaça para redes sem fio e entender a complexidade deste tipo de ataque e suas contra medidas é de suma importância, já que este é um ataque contra a disponibilidade do sinal e há diversos tipos de *jammers* (WILHELM et al., 2011).

Como explicado por Alturkostani et al. (2015) há diferentes tipos de *jammers*, e entre eles os que mais se destacam são: *jammers* constantes por ser o mais utilizado e os *jammers* inteligente, que, por analisar o tráfego, alvejam pacotes específicos e emitem ruídos para corromper o conteúdo dos mesmos.

Figura 2 – *Jammer* Constante



Fonte – Elaborada pelo autor.

2.5 Escuta Passiva

Neste tipo de ataque, o atacante monitora a rede sem fio utilizando um software em busca de informações que são trafegadas através de uma antena direcional no modo monitor e no canal onde a rede está operando. Estes tipos de ataques não podem ser facilmente detidos por medidas de seguranças de *software*, como proteger os dados trafegados através de criptografia, entretanto, este é um modo de mitigar. Assumindo uma rede sem a proteção de criptografia em seus pacotes enviados pela rede, o atacante ganha informações importantes utilizando esse ataque. O invasor consegue analisar os pacotes e descobrir seu remetente, destinatário, tamanho e tempo de transmissão. O impacto deste ataque não é um risco apenas à privacidade, mas sim é uma pré-condição para ataques mais nocivos (WELCH; LATHROP, 2003) (YUAN et al., 2008).

As *WMNs* são suscetíveis à escuta interna por meio de seus nós intermediários, onde um nó malicioso pode manter uma cópia de todas as informações encaminhadas para ele sem o conhecimento dos outros nós da rede, entretanto, este ataque não interfere diretamente na funcionalidade da rede, apenas compromete a privacidade e integridade dos dados. A criptografia dos dados é implementada utilizando uma chave para proteger os dados e manter sua integridade e privacidade (SEN, 2013).

2.6 *Man in The Middle*

Um ataque MITM (*Man in The Middle*) segundo [Hwang et al. \(2008\)](#) tem como conceito fundamental entrar no meio de uma conexão tornando-se uma ponte entre o cliente e o destinatário, portanto o mesmo envia os pacotes normalmente para o destinatário e pode realizar funções ilegais na rede, abrindo a possibilidade de diversos ataques.

Figura 3 – *Man in The Middle*



A seguir será explicado o fluxo de como o ataque ocorre na teoria.

1. A comunicação entre o cliente e o ponto de acesso é monitorada para conseguir as informações necessárias como o *SSID* e o canal do mesmo.
2. O atacante inicia um ponto de acesso com as mesmas informações do ponto de acesso autêntico.
3. O acesso entre o cliente e o ponto de acesso é interrompido por um jamming ou alguma técnica que cause a indisponibilidade do AP.

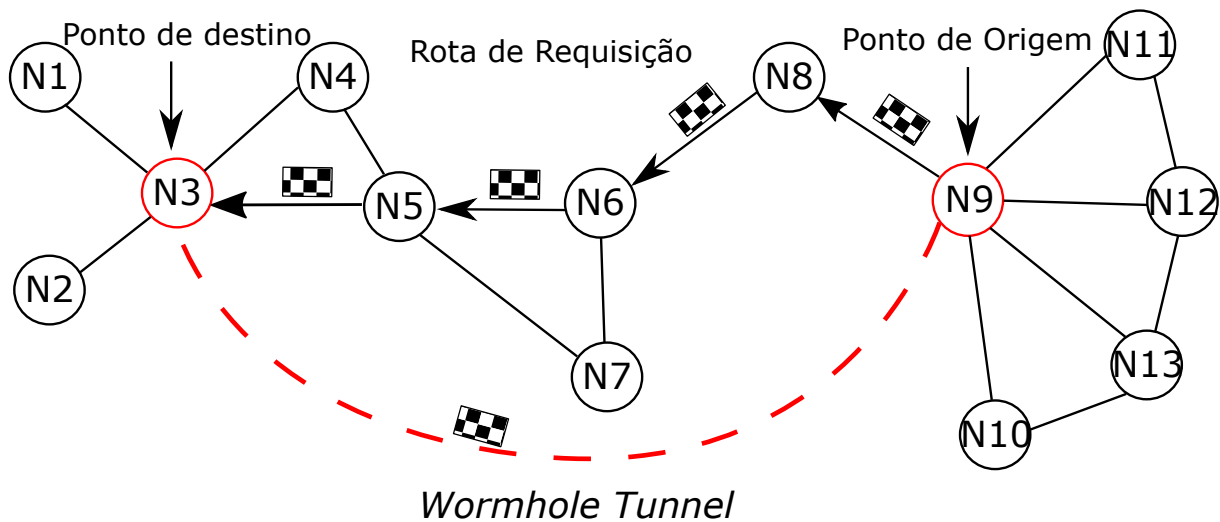
4. O cliente tenta se autenticar com o mesmo *SSID* da rede autêntica, entretanto não encontra a rede original e se conecta na rede do atacante que possui as mesmas informações.
5. Os pacotes enviados do cliente são interceptados pelo atacante e replicados para o ponto de acesso autêntico.

2.6.1 Wormhole Attack

O *wormhole attack* ou ataque de tunelamento é um dos mais poderosos e severos ataques em *WMNs*. Os métodos tradicionais de segurança, como criptografia e *digital signature* podem prevenir o comprometimento, integridade e privacidade dos pacotes que estão sendo trafegados, porém o *wormhole attack* é transparente para esses métodos (ZHOU et al., 2012).

O *wormhole* coloca o atacante em uma posição muito privilegiada, ele é capaz de explorar a rede e realizar diversos ataques, permitindo que o invasor obtenha acesso sem autorização, ou rompa do roteamento, ou até mesmo causar indisponibilidade na rede (HU; PERRIG; JOHNSON, 2003).

Figura 4 – *Wormhole Attack*



Fonte – Elaborada pelo autor.

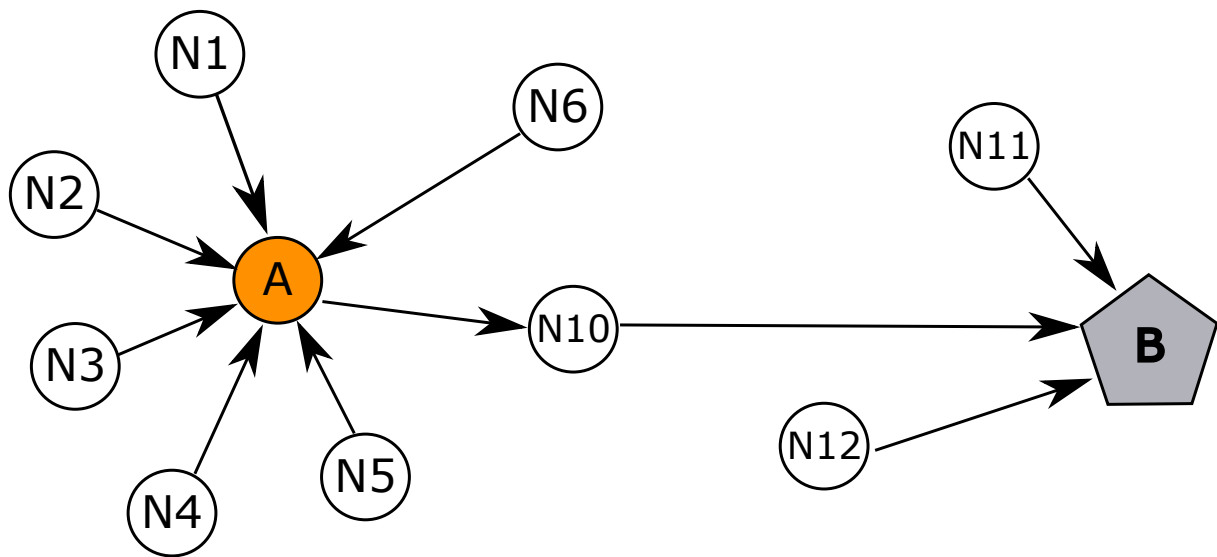
O invasor pode encaminhar as mensagens recebidas através de um meio com baixa latência e o replica em uma parte diferente da rede. Geralmente envolvem dois nós maliciosos distantes entre si transmitindo pacotes ao longo de um meio distinto disponível apenas para os nós maliciosos. Em alguns casos o atacante consegue romper totalmente o roteamento através de um *wormhole* bem implementado, e isso resulta na possibilidade de convencer os nós de que a rede está funcionando normalmente e que seriam múltiplos saltos a partir de uma estação base, sendo que eles estão apenas um ou dois saltos de

