Name: Shubhan Singh

Class: SE Comps B

Roll no.: 2022300118

CCN Experiment 7

Packet Crafting

Aim: Packet Crafting using Scapy

Objectives:

1. Gain hands-on experience with Scapy, a Python-based packet manipulation tool.
2. Understand the functionality and significance of protocols at the Application and Transport Layers.
3. Analyze and dissect packets to comprehend the structure and contents of different protocols.
4. Investigate the interaction between Application and Transport Layer protocols.
5. Develop skills in crafting custom packets for specific networking scenarios.

Problem Statement:

Kindly craft the following packets. Take a screenshot of results. Also take screenshots of the crafted packet you send.

1. **Ping (ICMP Echo Request)**:

- Craft an ICMP Echo Request packet using Scapy.
- Send the packet to a target IP address.
- Expect an ICMP Echo Reply packet in response from the target.

Code:

```python
from scapy.all import *

def ping(ip):
    icmp_request = IP(dst=ip) / ICMP()
    icmp_response = sr1(icmp_request, timeout=1)

    if icmp_response:
        print(f"Received reply from {ip}")
        print(icmp_response.show())
    else:
        print(f"No reply from {ip}")


ping('142.250.183.174')
```

Output:

```
shubhan@Shubhan: ~/progra

shubhan@Shubhan:~/programs/CCN$ sudo -E python3 scapy_icmp.py
Begin emission:
Finished sending 1 packets.
.*
Received 2 packets, got 1 answers, remaining 0 packets
Received reply from 142.250.183.174
###[ IP ]###
  version   = 4
  ihl       = 5
  tos       = 0x0
  len       = 28
  id        = 0
  flags     =
  frag      = 0
  ttl       = 110
  proto     = icmp
  chksum    = 0xcb36
  src       = 142.250.183.174
  dst       = 172.30.142.227
  \options   \
###[ ICMP ]###
     type      = echo-reply
     code      = 0
     chksum    = 0xffff
     id        = 0x0
     seq       = 0x0
     unused    = ''

None
shubhan@Shubhan:~/programs/CCN$
```

2. UDP Datagram:

- Craft a UDP packet with custom payload using Scapy.
- Send the UDP packet to a target listening on a specific UDP port.
- Expect a response from the target if the port is open and reachable.

Code:

```
from scapy.all import IP, UDP, sr1

def send_udp_packet(ip, port, payload):
    udp_packet = IP(dst=ip) / UDP(dport=port) / payload
    udp_response = sr1(udp_packet, timeout=1)

    if udp_response:
        print(f"Received reply from {ip}:{port}:
{udp_response.show()}")
    else:
        print(f"No reply from {ip}:{port}")

send_udp_packet('8.8.8.8', 53, 'My message')
```

Output:

```
shubhan@Shubhan: ~/progra
shubhan@Shubhan:~/programs/CCN$ sudo -E python3 scapy_udp.py
[sudo] password for shubhan:
Begin emission:
Finished sending 1 packets.
.
Received 1 packets, got 0 answers, remaining 1 packets
No reply from 8.8.8.8:53
```

(No response was received from server)

3. DNS Query:

- Craft a DNS query packet using Scapy to query a DNS server for a specific domain.
- Send the DNS query packet to the DNS server.
- Expect a DNS response containing the IP address associated with the queried domain.

Code:

```
from scapy.all import *

ip='8.8.8.8'
icmp_request = IP(dst=ip) / UDP(dport=53) /
DNS(rd=1,qd=DNSQR(qname='www.google.com'))
icmp_response = sr1(icmp_request, verbose=0)

if icmp_response:
    print(f"Received reply from {ip}")
    print(icmp_response.show())
else:
    print(f"No reply from {ip}")
```

