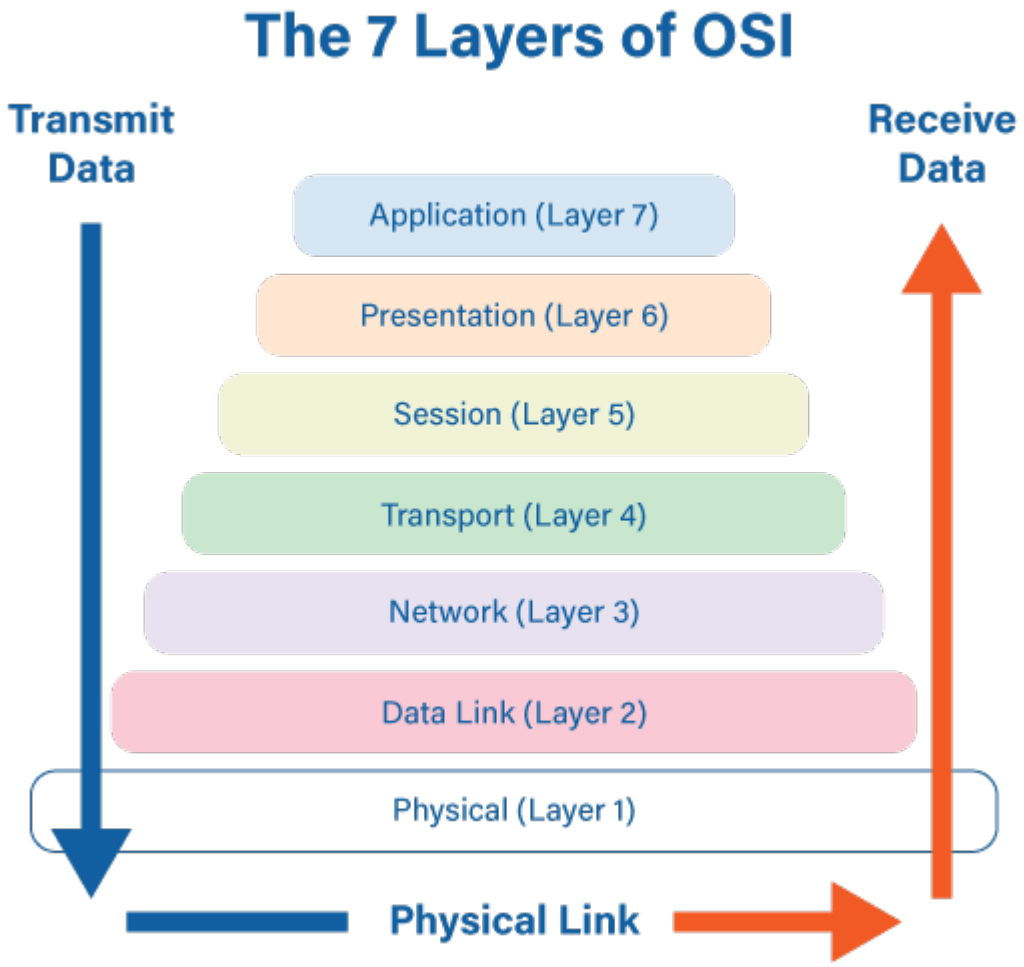
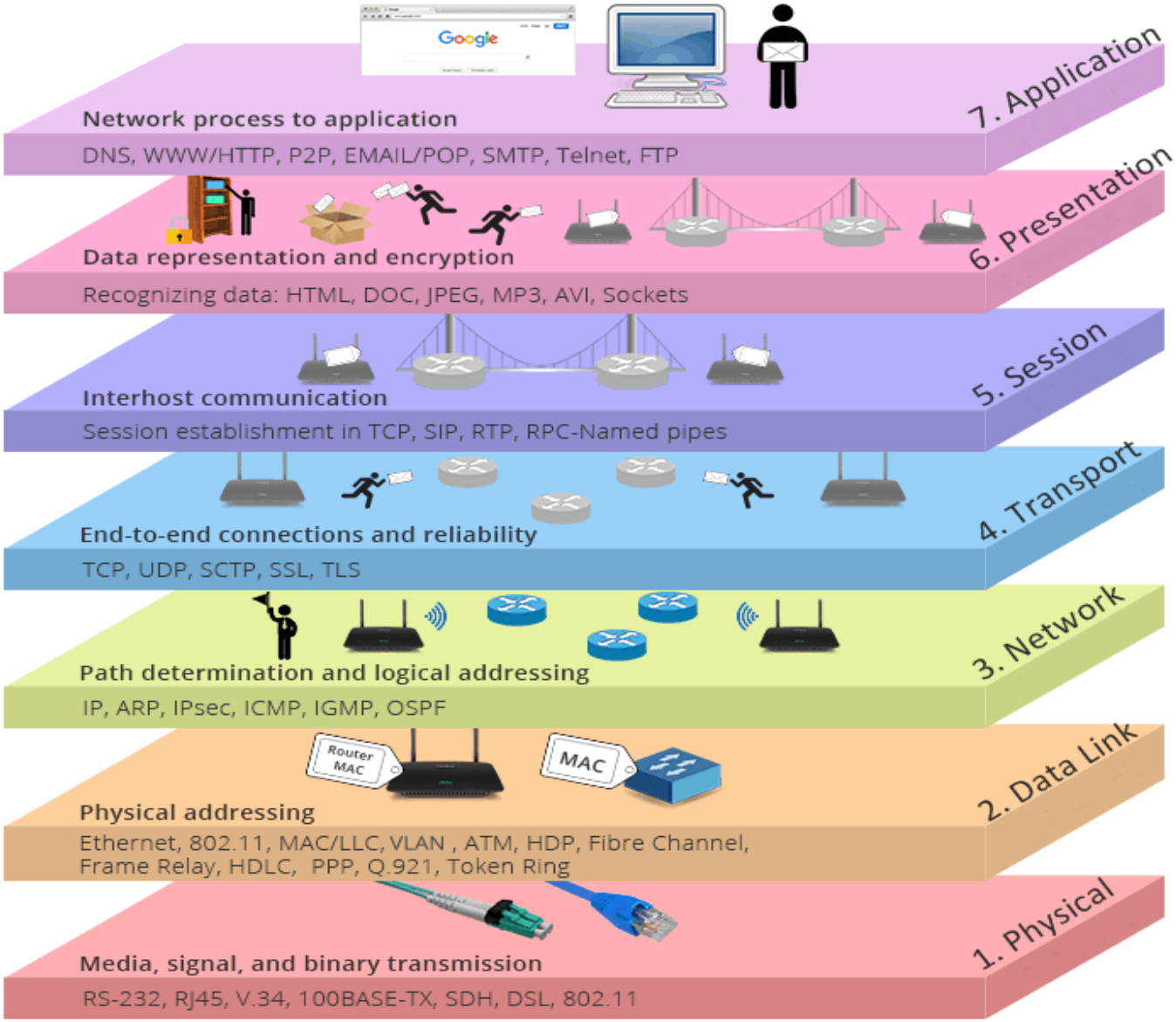