distância através do *wormhole*. O mesmo pode ser utilizado para criação de *sinkholes* pela potencial atratividade do roteamento de um nó comprometido ou malicioso (KARLOF; WAGNER, 2003).

2.6.2 Sinkhole Attack

Estes ataques comprometem a aparência de um nó fazendo com que se torne atrativo para os nós vizinhos. Fazem isto respeitando o algoritmo de roteamento da rede com o principal objetivo de prevenir que a estação principal receba o dado completo e correto (NGAI; LIU; LYU, 2006). O invasor pode encaminhar as informações para a estação principal através de um roteamento de alta qualidade fazendo com que os protocolos que visam a confiabilidade ou a baixa latência prefiram aquele nó comprometido para enviar as informações para a estação principal. Há a possibilidade do invasor utilizar um *wormhole* ao invés de um roteamento de alta qualidade. (SALEHI et al., 2013)

Figura 5 – Sinkhole Attack; (a) nó comprometido; (b) Estação principal.



Fonte – Elaborada pelo autor.

O *sinkhole attack* pode abrir oportunidades para diversos ataques diferentes, entre eles o encaminhamento seletivo das informações e o *blackhole*. O mesmo pode ser lançado realizando dois passos: primeiro, o nó comprometido identifica seus vizinhos que tem as menores distâncias em relação a estação principal, e em seguida envia informações de que o nó comprometido tem a menor distância. Realizando os passos anteriores, o atacante consegue mudar o mecanismo de roteamento e o nó adversário passa a funcionar como um *sinkhole*. Após um lançamento de sucesso, certificando que o tráfego da área selecionada passe pelo nó comprometido, o invasor pode suprimir ou modificar as mensagens originadas de qualquer nó na área (QI et al., 2012).

3 Desenvolvimento

Neste capítulo serão analisados, classificados e demonstrados as ameaças que violam os critérios de segurança de redes sem fio. Para isso será utilizado um cenário de testes isolado para a investigação e mitigação dos mesmos.

O critério utilizado para classificar os ataques serão: Explorabilidade, detectabilidade e os impactos.

3.1 Escuta Passiva

A escuta passiva é uma ameaça em redes *IoT* na qual não é possível detectar vide [seção 2.5](#). É uma ameaça que compromete a confidencialidade da rede e é classificada como uma vulnerabilidade de alto risco, por ser fácil de ser explorada e indetectável.

3.1.1 Metodologia do ataque

Para se realizar uma escuta passiva em uma *WMN* é necessário ter um adaptador em modo promíscuo, na qual permita apenas receber os pacotes que estão no alcance. Com o auxílio do software visualizador de pacotes *Wireshark* é possível visualizar os pacotes que o adaptador irá capturar, Vale ressaltar que o canal a ser observado deve ser a mesma que o da rede alvo.

Figura 6 – Adaptador USB da Atmel para baixa frequência.



Fonte – Elaborada pelo autor.

Não foi utilizada uma descoberta de redes automática para identificar em qual canal a rede está operando por causa do firmware desenvolvido no Adaptador USB da Atmel, logo foi percorrido todos os canais possíveis para descobrir se há alguma rede operando e em qual canal.

Após a identificação do canal e a rede a ser observada, ao analisar as informações dos pacotes capturados, é possível obter informações sensíveis da rede, como o *SSID*, endereço de destino e de origem e o conteúdo do pacote.

Figura 7 – *Pacote capturado pelo sniffer.*

```
> Frame 466351: 68 bytes on wire (544 bits), 66 bytes captured (528 bits) on interface 0
▼ IEEE 802.15.4 Data, Dst: 0x0000, Src: 0x0002
  ▼ Frame Control Field: 0x8861, Frame Type: Data, Acknowledge Request, Intra-PAN,
    Destination Addressing Mode: Short/16-bit, Source Addressing Mode: Short/16-bit
      .... .001 = Frame Type: Data (0x0001)
      .... .0... = Security Enabled: False
      .... .0... = Frame Pending: False
      .... .1. .... = Acknowledge Request: True
      .... .1.. .... = Intra-PAN: True
      .... 10.. .... = Destination Addressing Mode: Short/16-bit (0x0002)
      ..00 .... .... = Frame Version: 0
      10.. .... .... = Source Addressing Mode: Short/16-bit (0x0002)
    Sequence Number: 47
    Destination PAN: 0x1444
    Destination: 0x0000
    Source: 0x0002
  ▼ Data (57 bytes)
    Data: 01eb020000001101010200000000000000002000001010100...
    [Length: 57]
```

0000	61 88 2f 44 14 00 00 02	00 01 eb 02 00 00 00 11	a./D....
0010	01 01 02 00 00 00 00 00	00 00 02 00 00 01 01 01
0020	00 80 00 00 44 14 0f 00	00 00 00 01 0c 5f 76 00D..._v.
0030	00 0f 00 00 00 94 00 00	00 20 07 52 6f 75 74 65Route
0040	72 32		r2

Fonte – Elaborada pelo autor.

Como visto na [Figura 7](#), o *sniffer* utilizado obteve os pacotes que trafegavam na rede com sucesso. Com os pacotes obtidos foi possível obter o endereço da rede, o número do pacote, o endereço do remetente e destinatário e o conteúdo do pacote, com um estudo acima dos protocolos estudados para redes de *IoT* foi possível concluir que não há um meio de proteger as informações do cabeçalho, só há como proteger o conteúdo (*payload*) do pacote.

3.1.2 Implementação da mitigação

O principal meio para a mitigação deste problema é utilizar uma criptografia para garantir a privacidade, integridade e a autenticidade da informação.

A implementação de uma criptografia em microcontroladores pode ser feita na camada de *hardware*, ou também na camada de aplicação. Na camada de aplicação a implementação pode ser realizada através da inserção de um algoritmo de criptografia, entretanto pode ocorrer perdas de desempenho ao utilizar recursos do microcontrolador para encriptar e desencriptar. Já na camada de *hardware* a encriptação e a desencriptação é feita através de um *hardware* dedicado sem influenciar o desempenho do microcontrolador, logo recomenda-se o uso de uma criptografia com aceleração de *hardware*.

```
1 // Release transmission with security
2 appNwkAddrReq.dstAddr = 0;
3 appNwkAddrReq.dstEndpoint = APP_ENDPOINT_DYN_ADDR;
4 appNwkAddrReq.srcEndpoint = APP_ENDPOINT_DYN_ADDR;
5 appNwkAddrReq.options = NWK_OPT_ACK_REQUEST |
    NWK_OPT_ENABLE_SECURITY;
6 appNwkAddrReq.data = (uint8_t *) &AppMsgDynAddr;
7 appNwkAddrReq.size = sizeof(AppMsgDynAddr);
8 appNwkAddrReq.confirm = appRelsConf;
9
10 NWK_DataReq(&appNwkAddrReq);
```

O processo de encriptação e desencriptação não é explícito, ele apenas reconhece que o pacote deve ser encriptado na hora do envio ou desencriptado no recebimento pela opção *NWK_OPT_ENABLE_SECURITY* nas opções do pacote onde foi definido na linha 5 do trecho de código acima.

