UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO ENGENHARIA ELÉTRICA E ELETRÔNICA
CURSO ENGENHARIA ELETRÔNICA

Sidinei Lindomar da Rocha Junior

# **SEGURANÇA EM SISTEMAS IoT:** vulnerabilidades e mecanismos de prevenção

Florianópolis
2022

Sidinei Lindomar da Rocha Junior

# **SEGURANÇA EM SISTEMAS IoT:** vulnerabilidades e mecanismos de prevenção

Trabalho de Conclusão do Curso de Graduação em Engenharia Eletrônica do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Bacharel em Engenharia Eletrônica.

Orientador: Prof. Richard Demo Souza, Dr.

Florianópolis

2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Rocha Júnior, Sidinei Lindomar da
Segurança em sistemas de Internet of Things : vulnerabilidades e mecanismos de prevenção / Sidinei Lindomar da Rocha Júnior ; orientador, Richard Demo Souza, 2022.
36 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia Eletrônica, Florianópolis, 2022.

Inclui referências.

1. Engenharia Eletrônica. 2. Internet of things. 3. Segurança Digital. 4. Vulnerabilidades em IoT. I. Souza, Richard Demo . II. Universidade Federal de Santa Catarina. Graduação em Engenharia Eletrônica. III. Título.

Sidinei Lindomar da Rocha Junior

**SEGURANÇA EM SISTEMAS IoT:** vulnerabilidades e mecanismos de prevenção

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de "Bacharel" e aprovado em sua forma final pelo Curso de Engenharia Eletrônica

Florianópolis, 21 de março de 2022.


Documento assinado digitalmente
Fernando Rangel de Sousa
Data: 29/03/2022 10:32:13-0300
CPF: 884.649.114-91
Verifique as assinaturas em https://v.ufsc.br

Prof. Fernando Rangel de Sousa, Dr.
Coordenador do Curso

**Banca Examinadora:**


Documento assinado digitalmente
Richard Demo Souza
Data: 25/03/2022 13:50:54-0300
CPF: 004.267.379-89
Verifique as assinaturas em https://v.ufsc.br

Prof. Richard Demo Souza, Dr.
Orientador
Universidade Federal de Santa Catarina


Documento assinado digitalmente
BARTOLOMEU FERREIRA UCHOA FILHO
Data: 29/03/2022 15:33:22-0300
CPF: 476.362.114-91
Verifique as assinaturas em https://v.ufsc.br

Prof.(a) Bartolomeu Ferreira Uchôa-Filho, Dr.(a)
Avaliador
Instituição Universidade Federal de Santa Catarina


Documento assinado digitalmente
Victoria Dala Pegorara Souto
Data: 30/03/2022 11:18:15-0300
CPF: 024.018.410-67
Verifique as assinaturas em https://v.ufsc.br

Prof.(a) Victoria Dala Pegorara Souto, Dr.(a)
Avaliadora
Instituição Universidade Católica de Pelotas

Este trabalho é dedicado aos meus pais.

## AGRADECIMENTOS

Agradeço aos meus pais por todo ensinamento que me passaram durante a vida, pelo apoio em todos os momentos e pelos momentos que passamos juntos.

Um agradecimento especial para minha namorada e companheira Jéssica Vilvert Klöppel, que me ensinou muito sobre o mundo, me ajudou em momentos difíceis e me deu suporte para que possa continuar na busca do meu caminho.

Agradeço aos meus amigos de longa data por estarem ao meu lado e me apoiarem nas minhas decisões.

Agradeço aos amigos que a Universidade Federal de Santa Catarina trouxe, pelos momentos que passamos juntos e pela parceria.

Agradeço ao meu orientador pela paciência, pela ajuda e pelo auxílio para alcançar mais uma etapa importante do curso de Engenharia Eletrônica.

## RESUMO

O presente trabalho explora as vulnerabilidades mais recorrentes na literatura pesquisada sobre *Internet of Things* (IoT), bem como os mecanismos de prevenção disponíveis para cada uma delas. Apresenta ainda algumas perspectivas futuras com possíveis problemas e soluções relacionadas ao uso de sistemas IoT. Percebe-se a importância de estudos que considerem as necessidades de segurança destes sistemas, considerando que as vulnerabilidades podem expor dados sensíveis de usuários dos dispositivos conectados.



**Palavras-chave:** *Internet of Things*. Segurança Digital. Vulnerabilidades em IoT.

## ABSTRACT

The present work explores the most recurrent vulnerabilities in the literature researched on the Internet of Things (IoT), as well as the prevention mechanisms available for each of them. It also presents some future perspectives with potential problems and solutions related to the use of IoT systems. It is noticed the importance of studies that consider the security needs of these systems, considering that vulnerabilities can expose sensitive data of users of connected devices.



**Keywords:** Internet of Things. Digital Security. IoT vulnerability.

## LISTA DE ABREVIATURAS E SIGLAS

ABE - Attribute Based Encryption

DG INFSO - Information Society and Media Directorate-General of the European Commission

DG Connect - Directorate General Communication Networks, Content and Technology

DDoS – Distributed Denial of Service

DoS – Denial of Service

E2E – End to End

IEEE - Institute of Electrical and Electronics Engineers

IETF - Internet Engineering Task Force

ILP - Independent Link Padding

IoT – Internet of Things

JTAG – Joint Test Action Group

M2M – Machine to Machine

MQTT - Message Queue Telemetry Transport

SMQTT - Secure Message Queue Telemetry Transport

SNEP - Secure Network Encryption Protocol

SNR – Signal-to-Noise Ratio

SVM - Suport Vector Machine

TLS - Transport Layer Security

VPN -Virtual Private Network

WSN – Wireless Sensor Network

**SUMÁRIO**

# 1 INTRODUÇÃO

*Internet of Things* (IoT) é um sistema que faz medições de parâmetros do mundo real, como temperatura, luminosidade, batimento cardíacos etc, através de sensores ligados a um sistema embarcado que se conecta a internet (YANG *et al*., 2017). Maple (2017) traz algumas definições e interpretações sobre o tema, incluindo que, em 2014, pesquisadores especialistas do *Institute of Electrical and Electronics Engineers* (IEEE) o descreveram como uma rede de itens que contém sensores em um sistema embarcado, que é conectada à internet. Maple, 2017 apresenta ainda que, para a *Internet Engineering Task Force* (IETF), dispositivos de IoT podem ser variados objetos, como computadores, sensores, atuadores, refrigeradores, veículos etc. Ademais, *Information Society and Media Directorate-General of the European Commission* (DG INFSO[1]) afirma que '*thing*' é um objeto não identificado com precisão.

Atualmente, a maior parte do que se considera como sistemas IoT são dispositivos autônomos e sistemas isolados, como *smart watches, wearables,* telefones, controladores de luz e temperaturas de casas, etc (BUTUN *et al.,* 2020).

