

- Scenarios
- essays
- MITM
- ARP
- sniffing
- DoS/DDoS
- botnets

SNIFFERS, SESSION HIJACKING, AND DENIAL OF SERVICE ATTACKS

Prepared by:

Prof. Ahmed Redha Mahlous
PSU

Key Concepts

- Network sniffing and traffic analysis
- Session hijacking
- Denial of service (DoS)
- Distributed denial of service (DDoS) attacks
- Botnets

Sniffers

Sniffers

- An application or device designed to capture, or “sniff,” network traffic as it moves across the network
- A technology used to steal or observe information
- Allows viewing of email passwords, web passwords, File Transfer Protocol (FTP) credentials, email contents, and transferred files
- Network sniffing, aka eavesdropping, is a type of attack where an attacker captures the packets across a wire or across air (wireless connection).
- The main goal is to capture unencrypted credentials across the network.

Sniffing Threats

tool captures packets
monitor network traffic
ex: Wireshark, Tcpdump

- By placing a **packet sniffer** on a network in **promiscuous mode**, an attacker can capture and analyze all of the network traffic within a the same subnet

meaning: Physical Ports in company may be accessible

- Many enterprises' **switch ports** are open
- Anyone in the same physical location can plug into the network using an **Ethernet cable**

→ Attacker inside building
Can Plug into network

Normally,
• NIC (network card) only accepts packets sent to its own MAC address.
BUT in promiscuous mode:
• The NIC accepts ALL packets it sees.

So the attacker can:

- monitor other users' traffic
- steal sensitive data

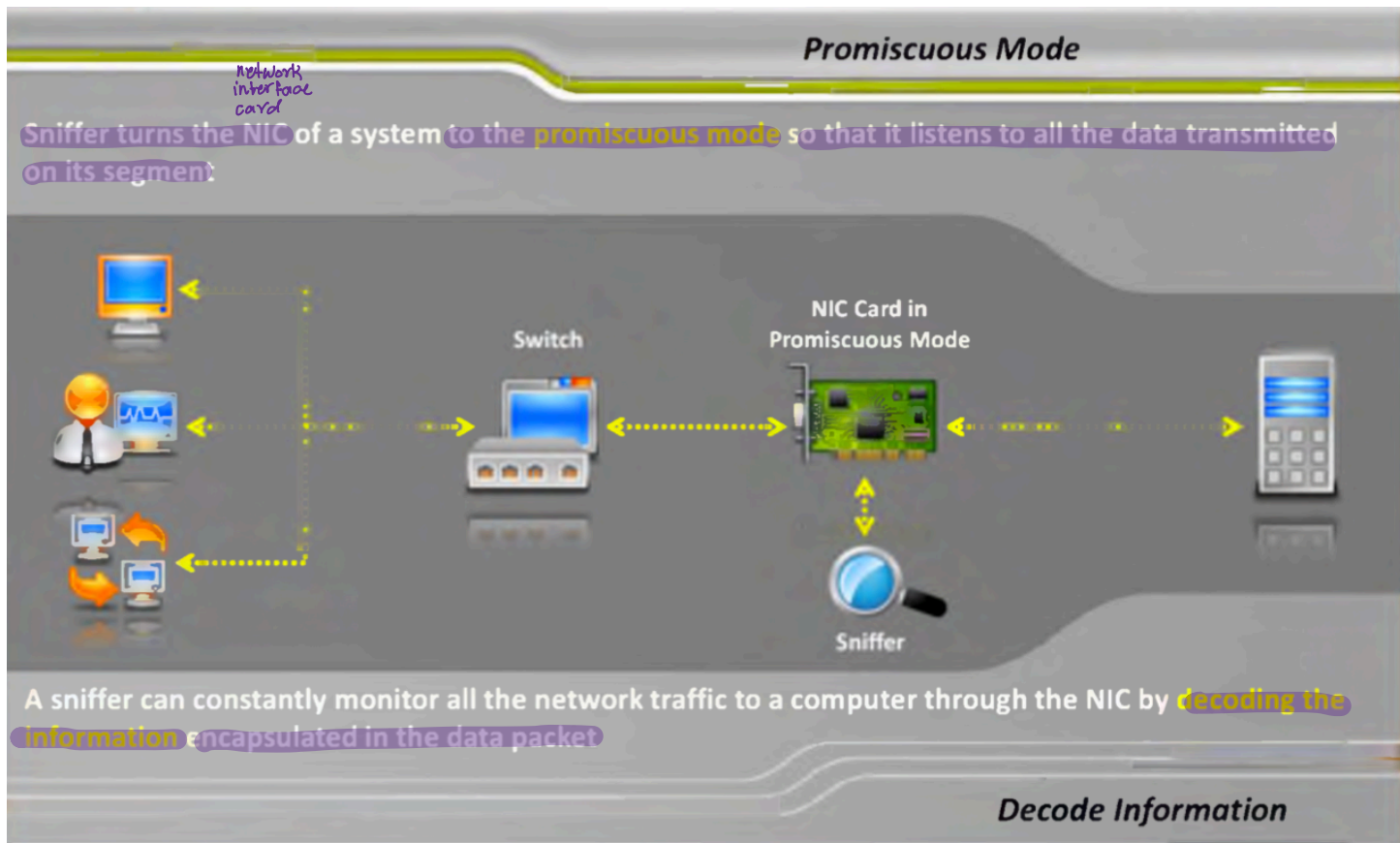
Important exam idea

Normal Mode
device reads:
• ONLY its own traffic

Promiscuous Mode
device reads:
• ALL traffic on network



How a Sniffer Works

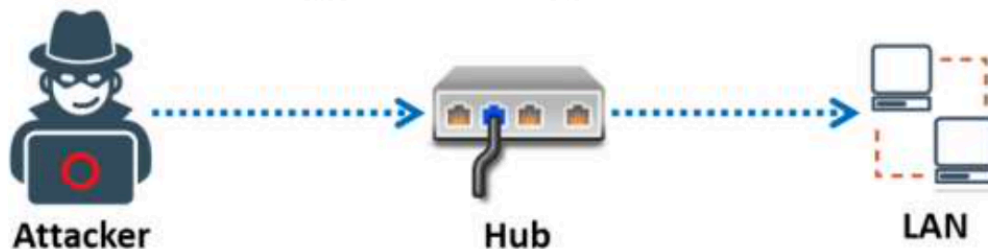


Types of Sniffing

1- Passive Sniffing:

- Passive sniffing means sniffing through a hub, on a hub the traffic is sent to all ports.
- It involves only monitoring of the packets sent by others without sending any additional data packets in the network traffic.
- In a network that use hubs to connect systems, all hosts on the network can see all traffic therefore attacker can easily capture traffic going through the hub.
- Hub usage is outdated today. Most modern networks use switches.

- **Passive sniffing involves sending no packets.** It simply captures and monitors the packets flowing in the network. A packet sniffer alone is not preferred for an attack because it works only in a **common collision domain**.
- A common collision domain is the sector of the network that is not switched or bridged (i.e., connected through a hub).
- Common collision domains are present in hub environments. A network that uses hubs to connect systems uses passive sniffing. In such networks, all hosts in the network can see all the traffic. Hence, it is easy to capture traffic through the hub using passive sniffing.



Attackers use the following passive sniffing methods to gain control over a target network:

- **Compromising physical security:** An attacker who succeeds in compromising the physical security of a target organization can walk into the organization with a laptop and try to plug into the network and capture sensitive information about the organization.
- **Using a Trojan horse:** Most Trojans have in-built sniffing capability. An attacker can install these on a victim's machine to compromise it. After compromising the victim's machine, the attacker can install a packet sniffer and perform sniffing.

Why Passive Sniffing less common these days?

Most modern networks use switches instead of hubs. A switch eliminates the risk of passive sniffing. However, a switch is still vulnerable to active sniffing.

Note: Passive sniffing provides significant stealth advantages over active sniffing.

Types of Sniffing (cont.)

2- Active Sniffing:

- Active sniffing is used in switched networks where traffic is not normally visible to all hosts. It involves actively interacting with the network by sending packets to manipulate traffic flow and capture data.

- Common techniques include:

1. ARP spoofing/poisoning to ^{goal} redirect traffic through the attacker, → Fake ARP replies
2. MAC flooding to ^{goal} overflow the switch's CAM table and force it into hub-like behavior → Flood switch with Fake MAC @

- The CAM (Content Addressable Memory) table in a switch maps MAC addresses to specific ports and manipulating it allows an attacker to intercept network traffic.

Types of Sniffing (cont.)

