Sistema para detecção e prevenção de ataques a caixas eletrônicos: Um estudo de caso

Luiz Alfredo Thomasini[1]

[1]Programa de Graduação em Ciência da Computação
Universidade do Rio dos Sinos (Unisinos)
São Leopoldo – RS – Brazil

{luizalfredo}@edu.unisinos.br

Resumo. *Este artigo compreende a segunda entrega do projeto aplicado em Internet das Coisas (IoT) como forma de comprovar o conhecimento adquirido na disciplina. A primeira parte do projeto teve como objetivo a especificação de um cenário como estudo de caso e uma solução baseada em IoT, que foi definido como um sistema de detecção e prevenção de ataques em caixas eletrônicos. Nesta segunda parte será apresentada uma prova de conceito da solução proposta.*

1. Introdução

Caixas eletrônicos ainda são amplamente utilizados no sistema financeiro. Devido ao seu valor e quantidade de cédulas de papel moeda que possuem armazenadas, são alvos frequentes de ataques e tentativas de roubo ou fraudes. Sendo assim, precisam de um sistema de segurança para monitoramento e detecção de possiveis ataques.

Estes equipamentos geralmente estão espalhados por diversos ambientes, como em agências bancárias, postos de gasolina, supermercados, entre outros. Para este problema, propomos uma solução baseada em sistemas distribuídos [Van Steen and Tanenbaum 2017] e monitoramento e sensoriamento desses aparelhos através de Internet das Coisas (IoT) [Li et al. 2015].

Retomando o desenvolvimento da primeira parte do projeto, esta segunda parte propõe a a construção de uma prova de conceito de uma solução que cumpra as seguintes etapas de um sistema IoT:

I Interação com o ambiente para sensoriamento;
II Comunicação de dispositivos para coleta de dados;
III Armazenamento e análise das informações;
IV Visualização dos resultados da análise;

Portanto, este artigo da seguinte forma: A seção 2 vai descrever a solução implementada na prova de conceito e listar adaptações feitas na especificação. A seção 3 vai descrever e aprofundar nos componentes de hardware e software que foram utilizados. Finalmente, a seção 4 vai apresentar um estudo de caso ponta-a-ponta simulado da prova de conceito.

2. Visão geral da solução

O protótipo do sistema implementado para o teste do conceito proposto se baseia num dispositivo de aquisição e transmissão de dados junto ao caixa eletrônico, baseado numa

placa de desenvolvimento microcontrolada ESP32 [esp 2024] que realiza leituras de movimento através de um sensor PIR Motion Sensor [pir 2024], e envia os dados para um broker MQTT [mqt 2024] hospedado em nuvem. Estes dados do sensor são consumidos e processados para armazenamento em um banco de dados e consultados por uma ferramenta de visualização dos dados temporais por profissionais de segurança de agências bancárias.

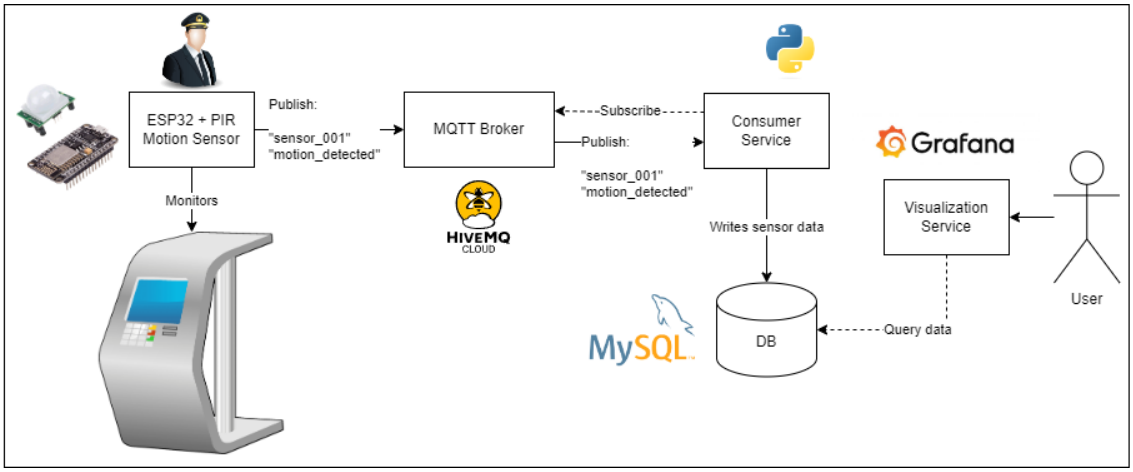

Figure 1. Diagrama da Solução

Um diagrama esquemático do sistema proposto pode ser visto na Figura 2, onde observa o dispositivo de monitoramento e transmissão de dados acoplado ao caixa eletrônico e a arquitetura de processamento e armazenamento que pode ser executada em containers e hospedada em provedores de nuvem, como a AWS.

