In the name of God

Sharif University of Technology
Department of Computer Engineering

CE 442: Data and Network Security

Mehdi Kharrazi *Ordibehesht 23rd, 1396*

Homework 3

Please upload your answers/report **in PDF format** to Quera. The HW file name should be **"NS-PA3-StudentID.pdf"**.

Begin by downloading the traces you will need to analyze. You can find the trace of part I and II according to your SID which is odd or even (e.g., CE442-odd or CE442-even). Trace files are posted on the course website. Be sure to download the trace that matches your student ID. The traces are different, so if you download the one for someone elses account, all your answers will be wrong!

you should only use the tools mentioned for each question when answering them. You will need to submit any code or filter used with your answers. the questions must not be answered by manually inspecting the trace with existing tools. Furthermore, in some cases you may find it useful to write one or more simple scripts to help.

Part I (Tcpdump)

1. **HTTP Sessions**

   For this problem, find all web servers that were successfully visited in the trace (that is, contacted via HTTP). Include any servers that engaged in a valid instance of the HTTP protocol.

2. **Traceroute Scanning**

   Traceroute is a utility for finding the addresses of the routers along the IP route between the host it is being run on and an arbitrary destination. Attackers sometimes use traceroute to find out about a victims network infrastructure (routers and possibly firewalls). Identify the host that is running traceroute for detecting routers on a path. Submit the IP address of the host running traceroute and the IP address of the destination of the traceroute path.

3. **Service Versions**

   Finding hosts running specific versions of servers is an important step in exploiting them; in general, older versions will have more vulnerability. For this problem, find the host running the oldest version of Apache. (Apache is the most widely used web server on the Internet.) Dont count Apache-Coyote as Apache; also, ignore any servers that dont specify their version. Submit that hosts IP address.

Part II (Wireshark)*

1. **Directory Traversal**

   One simple way people attempt to exploit a web server is by making requests for files outside the normal directories it serves using pathnames with sequences like ../../../. (Of course, a reasonably well-implemented web server will not fall for tricks like this.) Find a host that appears to be attempting this type of attack and submit its IP address.

2. **Password Guessing**

   If youve ever looked through the logs of an SSH server, you've likely seen attempts to login through brute force guessing of usernames and passwords. Of course, the same attack is possible for any type of protocol with password authentication. There is one host that attempted such an attack against a password protected FTP server. Find that host and submit the IP address of the attacker.

3. **DNS and Source Port Randomization**

   Recall that most clients now select a random UDP source port when making DNS queries to help prevent the Kaminsky attack. For this problem, look for clients which do not use a random source port. There are exactly two such DNS resolvers (not including MDNS6). As your answer to this question, submit the IP addresses of the two DNS resolvers (not counting MDNS) that use the same source port for all the DNS queries they make (and make more than 1 query).

4. **Unencrypted Usernames and Passwords**

   Next, find an unencrypted username and password. Note that we are interested in a real username and password, so failed login attempts dont count. Examples of some protocols that can send usernames and passwords without encryption are Telnet, FTP, HTTP, and POP3. List the username and password as your answer.

(*) Some of the questions in this part were obtained in whole, or partially, from prof. Paxson/Wagner at UC Berkeley.

Part III (iptables)

In this section of the homework, you will be using iptables to conduct a number of tasks, based on a described scenario. It is assumed that you have root access to a Linux box for this experiment and have iptables installed. If you don't have such access, then you should install a Linux distribution either on your machine or in a virtual machine. Consider the network topology depicted in Figure 1 and write iptables rules to enforce following requirements:

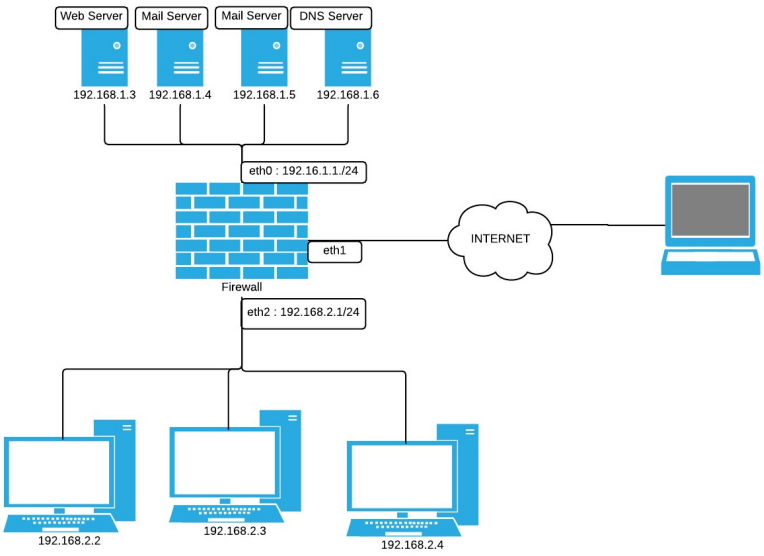

Figure 1: Network Topology

- Allow all connections between DMZ and Internal Network if initiated from Internal Network.
- Drop all connections from Internet to Internal Network
- Allow HTTP, DNS , ICMP, SSH and SMTP connections from Internet to DMZ.
- Route all HTTP requests from Internet to HTTP server dedicated in DMZ on port 80
- Allow 100 connections per minute from Internet to HTTP server.
- Block all connections from Internal Network to www.example.com .
- Log all rejected packets.