Lab - Creating a BASH Script for Scanning Vulnerable Ports

Overview

No matter how you slice it, knowledge is king when it comes to being a threat either as a cybersecurity professional or a hacker. You’re not going to fake it and no matter how much you think you know; someone else always knows more. We’ve all read stories of how a hacker managed to steal credit card numbers by the tens of thousands but often we don’t know enough about Linux and the black art of hacking to appreciate how they did it.

Hopefully, this second lab in the BASH scripting series will provide enough insight of how one hacker using Nmap and simple BASH script caused $86 million dollars of havoc for the credit card companies.

As pentesters and IT auditors, we come into contact we many different vendors and their programs. To allow remote access to these programs, software designers use their coding skills to leave a back door open for accessing the program remotely either for updates or for tech support. Nearly all business-grade programs have the means for remote access using a not so well-known port.

There will be times when you'll find yourself in a position knowing a port represents a serious vulnerability to the application or service. Often time the vulnerability is known but ignored for the sake of convenience.

The Saga of Max Vision

Max Bulter, aka Max Vision, found such a vulnerability in the Aloha Point-of-Sale (POS) system while servicing a client. The software vendor had a remote backdoor created their systems to better provide technical assistance to their customers.

Aloha systems provide POS’s for small-to-medium sized restaurants taking in thousands of credit card numbers each year. Learning of the built-in backdoor left by Aloha for providing customer tech support, Max created a simple but effective BASH script to scan the U.S. for systems with port 5505 open constantly.

Since Aloha systems are the only vendor using this port, finding an IP address using this port would indicate there was an Aloha POS system up and running somewhere in the U.S. using port 5505.

Once discovered MAX would execute an exploit against port 5505 and its service scavenging what credit card numbers he could find. MAX would than post the credit numbers for sale on his Carders Market forum to be sold to cybercriminals. Max sold over 2 million credit cards which racked up more than $86 million in fraudulent credit card charges.

SYBEROFFENSE

Lab - Creating a BASH Script for Scanning Vulnerable Ports

In this lab, we will create a BASH script for scanning for a specific vulnerable port and then generate a report showing what IP address has the vulnerable service running. The only difference is we'll play nice and not go looking for any credit card information.

Hardware Requirements

- One updated virtual install of Kali Linux

Part 1 – Build the script….

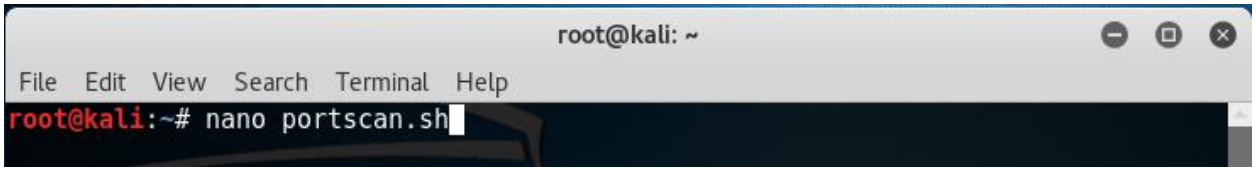
To create our script, we need a text editor. Any Linux text editors will work. In this lab, we'll use the nano editor built into Kali. You are free to use any text editor of your choice.

Step 1. Open a Terminal in Kali

Open you VM of Kali and then open a terminal. At the terminal type.

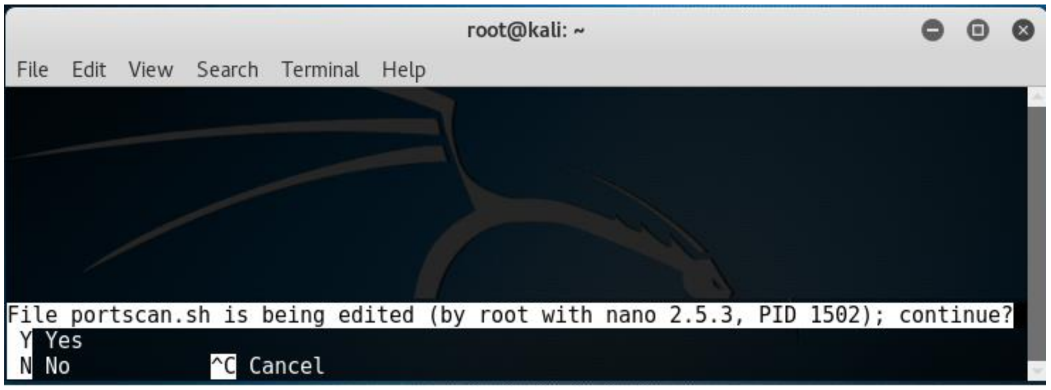
```
nano portscan.sh
```

You can name your script anything, in this example we to call it **portscan.sh**. With the following command we are saying open nano, create a file and name it portscan.sh.



```
root@kali:~# nano portscan.sh
```

This opens a blank file using the nano file editor for creating our script. Select Y for yes to begin the edit.



```
File portscan.sh is being edited (by root with nano 2.5.3, PID 1502); continue?
 Y Yes
 N No           ^C Cancel
```

Step 2 create the script

Type the following lines into our nano script file.

```
#!/bin/bash
```

The required opening for all BASH scripts.

```
nmap -sT 216.58.194.0/24 -p 5505 -oG aloha
```

Does an nmap connect scan (-sT) to the subnet of google.com and looks for the port 5505 open and sends the output (-oG) to a file called aloha.

```
cat aloha | grep open > alohaopen
```

Opens the file aloha and filters (grep) for lines that say open and stores those lines in a file called alohaopen.

