Network Security Tasks

Introduction

This assignment includes three parts:

1. Network Packet Analysis (25 pts).
2. TCP Reset Attack (35 pts).
3. DNS Poisoning Attack (40 pts).

In the second and third tasks, you are asked to conduct the corresponding attack in simulated network environment. As the virtual environment is built via Docker, you will need to install Docker in a Linux system. Please refer to this link https://docs.docker.com/engine/install/ to check out how to install Docker in Linux system.

Scripting Cheat Sheet

As the attack script is asked to be written in Python to manipulate the network packets, here is a Python example for sniffing packets with Scapy.

Install Scapy Library:

Scapy will be used in the assignment for packet manipulation, you can install it via:

```
$ python3 -m pip install scapy
```

Packet Sniffing:

```python
from scapy.all import * # Import anything from scapy library

target_src_ip = "1.2.3.4"
target_dst_ip = "5.6.7.8"

def send_rst(packet):
    pass

sniff(iface="br-123456789", filter=f"tcp and host {target_src_ip} and
{target_dst_ip}", prn=send_rst)
```

Scapy's `sniff` method can help us with sniffing the packet in given network interface with customized filter. For the above instance, three arguments are passed:

1. **iface:** The interface where the packets are captured. The interfaces' name can be viewed via `ifconfig -a` in Linux.
2. **filter:** It indicates what kind of packets will be captured. In this example, Scapy only captures the TCP packets with source ip address "1.2.3.4" and destination ip address "5.6.7.8".
3. **prn:** Callback function for each captured packet. This function will be called for every

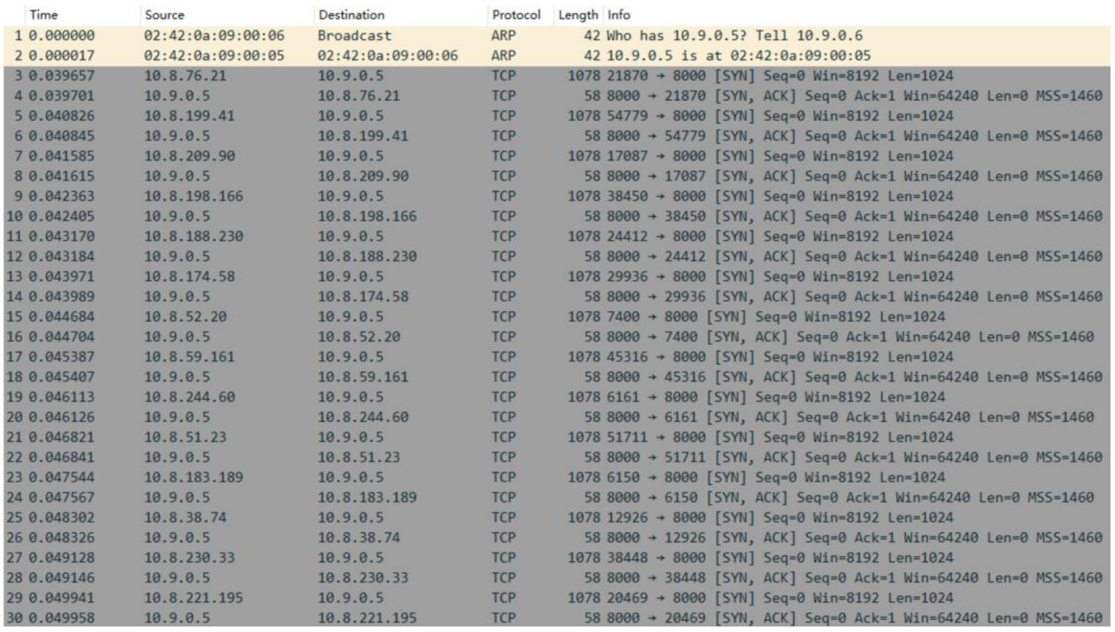
captured packet.

The complete instruction of Scapy’s `sniff` function can be viewed here https://scapy.readthedocs.io/en/latest/usage.html.

Network Packet Analysis (25 pts)

Description

Alice has a remote cloud server, but one day she suddenly found she failed to get access to the service deployed on the server. The IT manager captured the network traffic pcap file shown as follow. Can you help to find out what happened?