A ideia de conectar coisas à internet vai além do uso do termo *Internet of Things* (MAPLE, 2017). Maple, 2017 cita exemplos de equipamentos que foram conectados à internet nos anos 1980 e 1990, mas que não se enquadram nesse conceito, sendo o termo IoT empregado pela primeira vez em 1999, por Kevin Ashton. Butun *et al.* (2020) afimam que IoT teve uma evolução rápida em um campo que envolve a interconexão e a interação de objetos inteligentes, que podem ser dispositivos com sensores, capacidade de processamento e um meio de comunicação para fornecimento de serviços e aplicativos automatizados.

Hassija *et al*. (2019) indicam que o crescimento de dispositivos físicos conectados a internet está aumento rapidamente e falam que a estimativa para 2020 eram 8,4 bilhões de dispositivos conectados ao redor do mundo e que esse número alcançaria 20,4 bilhões em 2022.

[1] Em 2012 houve uma mudança de nomenclatura para *Directorate General Communication Networks, Content and Technology* (DG Connect)

Sistemas IoT são desenvolvidos para aumentar a qualidade da vida moderna. Uma das aplicações é na busca por soluções para auxiliar pessoas idosas e pessoas com deficiência a realizar tarefas de rotina, aumentando sua autonomia e confiança. Embora seu uso apresente mais benefícios do que riscos é preciso abordar o tema de segurança com cuidado (NESHENKO *et al.*, 2019).

O fato de estar presente em aplicações cotidianas faz com que os dispositivos de IoT estejam próximos às pessoas. Neshenko *et al.* (2019) indicam que a negligência na segurança nestes sistemas pode acarretar em exposição de dados sensíveis e informações confidenciais, como o vídeo de um monitor de bebê conectado à internet, e-mails, senhas, etc.

Dispositivos IoT vulneráveis representam um solo fértil para ameaças cibernéticas (FRUSTACI *et al.*, 2017). Um exemplo é o caso do *malware* Mirai que infectou dispositivos conectados à internet em 2016, a partir dos quais gerou requisições a servidores de internet, provocando ataques de negação de serviço conhecidos por *Denial of Service* (DoS) e *Distributed Denial of Service* (DDoS), que são ataques que utilizam um grande número de dispositivos infectados para fazer requisições a um servidor (FRUSTACI *et al.*, 2017; NESHENKO *et al.*, 2019).

Outro conceito importante para IoT é o de privacidade e proteção dos dados pessoais coletados, pois é preciso fornecer um controle do fluxo automático dos dados ao usuário final (FRUSTACI *et al.*, 2017). Para isso existem dificuldades técnicas que incluem limite de armazenamento, processamento computacional e custo energético que desafiam o atendimento de padrões de segurança nos dispositivos (NESHENKO *et al.*, 2019).

A importância da segurança nos sistemas IoT é evidenciada com a necessidade da proteção de dados coletados, para a qual deve-se ter confiabilidade, integridade e validade desses dados para que possam ser utilizados em diversas aplicações e estudos. Também é preciso garantir a funcionalidade do sistema como um todo, desde a execução até a entrega correta do produto final, isto é, manter os dispositivos funcionando, a comunicação bem estabelecida e protegida para não haver perdas de pacotes e dispositivos infectados. Para contribuir na mitigação dos danos causados pela falta de segurança é necessário entender o funcionamento das

vulnerabilidades presentes nos sistemas e implementar mecanismos de segurança para sua prevenção.

A fim de agregar informações sobre os desafios na segurança de sistemas de IoT, este trabalho objetiva explorar a temática, apresentando as vulnerabilidade mais recorrentes na literatura, seus mecanismos de prevenção e as perspectivas futuras.

# 2 VULNERABILIDADES NAS CAMADAS DE SISTEMAS IOT

Os sistemas IoT podem ser classificados em camadas de acordo com as suas funções. Frustaci *et al*. (2017) apresentam uma divisão nas seguintes camadas: 1) Percepção; 2) Transporte; e 3) Aplicação. Na mesma linha, de acordo com Neshenko *et al.* (2019), a arquitetura de sistemas IoT pode ser distribuída em três camadas denominadas dispositivos, *network* e aplicação. Em cada camada encontra-se desafios na segurança e proteção dos dados adquiridos, transportados e mostrados aos usuários.

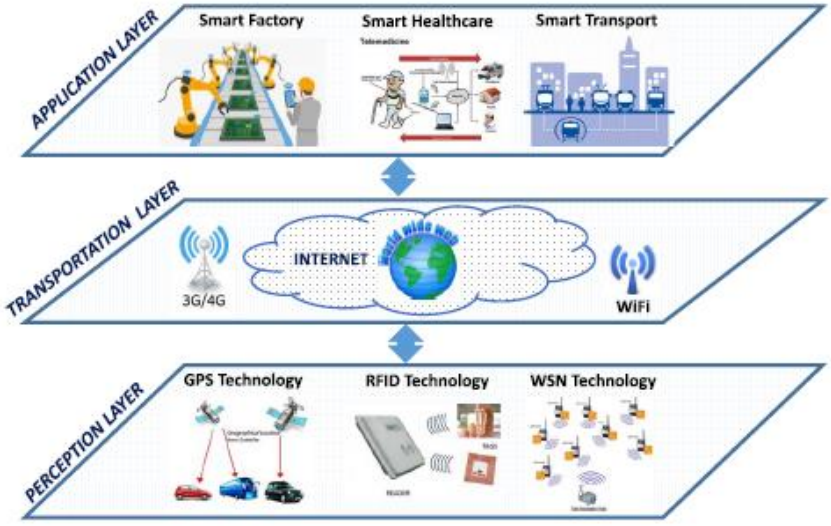
Existem classificações dos sistemas IoT que incluem mais camadas conforme apresentado por Butun, Österberg, Song, *et al.* (2020) que as classificam em: 1) Física; 2) MAC; 3) Network; 4) Transporte; 5) Aplicação. Já Hassija *et al.* (2019) subdividem o sistema de IoT em quatro camadas: 1) Sensoriamento; 2) Network; 3) Middleware; 4) Aplicação. Bem como Yang *et al.* (2017) que também as identificam como: 1) Percepção; 2) Network; 3) Transporte; 4) Aplicação.

Percebeu-se que as divisões em mais de três camadas são contempladas quando agrupadas nas categorias apontadas por Frustaci *et al*. (2017) e Neshenko *et al.* (2019). Por isso, optou-se por subdividir o sistema IoT em três camadas, cujas características e vulnerabilidades serão abordadas neste capítulo.

## **2.1** CAMADAS DE UM SISTEMA IOT

Como tem-se divergência de nomenclatura para denominar as camadas dos sistemas IoT, adotou-se a identificação utilizada por Frustaci *et al.* (2017), que consiste em: 1) Percepção; 2) Transporte; e 3) Aplicação (**Figura 1**).