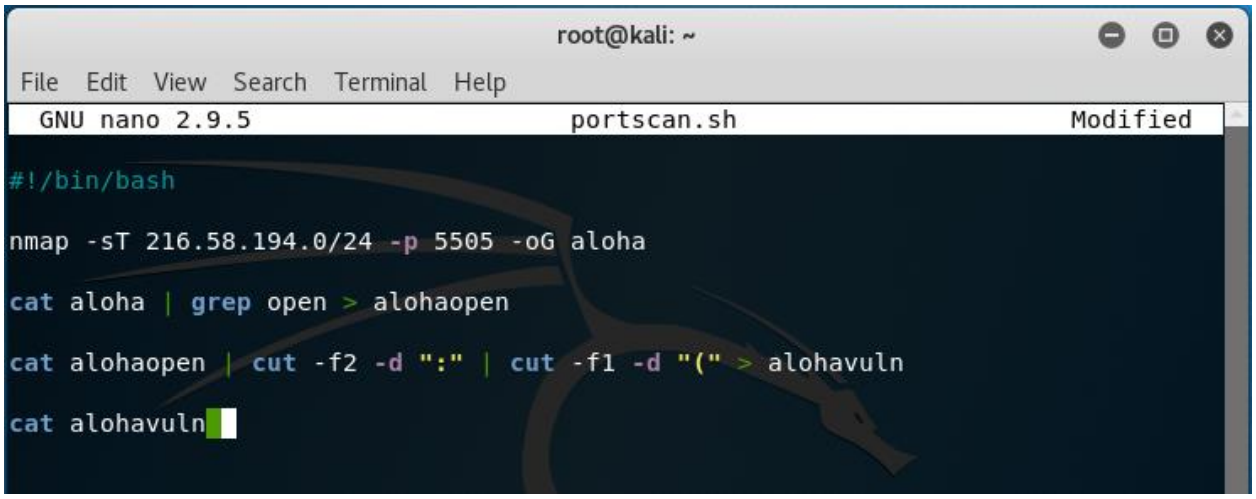
```
cat alohaopen | cut -f2 -d ":" | cut -f1 -d "(" > alohavuln
```

Opens the file alohaopen and cuts it at the second field (-f2) defined by the delimiter (-d) semicolon (":"), then pipes that to a second cut command that cuts the file at the first field (-f1) defined by the delimiter (-d) paren ("(") and saves it into a file named alohavuln.

```
cat alohavuln
```

Finally, we open and display the file that contains all the IP addresses of systems with port 5505 open.

Here is what we have done so far.

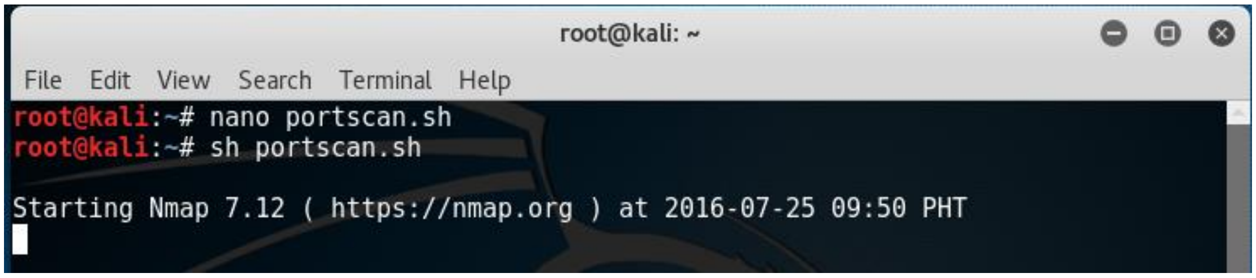


```
  GNU nano 2.9.5                   portscan.sh                        Modified

#!/bin/bash

nmap -sT 216.58.194.0/24 -p 5505 -oG aloha

cat aloha | grep open > alohaopen

cat alohaopen | cut -f2 -d ":" | cut -f1 -d "(" > alohavuln

cat alohavuln
```

We now need to save the script. Do a CTRL+X to save the file. Select Y and then hit enter. We are now back at the terminal prompt and ready to run the script.