2.1. Modelo do dispositivo de monitoramento

O microcontrolador ESP32 para o aquisição e transmissão de dados opera de acordo com o seguinte modelo para fazer o monitoramento do caixa eletrônico:

1. O dispositivo é inicializado e define uma variável chamada *MOTION* para o valor booleano **FALSE**;
2. Após inicializado, o dispositivo faz uma leitura do movimento em frente ao caixa eletrônico a cada 10 segundos;
3. Assim que alguma pessoa se aproxima do caixa eletrônico, o sensor realiza a leitura do movimento e define a variável para **TRUE**;
4. A cada leitura realizada, o dispositivo transmite os dados para o Broker MQTT (em um dia de 24 horas, teremos uma leitura feita a cada 10 segundos, o que contabiliza um total de 8640 leituras);

Algumas adaptações foram feitas nesse componente do sistema em relação a especificação. Originalmente era previsto o uso do microcontrolador ESP32-CAM para, além de identificar o movimento em frente ao caixa eletrônico, também realizar a autenticação da pessoa. Por limitações de hardware foi sábio trocar o módulo de processamento de imagem da ESP32-CAM pelo sensor PIR Motion Sensor que ainda mantém parte das características da solução proposta na parte um.

2.2. Modelo do sistema de processamento

Após a aquisição e transmissão dos dados, precisamos garantir que esses dados sejam processados entregues aos usuários para efetivamente gerar valor. Utilizamos uma arquitetura no modelo *Publisher-Subscriber* baseada no protocolo MQTT. No desenvolvimento da prova de conceito, utilizamos o Broker MQTT gerenciado na nuvem da HiveMQ [hiv 2024] pela simplicidade do uso e velocidade no desenvolvimento, mas essa decisão pode ser revisada para uma solução própria dependendo os critérios de carga e segurança do sistema. A partir de que os dados são publicados no tópico MQTT, o modelo segue da seguinte forma:

1. Um serviço *Python* é responsável por subscrever no tópico e consumir os dados;
2. O serviço faz a validação do conteúdo da mensagem;
3. Se a mensagem está no padrão, os dados são armazenados. Se não, a mensagem é descartada;

Neste componente também houve uma modificação: substituímos o servidor web baseado em HTTP proposto na especificação original por uma arquitetura *Pub-Sub* baseada em MQTT por simplicidade de uso e possibilidade de utilizar um serviço hospedado na nuvem.

2.3. Modelo de armazenameto e visualização dos dados

Por fim, queremos armazenar os dados de monitoramento em um banco de dados para consultá-los e extrair valor para os usuários. Modelamos uma tabela do banco de dados para armazenar cada leitura feita por cada dispotivo de monitoramento integrado ao sistema, a tabela 1 descreve os campos armazenados.

Campo	Descrição
ID	Identificador do registro no banco de dados. Auto incremental.
SENSOR_ID	Campo texto do identificador único do dispositivo de monitoramento. Obrigatório.
MOTION_DETECTED	Campo booleano da informação do sensor de movimento no momento da leitura.
TIMESTAMP	Campo datahora do momento da leitura pelo dispositivo de monitoramento. Adicionado automaticamente se nenhum valor for inserido.

Table 1. Descrição dos Campos

Uma pequena alteração deste componente foi em relação a tecnologia a tecnologia do banco de dados, que na especificação original era Postgres mas foi alterado para MySQL por familiaridade. Mantivemos o uso do Grafana [gra 2024] para a visualização. Algumas capturas de tela dos dados serão apresentadas na seção 4.

3. Detalhamento dos componentes

3.1. Componentes de hardware

Componentes de hardware utilizados na prova de conceito:

1. **ESP32** [esp 2024]: Microcontrolador de baixo consumo de energia com conectividade WiFi para conectar e energizar sensores, realizar a leitura e processamento dos dados.

2. **PIR Motion Sensor** [pir 2024]: Sensor *Pyroelectric ("Passive") InfraRed* de movimento com tecnologia infravermelho que detecta movimento de objetos através da radiação e temperatura. Possui pino de entrada de energia de 3-5V, pino GND e pino para leitura de dados. Consegue realizar a leitura de até 6 metros e é especialmente utilizado para detectar se algum humano se moveu em frente ao sensor. Custa em torno de R$10.

