

Lab 1

Build, Maintain and Test Network and Workstation Firewalls

Install, maintain and test a firewall configuration to protect cardholder data

Summary:

Firewalls control the transmission of data between an organization's trusted internal networks and untrusted external networks, as well as traffic between sensitive areas of the internal networks themselves. Requirement 1 of the PCI DSS requires systems to use firewalls to prevent unauthorized access. Where other system components provide the functionality of a firewall, they must also be included in the scope and assessment of this requirement.

| | |
|-----------------------------|---|
| Summary: | 1 |
| Lab Setup: | 2 |
| PCI_DSS Mapping: | 2 |
| NMap | 3 |
| Nmap Potential Port states: | 3 |
| Open | 3 |
| Closed | 3 |
| Filtered | 3 |
| Audit HowTo: | 4 |
| Running the Scan | 4 |
| Interpreting Results: | 5 |
| ToDo: | 6 |

Lab Setup:

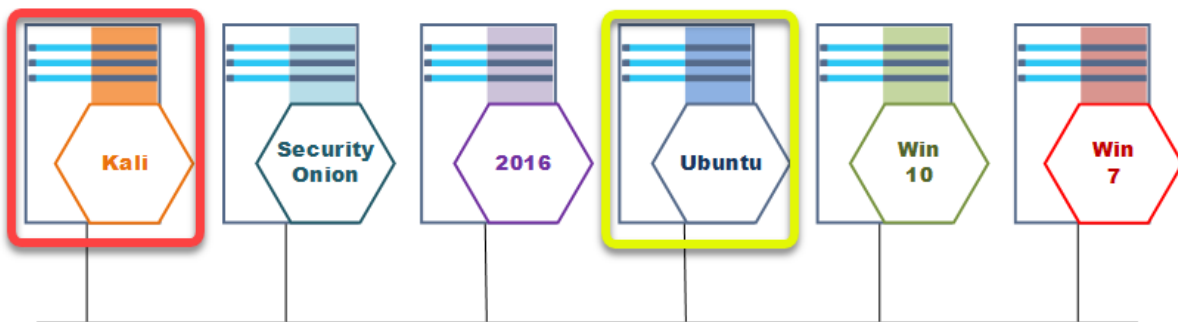
We will be using “nmap” from the Kali box to audit and confirm that our corporate firewall policy is being followed. We will primarily be working with the Ubuntu box as the target of the test but the same techniques apply to other OS.

PCI_DSS Mapping:

1.4 Install personal firewall software on any mobile and/or employee-owned devices that connect to the Internet when outside the network (for example, laptops used by employees), and which are also used to access the network.

Firewall configurations include:

- Specific configuration settings are defined for personal firewall software
- Personal firewall software is actively running
- Personal firewall software is not alterable by users of mobile and/or employee-owned devices.



Network: 192.168.14.0/24

NMap

The main purpose of a firewall is to control the ports that are available to connect to from the network. Each machine depending on what services it provides may have a different policy defining which ports should be open and available for connection.

The best way to test if a devices firewall is properly configured is to directly interact with it using a port scanner. In this case we will be using the open source tool “NMap” to perform our audit. Nmap sends connection requests to each port and based on the reply (or lack of it) decides the state of the port.

Nmap Potential Port states:

Note: Use “--reason” switch to make nmap list how it classified each port.

Open

- Port is accepting connections
- TCP: SYN/ACK
- UDP: [udp-response] The application protocol answers back

Closed

- Target answers request but port is closed
- TCP: RST
- UDP: [ICMP] port-unreachable

Filtered

- Target does not answer if the port is closed
- TCP: [no-response] SYN sent but not RST sent back
- Slows down scan big time

Audit HowTo:

Running the Scan

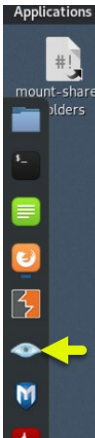
Nmap has many settings to customize the port scanning process. We only really need a few for the job we are trying to accomplish.

- -sS = TCP Syn scan
- -T4 = Set timing aggressive as we are not trying to avoid detection.
- -p- = Scan all 65535 ports
- -n = Do not do a reverse DNS lookup of target (speeds up scan)
- --reason = Displays the reason NMap indicated the port state it decided.
- --script=banner = If there is a banner available on the port, grab it.

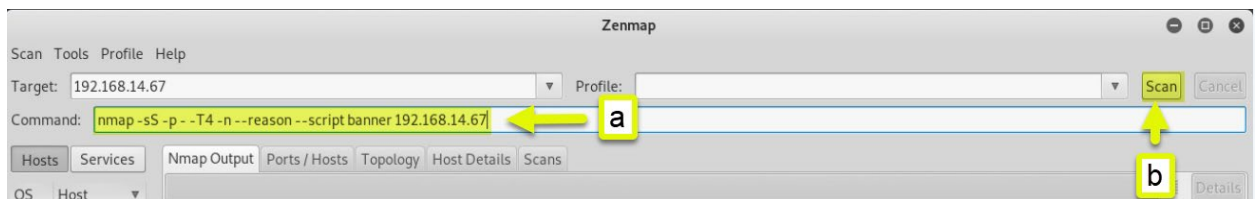
Command line:

```
nmap -sS -T4 -p- -n --reason --script=banner <ip address>
```

1. On the toolbar of the Kali Linux machine select “Zenmap (as root)”.



2. From the Zenmap console:
 - a. Type in the Command Line from above
Note: This will automatically fill in the “Target: “ box in the console.
 - b. Click scan.