3.2 Man In The Middle

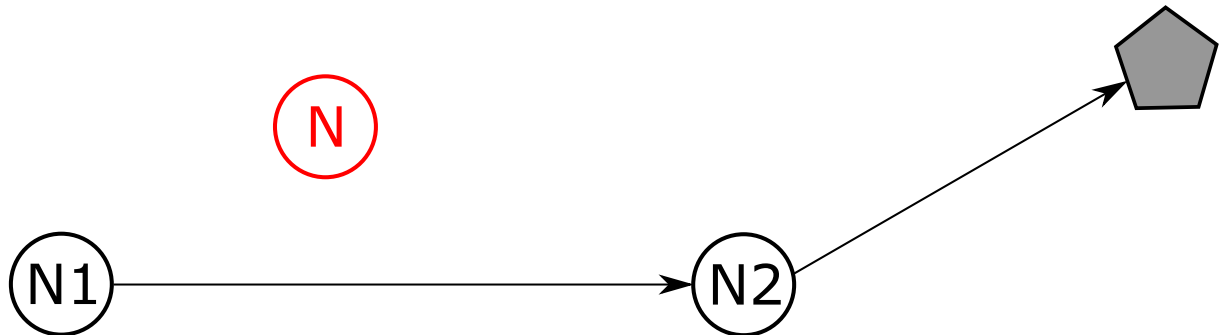
Este ataque compromete a integridade, confidencialidade e a disponibilidade da rede, pois o atacante terá acesso à rede como um dispositivo autenticado e o mesmo abre oportunidades para diversos tipos de ataques, logo é considerado uma vulnerabilidade de alto risco.

3.2.1 Metodologia do ataque

Utilizando a técnica de *sniffing* citado na [seção 2.5](#), é possível obter o *SSID* (*Service Set Identifier*) da rede alvo, e assim é possível criar um nó com um endereço aleatório onde ele pode se conectar na rede e pelo algoritmo de roteamento, o mesmo consegue

receber os pacotes que trafegam na rede. Com isso, o atacante consegue explorar essa falha e encadear diversos tipos de ataques diferentes que serão citados ao decorrer do trabalho.

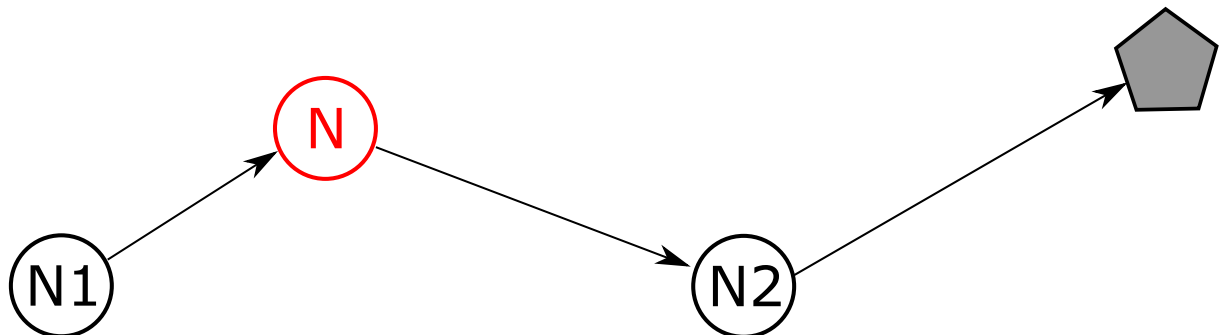
Figura 8 – Nó estrategicamente posicionado para realizar o ataque.



Fonte – Elaborada pelo autor.

Se a rede não possuir alguma proteção contra a leitura de pacotes, o invasor tem uma chance muito grande de obter a informação de como é a estrutura e a forma de comunicação entre os nós e tornando possível a realização de uma injeção de informações erradas na base de dados por enviar informações aparentemente autênticas ou até mesmo a injeção de comandos nos nós de forma ilegal. Fazendo assim com que o nó seja integrado na rede como mostra na [Figura 9](#).

Figura 9 – Nó malicioso acoplado na rede.



Fonte – Elaborada pelo autor.

Pode se classificar este ataque como um ataque de alta ameaça, de risco moderado e complexidade alta.

3.2.2 Implementações da mitigação

Investigando mais a fundo redes sem fio, foi concluído que é necessário para o bom funcionamento da rede um serviço de entrega de endereços, e para complementar é recomendado usar em conjunto com o padrão de *whitelist* documentado por [Villarreal et al. \(2013\)](#). O objetivo desta *whitelist* é armazenar o endereço único de todos os nós da rede,

para que apenas nós autênticos recebam um endereço definido pelo coordenador da rede, evitando assim que dispositivos maliciosos tenham acesso a esse serviço e consequentemente acesso à rede.

Cada nó terá que receber um endereço para se conectar com a rede, e seu endereço único gerado pela aplicação será gravado no ato. Se o nó coordenador receber um pacote de um nó que não está cadastrado na rede, o mesmo será descartado. Os nós da rede também terão essa política, eles apenas aceitarão pacotes que vem do endereço único do coordenador.

A implementação desta *whitelist* se dá por uma estrutura para o armazenamento dos endereços únicos, no nosso caso será utilizada uma lista de 128 posições de *unsigned int* de 16 bits (*uint16_t*) para armazenar 128 endereços únicos dos nós.

Tabela 2 – Exemplo da *whitelist*

Índice	Endereço único
0	51348
1	24867
2	84661
3	0

Como pode ser visto na [Tabela 2](#), o índice é o endereço do nó subtraído 1, e nesse índice está armazenado seu endereço únicos. Se a aplicação que roda no microcontrolador encontrar um índice cujo valor é 0, significa que o mesmo está disponível e pronto para ser armazenado um endereço únicos. No caso da *whitelist* acima, o endereço válido será o 4, pois é o índice 3 que está vago.

Ao receber um pacote, o nó deverá abrir o cabeçalho do pacote e verificar se seu endereço de identificação e seu endereço únicos são válidos, e essa informação está armazenada na *whitelist*, ou seja, o microcontrolador deverá percorrer essa lista para a verificação.

