ASSIGNMENT 7

Name: Yogita Mundankar
ROll no. B22CS068

TASK1 : Understanding Replay Attack:

A Replay Attack is a type of cyber attack where an attacker captures a legitimate data transmission and then replays (resends) it to trick a system into performing an action again. These attacks exploit the lack of unique request verification mechanisms in certain protocols.

How Replay Attacks Work?

- Packet Sniffing: An attacker captures network traffic using tools like Wireshark, tcpdump, or Scapy.
- Packet Injection: The attacker resends the captured packets at a later time to replay the legitimate action.
- Exploitation: The target system, believing the replayed packets to be legitimate, processes them again, causing unintended behavior.

Real-World Examples of Replay Attacks

- Authentication Bypass: An attacker replays a valid login request to gain unauthorized access.
- Financial Transactions: A hacker captures and replays a payment request to duplicate a money transfer.
- IoT & Smart Devices: Replay attacks can be used to open smart locks or trigger smart devices remotely.
- Session Hijacking: Replaying session tokens can allow attackers to take over user accounts.

TASK2:

Wireshark was already installed for previous labs.

Installing scapy:

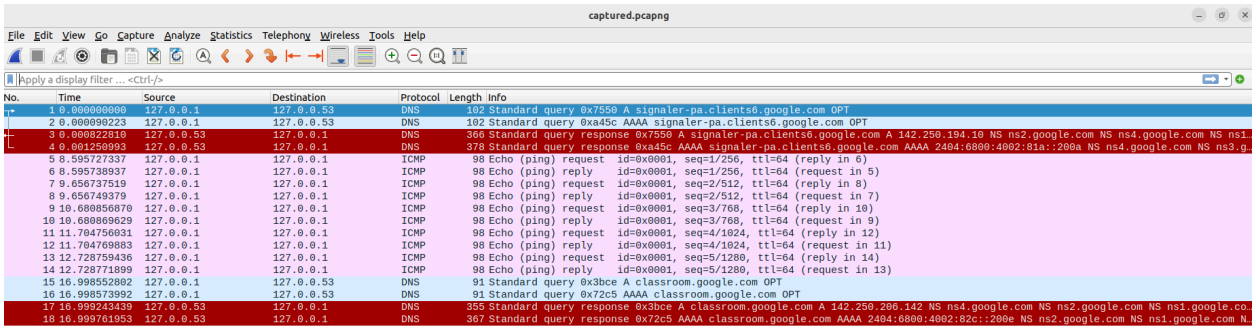
```
ModuleNotFoundError: No module named 'scapy'
yogita@yogita-Inspiron-5502:~/CyberSecurity/Lab7$ sudo python3 -m pip install scapy
Collecting scapy
  Downloading scapy-2.6.1-py3-none-any.whl (2.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.4/2.4 MB 2.8 MB/s eta 0:00:00
Installing collected packages: scapy
Successfully installed scapy-2.6.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting be
ur with the system package manager. It is recommended to use a virtual environment instead:
s://pip.pypa.io/warnings/venv
```

TASK 3:

Captured ICMP pings sent via loopback interface:

```
rtt min/avg/max/mdev = 0.033/0.044/0.050/0.007 ms
yogita@yogita-Inspiron-5502:~$ ping 127.0.0.1 -c 5
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.037 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.046 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.045 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.044 ms

--- 127.0.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4075ms
rtt min/avg/max/mdev = 0.037/0.043/0.046/0.003 ms
```