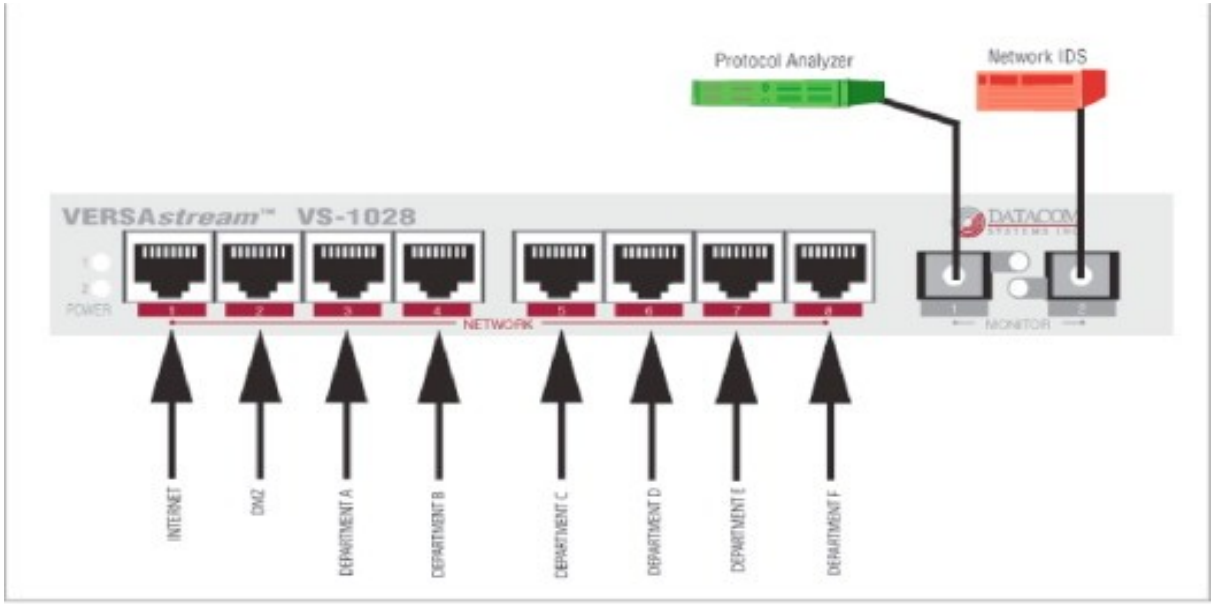

