Relatório Técnico: Simulação de Ataques e Estratégias de Defesa em Arquiteturas IoT Multicamadas

Curso: Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Disciplina: Tópicos Avançados em WEB I

Discente: Gabriel Lima, Rariel

Docente: Felipe Silva

Data: 24 de Agosto de 2025

Sumário

---

1. Descrição do Cenário

Este projeto simula um ambiente de "Casa Conectada", um cenário de IoT (Internet das Coisas) onde diversos dispositivos monitoram as condições do ambiente para prover segurança e conforto ao morador. O objetivo é demonstrar o impacto de ciberataques comuns nesse tipo de arquitetura e a eficácia de contramedidas de segurança.

Para representar este cenário, foram implementados três tipos de sensores virtuais:

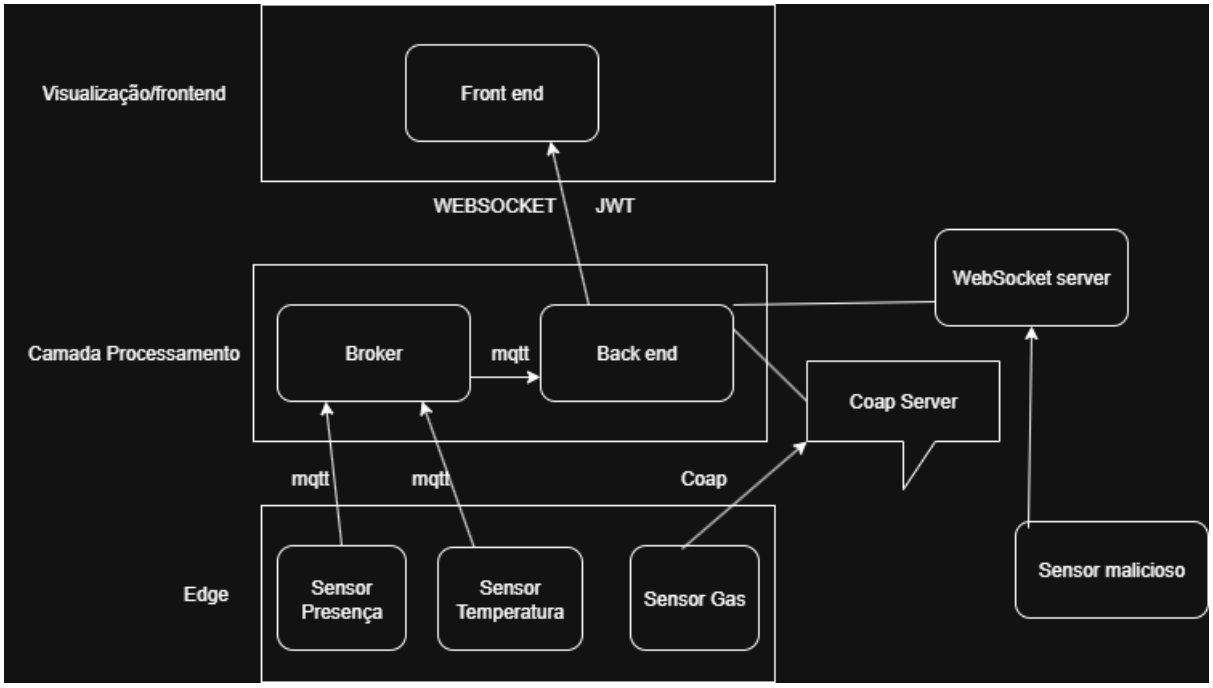
- **Sensor de Temperatura:** Monitora a temperatura ambiente, reportando dados em graus Celsius (°C). Utiliza o protocolo **MQTT** para comunicação, ideal para mensagens leves e eficientes.
- **Sensor de Presença:** Detecta a presença de pessoas em um cômodo, enviando valores booleanos (0 ou 1). Também utiliza o protocolo **MQTT**.
- **Sensor de Gás:** Mede a concentração de gás (ex: GLP) no ar, reportando em partes por milhão (ppm). Este sensor utiliza o protocolo **CoAP** (Constrained Application Protocol), adequado para dispositivos com recursos limitados e comunicação via UDP.

A escolha de múltiplos sensores e protocolos distintos (MQTT e CoAP) atende aos requisitos da atividade, permitindo a análise de diferentes vetores de comunicação e vulnerabilidades associadas.