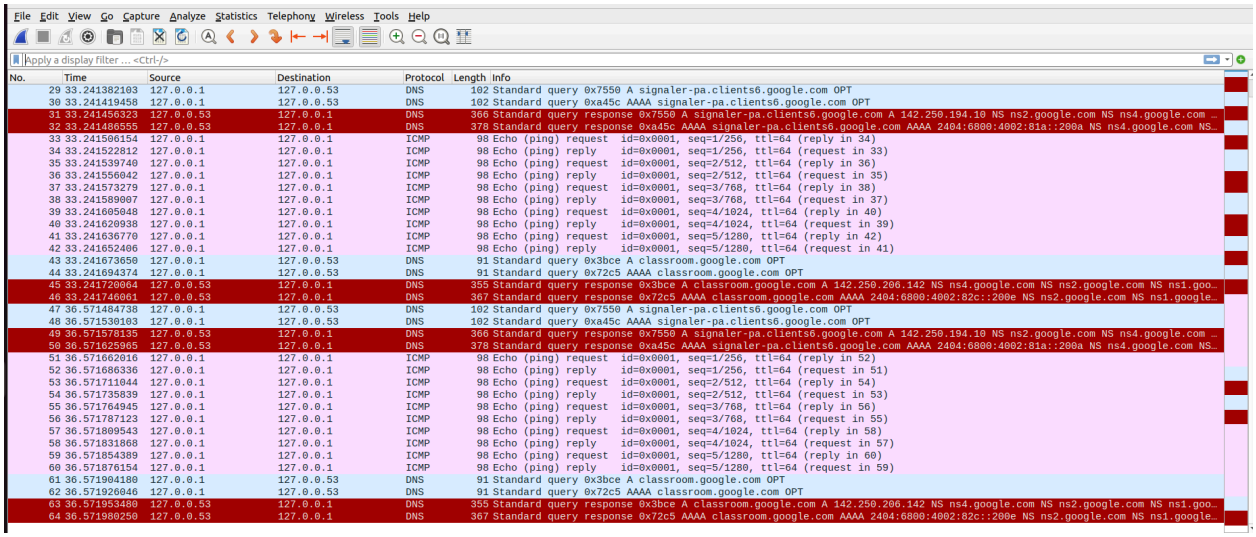


TASK 4:

Running Replay Attack:

```
.pypa.io/warnings/venv
yogita@yogita-Inspiron-5502:~/CyberSecurity/Lab7$ sudo python3 replayattack.py
.................
Sent 18 packets.
yogita@yogita-Inspiron-5502:~/CyberSecurity/Lab7$
```

After Replay:



No.	Time	Source	Destination	Protocol	Length	Info
29	33.241382103	127.0.0.1	127.0.0.53	DNS	102	Standard query 0x7550 A signaler-pa.clients6.google.com OPT
30	33.241419458	127.0.0.1	127.0.0.53	DNS	102	Standard query 0xa45c AAAA signaler-pa.clients6.google.com OPT
31	33.241456323	127.0.0.53	127.0.0.1	DNS	366	Standard query response 0x7550 A signaler-pa.clients6.google.com A 142.250.194.10 NS ns2.google.com NS ns4.google.com …
32	33.241486555	127.0.0.53	127.0.0.1	DNS	378	Standard query response 0xa45c AAAA signaler-pa.clients6.google.com AAAA 2404:6800:4002:81a::200a NS ns4.google.com NS…
33	33.241506154	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (reply in 34)
34	33.241522812	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) reply id=0x0001, seq=1/256, ttl=64 (request in 33)
35	33.241539740	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (reply in 36)
36	33.241556042	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) reply id=0x0001, seq=2/512, ttl=64 (request in 35)
37	33.241573279	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (reply in 38)
38	33.241589007	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) reply id=0x0001, seq=3/768, ttl=64 (request in 37)
39	33.241605048	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (reply in 40)
40	33.241620938	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) reply id=0x0001, seq=4/1024, ttl=64 (request in 39)
41	33.241636770	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (reply in 42)
42	33.241652406	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) reply id=0x0001, seq=5/1280, ttl=64 (request in 41)
43	33.241673650	127.0.0.1	127.0.0.53	DNS	91	Standard query 0x3bce A classroom.google.com OPT
44	33.241694374	127.0.0.1	127.0.0.53	DNS	91	Standard query 0x72c5 AAAA classroom.google.com OPT
45	33.241720064	127.0.0.53	127.0.0.1	DNS	355	Standard query response 0x3bce A classroom.google.com A 142.250.206.142 NS ns4.google.com NS ns2.google.com NS ns1.goo…
46	33.241746061	127.0.0.53	127.0.0.1	DNS	367	Standard query response 0x72c5 AAAA classroom.google.com AAAA 2404:6800:4002:82c::200e NS ns2.google.com NS ns1.google…
47	36.571484738	127.0.0.1	127.0.0.53	DNS	102	Standard query 0x7550 A signaler-pa.clients6.google.com OPT
48	36.571530103	127.0.0.1	127.0.0.53	DNS	102	Standard query 0xa45c AAAA signaler-pa.clients6.google.com OPT
49	36.571578135	127.0.0.53	127.0.0.1	DNS	366	Standard query response 0x7550 A signaler-pa.clients6.google.com A 142.250.194.10 NS ns2.google.com NS ns4.google.com …
50	36.571625965	127.0.0.53	127.0.0.1	DNS	378	Standard query response 0xa45c AAAA signaler-pa.clients6.google.com AAAA 2404:6800:4002:81a::200a NS ns4.google.com NS…
51	36.571662016	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (reply in 52)
52	36.571686336	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) reply id=0x0001, seq=1/256, ttl=64 (request in 51)
53	36.571711044	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (reply in 54)
54	36.571735839	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) reply id=0x0001, seq=2/512, ttl=64 (request in 53)
55	36.571764945	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (reply in 56)
56	36.571787123	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) reply id=0x0001, seq=3/768, ttl=64 (request in 55)
57	36.571809543	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (reply in 58)
58	36.571831868	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) reply id=0x0001, seq=4/1024, ttl=64 (request in 57)
59	36.571854389	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (reply in 60)
60	36.571876154	127.0.0.1	127.0.0.1	ICMP	98	Echo (ping) reply id=0x0001, seq=5/1280, ttl=64 (request in 59)
61	36.571904180	127.0.0.1	127.0.0.53	DNS	91	Standard query 0x3bce A classroom.google.com OPT
62	36.571926046	127.0.0.1	127.0.0.53	DNS	91	Standard query 0x72c5 AAAA classroom.google.com OPT
63	36.571953480	127.0.0.53	127.0.0.1	DNS	355	Standard query response 0x3bce A classroom.google.com A 142.250.206.142 NS ns4.google.com NS ns2.google.com NS ns1.goo…
64	36.571980250	127.0.0.53	127.0.0.1	DNS	367	Standard query response 0x72c5 AAAA classroom.google.com AAAA 2404:6800:4002:82c::200e NS ns2.google.com NS ns1.google…