	Time	Source	Destination	Protocol	Length	Info
1	0.000000	02:42:0a:09:00:06	Broadcast	ARP	42	Who has 10.9.0.5? Tell 10.9.0.6
2	0.000017	02:42:0a:09:00:05	02:42:0a:09:00:06	ARP	42	10.9.0.5 is at 02:42:0a:09:00:05
3	0.039657	10.8.76.21	10.9.0.5	TCP	1078	21870 → 8000 [SYN] Seq=0 Win=8192 Len=1024
4	0.039701	10.9.0.5	10.8.76.21	TCP	58	8000 → 21870 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
5	0.040826	10.8.199.41	10.9.0.5	TCP	1078	54779 → 8000 [SYN] Seq=0 Win=8192 Len=1024
6	0.040845	10.9.0.5	10.8.199.41	TCP	58	8000 → 54779 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
7	0.041585	10.8.209.90	10.9.0.5	TCP	1078	17087 → 8000 [SYN] Seq=0 Win=8192 Len=1024
8	0.041615	10.9.0.5	10.8.209.90	TCP	58	8000 → 17087 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
9	0.042363	10.8.198.166	10.9.0.5	TCP	1078	38450 → 8000 [SYN] Seq=0 Win=8192 Len=1024
10	0.042405	10.9.0.5	10.8.198.166	TCP	58	8000 → 38450 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
11	0.043170	10.8.188.230	10.9.0.5	TCP	1078	24412 → 8000 [SYN] Seq=0 Win=8192 Len=1024
12	0.043184	10.9.0.5	10.8.188.230	TCP	58	8000 → 24412 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
13	0.043971	10.8.174.58	10.9.0.5	TCP	1078	29936 → 8000 [SYN] Seq=0 Win=8192 Len=1024
14	0.043989	10.9.0.5	10.8.174.58	TCP	58	8000 → 29936 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
15	0.044684	10.8.52.20	10.9.0.5	TCP	1078	7400 → 8000 [SYN] Seq=0 Win=8192 Len=1024
16	0.044704	10.9.0.5	10.8.52.20	TCP	58	8000 → 7400 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
17	0.045387	10.8.59.161	10.9.0.5	TCP	1078	45316 → 8000 [SYN] Seq=0 Win=8192 Len=1024
18	0.045407	10.9.0.5	10.8.59.161	TCP	58	8000 → 45316 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
19	0.046113	10.8.244.60	10.9.0.5	TCP	1078	6161 → 8000 [SYN] Seq=0 Win=8192 Len=1024
20	0.046126	10.9.0.5	10.8.244.60	TCP	58	8000 → 6161 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
21	0.046821	10.8.51.23	10.9.0.5	TCP	1078	51711 → 8000 [SYN] Seq=0 Win=8192 Len=1024
22	0.046841	10.9.0.5	10.8.51.23	TCP	58	8000 → 51711 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
23	0.047544	10.8.183.189	10.9.0.5	TCP	1078	6150 → 8000 [SYN] Seq=0 Win=8192 Len=1024
24	0.047567	10.9.0.5	10.8.183.189	TCP	58	8000 → 6150 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
25	0.048302	10.8.38.74	10.9.0.5	TCP	1078	12926 → 8000 [SYN] Seq=0 Win=8192 Len=1024
26	0.048326	10.9.0.5	10.8.38.74	TCP	58	8000 → 12926 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
27	0.049128	10.8.230.33	10.9.0.5	TCP	1078	38448 → 8000 [SYN] Seq=0 Win=8192 Len=1024
28	0.049146	10.9.0.5	10.8.230.33	TCP	58	8000 → 38448 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
29	0.049941	10.8.221.195	10.9.0.5	TCP	1078	20469 → 8000 [SYN] Seq=0 Win=8192 Len=1024
30	0.049958	10.9.0.5	10.8.221.195	TCP	58	8000 → 20469 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460

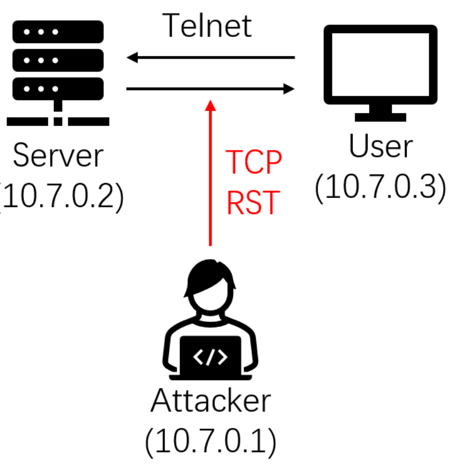
Questions

- (4 pts) What's the server's IP address?
- (4 pts) What's the attacker's real IP address?
- (5 pts) What kind of attack is the server suffering from?
- (12 pts) Explain the attack happened in this scenario.

TCP Reset Attack (35 pts)

Description

Consider the following network graph:



User and server has built up a stable TCP connection via Telnet. As an attacker in this LAN, you are asked to maliciously shut down the connection using TCP reset attack.

