TCP/IP Attack: Task1 Quiz

Started: 18 Apr at 13:55

Quiz instructions

Follow the instructions and answer the questions!

The requirement for passing this quiz is to score 35 points out of 40.

We assume that you have set up the lab VM on your computer.

Follow these instructions for the TCP/IP Attack assignment. In general, the instructions state the desired outcomes, and your task is to find out the exact commands you need to execute to achieve these outcomes. There are also corresponding questions, which expect you to give these commands as answers. Normally, you can issue a command with a set of flags in any order. However, the Canvas grading tool is not as flexible. Therefore, we require you to use a command with different flags in a specific order. You must pay close attention to the instructions and strictly write a command in the correct order as per the instructions. Otherwise, your answer will be considered incorrect.

Getting started

Before you work with the actual TCP/IP attacks, you will prepare the lab environment as follows:

- Launch the lab VM.
- On the lab VM, open one terminal window and run the following commands to start the LXC containers:
  - `lxc start attacker`
  - `lxc start inside-host`
  - `lxc start outside-host`
  - `lxc start firewall`
- Now, connect to these containers. You will open new terminal windows, one for each container, and run each command in a separate window:
  - `lxc exec attacker /bin/bash`

- `lxc exec inside-host /bin/bash`
- `lxc exec outside-host /bin/bash`
- `lxc exec firewall /bin/bash`

  Once you run the command, you will get into the root terminal in the container.

At this point, you should have four terminal windows opened. Each connects to one container.

1.1 SYN flooding attack

As stated in **RFC 4987** (https://tools.ietf.org/html/rfc4987), "*The SYN flooding attack is a denial-of-service method affecting hosts that run TCP server processes. The attack takes advantage of the state retention TCP performs for some time after receiving a SYN segment to a port that has been put into the LISTEN state. The basic idea is to exploit this behavior by causing a host to retain enough state for bogus half-connections that there are no resources left to establish new legitimate connections.*"

Your main objective as an attacker is to use the SYN flooding attack to cause a denial-of-service (DoS) on a telnet service running on the outside-host, and a legitimate user (i.e., inside-host) will not be able to access the telnet service. You will use the netwox tool number 76 to generate the SYN flooding attack and observe the results.

Before you begin the attack, you need to set up different containers.

On the **outside-host**, do the following:

- Check what services are listening on TCP sockets with the command:

  ```
  ss -ltn
  ```

  You should see a similar output to the output below with a DNS service (port 53) running locally (127.0.0.53) and an SSH service (port 22) running on both IPv4 (0.0.0.0) and IPv6 ([::]).

  ```
  root@outside-host:~# ss -ltn
  State    Recv-Q   Send-Q     Local Address:Port     Peer Address:Port  Proc
  ess
  LISTEN   0        4096       127.0.0.53%lo:53            0.0.0.0:*

  LISTEN   0        128              0.0.0.0:22            0.0.0.0:*
  ```

```
LISTEN   0        128              [::]:22             [::]:*
```

- Start a telnet service through xinetd (Extended Internet Service Daemon) with the command:

  ```
  service xinetd start
  ```

- Check what services are listening on TCP sockets again. Verify that it is now listening for telnet (port 23).

On the **firewall**, do the following:

- Add a default route to the attacker with the command:

  ```
  ip route add default via 10.0.20.3
  ```

  By default, netwox 76 will generate traffic with random IP addresses. Therefore, we add a default route to prevent the firewall from dropping the packets.

On the **attacker**, do the following:

- Disable routing with the command:

  ```
  sysctl -w net.ipv4.ip_forward=0
  ```

  Since the firewall is configured with a default route via the attacker, the firewall would send all traffic destined for random destinations to the attacker. The attacker would reply with destination unreachable when it receives the traffic with random destination addresses because the IP routing is enabled by default on all containers. . Thus, we disable routing to prevent the attacker to send replies to the outside-host.

