Modelagem de Ameaças com STRIDE e DREAD: Uma Análise preliminar aplicada a um sistema IoT

Luis Fernando Alves[1], **Pedro Amalfi**[1], **Santiago Martin**[1], **Claudio Schepke**[1], **Elder Rodrigues**[1], **Maicon Bernardino**[1]

[1]Universidade Federal do Pampa (UNIPAMPA) – Alegrete, RS – Brasil

{luisfads, pedroamalfi, santiagopereira}.aluno@unipampa.edu.br
{claudioschepke, elderrodrigues}@unipampa.edu.br, bernardino@acm.org

Resumo. *A crescente complexidade dos sistemas e o aumento das ameaças cibernéticas exigem práticas robustas de segurança. A modelagem de ameaças é uma abordagem eficaz para identificar e mitigar riscos em sistemas complexos. Este trabalho explora as metodologias STRIDE e DREAD aplicadas à modelagem de ameaças em um sistema IoT, com foco nas etapas de identificação e mitigação de ameaças. A análise preliminar ilustra como cada metodologia pode ser utilizada em diferentes contextos, ressaltando sua aplicabilidade para a segurança de sistemas. Ao longo do estudo, demonstramos como as duas metodologias podem complementar-se para aumentar a eficácia da proteção de sistemas complexos.*

Abstract. *The increasing complexity of systems and the rise of cyber threats require robust security practices. Threat modeling is an effective approach to identifying and mitigating risks in complex systems. This work explores the STRIDE and DREAD methodologies applied to threat modeling in an IoT system, focusing on the threat identification and mitigation steps. The preliminary analysis illustrates how each methodology can be used in different contexts, highlighting its applicability for systems security. Throughout the study, we demonstrated how the two methodologies can complement each other to increase the effectiveness of protecting complex systems.*

1. Introdução

A segurança de software tornou-se uma prioridade devido ao aumento dos ataques cibernéticos e à complexidade crescente dos sistemas. A Modelagem de Ameaças (*Threat Modeling*) é uma abordagem sistemática para identificar e mitigar ameaças ao longo do ciclo de vida do desenvolvimento de software. *Threat Modeling* envolve a identificação e avaliação de ameaças a um sistema, permitindo a definição de medidas de mitigação adequadas [Shostack 2014]. O conceito de *Threat Modeling* surgiu na década de 1990, inicialmente focado em sistemas operacionais e redes, mas evoluiu para abranger aplicações web e móveis. Metodologias como **STRIDE** desempenharam um papel crucial nessa evolução, oferecendo uma categorização de ameaças que inclui *Spoofing*, *Tampering* e *Denial of Service* [Infomach 2024]. Além de STRIDE, outras metodologias ganharam destaque, como **DREAD**, que avalia ameaças com base em critérios como potencial de dano (*Damage Potential*) e reprodutibilidade (*Reproducibility*) [Conviso 2024]. Este estudo visa explorar os conceitos e metodologias de *Threat Modeling*, discutir as abordagens STRIDE e DREAD, para demonstrar sua aplicação prática em um cenário de sistema IoT, com o objetivo de fornecer uma visão abrangente da segurança de software neste contexto.

2. Modelagem de Ameaças

Threat Modeling é uma prática estruturada de identificar e avaliar possíveis ameaças à segurança de um sistema, com o objetivo de determinar as medidas necessárias para mitigar essas ameaças[OWASP 2024]. Definimos ameaça como qualquer circunstância ou evento com potencial de impactar negativamente as operações organizacionais[National Institute of Standards and Technology 2006]. Essa técnica envolve a criação de modelos que representam as possíveis formas de ataque e a análise das vulnerabilidades existentes no sistema. Ela deve ser integrada em várias fases do desenvolvimento de software, desde a concepção até a implementação e manutenção. Os passos gerais para implementar *Threat Modeling* incluem:

1. **Identificação de Ativos**: Determinar quais ativos (dados, componentes, sistemas) precisam ser protegidos;
2. **Diagramação do Sistema**: Criar diagramas que representem os fluxos de dados e as interações entre os componentes do sistema;
3. **Identificação de Ameaças**: Usar metodologias como *STRIDE* para identificar possíveis ameaças a cada componente do sistema;
4. **Classificação de Ameaças**: Avaliar e priorizar as ameaças identificadas com base em critérios como impacto e probabilidade, utilizando metodologias como *DREAD*;
5. **Mitigação de Ameaças**: Definir e implementar medidas de mitigação para as ameaças mais críticas;
6. **Revisão e Atualização**: Revisar e atualizar continuamente o modelo de ameaças conforme o sistema evolui.