The 7 Layers of OSI
Transmit Data
Receive Data
Application (Layer 7)
Presentation (Layer 6)
Session (Layer 5)
Transport (Layer 4)
Network (Layer 3)
Data Link (Layer 2)
Physical (Layer 1)
Physical Link

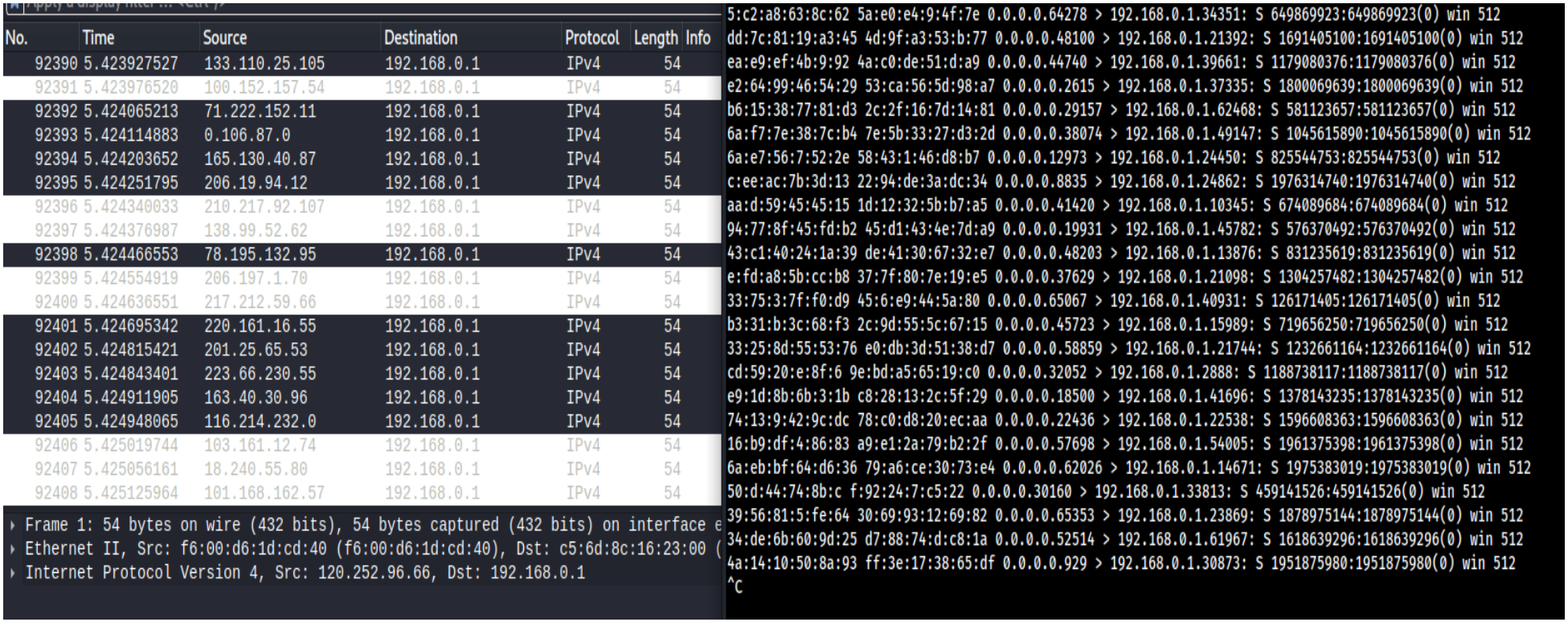

Google
7. Application
Network process to application
DNS, WWW/HTTP, P2P, EMAIL/POP, SMTP, Telnet, FTP
6. Presentation
Data representation and encryption
Recognizing data: HTML, DOC, JPEG, MP3, AVI, Sockets
5. Session
Interhost communication
Session establishment in TCP, SIP, RTP, RPC-Named pipes
4. Transport
End-to-end connections and reliability
TCP, UDP, SCTP, SSL, TLS
3. Network
Path determination and logical addressing
IP, ARP, IPsec, ICMP, IGMP, OSPF
2. Data Link
Router MAC
MAC
Physical addressing
Ethernet, 802.11, MAC/LLC, VLAN , ATM, HDP, Fibre Channel,
Frame Relay, HDLC, PPP, Q.921, Token Ring
1. Physical
Media, signal, and binary transmission
RS-232, RJ45, V.34, 100BASE-TX, SDH, DSL, 802.11

OSI (Open Source Interconnection) 7 Layer Model

Layer	Application/Example		Central Device/ Protocols			DOD4 Model
Application (7) Serves as the window for users and application processes to access the network services.	**End User layer** Program that opens what was sent or creates what is to be sent Resource sharing • Remote file access • Remote printer access • Directory services • Network management		**User Applications** SMTP		**GATEWAY** Can be used on all layers	Process
Presentation (6) Formats the data to be presented to the Application layer. It can be viewed as the "Translator" for the network.	**Syntax layer** encrypt & decrypt (if needed) Character code translation • Data conversion • Data compression • Data encryption • **Character Set Translation**		JPEG/ASCII EBDIC/TIFF/GIF PICT			
Session (5) Allows session establishment between processes running on different stations.	**Synch & send to ports** (logical ports) Session establishment, maintenance and termination • Session support - perform security, name recognition, logging, etc.		**Logical Ports** RPC/SQL/NFS NetBIOS names			
Transport (4) Ensures that messages are delivered error-free, in sequence, and with no losses or duplications.	**TCP** Host to Host, Flow Control Message segmentation • Message acknowledgement • Message traffic control • Session multiplexing	PACKET FILTERING	TCP/SPX/UDP			Host to Host
Network (3) Controls the operations of the subnet, deciding which physical path the data takes.	**Packets** ("letter", contains IP address) Routing • Subnet traffic control • Frame fragmentation • Logical-physical address mapping • Subnet usage accounting		**Routers** IP/IPX/ICMP			Internet
Data Link (2) Provides error-free transfer of data frames from one node to another over the Physical layer.	**Frames** ("envelopes", contains MAC address) [NIC card — Switch — NIC card] (end to end) Establishes & terminates the logical link between nodes • Frame traffic control • Frame sequencing • Frame acknowledgment • Frame delimiting • Frame error checking • Media access control		**Switch** **Bridge** **WAP** PPP/SLIP	Land Based Layers		Network
Physical (1) Concerned with the transmission and reception of the unstructured raw bit stream over the physical medium.	**Physical structure** Cables, hubs, etc. Data Encoding • Physical medium attachment • Transmission technique - Baseband or Broadband • Physical medium transmission Bits & Volts		**Hub**			

dsniff Toolkit (Layer-2 & Network Sniffing Toolkit)