At this point, the containers are configured and you are ready to begin the attack. Now, do the following and answer the questions below:

- On the attacker, use the netwox 76 command to generate SYN flood attacks on the telnet service of the outside-host using IP spoof at IP4/IP6 level only. Also, answer the first question below.
- Let the attack runs for five seconds. Then, on the outside-host, check all TCP sockets with the command:

  ```
  ss -tan
  ```

  Study the output and answer the next three questions below.
- On the inside-host, try to connect to telnet service on the outside-host with the command:

  ```
  telnet 10.0.10.2
  ```

  Observe what happens and answer the remaining three questions below.

Question 1

3 pts

What is the exact netwox 76 command that you need to generate SYN flood attacks on the telnet service of the outside-host using IP spoof at IP4/IP6 level only?

`netwox 76 -i` 10.0.10.2

`-p` 23

`-s` raw

Question 2

1 pts

What are the states of the outstanding telnet connections from the random IP addresses on the outside-host?

- ○ LAST_ACK
- ○ CLOSED
- ○ SYN_SENT
- ○ FIN_WAIT1
- ○ CLOSE_WAIT
- ○ ESTABLISHED
- ○ FIN_WAIT2
- ○ TIME_WAIT
- ◉ SYN_RECV
- ○ CLOSING
- ○ LISTEN

Question 3 1 pts

How many outstanding telnet connections from the random IP addresses on the outside-host?

64

Question 4 1 pts

What is the parameter that limits the number of outstanding telnet connections in this case?

- ○ The transmission queue length specified on the network interface (qlen shown in "ip addr" output)
- ○ The Recv-Q size of the listening application
- ○ The maximum TCP backlog specified in /proc/sys/net/ipv4/tcp_max_syn_backlog
- ○ The maximum backlog of the network interface card specified in /proc/sys/net/core/netdev_max_backlog
- ◉ The Send-Q size of the listening application

Question 5 1 pts

Can the inside-host access the telnet service on the outside-host while the attacker preforms the SYN flooding attack?

- ○ No
- ◉ Yes

Question 6 1 pts

If the inside-host can access the telnet service, then what do you think is the main reason that the SYN flooding attack does not work?

- ○ Because the ouside-host can always accept new connections from a new source IP address.
- ○ Because a SYN flooding attack only targets a Web service on HTTP protocol and won't work on a telnet service.
- ◉ Because by default a countermeasure to SYN flooding attack is enabled.
- ○ Because the target of a SYN flooding attack must reside on a local area network (LAN).

Question 7 1 pts

Assume that the SYN flooding attack to the telnet server uses a spoofed source IP address that belongs to a legitimate computer. When this legitimate computer, which has never sent the connection request, receives the SYN+ACK message from the telnet server. What will this computer do?

- ○ Send an ACK message to the telnet server to complete the connection.
- ◉ Send RST message to the telnet server to close the half-open connection.
- ○ Send a SYN message to initiate a new connection to the telnet server.
- ○ Send a FIN message to the telnet server to close the connection.

On the inside-host, terminate the telnet session by pressing `Ctrl-C`.

On the outside-host, disable the SYN cookie mechanism with the command:

```
sysctl -w net.ipv4.tcp_syncookies=0
```

Now, from the inside-host, try to telnet to the outside-host again and answer the following questions.

Question 8

1 pts

Can the inside-host access the telnet service on the outside-host while the attacker preforms the SYN flooding attack?

- (●) Yes
- ( ) No

Question 9

2 pts

Select statement(s) that describes SYN cookies.

- [x] SYN cookies can be used to counter SYN flooding attacks.
- [ ] Attacker can still use ACK flooding to trigger the server to allocate more resources for a completed connection.
- [ ] The main drawback of SYN cookies is that it requires the server to allocate more resources for the connections.
- [ ] A SYN cookie is initiated by a client when establising a connection to the server.
- [x] A SYN cookie is an initial sequence number sent in the SYN-ACK.

Question 10

1 pts

Assume that a server uses a SYN cookie mechanism. After the server receives a SYN message from a client, the server calculates a SYN cookie value 12345 and sends it to the client in a SYN+ACK message.

How will client verify itself to the server?

- ( ) By sending an ACK packet with value 12346 in the acknowledgement field to the server.
- (●) By sending an ACK packet with value 12345 in the sequence number field to the server

- ◯ By open a new connection to the server with a source port 12345 to send an ACK packet.
- ◯ By sending an ACK packet to the server on the destination port 12345.