Environment Setup

Please find the attachment **"TCP-Reset-Attack.zip"**. Before starting the attack, you should do the following first:

1. Decompress the zip file.
2. Change directory to where "docker-compose.yml" is located.
3. Run `docker-compose up` in command line.
4. Run `docker ps -a` in command line to check the containers' status (all of the three containers should be "up" at this step).
   a) Please ensure all the three containers are successfully deployed before moving on.
5. Switch into "user1-tcp-rst" container via `docker exec -it user1-tcp-rst bash` and start TCP connection to server via `telnet 10.7.0.2`.
   a) You need to log in the telnet with username (<u>user</u>) and password (<u>user</u>).
6. If you log in successfully, you can execute any command in the telnet panel.

Questions

Please answer the following questions in your report:

1. (5 pts) How does TCP reset attack work?
2. (10 pts) After building up a successful TCP connection via Telnet between user and server (step 5 in Environment Setup), execute `cat /home/user/flag` to get the file content. You need to provide a screenshot for this question.
3. (15 pts) Write a Python script to conduct TCP reset attack and explain how the code's

workflow is. Your code should be able to:

   a) Automatically sniff the packets in the LAN.
   b) Automatically generate the attack payload and send to ruin the TCP connection.

4. (5 pts) Suggestion an effective approach to detect and prevent TCP reset attack.

Tips:

- You should conduct the attack in the container \`attacker-tcp-rst\`.
- Your attack script should be written in Python code,
- After all the containers are up, you can transfer the files between your host and \`attacker-tcp-rst\` in the shared \`volumes\`, which is located in \`/volumes\` in \`attacker-tcp-rst\`.

DNS Poisoning Attack (40 pts)

Description

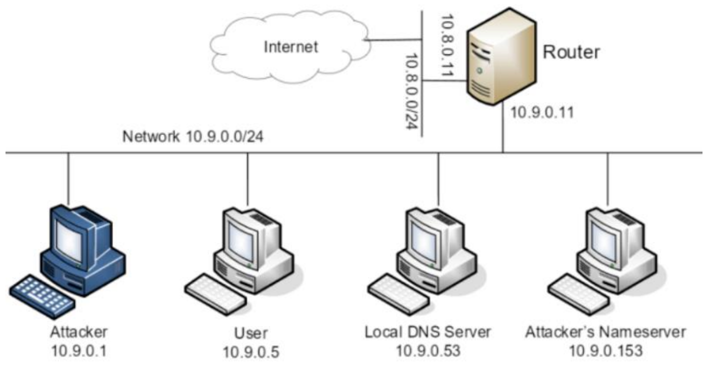
In this task, you are an attacker that has already invaded into a network environment that hosts a DNS server. You are required to conduct the DNS poisoning to attack the DNS server to trick victims who query this server for domain addresses.
We provided a script template, which currently monitors the current network to sniff all DNS packets and outputs their information to you.
You are required to complete the script to poison the DNS server and verify that, when a user queries a certain domain `www.example.com`, the DNS server would response with the address poisoned by the attacker (i.e., your script) instead of the correct address.

Environment Setup

Please find the attachment **"DNS-Poisoning-Attack.zip"**.
Decompress the zip file, change directory to where "docker-compose.yml" is located and type `docker-compose up`. It starts several docker containers to simulate the DNS server's machine, the victim user's machine, and the attacker's machine. Type Ctrl-C in the same terminal will stop the running containers.

Useful commands:

- `dig`: query a domain address, as taught in the class.
- Start a terminal to enter the user (victim)'s machine: `sudo docker exec -it victim-user bash`
- Start a terminal to enter the DNS server's machine: `sudo docker exec -it local-dns-server bash`
    - The only allowed operation on the DNS server's machine is `rndc flush`, which clears the DNS server's cache.
- Start a terminal to enter the attacker's machine: `sudo docker exec -it attacker bash`
    - Running the script: `python3 /volume/attack.py`
        - Running the provided script directly would start to monitor the network for sniffing DNS packets.
    - The script is placed in /volume in the attacker's machine, which are also mapped to the volume/ folder outside the docker containers. You can modify the script in volume/ to complete the attack and run it in the container.



Questions

- (10 pts) Please start the original script in the attack's machine, clear the cache in the DNS server's machine, and enter the user's machine to query the domain `www.example.com` by `dig`. When there is no attack, based on the script's and `dig` output, observe and report the complete DNS query process step-by-step.
- (10 pts) Based on the query process above, please report the process of how to poison the DNS server by sending it forged network packets.
- (20 pts) Please complete the script to send forged packets to the DNS server, explain the code you written in the script (i.e., don't have to modify the function `send_dns`, you should call it with proper arguments and explain them), and verify that the user has been tricked to visit a wrong address when querying for ` www.example.com `.