Data Communication and Computer Network Laboratory

Master of Computer Application
Second Year, First Semester
Session: 2023-24

Assignment - V

Date: 24/08/2023

Packet Crafting

Packet crafting, is the process of manually creating and customizing network packets with specific headers, payloads, and other fields. This technique is commonly used in network security, network testing, and network debugging scenarios. In this exercise you will be using the `scapy` library to create custom packets from scratch. The `scapy` library provides a high-level interface to construct packets, allowing you to set fields in various network layers, such as Ethernet, IP, TCP, UDP, and more.

Example 1: How to craft a custom packet with an Ethernet layer, IP layer, and TCP layer using `scapy`:

```
from scapy.all import Ether, IP, TCP

# Create an Ethernet layer with source and destination MAC addresses
eth_layer = Ether(src="00:11:22:33:44:55", dst="ff:ff:ff:ff:ff:ff")

# Create an IP layer with source and destination IP addresses
ip_layer = IP(src="192.168.1.100", dst="8.8.8.8")

# Create a TCP layer with source and destination ports
tcp_layer = TCP(sport=12345, dport=80)

# Combine all layers to create the custom packet
custom_packet = eth_layer / ip_layer / tcp_layer

# Show the summary of the custom packet
print(custom_packet.summary())

# Show the detailed information of the custom packet
print(custom_packet.show())
```

You can also set various attributes for each layer according to your requirements. The "/" operator is used to stack the layers and create the final packet. After creating the custom packet, you can send it over the network using `scapy`'s send() function:

```
from scapy.all import send

send(custom_packet)
```

Example 2: How to send an ICMP packet to a destination:

```
send(IP(dst="192.168.1.16")/ICMP()/"HI")
```

Example 3: How to send packets and receive answers.

`sr()/sr1()/srp()` functions are used for sending packets and receiving answers. The function `sr1()` is a variant that only returns one packet that answered the packet (or the packet set) sent. The packets must be layer 3 packets (IP, ARP, etc.). The function `srp()` do the same for layer 2 packets (Ethernet, 802.3, etc.). If there is no response, a None value will be assigned instead when the timeout is reached.

Further details on scapy usage can be found at https://scapy.readthedocs.io/en/latest/index.html

Now, use `scapy` to craft packets and perform the following tasks:

1. Write a python program which gets a network address from the user and generates all possible host IP addresses within that network. Then it sends dummy ICMP echo request message to all the hosts. Display only those hosts from which you receive corresponding ICMP echo reply message within a predefined time out period.
2. Write a python program which gets a host IP address from the user and sends TCP SYN segments to all the ports within the range 0 to 1023. Display only those port numbers from which you receive corresponding TCP SYN+ACK segment within a predefined time out period.

Your report should contain at least the following sections:

1. Problem Statement
2. Your solution approach, data structures used
3. Source code (with appropriate comments)
4. Sample run