## Network Configurations

1. Open the configuration file for the virtual machine using a text editor
2. Modify line 44 of the configuration file to become *ethernet0.networkName = "nat"*

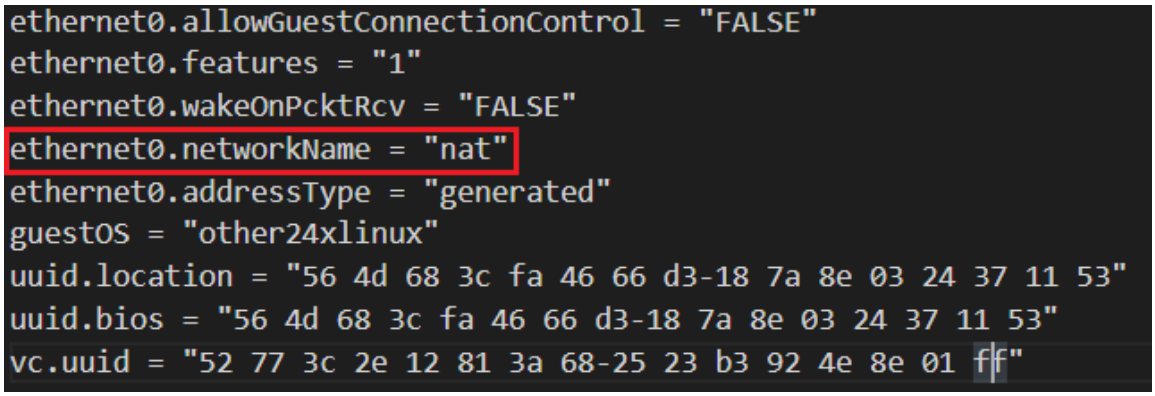


*Figure 1. Modifying configuration file for virtual machine*

## IP Discovery

1. We must find the IP address of the vulnerable target machine using *netdiscover*
2. The IP address of the vulnerable target machine is *192.168.44.128.*

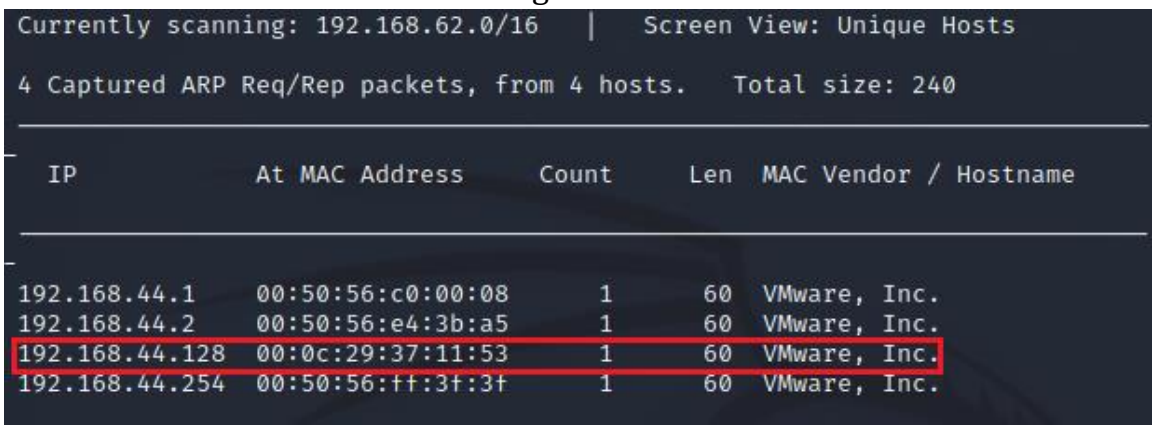


*Figure 2. Finding IP address using netdiscover*

## Port Scan

1. We will conduct a scan of all ports and service version on the IP address that we have found previously using *Nmap*. (The command that was used is *nmap -sv -T4 -p- 192.168.44.128 -vv)*
2. The more notable ports that we have discovered are ports 22,80,139 and 443.

```
PORT     STATE SERVICE     REASON  VERSION
22/tcp   open  ssh         syn-ack OpenSSH 2.9p2 (protocol 1.99)
80/tcp   open  http        syn-ack Apache httpd 1.3.20 ((Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b)
111/tcp  open  rpcbind     syn-ack 2 (RPC #100000)
139/tcp  open  netbios-ssn syn-ack Samba smbd (workgroup: TMYGROUP)
443/tcp  open  ssl/https   syn-ack Apache/1.3.20 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
1024/tcp open  status      syn-ack 1 (RPC #100024)
```