Um estudo onde é detalhado cada etapa do processo utilizado neste artigo bem como outras metodologias foi realizado em BITM[1][Alves et al. 2024].

2.1. Camadas e Componentes do Sistema

Para ilustrar as etapas de desenvolvimento e facilitar a compreensão das metodologias STRIDE e DREAD, foi criado um cenário de Monitoramento de Ambiente IoT. Esse cenário envolve a coleta e transmissão de dados de temperatura e umidade de sensores para um servidor local, acessado por um único usuário. Nesse contexto, foram identificados todos os componentes do sistema e suas respectivas funções, descritos a seguir: **Dispositivos**: Os dispositivos são responsáveis por coletar dados ambientais e processá-los antes de transmiti-los para o *gateway*. A Tabela 1 detalha os componentes e suas funcionalidades.

Tabela 1. Componentes dos Dispositivos

ID	Recurso	Componente	Funcionalidade
1	Sensor	DHT-22	Responsável por medir a temperatura e umidade do ar.
2	Microcontrolador	ESP32 LoRa v3 heltec	Processa os dados do sensor e envia as leituras para o gateway através do protocolo LoRa.
3	Software do Microcontrolador	Plataforma Arduino	Firmware responsável pela leitura dos dados do sensor e o envio através dos protocolos LoRa / MQTT para o gateway.

Comunicação: A comunicação no sistema é feita de forma sem fio entre os dispositivos e o *gateway*. A Tabela 2 descreve o protocolo utilizado e sua funcionalidade.

[1]BITM: `https://doi.org/10.5281/zenodo.13821928`

Tabela 2. Componentes de Comunicação

ID	Recurso	Componente	Funcionalidade
1	Protocolo de Comunicação	LoRa	Responsável pela transmissão de dados sem fio entre dispositivo e *gateway*.

Gateway: O *gateway* recebe, processa e armazena os dados enviados pelos dispositivos. Ele também gerencia a comunicação com o sistema central. A Tabela 3 lista os componentes do *gateway* e suas funções.

Tabela 3. Descrição dos Recursos e Componentes do Sistema

ID	Recurso	Componente	Funcionalidade
1	Hardware do Gateway	Raspberry Pi 4/ Robocore LoRaWan	Processamento dos dados obtidos pelo Gateway através da conexão LoRa.
2	Sistema Operacional do Gateway	Raspberry Pi OS	Sistema operacional baseado em Linux responsável por gerenciar o hardware e software.
3	Broker MQTT	Mosquitto	Gerencia os tópicos MQTT do sistema.
4	Ambiente de Execução	Node.js	Atua como interpretador da linguagem JavaScript.
5	Software do Sistema do Gateway	Aplicativo Node.js / Cliente MQTT.js	Responsável pela comunicação entre o broker e o cliente MQTT, persistência dos dados no servidor MySQL e interface web com o cliente e o sistema de autenticação.
6	Servidor Banco de Dados	MySQL	Sistema Gerenciador de Banco de Dados responsável por persistir as leituras enviadas pelo sensor e outros dados da aplicação.

2.2. Dependências Externas

Para o correto funcionamento do sistema, várias dependências externas precisam ser gerenciadas. Essas dependências permanecem sob o controle da organização, porém podem não estar sob o controle direto da equipe de desenvolvimento A Tabela 4 enumera essas dependências e os pontos de atenção associados a cada recurso.

