بسم الله الرحمن الرحیم

In the name of Allah

Network Management and Security[1]

Laboratory Manual

University of Tehran
School of Electrical and Computer Engineering

دانشگاه تهران

دانشکده‌ی مهندسی برق و کامپیوتر

Computer Network Lab

آزمایشگاه شبکه‌های کامپیوتری

Dr. Ahmad Khonsari
دکتر احمد خونساری
a_khonsari@ut.ac.ir

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

Muhammad Borhani
محمد برهانی
m.borhani@ut.ac.ir

AmirAhmad Khordadi
امیراحمد خردادی
a.a.khordadi@ut.ac.ir

Sina Kashipazha
سینا کاشی‌پزها
sina_kashipazha@ut.ac.ir

Hadi Safari
هادی صفری
hadi.safari@ut.ac.ir

November 14, 2019

۲۳ آبان ۱۳۹۸

[1]S. Panwar, S. Mao, J.-dong Ryoo, and Y. Li, "Network management and security," in TCP/IP Essentials: A Lab-Based Approach, Cambridge: Cambridge University Press, 2004, pp. 187–213.

Part I
SNMP Exercises

For the exercises in this section, the network topology is given in Figure 1.3, where all the hosts are connected in a single network segment using their IP addresses, i.e. from 128.238.66.100 to 128.238.66.102.

Table 1: The IP addresses of the hosts (Table 1.2)

Host	IP Address	Subnet Mask
h0	128.238.66.100	255.255.255.0
h1	128.238.66.101	255.255.255.0
h2	128.238.66.102	255.255.255.0
h3	128.238.66.103	255.255.255.0
h4	128.238.66.104	255.255.255.0
h5	128.238.66.105	255.255.255.0
h6	128.238.66.106	255.255.255.0
h7	128.238.66.107	255.255.255.0

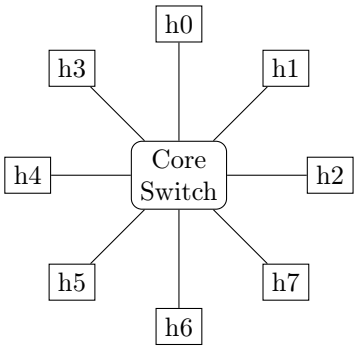

Figure 1: A single segment network (Figure 1.3)

Before the lab, you should:

1. Backup the original `snmpd` configuration file: `mv /etc/snmp/snmpd.conf /etc/snmp/snmpd.conf.save`.
2. Create a simple configuration file `/etc/snmp/snmpd.conf` with a single line defining a read-only community **guest**:

```
rocommunity guest
```

1 SNMP Service and MIB Struct

Use `pgrep snmpd` to check if `snmpd` is started. Try to stop and then start the SNMP agent daemon using `service snmpd start|stop`.

Study the `snmpd` configuration file `/etc/snmp/snmpd.conf`. Also study the default configuration file `/etc/snmp/snmpd.conf.save`. This file is well commented. Read the comments and study the configuration options.

Study the MIB files in the `/usr/share/snmp/mibs` directory. Examine the Interface MIB `IF-MIB.txt` and the TCP MIB `TCP-MIB.txt` to see the MIB objects and data types. Save these two files for the lab report.

Report

1. What is the community name used in this lab? What is the use of the community name?
2. What is the data type for the MIB object `ifMtu.2`? What is the definition of the MIB object `ifPhysAddress` and `ifInOctets`?
3. What is the data type and definition of `tcpRtoAlgorithm`? What values are allowed for `tcpRtoAlgorithm`? What is the definition of `tcpMaxConn`?

2 SNMP List Objects

Use `snmpwalk -v 2c -c guest localhost interface` to display the Interface MIB.

Use `snmpwalk -v 2c -c guest localhost tcp` to display the TCP MIB.

You may run `man snmpwalk` to find out the meanings of the options used in the commands. Compare the outputs with the MIB files you saved in the previous exercise. Also compare the outputs of the first command with that of `ifconfig -a`.

Retry the `snmpwalk` commands, but change **guest** to **public**. Can you display the MIBs this time?

Report

1. What is the MTU of the Ethernet interface? What is the MTU of the loopback interface? Justify your answer with the `snmpwalk` output and the `ifconfig` output.
2. Why did the `snmpwalk` command with a community name public fail?