---

2. Diagrama e Descrição da Arquitetura

A arquitetura do sistema foi projetada seguindo um modelo multicamadas, dividindo as responsabilidades para aumentar a modularidade, escalabilidade e segurança.



A arquitetura é composta pelas seguintes camadas:

1. **Camada de Dispositivos/Borda (Edge Layer):**
    - **Componentes:** Sensores simulados (Temperatura, Presença, Gás) e o sensor malicioso.
    - **Responsabilidade:** Coletar dados do ambiente. Os sensores legítimos são responsáveis por criptografar os dados na origem antes de enviá-los, garantindo a segurança desde a borda. A comunicação é feita via protocolos MQTT e CoAP.
2. **Camada de Comunicação e Processamento (Fog/Cloud Layer):**
    - **Componentes:** Broker MQTT (Mosquitto), Backend (FastAPI), Servidor CoAP.
    - **Responsabilidade:** Esta camada atua como o cérebro do sistema.
        - O **Broker MQTT** recebe as mensagens dos sensores de temperatura e presença.
        - O **Backend em FastAPI** se inscreve nos tópicos do broker para receber os dados, descriptografa as mensagens, processa-as, e as transmite via WebSockets para a camada de visualização. Ele também hospeda o servidor CoAP para receber dados do sensor de gás e um endpoint para gerar tokens JWT.
3. **Camada de Aplicação/Visualização (Cloud Layer):**
    - **Componentes:** Frontend (HTML, CSS, JavaScript) servido por um Nginx.

- **Responsabilidade:** Apresentar os dados aos usuários. O frontend estabelece uma conexão segura via WebSocket com o backend (autenticada por JWT), recebendo e exibindo os dados dos sensores e os alertas de segurança em gráficos e listas em tempo real.

Toda a arquitetura é orquestrada utilizando Docker e Docker Compose, garantindo que cada componente (broker, backend, frontend) opere em um contêiner isolado, facilitando a implantação e a interoperabilidade.

---

3. Simulação de Ameaças e Evidências

Para avaliar a robustez da arquitetura, foram simulados dois tipos de ataque, conforme solicitado na atividade.

3.1. Ataque de Interceptação de Dados (Envio em Texto Plano)

- **Descrição do Ataque:** Um dispositivo não autorizado (sensor_temperatura_malicioso_mqtt.py) foi introduzido na rede. Este sensor se conecta ao mesmo broker MQTT e publica mensagens no mesmo tópico que o sensor de temperatura legítimo. No entanto, o payload é enviado como um JSON em texto plano, sem qualquer criptografia. Um atacante que consiga acesso à rede poderia facilmente ler esses dados, violando a confidencialidade das informações.
- **Evidências:**
    1. **Log do Backend:** O log do backend mostra a detecção da ameaça. Ao receber uma mensagem que não pôde ser descriptografada, o sistema a verifica. Se for um JSON válido, ele a classifica como um ataque de "Dados em Texto Plano", gera um alerta e a descarta.
    2. **Alerta no Frontend:** O alerta gerado pelo backend é enviado em tempo real para a interface do usuário, notificando sobre a atividade maliciosa.

3.2. Ataque de Acesso Não Autorizado (Token JWT Inválido)

- **Descrição do Ataque:** Este ataque simula uma tentativa de um cliente não autorizado de se conectar à stream de dados em tempo real. O acesso ao WebSocket do backend é protegido e requer um token JWT válido. Para simular o ataque, o código do cliente no script.js foi modificado para deliberadamente corromper o token antes de tentar a conexão.
- **Evidências:**
    1. **Log do Backend:** O backend valida cada token recebido. Ao receber o token inválido, ele recusa a conexão e registra o evento, informando que a tentativa falhou devido a um "Token inválido".

2. **Console do Navegador:** O navegador do cliente que tentou se conectar com o token inválido exibe um erro no console, indicando que a conexão WebSocket falhou. Isso demonstra que a barreira de proteção funcionou como esperado.

---

4. Estratégias de Defesa Implementadas

Para mitigar os ataques simulados e proteger o sistema, foram implementados três mecanismos de defesa principais.

4.1. Criptografia de Ponta a Ponta com AES-GCM

Para garantir a

confidencialidade e integridade dos dados em trânsito, todas as mensagens enviadas pelos sensores legítimos são criptografadas na origem usando o algoritmo AES-GCM (Advanced Encryption Standard in Galois/Counter Mode). O backend é o único com a chave secreta para descriptografar essas mensagens. Isso impede que um atacante que intercepte o tráfego de rede (ataque

Man-in-the-Middle) consiga ler ou alterar os dados sem ser detectado.