**Figura 1**: Camadas do Sistemas IoT.




Fonte: Frustaci *et al.* (2017)

### 2.1.1 Percepção

A camada de percepção é responsável pela aquisição dos dados, onde está presente o *hardware* do sistema que executa as medições através do sensoriamento. Esta camada trata do sensoriamento físico do sistema que faz a aquisição dos dados e transmissão para um ponto central, ou *gateway*, geralmente utilizando radiofrequência, através de uma rede de sensores sem fio (WSN) (FRUSTACI *et al.*, 2017). Note que ela corresponde à camada de dispositivo apresentada por *Neshenko et al. (2019),* que também é responsável pelo sensoriamento, capturando dados ciber-físicos do meio.

Hassija *et al.* (2019) classificam essa camada como sensoriamento e indicam que sua principal função é lidar com sensores e atuadores, sendo que os sensores detectam os fenômenos físicos e os atuadores realizam determinada ação de acordo com os dados lidos. Yang *et al.* (2017) seguem a mesma linha, afirmando que a camada de percepção é responsável pela coleta de dados. Este autor propôs a separação da camada de transporte na rede de sensores sem fio da camada de percepção denominando-a como *Network Layer*.

### 2.1.2 Transporte

Após a aquisição dos dados e envio ao *gateway* pela camada de percepção, a camada de transporte tem por finalidade fazer a transmissão dos dados coletados para sistemas de comunicação que utilizam redes de acesso como 3G, WiFi ou a Internet em geral (FRUSTACI *et al.*, 2017; YANG *et al.*, 2017).

Neshenko *et al.* (2019) indicam que a camada de *network* trata dos protocolos de comunicação nos transportes dos dados entre a camada de dispositivos e a de aplicação. Hassija *et al.* (2019) afirmam que a transmissão da informação recebida pela camada de sensoriamento para uma unidade de processamento é a principal função desta camada, denominada por eles como *network*.

### 2.1.3 Aplicação

A camada de aplicação mostra, ao cliente final, os resultados obtidos dos dados coletados pela camada de percepção e transportados pela camada de transporte. Nesta etapa são mostrados dados como temperatura, umidade do ar, entre outros, de acordo com as definições do sistema. Esta camada faz a interface do sistema com o usuário final, provendo serviços para que possa tomar decisões (FRUSTACI *et al.*, 2017; NESHENKO *et al.,* 2019).

Hassija *et al.* (2019) dividem a camada de aplicação em duas, uma de *middleware,* que fornece recursos computacionais e armazenamento e a outra, denominada aplicação, que lida diretamente com a entrega dos dados ao cliente final.

Além de mostrar os dados brutos, nessa camada podem ser feitos processamentos que, com auxílio de algoritmos provenientes da ciência de dados e aprendizagem de máquina, permitem gerar um relatório de conclusão para o usuário final.

## 2.2 VULNERABILIDADES

Os estudos sobre os sistemas IoT apontam para diversas vulnerabilidades a ataques em cada camada, que ameaçam a sua segurança (NESHENKO *et al.*,

2019). Existem também vulnerabilidades que acontecem em mais de uma camada. Apresenta-se a seguir algumas vulnerabilidades distribuídas em cada camada, mostrando uma visão geral dos riscos que apresentam. As vulnerabilidades foram selecionadas de acordo com a recorrência de citação nos artigos levantados (**Tabela 1**) e aparição na mídia, como é o caso de ataques DDoS com Mirai *botnet,* (WOOLF, 2016).

**Tabela 1:** Vulnerabilidades recorrentes nos artigos levantados.

| Percepção | Transporte | Aplicação |
|---|---|---|
| Danos físicos | Sinkhole | Vazamento de dados |
| Jamming-DoS | Wormhole | Negação de serviço |
| Captura de nó | Espionagem | |
| Autenticação | Man in the middle | |
| Transmissão de dados | MQTT[2] exploit | |
| Privação de Stand by | | |
| Redução de eficiência energética | | |

Fonte: Autor

### 2.2.1 Camada de Percepção

Na camada de percepção encontram-se os componentes físicos do *hardware* de aquisição de dados, assim como a estruturação dos nós de comunicação dos dispositivos. Nesta camada pode-se observar as seguintes vulnerabilidades:

#### ***2.2.1.1*** *Ataques físicos*

A parte física do sistema pode sofrer ataques, pois utiliza *hardwares,* como sensores e atuadores. Por exemplo, um dos nós dos sistemas pode sofrer uma destruição física, tornando-o inoperante (BUTUN; ÖSTERBERG; SONG *et al.*, 2020).

Também pode acontecer um sequestro do dispositivo físico por pessoa mal intencionada adulterando o circuito eletrônico projetado. Sobre a posse do dispositivo o atacante poderá: 1) Ganhar acesso a dados sensíveis do sistema

[2] *Message Queue Telemetry Transport* (MQTT).

projetado; 2) Obter acesso a chaves criptográficas compartilhadas pelos nós do sistema para autenticação; 3) Inserir dados falsos para manipulação do resultado final do sistema a partir das chaves criptográficas. (FRUSTACI *et al.*, 2017; BUTUN; ÖSTERBERG; SONG *et al.*, 2020).

#### ***2.2.1.2*** *Problemas de autenticação*

O *hardware* dos sistemas IoT, na camada de percepção, apresenta restrições em alimentação de energia e poder computacional. A partir dessas restrições tem-se dificuldade na implementação de algoritmos de autenticação complexos gerando uma vulnerabilidade que pode ser explorada. Após a invasão do dispositivo o atacante poderá violar a integridade dos dados coletados ou anexar nós com dispositivos maliciosos na rede (NESHENKO *et al.*, 2019).

Com a invasão abre-se caminho para o monitoramento dos dados e recursos do sistema por pessoas não autorizadas. Os ataques podem ser feitos através de força bruta até que se consiga a autorização e o controle do *hardware* (NESHENKO *et al.*, 2019).

Devido à limitação de poder computacional abre-se a porta para que os dispositivos na camada de percepção sejam invadidos transmitindo dados alterados para a próxima camada ou disponibilizando os dados adquiridos para entes não autorizados. Assim, quebra-se o conceito de integridade e confiabilidade da segurança da informação.

#### ***2.2.1.3*** *Transmissão de dados*

A vulnerabilidade na transmissão de dados está diretamente interligada com a comunicação entre os nós e *o gateway.* Nessa comunicação podem acontecer ataques como *Jamming DoS*, colisão de dados e privação de *stand-by*.

Um dispositivo, que não faz parte do sistema, pode enviar sinais de rádio, na mesma região dos nós. Ele utiliza a mesma frequência, reduzindo a *Signal-to-Noise ratio* (SNR) do sinal recebido, o que gera uma degradação dos dados transmitidos, causando o efeito denominado *Jamming DoS* (BUTUN; ÖSTERBERG; SONG *et al.*,

2020). O resultado da diminuição da SNR interfere na comunicação entre os nós e o *gateway,* ocasionando erros de transmissão. Por mais que *Jamming DoS* tenha sido tratado pelos autores como um ataque físico, entende-se que ele tem maior impacto na transmissão de dados devido a degradação da SNR.