TASK 5:

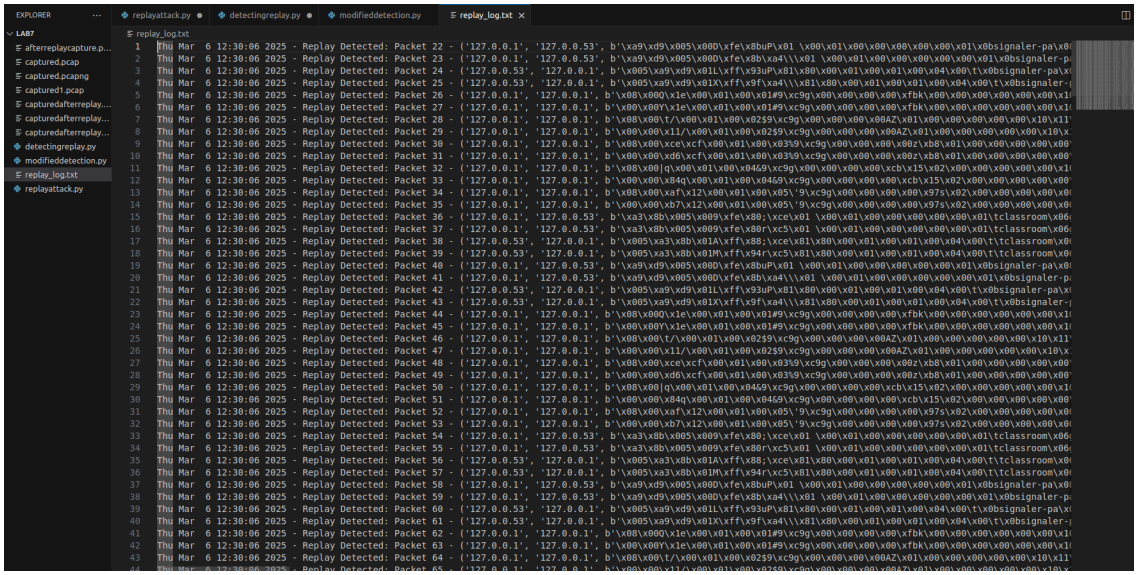
Running the detecting script

```
Sent 18 packets.
yogita@yogita-Inspiron-5502:~/CyberSecurity/Lab7$ sudo python3 detectingreplay.py
Analyzing afterreplaycapture.pcap for replay attacks...
⚠ Replay detected: Packet 46 (Count: 2)
⚠ Replay detected: Packet 47 (Count: 2)
⚠ Replay detected: Packet 48 (Count: 2)
⚠ Replay detected: Packet 49 (Count: 2)
⚠ Replay detected: Packet 50 (Count: 2)
⚠ Replay detected: Packet 51 (Count: 2)
⚠ Replay detected: Packet 52 (Count: 2)
⚠ Replay detected: Packet 53 (Count: 2)
⚠ Replay detected: Packet 54 (Count: 2)
⚠ Replay detected: Packet 55 (Count: 2)
⚠ Replay detected: Packet 56 (Count: 2)
⚠ Replay detected: Packet 57 (Count: 2)
⚠ Replay detected: Packet 58 (Count: 2)
⚠ Replay detected: Packet 59 (Count: 2)
⚠ Replay detected: Packet 60 (Count: 2)
⚠ Replay detected: Packet 61 (Count: 2)
⚠ Replay detected: Packet 62 (Count: 2)
⚠ Replay detected: Packet 63 (Count: 2)
yogita@yogita-Inspiron-5502:~/CyberSecurity/Lab7$
```