Step 3 run the Script

```
sh portscan.sh
```

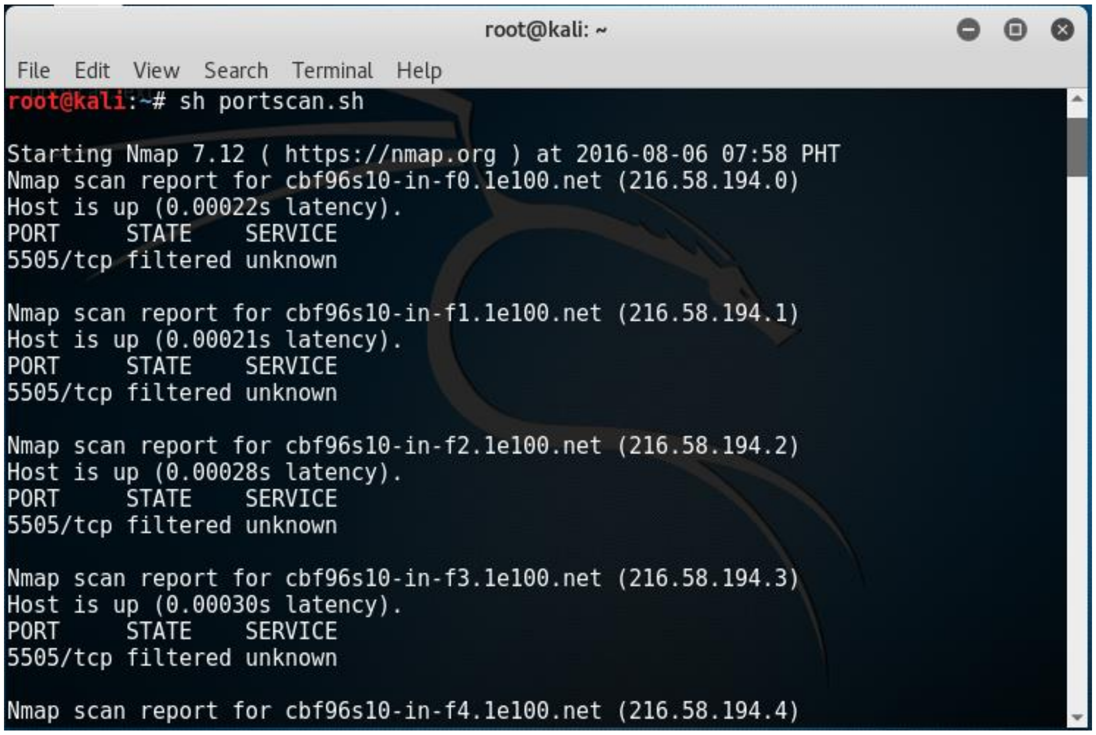


```
root@kali:~# nano portscan.sh
root@kali:~# sh portscan.sh

Starting Nmap 7.12 ( https://nmap.org ) at 2016-07-25 09:50 PHT
```

Allow the scan to complete. In our example, we're only scanning 255 addresses, so should only take a few minutes. In this example, we used the IP range used by Google.com, but it could have been any IP range.

A real-world scan conducted by Max Vision would have millions of addresses in which he might wait days, perhaps weeks for the results.

Step 4 look at the results



```
root@kali:~# sh portscan.sh

Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-06 07:58 PHT
Nmap scan report for cbf96s10-in-f0.1e100.net (216.58.194.0)
Host is up (0.00022s latency).
PORT     STATE    SERVICE
5505/tcp filtered unknown

Nmap scan report for cbf96s10-in-f1.1e100.net (216.58.194.1)
Host is up (0.00021s latency).
PORT     STATE    SERVICE
5505/tcp filtered unknown

Nmap scan report for cbf96s10-in-f2.1e100.net (216.58.194.2)
Host is up (0.00028s latency).
PORT     STATE    SERVICE
5505/tcp filtered unknown

Nmap scan report for cbf96s10-in-f3.1e100.net (216.58.194.3)
Host is up (0.00030s latency).
PORT     STATE    SERVICE
5505/tcp filtered unknown

Nmap scan report for cbf96s10-in-f4.1e100.net (216.58.194.4)
```

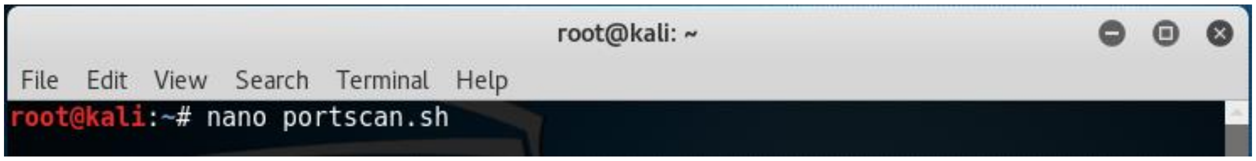
Part 2 – Modify the Script

With a slight modification, we could easily create a time-saving BASH script to speed up the process of scanning a client's network IP address range searching for a port and service. Let's see how we do this.

