Cybersecurity Lab II

Lab 9

TCP Attacks

The goal of the lab is to reproduce and launch:

(a) TCP SYN flooding attack,
(b) TCP RST attack, and
(c) TCP session hijacking attack.

1. Environment (20%)

Setup. You need to run three VMs to perform this lab: an attacker, a victim, and an observer. For the purposes of this assignment, all three VMs should be placed on the same LAN.

A Vagrantfile has been shared with you. The file will allow you to run three machines, box1, box2, and box3. For this assignment, box2 is the Man-in-the-Middle attacker who should have sniffing and interception capabilities. By default, the vagrant boxes are configured with username "vagrant" and password "vagrant". You can turn on and access box1 with:

```
$ vagrant up box1

$ vagrant ssh box1
```

You need to set up the machines (as superuser) so that:

- Whenever box1 and box3 send IP packets to each other, they must pass through box2. To do this, you will need to set up the routing table (type “`ip route`”) correctly. For box1 and box3, add an ip route ("`ip route add`") to each other via the IP of box2.
- You need to enable IP forwarding (for IPv4) in box2. This is done by changing a kernel parameter.
- You need to stop box2 from sending ICMP redirects. This is also done by changing a kernel parameter for the specific network interface.

If you made a mistake, it is possible that either box1 or box3 would receive a redirected route that is cached, preventing further progress. You can observe if such a route exists using "`ip route get <targetip>`", and remove the route with "`ip route flush cache`".

If you accidentally broke the network connectivity of a machine (this would kill the SSH connection as well), you can restart the machine with “`vagrant reload box1`”.

Demonstrate that you have succeeded with the proper screenshots: show routing tables and tcpdump logs.

2. Tasks

Task 1: TCP SYN Flooding Attack (30%)

TCP SYN flooding is a DoS attack that sends many SYN packets to the victim machine (e.g., a web server). The attacker does not complete the three-way handshake protocol, compromising the target's resources with spoofed packets.

In this task, you need to reproduce and demonstrate the SYN flooding attack.

- box1 is running a telnet server. This can be installed with "`sudo apt-get install telnetd`". You can also observe half-open connections on box1 with "`netstat -tna`".
- box2 floods box1 with SYN packets. You need to use scapy to construct these packets.
- box3 will attempt to access box1 during the attack as a telnet client.

SYN Cookie Countermeasure. SYN cookie is a defense mechanism to counter the SYN flooding attack. The mechanism detects that the machine is under the SYN flooding attack. You can use the `sysctl` command to turn on/off the SYN cookie mechanism:

```
$ sudo sysctl -w net.ipv4.tcp_syncookies=0  # (turn off SYN cookie)
$ sudo sysctl -w net.ipv4.tcp_syncookies=1  # (turn on SYN cookie)
```

You need to run your attacks with the SYN cookie mechanism on and off, and compare the results. **Submit the scapy code you run on box2 as tcp_synflood.py.**

The attack will not succeed if box2 closes connections after receiving a SYN-ACK from box1. You can tell its firewall to drop these RSTs by adding a rule to iptables:

```
$ sudo iptables -A OUTPUT -p tcp --tcp-flags RST RST -s 192.168.56.12 -j DROP
```

You may also want to change the kernel parameter "net.ipv4.tcp_max_syn_backlog", the maximum number of half-open connections, to demonstrate the attack.

Questions

For each case (SYN cookie on/off):
(1.a) Describe how you know whether the attack is successful or not.
(1.b) Show the outputs of "`netstat -tna`"; compare the two results, noting observations.
(1.c) What happens to TCP sessions (e.g., `telnet`) that were established before the attack?

Task 2: TCP RST Attack (20%)

TCP RST attack terminates an existing connection between two TCP endpoints. The attacker, box2, can spoof a RST packet from box1 to box3, breaking this existing connection. To succeed in this attack, attackers need to correctly construct the TCP RST packet.

In this task, you need to launch a TCP RST attack to break an existing `telnet` connection between A and B. After that, try the same attack on an `ssh` connection. To simplify the lab, we assume that the attacker and the victims are on the same LAN.

<u>Use scapy to construct and send the TCP RST packet, and submit your code as tcp_rst.py.</u>

Questions

(2.a) What header fields did you need to modify? How did you calculate these fields? Show screenshots of the steps you carried out to calculate the fields.
(2.b) Show screenshots of the `telnet` and `ssh` sessions after launching the attack.
(2.c) Did your attack on an `ssh` session succeed? How is this related to the encryption done by `ssh`? Explain.

Task 3: TCP Session Hijacking Attack (30%)

The goal of this attack is to inject malicious contents into an existing TCP session. If this is a `telnet` session, attackers can inject malicious commands (e.g., deleting an important file) into this session, causing the victims to execute the malicious commands.

In this task, you need to demonstrate how you can hijack a `telnet` session between two computers. Your goal is to get the `telnet` server to delete an existing file.
Execute the following command on the server machine:

```
$ touch ~/myfile.txt
```

The injected command is: `rm ~/myfile.txt`

Use scapy to hijack the existing `telnet` session, and submit your code as tcp_session.py.

Questions

(3.a) What header fields did you need to modify? How did you calculate these fields? Show screenshots of the steps you carried out to calculate the fields.
(3.b) Show and explain a screenshot of packet capture after launching the attack, demonstrating that it is successful.
(3.c) Can you use this attack to hijack an `ssh` session? Explain.

3. Submission

You are required to submit:
(1) All source code you implemented to complete the tasks.
(2) A detailed report.
The files should be compressed in a single (.zip) archive.
If you are not able to set up the environment, you may still receive full credit for the other tasks by setting up a reasonable attack environment; be sure to explain.