- ❑ Active sniffing is used to capture network traffic in switched LAN environments by actively injecting packets into the network. Unlike hub-based networks, where data is broadcast to all devices, switches forward packets only to the intended destination based on MAC addresses. As a result, passive sniffing is ineffective in switched networks.
- ❑ To overcome this limitation, attackers use active techniques such as ARP spoofing or MAC flooding to manipulate traffic and redirect it through their system. Switches rely on a Content Addressable Memory (CAM) table, which maps MAC addresses to specific ports. By exploiting this mechanism, an attacker can intercept traffic that would not normally be accessible.
- ❑ A sniffer can then capture and analyze packets, potentially exposing sensitive information such as credentials or confidential data

Detectability: Leaving Fingerprints

Passive Sniffing:

Passive sniffing is very difficult to detect because it does not generate additional network traffic. Detection is possible only through advanced techniques such as:

- **Monitoring NIC mode:** Identifying network interfaces running in promiscuous mode using specialized tools (rarely performed in practice)
- **Analyzing host performance:** Unusual CPU or disk usage may indicate packet capture activity, though this method is indirect and unreliable

Active Sniffing:

Active sniffing is easier to detect than passive sniffing because it introduces abnormal traffic into the network. Security systems such as Intrusion Detection Systems (IDS) can identify signs of attack, including:

- **Unusual ARP traffic:** Large numbers of ARP replies without corresponding requests or inconsistent IP-to-MAC mappings
- **MAC address flapping:** A MAC address appearing on multiple switch ports in a short time
- **Duplicate IP addresses:** Network conflicts caused by spoofed ARP responses

Protocols Vulnerable to Sniffing

**Telnet
and
Rlogin**

- ☐ Keystrokes including usernames and passwords are sent in clear text

IMAP

- ☐ Passwords and data are sent in clear text

HTTP

- ☐ Data is sent in clear text

**SMTP
and
NNTP**

- ☐ Passwords and data are sent in clear text

POP

- ☐ Passwords and data are sent in clear text

FTP

- ☐ Passwords and data are sent in clear text

Switch Basics & CAM Table

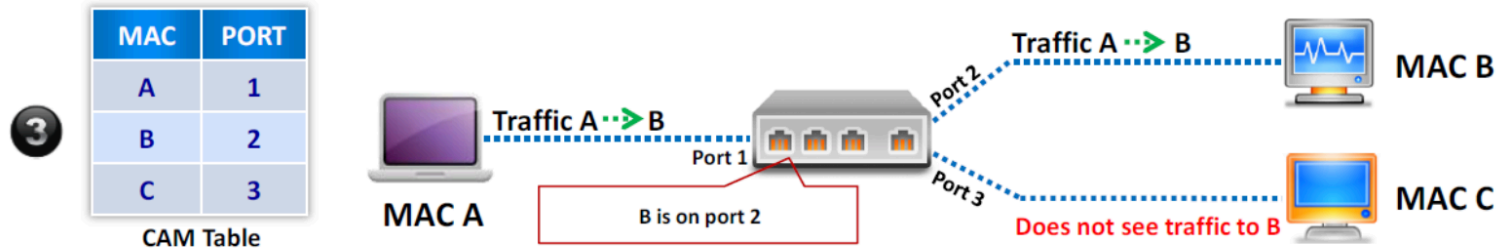
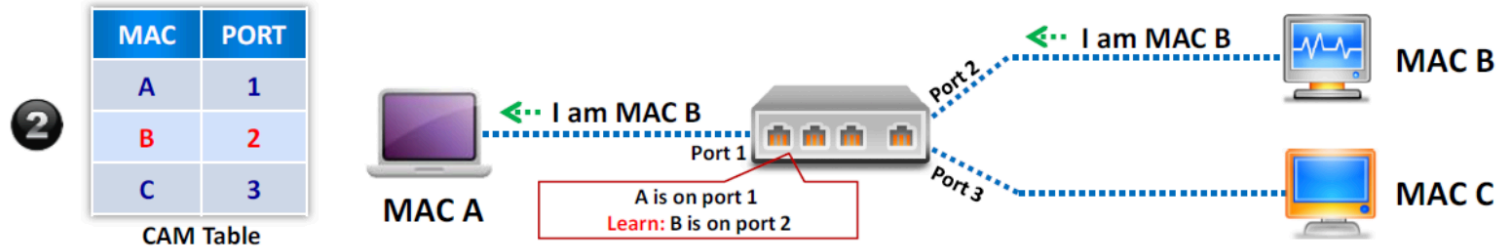
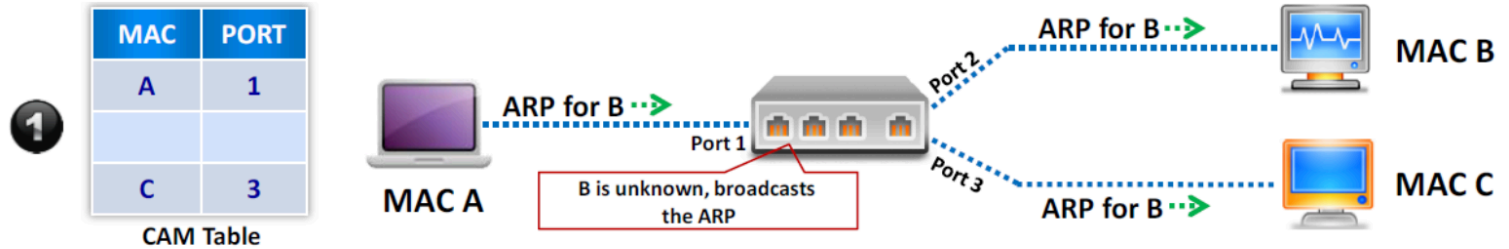
- Each switch maintains a Content Addressable Memory (CAM) table with a fixed size, where entries are dynamically updated.
- The CAM table stores mappings between MAC addresses and switch ports, along with associated VLAN information.
- This table enables the switch to efficiently forward frames to the correct destination. When a frame arrives, the switch learns the source MAC address and updates the CAM table accordingly.
- If the destination MAC address is not found in the table, the switch floods the frame to all ports within the same VLAN.

CAM Table

vlan	MAC Add	Type	Learn	Age	Ports
255	00d3.ad34.123g	Dyna mic	Yes	0	Gi5/2
5	as23.df45.45t6	Dyna mic	Yes	0	Gi2/5
5	er23.23er.t5e3	Dyna mic	Yes	0	Gi1/6



How CAM Works



ARP Protocol Basics

Purpose:

- Resolves IP addresses to MAC addresses
- Enables communication within a local network (LAN)
- Bridges the Network Layer (IP) and Data Link Layer (MAC)

How It Works:

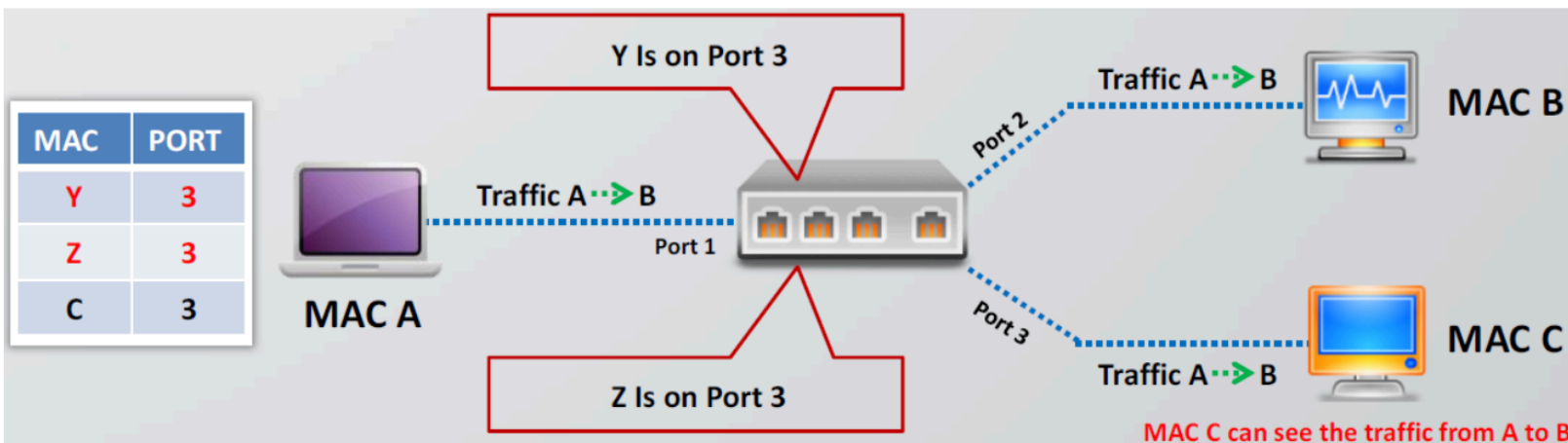
1. The host checks its ARP cache for an existing mapping
2. If not found, it broadcasts an ARP Request: → “Who has IP X.X.X.X?”
3. The target host replies with an ARP Reply (unicast): → “IP X.X.X.X is at MAC XX:XX:XX:XX:XX:XX”
4. The mapping is stored in the ARP cache (duration varies by OS)

Key Characteristics:

- **Stateless protocol:** no session or verification mechanism
- **No authentication:** replies are automatically trusted → vulnerable to spoofing
- **Local scope only:** operates within a LAN (does not cross routers)
- **Broadcast/Unicast behavior:**
 - Requests → broadcast
 - Replies → unicast
- Hosts maintain an **ARP cache** for recently resolved addresses

MAC Flooding attack

- MAC flooding is a Layer 2 attack that targets a network switch by overwhelming its Content Addressable Memory (CAM) table.
- In this attack, the attacker sends a large number of frames with random, spoofed source MAC addresses. This causes the CAM table to fill up, forcing legitimate entries to be removed.
- When the CAM table becomes full, the switch can no longer learn new MAC addresses. As a result, it begins to flood unknown unicast frames out of all ports, behaving similarly to a hub.
- This allows an attacker to capture network traffic from other devices and perform passive sniffing.