3 SNMP Remote Access

Execute `tcpdump udp port 161` or `wireshark` to capture SNMP messages.

Run from *h1* host `telnet h0 echo` (use *ctrl+]* to terminate `telnet` and then `quit`) and while telnet is running, try again to run `snmpwalk -v 2c -c guest localhost tcp` on *h0*.

Try to run `snmpwalk -v 2c -c guest h0 tcp` from *h2*.

Execute `snmpget -v 2c -c guest remote-host IF-MIB::ifMtu.1` to get the MIB object `IF-MIB::ifMTU.1` from a remote machine.

Save the `snmpget` output and terminate `tcpdump`.

Use `wireshark` to analyze the format of the captured SNMP Get and Response messages. Print the messages for the lab report.

Report

1. What is the port number used by the SNMP agent?
2. What are the full text-based and numerical object ID's of the MIB object `interface.ifMTU.2`? What was the value returned? Justify the answer using Figure 9.3 of reference book and the `ifconfig` output.
3. Draw the format of one of the SNMP messages saved, including the name and value of each field.

Part II
Exercises on Secure Applications

4 Plain Transfer

Execute `tcpdump -enX -s 100` or `wireshark` to capture packets between your machine and a remote machine.

Execute `ftp remote-host` . When prompted, type *1111* for the login ID, and *2222* for the login password. Then terminate `tcpdump` and `ftp`.

Use `wireshark` to analyze and print the packets that carry the login ID and the password for the lab report.

Repeat the above experiment, but use `telnet` and the `wireshark` to analyze and print output.

Report

1. Can you see the login ID and the password in the FTP experiment? Submit the two packets you printed.
2. Can you see the login ID and the password in the TELNET experiment? Submit the packets you printed.
3. What is the difference between FTP and TELNET in their transmission of user ID's and passwords? Which one is more secure?

5 Secure Transfer

Start `ssh` service on target machine by `service ssh start` .

In this exercise, students pair up to work together using two workstations.

Execute `tcpdump -enX -s 100` or `wireshark` host your host and remote host to capture packets between your machine and a remote machine.

Execute `sftp` *`remote-host`* . When prompted, type *yes* to continue the connection and *1111* for the login password. Then terminate `tcpdump` .

Use `wireshark` to analyze and print one or two `SSH` packets for the lab report.

Repeat the above experiment, but use `ssh` and the `wireshark` to analyze and print output.

Report

1. 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?
2. What is the client protocol (and version) used in both cases?
3. 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 `tcpdump` output and the `/etc/services` file.

Part III

Exercises on Firewalls and `iptables`

In this exercise using two workstations.

6 Firewall Basic

Execute `iptables -L -v` 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

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

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

On both machines, execute `tcpdump` . Then, `telnet` to the host where the rule is set from the remote machine. Save the tcpdump output for the lab report.

Report

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

7 Firewall Action

Delete the rule created in the last exercise, by:

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

Then, append a new rule to the `INPUT` chain:

```
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 `tcpdump`, and then `telnet` to the host where the rule is set from the remote machine. Save the wireshark output for the lab report.

Report

1. Explain the difference between the `tcpdump` outputs of this exercise and the previous exercise. How many attempts did TCP make this time?

Part IV

Exercises on Secure Apache Server

In the exercises in this section, using two workstations from *term* and *gui* hosts. Run `apache2` service on *server*

8 Generate Certificate

Run `man openssl` to study the OpenSSL command line tool.

Create a new private key for the Apache server, using:

```
openssl genrsa 1024 > /etc/apache2/ssl/server.key
```

To create a self-signed certificate, go to the `/etc/apache2/ssl/` directory, and execute:

```
openssl req -new -x509 -days 365 -key /etc/apache2/ssl/server.key -out
    /etc/apache2/ssl/server.crt
```

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/apache2/ssl/server.crt`

You can generate private key and self-signed certificate whit one command by: [1]

