Activity: Recon and Defense (Network Security I)

Instructors : Kunwadee Sripanidkulchai, Ph.D.

Overview

In this activity, practice our reconnaissance and defense skills. There are 3 parts to this activity: (1) Prepare your target, (2) Recon, and (3) Defense.

NMAP ON NOTEBOOK2
LINUX VM ON NOTEBOOK1

GOALS FOR TODAY

1) Install a web server here

2) Run nmap

3) Defense: block packets

Writing firewall rules is a challenge in a large environment.
The rule of thumb is to first DROP,
then ACCEPT only specific things in.

Part I: Prepare your target

Pair up with another student, and have one student be the target and one student be the attacker. Your target is the Linux VM in one of your notebooks. Install a web server (Apache httpd) in your VM if you do not already have one installed. Also, install an ssh server (sshd) so that users can remotely login to the VM. The following instructions are for Debian/Ubuntu. If you use a different flavor of Linux (my favorite is RedHat), I trust that you probably know how to adjust the commands accordingly.

1. You may need to use sudo if you are not root.

```
$ apt-get update
$ apt-get install apache2
$ apt-get install openssh-server
```

To start and enable the Debian ssh server to start at system reboot, run these commands:

```
$ systemctl start ssh.service
$ systemctl enable ssh.service
```

And run this to check the status of your ssh service.

```
$ systemctl status ssh.service
```

2. Test that the web server is available.

Test 1: Use a browser in your target VM to visit the web server on the target VM.

Test 2: Use a browser in your target notebook (notebook 1) to visit the web server on the VM. If you cannot reach the web server, modify your VM network settings in VirtualBox to "Bridge". Google for how to modify the setting if you cannot find it in VirtualBox.

3. Connect both target (notebook 1) and attacker (notebook 2) to the same network. If you are on campus, DO NOT USE chula wifi. Use a hotspot (shared from one of your phones). Ask the instructor why chula wifi will not work for this activity.

Check that your target VM's network interface has an assigned IP address.

```
$ ifconfig -a
```

If it doesn't, refresh the network configuration. On my Debian VM, the interface is called "enp0s3".

```
$ ifdown enp0s3
$ ifup enp0s3
```

Check again that your VM has an assigned IP address.

```
$ ifconfig -a
```

Note your VM's IP address. From now on, your target VM, your target host (notebook 1), and your attacker (notebook 2) can communicate. Read here for more info: https://www.virtualbox.org/manual/ch06.html#networkingmodes

4. Determine where the http access logs are stored on your VM. Mine is at /var/log/apache2/access.log. You will need to look at this to see the results of the scans.

5. Test that the ssh server on your VM is available by running ssh from the target notebook. Mac users may use Terminal. Windows users need to use putty, cygwin, CMDer, WSL, etc.

```
    $ ssh -l your_username your_vm_ip_address
```

6. Test that the attacker can access the web server on your VM. Use a browser in your attacker notebook (notebook 2) to visit the web server on the VM.

Part II. Reconnaissance

We will use nmap to scan your notebooks and the Linux VM to see what nmap can learn from our default machine settings.

1. Download nmap and install it on the attacker notebook. We will run nmap using the "Intense scan" profile. https://nmap.org/download.html
2. nmap comes built-in with many scanning profiles. Run nmap using the "Intense scan" profile from the attacker notebook → your target Linux VM's IP address.
3. Run nmap from the attacker notebook against itself (localhost).
4. Run nmap from the attacker notebook against the target notebook (not the VM, but the host).
5. (Optional) For computer network enthusiasts (ผู้ชื่นชอบ Computer Networks), run Wireshark on both notebooks. You will see all the details in the packets that nmap uses to perform the scan, and all the responses back from your VM. There are lots of details here that you may find to be interesting.

Answer these questions and provide corresponding evidence/screenshots:
Q1. Notice the open ports on all 3 devices (the attacker notebook, the target notebook, and the target Linux VM). Does anything look suspicious, i.e., some ports that you are not aware of that are open on the VM or on your notebooks?

Q2. Look at the information provided by nmap about your OS's on all 3 devices. Is the information correct? Why is it or why is it not correct?

Q3. What do you think about the information you can get using nmap? Scary?

Q4. Look at the access.log file for the web server in your Linux VM. What IP addresses do you see accessing the web server? Which devices do these IP addresses belong to?

Q5. Find the nmap scan in the web server log. Copy the lines from the log file that were created because of the nmap scan.

Part III. Defense

Our goal is to defend our target VM from future port scans and allow only certain services to be accessed by users. We will use iptables. iptables is a basic firewall on Linux. We can only configure it to DROP/ACCEPT packets based on their IP address/port/protocol information.
The policy we want to implement is:

- DROP all incoming network traffic
- ACCEPT web traffic from all sources (because when we have a web server we usually want the world to be able to access it)
- ACCEPT ssh traffic from only our target notebook's (host) IP address (because only server admins/users should be able to remotely access the server, so we can give access to them if they are logging in from specific IP addresses)

1. From the scans, you see that our VM responds to pings. If you cannot find this result in the scan, try rerunning the scan but this time select the "Ping scan" profile. Also, ping your VM from your notebook.

Given that pings reveal information about us, we will block pings using iptables in order to prevent nmap from learning too much about us.

2. How to use iptables in a minute:
iptables has three built-in chains INPUT, OUTPUT and FORWARD. You install rules into the chains. For example, to interact with packets coming into the machine, you add rules to the INPUT chain. If you want to interact with packets coming out of the machine to the rest of the world, you add rules to the OUTPUT chain.

Chains are combined into tables. We will be working with the default "filter" table.

Each of the chains filters data packets based on

- Source and Destination IP
- Source and Destination Port number
- Network interface
- State of the packet

Target action for the rule:

- ACCEPT
- DROP
- REJECT
- QUEUE
- RETURN
- LOG

For example, if I run the command

```
$ iptables -L
```

I get the following result showing that my current policy is to allow everything in and everything out.

3. Writing firewall rules is a challenge in a large environment. The rule of thumb is to first DROP, then ACCEPT only specific things in.

Write firewall rules to
DROP all incoming packets, but
ACCEPT packets destined to the web server (http) from anywhere, and
ACCEPT packets destined to the ssh server only from your notebook.

Hint: You only need to add rules to the INPUT chain.

Good examples on the specific rules here:
http://coewww.rutgers.edu/www1/linuxclass2015/lessons/Security/sec_12.php

4. After you successfully install your rule(s), test your firewall.

Test 1: Ping your VM from your two notebooks. You should not see any responses.
Test 2: Access the web server on your VM from your browser on your 2 notebooks. You should be able to get the same web page as before.
Test 3: ssh from your target notebook (notebook 1) into the VM. You should be able to get the same results as before.

If your configured firewall fails all 3 tests, try again.

5. Run nmap against your VM again. Compare the reported results from your new scan to your previous scan before using iptables.

Answer these questions and provide corresponding evidence/screenshots:

Q6. After you successfully install your iptable rule(s), how do the reported results from your new nmap scan compare to your previous scan before using iptables? Look to see if OS detection, port open results, etc. have changed. Something(s) have definitely changed.

Q7. Notice that nmap can still figure out you have Apache httpd running. Look at the access.log file for the web server in your Linux VM. Are the logs the same as in Part II?

Q8. Explain whether or not you could prevent nmap from reaching the web server while still allowing legitimate clients to get service. Will a firewall be sufficient for this? Or do you need some other device? Please think critically about this.

Q9. What are your firewall rules? Run iptables -L on your VM and enter the output here.