Como trata-se de uma vulnerabilidade, um atacante pode transmitir um sinal pelo mesmo canal de algum nó conectado ao sistema, dessa forma, se a transmissão maliciosa e a original colidirem, o receptor da mensagem pode encontrar problemas na identificação e interpretação do pacote recebido (BUTUN; ÖSTERBERG; SONG *et al.*, 2020).

Os sistemas IoT consomem mais energia quando enviam dados dos nós para o *gateway.* Enquanto não ocorre a transmissão, os dispositivos podem entrar em modo *stand-by,* que é uma forma de aumentar a eficiência energética. O ataque de privação nessa situação*,* chamado de *Denial of Sleep,* pode ser executado por ataques de colisão ou trocas de informação contínua fazendo com que o dispositivo não entre no modo *stand-by*, reduzindo a eficiência energética e seu tempo de vida (BUTUN; ÖSTERBERG; SONG *et al.*, 2020). Na categoria de ataque, *Denial of Sleep*, os adversários buscam drenar a bateria do dispositivo para inviabilizar o nó, o que pode ser executado através de códigos maliciosos contendo *loops* infinitos ou por um consumo de energia artificial (HASSIJA *et al.*, 2019).

### 2.2.2 Camada de Transporte

Na camada de transporte faz-se a transmissão de dados adquiridos na camada de percepção até a aplicação. Nesta camada pode-se observar, por exemplo, as seguintes vulnerabilidades:

#### ***2.2.2.1*** *Ataques de rotas*

Durante o transporte dos dados adquiridos pela camada de percepção até a chegada à camada de aplicação os sistemas podem sofrer ataques nas rotas. Hassija *et al.* (2019) afirmam que nestes ataques nós maliciosos dentro da aplicação IoT tentam redirecionar o caminho durante a transmissão de dados. Os autores

mencionam ataques como *sinkhole* e *wormhole* que buscam fazer os redirecionamentos de rotas através dos nós.

No ataque *Sinkhole* um nó malicioso gera uma rota artificial indicando que é o caminho mais eficiente para envio de pacotes de dados até o destino. A partir disso os nós vizinhos passam a enviar os pacotes através do nó malicioso, que não destrói os pacotes, mas recebe todos os dados. Já no *wormhole* tem-se um túnel entre dois nós para transmissão de pacotes de maneira mais rápida. Um nó malicioso pode espionar e transmitir dados para outro nó malicioso presente na rede através de um canal fora de banda. Os pacotes que são enviados através da rota normal chegam ao destino atrasados em relação aos que utilizaram o *wormhole* e podem ser descartados. (BUTUN; ÖSTERBERG; SONG *et al.*, 2020).

#### ***2.2.2.2*** *Ataques de acesso*

Um atacante pode obter acesso aos nós do sistema de IoT e após estar conectado, pode permanecer indetectável por não interagir com o sistema. Este tipo de ataque não está relacionado à destruição dos pacotes de dados que são enviados entre os nós e não busca causar danos ao sistema, seu principal objetivo é ter acesso aos dados por parte de pessoas não autorizadas (HASSIJA *et al*., 2019).

Butun, Österberg, Song *et al.*, (2020) classificam ataques que não estão relacionados a danos no sistema e destruição de pacotes como ataques passivos. Espionagem *(eavesdropping)* é um ataque difícil de ser detectado, pois o atacante está ouvindo a comunicação, roubando os dados e obtendo informações sobre o sistema como um todo, como localização de nós, *gateways*, centro de distribuição de chaves de acesso, etc.

#### ***2.2.2.3*** *MQTT e Man in the middle*

MQTT é uma especificação de protocolo de transporte para transmissão de mensagens. Ele é baseado em *publish-and-subscribe,* no qual os pacotes de publicações são enviados do servidor ao cliente que tem assinatura cadastrada (OASIS *et al*., 2019)*,* fazendo a conectividade de dispositivos que utilizam baixo

recurso energético. Este protocolo não utiliza uma camada de segurança como padrão, permitindo ao desenvolvedor da aplicação a escolha de tecnologias, de autenticação e de autorização, visto que o contexto da aplicação estabelece os padrões de segurança que devem ser adotados pelos desenvolvedores do sistema (OASIS, 2019; BUTUN; ÖSTERBERG; SONG *et al.*, 2020).

Tratando-se do protocolo MQTT, Oasis (2019) lista uma série de ameaças que os desenvolvedores de sistemas IoT devem considerar, como por exemplo: 1) Dispositivos podem estar comprometidos (sob o comando de atacante externo); 2) Dados de clientes e servidores podem estar acessíveis; 3) Ataques de negação de serviços; 4) Comunicações interceptadas, alteradas e reencaminhadas.

O ataque denominado *man in the middle* – quando um atacante se posiciona entre a comunicação de um servidor e o usuário final - é caracterizado quando um usuário não autorizado consegue o controle do *broker* MQTT – gerenciador responsável por autorizar o acesso de clientes inscritos às mensagens transmitidas (THE HIVEMQ TEAM, 2019) - assumindo o comando das comunicações do sistema (HASSIJA *et al.*, 2019).

### 2.2.3 Camada de Aplicação

Na camada de aplicação encontram-se os dados adquiridos e transportados pelo sistema de IoT. Nessa camada o usuário final tem acesso para visualizar, aplicar estudos de ciência de dados e gerar relatórios conforme for necessário. Nela pode-se observar as seguintes vulnerabilidades:

#### ***2.2.3.1*** *Vazamento de dados*

Os sistemas IoT trabalham diretamente com dados desde sua aquisição e transmissão até seu consumo pela camada de aplicação. Como as aplicações de IoT trabalham com dados privados e críticos, os vazamentos de dados geram incertezas ao usuário sobre sua disponibilização (HASSIJA *et al.*, 2019).

Um atacante pode obter acesso a informações críticas, como e-mails, senhas, dados do usuário em geral, a partir do conhecimento de vulnerabilidades do serviço disponível (FRUSTACI *et al.*, 2017).

Como em sistemas IoT tem-se intenso movimento de dados na aquisição na camada de percepção e no transporte pela rede até chegar na aplicação para disponibilização ao usuário final, os dados em trânsitos tornam-se mais vulneráveis a vazamentos.

#### ***2.2.3.2*** *Negação de serviço (DoS)*

DoS trata-se de um ataque para degradar a disponibilidade de serviços de rede. Pode ser explicado como uma situação que consome recursos simultâneos e diminui a capacidade de comunicação da rede, podendo gerar funcionamento incorreto ou resposta fora do tempo hábil. Quando tem-se muitas requisições a um servidor, provenientes de pacotes de informações dos nós, pode-se acarretar um consumo de recursos de resposta, o que impede a entrega de serviços ao usuário final requerente (BUTUN; ÖSTERBERG; SONG *et al.*, 2020).

Ataques de negação de serviços necessitam de um número alto de requisições em um servidor para então degradar o serviço entregue ao usuário final. Pode-se utilizar *Botnets,* como por exemplo o *malware* Mirai, que a partir de um bot faz um escâner para identificar dispositivos da rede do sistema de IoT com usuários e senhas padrões e recorre a ataque de força bruta para invasão (KOLIAS *et al.*, 2017).