We now see how easy it was for Max Vision to get a hold of 2 million credit card numbers using just a basic BASH file script to run a Nmap scan looking for a specific port. Let's see how we can turn Max's basic script into a more advance useful script for pentesting a network.

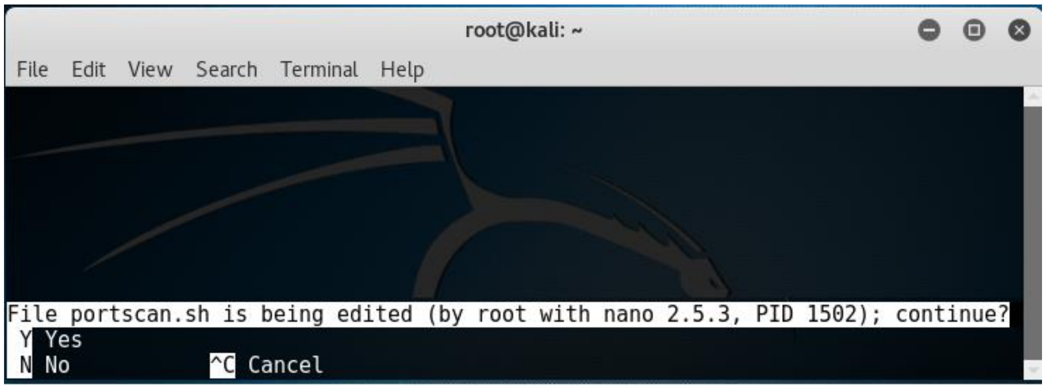
Step 1 Adding in text boxes and variables

Open a new terminal and open your postscan.sh script for editing using nano.



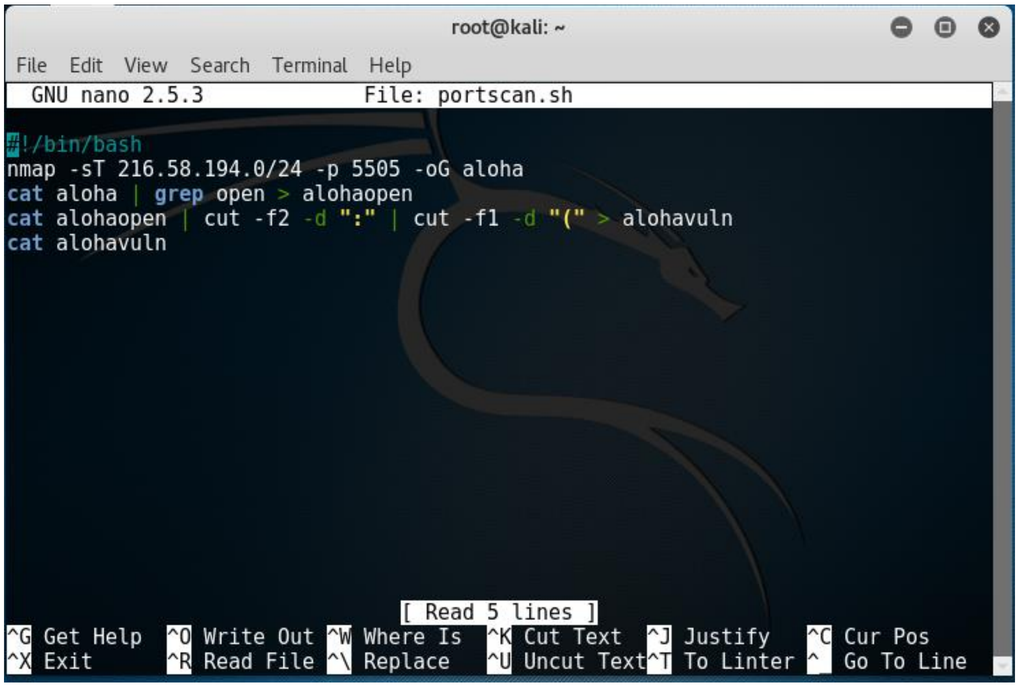
```
root@kali:~# nano portscan.sh
```

Select Y to continue with the editing.



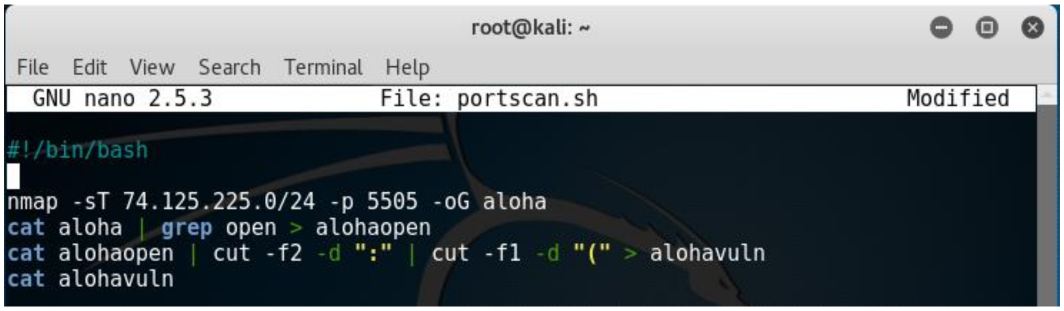
```
File portscan.sh is being edited (by root with nano 2.5.3, PID 1502); continue?
 Y Yes
 N No             ^C Cancel
```