*Figure 3. Ports found from nmap scanning*

## SMB Scan

1. We use *Metasploit* to scan the version of *Samba* that port 139 is using.
2. The *Samba* version that is used is 2.2.1a

```
msf6 > use auxiliary/scanner/smb/smb_version
msf6 auxiliary(scanner/smb/smb_version) > set rhosts 192.168.44.128
rhosts ⇒ 192.168.44.128
msf6 auxiliary(scanner/smb/smb_version) > run

[*] 192.168.44.128:139    - SMB Detected (versions:) (preferred dialect:) (signatures:optional)
[*] 192.168.44.128:139    -   Host could not be identified: Unix (Samba 2.2.1a)
[*] 192.168.44.128:       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

*Figure 4. Scanning for version of Samba used using msf*

## Web Scan

1. We use *gobuster* to enumerate the directories of the *Apache* website that we have found using *nmap* earlier.
2. The results *index.html* and *mrtg* looks the most promising.

```
2021/06/06 09:40:29 Starting gobuster in directory enumeration mode
===============================================================
/cgi-bin/             (Status: 403) [Size: 272]
/index.html           (Status: 200) [Size: 2890]
/icons/               (Status: 200) [Size: 9472]
/doc/                 (Status: 403) [Size: 268]
/test.php             (Status: 200) [Size: 27]
/manual/              (Status: 200) [Size: 643]
/usage/               (Status: 200) [Size: 4279]
/mrtg/                (Status: 200) [Size: 17318]

