CS473 - Homework # 1 Spring 2019

Instructors: Mobin Javed and Fareed Zaffar

Due: 27th February 2019 at 11.55pm

The focus of this homework is hands-on network security. You will learn:
1. Scanning a network using the `Nmap` tool
2. Building your own port scanner
3. Performing reconnaissance on a webserver

Question 1: Nmap and Wireshark

(1) Port Scanning using Nmap

Port scanning is a method that can be used by an attacker to probe which ports are open on a given host, learning details about which software the server is running on publicly-addressable interfaces. With this information, an attacker gains a better understanding of where and how to attack the victim server. Port scanning takes advantage of conventions in TCP and ICMP that seek to provide a sender with (perhaps too much!) information on why their connection failed. In this part, you will use the Nmap tool (`https://nmap.org`) to scan the server scanme.nmap.org. By doing so, you should be able to see the powerful information that a simple scan can reveal. In your scan, make sure to:

1. Only scan scanme.nmap.org! Do not scan any other servers. You should only scan a server if you have explicit permission from the server operator to do so.
2. Record the traffic with Wireshark (see part 2)
3. Use a TCP SYN scan. (Hint: Read the Nmap man pages to find the appropriate flag to use).
4. Enable OS detection, version detection, script scanning, and traceroute. (Hint: This is a single flag).
5. Do a quick scan (-T4).
6. Scan all ports.

When you get the result, report the following about the target server (scanme.nmap.org) based on the results of the scan:

1. What is the full command you used to run the port scan (including arguments)?
2. What is the IP address of scanme.nmap.org?
3. What operating system is the target server running? What version number?
4. What ports are open on the target server? What applications are running on those ports? (For this part, you only need to report the service name printed by Nmap.)
5. The target machine is running an SSH server. What SSH software and version is being used?
6. The target machine is also running a web server. What web server software and version is being used? What ports does it run on?

Please answer all questions briefly; no response should take more than two sentences.

(2) Wireshark Packet Sniffing

Wireshark is a tool to monitor local network traffic. Wireshark has access to complete header information of all packets on a monitored interface and presents a helpful GUI for understanding the structure of different protocols. Because of this it can be a valuable debugging tool for networking projects, as you will see later in Question 3.

Use the Wireshark packet analyzer (`https://www.wireshark.org/`) to examine the traffic generated by Nmap during the scan in Part (1).

You will need to install and start Wireshark and record traffic on the interface Nmap will use to scan before actually running the scan.

When you get the result, take a look at the Wireshark capture. Use Wireshark filtering functionality to look at how Nmap scans a single port. A primer on Wireshark is available in the homework directory (please see `network traffic analysis primer.pdf`).

Report the following about the target server based on the results of the scan:

1. What does it mean for a port on scanme.nmap.org to be “closed”? More specifically, what is the TCP packet type, if any, the server gives in response to a SYN packet sent to port that is “closed”?
2. What does it mean for a port on scanme.nmap.org to be “filtered”? More specifically, what is the TCP packet type, if any, the server gives in response to a SYN packet sent to port that is “filtered”?
3. In addition to performing an HTTP GET request to the web server, what other HTTP request types does Nmap send?

Once again, please answer all questions briefly; no response should take more than three sentences.

Question 2: Port Scanner

In this question, you will build your own port scanner.

(1) Build a Port Scanner

Port scanning is a technique used to find network hosts that have services listening on one or more target ports. It can be used offensively to locate vulnerable systems in preparation for an attack, or defensively for research or network administration.

Your port scanner, `PortScan`, will probe all 2^{16} TCP ports on a targeted host, and report the ports that accept connections. Your scanner should not require superuser (root) privileges and can attempt to establish full TCP connections to the tested ports.

Your scanner should scan the ports in order (i.e., from 0 to 65535) as quickly as possible. That is, you should not pause or sleep between probes.

For each open port, `PortScan` should report both the port number and the service that normally runs on that port. The latter can be found by using the getservbyport() and socket.getservbyport() calls in C and Python, respectively. `PortScan` should also report how long it took to probe all ports, the number of ports that were found to be open, and the scan rate (ports scanned per second).

The command-line usage for `PortScan` should be:

```
PortScan target or python PortScan.py target
```

for C/C++ and Python, respectively, where target is the hostname or IP address of the machine that is to be scanned.

Question 3: Reconnaissance

Before attacking a machine, it is vital to do reconnaissance (recon) on the target system. This helps us gather and log various important pieces of information pertaining to the system or the target application. Recon also helps us develop a better understanding of what configuration the target server is using. If done correctly, recon can save ethical hackers a lot of time in later stages and makes the process more precise and meaningful. It is pointless going into a security audit/pen test and just trying random attack vectors, as quite often, such an attempt will not work and waste a substantial amount of time. Hence, in this lab, we will see how to properly and systematically conduct a thorough evaluation of the vulnerabilities and misconfigurations plaguing a website, server or application. To this end, we will be working with some tools that are designed to actively scan and check servers (active recon). Particularly, we will be using the Nikto tool, which has become the “go-to” tool that attackers and testers use nowadays for active recon.

We have setup a virtual machine for you that has quite a few vulnerabilities. You need to download the image of the vulnerable VM appliance and import it to VirtualBox using the following link:
https://drive.google.com/file/d/1ZGdDmMd3Npy4guNn9tYGX6LHALgqxF9r/view?usp=sharing

Once this VM is up and running, you can run Nikto and Nmap on it from your Kali Linux VM which you can download from:
https://images.offensive-security.com/virtual-images/kali-linux-2018.4-vbox-i386.ova

The login credentials for both VMs are:
Vulnerable VM:
Username: targetVM
Password: targetVM

Kali Linux (Attacker) VM:
Username: root
Password: toor

For setting up the network correctly in VirtualBox:
Both VMs (target and attacker) need to be part of the same NAT network. Go to VirtualBox, create a new NAT network. Then go to the individual settings of each VM and have them

join this new NAT network. Everything needs to be done via the VirtualBox GUI.

A primer on Nmap and Nikto is available in the homework directory (please see `recon-primer.pdf`).

Once you are finished with the Nmap and Nikto scans, you are required to submit a report with the following tasks:

A. Run Nmap on the VM image and list all the open ports and services. Explain any three services from the list. Do you see any “unknown” services in the scan result? Why is Nmap unable to determine which service this is (in the case of unknown)?

B. Run Nikto on the aforementioned server and list all the vulnerabilities that the tool outputs (just the name or the code).

C. As mentioned above, not all reported items will be vulnerabilities. Some will be warnings and some might be just pieces of information that are harmless. Hence, create categories and try to organize the information into these categories. You need to have a very good understanding of what is a problem and what is not. Find at least one vulnerability that is a false positive (not really a vulnerability) and three actual vulnerabilities.

D. Distinguish a web-specific vulnerability from a host or OS-based vulnerability. Give one example of each from your scan and then explain both in detail.

E. Lastly, give an explanation on how we can solve at least three of the identified vulnerabilities. The goal is to determine how well you understand the output of a security tool and how you would approach the problems reported, to solve them.