Your portscan.sh file is now ready for editing.



```
  GNU nano 2.5.3              File: portscan.sh

#!/bin/bash
nmap -sT 216.58.194.0/24 -p 5505 -oG aloha
cat aloha | grep open > alohaopen
cat alohaopen | cut -f2 -d ":" | cut -f1 -d "(" > alohavuln
cat alohavuln

                                [ Read 5 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Linter ^_ Go To Line
```

Use your keyboard arrows to move the prompt up and down and back and forth as needed. Move the cursor to the beginning of the top line and hit enter.



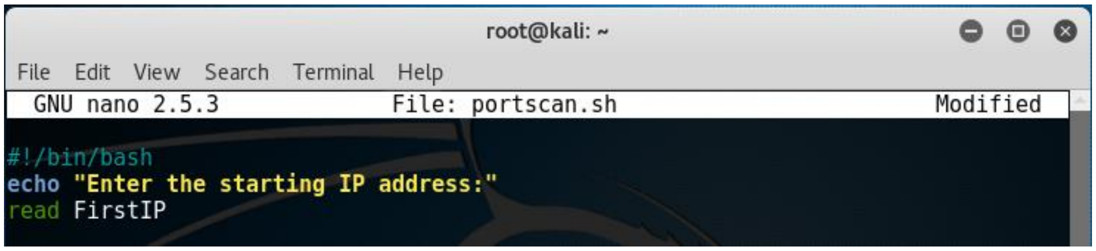
```
  GNU nano 2.5.3              File: portscan.sh                        Modified

#!/bin/bash

nmap -sT 74.125.225.0/24 -p 5505 -oG aloha
cat aloha | grep open > alohaopen
cat alohaopen | cut -f2 -d ":" | cut -f1 -d "(" > alohavuln
cat alohavuln
```

We are now going to remove the specified IP range and specific port with three variables called "FirstIP" "LastIP" and "port." You can name the variables as you please, but they should be easily associated with what it is your trying to accomplish within the script.

For the variables to work, we need to be prompted for the three values. To do this, we need to use the echo command for the prompt and the read command for the variables to work.

Type in `echo "Enter the starting IP address:"` The user will now be prompted for the first ip address. The command read "FirstIP" replace the variable "FirstIP" with this information.

So far here's what we got:



```
  GNU nano 2.5.3              File: portscan.sh                     Modified

#!/bin/bash
echo "Enter the starting IP address:"
read FirstIP
```

Create an echo command for the entering the last IP address, and a read command for the variable 'LastIP.'

```
echo "Enter the Last IP address:"
```

```
read LastIP
```

Next, we create a variable called 'LastIPOctet,' which it is equal to the value after the third period in the last IP address. When a user enters 192.168.1.255 into LastIP, LastIPOctet is equal to 255. That value is then entered into the nmap command, as 192.168.1.1-255 thus scanning the entire range specified by the user.

```
LastIPOctet=${LastIP##*.}
```

On the next line, Create an echo command for the '$LastIPOctet'

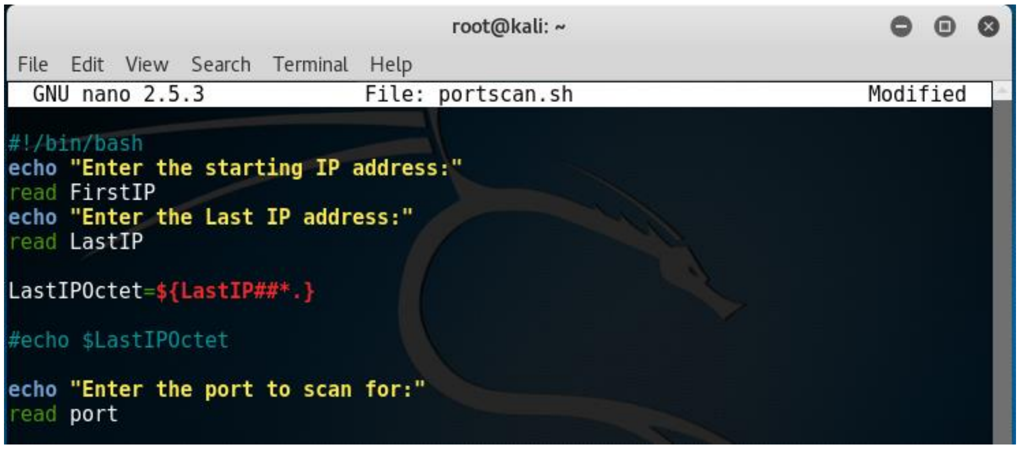
```
#echo $LastIPOctet
```

Create an echo command for the inputting the port number along with a read command for this variable.

```
echo "Enter the port to scan for:"
```

```
read port
```