3.3 Flooding

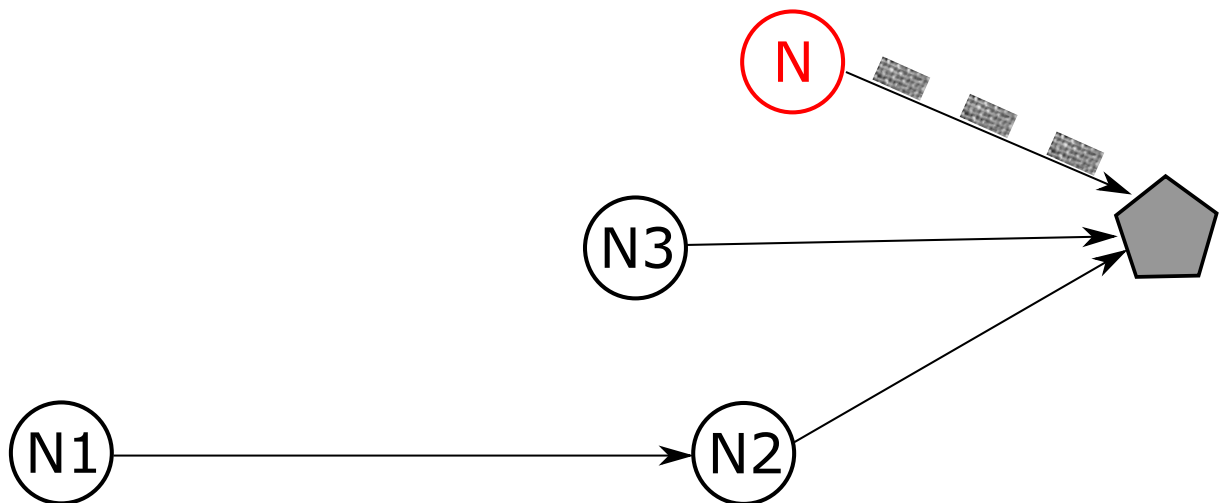
O *flooding* é uma vulnerabilidade da camada física que é relativamente fácil de ser lançada na camada de transporte de uma rede, que requer o mínimo de informações da rede que são facilmente obtidas através de um *sniffer* visto na [seção 2.5](#). Tendo em mãos o *SSID* da rede, o que é mais demorado de se conseguir na rede é o canal em que a mesma está operando, já que o *sniffer* tem que vasculhar canal por canal em busca de pacotes (O que em alguns casos pode demorar). É uma vulnerabilidade de médio risco na qual tem o objetivo de causar a indisponibilidade da rede segundo: [seção 2.3](#).

3.3.1 Metodologia do ataque

Para iniciar o ataque, o invasor precisa utilizar a técnica citada na [seção 2.6](#), entretanto o mesmo não precisa estar autenticado à rede, o mesmo precisa apenas estar na mesma rede com um endereço válido para conseguir enviar os pacotes em um *endpoint* aberto.

O atacante por meio da repetição de alguma requisição ou pacote, pode levar a rede ou algum nó à exaustão. O atacante pode fazer desde uma requisição de conexão repetitivamente ou enviar diversos pacotes vazios para ocupar a rede com o objetivo de fazer com que os nós deixem de responder, ou fazer o *gateway* se sobrecarregar e consequentemente deixando de funcionar como exemplificado na [Figura 10](#).

Figura 10 – Nó malicioso realizando *flooding*.



Fonte – Elaborada pelo autor.

Pode se classificar este ataque como um ataque de baixa ameaça, com risco moderado e complexidade moderada.

3.3.2 Implementações da mitigação

Como este ataque tem como principal objetivo causar exaustão na rede, um coordenador potente para atender diversas requisições pode ser uma solução, entretanto se combinado com uma limitação de resposta, se torna mais efetivo ainda. Para limitar a resposta, foi criada e implementada uma política na qual o coordenador só irá responder nós autênticos e listados na *whitelist* citado na [seção 2.6](#).

As políticas nos dispositivos da rede em malha tem importância na mitigação desta vulnerabilidade. Os *End Devices* são dispositivos que geram informações, que por ficarem em baterias eles não ficam sempre disponíveis, os mesmos ficam ausentes até o envio de alguma informação. Já os *Routers* são responsáveis pela propagação da informação pela

rede até chegar no seu destino, todavia, diferente dos *End Devices*, eles ficam ligados em uma fonte de energia estável, então não há problemas de energia com os mesmos.

Visando que um ataque desse tipo é emitido para causar a exaustão contínua, foi implementado um sistema na qual se um endereço enviar uma requisição e o endereço único não conferir com o cadastrado na *whitelist* citado na [seção 3.2](#), o coordenador emitirá um alerta para notificar a aplicação de um possível ataque.

3.4 *Jamming*

Jamming é um ataque que atinge a camada física na rede e tem o objetivo causar a indisponibilidade da rede. É uma ameaça de alto risco e que pode ser utilizado para desencadear outros ataques, como por exemplo é possível realizar um *Man in The Middle* citado na [seção 2.6](#) despercebido se a rede não possuir alguma implementação de segurança que bloqueie a conexão do nó invasor. Este ataque é de alto risco tanto para a confidencialidade tanto para a disponibilidade da rede. Pode ser considerada uma vulnerabilidade de risco médio.

3.4.1 Metodologia do ataque

Para realizar um *jamming* é necessário apenas uma placa programada para transmitir em uma potência alta diversos ruídos na rede. Em uma rede pequena como em *Smart Home* causa indisponibilidade total dependendo do tamanho físico da rede, já em redes maiores, como *smart cities*, causa a indisponibilidade de alguns nós na rede. O tipo mais comum de *jamming* a ser lançado é o contínuo e usualmente são lançado nos coordenadores, já que o mesmo é o concentrador de toda a rede e é ele o responsável por receber e enviar para a aplicação todas as informações.

A [Figura 11](#) exemplifica o ataque, o nó vermelho sendo o nó malicioso e o círculo pontilhado mostra a área de atuação do *jamming*, na qual bloqueia a comunicação do nó N1 com o resto da rede.

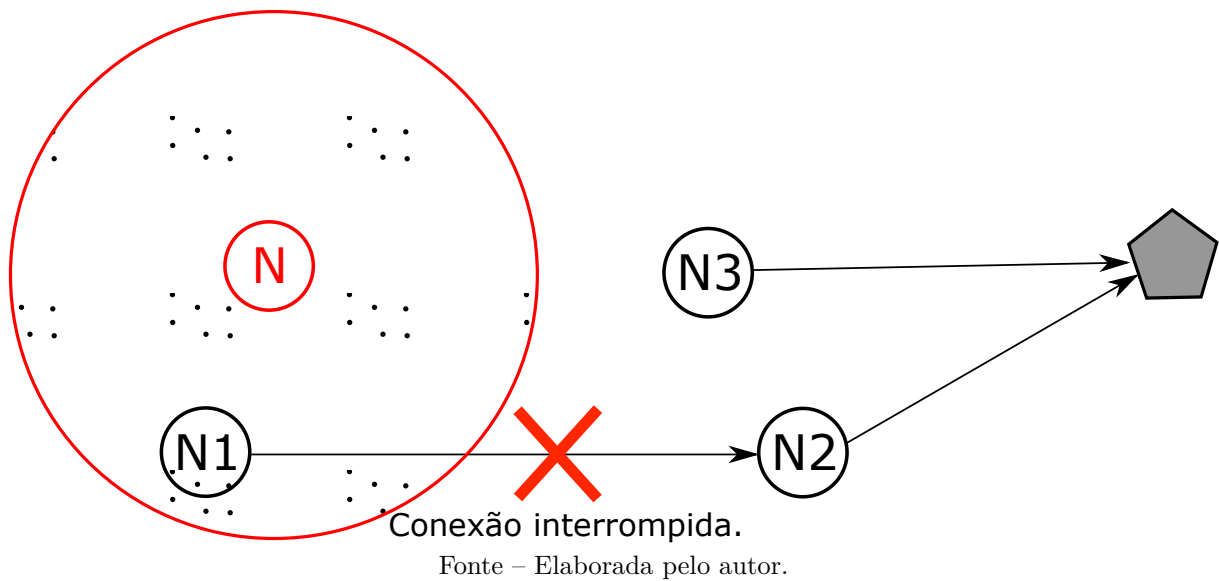
Pode se classificar o *jamming* como um ataque de alta ameaça, com risco moderado e de complexidade moderada.

3.4.2 Implementações da mitigação

Como todos ataques na camada física, o *jammer* ainda não há meio de evitar, entretanto há a possibilidade de se detectar e desativar ataques desse tipo.