===============================================================
2021/06/06 09:42:39 Finished
```

*Figure 5. Directory enumeration using gobuster*

## Exploiting SMB vulnerability

1. We will first search for a vulnerability related to *Samba 2.2.1a* using *searchsploit*
2. It seems that we found an RCE attack related to *Samba 2.2.1a*

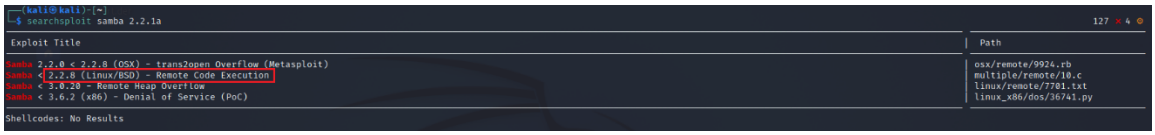


*Figure 6. Finding for existing vulnerabilities*

3. We will now copy the exploit to a working folder and compile the exploit.


*Figure 7. Copying and compiling the exploit*

4. Afterwards, we will execute the exploit and obtain root access.

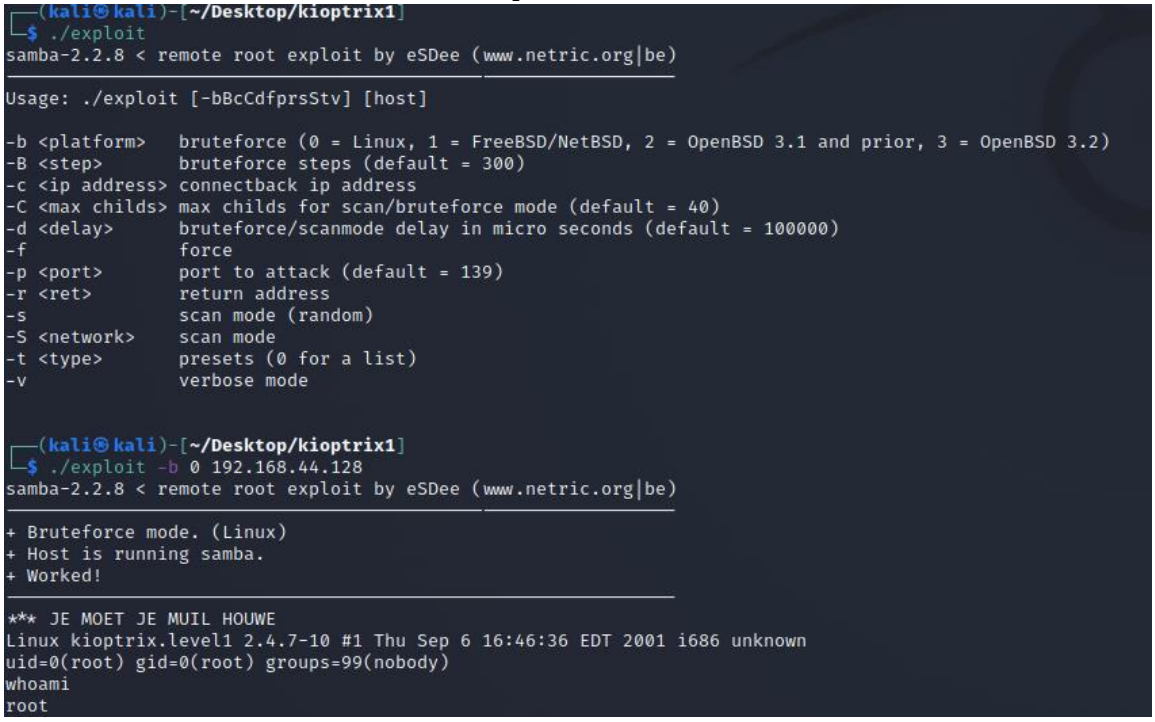


*Figure 8. Obtaining root access*

## Exploiting Apache/mod_ssl vulnerability

1. We will look at the *test.php* file that was found during directory enumeration using *gobuster.* However, it does not yield any potential findings.