A partir dos dispositivos infectados com o *malware*, um *botmaster* consegue gerar um comando para que a rede maliciosa dispare requisições ao servidor impedindo a resposta a requisições feitas por usuários reais. Como afirmam Hassija *et al.* (2019), esses ataques com *botnets* são chamados de interrupção ilegal ou DDoS. Existem registros, como o *malware* Mirai e *botnets* baseados nele, de casos de ataques a aplicações IoT (FRUSTACI *et al.,* 2017; NESHENKO *et al*., 2019). Quando acontecem as requisições dos dispositivos infectados o usuário legítimo tem sua conexão interditada, pois os dispositivos mantém a rede e o servidor ocupados.

# 3 PREVENÇÃO DA EXPLORAÇÃO DAS VULNERABILIDADES

Neste capítulo serão abordadas técnicas e soluções utilizadas para prevenção da exploração das vulnerabilidades em sistemas IoT descritas anteriormente.

## 3.1 FORMAS DE PREVENÇÃO EM SISTEMAS IOT

Frustaci *et al*. (2017) trazem que é fundamental, para segurança em IoT, que sejam adotados padrões de comunicação seguros de forma que os dados em trânsito sejam confidenciais, desenvolver sistemas mais seguros contra ataques cibernéticos com utilização de criptografia e mecanismos de autenticação para gerenciamento de dispositivos conectados a rede.

### 3.1.1 Camada de percepção

Na camada de percepção encontram-se os dispositivos físicos e sensores que fazem parte da aquisição de dados e que sofrem ataques. A partir dos tipos de ataques apresenta-se as técnicas de proteção correspondentes:

#### *3.1.1.1 Ataques físicos*

Como a camada de percepção é composta por *hardware* e esse pode ser destruído, a melhor maneira para proteger os nós de destruição é através de uma camuflagem e ocultação do *hardware.* Em caso de ataque, para solucionar o mal funcionamento deve-se utilizar uma distribuição massiva de nós para que quando ocorra de um dispositivo não funcionar outros possam cobrir a rede (BUTUN; ÖSTERBERG; SONG *et al.*, 2020).

Quanto à prevenção de sequestro do dispositivo físico do sistema de IoT, uma solução é a utilização de *hardware* anti-sequestro, que apaga a memória e qualquer armazenamento de dados a partir de tentativas de violação, para não ocorrer o vazamento. Outra maneira de prevenir é desabilitar a interface *Joint Test*

*Action Group* (JTAG) dos sensores presentes no sistema (FRUSTACI *et al.*, 2017; BUTUN; ÖSTERBERG; SONG *et al.*, 2020).

As soluções para a prevenção desses ataques físicos vem acompanhadas do custo de implementação adicional principalmente na camada de hardware e na distribuição e uso de mais dispositivo de nós.

#### ***3.1.1.2*** *Autenticação*

Devido ao baixo poder computacional dos dispositivos de IoT, na camada de percepção, utiliza-se técnicas para autenticação de nós na rede para validação de pertencimento ao sistema.

Hafeez *et al.* (2016 *apud* Neshenko, 2019) propuseram uma *SecureBox*, uma plataforma para proteção da rede IoT. A ideia da *SecureBox* é verificar se a política de segurança é satisfeita a partir do *request* do nó, em caso negativo aconteceria o isolamento do dispositivo requerente e um aviso ao usuário para verificação. No entanto, é uma solução ainda teórica, necessitando de uma experimentação empírica completa.

Porambage *et al.* (2014) trazem um protocolo de autenticação e distribuição de chave visando dispositivos com alta restrição de recursos. O protocolo proposto é denominado *PauthKey* e consiste em duas fases: 1) Fase de registro, na qual os sensores vão obter as credenciais de segurança do *cluster head*, pré-requisito para autenticação; 2) Fase de autenticação, que estabelece a autenticação do nó e a chave para comunicação mútua. Esse protocolo permite comunicação entre nós de mesmo *cluster,* de *clusters* diferentes e comunicação direta do usuário final com o nó.

#### ***3.1.1.3*** *Transmissão de dados*

Como a transmissão de dados está interligada com a comunicação entre os nós e o *gateway,* podem ocorrer ataques de colisão de dados e na privação de *stand-by.* A prevenção para regiões onde ocorrem *Jamming DoS* seria a implementação de protocolos que mapeiam as rotas de maior ocorrência e possam

desviar as rotas de envio de pacotes de dados remediando o ataque na sobreposição de ondas de rádio (BUTUN; ÖSTERBERG; SONG *et al.*, 2020).

Como prevenção a ataques de colisão, a implementação de uma taxa de solicitação (*request rate*) limitada para cada nó do sistema pode descartar requisições extras provenientes de nós atacantes. Outra solução é utilizar a técnica de multiplexação por divisão de tempo, na qual cada nó tem um intervalo de tempo restrito para envio do pacote de dados prevenindo um abuso de uso do canal pelo nó atacante (BUTUN; ÖSTERBERG; SONG *et al.*, 2020).

Na prevenção ao ataque de privação do modo *stand-by,* Liu *et al.* (2005 *apud* Butun, Östemberg, Song, 2020) propuseram que cada nó participa da detecção dos nós anormais, analisando e avaliando o tráfego dos pacotes de dados. A detecção de anomalias distribuídas e cooperativas é encontrada a partir da análise de um vetor de características de cada nó realizando uma análise cruzada de recursos.

### 3.1.2 Camada de transporte

A camada de transporte faz a transmissão dos dados coletados pela camada de percepção. A partir dos ataques apresentados anteriormente buscou-se técnicas para mitigação das vulnerabilidades dessa camada.

#### ***3.1.2.1*** *Ataques de rotas*

Como o *sinkhole* atrai os pacotes dos nós vizinhos e funciona como um sumidouro de pacotes, é necessário fazer a detecção da rota feita pelos pacotes. Shafiei *et al*. (2014 *apud* Butun, Östemberg, Song, 2020) mostraram dois caminhos para detecção de um *sinkhole*, com o qual os nós vizinhos tendem a gastar mais energia enviando pacotes ao nó suspeito e então tem-se uma região, da rede de sensores, com um buraco de energia. A primeira ideia é usar geoestatística para localizar o *sinkhole* e assim fazer um *bypass* dos sensores por essa rota. O outro método seria fazer um monitoramento para detectar as áreas com menor nível médio de energia utilizando essas rotas na transmissão.

Outra maneira de detecção do *sinkhole* é fazer com que múltiplos nós da rede enviem mensagem ao nó suspeito e fazer uma avaliação da resposta proveniente

confirmando o comprometimento do nó suspeito conforme afirmam Zhang *et al*. (2014 *apud* Butun, Östemberg, Song, 2020).