Figure 2. PIR Motion Sensor

3.2. Componentes de software

Componentes de software utilizados na prova de conceito:

1. **Docker** [doc 2024]: Serviço open-source de containers ideal para rodar as aplicações e garantir a portabilidade da nossa prova de conceito, como veremos no setup do Estudo de Caso.
2. **Grafana** [gra 2024]: Serviço open-source de visualização de dados com gerenciamento de conexão a banco de dados, construção de dashboards e criação de alertas.
3. **MQTT** [mqt 2024]: Protocolo padrão para comunicação de aplicações IoT devido a arquitetura desacoplada e que exige pouca largura de banda.
4. **MySQL** [mys 2024]: Serviço de banco de dados SQL open-source.
5. **Python** [pyt 2024]: Linguagem de programação utilizada para desenvolver os serviços devido a velocidade de escrita e disponibilidade de bibliotecas open-souce para integração com os outros serviços.

4. Estudo de Caso

Considere que desejamos monitorar um caixa eletrônico que fica dentro de uma agência bancária. Esta agência opera em um período comercial das 06 horas da manhã até às 19h da noite. Durante esse período, pessoas vão até o caixa para sacar cédulas de papel moeda, pagar boletor, realizar cadastros, etc. Queremos monitorar o movimento em frente a este caixa eletrônico para encontrar movimentações suspeitas que não estão relacionadas a estes usos comuns do equipamento. Para isso, vamos simular essa situação de acordo com alguns parâmetros e analisar os dados para ver que informações conseguimos extrair dos dados.

4.1. Simulação

No nosso simulador, simulamos as leituras do nosso dispositivo de monitoramento. Definimos o horário comercial da agência bancária no período das 06 horas da manhã até às 19h da noite. A movimentação no caixa eletrônico é dada por uma probabilidade de uma pessoa aparecer a cada instante de leitura. O tempo que a pessoa fica no caixa é dado por uma distribuição uniforme: durante esse período, a informação lida pelo sensor é **TRUE**. Definimos que uma atividade suspeita como um tempo exagerado que o movimento é percebido no caixa eletrônico que não condiz com as atividades comuns listadas acima. A tabela 2 descreve os parâmetros configuráveis do nosso simulador:

Parâmetro	Descrição
DAYS_TO_SIMULATE	Quantos dias queremos simular.
READ_INTERVAL_SECONDS	Qual o intervalo de cada leitura (segundos).
MOTION_PROBABILITY	Qual a probabilidade de uma pessoa aparecer no caixa eletrônico.
OFFHOURS_PROBABILITY	Qual a probabilidade de uma pessoa aparecer no caixa eletrônico fora do horário comercial.
SUSPICIOUS_PROBABILITY	Qual a probabilidade de acontecer uma atividade suspeita no caixa eletrônico.
NORMAL_DURATION_MINUTES	Parâmetros de mínimo e máximo da nossa distribuição uniforme.
SUSPICIOUS_DURATION_MINUTES	Tempo da atividade suspeita (minutos).

Table 2. Parâmetros da simulação

4.2. Setup

Para executar a simulação, devemos seguir uma série de passos:

1. Inicializar o banco de dados e criar a tabela para armazenar os dados do sensor.
2. Inicializar o grafana e configurar a consulta dos dados.
3. Inicializar o serviço consumidor e conectar no Broker MQTT.
4. Configurar os parâmetros da simulação e inicializar o simulador.

Para simplificar este processo, disponibilizamos o ambiente configurado em um arquivo *docker-compose* e um script *setup.py* para inicializar as tabelas do banco de dados e testar as integrações (repositório dispoível em).

4.3. Resultados e discussões

Executamos uma simuação com os parâmetros apresentados na tabela 3 e observamos o comportamento em um painel no Grafana.

As Figuras 3 e 4 mostram o comportamento capturado em frente ao caixa eletrônico no período de 5 dias e 1 dia, respectivamente. O eixo X é o tempo da leitura do dado, no nosso caso tempos um datapoint a cada 10 segundos. O eixo Y é o valor do sensor: **FALSE** quando não há ninguém em frente ao caixa eletrônico, ou **TRUE** quando há movimento.

Como vemos na figura 3, não foi registrado nenhum movimento suspeito fora do horário comercial da agência bancária. Desta visão, não conseguimos detectar nada excepcional ocorrendo. Por outro lado, a Figura 4 mostra os dados de leitura em um período de 1 dia. Neste caso, vemos que foi registrado uma movimentação suspeita no período da tarde em frente ao caixa eletrônico.

