

# Pen-test

## 1. Introduction

**Date du test d'intrusion :** 27/10/2024

**Client :** Gothime Manager

**Testeurs :** MAR\_10

**Objet du test :** Ce test d'intrusion vise à évaluer la sécurité du site web et de l'API associée afin d'identifier et de documenter les vulnérabilités existantes et d'orienter les recommandations de sécurité.

## 2. Résumé Exécutif

Le test d'intrusion a permis de détecter plusieurs vulnérabilités dans l'application web et l'API. Ces vulnérabilités peuvent permettre des accès non autorisés, des manipulations de données sensibles, ou une compromission du système.

Voici un résumé des résultats :

### **Vulnérabilités critiques :**

- Aucune détection de brute-force
- Informations envoyées en clairs lors des requêtes

### **Vulnérabilités majeures :**

- La base de donnée hébergée sur la même machine que le web
- Lors d'une authentification, les identifiants en clairs sont envoyées au serveur

### **Vulnérabilités modérées :**

- Aucun filtre lors du scan des QRCode

Les principales recommandations incluent l'amélioration de la validation des entrées, la sécurisation des configurations serveur, et la mise à jour des composants logiciels.

### 3. Méthodologie

Les tests ont été réalisés selon les méthodes d'analyse suivantes :

1. **Reconnaissance passive et active** : Collecte d'informations sur le site et l'API.
2. **Scan de vulnérabilités** : Analyse des vulnérabilités connues.
3. **Exploitation des vulnérabilités** : Tentative d'exploitation des faiblesses détectées.
4. **Post-exploitation** : Test de persistance et d'escalade de privilèges.
5. **Rapport et recommandations** : Documentation des vulnérabilités et recommandations de sécurité.

## 4. Détails des Vulnérabilités

## 4.1 Vulnérabilités Critiques

#### 4.1.1 Absence de détection de force brute

**Description :** L'application ne détecte pas les tentatives de force brute, permettant la récupération non autorisée des identifiants administrateurs

[illegible]

**Impact :** Risque élevé d'accès non autorisé aux comptes administrateurs et aux données sensibles.

**Recommandation :** Mettre en place un système de détection et de blocage des tentatives de force brute, ainsi qu'une politique de mots de passe robuste.

## 4.1.2 Les requêtes d'authentification sont envoyées en clairs

**Description :** Lors de l'authentification, la requête de login est envoyée en clair, avec les identifiants sans aucune obfuscation.