Como o funcionamento do *wormhole* é diferente do *sinkhole*, conforme mencionado, tem-se a criação de um túnel entre dois nós para transmissão de pacotes mais rapidamente por um canal fora de banda, isso torna a detecção do *wormhole* mais difícil (BUTUN; ÖSTERBERG; SONG *et al.*, 2020). Para sua detecção e defesa, Hu, Perrig, Johnson et al. (2003) propõe o uso de *packet leashes*, que são informações adicionadas no pacote de dados para restringir a distância máxima de transmissão, onde podem ser adicionadas informações temporais ou geográficas. O *leash* geográfico garante que o receptor esteja a uma distância máxima do nó transmissor, já o *leash* temporal garante que o pacote tem um limite de vida útil que restringe a distância máxima de viagem. Ambos métodos impedem o *wormhole,* pois permitem ao receptor avaliar se o pacote viajou além do permitido.

A utilização de chaves baseadas na localização dos nós pode ser usada na prevenção do *wormhole*, uma vez que cada pacote é autenticado pela sua chave, que inclui informação da sua posição. Utilizar essa autenticação excluiria o efeito do ataque por testar a autenticação da posição, fazendo com que nós atacantes sejam negados a conectar na rede pela invalidade de sua chave (ZHANG *et al*., 2006 *apud* Butun, Östemberg, Song, 2020).

#### ***3.1.2.2*** *Ataques de acessos*

Os ataques de acessos tem por finalidade o monitoramento dos dados que estão trafegando. Hassija *et al*. (2019) trazem o modelo de *fog computing*[3] com uma infraestrutura descentralizada para análise de dados podendo ser utilizado para processamento e armazenamento de dados sensíveis de forma rápida e eficiente. Seu principal objetivo é aumentar a segurança, agir na prevenção de roubo de dados, minimizar dados armazenados na nuvem e aumentar a eficiência dos aplicativos de IoT.

[3] O conceito de f*og computing* foi introduzido em 2012 pela Cisco, é uma camada como a *cloud computing*. *Fog computing* busca tratar os dados gerados por dispositivos IoT localmente para melhor gerenciá-los (HASSIJA *et al.*, 2019).

Tratando-se de monitoramento de dados, utilizar *fog nodes*[4], faz a comunicação permanecer entre o usuário final e o nó, diminuindo a rota da informação pela rede, o que, por consequência, diminui as chances de espionagem (HASSIJA *et al.,* 2019, p. 14).

Ataques de espionagem, nos quais o invasor tem acesso aos dados mas não os destrói, são dificilmente detectados. Esse acesso a informação passiva pode ser solucionado utilizando criptografia na transmissão dos dados (BUTUN; ÖSTERBERG; SONG *et al.*, 2020). Os autores apresentam alguns exemplos de algoritmos de criptografia que podem ser utilizados nesse tipo de prevenção, como: TinySec para rede WSN e *Secure Network Encryption Protocol* (SNEP).

#### ***3.1.2.3*** *MQTT e Man in the middle*

Para prevenir os ataques MQTT e Man in the middle pode-se utilizar a *fog computing*, que faz uma camada de segurança entre o sistema IoT e seu usuário final. Ela pode identificar os ataques e ameaças evitando que as atividades fora do padrão atinjam o sistema (HASSIJA *et al.,* 2019).

Singh *et al.* (2015 *apud* Butun, Östemberg, Song, 2020), propuseram um protocolo de segurança chamado *Secure-MQTT* (SMQTT), que se baseia na encriptação *Attribute Based Encryption* (ABE) de curvas elípticas. Ele utiliza o suporte da criptografia de transmissão, e com um ciclo de encriptação a mensagem é disponibilizada a vários destinatários sendo adequada para dispositivos IoT.

Oasis (2019) traz uma lista, não normativa, de preocupações em segurança, cujos tópicos a serem considerados deverão ser selecionados dependendo das características do sistema:

1. **Autenticação de cliente pelo servidor:** Os pacotes de conexões contêm o usuário e a senha, deixando a cargo de quem implementa a solução o modo que utiliza esses dados para autenticação, outros métodos podem ser usados para autenticação como *tokens*, Oauth, etc. Pode-se utilizar o método *Virtual Private*

[4] *Fog node* pode ser qualquer dispositivo que tenha armazenamento computacional e conexão na rede, por exemplo pode ser um roteador, um *switch* ou uma câmera de segurança (HASSIJA *et al.*, 2019).

*Network* (VPN) entre o cliente e o servidor gerando confiabilidade nos dados que estão sendo recebidos.

2. **Autorização do cliente pelo servidor:** Após autenticado no servidor, ele deverá checar se o cliente está autorizado a conectar. A autorização é baseada em dados do cliente, como usuário, IP ou resultados de mecanismos de autenticação.

3. **Integridade de mensagens e controle de pacotes:** As aplicações podem utilizar *hashs* nas mensagens garantindo a integridade dos pacotes enviados. A *Transport Layers Security* (TLS)[5] fornece algoritmos de *hash* para verificação de integridade de pacotes. Outro mecanismo para garantir a integridade seria o uso de VPN.

4. **Privacidade de mensagens e controle de pacotes:** Uma aplicação pode criptografar independentemente suas mensagens, trazendo privacidade aos pacotes enviados. A TLS também pode fornecer uma criptografia de dados enviados pela rede e outro mecanismo útil é novamente o uso de VPN, fazendo a conexão cliente-servidor segura.

A implementação de mecanismos de criptografia de dados auxiliam na prevenção de ataques como Man in the Middle, uma vez que não estão sendo transmitidos dados em textos simples, mas sim criptografados.

### 3.1.3 Camada de aplicação

A camada de aplicação mostra ao usuário final os dados coletados pelo sistema IoT. A partir dos ataques vistos anteriormente buscou-se técnicas para mitigação das vulnerabilidades dessa camada.

#### ***3.1.3.1*** *Vazamento de dados*

A partir do conceito de *fog computing,* Hassija *et al.* (2019) mostram que o armazenamento e gerenciamento de dados tem melhor performance usando *fog nodes.* Os dados ficam mais protegidos quando armazenados dessa forma e não no dispositivo do usuário final. Hassija *et al.* (2019) trazem também que o uso de dados

[5]Os algoritmos podem ser acessados no documento The Transport Layer Security (TLS) Protocol Version 1.3 (RESCORLA, 2018).

encriptados, autenticação do usuário e gerenciamento de privacidade são alguns mecanismos que previnem o vazamento de dados e fazem com que a aplicação seja mais segura.

Como alternativa de mecanismo de prevenção, Premsankar, Di Francesco e Taleb *et al.* (2018 *apud* Hassija 2019) sugerem o uso do *edge computing,* no qual os dados são armazenados e processados nos dispositivos ou na rede local. Esse armazenamento diminui o movimento dos dados prevenindo os vazamentos e roubos.