Interpreting Results:

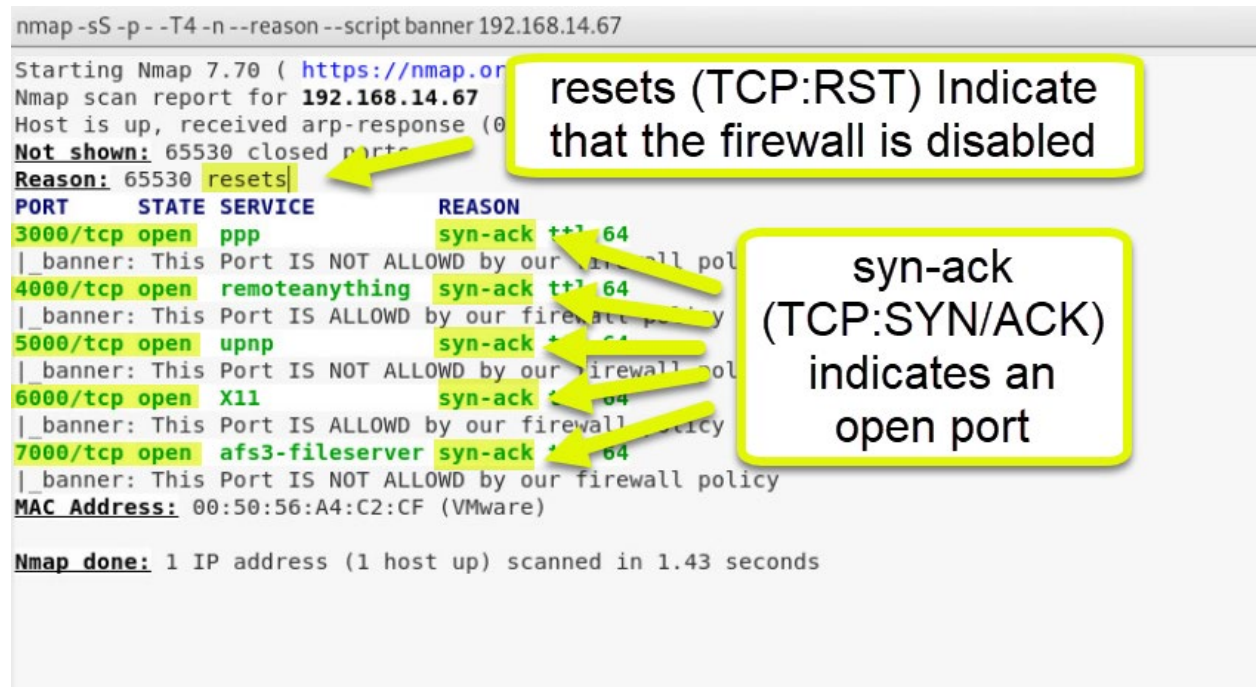
In the case of this scan we can see that there are 5 ports open (3000,4000,5000,6000,7000) and 65530 closed. What we are interested in is the fact that all 65530 closed ports responded with a "reset". This indicates that the firewall on the target is disabled (likely against policy) and needs to be enabled.

Once the firewall is up you should only receive responses from open ports. This list should be matched against the firewall policy in an audit and all ports not in the policy should be closed.

```
nmap -sS -p - -T4 -n --reason --script banner 192.168.14.67

Starting Nmap 7.70 ( https://nmap.org )
Nmap scan report for 192.168.14.67
Host is up, received arp-response (0.000s)
Not shown: 65530 closed ports
Reason: 65530 resets|
PORT      STATE SERVICE      REASON
3000/tcp  open  ppp          syn-ack ttl=64
|_banner: This Port IS NOT ALLOWD by our firewall policy
4000/tcp  open  remoteanything syn-ack ttl=64
|_banner: This Port IS ALLOWD by our firewall policy
5000/tcp  open  upnp        syn-ack ttl=64
|_banner: This Port IS NOT ALLOWD by our firewall policy
6000/tcp  open  X11         syn-ack ttl=64
|_banner: This Port IS ALLOWD by our firewall policy
7000/tcp  open  afs3-fileserver syn-ack ttl=64
|_banner: This Port IS NOT ALLOWD by our firewall policy
MAC Address: 00:50:56:A4:C2:CF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.43 seconds
```



ToDo:

You company policy indicates that only ports 4000 and 6000 are allowed open on the target machine 192.168.14.67. You need to audit the machine to ensure the policy is being followed.

1. Go ahead and do the scan above. What information can you gain from looking at the results of the scan? _____
2. Now go to the Ubuntu machine
 - a. Open a command shell



- b. Run the following script.

```
sudo LabFiles/Lab1/enable-firewall.sh
```

```
sysadmin@grc-ubuntu:~$ sudo LabFiles/Lab1/enable-firewall.sh  
[sudo] password for sysadmin:
```

3. Now rerun the scan. Is the machine now working within the firewall policy? _____
Note: This scan will take MUCH longer than the first one (about 11 minutes).