No.	Time	Source	Destination	Protocol	Length	Info
1786	30.193164	127.0.0.1	127.0.0.1	TCP	44	53713 → 53714 [ACK] Seq=1 Ack=37 Win=47778 Len=0
1787	30.193206	127.0.0.1	127.0.0.1	TCP	45	53714 → 53713 [PSH, ACK] Seq=37 Ack=1 Win=65535 Len=1
1788	30.193300	127.0.0.1	127.0.0.1	TCP	44	53713 → 53714 [ACK] Seq=1 Ack=38 Win=47778 Len=0
1789	30.193312	127.0.0.1	127.0.0.1	TCP	56	54414 → 8080 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
1710	30.193357	127.0.0.1	127.0.0.1	TCP	45	53714 → 53713 [PSH, ACK] Seq=38 Ack=1 Win=65535 Len=1
1711	30.193378	127.0.0.1	127.0.0.1	TCP	44	53713 → 53714 [ACK] Seq=1 Ack=39 Win=47777 Len=0
1712	30.193390	127.0.0.1	127.0.0.1	TCP	56	8080 → 54414 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
1713	30.193420	127.0.0.1	127.0.0.1	TCP	44	54414 → 8080 [ACK] Seq=1 Ack=1 Win=2161152 Len=0
1714	30.193444	127.0.0.1	127.0.0.1	TCP	45	53714 → 53713 [PSH, ACK] Seq=39 Ack=1 Win=65535 Len=1
1715	30.193557	127.0.0.1	127.0.0.1	TCP	44	53713 → 53714 [ACK] Seq=1 Ack=40 Win=47776 Len=0
1716	30.193602	127.0.0.1	127.0.0.1	TCP	45	53714 → 53713 [PSH, ACK] Seq=40 Ack=1 Win=65535 Len=1
1717	30.193617	127.0.0.1	127.0.0.1	TCP	44	53713 → 53714 [ACK] Seq=1 Ack=41 Win=47775 Len=0
1718	30.193631	127.0.0.1	127.0.0.1	HTTP/3	572	POST http://localhost:4000/api/login HTTP/3.1.1, JSON (application/json)
1719	30.193644	127.0.0.1	127.0.0.1	TCP	44	8080 → 54414 [ACK] Seq=1 Ack=534 Win=2160640 Len=0
1720	30.193702	127.0.0.1	127.0.0.1	TCP	45	53714 → 53713 [PSH, ACK] Seq=41 Ack=1 Win=65535 Len=1
1721	30.193713	127.0.0.1	127.0.0.1	TCP	44	53713 → 53714 [ACK] Seq=1 Ack=42 Win=47774 Len=0
1722	30.193731	127.0.0.1	127.0.0.1	TCP	45	53714 → 53713 [PSH, ACK] Seq=42 Ack=1 Win=65535 Len=1
1723	30.193740	127.0.0.1	127.0.0.1	TCP	44	53713 → 53714 [ACK] Seq=1 Ack=43 Win=47773 Len=0
1724	30.193802	127.0.0.1	127.0.0.1	TCP	45	53714 → 53713 [PSH, ACK] Seq=43 Ack=1 Win=65535 Len=1
1725	30.193829	127.0.0.1	127.0.0.1	TCP	44	53713 → 53714 [ACK] Seq=1 Ack=44 Win=47772 Len=0
1726	30.193860	127.0.0.1	127.0.0.1	TCP	45	53714 → 53713 [PSH, ACK] Seq=44 Ack=1 Win=65535 Len=1

```
> Frame 1718: 577 bytes on wire (4616 bits), 577 bytes captured (4616 bits) on interface \Device\NPF_{...}_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 54414, Dst Port: 8080, Seq: 1, Ack: 1, Len: 533
> Hypertext Transfer Protocol
  JavaScript Object Notation: application/json
  object
    Member: email
      [Path with value: /email:pentest@gmail.com]
      [Member with value: email:pentest@gmail.com]
      String value: pentest@gmail.com
      Key: email
      [Path: /email]
    Member: password
      [Path with value: /password:monmotdepasse]
      [Member with value: password:monmotdepasse]
      String value: monmotdepasse
      Key: password
      [Path: /password]
```

```
0000 02 00 00 00 45 00 02 3d 34 98 40 00 00 06 00 00 ... Err= 4 @-----
0010 7f 00 00 01 7f 00 00 01 d4 8e 1f 80 7e 88 1b 9c .....
0020 26 06 04 dc 50 18 20 fa 81 69 00 00 50 af 53 54 &...P...:1: POST
0030 20 68 74 74 70 3a 2f 2f 6c 6f 63 61 6c 68 6f 73 http://localhost
0040 74 3a 34 30 30 30 2f 61 70 69 2f 6c 6f 67 69 6e t:4000/api/login
0050 20 48 54 54 50 2f 31 2e 31 8d 8a 48 6f 73 74 3a HTTP/1.1: Host:
0060 20 6c 6f 63 61 6c 68 6f 73 74 3a 34 30 30 30 0d localho st:4000
0070 0a 55 73 65 72 2d 41 67 65 6e 74 3a 2a 0d 6f 7a User-Agent: Mozilla/5.0 (Windows
0080 69 6c 61 2f 35 2a 30 20 57 69 6e 64 6f 77 lla/5.0 (window
0090 73 20 4e 54 20 31 30 2e 30 3b 20 57 69 6e 36 34 s NT 10.0; Win64
00a0 30 20 70 30 34 30 20 72 76 3a 31 33 31 2e 30 20 ; 64; rv:11.0)
00b0 20 47 65 63 6b 6f 2f 32 30 31 30 30 31 30 31 20 Gecko/2.0100101
00c0 46 69 72 65 66 6f 78 2f 31 33 31 2e 30 0d 0a 41 Firefox/131.0.A
00d0 63 65 70 74 3a 20 61 70 70 6c 69 63 61 74 69 on/json, text/pl
00e0 6f 6e 2f 6a 73 6f 6e 2c 20 74 65 70 74 2f 70 6c on/json, text/pl
00f0 61 69 6e 2c 20 2a 2f 2a 0d 0a 41 63 63 65 70 74 ain, "" -Accept
0100 2d 4c 61 6e 67 75 61 67 65 3a 20 66 72 2c 65 72 -Language: fr,fr
0110 2d 46 52 3b 71 3d 30 2e 30 2c 65 6e 2d 55 53 3b -Fgq=0.8,en-US;
0120 71 3d 30 2e 35 2c 65 6e 30 71 3d 30 2e 33 0d 0a q=0.5,en;q=0.3
0130 41 63 65 70 74 2d 45 6e 63 6f 64 69 6e 67 3a -Accept-encoding:
0140 20 67 7a 69 70 2c 20 64 65 66 6c 61 74 65 0d 0a gip, deflate
0150 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 61 70 Content-Type: ap
0160 70 6c 69 63 61 74 69 6f 6e 2f 6a 73 6f 6e 0d 0a plication/json
0170 41 75 74 68 6f 72 69 7a 61 74 69 6f 6e 3a 20 42 Authorization: B
0180 65 61 72 65 72 20 6e 75 6c 6c 0d 0a 43 6f 6e 74 earer null: Cont
0190 65 6e 74 2d 4c 65 6e 67 74 68 3a 20 35 36 0d 0a ent-length: 56
01a0 4f 72 69 67 69 6e 3a 20 68 74 74 70 3a 2f 2f 6c Origin: http://l
01b0 6f 63 61 6c 68 6f 73 74 3a 35 31 37 33 0d 0a 43 ocalhost:5173-C
01c0 6f 6e 65 63 74 69 6f 6e 3a 20 65 65 65 70 2d 0nnection: keep
01d0 61 6c 69 76 65 0d 0a 52 65 66 65 72 65 72 3a 20 alive: R eferer:
01e0 68 74 74 70 3a 2f 2f 6c 6f 63 61 6c 68 6f 73 74 http://localhost
01f0 3a 35 31 37 33 2f 0d 0a 50 72 69 6f 72 69 74 79 :5173/ - Priority
0200 3a 20 75 3d 30 0d 0a 0d 0a 70 22 65 6d 61 69 6c : u=0... ("email
0210 22 3a 22 70 65 6e 74 65 73 74 40 67 6d 61 69 6c : "pentest@gmail
0220 2e 63 6f 6d 22 2c 22 70 61 73 73 77 6f 72 6d 22 : "cm", "password"
0230 3a 22 6d 6f 6e 6d 6f 74 64 65 70 61 73 73 65 22 : "monmot depasse"
0240 70 }
```