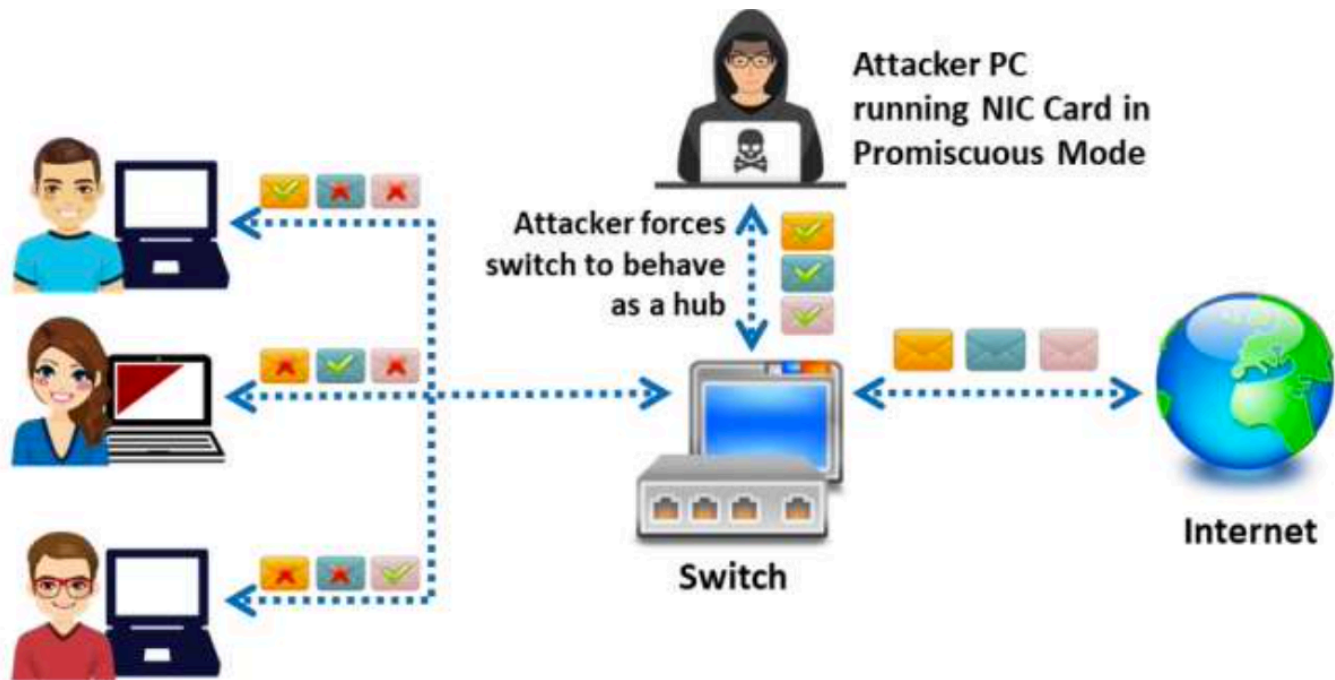


Figure 8.2: Working of a sniffer

How to Defend against MAC Attacks

Port Security Definition:

- Port security is a switch feature that restricts access to a port by limiting the MAC addresses allowed to connect.
- When enabled, the switch forwards traffic only from authorized MAC addresses.

Security Violation Conditions occur:

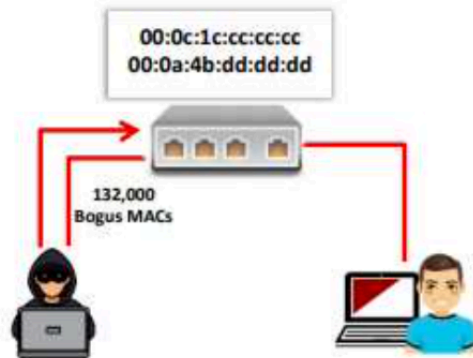
- ❑ When a port is configured as secure and the maximum number of secure MAC addresses is reached.
- ❑ When a machine's MAC address attempting to access the port does not match any identified secure MAC addresses.

Methods for Configuring Secure MAC Addresses:

- ✓ Manually configure all secure MAC addresses using the ***switch port port-security mac-address*** command.
- ✓ Allow the port to dynamically learn and configure secure MAC addresses from connected devices.
- ✓ Configure a specific number of addresses manually and allow the remainder to be dynamically learned.

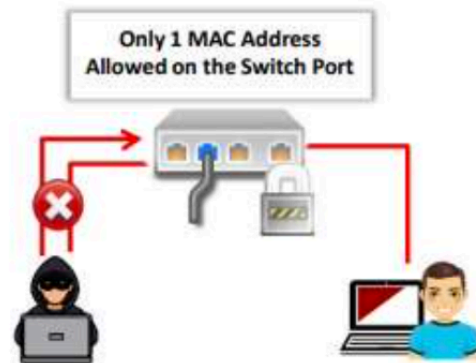
Benefits of Port Security:

- Limits MAC flooding attacks.
- Locks down ports to authorized devices only.
- Sends SNMP traps for security monitoring and alerts.



Configuring Port Security on Cisco Switch:

- switchport port-security
- switchport port-security maximum 1 vlan access
- switchport port-security violation restrict
- switchport port-security aging time 2
- switchport port-security aging type inactivity
- snmp-server enable traps port-security trap-rate 5



Port security can be used to **restrict inbound traffic** from only a selected set of MAC addresses and limit MAC flooding attack

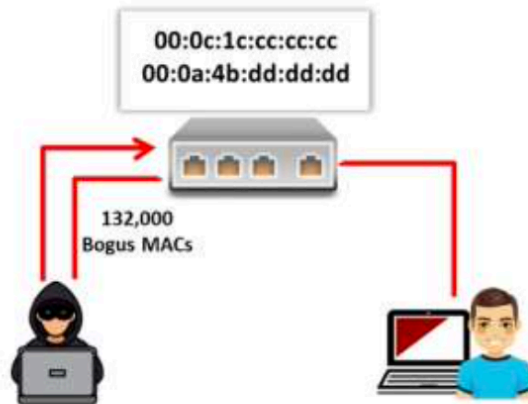


Figure 8.23: Flooding CAM tables

As shown in the figure, the number of MAC addresses allowed on the switch port is limited to one; therefore, the MAC requests are recognized as flooding. Port security locks down the port and sends an SNMP trap.

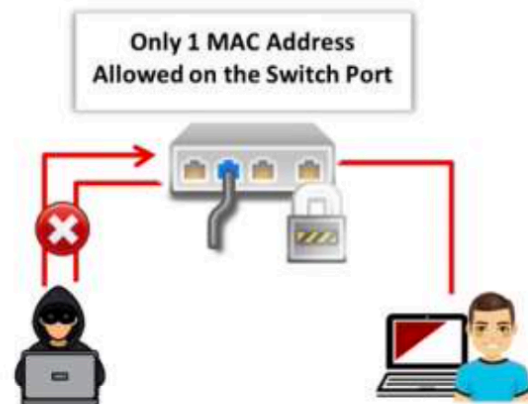


Figure 8.24: Blocking MAC flooding

ARP Poisoning / ARP Spoofing

- **Definition** :ARP poisoning is a cyberattack in which an attacker sends forged ARP messages onto a local network to manipulate ARP tables (ARP cache) of other devices.
- **Primary Primary Goal**: To associate the attacker's MAC address with the IP address of a legitimate device (e.g., the default gateway or another host)
- **Result**: Once successful, network traffic is redirected through the attacker's machine, enabling them to:
 - Intercept and steal data (Man-in-the-Middle attack)
 - Launch Denial-of-Service (DoS) attacks
 - Perform session hijacking

How does ARP Poisoning Works

■ Step 1: Reconnaissance

- The attacker identifies the IP addresses of:
 - The target victim
 - The default gateway

■ Step 2: The Poisoning

- The attacker sends forged ARP replies:

To the victim:

→ “I am the gateway. My MAC address is [Attacker’s MAC].”

To the gateway:

→ “I am the victim. My MAC address is [Attacker’s MAC].”