Here is what we have so far:

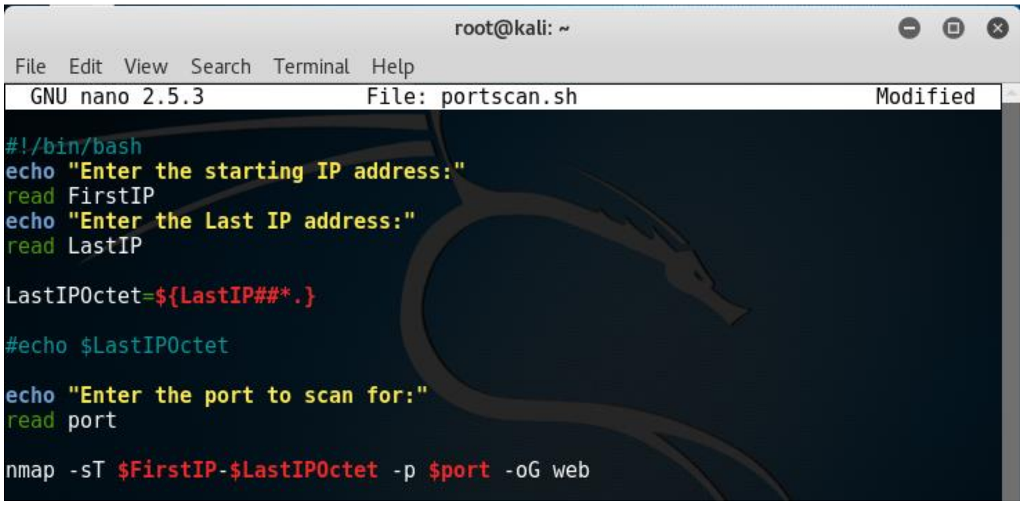

root@kali: ~
File Edit View Search Terminal Help
GNU nano 2.5.3 File: portscan.sh Modified

```
#!/bin/bash
echo "Enter the starting IP address:"
read FirstIP
echo "Enter the Last IP address:"
read LastIP

LastIPOctet=${LastIP##*.}

#echo $LastIPOctet

echo "Enter the port to scan for:"
read port
```

Next, edit the Nmap command in our script to use the variables we just created and filled. When we want the value stored in the variable, we can simply preface the variable name with the $, such as $port. So, to use Nmap to scan a range of IP addresses starting with the first user input IP through the second user input IP and look for a port input by the user, we can re-write the Nmap command like this:

```
nmap -sT $FirstIP-$LastIPOctet -p $port -oG web
```

So far, we have this:


root@kali: ~
File Edit View Search Terminal Help
GNU nano 2.5.3 File: portscan.sh Modified

```
#!/bin/bash
echo "Enter the starting IP address:"
read FirstIP
echo "Enter the Last IP address:"
read LastIP

LastIPOctet=${LastIP##*.}

#echo $LastIPOctet

echo "Enter the port to scan for:"
read port

nmap -sT $FirstIP-$LastIPOctet -p $port -oG web
```

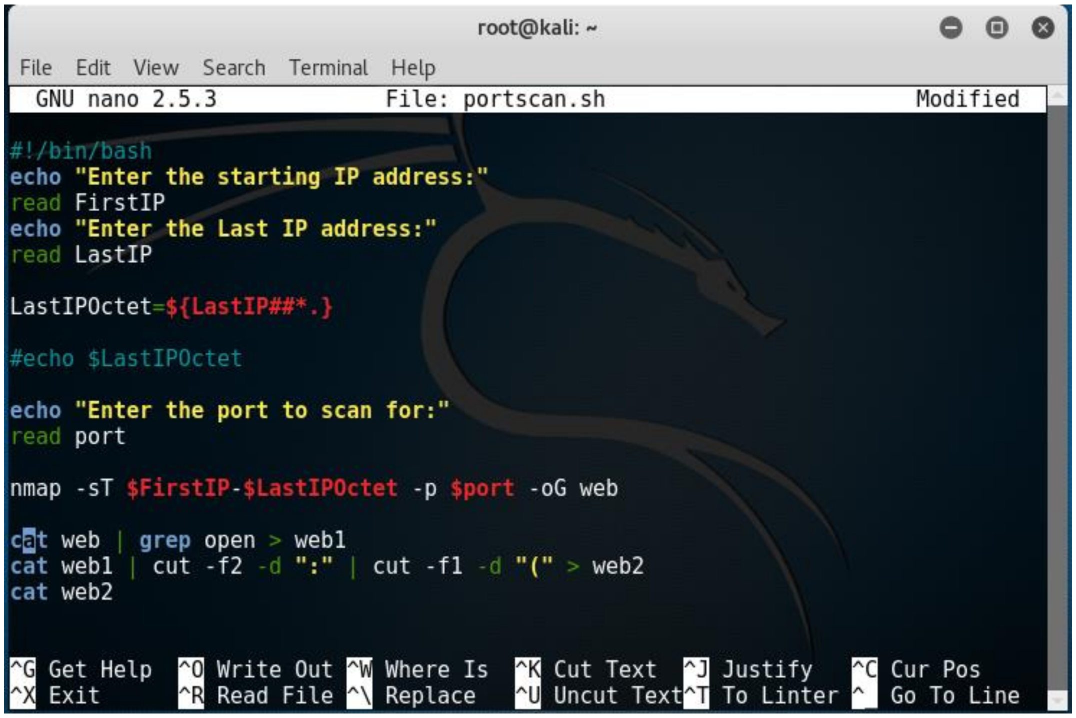
We next modify the `cat grep` commands to filter and let us see the contents of our scan.