#### ***3.1.3.2*** *Negação de serviço (DoS)*

Ataques de negação de serviço degradam a funcionalidade do sistema IoT como um todo. Pode-se utilizar mecanismo de *fog computing* e *cloud computing* para mitigar os ataques DoS, pois não existe apenas uma solução para detecção e solução de ataques dessa categoria (HUSSAIN *et al.*, 2020). O aprendizado de máquina tem sido pesquisado para soluções de problemas de segurança em IoT e algoritmos de aprendizado de máquina tem sido utilizados na detecção e segurança em ataques de negação de serviço (HASSIJA *et al.*, 2019).

Pode-se utilizar algoritmos de aprendizado de máquina na detecção e mitigação de ataques DDoS/DoS, pois esses algoritmos levam em conta o comportamento do atacante e não apenas o fluxo de tráfego pela rede. Os métodos devem considerar que falsos positivos podem acontecer mesmo com solicitações verídicas (HUSSAIN *et al.*, 2020).

Utilizando métodos de aprendizado de máquina na detecção de DDoS, Doshi *et al.* (2018 *apud* Hussain 2020) fazem uma comparação entre os métodos *K-nearest neighboors*, *decision trees*, *neural network*, *random forrest* e *Suport Vector Machine* (SVM) alcançando uma taxa de sucesso de 99% na detecção dos ataques. A partir da detecção podem ser adotados métodos para fazer um *bypass* e algoritmos para não atender as requisições dos nós atacantes.

# 4 PERSPECTIVAS PARA SEGURANÇA DE SISTEMAS IOT

Os estudos indicam o desenvolvimento de novos mecanismos de segurança em sistemas IoT, que possibilitam a proteção de transferência de dados e outros recursos da rede, bem como indicam o surgimento de novas vulnerabilidades em todas as camadas do sistema.

Neste capítulo serão abordadas algumas possibilidades para o futuro da segurança em sistemas IoT, considerando outras possíveis vulnerabilidades nas camadas adotadas no presente trabalho e, quais áreas do sistema devem ser focadas para proteção da aplicação e dos dados adquiridos, armazenados e utilizados.

## **4.1** CAMADA DE PERCEPÇÃO

Nesta camada, dentre outras possíveis soluções apresentadas por Butum *et al.* (2020) para as vulnerabilidades já conhecidas, estão: a Distribuição de chaves privadas; Mecanismo de confiança, que deve incluir métricas de confiança, como perda de pacotes, contagem de saltos, alcance de transmissão de rádio, taxa de consumo de energia, latência do link de dados, qualidade do caminho, etc; Agregação de dados; Atualização de *firmware*; e Reconhecimento digital. No entanto, o próprio autor chama a atenção para a dificuldade em como executar algumas soluções como a distribuição das chaves privadas, pois perder-se-ia escalabilidade com a adição de novos nós, bem como desafios na agregação de dados, visto que com esta solução, eles ainda estariam passíveis de violação.

Frustaci *et al.* (2017), também chamam a atenção sobre os problemas em soluções que envolvem segurança do hardware, mecanismo de confiança, protocolos de rotas inseguros, mecanismo de gerenciamento de chaves criptográficas e soluções anti malware.

Hassija *et al.* (2019) atentam sobre a segurança de dados e a privacidade do usuário com o *edge computing,* onde os dados podem ser vazados e utilizados por um atacante para obtenção de informações, como horários de presença em casa de acordo com o uso da eletricidade ou da água. Ele indica que os dados devem ser

totalmente do usuário final para que possa decidir quais deles poderão ser compartilhados.

## **4.2** CAMADA DE TRANSPORTE

Como outra possível solução para a prevenção de problemas nesta camada, Apthorpe (2017 *apud* BUTUM *et al.* 2020) propôs uma metodologia para o tráfego de rede. O *Independent Link Padding* (ILP) funciona no sentido de moldar o tráfego de dados sem violar a taxa pré-determinada, isto é, sem causar perdas das atividades de comunicação. Com isso o usuário final tem sua privacidade garantida contra-ataques de coleta de informações passivas.

## **4.3** CAMADA DE APLICAÇÃO

Para esta camada, Hassija *et al.* (2019) apontam como ponto de atenção que a seleção do algoritmo correto é vital para não diminuição da eficiência e precisão, em função de existirem diversos algoritmos de aprendizado de máquina, visto que o sucesso dos algoritmos de aprendizado máquina depende da escolha de algoritmo e dos dados utilizados.

## **4.4** MULTICAMADAS

Para além dos possíveis problemas e soluções que atingem cada camada, há fatores que podem envolver duas ou mais camadas. Wu *et al.* (2016 *apud* BUTUN, 2020) mostram uma solução que envolve uma estrutura de segurança hierárquica, com um sistema de detecção de ataques com regras simples sendo executado nos nós sensores e detecção de ataques executadas em coletores e na estação base com regras mais complexas.

Hassija *et al.* (2019) apontam que a segurança da *blockchain* depende do método de implementação e do uso do software e do hardware na aplicação. Assim, como os dados de transações feitos em uma *blockchain* são públicos, pode existir a possibilidade de que as informações privadas sejam vazadas. Uma *blockchain* utiliza

algoritmos de segurança que fornecem solução descentralizada, portanto, quando se aumenta o número de mineradores o tamanho de uma *blockchain* aumenta. Isto implica diretamente no custo energético, no *delay* e poder computacional, o que não é adequado para dispositivos limitados de IoT (HASSIJA *et al.*, 2019; BUTUN *et al.*, 2020).

Neshenko et al. (2019) também citam problemas que vão além do que pode ser previsto nos mecanismos de segurança, que incluem o comportamento do usuário no uso do sistema, como ao não atualizar as senhas padrões disponibilizadas, por exemplo. Os autores apontam para a falta de identificação em grande escala de dispositivos explorados e de solução em grande escala para mapear os problemas de segurança em IoT. Mostram a necessidade de obter-se protocolos de segurança padronizados. Por fim, alertam sobre a falta de segurança no desenvolvimento de softwares, uma vez que sistemas IoT utilizam aplicativos de software personalizados, sendo que é necessário uma metodologia para que se possam verificar a segurança do código implantado.

# 5 CONCLUSÃO

Neste trabalho buscou-se apresentar alguns dos problemas de segurança recorrentes na literatura sobre sistemas IoT. Primeiro definiu-se o sistema IoT em três camadas: 1) Percepção; 2) Transporte; 3) Aplicação. A seguir indicou-se as vulnerabilidades mais recorrentes, em cada camada do sistema, os mecanismos de detecção e solução que são usados para mitigar os problemas de segurança e uma direção futura sobre alguns problemas de segurança em IoT.

A camada de percepção, que faz a aquisição dos dados utilizando sensores e atuadores, utiliza *hardware* com baixo poder computacional e limitações energéticas e de armazenamento devido à quantidade de dispositivos que podem estar disponíveis na rede. O fato da utilização de *hardware* exposto a torna mais susceptível a problemas físicos, conforme visto anteriormente. O baixo poder computacional gera dificuldades na implementação de algoritmos de criptografia seguros e que não tenham sua chave descoberta, o baixo poder computacional somado às limitações energéticas, é um obstáculo para implementação de criptografia de ponta a ponta eficiente. Outro ponto importante é que a rede de dispositivos precisa identificar nós problemáticos (comprometidos). Mecanismos de confiança são baseados em chaves criptográficas, então são necessários métodos confiáveis e que tragam escalabilidade, uma vez que a rede de dispositivos poderá aumentar ou diminuir com a inserção ou remoção de nós.