1.2 TCP reset attack

As described in **RFC 3360** (https://tools.ietf.org/html/rfc3360) , "*TCP uses the RST (Reset) bit in the TCP header to reset a TCP connection. Resets are appropriately sent in response to a connection request to a nonexistent connection, for example. The TCP receiver of the reset aborts the TCP connection, and notifies the application.*" However, an adversary can exploit TCP resets to terminate established TCP connections and cause interruptions to the ongoing communications. For example, an attacker can inject a forged TCP reset packet to a BGP speaker, forcing the BGP speaker to drop the BGP routing information and affecting routing paths throughout the Internet.

Your main objective as an attacker is to use the TCP reset attack to terminate an established telnet connection between the inside-host and the outside-host. You will use netwox tool number 40 to send a forge TCP reset packet to the inside-host and observe the results. Note that we cannot use netwox tool number 78 that resets every TCP session matching a filter since the attacker cannot sniff traffic destined to other containers in our setup.

Now, do the following

- On the **lab VM**, start wireshark capture on the lxc-int-br interface
- On the **inside-host**, start a telnet connection to the outside-host. Once the session is established, you should see a login prompt. Make sure to log in using the following credentials:
  USERNAME: ubuntu
  PASSWORD: ubuntu
  After logging in, you should see the welcome message and a prompt on the outside-host.
- On **Wireshark**, study the last TCP message for the ongoing telnet session in the capture screen. Also, select this packet, then press right-click on the mouse and choose to **mark this packet**.
  You will need the actual TCP sequence number when you forge a TCP reset message. However, Wireshark, by default, displays the TCP sequence number as a relative number (i.e., the actual sequence number minus the initial sequence number). To see the actual sequence number in a packet, you must change the settings by right-click on the packet list pane, select `Protocol Preferences`, and uncheck `Relative sequence numbers`.

- On the **attacker**, use netwox 40 to forge a TCP reset message from the outside-host to the inside-host and answer the questions below. Keep in mind that you must correctly construct the TCP reset message to successfully perform a TCP reset attack.

Question 11

10 pts

Match the correct descriptions for different parameters of netwox tool number 40.

-j	IP4 TTL
-l	IP4 source
-m	IP4 destination
-o	TCP source port
-p	TCP destination port
-q	TCP seqnum
-r	TCP acknum
-z	set TCP ACK bit
-A	set TCP PSH bit
-B	set TCP RST bit
-C	set TCP SYN bit

-D	set TCP FIN bit
-E	TCP window
-H	TCP data

Question 12

3 pts

Fill in the missing information for the netwox 40 command to forge a TCP reset message from the outside-host to the inside-host below:
NOTE: We do not ask you to fill in <DST-PORT>, <SEQ_NO> and <ACK_NO> in this question since we do not know in advance the correct values that you need to use. However, you must use the correct information for a successful TCP reset attack.

```
netwox 40 --ip4-ttl 64 --ip4-src 10.0.10.2
 --ip4-dst 10.0.20.2
 --tcp-src 23
 --tcp-dst <DST-PORT> --tcp-seqnum <SEQ_NO> --tcp-acknum <ACK_NO> -z -B --spoo
fip raw
```

Question 13

1 pts

Unlike telnet connections that send message in clear text, SSH connections are encrypted.

Would you be able to perform the TCP reset attack on an SSH connection in the same way as with a telnet connection?

- ◯ Yes. Because SSH encrypts only the data in TCP packets, not the header.

- ○ No. Because the attacker requires to have a secret key to encrypt the TCP parameters when creating a forged message.
- ○ No. Because it is not possible to see TCP related parameters of an SSH connection.
- ◉ Yes. But you can only do it by guessing the TCP sequence number using a brute-force attack.

Once you successfully perform a TCP reset attack, the telnet session is terminated and the message below should appear on the inside-host's terminal:

```
Connection closed by foreign host.
```

On Wireshark, you should observe the forged TCP reset message. Also, select this packet, then press right-click on the mouse and choose to **mark this packet**.