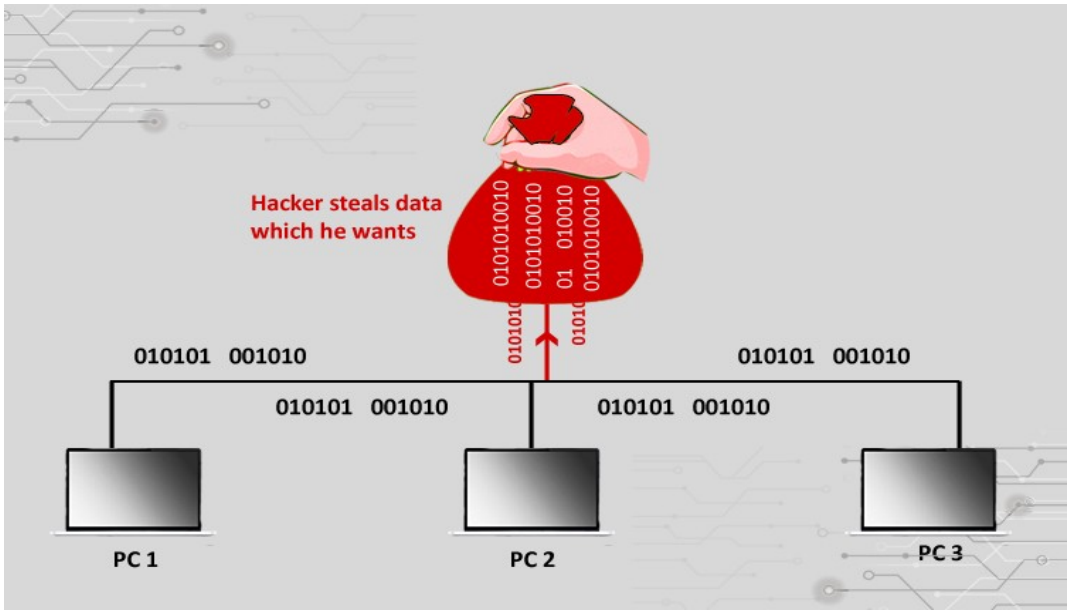




No.	Time	Source	Destination	Protocol	Length	Info
92390	5.423927527	133.110.25.105	192.168.0.1	IPv4	54	
92391	5.423976520	100.152.157.54	192.168.0.1	IPv4	54	
92392	5.424065213	71.222.152.11	192.168.0.1	IPv4	54	
92393	5.424114883	0.106.87.0	192.168.0.1	IPv4	54	
92394	5.424203652	165.130.40.87	192.168.0.1	IPv4	54	
92395	5.424251795	206.19.94.12	192.168.0.1	IPv4	54	
92396	5.424340033	210.217.92.107	192.168.0.1	IPv4	54	
92397	5.424376987	138.99.52.62	192.168.0.1	IPv4	54	
92398	5.424466553	78.195.132.95	192.168.0.1	IPv4	54	
92399	5.424554919	206.197.1.70	192.168.0.1	IPv4	54	
92400	5.424636551	217.212.59.66	192.168.0.1	IPv4	54	
92401	5.424695342	220.161.16.55	192.168.0.1	IPv4	54	
92402	5.424815421	201.25.65.53	192.168.0.1	IPv4	54	
92403	5.424843401	223.66.230.55	192.168.0.1	IPv4	54	
92404	5.424911905	163.40.30.96	192.168.0.1	IPv4	54	
92405	5.424948065	116.214.232.0	192.168.0.1	IPv4	54	
92406	5.425019744	103.161.12.74	192.168.0.1	IPv4	54	
92407	5.425056161	18.240.55.80	192.168.0.1	IPv4	54	
92408	5.425125964	101.168.162.57	192.168.0.1	IPv4	54	

```
▸ Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface e
▸ Ethernet II, Src: f6:00:d6:1d:cd:40 (f6:00:d6:1d:cd:40), Dst: c5:6d:8c:16:23:00 (
▸ Internet Protocol Version 4, Src: 120.252.96.66, Dst: 192.168.0.1
```

```
5:c2:a8:63:8c:62 5a:e0:e4:9:4f:7e 0.0.0.0.64278 > 192.168.0.1.34351: S 649869923:649869923(0) win 512
dd:7c:81:19:a3:45 4d:9f:a3:53:b:77 0.0.0.0.48100 > 192.168.0.1.21392: S 1691405100:1691405100(0) win 512
ea:e9:ef:4b:9:92 4a:c0:de:51:d:a9 0.0.0.0.44740 > 192.168.0.1.39661: S 1179080376:1179080376(0) win 512
e2:64:99:46:54:29 53:ca:56:5d:98:a7 0.0.0.0.2615 > 192.168.0.1.37335: S 1800069639:1800069639(0) win 512
b6:15:38:77:81:d3 2c:2f:16:7d:14:81 0.0.0.0.29157 > 192.168.0.1.62468: S 581123657:581123657(0) win 512
6a:f7:7e:38:7c:b4 7e:5b:33:27:d3:2d 0.0.0.0.38074 > 192.168.0.1.49147: S 1045615890:1045615890(0) win 512
6a:e7:56:7:52:2e 58:43:1:46:d8:b7 0.0.0.0.12973 > 192.168.0.1.24450: S 825544753:825544753(0) win 512
c:ee:ac:7b:3d:13 22:94:de:3a:dc:34 0.0.0.0.8835 > 192.168.0.1.24862: S 1976314740:1976314740(0) win 512
aa:d:59:45:45:15 1d:12:32:5b:b7:a5 0.0.0.0.41420 > 192.168.0.1.10345: S 674089684:674089684(0) win 512
94:77:8f:45:fd:b2 45:d1:43:4e:7d:a9 0.0.0.0.19931 > 192.168.0.1.45782: S 576370492:576370492(0) win 512
43:c1:40:24:1a:39 de:41:30:67:32:e7 0.0.0.0.48203 > 192.168.0.1.13876: S 831235619:831235619(0) win 512
e:fd:a8:5b:cc:b8 37:7f:80:7e:19:e5 0.0.0.0.37629 > 192.168.0.1.21098: S 1304257482:1304257482(0) win 512
33:75:3:7f:f0:d9 45:6:e9:44:5a:80 0.0.0.0.65067 > 192.168.0.1.40931: S 126171405:126171405(0) win 512
b3:31:b:3c:68:f3 2c:9d:55:5c:67:15 0.0.0.0.45723 > 192.168.0.1.15989: S 719656250:719656250(0) win 512
33:25:8d:55:53:76 e0:db:3d:51:38:d7 0.0.0.0.58859 > 192.168.0.1.21744: S 1232661164:1232661164(0) win 512
cd:59:20:e:8f:6 9e:bd:a5:65:19:c0 0.0.0.0.32052 > 192.168.0.1.2888: S 1188738117:1188738117(0) win 512
e9:1d:8b:6b:3:1b c8:28:13:2c:5f:29 0.0.0.0.18500 > 192.168.0.1.41696: S 1378143235:1378143235(0) win 512
74:13:9:42:9c:dc 78:c0:d8:20:ec:aa 0.0.0.0.22436 > 192.168.0.1.22538: S 1596608363:1596608363(0) win 512
16:b9:df:4:86:83 a9:e1:2a:79:b2:2f 0.0.0.0.57698 > 192.168.0.1.54005: S 1961375398:1961375398(0) win 512
6a:eb:bf:64:d6:36 79:a6:ce:30:73:e4 0.0.0.0.62026 > 192.168.0.1.14671: S 1975383019:1975383019(0) win 512
50:d:44:74:8b:c f:92:24:7:c5:22 0.0.0.0.30160 > 192.168.0.1.33813: S 459141526:459141526(0) win 512
39:56:81:5:fe:64 30:69:93:12:69:82 0.0.0.0.65353 > 192.168.0.1.23869: S 1878975144:1878975144(0) win 512
34:de:6b:60:9d:25 d7:88:74:d:c8:1a 0.0.0.0.52514 > 192.168.0.1.61967: S 1618639296:1618639296(0) win 512
4a:14:10:50:8a:93 ff:3e:17:38:65:df 0.0.0.0.929 > 192.168.0.1.30873: S 1951875980:1951875980(0) win 512
^C
```