On the next line type: `cat web | grep open > web1` Hit enter.

On the next line type: `cat web1 | cut -f2 -d ":" | cut -f1 -d "(" > web2`
Hit Enter

On the next line type: `cat web2`

So finally, we have this:



```
  GNU nano 2.5.3              File: portscan.sh                    Modified

#!/bin/bash
echo "Enter the starting IP address:"
read FirstIP
echo "Enter the Last IP address:"
read LastIP

LastIPOctet=${LastIP##*.}

#echo $LastIPOctet

echo "Enter the port to scan for:"
read port

nmap -sT $FirstIP-$LastIPOctet -p $port -oG web

cat web | grep open > web1
cat web1 | cut -f2 -d ":" | cut -f1 -d "(" > web2
cat web2

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Linter ^_ Go To Line
```

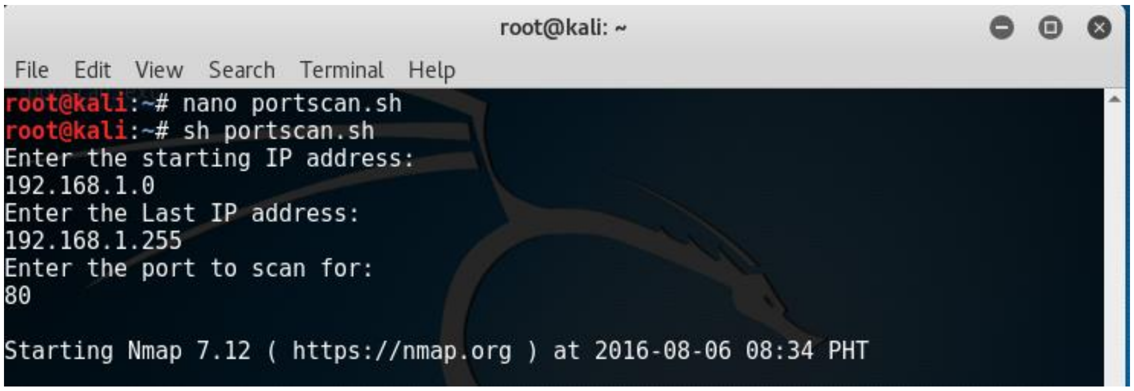
We can save the script and is all is well with its contents we should be prompted for three pieces of information and read out of the scan results.

Save the file (CTRL+X), run the script.

For testing purposes, I scanned my home network of 192.168.1.0 looking for port 80. You can do an IPCONFIG from your Windows or MAC host or IFCONFIG from Kali to find the first three octets of your home network. Normally, the private IP assigned to any LAN using a class C subnet would be 192.168.0.0 or 192.168.1.0 but your network IP could be something different.