Parâmetro	Descrição
DAYS_TO_SIMULATE	5
READ_INTERVAL_SECONDS	10
MOTION_PROBABILITY	0.01 (1%)
OFFHOURS_PROBABILITY	0.1 (10%) (multiplicado por 1%)
SUSPICIOUS_PROBABILITY	0.1 (10%) (multiplicado por 1%)
NORMAL_DURATION_MINUTES	(5, 15)
SUSPICIOUS_DURATION_MINUTES	50

Table 3. Parâmetros utilizados no experimento

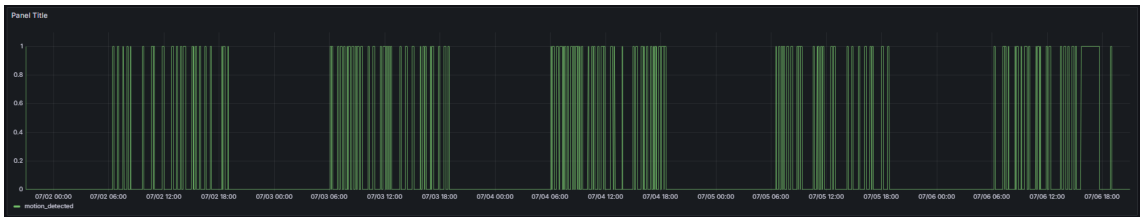

Figure 3. Resultados 5 dias.

A partir dessas visualizações, é difícil de tirar conclusões sobre se houve ou não tentativa de ataque ou fraude no caixa eletrônico que foi registrado no último dia de leitura. Para os usuários responsáveis pela segurança dos caixas eletrônicos essa informação pode servir de alerta que devem aprofundar na investigação, recuperando gravações de câmeras de segurança ou observando o comportamento por mais alguns dias e semanas.

5. Conclusão

O sistema mostra ser eficiente para gravar a movimentação em frente a caixas eletrônicos e encontrar comportamentos que fogem aos padrões identificados. Como evoluções futuras, podemos imaginar acomplar novos sensores aos caixas eletrônicos para enriquecer ainda mais a informação. Podemos evoluir também a parte de análise dos dados para entregar ainda mais valor aos usuários, como por exemplo: identificar padrões comportamentais que possam gerar insights para otimizar custos; automatizar análises para encontrar comportamentos fora do comum e alertar os usuários, removendo a necessidade de fazer a análise manual dos dashboards.

No geral, o estudo de caso exemplifica uma aplicação prática do sistema de monitoramento e como ele agrega valor para os usuários.

References

(2024). Docker (https://www.docker.com/).

(2024). Esp32 (https://www.espressif.com/en/products/socs/esp32).

(2024). Grafana (https://grafana.com/).

(2024). Hivemq (https://www.hivemq.com/).

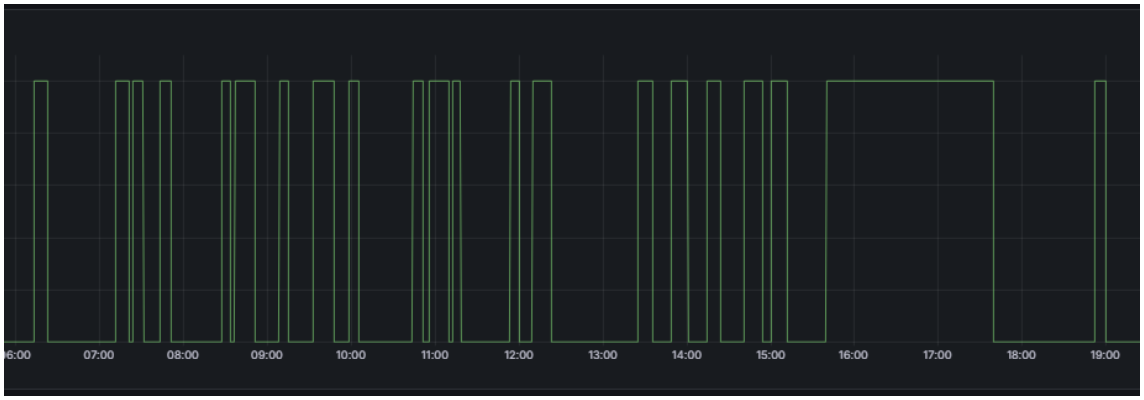

Figure 4. Resultados 1 dia.

(2024). Mqtt (https://mqtt.org/).

(2024). Mysql (https://www.mysql.com/).

(2024). Pir motion sensor. pyroelectric (”passive”) infrared sensors.

(2024). Python (https://www.python.org/).

Li, S., Xu, L. D., and Zhao, S. (2015). The internet of things: a survey. *Information systems frontiers*, 17:243–259.

Van Steen, M. and Tanenbaum, A. S. (2017). *Distributed systems*. Maarten van Steen Leiden, The Netherlands.