**Impact :** Risque élevé de sniffing du réseau et de récupération des identifiants.

**Recommandation :** Mettre en place un système directement dans le front-end qui obfusque ( ou hashage ) les identifiants avant de les envoyées au serveur web.

## 4.1.2 Authentification Insecure

**Description :** Le mécanisme d'authentification de l'API permet la réutilisation des jetons expirés, ce qui peut permettre un accès non autorisé.

**Impact :** Risque d'accès non autorisé aux comptes d'utilisateurs et aux données sensibles.

**Recommandation :** Utiliser des jetons avec un délai d'expiration court et activer le renouvellement sécurisé des jetons

## 4.2 Vulnérabilités Majeures

### 4.2.1 Base de données hébergée sur la même machine que le serveur web

**Description :** La base de données et le serveur web sont hébergés sur la même machine, ce qui compromet la sécurité, l'isolation des données, mais compromet aussi la performance et la disponibilité de l'application web.

```
(kali㉿kali)-[~]
└─$ nmap 192.168.1.5 -sV -Pn
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-27 11:3
Nmap scan report for xps (192.168.1.5)
Host is up (0.014s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
4000/tcp  open  http-proxy  Ncat http proxy (Nmap 4.85BETA1 or
5432/tcp  open  postgresql  PostgreSQL DB 9.6.0 or later
8081/tcp  open  http        PHP cli server 5.5 or later (PHP 7.
SF-Port5432-TCP:V=7.94SVN%I=7%D=10/27%Time=671E5D96%P=x86_64-
SF:r(SMBProgNeg,90,"E\0\0\0\x8fSFATAL\0VFATAL\0C0A000\0MunSUP
SF:ntend\x20protocol\x2065363\0.19778:\x20server\x20supports\x
SF:x203\0\0Fbackend_startup\0.c\0L679\0RProcessStartupPacket\

Nmap done: 1 IP address (1 host up) scanned in 180.26 seconds
```

**Impact :** Risque accru de compromission totale du système en cas d'intrusion.

**Recommandation :** Séparer physiquement la base de données du serveur web et mettre en place un réseau segmenté pour isoler les composants critiques.

## 4.3 Vulnérabilités Modérées

### 4.3.1 Absence de filtrage lors du scan du QRCode

**Description :** L'application ne filtre pas les données lors du scan des QR codes, ce qui peut permettre l'injection de contenu malveillant.

**Impact :** Risque d'exécution de code arbitraire ou d'accès non autorisé aux données.

**Recommandation :** Implémenter une validation stricte des données issues des QR codes et limiter les actions possibles après un scan.

## 5. Recommandations Générales

- **Renforcer l'authentification** : Activer une authentification multi-facteurs pour les comptes utilisateurs.
- **Chiffrer les données sensibles** : Utiliser HTTPS et chiffrer les données sensibles lors de leur transport ( requête client → serveur & serveur → client )
- **Surveillance et Alertes** : Mettre en place un système de détection d'intrusion pour surveiller les comportements anormaux et détecter rapidement toute tentative d'intrusion.

## 6. Conclusion

Le test d'intrusion a révélé plusieurs faiblesses de sécurité qui nécessitent une action immédiate pour protéger les données des utilisateurs et maintenir la sécurité de l'application. Les correctifs recommandés devraient être appliqués rapidement, et un suivi de sécurité périodique est conseillé pour détecter d'éventuelles nouvelles vulnérabilités.

## 7. Annexes

### Liste des outils utilisés

- Burp Suite
- Wappalazer
- OWASP ZAP
- Nmap
- Dirbuster & Gobuster
- Metasploit