Stratégie de sécurité - Projet IoT

AUTHORS

- GAMBART Louis
- MECENE Guillaume

Introduction

Dans ce document, nous présentons les différentes idées et solutions de sécurité que nous pouvons envisager pour notre projet IoT.

Certaines seront appliquées, et d'autres non, selon la faisabilité et la cohérence avec notre système.

Rappel du contexte

Le projet vise à concevoir un système complet de surveillance environnementale permettant :

- la **mesure locale** de différents paramètres (humidité du sol, température, luminosité…),
- l'**affichage sur un écran LCD**,
- la **transmission longue portée via LoRa**,
- l'envoi des données vers un **serveur MQTT** via ESP32,
- l'exploitation des données dans un **dashboard externe**.

1. Obectifs de sécurité

Notre système doit garantir :

- l'intégrité des données,
- l'authenticité des messages,
- la disponibilité du système,
- la confidentialité des communications,
- la robustesse face aux erreurs ou pannes matérielles.

2. Modèle de menace

Les risques identifiés :

- Envoi de fausses données via LoRA,

- Interception ou modification du trafic MQTT,
- Manipulation physique de l'Arduino,
- Attaques depuis le réseau WiFi,
- Corruption des données dans le dashboard.

3. Sécurité du noeud capteur

3.1. Sécurité physique

- Placer l'Arduino dans un boitier fermé,
- Limiter l'accès aux ports et câbles,
- Fixer les composants pour éviter les débranchements accidentels.

3.2. Sécurité logicielle

- Utiliser le Watchdog pour redémarrer automatiquement en cas de blocage,
- Vérifier les données des capteurs (valeurs aberrantes ignorées),
- Retourner un état clair (`state = ERREUR`) si un capteur est défaillant.

4. Sécurité de la liaison LoRA

4.1. Authentification et intégrité

Chaque message LoRA doit contenir :

- les valeurs des capteurs
- un compteur ou timestamp,
- un HMAC-SHA256 basé sur une clé partagée.

Cela permet de s'assurer que le message n'a pas été modifié et qu'il vient du bon émetteur.

4.2. Protection contre les relectures

L'ESP32 vérifiera que le compteur `t` augmente à chaque message. Si une trame est ancienne ou déjà reçue, elle sera rejetée.

4.3. Exemple de trame

```
{ "soil": 23, "temp": 21.4, "hum": 55, "lux": 1200, "state": 0,
"t": 1025, "mac": "..." }
```

5. Sécurité de la passerelle

5.1. Sécurité WiFi

- Utiliser WPA2 ou WPA3,
- Mot de passe long et robuste,
- Pas de connexion à un réseau non fiable.

5.2. Durcissement de l'ESP32

- Désactiver les ports debug en production,
- Gérer proprement les erreurs
- Publier uniquement si la trame LoRA est valide.

6. Sécurité MQTT

6.1. Authentification

- Utiliser un compte dédié à l'ESP32,
- Mettre en place des ACL stricts :
  - ESP32 : PUBLISH
  - Dashboard : SUBSCRIBE

6.2. Chiffrement

- Utiliser TLS (MQTTS),
- Les certificats auto-signés sont suffisants en local.

6.3. Cloisonnement réseau

- Ne pas exposer le broker sur Internet,
- Le rendre accessible uniquement via le LAN ou un VPN.

7. Sécurité du dashboard

- Protéger l'accès par mot de passe,
- Utiliser HTTPS pour un accès extérieur,
- Mettre à jour les outils du dashboard,
- Se protéger des injections (si possibilité d'entrée de données dans le dashboard).

8. Supervision et détection d'anomalies

8.1. Monitoring

- Envoyer régulièrement un message `plante1/system/last_update`,
- Alerter si aucune donnée n'est reçue pendant un certain temps,
- Journaliser les trames LoRA invalides.

8.2. Détection d'anomalies

- Identifier les données incohérentes (capteur défaillant)
- Détecter une activité anormale.

Conclusion

Cette stratégie permet de couvrir la sécurité :

- du matériel,
- des communications radio,
- de la passerelle réseau,
- du serveur MQTT,
- et de l'interface utilisateur.

Elle vise à assurer un système fiable, authentique et résistant aux attaques tout en restant compatible avec notre projet IoT.