

LAB 13

CRYPTOGRAPHY AND **NETWORK SECURITY**

ANALYSE THE DATA PACKET WITH WIRESHARK PACKET TRACER

NAME: OM SUBRATO DEY REG NO.: 21BAI1876

COLAB LINK:

<https://colab.research.google.com/drive/1ONarxyrFl19EDAnCQ7bN6gNofQ4Aa-iZ?usp=sharing>

We will directly be using a downloaded pcap format file from the link accessible:

<https://www.cloudshark.org/captures/767a93d720ad>

So, above link consists of the wireshark file.

The screenshot displays the CloudShark web interface for a capture named 'wireshark-capture-ipsec-ikev2.pcap'. The interface shows a list of four packets, all of which are IKEv2 messages between 192.168.12.1 and 192.168.12.2. Packet 1 is an IKE_SA_INIT Initiator Request, packet 2 is an IKE_SA_INIT MID=00 Responder Response, packet 3 is an IKE_AUTH MID=01 Initiator Request, and packet 4 is an IKE_AUTH MID=01 Responder Response.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.12.1	192.168.12.2	ISAKMP	499	IKE_SA_INIT MID=00 Initiator Request
2	0.001935	192.168.12.2	192.168.12.1	ISAKMP	499	IKE_SA_INIT MID=00 Responder Response
3	0.003885	192.168.12.1	192.168.12.2	ISAKMP	294	IKE_AUTH MID=01 Initiator Request
4	0.005636	192.168.12.2	192.168.12.1	ISAKMP	278	IKE_AUTH MID=01 Responder Response

Below the packet list, the details of the first packet (Frame 1) are shown. It is an Ethernet II frame from Cisco_89:7a:36 to Cisco_9c:e3:20, encapsulating an Internet Protocol Version 4 packet from 192.168.12.1 to 192.168.12.2. The payload is a User Datagram Protocol (UDP) packet with source port 500 and destination port 500, which is an Internet Security Association and Key Management Protocol (ISAKMP) message.

The packet bytes are displayed in hexadecimal and ASCII. The ASCII column shows the beginning of the IKEv2 message structure, including the 'z6..E..' magic number and the 'W...../.....' field.

CODE:

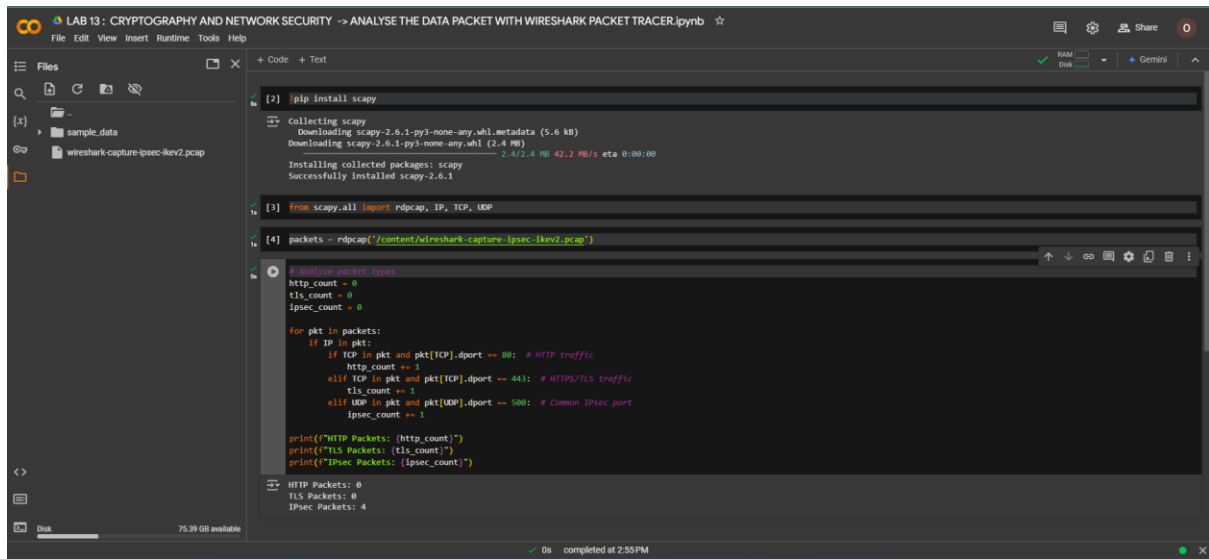
```
!pip install scapy
from scapy.all import rdpcap, IP, TCP, UDP
packets = rdpcap('/content/wireshark-capture-ipsec-ikev2.pcap')

# Analyze packet types
http_count = 0
tls_count = 0
ipsec_count = 0

for pkt in packets:
    if IP in pkt:
        if TCP in pkt and pkt[TCP].dport == 80: # HTTP
            traffic
                http_count += 1
        elif TCP in pkt and pkt[TCP].dport == 443: #
            HTTPS/TLS traffic
                tls_count += 1
        elif UDP in pkt and pkt[UDP].dport == 500: #
            Common IPsec port
                ipsec_count += 1

print(f"HTTP Packets: {http_count}")
print(f"TLS Packets: {tls_count}")
print(f"IPsec Packets: {ipsec_count}")
```

OUTPUT:



The screenshot shows a Jupyter Notebook environment with the following content:

```
[2] !pip install scapy

Collecting scapy
  Downloading scapy-2.6.1-py3-none-any.whl.metadata (5.6 kB)
  Downloading scapy-2.6.1-py3-none-any.whl (21.4 MB)
    ----- 2.4/2.4 MB 42.2 MB/s eta 0:00:00
Installing collected packages: scapy
Successfully installed scapy-2.6.1

[3] from scapy.all import rdpcap, IP, TCP, UDP

[4] packets = rdpcap('/content/wireshark-capture-ipsec-ikev2.pcap')
```

The next cell contains a script to analyze the packet types:

```
# Analyze packet types
http_count = 0
tls_count = 0
ipsec_count = 0

for pkt in packets:
    if IP in pkt:
        if TCP in pkt and pkt[TCP].dport == 80: # HTTP traffic
            http_count += 1
        elif TCP in pkt and pkt[TCP].dport == 443: # HTTPS/TLS traffic
            tls_count += 1
        elif UDP in pkt and pkt[UDP].dport == 500: # Common IPsec port
            ipsec_count += 1

print(f"HTTP Packets: {http_count}")
print(f"TLS Packets: {tls_count}")
print(f"IPsec Packets: {ipsec_count}")
```

The output of the script is displayed at the bottom of the cell:

```
HTTP Packets: 0
TLS Packets: 0
IPsec Packets: 4
```

The status bar at the bottom indicates that the cell was completed at 2:55 PM.