```
openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout /etc/apache2/ssl/server.key -out
    /etc/apache2/ssl/server.crt
```

Let's go over exactly what this means.

- `openssl` : This is the basic command line tool provided by OpenSSL to create and manage certificates, keys, signing requests, etc.
- `req` : This specifies a sub-command for X.509 certificate signing request (CSR) management. X.509 is a public key infrastructure standard that SSL adheres to for its key and certificate management. Since we are wanting to create a new X.509 certificate, this is what we want.

[1] You can use simple command as `make-ssl-cert generate-default-snakeoil --force-overwrite` without editing default apache ssl config.

- `-x509` : This option specifies that we want to make a self-signed certificate file instead of generating a certificate request.
- `-nodes` : This option tells OpenSSL that we do not wish to secure our key file with a passphrase. Having a password protected key file would get in the way of Apache starting automatically as we would have to enter the password every time the service restarts.
- `-days 365` : This specifies that the certificate we are creating will be valid for one year.
- `-newkey rsa:1024` : This option will create the certificate request and a new private key at the same time. This is necessary since we didn't create a private key in advance. The rsa:1024 tells OpenSSL to generate an RSA key that is 1024 bits long.
- `-keyout` : This parameter names the output file for the private key file that is being created.
- `-out` : This option names the output file for the certificate that we are generating.

The questions portion looks something like this:

```
Country Name (2 letter code) [AU]:IR
State or Province Name (full name) [Some-State]:Tehran
Locality Name (eg, city) []:Tehran
Organization Name (eg, company) [Internet Widgits Pty Ltd]:University of Tehran
Organizational Unit Name (eg, section) []:ECE Department
Common Name (e.g.\ server FQDN or YOUR name) []:ece.ut.ac.ir
Email Address []:netlabut.ac.ir
```

Save the output for the lab report.

9 HTTPS Handshake and Request

Restart the Apache server to load the new key and the new certification: `service apache2 restart` .

Execute `wireshark` to capture the packets between your host and a remote host.

On the remote host, start the *Mozilla* web browser. After typing in the URL `https://your-host`, a dialog window titled *Connection is not not secure* or *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. Add "Certificate" to exceptions and continue browsing.

Click the *View Certificate* button. Then a *Certificate Viewer* window pops up, displaying detailed information about the received certificate. Examine the certificate and dump the window into a picture if necessary. Save the pictures for the lab report.

Click the *Continue* button in the *Website Certified by an Unknown Authority* dialog window to accept the certificate. Then terminate `wireshark` and Mozilla.

Use `wireshark` to examine the operation of SSL.

Report

1. What is the port number used by the secure Apache server?
2. Compare the general information of the received certificate with the `openssl` output saved in the previous exercise. Are they consistent?
3. What is the Subject of the received certificate? Who is the Issuer of this certificate? Are they the same?
4. What is the *Certificate Signature Algorithm* used to generate and distribute this certificate?
5. When was the certificate signed? When will it expire?

Part V
Exercises on Auditing and Intrusion Detection

Start `apache2` service and browse several page in the *Mozilla*. Open random URL that not exist on the web server to generate log for apache. Browse again page as `securehttp`.

10 Trace Log

Go to `log` folder at `/var/log/` to examine the log files in your host. If a log (e.g. the *Apache Access Log* at `/var/log/apache2/access.log`) is too long, use `grep keyword access.log` (e.g. `GET`) in the console to display those log entries containing the keyword. Enter the keyword `failed` to display logged failures.

See other log files.

11 Log Analyzer

Linux uses a utility called `webalizer` to analyze the web server log files. `Webalizer` reads the `apache2` log files and creates a set of web reports on server statistics. Another utility is `goaccess` that analyze various access type and list it.

To analyze local log use this commands:

```
webalizer
```

To view the reports of the `webalizer` , start *Mozilla* and enter the URL `http://remote-host/usage/index.html`. Examine the web statistics displayed in the browser. Also click on the month links in the Summary by Month table to see the statistics of each month.

To analyze local log use this commands:

```
goaccess -a > /var/www/html/report.html
```

To view the report of the `goaccess` , start *Mozilla* and enter the URL `http://remote-host/report.html`. Examine the web statistics displayed in the browser. Also click on the side panel to see other section.

Report

1. List the most frequently visited pages at the local Apache server during the most recent month, respectively.
2. List the web pages that have the most number of bytes transferred by the local during the most recent month, respectively.

12 System Status

Execute `netstat -l` to display the listening sockets in your host.

Execute htis commands to see the system services info [2] and their status. Save the output for the lab report.

- `service --status-all`
- `systemctl list-units --type=service --state=running`
- `systemctl list-unit-files --type service --state=enabled`

[2] The old Linux OS service loaded in `/etc/init.d/` and `/etc/init/` directories

Report

1. Is the `rlogin`, `ssh` and `apache2` services enabled in your host?