In the name of Allah

الله

Network management and security Laboratory Manual

University of Tehran

School of Electrical and Computer Engineering

Computer Network Lab

Dr. Ahmad Khonsari -
a_khonsari@ut.ac.ir

Amir Haji Ali Khamseh'i -
khamse@ut.ac.ir

Sina Kashi pazha -
sina_kashipazha@ut.ac.ir

Mohammad Ali Shahsavand -
mashahsavand@ut.ac.ir

Amirahmad Khordadi -
a.a.khordadi@ut.ac.ir

November 29, 2018

۸ آذر ۱۳۹۷

1 Exercises on secure applications

1.1 Excercise One

In this excercise we will study security vulnerability of ftp and telnet protocol . To do so, wi will create mininet topology with single hub[1] connected to three hosts then we will connect from h1 to h2 through ftp and telnet connection and capture h1 password on h3. Let's do it.

- Start pox controller with below command to force mininet switches act like hub.

```
$ python pox.py opeenflow.of_01 --address=127.0.0.1 --port=6337 forwarding.hub
```

- Run below command to start mininet with one single switch and three hosts and connect it to pox controller.

```
$ sudo mn --topo single,3 --controller remote,ip=127.0.0.1,port=6633
```

- start ftp server on h2 with :

```
# /usr/sbin/vsftpd
```

- Run **wireshark &** on h3
- Login to h2 ftp from h1

```
h1> ftp mininet@10.0.0.2
```

- Capture h1 password on wireshark output

Repeat the above experiment, but use telnet to connect from h1 to h2 and capture h1 password on h3. [2]

Lab Report

- Can you see the login ID and the password in the FTP experiment? Submit the two packets you captured.
- Can you see the login ID and the password in the TELNET experiment? Submit the packets you captured.
- What is the difference between FTP and TELNET in their transmission of user IDs and passwords? Which one is more secure?

1.2 Excercise Two

Run previous mininet topology and connect it to pox controller but rather than using ftp and telnet use ssh and sftp as described in below steps.

- ~~start ftp server on h2 with~~ restart ssh service on h2 to enable ssh and sftp service on it with:

```
h2> service ssh restart
```

- Run **wireshark &** on h3
- Login to h2 sftp from h1 by:

```
h1> sftp mininet@10.0.0.2
```

- Capture packets on wireshark output

Repeat the above experiment, but use ssh and save the wireshark output for lab report.

[1] hub forwards incoming packets to all of its ports, which means it always floods packets
[2] Don't foget to restart xinetd with '/etc/init.d/xinetd restart' on h2 to start telnet server.

Lab Report

- In each experiment, can you extract the password from the tcpdump output? Can you read the IP, TCP, SSH headers? Can you read the TCP data?
- What is the client protocol (and version) used in both cases?
- What is the port number used by the ssh server? What is the port number used by the sftp server? Justify your answer using the wireshark output and the /etc/services file.

2 Exercises on Firewalls and Iptables

Excercise Three

Start mininet with default topology and Execute iptables -L -v on h1 and h2 to list the existing rules in the filter table. Save the output for the lab report.

Append a rule to the end of the INPUT chain, by executing

```
h2> iptables -A INPUT -v -p TCP --dport 23 -j DROP
```

on h2. Run iptables -L -v again on both hosts to display the filter table. Save the output.

- Start telnet server on h2 with '/etc/init.d/xinetd restart'.
- Capture packets on both hosts with wireshark
- Try to login with telnet from h1 to h2

Lab Report

- Can you telnet to the host from the remote machine?
- From the wireshark output, how many retries did telnet make? Explain the exponential backoff algorithm of TCP timeout and retransmission.

Excercise Four

Keep previous mininet running and delete the rule created in the last exercise on h2, by:

```
h2> iptables -D INPUT -v -p TCP --dport 23 -j DROP
```

Then, append a new rule to the INPUT chain:

```
h2> iptables -A INPUT -v -p TCP --dport 23 -j REJECT --reject-with tcp-reset
```

Execute **iptables -L -v** to display the new rule. On both machines in your topology, restart wireshark output, and then telnet from h1 to h2. Save the wireshark output for the lab report.

Lab Report

- Explain the difference between the wireshark outputs of this exercise and the previous exercise. How many attempts did TCP make this time?

3 Exercises on secure Apache server

In the exercises in this section you don't need to create mininet topology, run command on your ubuntu terminal.

Excercise Four

Run man openssl to study the OpenSSL command line tool. Create a new private key for the Apache server, using:

openssl genrsa 1024 > /etc/httpd/conf/ssl.key/server.key

To create a self-signed certificate, go to the /etc/httpd/conf directory, and execute: make testcert.

Then you will be asked a number of questions, regarding the location, affiliation, etc. of the Apache server. After you type in the answers, a self-signed certificate is created at **/etc/httpd/conf/ssl.crt/server.crt**.

Save the make output for the lab report.

Excercise Five

Restart the Apache server to load the new key and the new certification: **/etc/rc.d/init.d/httpd restart**.

Execute wirshark on your host to capture the packets between your host and a remote host. On your host, start the Mozilla web browser. After typing in the URL https://<your host IP>, a dialog window titled Website Certified by an Unknown Authority will pop up, reporting the reception of a certificate signed by an unknown authority and asking if you want to continue.

Click the Advance button. Then a Certificate Viewer window pops up, displaying detailed information about the received certificate. Examine the certificate and confirm it's exception. Save the pictures for the lab report.

Use wireahark output and examine the operation of SSL.

Lab Report

- What is the port number used by the secure Apache server?
- Compare the general information of the received certificate with the make output saved in the last exercise. Are they consistent?
- What is the Subject of the received certificate? Who is the Issuer of this certificate? Are they the same?
- What is the Certificate Signature Algorithm used to generate and distribute this certificate?
- When was the certificate signed? When will it expire?