Tabela 4. Dependências Externas

ID	Recurso/Dependências	Pontos de atenção
1	Sensor	Possibilidade de manipulação das leituras
2	Dispositivo Microcontrolador	Controle das portas de comunicação e interfaces
3	Software do Microcontrolador	Risco de *spoofing* na comunicação
4	Protocolo de Comunicação	Vulnerabilidades relacionadas à criptografia, ataques DDoS, *jamming* e interceptação por *Man in the Middle*
5	Hardware do *Gateway*	Segurança nas portas e interfaces de comunicação
6	Sistema Operacional do *Gateway*	Autenticação e controle de acesso ao sistema
7	*Broker MQTT*	Gerenciamento de autenticação, autorização, criptografia e proteção contra injeção de mensagens
8	Ambiente de Execução	Ausência de atualizações e *patches* de segurança
9	Software do Sistema do *Gateway*	Vulnerabilidades na interface web e processos de login
10	Servidor Banco de Dados	Exposição a ataques de *SQL Injection*

2.3. Pontos de Entrada e Saída

Os pontos de entrada e saída do sistema são vulnerabilidades potenciais que precisam ser monitoradas para garantir a segurança do sistema. Os pontos de entrada são interfaces por onde dados ou comandos ingressam no sistema, enquanto os pontos de saída são os locais por onde os dados são liberados. Embora os pontos de saída não sejam diretamente vulneráveis a ataques, eles podem ser explorados para obter informações que facilitem

futuras invasões ou ataques ao sistema. As Tabelas 5 e 6 detalham esses pontos para cada recurso.

Tabela 5. Pontos de Entrada

ID	Recurso	Ponto de Entrada	Nível de Confiança
1	Sensor	Manipulação das Leituras	3, 4, 5, 12
2	Dispositivo Microcontrolador	Portas/Interface de comunicação	3, 4, 5, 12
3	Software do Microcontrolador	*Spoofing* de comunicação	2, 3, 5
4	Protocolo de Comunicação	Criptografia, DDoS, *Jamming e Man in the Middle*	2, 3
5	Hardware do *Gateway*	Portas/Interface de comunicação	3, 4, 5
6	Sistema Operacional do *Gateway*	Autenticação / Acesso ao Sistema	2, 10
7	*Broker MQTT*	Autenticação e autorização / criptografia / injeção de mensagens	2, 6, 9
8	Ambiente de Execução	Falta de atualização e *patches*	1, 2, 7
9	Software do Sistema do *Gateway*	Interface Web / Login	1, 2
10	Servidor Banco de Dados	*SQL Injection*	1, 2, 8, 11

Tabela 6. Pontos de Saída

ID	Recurso	Ponto de Saída
1	Software do Sistema do *Gateway*	Coleta de dados de erros/debug ativo
2	Servidor Banco de Dados	*SQL Injection*

2.4. Ativos e Níveis de Confiança

Os ativos do sistema incluem dados, informações e componentes de hardware e software que precisam ser protegidos. Sendo classificados como físicos, como uma lista de clientes, ou abstratos, como a reputação de uma organização. Os níveis de confiança definem os direitos de acesso concedidos pelo aplicativo a entidades externas, relacionando-os com pontos de entrada e ativos, permitindo especificar os privilégios necessários para acessar cada um. A Tabela 7 lista esses ativos e seus níveis de confiança pode ser visto na Tabela 8.

2.5. Diagrama de Fluxo de Dados

Os Diagramas de Fluxo de Dados (DFDs) oferecem uma representação visual do processamento de dados em uma aplicação, destacando componentes críticos e o fluxo de dados. Eles ajudam a entender o funcionamento da aplicação ao decompor o sistema em subsistemas e mostrar a movimentação dos dados. O sistema é representado por três DFDs. A **Figura 1** mostra uma visão geral do sistema IoT, incluindo o microcontrolador que coleta dados ambientais e os envia via LoRa/MQTT para o *Gateway*. A **Figura 2** detalha o microcontrolador, dividindo-o entre sensores que coletam dados e o hardware/software que transmite os dados para o *Gateway* via LoRa/MQTT. Por fim, a **Figura 3** detalha o *Gateway*, que é composto por três elementos principais: o processo do *Gateway*, que recebe dados via LoRa/MQTT e interage com o banco de dados; o Banco de Dados, responsável por armazenar os dados e se comunicar com o Web Site; e, por fim, o Web Site juntamente com o ambiente Node.JS, que processa as requisições dos usuários, gerencia arquivos estáticos e se comunica com o banco de dados.