At this point, you should have two packets marked on Wireshark. Now, do the following:

- Stop the Wireshark capture.
- From Wireshark's `File` menu, select `Export Specified Packets...`
- In the Packet Range section, select `Marked packets only`.
  In the "File name:" box, type in `reset` (all characters in small letters)
  Then, click `Save`.

Now, you should have a file called `reset.pcapng` that contains two packets on the home directory of your lab VM. You will submit this file as part of Task 1 submission after you complete the Task 1 assignment.

IMPORTANT: This is an individual assignment and we expect every student to submit a file with unique packets. Students who submit files with the same packets are considered cheating.

1.3 TCP session hijacking

TCP is a connection-oriented protocol and treats data as a stream, so each octet in the TCP session has a unique sequence number, identifying its position in the stream. Although a TCP connection is normally established between two end-hosts, and it is supposed to be used by these two hosts. An attacker may utilize a TCP sequence prediction attack to hijack the established connection.

Your main objective as an attacker is to hijack a TCP connection that is established between the inside-host and the outside-host. For ease of understanding, we will use netcat instead of telnet in this experiment. You will run a netcat server that is actively listening for a connection on the outside-host and use netcat on the inside-host to connect to the netcat server. Then, you will use netwox tool number 40 to send a forge TCP packet from the attacker to the outside-host to hijack the netcat connection.

Now, do the following:

- On the **lab VM**, restart Wireshark capture on the lxc-int-br interface
- On the **outside-host**, start a netcat server that listens on port 9090 with the command:

  ```
  netcat -l 10.0.10.2 9090
  ```

- On the **inside-host**, start a netcat connection to the outside-host with the command:

  ```
  netcat 10.0.10.2 9090
  ```

  Once the connection is established, you can send a message from the inside-host to the outside-host by simply writing the message on the inside-host. Note that the message is buffered and will be sent only after you enter a new line (by pressing the enter key).
  Now, on the inside-host, type "hello", then press enter and observe the netcat server on the outside-host:

  ```
  hello
  ```

  Also, on **Wireshark**, **mark the last TCP message** that is related to the established netcat connection and answer the first question below.
- On the **attacker**, use netwox 40 to forge a TCP message to the outside-host to hijack the netcat connection. The message should contain the word **"HIJACK" written in capital letters followed by a special character for "newline"**. You will also **set the TCP window to 502**. Keep in mind that you must correctly construct the TCP message to successfully perform the TCP session hijack. You can obtain relevant parameters from the Wireshark capture.
  You will also answer the next two questions related to the netwox 40 command below.
- Once you successfully hijack the TCP connection, you should see the "HIJACK" message on the outside-host, as shown below:

  ```
  root@outside-host:~# netcat -l 10.0.10.2 9090
  hello
  HIJACK
  ```

- On **Wireshark**, there should be two new TCP messages that were generated after the last message you have marked. **Mark the two new TCP messages**,

study the messages, and answer the next two questions below.

- On the **inside-host**, try to send a message "hi" from the netcat session. Observe the netcat output on the outside-host.
  Also, On **Wireshark**, you should see many new TCP messages. **Mark the first two TCP messages**, study the message, and answer the remaining questions below.

Question 14

1 pts

After you send "hello" message from the inside-host, study the last TCP message that you see on Wireshark.

Select the description that is the most suitable for this message.

- ○ The message indicates a connection termination.
- ○ The message is a part of the TCP three-way handshake
- ○ The message is a duplicate acknowledgement.
- ◉ The message is an acknowledge message.
- ○ The message triggers a retransmission of unacknowledged packet.
- ○ The message contains the "hello" message sent to the outside-host.

Question 15

5 pts

You will use the netwox 40 command below to forge a TCP message to the outside-host to hijack the netcat connection. Fill in the missing parts of the command.

```
netwox 40 -j 64 -l [10.0.20.2]
-m [10.0.10.2]
```

`-o <SKIP1> -p` 9090

`-q <SKIP2> -r <SKIP3> -a raw -E` 502

`-H` 48494a41434b0a0

`<TCP_FLAGS>`

In this question, we skip some values (with names begin with SKIP) since we do not know them in advance. However, you will need to fill in the correct values when you run the netwox 40 command to successfully hijack the connection.