Output: (On next page)

```
shubhan@Shubhan: ~/progra

shubhan@Shubhan:~/programs/CCN$ sudo -E python3 scapy_dns.py
Received reply from 8.8.8.8
###[ IP ]###
  version   = 4
  ihl       = 5
  tos       = 0x0
  len       = 76
  id        = 2365
  flags     =
  frag      = 0
  ttl       = 52
  proto     = udp
  chksum    = 0x3253
  src       = 8.8.8.8
  dst       = 172.30.142.227
  \options   \
###[ UDP ]###
     sport     = domain
     dport     = domain
     len       = 56
     chksum    = 0xeba
###[ DNS ]###
        id        = 0
        qr        = 1
        opcode    = QUERY
        aa        = 0
        tc        = 0
        rd        = 1
        ra        = 1
        z         = 0
        ad        = 0
        cd        = 0
        rcode     = ok
        qdcount   = 1
        ancount   = 1
```

```
###[ DNS ]###
        id        = 0
        qr        = 1
        opcode    = QUERY
        aa        = 0
        tc        = 0
        rd        = 1
        ra        = 1
        z         = 0
        ad        = 0
        cd        = 0
        rcode     = ok
        qdcount   = 1
        ancount   = 1
        nscount   = 0
        arcount   = 0
        \qd        \
         |###[ DNS Question Record ]###
         |  qname     = 'www.google.com.'
         |  qtype     = A
         |  qclass    = IN
        \an        \
         |###[ DNS Resource Record ]###
         |  rrname    = 'www.google.com.'
         |  type      = A
         |  rclass    = IN
         |  ttl       = 288
         |  rdlen     = 4
         |  rdata     = 142.250.183.132
        ns        = None
        ar        = None

None
shubhan@Shubhan:~/programs/CCN$
```

4. HTTP GET Request:

- Craft an HTTP GET request packet using Scapy to retrieve a specific web page from a web server.
- Send the HTTP GET request to the web server.
- Expect an HTTP response containing the requested web page content.

Code:

```
from scapy.all import *
from scapy.layers.http import HTTP, HTTPRequest
from scapy.layers.inet import IP, TCP

web_server = "www.google.com"

ip = IP(dst=web_server)

tcp = TCP(dport=80)

http_get = "GET / HTTP/1.1\r\nHost: " + web_server +
"\r\n\r\n"

packet = ip/tcp/http_get

response = sr1(packet)

if response is None:
    print("No response received.")
else:
    print("Response received:")
    response.show()
```

Output: (on next page)

```
shubhan@Shubhan: ~/progra

shubhan@Shubhan:~/programs/CCN$ sudo -E python3 scapy_http.py
[sudo] password for shubhan:
Begin emission:
Finished sending 1 packets.
.*
Received 2 packets, got 1 answers, remaining 0 packets
Response received:
###[ IP ]###
  version   = 4
  ihl       = 5
  tos       = 0x28
  len       = 44
  id        = 0
  flags     = DF
  frag      = 0
  ttl       = 54
  proto     = tcp
  chksum    = 0xb8e3
  src       = 142.250.193.196
  dst       = 172.30.142.227
  \options   \
###[ TCP ]###
     sport     = http
     dport     = ftp_data
     seq       = 2601157537
     ack       = 1
     dataofs   = 6
     reserved  = 0
     flags     = SA
     window    = 65535
     chksum    = 0xed9e
     urgptr    = 0
     options   = [('MSS', 1370)]

shubhan@Shubhan:~/programs/CCN$
```

5 . Traceroute

- Craft UDP packets with increasing TTL (Time-to-Live) values using Scapy.
- Send these packets towards a destination IP address.
- Observe the ICMP Time Exceeded messages returned by intermediate routers to map the network path to the destination.

Code:

```python
from scapy.all import *

def traceroute(dest_ip, max_hops=30):
    ttl = 1
    while True:
        packet = IP(dst=dest_ip, ttl=ttl) / UDP(dport=33434)
        reply = sr1(packet, verbose=0, timeout=2)
        if reply is None:
            print(f"{ttl}. no reply")
        elif reply.type == 3:
            print(f"{ttl}. {reply.src}")
            break
        else:
            print(f"{ttl}. {reply.src}")
        ttl += 1
        if ttl > max_hops:
            break

traceroute("8.8.8.8")
```

Output:

```
shubhan@Shubhan:~/programs/CCN$ sudo -E python3 scapy_tracert.py
1. 172.30.128.1
2. 192.168.187.139
3. no reply
4. 10.71.5.13
5. 172.26.76.246
6. 172.26.76.226
7. 192.168.53.176
8. no reply
9. no reply
10. 72.14.211.138
11. no reply
12. 8.8.8.8
shubhan@Shubhan:~/programs/CCN$
```