Tabela 7. Ativos

ID	Nome	Descrição	Nível de Confiança
1	Dados armazenados no SGBD	Dados armazenados no servidor de banco de dados	2, 8, 11
1.1	Dados dos sensores	Leitura dos Sensores Armazenados no sistema	2, 8
2	Dados e informações do software do sistema	Login dos usuários e dados do sistema	1, 2, 9, 10, 11
2.1	Informações de Login	Credenciais de login e sua senha	1, 2, 9, 10, 11
2.2	Dados dos sensores	Leitura dos Sensores Armazenados no sistema	1, 2
2.3	Código-Fonte do sistema	Código-Fonte do sistema contendo informações críticas da funcionalidade do sistema	2, 7
3	Dados e Informações do Broker	Informações críticas armazenadas no Broker	2, 6, 9
3.1	Autenticação do Broker	Senhas de autenticação do Broker	2, 6, 9
3.2	Configurações do Broker	Arquivo de configurações do Broker - Mosquitto	2, 6, 9
4	Dados e informações do sistema operacional	Informações e Arquivos do sistema operacional	2, 10
4.1	Autenticação do sistema operacional	Informações de login do sistema operacional	2, 10
4.2	Arquivos e Configurações	Arquivos e configurações armazenados no sistema operacional	2, 10
4.3	Controle do sistema operacional	Controle do funcionamento do sistema operacional	2, 10
5	Hardware do Gateway	Ponto de acesso a invasão via hardware	3, 4, 5
5.1	Hardware Hacking	Acesso a portas de entrada de comunicação, módulos de comunicação, modificação do sistema visando Hardware Hacking	3, 4, 5
6	Software do Firmware	Dump do firmware	2, 3, 5
6.1	Dump do firmware	Dump do firmware visando engenharia reversa e/ou modificação	2, 3, 5
7	Hardware do Microcontrolador	Ponto de acesso a invasão via hardware	3, 4, 5, 12
7.1	Hardware *Hacking*	Acesso a portas de entrada de comunicação, módulos de comunicação, modificação do sistema visando Hardware *Hacking*	3, 4, 5, 12
7.2	*Firmware Tampering*	Manipulação do *Firmware* armazenado na memória do microcontrolador	3, 12
7.3	*Spoofing* nas portas de comunicação	Manipulação das portas de comunicação via *spoofing*	3, 12
8	Sensores	Manipulação dos sensores	3, 4, 5, 12
8.1	*Spoofing* dos sensores	Falsificação das leituras	3, 4, 5, 12
8.2	Hardware *Hacking*	Modificação dos sensores e/ou do seu *firmware*	3, 12
9	Comunicações	Sobrecarga e distorção de sinais	2, 3
9.1	Disponibilidade de Comunicações	DDoS e *Jamming* do canal de comunicação LoRa	2, 3
10	Reputação do Sistema	Depreciação da reputação da organização	2, 3, 4, 5
10.1	Vazamento de dados	Roubo e apropriação de informações confidenciais	2, 3, 4, 5
10.2	Comprometer a integridade dos dados	Sequestro e/ou corrompimento dos dados	2, 3, 4, 5
10.3	Danos legais e a reputação do sistema	Vazamento de informações que comprometam a LGPD e a reputação da organização/sistema	2, 3, 4, 5

Tabela 8. Níveis de Confiança

ID	Nome	Descrição
1	Usuário do Sistema IoT	Usuário com credenciais válidas para acessar o sistema IoT
2	Administrador do Sistema de Software	Desenvolvedor com acesso ao sistema de software (SO, SGBD, *Broker*, *firmware*, ambiente de execução e sistema web)
3	Administrador do Sistema de Hardware	Desenvolvedor com acesso ao sistema de hardware (*gateway*, microcontrolador, sensor)
4	Instalador do sistema	Responsável por instalar/testar o sistema
5	Equipe de manutenção e suporte	Responsáveis por executar manutenções e suporte ao usuário
6	Processo do *Broker*	Processo que executa o *Broker MQTT Mosquitto*
7	Processo do Ambiente de Execução	Processo que executa o Ambiente de Execução Node
8	Processo do Banco de Dados	Processo que executa o SGBD *MySQL*
9	Conta do *Broker*	Informações de autenticação do *Mosquitto*
10	Conta do SO	Conta de usuário do Linux
11	Conta do SGBD	Conta de usuário do *MySQL*
12	Sistema Embarcado	*Firmware* executado no microcontrolador