Moreover, the command above is still not complete. You need to specify which TCP flags (denoted as <TCP_FLAGS>) that you need to set in the forged TCP message. You will answer which TCP flags you need in the next question.

Question 16

2 pts

Choose all TCP flags that you must set in the forged message above.

- [ ] SYN: Synchronization flag
- [ ] No answer text provided.
- [ ] FIN: Finished flag
- [ ] No answer text provided.
- [ ] ECE: ECN-Echo flag
- [ ] URG: Urgent flag
- [ ] NS: Nounce sum flag
- [ ] RST: Reset flag
- [x] PSH: Push flag
- [ ] CWR: Congestion window reduced flag
- [x] ACK: Acknowledgment flag

Question 17

1 pts

Assume that you use the correct parameters to hijack the TCP connection, observe the forged message that you sent.

What is the TCP segment length of the forged message?

7

Question 18

1 pts

On Wireshark, change the protocol preference to use relative sequence numbers.

Look at the netcat server's response to your forged message. What is the relative acknowledgment number in the message?

14

Question 19

1 pts

When you sent a message "hi" from the netcat session on the inside-host, did the message show up on the outside-host?

- ○ Yes
- ◉ No

Question 20

1 pts

Look at the Wireshark info column of the first TCP message that was generated by the inside-host when you sent the message "hi", you should see that it was

marked as "TCP Spurious Retransmision".

What is the best description for "TCP Spurious Retransmission" in this context?

- ○ It means that the TCP retransmission mechanism is disabled.
- ◉ It means that Wireshark has seen the ACK for the data already.
- ○ It means that the message is lost and the inside-host should send the message again.
- ○ It means that an error occurs and the message should be retransmitted.

Question 21

1 pts

Look at the Wireshark info column of the second TCP message, which is a response to the "hi" message sent by the inside-host, you should see that it was marked as "TCP DUP ACK".
Also, observe the relative acknowledgment number of this message and the next sequence number in the previous message.

What is the best description for "TCP DUP ACK" in this context?

- ○ It means Wireshark detects duplicated messages and marks all messages with "TCP DUP ACK".
- ○ It means Wireshark observes that the segment number has already been acknowledged.
- ○ It means Wireshark detects that a TCP retransmission is needed.
- ◉ It means Wireshark observes that an acknowledgement message was duplicated.
- ○ It means Wireshark observes that an acknowledgement message is delayed.

You may observe that the netcat connection on the inside-host hangs after it was being hijacked. You can resolve this by doing the following:

- On the lab VM, open a new terminal and make a second connection to the inside-host with the command:

```
lxc exec inside-host /bin/bash
```

- From this second connection, kill the netcat client process that hung on the first connection with the command:

```
killall netcat
```

- Now, the first connection to the inside-host should be responsive, and you can close the second connection to the inside-host with the command:

```
exit
```

On Wireshark, you should have five packets marked. You can now export the marked packets and save them to a file named: `hijack` (all characters in small letters).

Then, you should have a file called `hijack.pcapng` that contains five packets on the home directory of your lab VM. You will submit this file as part of Task 1 submission after you complete the Task 1 assignment.

IMPORTANT: This is an individual assignment and we expect every student to submit a file with unique packets. Students who submit files with the same packets are considered cheating.

At this point, you are done with Task 1. You should also have two capture files, i.e., `reset.pcapng` and `hijack.pcapng` that you need to submit as part of Task 1. You can now proceed to the Task 1 submission page and follow the instructions to create a tarball for uploading to Canvas.

After you are done with the submission, if you want to continue with Task 2, you may proceed directly to Task 2 instructions. Otherwise, you can follow the clean-up instructions below to shut down the containers and lab VM.

Cleaning up

Shut down all containers by running the command below on each of terminal that is connected to a container:

```
poweroff
```

It may take some time before the container is actually powered off. You can also try to run the command again if the container does not power off.

After all containers are powered off, you can shut down the lab VM.

Saved at 10:42 Submit quiz