4

The **dsniff toolkit** is a **collection of network auditing and penetration-testing tools** designed to **intercept, analyze, and manipulate network traffic**, primarily within **local area networks (LANs)**. It is widely referenced in **ethical hacking curricula** to demonstrate weaknesses in **unencrypted and trust-based network protocols**.

---

1. What is dsniff?

- A **suite of command-line tools** for **traffic sniffing, MITM, and Layer-2 attacks**
- Commonly used on **Linux/Kali Linux**
- Operates mainly at **OSI Layer 2 (Data Link)** and **Layer 3/4**
- Intended for **security auditing and education** (authorized environments only)

---

2. Why dsniff is Important (Student Perspective)

- Shows how **unencrypted protocols leak credentials**
- Demonstrates **CAM table attacks** and **ARP-based MITM**
- Explains why **switches ≠ secure by default**
- Reinforces the need for **encryption (TLS/HTTPS)** and **network hardening**

---

3. Key Components of the dsniff Toolkit

3.1 dsniff

- Core sniffer
- Extracts **plaintext credentials** from network traffic
- Targets protocols like:
  - FTP
  - Telnet
  - HTTP (non-TLS)
  - POP3 / IMAP / SMTP

📌 **Concept:** Sniffs traffic and prints usernames/passwords in real time.

3.2 macof

- Performs **MAC flooding (CAM table overflow)**
- Sends thousands of frames with **random source MACs**
- Forces a switch to **flood traffic** (behave like a hub)

📌 **Attack Type:** CAM Table Attack
📌 **Impact:** Enables packet sniffing on switched networks

3.3 arpspoof

- Conducts **ARP poisoning**
- Redirects traffic between victim and gateway through attacker
- Enables **Man-in-the-Middle (MITM)** attacks

📌 **Attack Type:** ARP Spoofing
📌 **Impact:** Traffic interception and manipulation

3.4 dnsspoof

- Fakes **DNS responses**
- Redirects victims to malicious or fake websites

📌 **Attack Type:** DNS Spoofing
📌 **Impact:** Phishing, credential theft

3.5 filesnarf

- Sniffs and captures files transferred over:
  - NFS
  - SMB
  - FTP

📌 **Attack Type:** Data leakage
📌 **Impact:** Unauthorized file access

3.6 mailsnarf

- Captures emails transmitted in plaintext
- Targets SMTP, POP3, IMAP (without TLS)