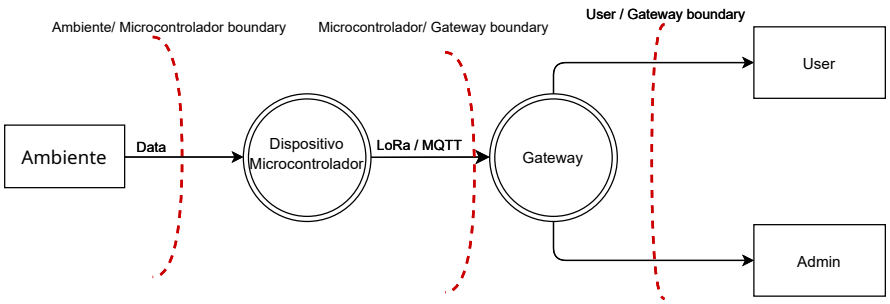

Figura 1. Diagrama de Fluxo de Dados do Sistema IOT, Visão Geral

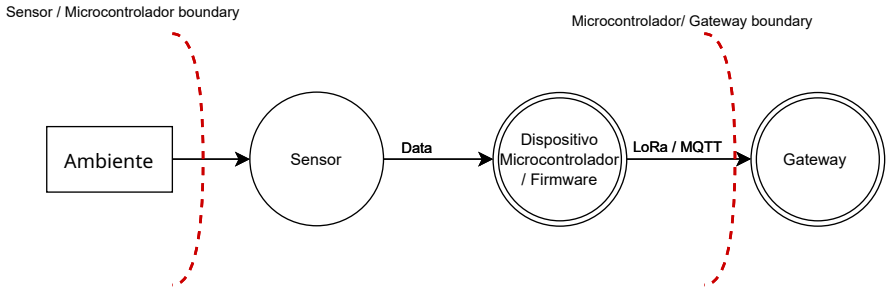

Figura 2. Diagrama de sistema de sensores

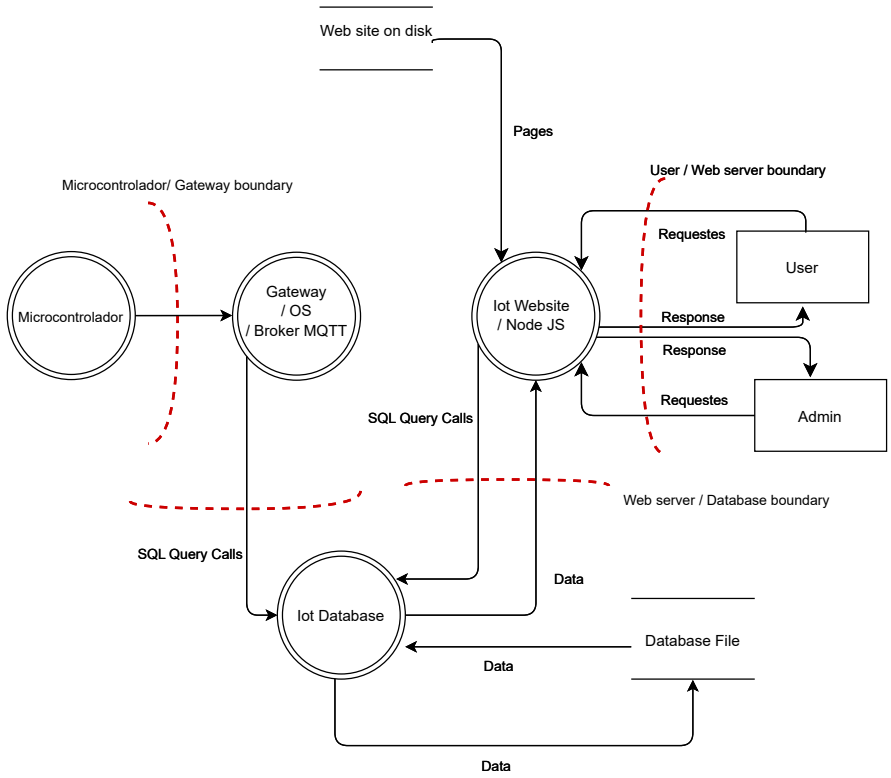

Figura 3. Diagrama de sistema de Gateway

2.6. Identificação de ameaças com STRIDE

Com base no cenário descrito anteriormente (Sessão 2.1). Cada componente foi classificado em sua camada correspondente, e os pontos críticos foram analisados para melhorar a segurança, considerando limitações externas, como requisitos do fabricante. Aplicando a metodologia *STRIDE* (Spoofing (falsificação), Tampering (adulteração), Repudiation (repúdio), Information disclosure (divulgação de informações), Denial of service (negação de serviço), Elevation of privilege (elevação de privilégio)), a Tabela 9 mostra uma lista de ameaças, categorizadas por tipo, pontos de entrada e controles de segurança recomendados para mitigação.