4.2. Autenticação e Autorização via JWT (JSON Web Tokens)

Para proteger a camada de visualização, foi implementado um sistema de autenticação baseado em tokens.

1. O cliente frontend primeiro solicita um token a um endpoint

/token no backend.

2. O backend gera um JWT assinado com uma chave secreta e com tempo de expiração.
3. O cliente deve incluir este token ao tentar estabelecer uma conexão WebSocket. O backend valida a assinatura e a expiração do token antes de aceitar a conexão. Isso garante que apenas clientes autenticados possam receber os dados dos sensores, prevenindo o acesso não autorizado.

4.3. Monitoramento e Alertas de Segurança em Tempo Real

O sistema possui um mecanismo de monitoramento ativo. O backend foi programado para identificar anomalias, como o recebimento de mensagens não criptografadas no tópico MQTT. Ao detectar tal evento, ele:

1. Registra o incidente em seus logs.
2. Gera uma mensagem de alerta estruturada.
3. Transmite esse alerta via WebSocket para todos os clientes conectados.

Isso permite uma resposta rápida a possíveis ameaças, transformando a interface de visualização de dados também em um painel de monitoramento de segurança.

---

5. Análise Crítica

5.1. Análise de Segurança

A simulação demonstrou que, sem as devidas contramedidas, uma arquitetura IoT é altamente vulnerável. O ataque de envio de dados em texto plano expôs informações sensíveis, enquanto o ataque de acesso não autorizado poderia levar ao vazamento de todo o fluxo de dados.

- **Antes das defesas:** O sistema era frágil. Qualquer dispositivo na rede poderia publicar dados falsos ou ler informações, e qualquer cliente web poderia se conectar para visualizar os dados.
- **Após as defesas:** A implementação de criptografia e autenticação com JWT elevou drasticamente o nível de segurança. A criptografia garante que os dados sejam ininteligíveis para quem os intercepta. A autenticação com JWT assegura que apenas usuários autorizados tenham acesso à visualização. O sistema de alertas adiciona uma camada de defesa proativa, permitindo a identificação de atividades suspeitas.

5.2. Análise de Interoperabilidade

A escolha de tecnologias e padrões abertos foi fundamental para a interoperabilidade do sistema.

- **Protocolos Padrão:** O uso de MQTT e CoAP permite que dispositivos de diferentes fabricantes se comuniquem com o backend sem a necessidade de adaptações complexas.
- **Conteinerização:** O uso de Docker e Docker Compose desacopla os serviços. O broker MQTT, o backend e o frontend rodam em contêineres independentes, podendo ser atualizados, substituídos ou escalados individualmente, sem impactar o resto do sistema. Essa abordagem modular é um pilar para sistemas interoperáveis e de fácil manutenção.

5.3. Análise de Privacidade e Conformidade com a LGPD

Os dados coletados por sensores em uma casa conectada podem ser extremamente pessoais, revelando hábitos e rotinas dos moradores. A Lei Geral de Proteção de Dados (LGPD) exige que dados pessoais sejam tratados com segurança e para finalidades específicas.

- **Impacto da Criptografia:** A criptografia de ponta a ponta é uma medida técnica essencial para a conformidade com a LGPD, pois protege os dados contra acessos não autorizados, minimizando o risco de vazamentos.
- **Boas Práticas:** Além da criptografia, outras práticas alinhadas à LGPD poderiam ser implementadas, como a anonimização de dados quando possível, políticas claras de retenção de dados (descartando informações antigas que não são mais necessárias) e a implementação de consentimento explícito do usuário para a coleta de cada tipo de dado.

---

6. Conclusão

Este trabalho demonstrou com sucesso a implementação de uma arquitetura IoT multicamadas para o cenário de uma casa conectada, a simulação de ataques cibernéticos relevantes e a aplicação de estratégias de defesa eficazes. A utilização de criptografia AES-GCM e autenticação via JWT provou ser robusta na proteção da confidencialidade e do controle de acesso. O projeto ressalta a importância crítica de se considerar a segurança desde a concepção (*Security by Design*) em sistemas IoT, garantindo não apenas a funcionalidade, mas também a privacidade e a confiança do usuário final.