📌 **Impact:** Email confidentiality breach

3.7 msgsnarf

- Sniffs instant-messaging traffic
- Extracts chat messages (legacy protocols)

---

4. Attacks Demonstrated Using dsniff

Attack	Tool Used	OSI Layer
MAC Flooding	macof	Layer 2
ARP Spoofing	arpspoof	Layer 2
MITM	arpspoof + dsniff	Layer 2/3
Credential Sniffing	dsniff	Layer 4
DNS Spoofing	dnsspoof	Layer 7
File Sniffing	filesnarf	Layer 7

5. Security Weaknesses Exposed by dsniff

- Trust-based LAN communication
- Plaintext protocols
- Weak switch configurations
- Lack of authentication at Layer-2

📌 **Key Insight:**

dsniff does not “break” encryption—it **exploits the absence of encryption**.

---

6. Detection & Prevention

Detection

- IDS/IPS alerts
- ARP table inconsistencies
- Duplicate MAC addresses
- Abnormal traffic flooding

Prevention

- Use **HTTPS / TLS**
- Enable **Switch Port Security**
- Enable **Dynamic ARP Inspection (DAI)**
- Use **802.1X authentication**
- Avoid plaintext protocols

---

7. Ethical & Legal Note (Very Important)

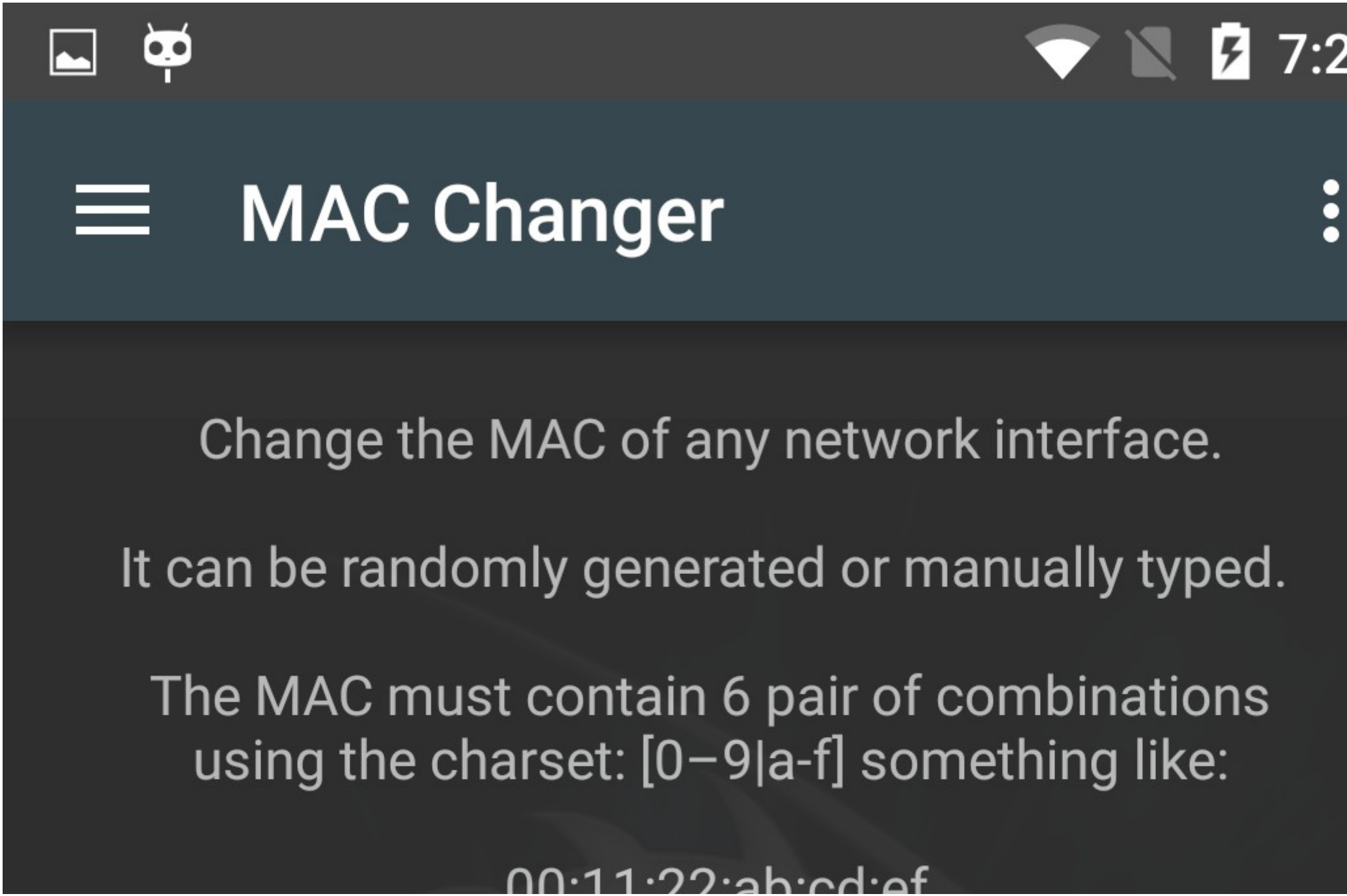
- dsniff must be used:
  - Only in **authorized lab environments**
  - With **written permission**
- Unauthorized usage is **illegal** and punishable