Tabela 9. Lista de ameaças STRIDE

Tipo	Pontos de Entrada	Controle de Segurança
Spoofing	Dados armazenados no SGBD, Dados e informações do software do sistema, Dados e Informações do *Broker*, Dados e informações do sistema operacional, Sensores, Comunicações	Autenticação
Tampering	Dados armazenados no SGBD, Dados e informações do software do sistema, Dados e Informações do *Broker*, Dados e informações do sistema operacional, Hardware do *Gateway*, Software do *Firmware*, Hardware do Microcontrolador, Sensores, Comunicações, Reputação do Sistema	Integridade
Repudiation	Dados armazenados no SGBD, Dados e informações do software do sistema, Dados e Informações do *Broker*, Dados e informações do sistema operacional, Hardware do *Gateway*, Software do *Firmware*, Hardware do Microcontrolador, Sensores, Comunicações, Reputação do Sistema	Não Repúdio
Information Disclosure	Dados armazenados no SGBD, Dados e informações do software do sistema, Dados e Informações do *Broker*, Dados e informações do sistema operacional, Hardware do *Gateway*, Software do *Firmware*, Hardware do Microcontrolador, Sensores, Comunicações, Reputação do Sistema	Confidencialidade
Denial of Service	Comunicações	Disponibilidade
Elevation of Privilege	Dados armazenados no SGBD, Dados e informações do software do sistema, Dados e Informações do *Broker*, Dados e informações do sistema operacional, Comunicações	Autorização

2.7. Análise e classificação de ameaças com DREAD

A lista de ativos foi analisada utilizando o método *DREAD* que classifica as ameaças com base em cinco critérios:

- **Damage Potential (Potencial de Dano):** avalia o impacto que a ameaça pode causar se explorada. Isso inclui possíveis prejuízos financeiros, perda de dados, interrupção de serviços e comprometimento de informações sensíveis. Quanto maior o dano potencial, maior é a classificação de risco.
- **Reproducibility (Reprodutibilidade):** mede a facilidade com que um ataque pode ser replicado. Se a ameaça pode ser explorada de maneira consistente e repetitiva, sem a necessidade de condições específicas ou raras, ela apresenta alta reprodutibilidade. Quanto mais fácil for reproduzir o ataque, maior será a gravidade atribuída.
- **Exploitability (Explorabilidade):** refere-se à facilidade com que a vulnerabilidade pode ser explorada por um atacante. Isso inclui o nível de habilidade necessário, a disponibilidade de ferramentas automatizadas para explorar a ameaça e o custo de execução do ataque. Vulnerabilidades que são fáceis de explorar representam um risco mais elevado.

- **Affected Users (Usuários Afetados):** indica o número de usuários que podem ser impactados se a ameaça for explorada. Quanto maior o número de usuários afetados, mais crítica é a vulnerabilidade. Este critério ajuda a avaliar o alcance do problema, considerando não apenas a quantidade, mas também a importância dos usuários ou sistemas impactados.
- **Discoverability (Detectabilidade):** avalia a facilidade com que uma vulnerabilidade pode ser encontrada por um atacante. Se a vulnerabilidade é facilmente detectável por meio de métodos comuns, como varreduras automáticas ou inspeção visual, a detectabilidade é alta. Vulnerabilidades que são difíceis de descobrir tendem a ter um risco menor, pois exigem um esforço maior para serem identificadas.

Cada ameaça recebe uma pontuação de zero a 10 em cada critério, e a soma dessas pontuações gera um valor total para cada ameaça. As ameaças com pontuações mais altas foram priorizadas para mitigação. Esse processo permite uma revisão e atualização contínuas do modelo de ameaças conforme o sistema evolui. Com base na importância dos recursos para a organização, a equipe de desenvolvimento atribuiu a seguinte pontuação às ameaças, utilizando o método *DREAD*. Esse processo de classificação permitiu priorizar as ameaças mais críticas para mitigação, garantindo uma revisão e atualização contínuas do modelo de ameaças à medida que o sistema evolui. A Tabela 10 mostra a análise e classificação das ameaças com base no DREAD.