TASK 6:

```
⚠ Replay detected: Packet 63 (Count: 2)
yogita@yogita-Inspiron-5502:~/CyberSecurity/Lab7$ sudo python3 modifieddetection.py
Analyzing capturedafterreplay1.pcap for replay attacks...
⚠ Replay detected: Packet 22 (Time Diff: 1.47s)
⚠ Replay detected: Packet 23 (Time Diff: 1.47s)
⚠ Replay detected: Packet 24 (Time Diff: 1.47s)
⚠ Replay detected: Packet 25 (Time Diff: 1.47s)
⚠ Replay detected: Packet 26 (Time Diff: 1.47s)
⚠ Replay detected: Packet 27 (Time Diff: 1.47s)
⚠ Replay detected: Packet 28 (Time Diff: 1.47s)
⚠ Replay detected: Packet 29 (Time Diff: 1.47s)
⚠ Replay detected: Packet 30 (Time Diff: 1.47s)
⚠ Replay detected: Packet 31 (Time Diff: 1.47s)
⚠ Replay detected: Packet 32 (Time Diff: 1.47s)
⚠ Replay detected: Packet 33 (Time Diff: 1.47s)
⚠ Replay detected: Packet 34 (Time Diff: 1.47s)
⚠ Replay detected: Packet 35 (Time Diff: 1.47s)
⚠ Replay detected: Packet 36 (Time Diff: 1.47s)
⚠ Replay detected: Packet 37 (Time Diff: 1.47s)
⚠ Replay detected: Packet 38 (Time Diff: 1.47s)
```

LOG FILE:



The modified detection Code does the following:

Logs suspicious activity – Instead of just printing, it may store detected replay attacks in a log file.

Uses timestamps – To check if packets are replayed within a short time window.

Handles nonces (random numbers) – If packets contain authentication mechanisms, it checks for duplicate nonces.

Implements encryption checks – Ensures replayed packets are not just modified versions of old packets.

Observations

To analyze replay attacks, we captured a sequence of network packets, then replayed them using a script. The following observations were made:

- The replayed packets had identical source and destination addresses as the original packets.
- The timestamps of the replayed packets were different but closely matched the original packets.
- No built-in mechanisms in the captured protocol prevented replay attacks.
- Without a detection mechanism, replayed packets were treated as legitimate by the network.

The modified detection script identified suspicious packets based on timestamps and content similarity. Packets replayed within a short time window were flagged.

Findings

From our analysis, key findings include:

- **Replay attacks are feasible** in unprotected communication channels.
- **Timestamps alone** are not sufficient unless properly validated.
- **Packet uniqueness must be ensured** using cryptographic techniques.
- **Logging and monitoring** aid in detecting suspicious activities.

Suggested Preventive Measures

To mitigate replay attacks, the following techniques can be implemented:

1) Timestamps

- Every transmitted packet should include a timestamp.

- The receiver must validate timestamps and discard packets outside an acceptable time window.
- Time synchronization between sender and receiver should be maintained.

2) Nonces (Unique Identifiers)

- A unique, randomly generated nonce should be attached to each communication.
- The receiver must verify that each nonce is used only once.
- Previously seen nonces should be stored and checked against incoming packets.

3) Cryptographic Signatures

- Use cryptographic hashing (e.g., HMAC) to sign packets.
- The receiver verifies the signature to ensure message integrity.
- Any tampered or replayed packet fails validation.

4) Secure Communication Protocols

- Implement TLS or IPSec to encrypt network communication.
- Use mutual authentication to ensure both parties are verified.
- Adopt secure session management techniques to prevent session hijacking.

5) Logging and Real-Time Monitoring

- Maintain logs of all detected replay attempts.
- Implement an intrusion detection system (IDS) to identify replay attacks.
- Alert administrators in real time when suspicious activity is detected.