---

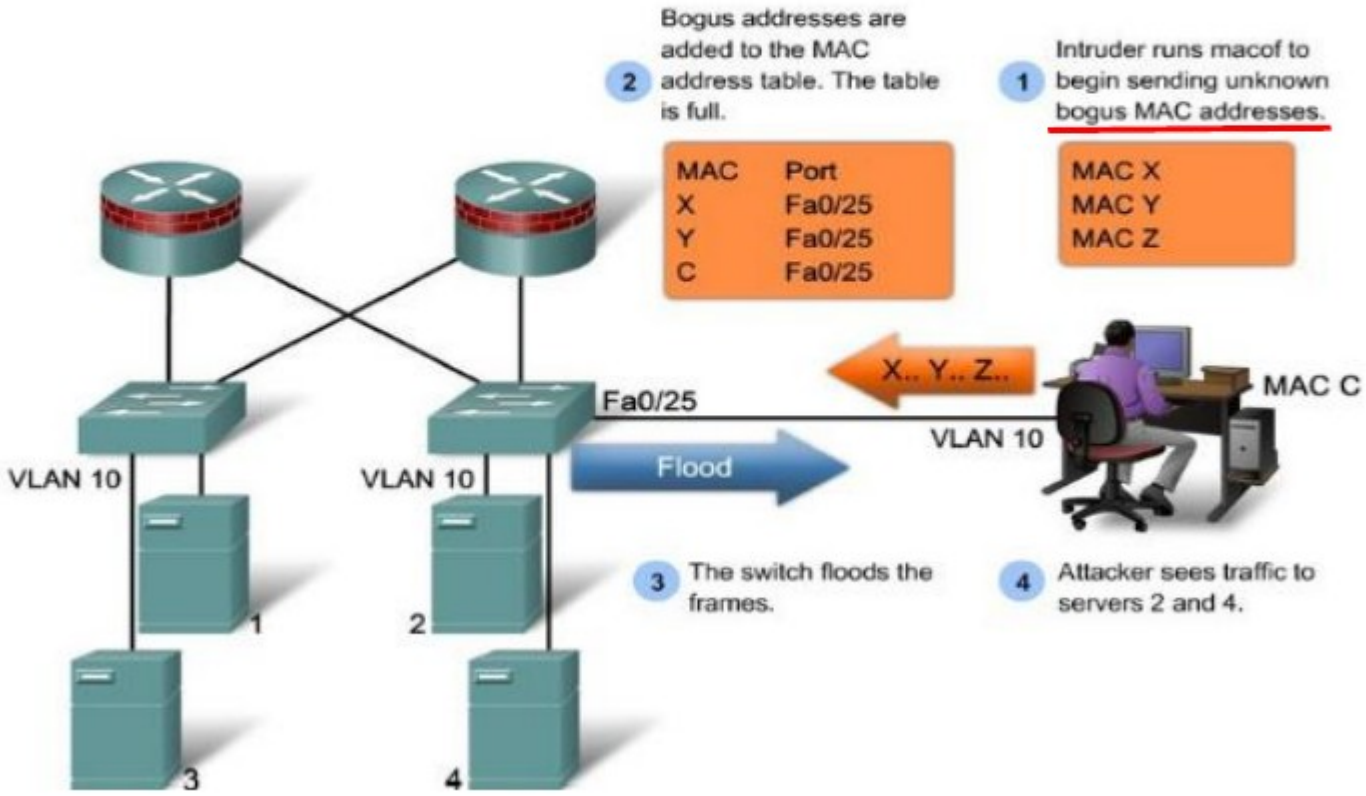
8. Summary - The **dsniff toolkit** is a collection of ***network auditing tools*** used to perform sniffing, MAC flooding, ARP spoofing, and man-in-the-middle attacks in local networks. Tools such as macof overflow a switch's CAM table, while arpspoof enables traffic interception by poisoning ARP caches. dsniff demonstrates vulnerabilities in plaintext protocols and trust-based LAN communication, highlighting the importance of encryption and switch security mechanisms.