Para detectar ataques desse tipo, a identificação dos nós é muito importante, já que há a possibilidade de listar os nós que foram afetados e assim lista-los para a investigação e

Figura 11 – Nó malicioso realizando *jamming* com sua área de abrangência.



a localização aproximada do nó atacante. Pode ser utilizado uma placa de rede sem fio de baixa frequência e um sistema computacional móvel para se obter o sinal do nó atacante e delimitar ainda mais o raio de procura do nó malicioso até se obter a localização do nó e assim desativá-lo.

Para garantir a disponibilidade dos nós, foi implementado um mecanismo de *Keepalive* para garantir que todos os *Routers* retornem sua disponibilidade e informações, para realizar um controle efetivo da saúde da rede para o coordenador, e assim na aplicação gerar um status de cada *Router* para caso algum nó vier a ficar *offline*, será possível deduzir que há algo de errado acontecendo, entretanto, utilizando o mecanismo de *Keepalive*, abre oportunidades para o *sniffing* documentado na [seção 2.5](#), já que os roteadores irão enviar pacotes de *keepalive* com um intervalo de tempo fazendo com que o *sniffer* obtenha uma quantidade maior de pacotes.

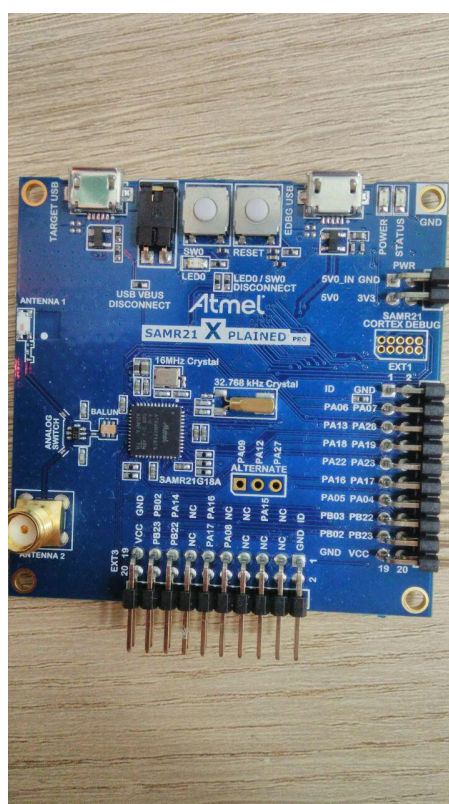
Vale a pena ressaltar que a base de informação para todos os ataques citados é o levantamento de informações da rede utilizando a técnica de *sniffing* citado na [seção 2.5](#).

4 Casos de Testes

Os principais componentes do cenário de testes serão transceptores da *Microchip Atmel* conectados em uma rede sem fio na topologia malha utilizando o protocolo *Lightweight Mesh (LWMESH)*.

Os transceptores da *Microchip Atmel* utilizados nos testes é o *SMART SAM R21 Xplained Pro*.

Figura 12 – *SMART SAM R21 Xplained Pro*



Fonte – Elaborado pelo autor.

Tabela 3 – Especificações técnicas do microcontrolador SMART SAM R21

Processador:	32-bit ARM [®] Cortex [®] - M0+ de 48MHz
Transceptor:	Ultra-low-power 2.4GHz ISM
Memória Flash:	256KB
SRAM:	32KB

4.1 Projeto de Iluminação Pública (*StreetLight*)

O projeto demonstrativo de iluminação pública é uma rede *mesh* na qual todos os nós são *Routers* e há um coordenador na qual recebe as informações dos nós para o gerenciamento da saúde da rede e também recebe os comandos para ligar e desligar as luzes, transmitindo assim para o respectivo microcontrolador. A disposição física dos nós nos testes é a mostrada na [Figura 13](#).

Figura 13 – *Routers* simulando postes



Fonte – Elaborado pelo autor.

4.1.1 Topologia da Rede

A rede utiliza a topologia malha onde os nós são *Routers* e tem como coordenador um microcontrolador na central estipulada. O protocolo utilizado para este caso foi o *LWMesh*, que é uma pilha de protocolo semelhante ao *TCP/IP*, com 3 camadas, sendo elas a camada física, a camada de rede e a de aplicação. Vale ressaltar que este protocolo foi desenvolvido para ser utilizada em redes de sensores em malha.

A rede utiliza o conceito de *fog computing* que não é diretamente ligado à internet. A rede interna envia as informações para o coordenador, que é o único com contato com a internet que por sua vez irá enviar as informações para o servidor.

4.1.2 Comportamento dos componentes da Rede

Os *Routers* estão ligados a um poste na qual estão teoricamente sempre com energia elétrica, evitando assim o consumo exaustivo da bateria contra os ataques citados na [seção 3.3](#) e na [seção 2.4](#). Todos os microcontroladores utilizados possuem uma criptografia com aceleração de *hardware* utilizando *AES-128* cujo núcleo está de acordo com padrão FIPS197 ([US DEPARTMENT OF COMMERCE/NIST, 2001](#)).

Os *Routers* tentam enviar os dados, se eles não receberem o pacote de *Acknowledgement (ACK)* do pacote que foi enviado, os mesmos guardarão as informações e tentarão enviar posteriormente até que o coordenador envie o pacote de *ACK* dos dados para confirmar o recebimento.

Os nós da rede possuem todos os meios de mitigação citados no [Capítulo 3](#), e a seguir será documentado uma bateria de testes de penetração e de disponibilidade.

4.1.3 Teste de *Sniffing*

No teste de escuta passiva será colocado um *sniffer* para capturar os pacotes que trafegam na rede, e espera-se que os pacotes estejam protegendo devidamente seu conteúdo.

Figura 14 – Pacote capturando utilizando *sniffer* na rede *Streetlight*

```
> Frame 1264: 26 bytes on wire (208 bits), 24 bytes captured (192 bits) on interface 0
  IEEE 802.15.4 Data, Dst: 0x0000, Src: 0x0002
    Frame Control Field: 0x8861, Frame Type: Data, Acknowledge Request, Intra-PAN, Destination Addressing Mode: Short/16-bit, Source Addressing Mode: Short/16-bit
      Sequence Number: 76
      Destination PAN: 0x1234
      Destination: 0x0000
      Source: 0x0002
    Lightweight Mesh, Sequence Number: 75
      Frame control field: 0x03
      Sequence Number: 75
      Network Source Address: 0x0002 (Routing node)
      Network Destination Address: 0x0000 (Unicast) (Routing node)
      Source Endpoint: 3
      Destination Endpoint: 3
      Message Integrity Code: 0xf11877d7
      [Expert Info (Note/Protocol): No encryption key set - can't decrypt]
      Data (4 bytes)
        Data: 4ee46a86
        Text: N\357\277\275j\357\277\275
        [Length: 4]

0000  61 88 4c 34 12 00 00 02 00 03 4b 02 00 00 00 33  a.L4.... ..K....3
0010  4e e4 6a 86 d7 77 18 f1                          N.j..w..
```

Fonte – Elaborado pelo autor.

No teste foi possível obter as seguintes informações: o endereço de destino e origem, o *SSID* ou *PANID* da rede e o *endpoint* utilizado, entretanto o conteúdo do pacote foi devidamente protegido como pode-se observar na [Figura 14](#).

O objetivo de proteger o conteúdo do pacote foi cumprido com sucesso, entretanto com as demais informações que não foram possíveis de proteger, pode-se arriscar a tarefa de penetrar na rede como um nó não autenticado, todavia como não foi possível visualizar como a rede se comunica, o atacante não conseguirá ver a estrutura dos pacotes, logo não conseguirá enviar comandos para os nós.