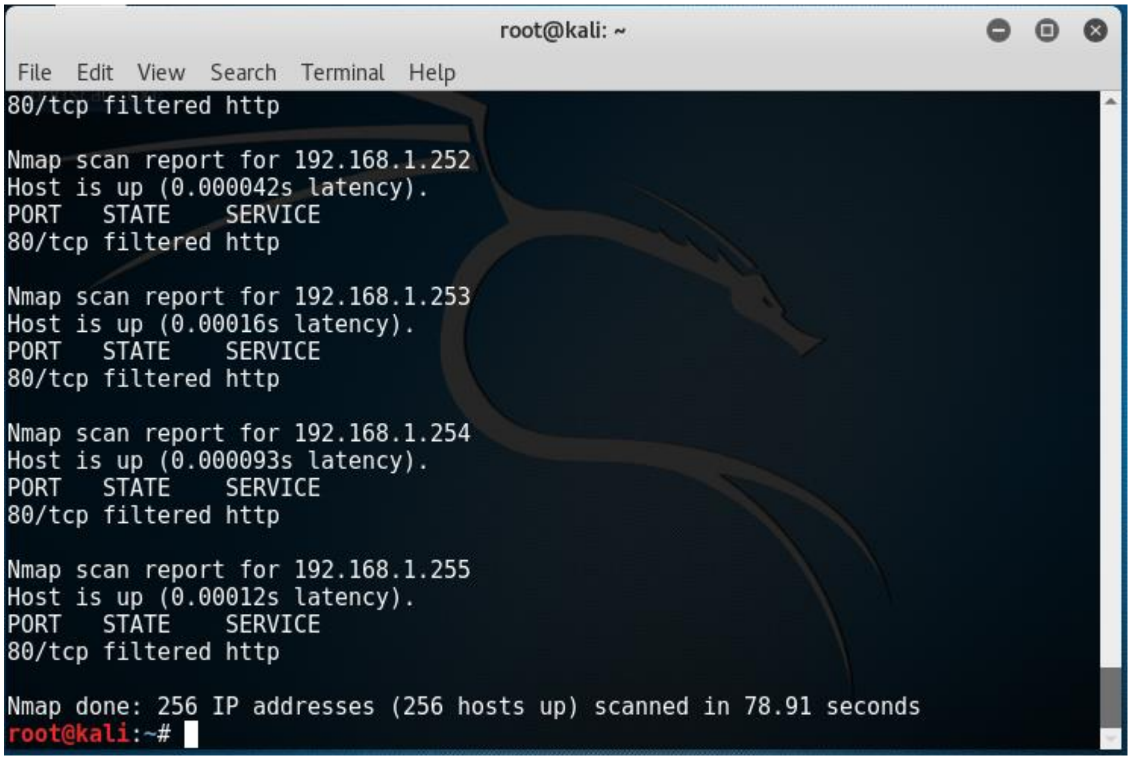
I run the script, and I am prompted for the three pieces of information and script to scan my network for all 256 IP address looking for port 80.

SYBEROFFENSE



```
root@kali:~# nano portscan.sh
root@kali:~# sh portscan.sh
Enter the starting IP address:
192.168.1.0
Enter the Last IP address:
192.168.1.255
Enter the port to scan for:
80

Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-06 08:34 PHT
```

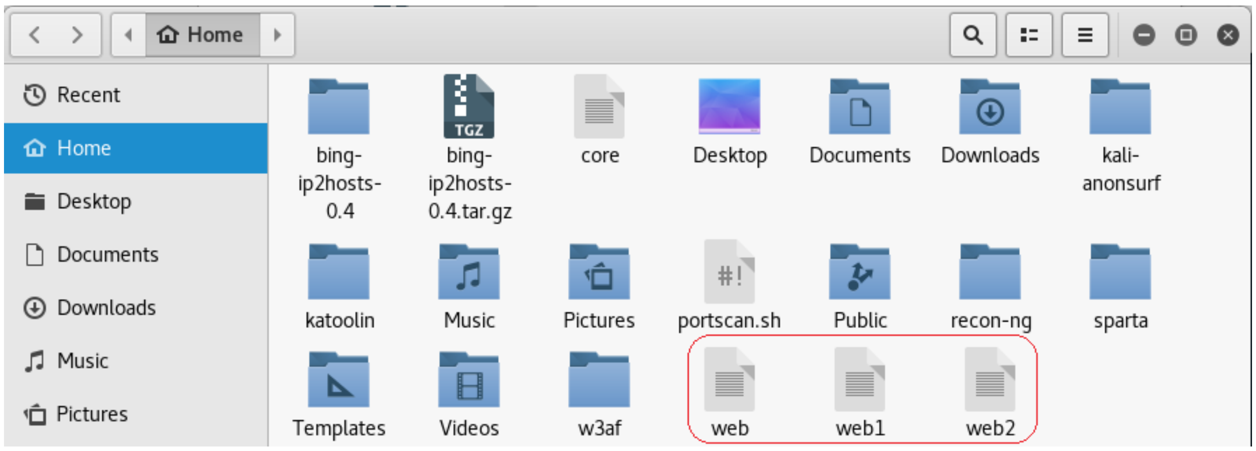
The script takes a few minutes to run and when it completes the filtered results are posted to my terminal screen.



```
80/tcp filtered http

Nmap scan report for 192.168.1.252
Host is up (0.000042s latency).
PORT   STATE    SERVICE
80/tcp filtered http

Nmap scan report for 192.168.1.253
Host is up (0.00016s latency).
PORT   STATE    SERVICE
80/tcp filtered http

Nmap scan report for 192.168.1.254
Host is up (0.000093s latency).
PORT   STATE    SERVICE
80/tcp filtered http

Nmap scan report for 192.168.1.255
Host is up (0.00012s latency).
PORT   STATE    SERVICE
80/tcp filtered http

Nmap done: 256 IP addresses (256 hosts up) scanned in 78.91 seconds
root@kali:~#
```

We generated three output files with this script that can be found in our home directory.

SYBEROFFENSE



The first web file will contain scan results. Since the scan found no IP running port 80, the two additional files, web1 and web2 will be empty. To see scan results in both web1 and web2, run Metasploitable2 on the same network as Kali and run the scan a second time.

If you would like to take the prompts to the next level, you can have them appear as a text dialog box, and if you use the same IP range every week, you can have a radio button next to the IP arrange and a manual option.

End of the lab!