- **These fake replies are sent continuously to maintain the attack.**

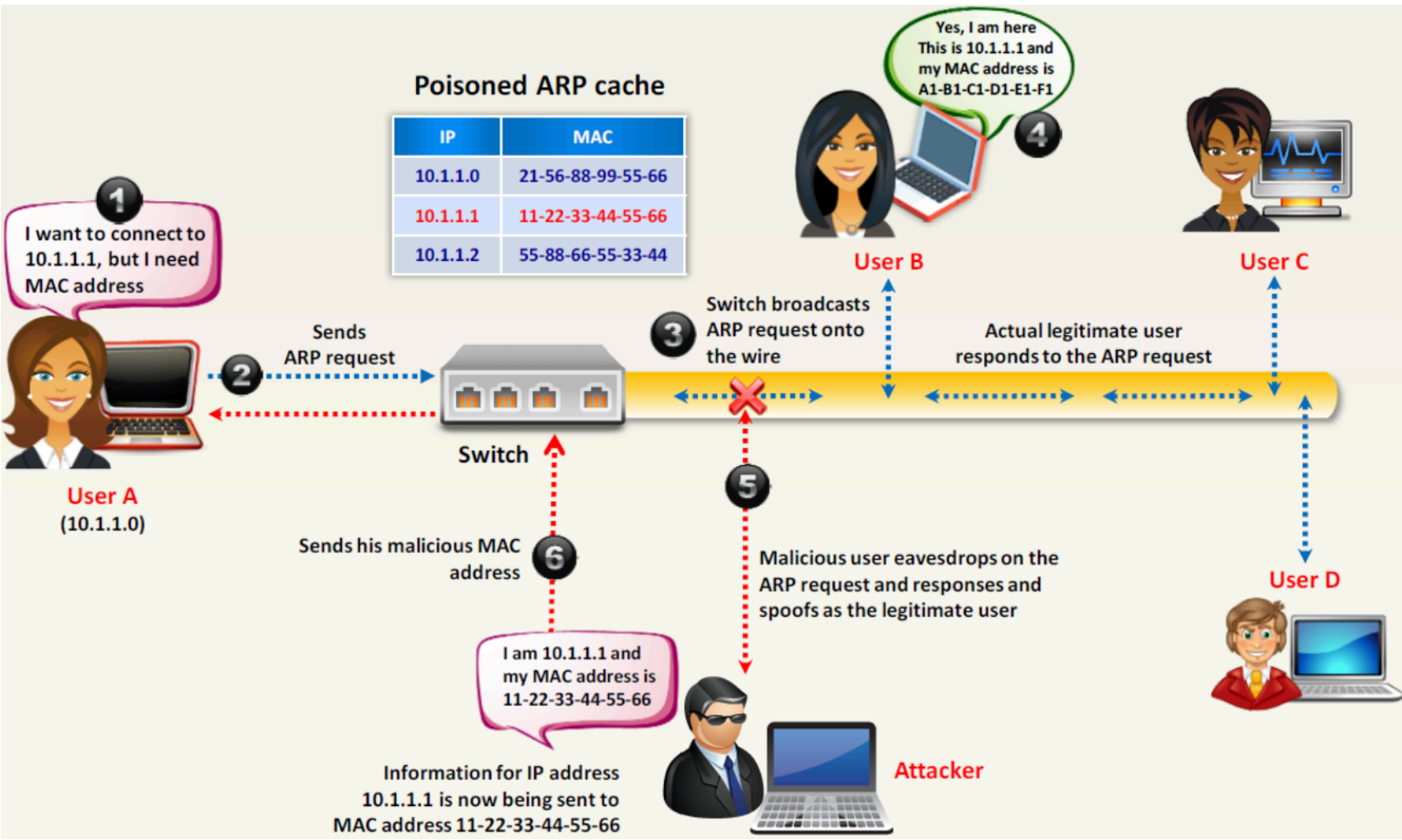
■ Step 3: The Redirection

- Both the victim and the gateway update their ARP caches with false mappings
- Traffic between them is now redirected through the attacker’s machine

■ Step 4: The Attack

- Sniff (capture sensitive data)
- Modify (alter packets in transit)
- Block (drop traffic → DoS attack)

How Does ARP Spoofing Work



Why is ARP Poisoning Possible?

The Core Vulnerability: **ARP is a Trust-Based Protocol**

- **No Authentication:** Devices automatically trust ARP replies without verifying their legitimacy.
- **Stateless Behavior:** Devices accept ARP replies even if they did not send a request (unsolicited responses)
- **No Integrity Verification:** There is no mechanism to validate that the IP–MAC mapping is correct
- **Result:** This lack of security allows attackers to inject forged ARP replies and poison ARP caches, enabling traffic interception and Man-in-the-Middle attacks.

ARP Spoofing Prevention

1. Encryption (Mitigation, not prevention)

- Use a **VPN (Virtual Private Network)**
- Encrypts traffic, making intercepted data **unreadable** to attackers
- Does **not stop ARP spoofing**, but reduces its impact

2. Static ARP Entries

- Manually bind IP addresses to MAC addresses
- Prevents devices from accepting fake ARP replies
- **Limitation:** Not scalable for large networks

3. Packet Filtering & Monitoring

- Use **IDS/IPS systems** to detect abnormal ARP activity

Identify:

- Conflicting IP–MAC mappings
 - Suspicious ARP traffic patterns
- Block malicious packets before they reach hosts

ARP Spoofing Prevention (Cont..)

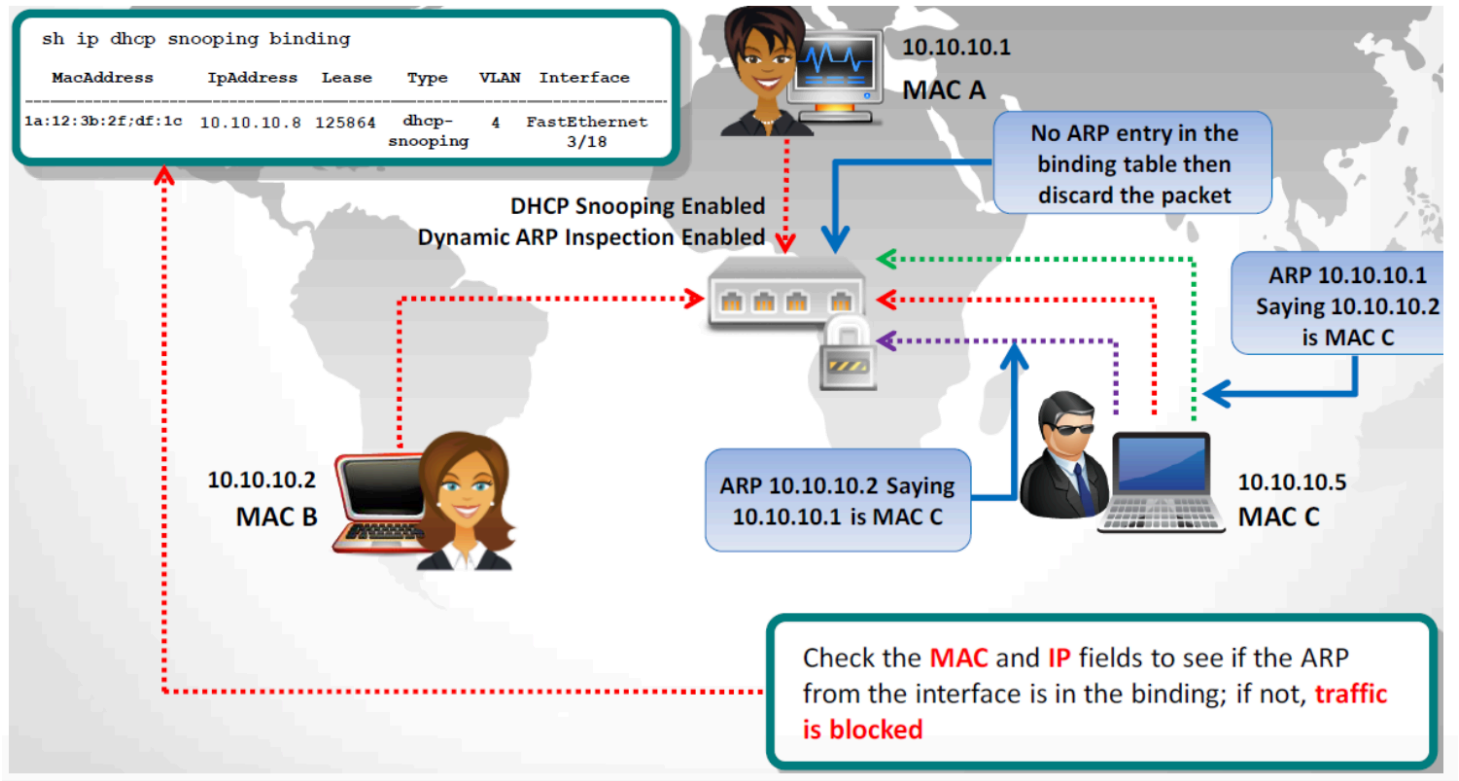
4. Dynamic ARP Inspection (DAI)

- Validates ARP packets using the **DHCP Snooping binding table**
- Drops invalid or spoofed ARP packets
- One of the **most effective network-level defenses**

5. Security Testing (Penetration Testing)

- Perform **controlled ARP spoofing tests** to evaluate defenses
- Identify vulnerabilities and improve security measures

ARP Spoofing Prevention (cont..)



Detection Tip:

Monitor networks for a single MAC address associated with multiple IP addresses or use tools like Wireshark to spot unusual ARP traffic.

Here is a simple way to detect that a specific device's ARP cache has been poisoned, using the command line. Start an operating system shell as an administrator. Use the following command to display the ARP table, on both Windows and Linux:

```
arp -a
```

The output will look something like this:

Internet Address	Physical Address
192.168.5.1	00-14-22-01-23-45
192.168.5.201	40-d4-48-cr-55-b8
192.168.5.202	00-14-22-01-23-45

If the table contains two different IP addresses that have the same MAC address, this indicates an ARP attack is taking place. Because the IP address 192.168.5.1 can be recognized as the router, the attacker's IP is probably 192.168.5.202.

Sniffing Tools

Wireshark

Tcpdump

Windump

Omnipeek

Dsniff

EtherApe

MSN Sniffer

NetWitness
NextGen

Throwing
Star LAN
Tap

Sniffing Countermeasures



Session Hijacking

What is Session Hijacking?

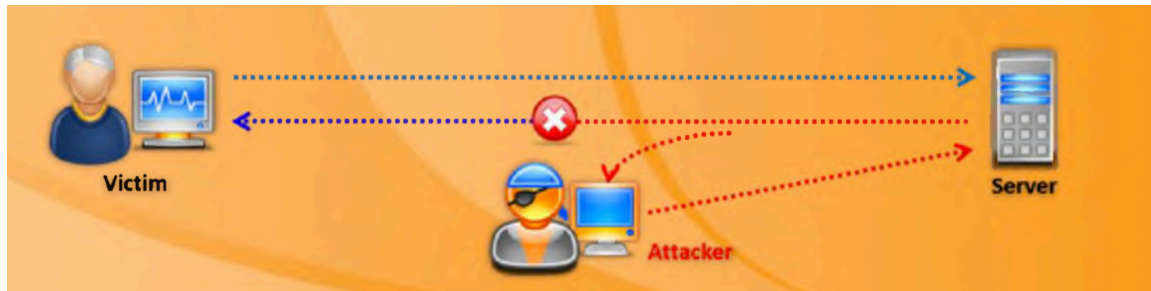
Session hijacking refers to an attack where an attacker takes over an active session valid TCP communication session between a client and a server.