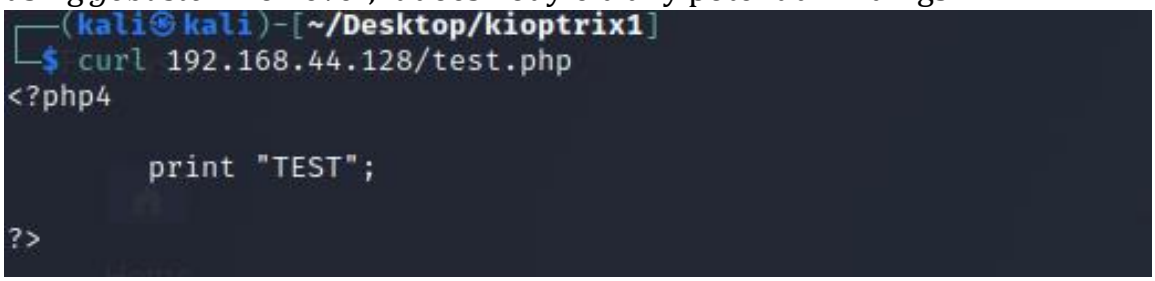


*Figure 9. Findings from test.php file*

2. We will try to find for exploits for the *mrtg* module. However, there are no potential exploits found for it.

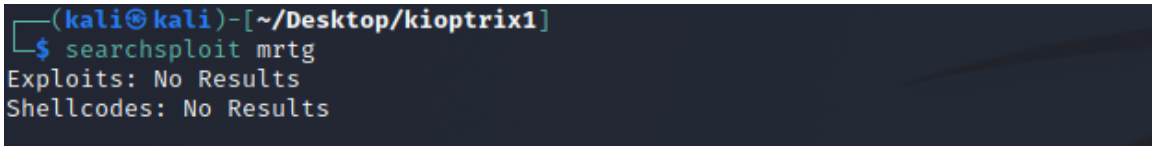


*Figure 10. Findings for mrtg module*

3. However, we notice that ports 80 and 443 uses *Apache and mod_ssl* from our port scans above.
4. We will search for exploits relating to *Apache* and *mod_ssl.* It seems that we have found a remote buffer overflow attack for this scenario that can potentially give us root access to the vulnerable machine.


*Figure 11. Looking for exploits for Apache and mod_ssl*

5. We will now copy the exploit to a working folder, modify the exploit code accordingly and compile the exploit.


*Figure 12. Compiling the exploit*

6. We will now execute the exploit

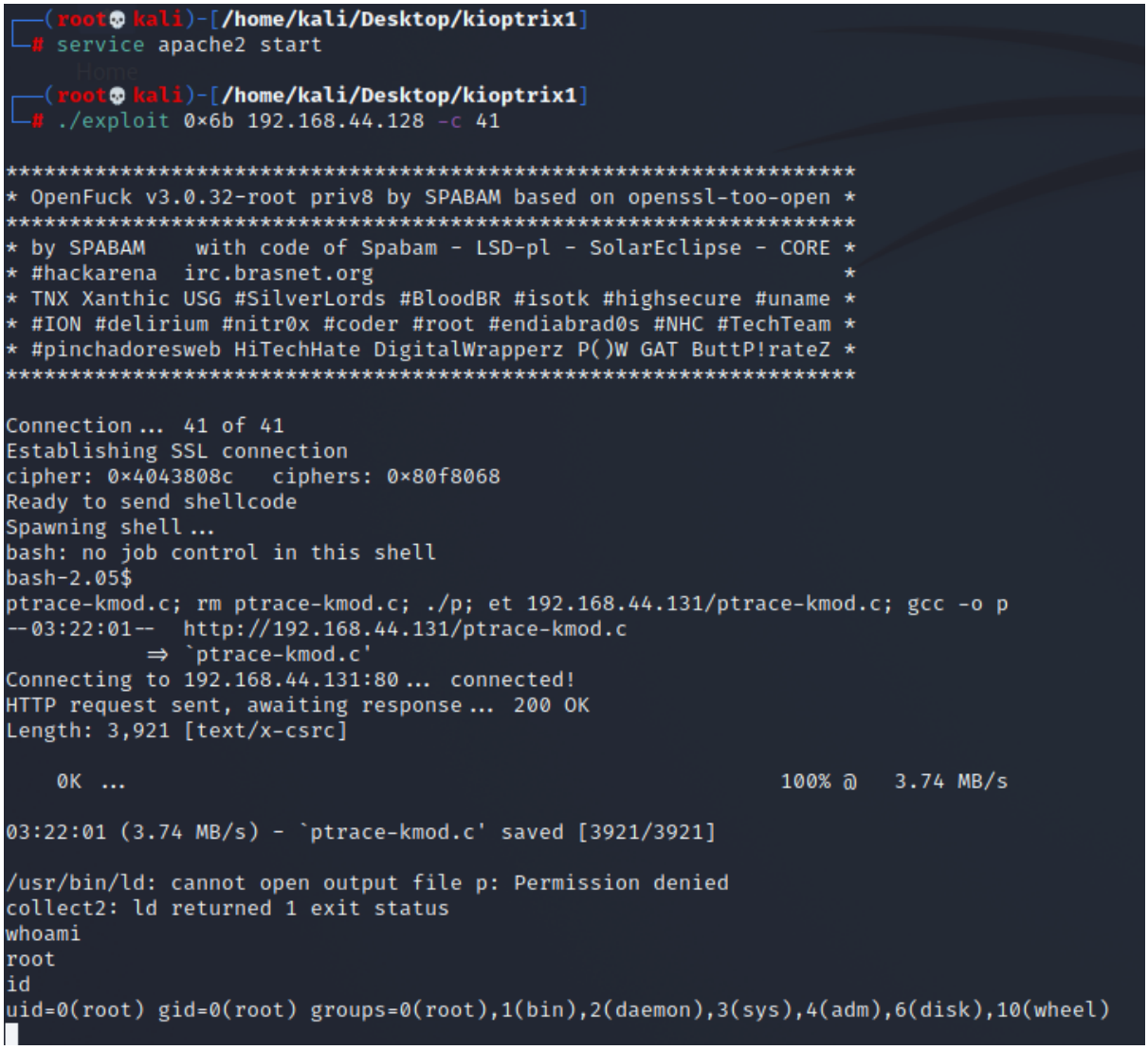


*Figure 13. Executing the exploit*