Tabela 10. Análise e classificação de ameaças DREAD

Ameaça aos Recursos	D	R	E	A	D	Total
Reputação do Sistema	5	10	10	10	10	45
Comunicações	7	10	10	8	10	45
Dados dos sensores	8	10	10	10	4	42
Hardware dos Sensores	6	10	10	4	10	40
Dados e informações do sistema operacional	10	2	3	10	2	27
Dados e informações do software do sistema web	8	3	3	10	2	26
Código-Fonte do sistema	10	3	3	8	2	26
Informações de Login do software do sistema web	1	8	9	6	1	25
Dados armazenados no SGBD	7	3	3	8	2	23
Software do *Firmware*	9	1	1	10	1	22
Dados e Informações do *Broker*	2	2	3	10	2	19
Hardware do *Gateway*	8	2	1	2	1	14
Hardware do Microcontrolador	8	2	1	2	1	14

2.8. Contramedidas e Mitigação

Com base nas análises das metodologias STRIDE e DREAD, a Tabela 11 apresenta as técnicas de mitigação e contramedidas. As ameaças são classificadas em: **Ameaças Não Mitigadas**, que correspondem a vulnerabilidades sem contramedidas eficazes, podendo resultar em impactos significativos se exploradas; **Ameaças Parcialmente Mitigadas**, que se referem a ameaças com algumas contramedidas que reduzem a exploração potencial, mas que ainda apresentam riscos limitados; e **Ameaças Totalmente Mitigadas**, que são aquelas com contramedidas adequadas, eliminando as vulnerabilidades e minimizando o risco. Essas classificações ajudam a definir o perfil de ameaça e a orientar a proteção do sistema.

Tabela 11. Contramedidas e Mitigação

Vulnerabilidade	Ameaças	Técnicas de Mitigação
Reputação do Sistema	Negação de serviço, Repúdio	Trilhas de auditoria, Qualidade do serviço
Comunicações	Negação de serviço	Filtragem, Limitação, Qualidade do serviço
Dados dos sensores	*Spoofing* dos sensores, Adulteração de dados	Autenticação apropriada, *Hashes*, MACs
Hardware dos Sensores	Hardware *Hacking*	Proteção fisica do hardware
Dados e informações do sistema operacional	Adulteração de dados, Divulgação de informação	*Hashes*, Criptografia
Dados e informações do software do sistema web	Adulteração de dados, Divulgação de informação	Assinaturas digitais, Proteção dos segredos
Código-Fonte do sistema	Adulteração do funcionamento legitimo do sistema, Adulteração de dados, Divulgação de informação	Assinaturas digitais, Criptografia
Informações de Login do software do sistema web	Falsificação de identidade	Autenticação apropriada
Dados armazenados no SGBD	Adulteração de dados, Divulgação de informação	Autorização apropriada, Criptografia
Software do *Firmware*	Dump do firmware, Adulteração de dados	Assinaturas digitais
Dados e Informações do *Broker*	Adulteração de dados, Divulgação de informação	*Criptografia*, Autorização apropriada
Hardware do *Gateway*	Hardware *Hacking*	Proteção fisica do hardware
Hardware do Microcontrolador	Hardware *Hacking*	Proteção fisica do hardware

3. Trabalhos Relacionados

A literatura sobre *Threat Modeling* apresenta várias abordagens e metodologias. Esta seção revisa trabalhos significativos na área. Uma abordagem inicial importante foi a metodologia STRIDE, descrita por Howard e LeBlanc (2002), que categoriza ameaças e orienta a análise de segurança durante o desenvolvimento de software. A metodologia DREAD, promovida por Shostack (2014), adiciona uma dimensão de avaliação de risco ao STRIDE, ajudando a priorizar ameaças com base em impacto e probabilidade. Estudos de caso mostram a eficácia da modelagem de ameaças. Awojana (2018) aplicou *Threat Modeling* a um sistema de gerenciamento de aplicações web. A integração com práticas DevSecOps é abordada por Francis (2019), que discute a automação de *Threat Modeling* em pipelines de desenvolvimento contínuo. Um estudo abrangente foi realizado na BITM[2], explorando metodologias como STRIDE e DREAD. A revisão revelou que **STRIDE** é mais adequada para categorizar ameaças detalhadamente. Enquanto o **DREAD** se mostra mais valiosa para priorização de riscos; A escolha da metodologia deve considerar o contexto específico e os requisitos do sistema. Compreender as vantagens e desvantagens de cada abordagem contribui para um desenvolvimento mais seguro e resiliente.