How It Works:

- Authentication typically occurs only at the start of a session
- The server then relies on a session identifier (session ID / cookie) to maintain the connection
- The attacker captures this session ID (e.g., through sniffing or MITM attacks)
- The attacker reuses the session ID to impersonate the legitimate user

Impact:

- Unauthorized access to systems or accounts
- Identity theft
- Data theft or manipulation
- Fraudulent actions



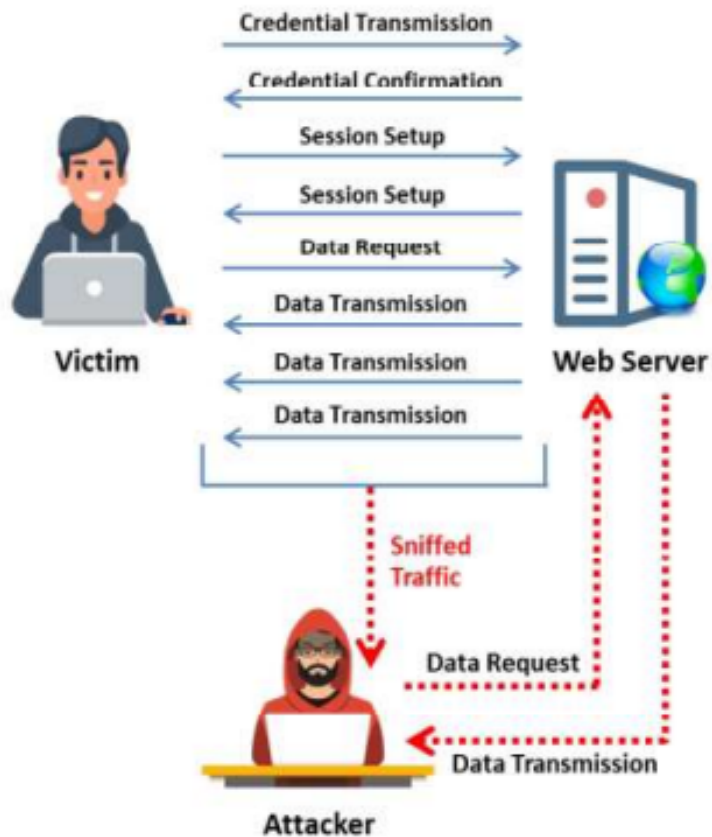
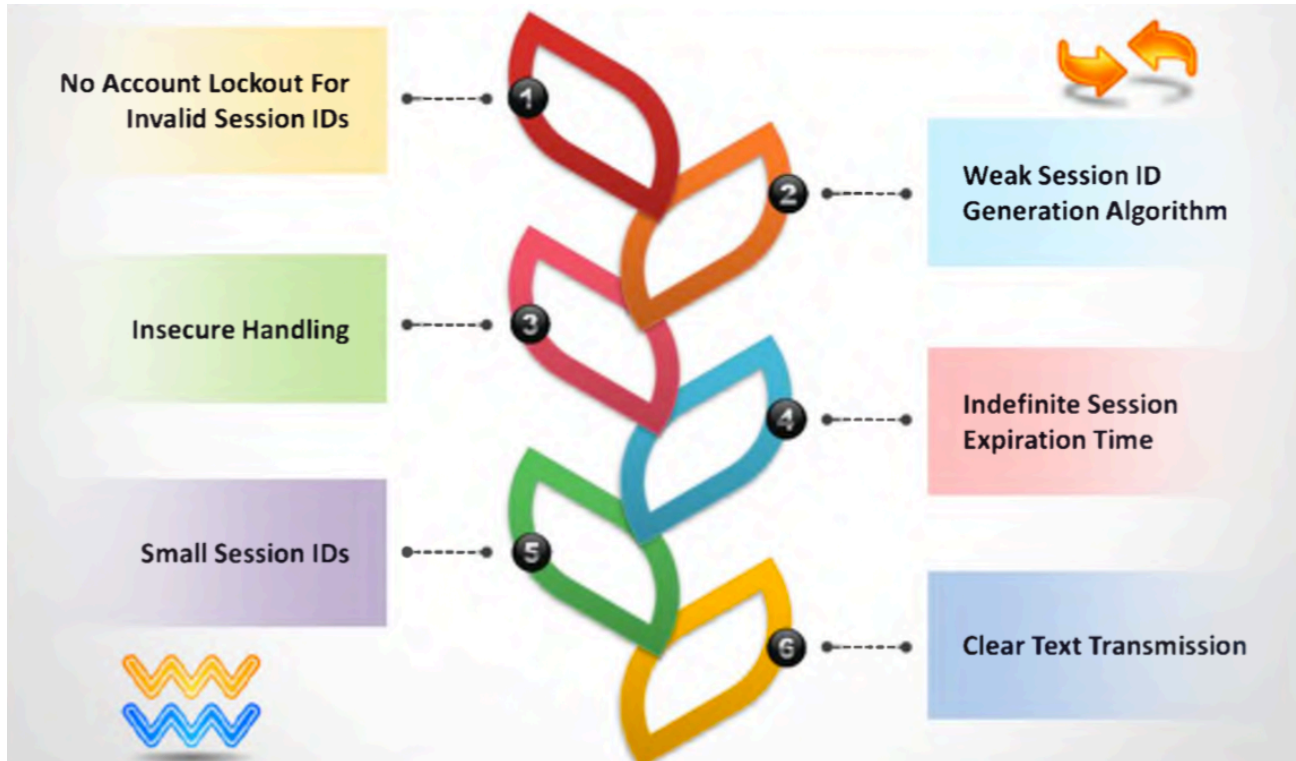


Figure 11.1: Example of session hijacking

Why Session Hijacking Is Successful



Example: Linear Algorithm Generating Weak Session IDs

```
import time

def generate_session_id(user_id):
    timestamp = int(time.time()) # current time in seconds
    print(f"[DEBUG] Current UNIX timestamp: {timestamp}")
    return f"SID-{user_id}-{timestamp}"

# Example usage
session_id = generate_session_id("user123")
print("Generated Session ID:", session_id)
```

Session ID output:

```
[DEBUG] Current UNIX timestamp: 1713622405
Generated Session ID: SID-user123-1713622405
```


Improvement using secure, random, cryptographic session ID generator

```
import secrets

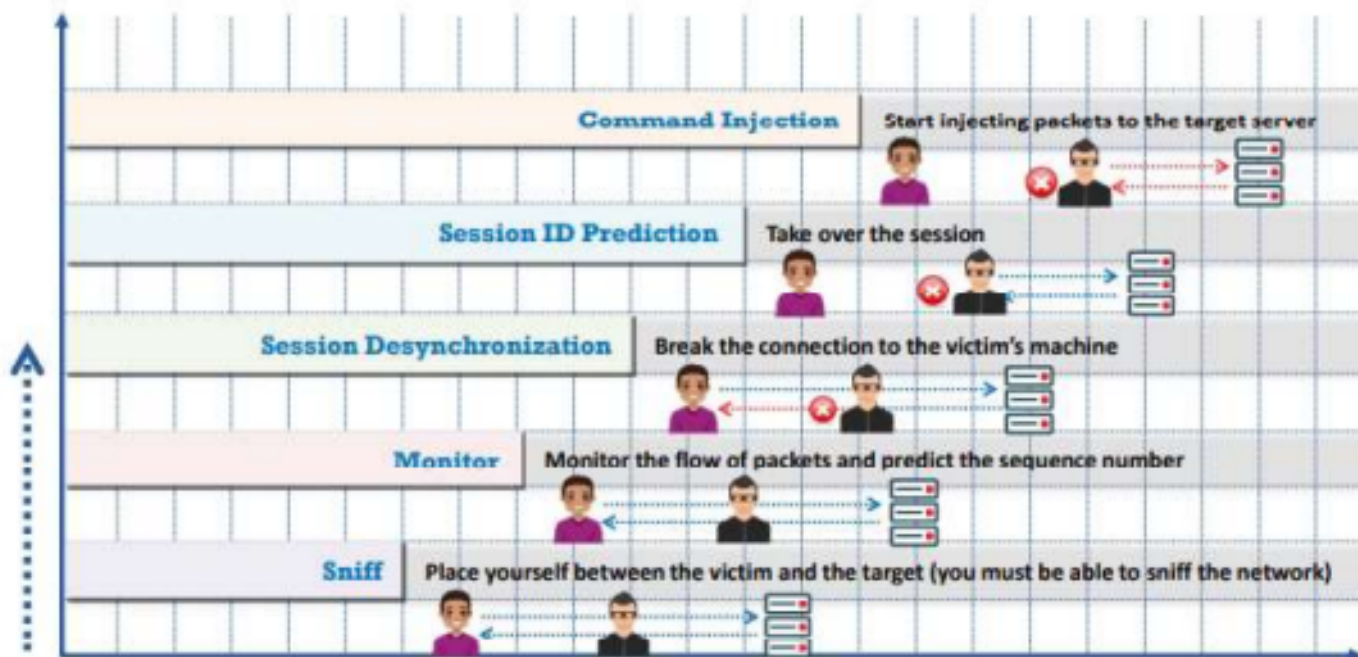
def generate_secure_session_id():
    return secrets.token_hex(32)

print(generate_secure_session_id())
```

Session ID output:

a1f5b9c3e90f15d27dc1f9c0f4bc2eb9fd40b8e6d1c7ad6d70a7f6633a9634f8

Session Hijacking Process



An attacker implements various techniques such as stealing, guessing, and brute forcing to obtain a valid session ID, which helps in acquiring control over a valid user's session while it is in progress.