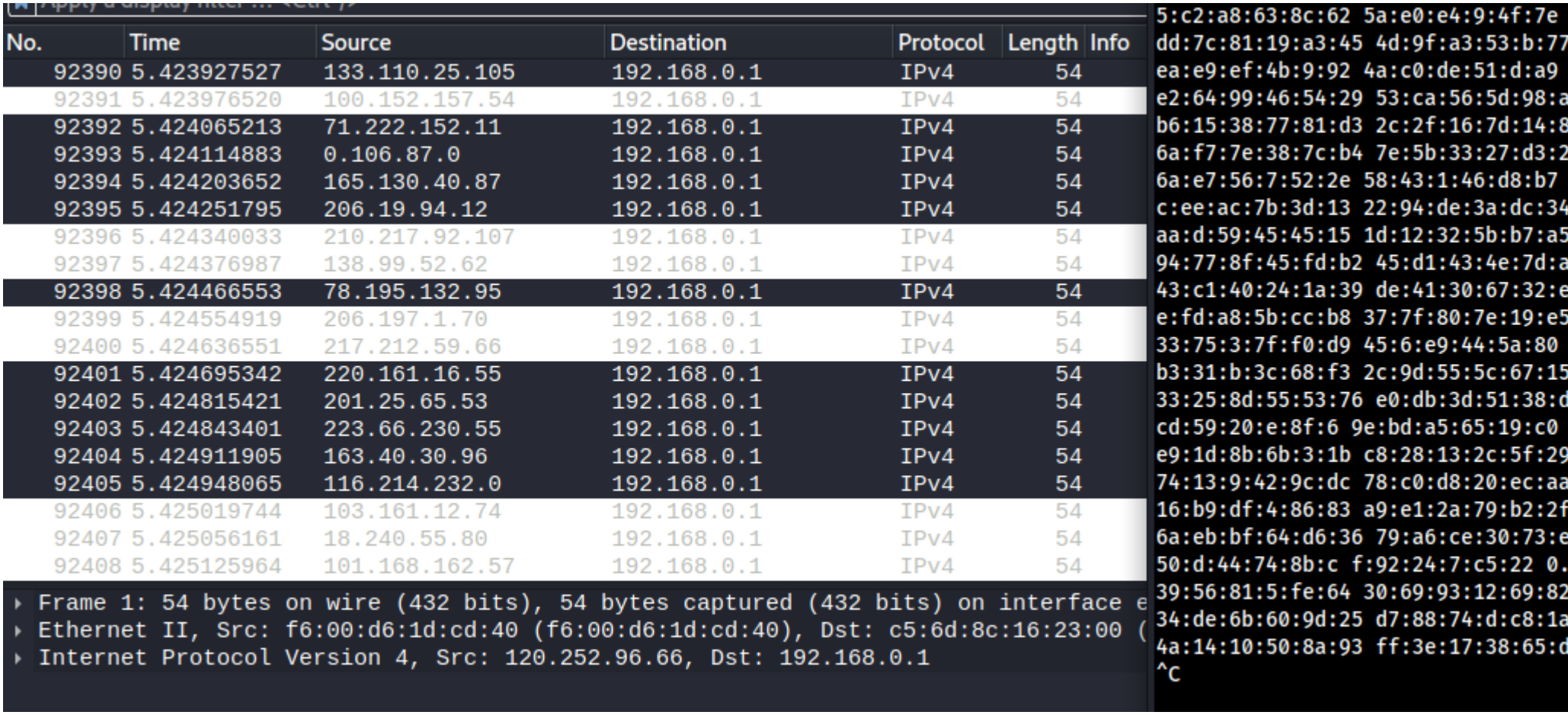
Below is a **clean, exam-safe, student-appropriate list of tools** used for **MAC Spoofing** and **MAC Flooding (CAM Table Attack)**.
✔️ **Names + purpose only** (no commands), suitable for **notes, exams, viva, and lab theory**.

Tools Used for MAC Spoofing & MAC Flooding Attacks


7:2
MAC Changer
Change the MAC of any network interface.
It can be randomly generated or manually typed.
The MAC must contain 6 pair of combinations
using the charset: [0–9|a-f] something like:
00:11:22:ab:cd:ef

MAC Address Table Overflow Attack





4

1 Tools for MAC Spoofing

1. macchanger

- **Most commonly used MAC spoofing tool**
- Allows changing:
  - Random MAC address

    - Specific MAC address
    - Reset to original MAC
- Works at **Layer 2**
- Widely used in **ethical hacking labs**

📌 **Purpose:**
Impersonation, bypassing MAC filtering, privacy testing

---

2. ifconfig / ip (Linux utilities)

- Built-in system tools
- Can manually modify MAC address
- Useful for **demonstration and learning**

📌 **Purpose:**
Manual MAC address modification

---

3. NetworkManager (nmcli)

- MAC spoofing at connection level
- Often used in modern Linux systems
- Supports persistent spoofing

📌 **Purpose:**
Stealth MAC identity change

---

2 Tools for MAC Flooding (CAM Table Attacks)

1. macof

(Part of dsniff toolkit)

- Classic **MAC flooding tool**
- Sends thousands of frames with fake MAC addresses
- Overflows switch CAM table
- Forces switch to behave like a hub

📌 **Purpose:**
CAM table overflow → packet sniffing

---

2. Yersinia

- **Advanced Layer-2 attack framework**
- Supports attacks on:
  - CAM tables
  - ARP
  - STP
  - DHCP
- GUI + CLI based

📌 **Purpose:**
Educational demonstration of Layer-2 protocol attacks

---

3. Scapy

- Python-based packet crafting tool
- Can generate custom Ethernet frames
- Used in **research and advanced labs**

📌 **Purpose:**
Custom MAC flooding and spoofing simulations

3 Tool Classification (Exam-Friendly Table)

Tool	Attack Type	Layer	Usage
macchanger	MAC Spoofing	Layer 2	Change MAC address
ifconfig / ip	MAC Spoofing	Layer 2	Manual MAC change
macof	MAC Flooding	Layer 2	CAM table overflow
Yersinia	MAC Flooding	Layer 2	Switch attack simulation
Scapy	Both	Layer 2	Packet crafting

4 Ethical Hacking Note (Important)

- These tools are used:
  - In **controlled lab environments**
  - With **written authorization**
- Unauthorized usage is **illegal and punishable**

5-Mark Exam Answer (Ready to Write)

MAC spoofing tools such as macchanger and ifconfig are used to modify a device's MAC address to impersonate another system. MAC flooding attacks are performed using tools like macof and Yersinia, which overflow the switch's CAM table with fake MAC addresses, forcing it to flood traffic. These attacks operate at Layer 2 and demonstrate weaknesses in MAC-based security.

s