4.1.4 Teste de *Man In The Middle*

No teste de *MITM* será colocado um nó malicioso para tentar se integrar na rede e desta forma enviar comandos para o resto da rede, e espera-se que a rede não aceite os pacotes enviados e que utilize o nó malicioso para rotear as informações para o seu destino.

Para o atacante, sem a informação da estrutura dos pacotes e de como os dados trafegam, será tecnicamente inviável realizar esta abordagem na rede, entretanto se considerarmos que o atacante tenha conhecimento do método de autenticação que é realizado na rede e obtenha um endereço *MAC* válido, é possível penetrar na rede com um endereço *MAC* aleatório, entretanto ele não conseguirá enviar comandos para os demais nós da rede como mostra na [Figura 15](#), logo, podemos ver na última linha que o coordenador nocivo retornou 0, o que significa que não foi validado o pacote enviado.

Figura 15 – Coordenador inválido enviando um pacote malicioso.

```
*****STREET LIGHT*****LF
COORD U2LF
ShortAddr 0LFCR
MacAddr 43103LFCR
APP Init - >>> ShortAddr 0LF
MacAddr 43103LF
PanID 4660LF
gID 0LF
SEND MSG_CMD_STRLIGHTLF
#0LF
```

Fonte – Elaborado pelo autor.

Isso acontece porque seu endereço *MAC* não será correspondente ao do nó autêntico como mostra na figura 16. Se o nó apenas se conectar na rede, a aplicação notificará ao receber a informação de que um endereço enviou uma quantidade maior de *keepalive* em um tempo determinado ou que um nó enviou algum comando com o endereço *MAC* inválido.

Figura 16 – Nó alvo recebendo o pacote e checando o endereço *MAC* de quem enviou.

```
*****STREET LIGHT*****LF
DEVICE U2LF
ShortAddr 2LFCR
MacAddr 32815LFCR
APP Init - >>> ShortAddr 2LF
MacAddr 32815LF
PanID 4660LF
gID 29577LF
gID 29577LF
MsgMac 43103LF
```

Fonte – Elaborado pelo autor.

Na imagem acima interpreta-se *gID* sendo o endereço do coordenador autêntico e o *MsgMac* como o endereço *MAC* do coordenador falso que enviou o pacote malicioso.

O nó e o coordenador rejeitaram o pacote enviado pelo nó malicioso e a aplicação mantém o controle dos nós da rede utilizando o mecanismo de *keepalive*, logo, se um nó malicioso entra na rede ou tenta enviar um pacote malicioso, o pacote não será aceito e a aplicação alertará o usuário. A vulnerabilidade foi mitigada com sucesso.

4.1.5 Teste de *flooding*

Para o teste de *flooding* espera-se que o nó aguarde a carga de dados enviados por um dispositivo malicioso, e que utilize-se de métodos para evitar que o nó se sobrecarregue.

Figura 17 – Pacotes do *flooding* capturados pelo *sniffer*

No.	Time	Source	Destination	Protocol	Lengt	Info
1645	13:28:17.154755	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1649	13:28:17.667756	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1653	13:28:18.172817	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1657	13:28:18.681058	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1659	13:28:19.213487	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1663	13:28:20.228698	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1667	13:28:20.735023	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1671	13:28:21.233849	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1675	13:28:21.740219	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1679	13:28:22.254432	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1683	13:28:22.763738	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1687	13:28:23.268704	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1691	13:28:23.765611	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1695	13:28:24.775367	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1699	13:28:25.296737	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1703	13:28:25.808566	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1707	13:28:26.311698	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1711	13:28:26.810551	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1715	13:28:27.316910	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1719	13:28:27.823768	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1723	13:28:28.328626	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1727	13:28:28.827980	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1731	13:28:29.333838	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1735	13:28:29.853706	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1739	13:28:30.328042	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1743	13:28:30.832677	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1747	13:28:31.329766	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1751	13:28:32.334618	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1755	13:28:32.830697	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1759	13:28:33.337517	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1763	13:28:33.840981	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1767	13:28:34.346991	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1771	13:28:34.851760	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1775	13:28:35.354849	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1777	13:28:35.891684	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1781	13:28:36.397839	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005
1785	13:28:36.894062	0x0005	0x0000	LwMesh	30	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0005

Fonte – Elaborado pelo autor.

O teste da vulnerabilidade de *flooding* foi realizado e como previsto anteriormente no Capítulo 3, o nó continuou recebendo os pacotes, entretanto não processou os mesmos, pois a estrutura dos pacotes não foi a mesma da rede. Utilizando o *header* do pacote para filtrar a entrada de pacote, foi possível emitir um alerta para a aplicação, logo se o atacante tiver a estrutura de como os dados trafegam, ele ainda não conseguiria fazer com que o nó processasse o pacote.

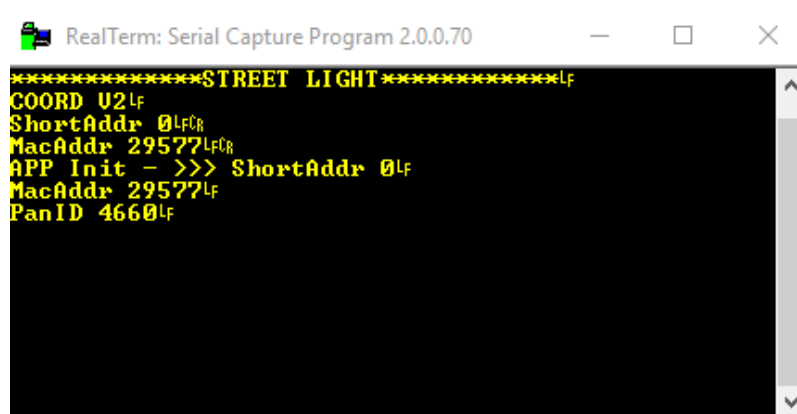
De acordo com o que foi realizado, não foi possível evitar o ataque, entretanto foi possível detectar o ataque e diminuir o impacto que ele causa na rede. Como meio de mitigação foi enviado um alerta para a aplicação para o usuário ficar ciente que está acontecendo uma ataque.

4.1.6 Teste de *Jamming*

O ambiente simulado foi uma *smart home*, um ambiente de $25m^2$ onde há um coordenador e um nó, o *jammer* será colocado próximo ao coordenador. Espera-se que o coordenador não receba nenhum dado.

Ao iniciar o *jammer*, a rede se tornou totalmente indisponível, já que foi lançada em um perímetro relativamente pequeno e perto do coordenador.

Figura 18 – Comportamento do coordenador no campo do *jamming*

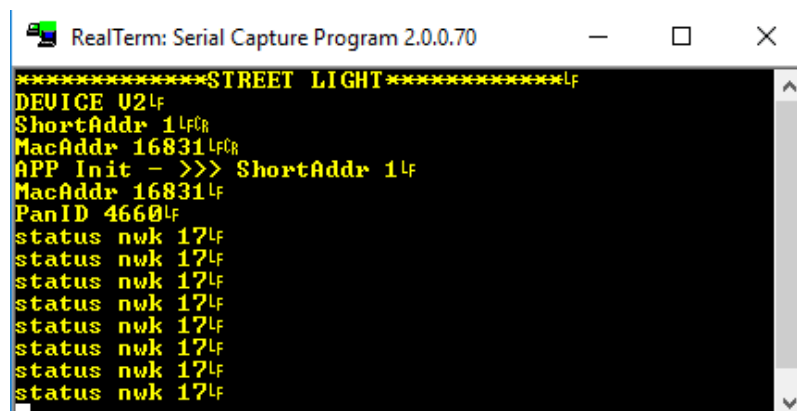


```
RealTerm: Serial Capture Program 2.0.0.70
*****STREET LIGHT*****Lf
COORD U2Lf
ShortAddr 0Lf
MacAddr 29577Lf
APP Init - >>> ShortAddr 0Lf
MacAddr 29577Lf
PanID 4660Lf
```

Fonte – Elaborado pelo autor.