Stealing:

Attackers use different techniques to steal session IDs.

- An attacker can steal the session key through physical access by, for example, acquiring the files containing session IDs or memory contents of either the user's system or the server.
- The attacker can also use sniffing tools such as Wireshark or SteelCentral Packet Analyzer to sniff the traffic between the client and server to extract the session IDs from the packets.

Guessing:

An attacker attempts to guess the session IDs by observing session variables. In the case of session hijacking, the range of session ID values that can be guessed is limited. Thus, guessing techniques are effective only when servers use weak or flawed session-ID generation mechanisms.

Brute forcing:

In the brute-force technique, an attacker obtains session IDs by attempting all possible permutations of session ID values until finding one that works. An attacker using a digital subscriber line (DSL) can generate up to 1,000 session IDs per second. This technique is most useful when the algorithm that produces session IDs is non-random.

Identifying an Active Session

- Session hijacking builds on sniffing techniques
- It involves not only observing traffic but also taking control of an active session
- The attacker targets a session that has already been authenticated
- For a successful attack, the attacker must identify a valid and active session

This is challenging due to:

- ✓ Network segmentation
- ✓ Switched networks
- ✓ Encryption mechanisms

Identifying an Active Session (Cont.)

■ Sequence number facts:

- TCP sequence numbers are 32-bit values
→ Over 4 billion possible combinations
- They ensure packets are:
 - Ordered correctly
 - Reassembled properly at the destination

To hijack a TCP session, the attacker must:

Predict or obtain valid sequence numbers

Identifying an Active Session (Cont.)

■ Sequence number prediction:

❑ A TCP session begins with the 3-way handshake:

- 1) Client → Server: SYN
- 2) Server → Client: SYN/ACK
- 3) Client → Server: ACK

❑ During this process:

- Each side selects an Initial Sequence Number (ISN)
- Modern systems use pseudo-random ISNs

❑ If sequence numbers are predictable:

- ✓ The attacker can forge packets
- ✓ Establish a connection using a spoofed IP address
- ✓ Potentially hijack the session

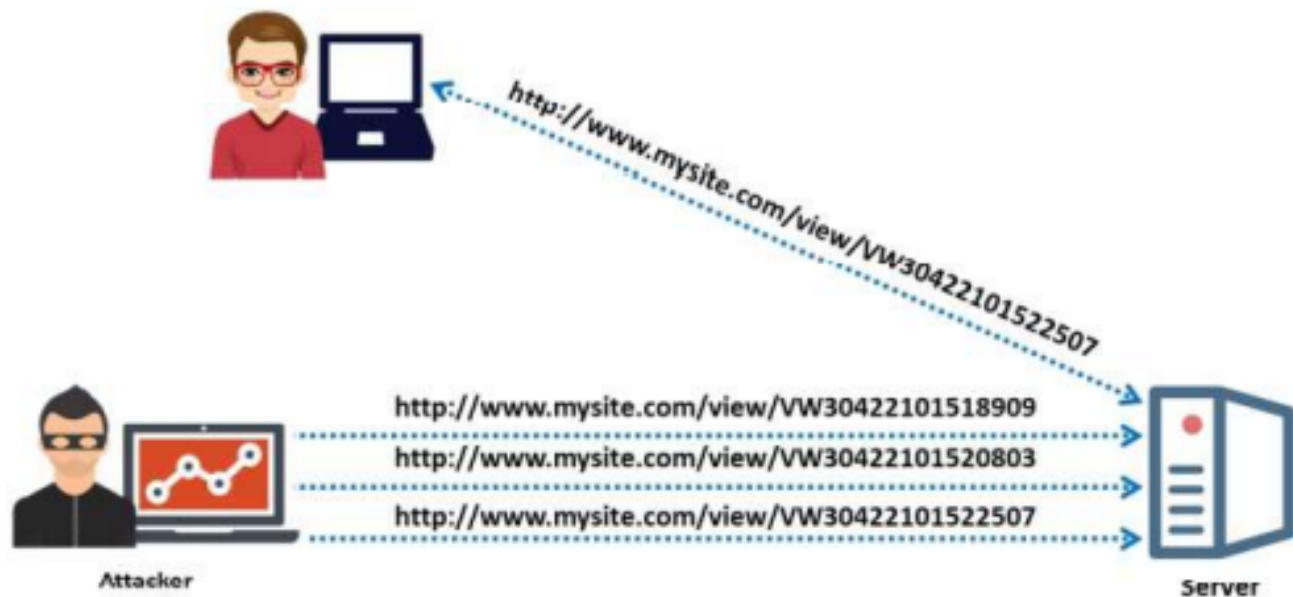
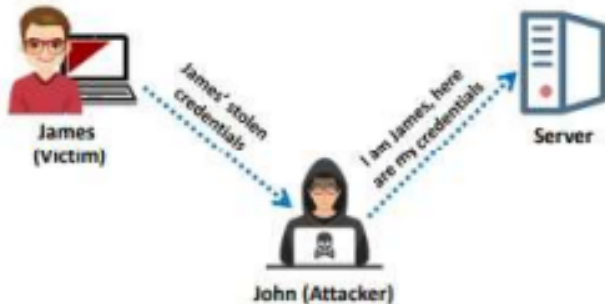


Figure 11.7: Brute-forcing attack on the session ID of a user

Spoofing vs Hijacking

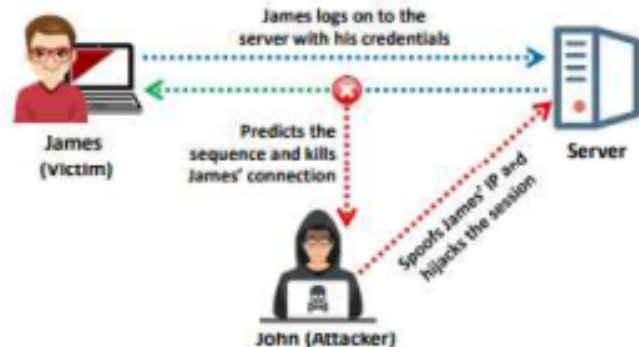
Spoofing Attack

- An attacker **pretends to be another user** or machine (victim) to gain access
- The attacker does not seize control of an existing active session; instead, he or she initiates a new session using the victim's **stolen credentials**



Hijacking

- Session hijacking is the process of seizing control of an **existing active session**
- The attacker relies on the **legitimate user** to create a connection and authenticate



Session Hijacking Tools

Ettercap

- A multiplatform tool that can perform man-in-the-middle attacks, ARP spoofing, and session hijacking

Hunt

- A commonly used tool designed to work on Ethernet-based networks in passive and active modes

Juggernaut

- A Linux network sniffer that provides the ability to hijack TCP sessions

Session Hijacking Tools (Cont.)

Paros
HTTP
Hijacker

- A Java utility that is an HTTP/HTTPS proxy that allows you to intercept and edit HTTP messages in real time

IP-
Watcher

- A commercial-grade tool that can perform session hijacking and monitor connections

T-sight

- A commercial offering that can hijack TCP sessions on a network

Thwarting Session Hijacking Attacks

Be proactive — use encryption

Configure routers to block spoofed traffic from outside the protected network

Use an intrusion detection system (IDS)

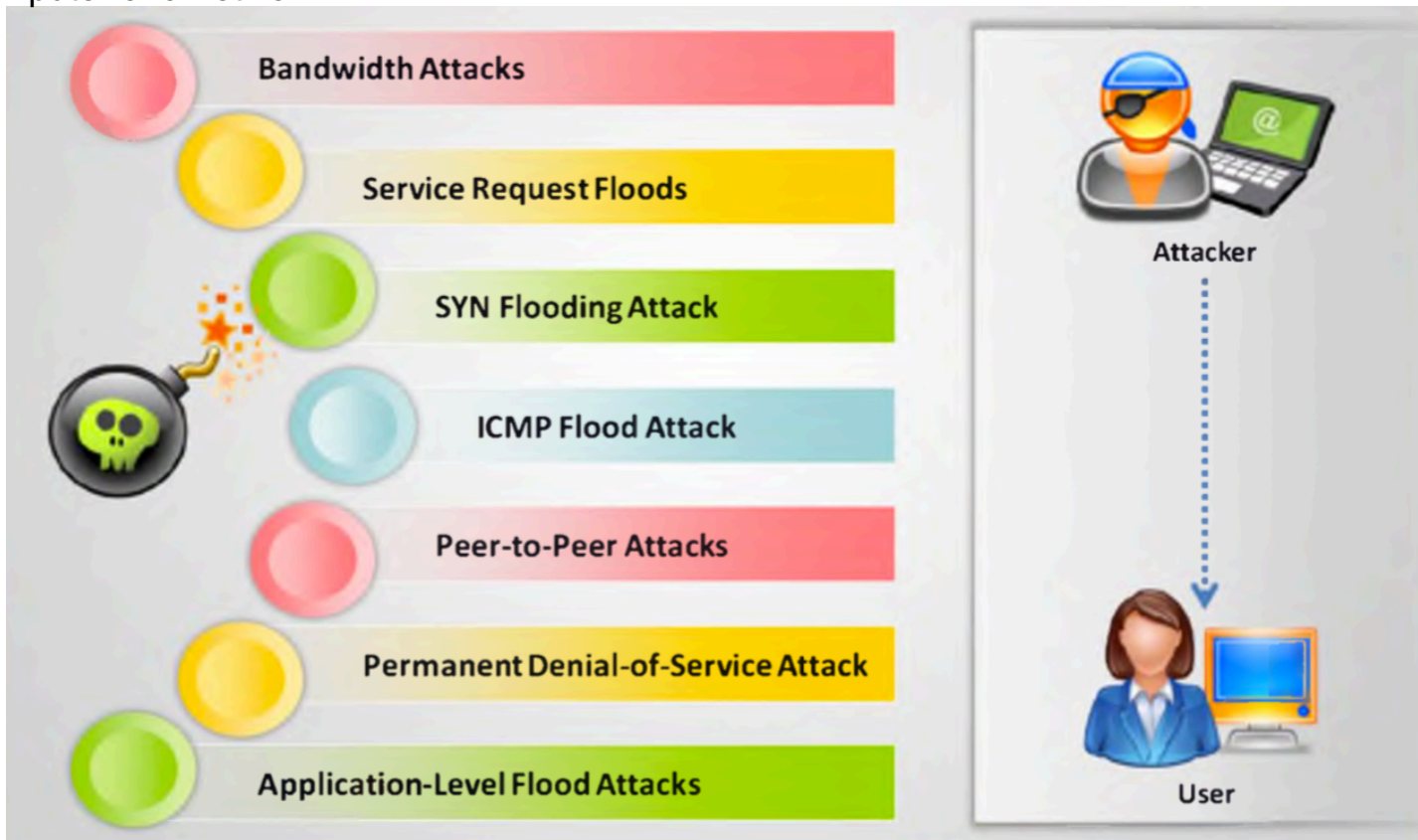
Denial of Service (DOS)

