Experiment No. 8

Title: To study analysis of network using Scapy

Batch: B-1 **Roll No: 16010422234** **Name: Chandana Ramesh Galgali**

Experiment No.:8

Aim: To study analysis of network using Scapy Library

Resources needed: Python IDE

Theory:

Traffic analysis is the process of intercepting and analyzing network traffic in order to deduce information from communication. The size of the packets exchanged between two hosts, details of the systems communicating, time and duration of communication are some of the valuable information to an attacker. Following is required to be studied under this experiment:

- Networking basics
- Raw socket programming
- Packet sniffing with Scapy
- Packet injection with Scapy
- Parse DNS traffic with Scapy
- OS fingerprinting with Scapy

Activities:

Students are expected to sit in pair-neighboring PC's and should perform the communication)

Result: (script and output)
Code:

```
from scapy.all import *

def packet_handler(packet):
    if packet.haslayer(IP):
        src_ip = packet[IP].src
        dst_ip = packet[IP].dst
        print(f"Source IP: {src_ip} --> Destination IP: {dst_ip}")
        if packet.haslayer(TCP):
            src_port = packet[TCP].sport
            dst_port = packet[TCP].dport
            print(f"Source Port: {src_port} --> Destination Port: {dst_port}")
```

```
        if packet.haslayer(Raw):
            http_data = packet[Raw].load
            print(f"HTTP Data: {http_data}")
# Set the network interface to monitor
interface = "eth0"
# Start sniffing packets
sniff(iface=interface, prn=packet_handler)
```

Output:

```
Source Port: 33484 --> Destination Port: 8080
Source IP: 172.28.0.12 --> Destination IP: 172.28.0.1
Source Port: 8080 --> Destination Port: 33484
HTTP Data: b'[\xaa\xc2\x0f#\xa4\x92\x03\xbe\xad\xbf\x07\xc8\xa2\x01MP#\x18\xf1`U\x16\x00*r\x17tb\xa7\x02\x9d\xc2\t\xba\xb3\xa4c\xf8\xe9GL\xf1\x05\x0
Source IP: 172.28.0.1 --> Destination IP: 172.28.0.12
Source Port: 33484 --> Destination Port: 8080
Source IP: 172.28.0.12 --> Destination IP: 172.28.0.1
Source Port: 8080 --> Destination Port: 33484
HTTP Data: b'_e\x90\x14\xc1\x02\xf5H\x178\xa2\xc24\x06]TQv\x05\xfa\xe7\xe2\x0f\xfc\xe3\xf7\xa8\x7f\x89\xdf\x84\x9f\x90V\xdf\xb6v\xf1\x1d\xd5\xc1\x8d
Source IP: 172.28.0.1 --> Destination IP: 172.28.0.12
Source Port: 33484 --> Destination Port: 8080
Source IP: 172.28.0.12 --> Destination IP: 172.28.0.1
Source Port: 8080 --> Destination Port: 33484
HTTP Data: b'\xd3\xe8\x7f\xabX\xe0x\xc6\xf7x+o\xcc\xb6WAN\x8a\xf3G\xadU\xa2\x972>\xc4~\x184\xc6\x87\xcf\xd8G\xe4\x0b\x01\x88J\xceR\x93L"\xd0\xbbq\xb
Source IP: 172.28.0.1 --> Destination IP: 172.28.0.12
Source Port: 33484 --> Destination Port: 8080
Source IP: 172.28.0.12 --> Destination IP: 172.28.0.1
Source Port: 8080 --> Destination Port: 33484
HTTP Data: b'``\x82\x1er\xbb\x0e\x8e\xc4\x9a\x96\x08\x8b\xf8\xe8\x1f2\xa5\x01(V\x0f*\x16\xf4\x12\xa4\nN\xc2DlE\x8c\xd1.\x7f\xcf\x10\x90"\xa2\xd1\xfd|
Source IP: 172.28.0.1 --> Destination IP: 172.28.0.12
Source Port: 33484 --> Destination Port: 8080
HTTP Data: b'POST /api/kernels/640271bd-58df-4dd2-9bfa-7679d292aaa0/interrupt HTTP/1.1\r\nHost: colab.research.google.com\r\nUser-Agent: Mozilla/5.0
Source IP: 172.28.0.12 --> Destination IP: 172.28.0.1
Source Port: 8080 --> Destination Port: 33484
<Sniffed: TCP:527 UDP:0 ICMP:0 Other:0>
```

Outcomes: Demonstrate handling database with python and to understand network programming with Python scapy.

Conclusion: (Conclusion to be based on the objectives and outcomes achieved)

The study of network analysis using the Scapy library proved to be a valuable learning experience, equipping us with the necessary skills and knowledge to effectively analyze and understand network traffic, identify potential security risks, and take appropriate measures to ensure network integrity and security.

References:

1. https://www.oreilly.com/library/view/effective-python penetration/9781785280696/ch02.html