O coordenador não conseguiu obter nenhum pacote como mostra a [Figura 18](#), pois o mesmo estava na área de atuação do jamming, diferente do *router*, que não estava no alcance do *jammer* e conseguiu enviar os pacotes como podemos ver a [Figura 19](#), entretanto o mesmo retornou um erro informando que não foi possível alcançar o destinatário do pacote, como mostra na [Figura 19](#).

Figura 19 – Comportamento do *router* fora do campo do *jamming*



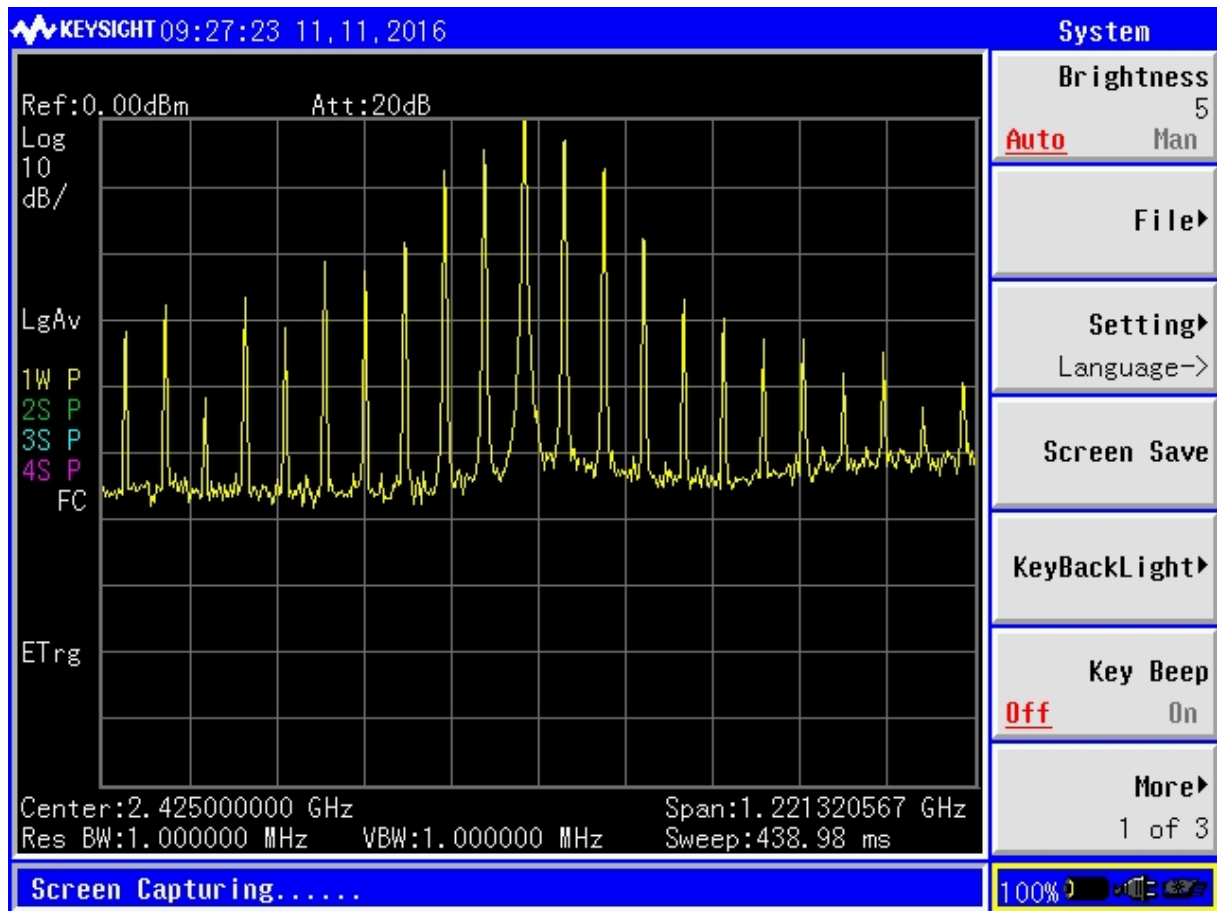
```
RealTerm: Serial Capture Program 2.0.0.70
*****STREET LIGHT*****Lf
DEVICE U2Lf
ShortAddr 1Lf
MacAddr 16831Lf
APP Init - >>> ShortAddr 1Lf
MacAddr 16831Lf
PanID 4660Lf
status nwk 17Lf
status nwk 17Lf
status nwk 17Lf
status nwk 17Lf
status nwk 17Lf
status nwk 17Lf
status nwk 17Lf
status nwk 17Lf
```

Fonte – Elaborado pelo autor.

O *jamming* dos testes age na frequência de $2.4GHz$ no canal 15, no mesmo canal

da rede de testes, a potência do ruído emitido do *jammer* pode ser medido com o uso de um analisador de espectro como pode ser visto na figura 20.

Figura 20 – Análise da potência do ruído emitido utilizando um analisador de espectro.



Fonte – Elaborado pelo autor.

O teste mostrou que não foi possível neutralizar esta vulnerabilidade, entretanto foi possível mitigar fazendo com que os *routers* tentem enviar as informações até o coordenador responder com o pacote de *ACK*, ou seja, quando o *jamming* que foi detectado for desativado, a rede fluirá normalmente e enviará os dados que não foram enviados, sendo assim, sem a perda de dados.

4.2 Considerações

Se o atacante tiver conhecimento de como a rede funciona e como se comunica, para ele lançar qualquer ataque será fácil, logo recomenda-se assegurar as informações no meio físico para que não haja quebra de sigilo ou vazamento de informações sensíveis, desde controle de quem acessa o código fonte dos dispositivos até quem tem acesso ao interior da empresa.

5 Conclusão

Este trabalho teve como objetivo analisar redes de *IoT* utilizando o protocolo *LWMesh* e mitigar as principais vulnerabilidades e falhas de segurança. As pesquisas realizadas mostraram de forma satisfatória que uma rede de sensores na topologia malha, utilizando o protocolo de comunicação *LWMesh*, é possível ter a maior parte de suas vulnerabilidades mitigadas, entretanto foi mostrado que ataques lançados na camada física da rede não são possíveis de serem eliminados, mas sim mitigados pelo impacto de suas consequências na rede.

Os mecanismos implementados na rede de testes provam de forma satisfatória que a rede tem a segurança básica, como o sistema de endereçamento para autenticar os nós, o padrão de *whitelist* para evitar que nós não autenticados tenham acesso à rede, criptografia para não permitir que escutas passivas consigam obter informações privadas e o mecanismo de *Keepalive* para a aplicação ficar ciente sobre o estado dos nós e consequentemente o estado da rede. Pode se levar em consideração que a segurança do meio físico também é muito importante para que não ocorra o vazamento de informações sensíveis, quebra de sigilo ou até mesmo a intrusão no perímetro aonde se localiza o coordenador, logo é possível dizer que a segurança do meio físico também é de grande importância. Com tudo isso, espera-se contribuir para a segurança em redes de *IoT* e de sensores.

As redes de dispositivos *IoT* no levantamento de vulnerabilidades se apresentaram muito frágeis e teoricamente fácil de causar indisponibilidade no serviço na mesma, entretanto por ser uma área ainda em estudo e em constante crescimento, a tendência é que com o tempo, essas redes sejam estudadas a fim de tornar as mesmas mais seguras. Este trabalho teve este propósito, de garantir a proteção da informação trafegada nesta rede utilizando mecanismos conhecidos para tornar a rede um canal seguro e disponível para as informações serem trafegadas e informar ao usuário possíveis indisponibilidades na rede.