Os dispositivos que compõem a camada de percepção são adquiridos por usuários finais que desejam monitorar sinais vitais, aumentar o nível de segurança de suas casas através de câmeras, otimizar processos cotidianos ou de trabalho, entre outras aplicações. É importante se atentar a alguns requisitos de segurança na utilização dos sistemas, como a troca de senhas padrões, o que dificulta possíveis invasões. Além de manter os dispositivos atualizados, uma vez que as novas versões disponíveis, em geral, servem para correção de problemas de desenvolvimento e incremento de segurança para vulnerabilidades já descobertas. Portanto, é necessário a conscientização do usuário, para que cumpra estas e outras etapas que auxiliam sua própria segurança.

Com o crescente número de dispositivos conectados à internet, pode haver um aumento nos ataques DoS, exigindo a necessidade de estudos para proteção da

rede de sensores e os métodos para avaliação dos *requests* nos servidores que transportam os dados até o cliente final. Conforme visto em 2016 com o *malware* Mirai[6], que infectou dispositivos que utilizavam senhas padrões. Os dispositivos infectados se comportavam como zumbis para fazer ataques de negação de serviços. Portanto, se não houver conscientização dos usuários aliada a mecanismos de defesa, podem ocorrer ataques em maior escala.

Para trabalhos futuros, sugere-se pesquisas sobre algoritmos de criptografia confiáveis que necessitem de baixo poder computacional, mecanismos de gerenciamento de chaves criptográficas para escalabilidade da rede, mitigação de ataques DoS utilizando aprendizado de máquina e possíveis maneiras de conscientização dos usuários de sistemas IoT na importância da questão de segurança.

[6] Mirai writes new chapter in the history of DDOS attacks (ILASCU, 2017).

## REFERÊNCIAS

BUTUN, I.; ÖSTERBERG, P.;SONG, H. Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures. **IEEE Communications Surveys & Tutorials**, v. 22, n. 1, p. 616-644, novembro 2020. DOI: 10.1109/COMST.2019.2953364. Disponível em: https://ieeexplore.ieee.org/document/8897627. Acesso em: 05 dez. 2021.

FRUSTACI, M.; PACE, P.; ALOI, G.; FORTINO, G. Evaluating Critical Security Issues of the IoT World: Present and Future Challenges. **IEEE Internet of Things Journal**, v. 5, n. 4, p. 2483-2495, agosto 2018. DOI: 10.1109/JIOT.2017.2767291. Disponível em: https://ieeexplore.ieee.org/document/8086136. Acesso em: 05 dez. 2021.

HASSIJA, V.; CHAMOLA, V.; SAXENA, V.; D. Jain, D.; GOYAL, P.; SIKDAR, B. A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. **IEEE Access**, v. 7, p. 82721-82743, 2019, junho 2019 DOI: 10.1109/ACCESS.2019.2924045. Disponível em: https://ieeexplore.ieee.org/document/8742551. Acesso em: 05 dez. 2021.

HU, Y.; PERRIG, A.; JOHNSON, D. B., Packet leashes: a defense against wormhole attacks in wireless networks. *In*: Annual Joint Conference of the IEEE Computer and Communications Societies, 22, 2003, São Francisco, CA. **IEEE**, v. 3, p. 1976-1986. DOI: 10.1109/INFCOM.2003.1209219. Disponível em: https://ieeexplore.ieee.org/document/1209219. Acesso em: 01 mar. 2022.

HUSSAIN, F.; HUSSAIN, R.; HASSAN S. A.; HOSSAIN E. Machine Learning in IoT Security: Current Solutions and Future Challenges. **IEEE Communications Surveys & Tutorials**, v. 22, n. 3, p. 1686-1721, abril 2020, DOI: 10.1109/COMST.2020.2986444. Disponível em: https://ieeexplore.ieee.org/document/9060970. Acesso em: 02 mar. 2022.

ILASCU, I. Mirai Writes New Chapter in the History of DDoS Attacks. **Bitdefender**, Romenia, 2017. Disponível em: https://www.bitdefender.com/blog/hotforsecurity/mirai-writes-new-chapter-history-ddos-attacks. Acesso em: 10 mar. 2022.

NESHENKO, N.; BOU-HARB, E.; CRICHIGNO, J.; KADDOUM, G.; GHANI, N. Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations. **IEEE Communications Surveys & Tutorials**, Singapore, v. 21, n. 3, p. 2702-2733, abril 2019. DOI: 10.1109/COMST.2019.2910750. Disponível em: https://ieeexplore.ieee.org/document/8688434. Acesso em: 05 dez. 2021.

OASIS. **MQTT Version 5.0**. BENKS, A. *et al* (editores). Wouburn (MA): OASIS Open, 2019. 137p. Disponível em: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf. Acesso em: 19 dez. 2021.

PORAMBAGE P.; SCHMITT C.; KUMAR P.; GURTOV A.; YLIANTTILA M. PAuthKey: a pervasive authentication protocol and key establishment scheme for wireless sensor networks in distributed iot applications. **International Journal of Distributed Sensor Networks**, v. 10, n. 7, 14 p., julho 2014. DOI:10.1155/2014/357430. Disponível em: https://journals.sagepub.com/doi/10.1155/2014/357430. Acesso em: 01 mar. 2022.

RESCORLA E. The Transport Layer Security (TLS) Protocol Version 1.3. **RFC**, Califórnia, v. 8446, 160 p., agosto 2014. DOI:10.17487/RFC8446. Disponível em: https://www.rfc-editor.org/rfc/pdfrfc/rfc8446.txt.pdf. Acesso em: 10 mar. 2022.

THE HIVEMQ TEAM. MQTT Client and Broker and MQTT Server and Connection Establishment Explained - MQTT Essentials: Part 3. *In*: HIVEMQ. **MQTT Essentials.** Bavaria: HiveMQ GmbH, 2019. Disponível em: https://www.hivemq.com/blog/mqtt-essentials-part-3-client-broker-connection-establishment/. Acesso em: 25 mar. 2022.

WOOLF, N. DDoS attack that disrupted internet was largest of its kind in history, experts say. **The Guardian,** San Francisco, 26 out. 2016. Disponível em: https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet. Acesso em: 19 dez. 2021.

YANG, Y.; WU, L.; YIN, G.; LI, L.; ZHAO, H. A Survey on Security and Privacy Issues in Internet-of-Things, **IEEE Internet of Things Journal**, v. 4, n. 5, p. 1250-1258, outubro 2017. DOI: 10.1109/JIOT.2017.2694844. Disponível em: https://ieeexplore.ieee.org/document/7902207. Acesso em: 05 dez. 2021.