4. Considerações Finais

A modelagem de ameaças é fundamental para garantir a segurança de sistemas IoT complexos. Este estudo demonstrou que as metodologias STRIDE e DREAD, quando aplicadas de forma complementar, permitem uma identificação e mitigação eficaz de ameaças, com STRIDE categorizando-as e DREAD priorizando os riscos. A escolha da abordagem depende da complexidade do sistema e dos recursos disponíveis, sendo crucial uma revisão contínua do modelo de ameaças conforme o sistema evolui. Com o aumento da adoção de sistemas IoT é cada vez mais necessário que as organizações mantenham uma abordagem proativa na atualização de suas estratégias de segurança.

[2]BITM: `https://doi.org/10.5281/zenodo.13821928`

Tabela 12. Tabela Comparativa dos Trabalhos Relacionados

Estudo	Objetivos	Contribuições	Escopo
[Howard and LeBlanc 2002]	Categorizar ameaças e orientar análise	Introdução do STRIDE para categorização de ameaças	Desenvolvimento de software
[Shostack 2014]	Avaliação de riscos e priorização	Extensão do STRIDE com DREAD para avaliação de risco	Desenvolvimento de software
[Deng et al. 2011]	Identificação de ameaças à privacidade	Proposta da metodologia LINDDUN	Privacidade em software
[Awojana 2018]	Aplicação de *Threat Modeling*	Estudo de caso em sistema de gerenciamento de aplicações web	Aplicações web
[Wuyts et al. 2014]	Avaliação da LINDDUN em projetos grandes	Estudo empírico da LINDDUN	Projetos de software de grande escala
[Ahmed and Francis 2019]	Integração com DevSecOps	Automação de *Threat Modeling*	Pipelines de desenvolvimento contínuo

Disponibilidade de Dados

Os dados gerados e/ou analisados durante este estudo estão disponíveis no [repositório Zenodo] (`https://doi.org/10.5281/zenodo.13821928`), sob o projeto intitulado "Basic IoT Threat Modeling (BITM)". Os dados estão acessíveis gratuitamente para pesquisas e análises futuras, sob a licença específica associada ao repositório.

Referências

Ahmed, Z. and Francis, S. C. (2019). Integrating security with devsecops: Techniques and challenges. In *Proceedings of the 2019 International Conference on Digitization (ICD)*, pages 178–182. IEEE.

Alves, L. F., Amalfi, P., and Martin, S. (2024). Basic iot threat modeling (bitm). `https://doi.org/10.5281/zenodo.13821928`. Accessed: October 15, 2024.

Awojana, T. B. (2018). Threat modelling and analysis of web application attacks.

Conviso (2024). Threat modeling. Accessed: 2024-06-17.

Deng, M., Wuyts, K., Scandariato, R., Preneel, B., and Joosen, W. (2011). Linddun: A privacy threat modeling framework. In *Proceedings of the 7th Symposium on Usable Privacy and Security*.

Howard, M. and LeBlanc, D. (2002). *Writing Secure Code*. Microsoft Press.

Infomach (2024). Conheça o modelo de ameaça stride. Accessed: 2024-06-17.

National Institute of Standards and Technology (2006). Minimum Security Requirements for Federal Information and Information Systems. Technical Report FIPS PUB 200, NIST. Accessed: October 15, 2024.

OWASP (2024). Threat modeling process. `https://owasp.org/www-community/Threat_Modeling_Process`. Accessed: October 15, 2024.

Shostack, A. (2014). *Threat Modeling: Designing for Security*. Wiley.

Wuyts, K., Scandariato, R., and Joosen, W. (2014). Empirical evaluation of a privacy-focused threat modeling methodology. *Journal of Systems and Software*, 96:122–138.