Denial of Service (DoS) Attacks

- Denial-of-service (DoS) is an attack that prevents authorized users from accessing a computer or network.
- DoS attacks target the network **bandwidth** or **connectivity**.
- **Bandwidth** attacks **overflow the network with a high volume of traffic** using existing network resources, thus depriving legitimate users of these resources.
- **Connectivity** attacks **overflow a computer with a large amount of connection requests**, consuming all available operating system resources, so that the computer cannot process legitimate user requests.

DoS Attack Techniques :

There are seven kinds of techniques that are used by the attacker to perform DOS attacks on a computer or a network.



Bandwidth Attacks

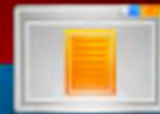
A single machine cannot make enough requests to overwhelm network equipment; hence DDoS attacks were created where an attacker uses **several computers** to flood a victim



Attackers use botnets and carry out DDoS attacks by flooding the network with **ICMP ECHO packets**



When a DDoS attack is launched, flooding a network, it can cause network equipment such as **switches** and **routers** to be overwhelmed due to the significant statistical change in the **network traffic**



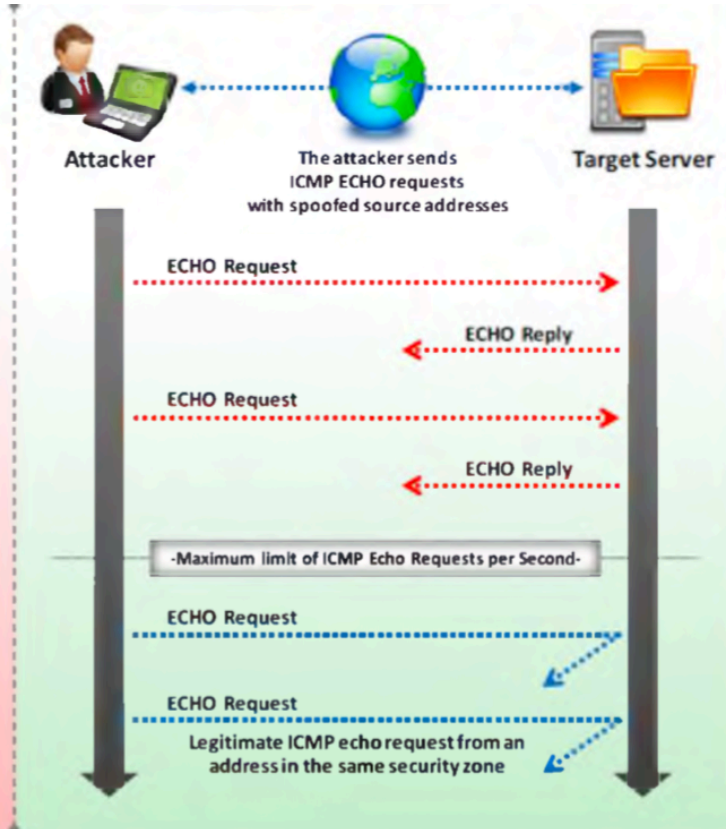
Basically, all bandwidth is used and no bandwidth remains for **legitimate use**

Service Request Floods



ICMP Flood Attack

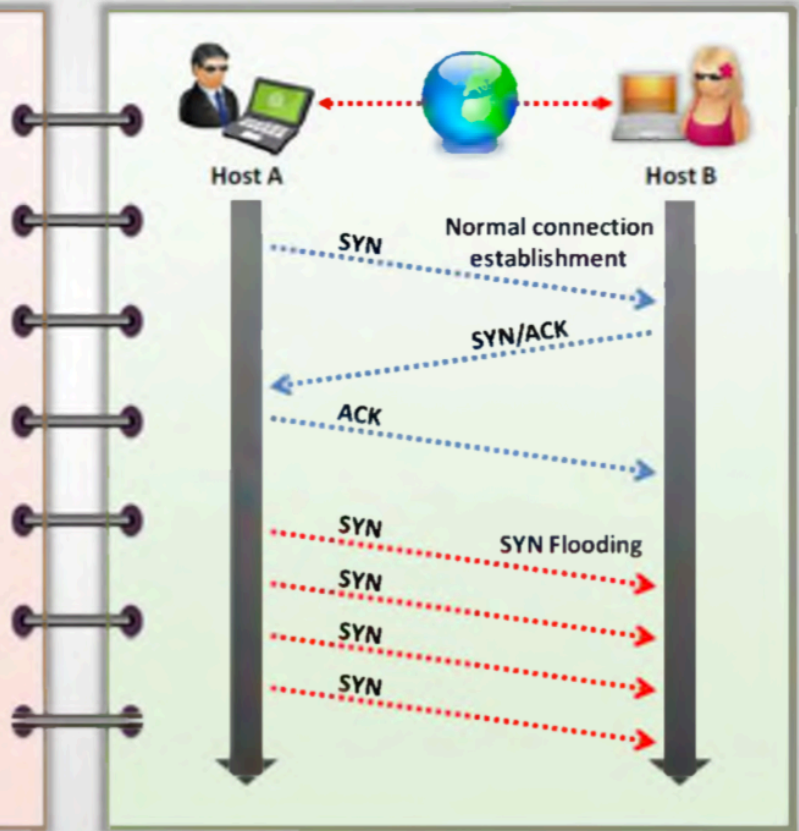
- ICMP is a type of DoS attack in which perpetrators send a large number of **packets with fake source addresses** to a target server in order to crash it and cause it to stop responding to TCP/IP requests
- After the ICMP threshold is reached, the router rejects further ICMP echo requests from all addresses in the **same security zone** for the remainder of the current second and the next second as well



- **ICMP flood** — Comes in two variants: Smurf attack and ping flood
- **Smurf attack** — Carried out when a large amount of traffic is directed to the broadcast address of a network instead of to a specific system; because the attacker configures the packet with the intended victim as the source, all hosts on the network respond to the victim instead of to the attack
- **Ping flood** — Carried out by sending a large number of ping packets to the victim with the intent of overwhelming the victim; very simple attack

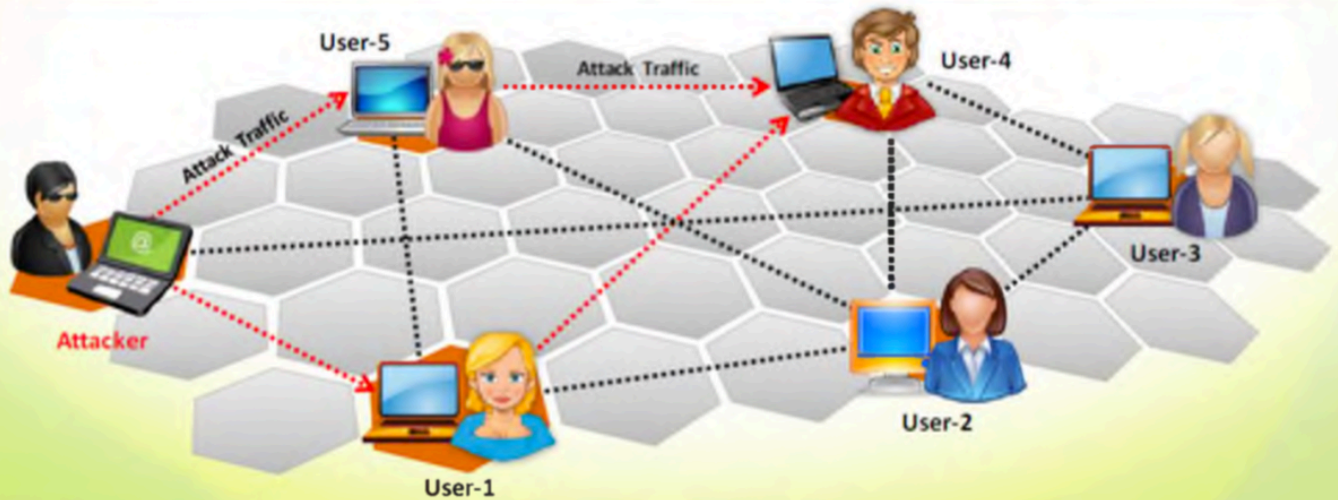
SYN Flooding

- SYN Flooding takes advantage of a flaw in how most hosts implement the TCP **three-way handshake**
- When Host B receives the SYN request from A, it must keep track of the partially-opened connection in a "**listen queue**" for at least 75 seconds
- A malicious host can exploit the small size of the listen queue by sending multiple SYN requests to a host, but **never replying** to the SYN/ACK
- The victim's listen queue is **quickly filled up**
- This ability of **removing a host** from the network for at least 75 seconds can be used as a denial-of-service attack



