

# Labo 2 de Cybersécurité

## Analyse de vulnérabilités du système PI2C Championship

Martin Fockedey

### Introduction

Ce laboratoire a pour objectif de vous confronter à une situation réaliste d'analyse de sécurité de système IT. L'exercice s'inspire de situations professionnelles réelles où l'analyste de sécurité se voit confier un système supposé fonctionnel, mais dont le niveau de sécurité n'a pas été évalué.

Le système étudié est un *Raspberry Pi* configuré pour héberger une variante du gestionnaire de tournoi développé dans le cadre du projet BA2 par monsieur Lurkin. Les enseignants souhaitent commercialiser cet appareil comme un outil pédagogique permettant aux professeurs d'organiser des tournois interactifs dans leurs classes.

Votre rôle, en tant qu'analyste de sécurité, est de réaliser une évaluation complète du système, d'en identifier les faiblesses et d'en démontrer, si possible, l'exploitabilité.

### 1. Contexte détaillé

Les enseignants de l'ECAM, avec **DLH** comme responsable principal du projet, souhaitent mettre sur le marché un produit pédagogique innovant : un Raspberry préconfiguré qui offre son propre réseau Wi-Fi et sert de plateforme de jeu.

Chaque Raspberry embarque :

- un gestionnaire de tournoi permettant à des étudiants de s'affronter (un code adapté de celui de M. Lurkin et le code d'un nouveau jeu) ;
- un serveur web contenant des règles du jeu et un client d'exemple ;
- son propre réseau wifi ;

L'idée est simple : un professeur apporte son Raspberry en classe, l'allume, attend que les étudiants s'y connectent en Wi-Fi, puis laisse le gestionnaire organiser automatiquement les matchs.

Cependant, ce système n'a pas été conçu selon des standards de sécurité. DLH souhaite donc vérifier si une commercialisation serait envisageable sans risque.

## **2. Fonctionnement complet du gestionnaire de partie**

Le gestionnaire organise un tournoi en plusieurs étapes :

### **2.1 Inscription des joueurs**

Les participants doivent s'inscrire auprès du gestionnaire. Celui-ci crée une identité unique pour chaque joueur et vérifie périodiquement leur disponibilité.

### **2.2 Création automatique des matchs**

Dès que deux joueurs ou plus sont inscrits, le gestionnaire génère un calendrier de matchs :

- chaque joueur affronte tous les autres ;
- les matchs sont joués deux fois : une fois dans chaque rôle ;
- le serveur gère automatiquement le déroulement du tournoi.

### **2.3 Déroulement des matchs**

Pendant un match :

- le serveur envoie tour à tour l'état du jeu à chaque joueur ;
- chaque joueur répond par un coup ;
- un coup invalide, trop tardif ou l'absence de réponse entraîne une pénalité, après un certain nombre de pénalités, le joueur perd le match ;

### **2.4 Enregistrement des résultats**

À la fin de chaque match :

- le résultat est consigné ;

- le gestionnaire passe automatiquement au match suivant ;
- à la fin du tournoi, un classement général est produit.

## 3. Infrastructure matérielle et logicielle

Chaque groupe reçoit l'accès à un Raspberry (sans accès physique) et doit analyser les services exposés

### 3.2 Information

- Port **3000** : gestionnaire de tournoi (API / serveur logique).<sup>1</sup>
- Port **80** : interface web et règles du jeu.
- Adresse IP du Raspberry dans son réseau : 10.42.0.1.

## 4. Modélisation formelle de la menace

Vous devez adopter le point de vue d'un **étudiant malveillant**. Trois objectifs sont considérés comme pertinents :

1. **Gagner sans respecter les règles du jeu.**
2. **Faire perdre un autre étudiant.**
3. **Interrompre ou perturber le tournoi.**

Les vulnérabilités non exploitables dans ce cadre sont hors-scope. Exemple :

- Le serveur web transmet les règles en HTTP non chiffré. S'il n'est pas possible d'exploiter cet élément pour atteindre les objectifs ci-dessus, alors cela n'est pas pertinent.

## 5. Difficultés attendues

Cet exercice est volontairement conçu pour simuler un audit réel. Ainsi :

- vous ne savez pas combien de vulnérabilités existent ;
- certaines pourront se trouver dans des couches inattendues (OS, réseau, protocoles, configuration...);
- vous n'avez pas accès au code source du gestionnaire de parties, sauf en échange d'un coût ;

---

1. Il sera sûrement nécessaire de désactiver ou de reconfigurer votre pare-feu pour pouvoir participer à une compétition

- l'interface du gestionnaire est purement réseau : il n'existe pas d'interface graphique ;
- certaines vulnérabilités requièrent des connaissances qui n'ont pas été revues dans le cadre de ce cours.

## 6. Votre mission complète

Votre travail doit consister en une **analyse de sécurité exhaustive**. Il doit inclure :

### 6.1 Recherche de vulnérabilités

Sur toutes les couches du système :

- réseau,
- API,
- OS,
- configuration du Raspberry,
- logique du gestionnaire,
- protocoles appliqués aux échanges,
- interface web,
- Et possiblement d'autres

### 6.2 Validation et démonstration

Pour chaque vulnérabilité :

- déterminer si elle est exploitable ;
- fournir une démonstration claire (capture réseau, preuve de concept, manipulation. . .) ;
- expliquer le potentiel d'exploitation ;
- documenter un scénario d'attaque complet si possible.

### 6.3 Évaluation du risque

Chaque vulnérabilité doit être accompagnée :

- d'une explication détaillée ;
- d'un score CVSS basé sur la plateforme NIST ;
- d'une appréciation du danger dans le contexte de l'exercice.

## 6.4 Proposition de correction

Pour chaque faille :

- proposer une solution réaliste et adaptée ;
- expliquer son principe ;
- préciser si elle nécessite une modification du code, de la configuration, du réseau ou du matériel.

## 7. Rapport final

Le rapport doit contenir :

- les outils utilisés ;
- les tentatives réussies ou non (justification requise) ;
- la liste exhaustive des vulnérabilités ;
- l'exploitation des vulnérabilités (PoC si applicable) ;
- une analyse de risque ;
- des contre-mesures précises ;
- des explications *sans contenu inutile*. Le style doit être clair, concis et professionnel.

Le rapport doit, au maximum, faire 10 pages (les images peuvent être mises en annexe). Deadline lundi 8/12 sur Teams à minuit.

## 8. Indices

Chaque groupe a droit à trois indices, chacun coûtant 2,5 points. Il est possible de demander le code source complet du gestionnaire, mais cela coûte 2 indices ou 5 points.

## 9. Barème et notation

Le score maximal atteignable grâce au travail est de 30 points. Cependant, seuls les 20 premiers sont retenus.

Score obtenu	Score final
12/30	12/20
25/30	20/20