Sugere-se uma pesquisa sobre mais mecanismos de proteção para melhorar a segurança já existente e um estudo mais fundo sobre o mecanismo de roteamento para evitar ataques mais complexos, como os ataques de roteamento. Para trabalhos futuros recomenda-se o estudo da segurança em outros protocolos de redes sem fio *Low Energy* baseados em *IEEE 802.15.4* que suportam *IPv6*, como o *Thread* e o *6LoWPAN*, estudo da segurança em redes sem fio utilizando *bluetooth low energy* baseados em *IEEE 802.15.1*, estudo de algoritmos de compactação leves para microcontroladores, pode-se trabalhar também com a segurança física dos dispositivos e com a segurança na aplicação em si que está responsável pelo recebimento dos dados e a exibição dos mesmos. Um ponto

importante a ser trabalhado futuramente são os métodos de mitigação de *jamming* e *friendly jamming*.

Referências

- ALTURKOSTANI, Hani; CHITRAKAR, Anup; RINKER, Robert; KRINGS, Axel. On the design of jamming-aware safety applications in vanets. In: *Proceedings of the 10th Annual Cyber and Information Security Research Conference*. New York, NY, USA: ACM, 2015. (CISR '15), p. 7:1–7:8. ISBN 978-1-4503-3345-0. Disponível em: <http://doi.acm.org/10.1145/2746266.2746273>. Citado na página 21.
- ATZORI, Luigi; IERA, Antonio; MORABITO, Giacomo. The internet of things: A survey. *Computer Networks*, v. 54, n. 15, p. 2787 – 2805, 2010. ISSN 1389-1286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128610001568>. Citado 3 vezes nas páginas 15, 16 e 17.
- COMMITTEE, LAN/MAN Standards et al. Part 15.4: wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans). *IEEE Computer Society*, 2006. Citado na página 19.
- HU, Yih-Chun; PERRIG, Adrian; JOHNSON, David B. Packet leases: a defense against wormhole attacks in wireless networks. In: IEEE. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*. IEEE Societies. [S.l.], 2003. v. 3, p. 1976–1986. Citado na página 23.
- HWANG, H.; JUNG, G.; SOHN, K.; PARK, S. A study on mitm (man in the middle) vulnerability in wireless network using 802.1x and eap. In: *Information Science and Security, 2008. ICISS. International Conference on*. [S.l.: s.n.], 2008. p. 164–170. Citado na página 22.
- KARLOF, Chris; WAGNER, David. Secure routing in wireless sensor networks: Attacks and countermeasures. *Ad hoc networks*, Elsevier, v. 1, n. 2, p. 293–315, 2003. Citado na página 24.
- KREBS, Brian. *KrebsOnSecurity Hit With Record DDoS*. 2016. Disponível em: <https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/>. Citado na página 17.
- MALGERI, John. Cyber security: A national effort to improve. In: *2009 Information Security Curriculum Development Conference*. New York, NY, USA: ACM, 2009. (InfoSecCD '09), p. 107–113. ISBN 978-1-60558-661-8. Disponível em: <http://doi.acm.org/10.1145/1940976.1940998>. Citado na página 17.
- MATUSZAK, Gary; BELL, Greg; LE, Danny. *Security and the IoT ecosystem*. [S.l.], 2015. Citado na página 17.
- NGAI, Edith CH; LIU, Jiangchuan; LYU, Michael R. On the intruder detection for sinkhole attack in wireless sensor networks. In: IEEE. *Communications, 2006. ICC'06. IEEE International Conference on*. [S.l.], 2006. v. 8, p. 3383–3389. Citado na página 24.
- PFLEEGER, Charles P; PFLEEGER, Shari Lawrence. *Security in computing*. [S.l.]: Prentice Hall Professional Technical Reference, 2002. Citado na página 20.

- QI, Jin; HONG, Tang; XIAOHUI, Kuang; QIANG, Liu. Detection and defence of sinkhole attack in wireless sensor network. In: IEEE. *Communication Technology (ICCT), 2012 IEEE 14th International Conference on*. [S.l.], 2012. p. 809–813. Citado na página 24.
- SALEHI, S Ahmad; RAZZAQUE, Md Abdur; NARAEI, Parisa; FARROKHTALA, Ali. Detection of sinkhole attack in wireless sensor networks. In: IEEE. *Space Science and Communication (IconSpace), 2013 IEEE International Conference on*. [S.l.], 2013. p. 361–365. Citado na página 24.
- SEN, Jaydip. Security and privacy issues in wireless mesh networks: A survey. In: *Wireless networks and security*. [S.l.]: Springer, 2013. p. 189–272. Citado na página 21.
- SIDDIQUI, Muhammad Shoaib; HONG, Choong Seon. Security issues in wireless mesh networks. In: IEEE. *Multimedia and Ubiquitous Engineering, 2007. MUE'07. International Conference on*. [S.l.], 2007. p. 717–722. Citado na página 17.
- STRATEGY, ITU; UNIT, Policy. Itu internet reports 2005: The internet of things. Geneva: International Telecommunication Union (ITU), 2005. Citado 2 vezes nas páginas 15 e 16.
- US DEPARTMENT OF COMMERCE/NIST. *NIST FIPS PUB 197: Advanced Encryption Standard (AES)*. 2001. Citado na página 35.
- VILLARREAL, Isaura N. Bonilla; FERNANDEZ, Eduardo B.; LARRONDO-PETRIE, Maria M.; HASHIZUME, Keiko. Whitelisting firewall pattern (wlf). In: *Proceedings of the 20th Conference on Pattern Languages of Programs*. USA: The Hillside Group, 2013. (PLoP '13), p. 11:1–11:6. ISBN 978-1-941652-00-8. Disponível em: <<http://dl.acm.org/citation.cfm?id=2725669.2725683>>. Citado na página 28.
- WELCH, Donald J; LATHROP, Scott. A survey of 802.11 a wireless security threats and security mechanisms. *United States Military Academy West Point*, 2003. Citado na página 21.
- WILHELM, Matthias; MARTINOVIC, Ivan; SCHMITT, Jens B.; LENDERS, Vincent. Short paper: Reactive jamming in wireless networks: How realistic is the threat? In: *Proceedings of the Fourth ACM Conference on Wireless Network Security*. New York, NY, USA: ACM, 2011. (WiSec '11), p. 47–52. ISBN 978-1-4503-0692-8. Disponível em: <<http://doi.acm.org/10.1145/1998412.1998422>>. Citado na página 21.
- WOOD, Anthony D; STANKOVIC, John A. Denial of service in sensor networks. *Computer*, IEEE, v. 35, n. 10, p. 54–62, 2002. Citado na página 20.
- YUAN, Xiaohong; WRIGHT, Omari T.; YU, Huiming; WILLIAMS, Kenneth A. Laboratory design for wireless network attacks. In: *Proceedings of the 5th Annual Conference on Information Security Curriculum Development*. New York, NY, USA: ACM, 2008. (InfoSecCD '08), p. 5–12. ISBN 978-1-60558-333-4. Disponível em: <<http://doi.acm.org/10.1145/1456625.1456629>>. Citado na página 21.
- ZHOU, Jie; CAO, Jiannong; ZHANG, Jun; ZHANG, Chisheng; YU, Yao. Analysis and countermeasure for wormhole attacks in wireless mesh networks on a real testbed. In: IEEE. *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*. [S.l.], 2012. p. 59–66. Citado na página 23.