Peer-to-Peer Attacks

- Using peer-to-peer attacks, attackers **instruct clients of peer-to-peer file sharing hubs** to disconnect from their peer-to-peer network and to connect to the victim's fake website
- Attackers **exploit flaws** found in the network using DC++ (Direct Connect) protocol, that is used for sharing all types of files between instant messaging clients
- Using this method, attackers launch **massive denial-of-service attacks** and compromise websites



Application-Level Flood Attacks

- Application-level flood attacks result in the **loss of services** of a particular network, such as emails, network resources, the temporary ceasing of applications and services, and more
- Using this attack, attackers **destroy programming source code and files** in affected computer systems



Using application-level flood attacks, attackers attempts to:

- Flood** web applications to legitimate user traffic
- Disrupt** service to a specific system or person, for example, blocking a user's access by repeating invalid login attempts
- Jam** the application-database connection by crafting malicious SQL queries



Attacker exploiting application source code

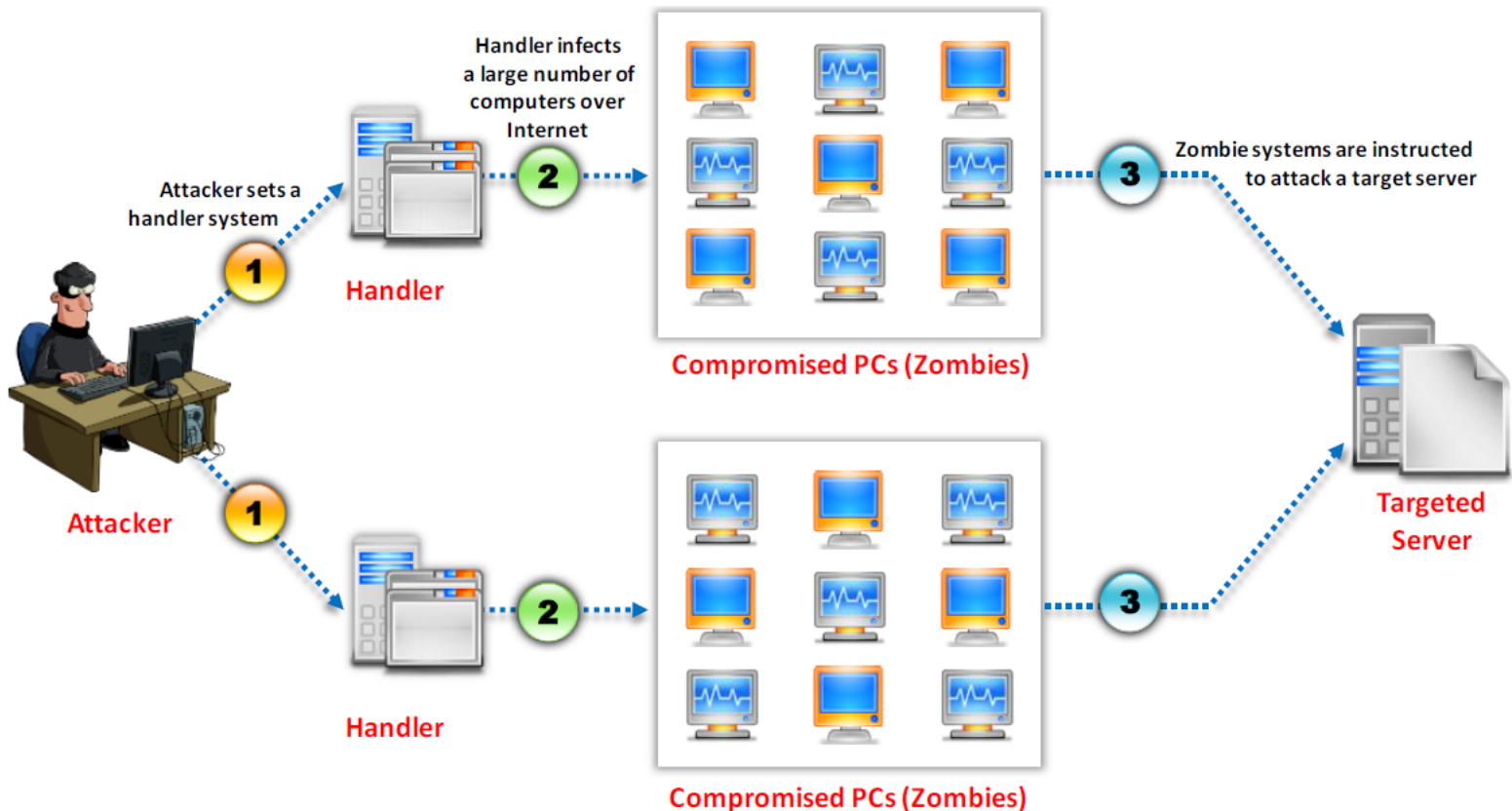


Attacker

Victim

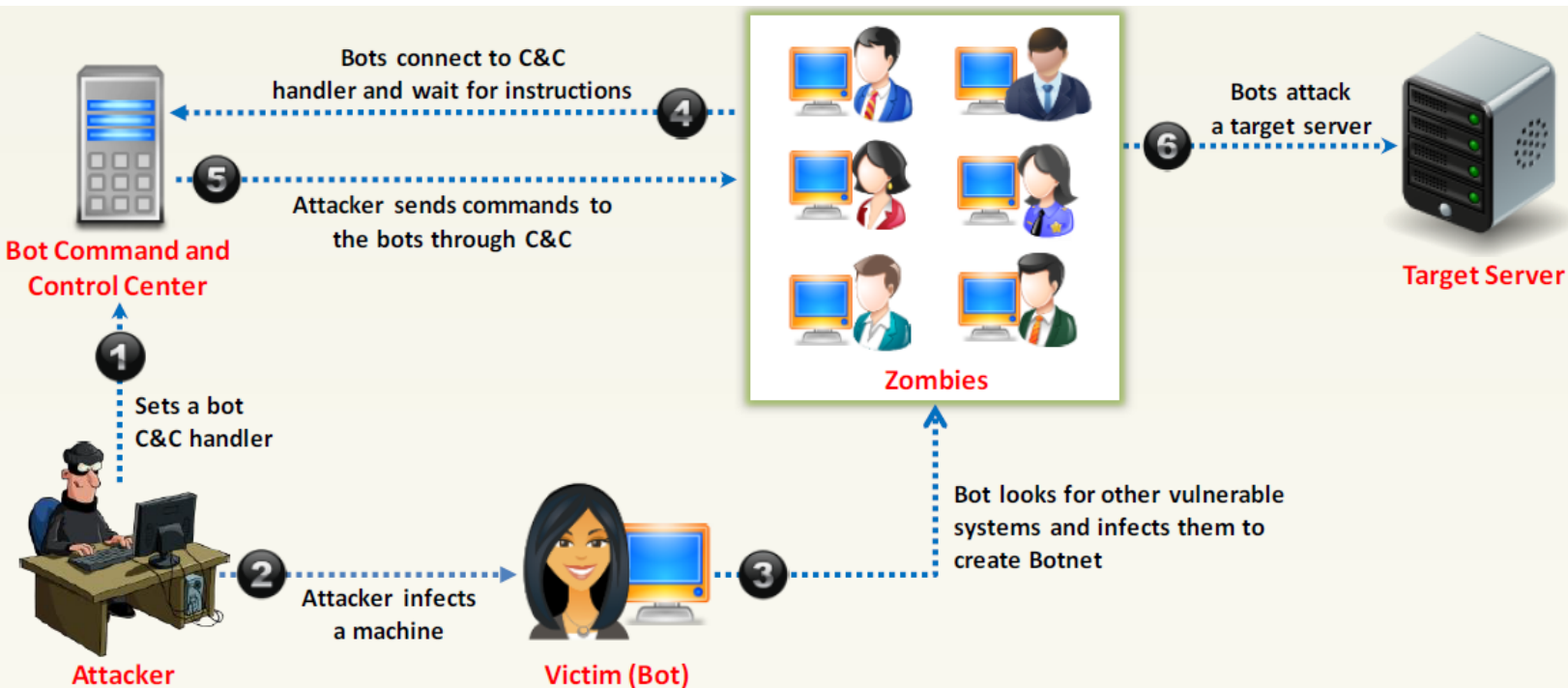
What are Distributed Denial of Service Attacks?

- A distributed denial-of-service (DDoS) attack involves a multitude of compromised systems attacking a single target, thereby causing denial of service for users of the targeted system.



Botnet

- Bots are software applications that run automated tasks over the Internet and perform simple repetitive tasks, such as web spidering and search engine indexing.
- A botnet is a huge network of the compromised systems and can be used by an attacker to launch denial-of-service attacks.



Summary

- Network sniffing and traffic analysis
- Session hijacking
- Denial of service (DoS)
- Distributed denial of service (DDoS) attacks
- Botnets