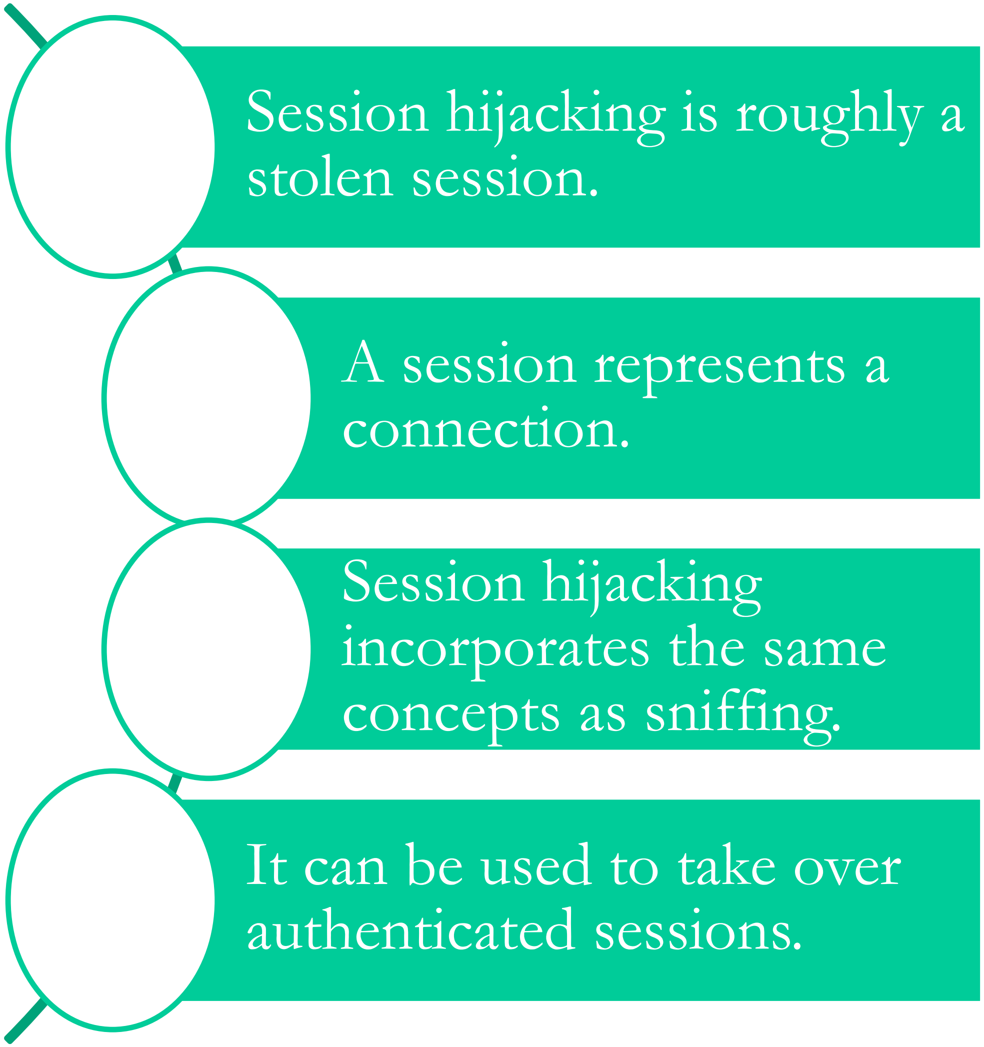
SESSION HIJACKING

Chapter 12

What Is Session Hijacking?

Session hijacking, also known as TCP session hijacking, is a method of taking over a web user session by surreptitiously obtaining the session ID and masquerading as the authorized user.



Session hijacking is roughly a stolen session.

A session represents a connection.

Session hijacking incorporates the same concepts as sniffing.

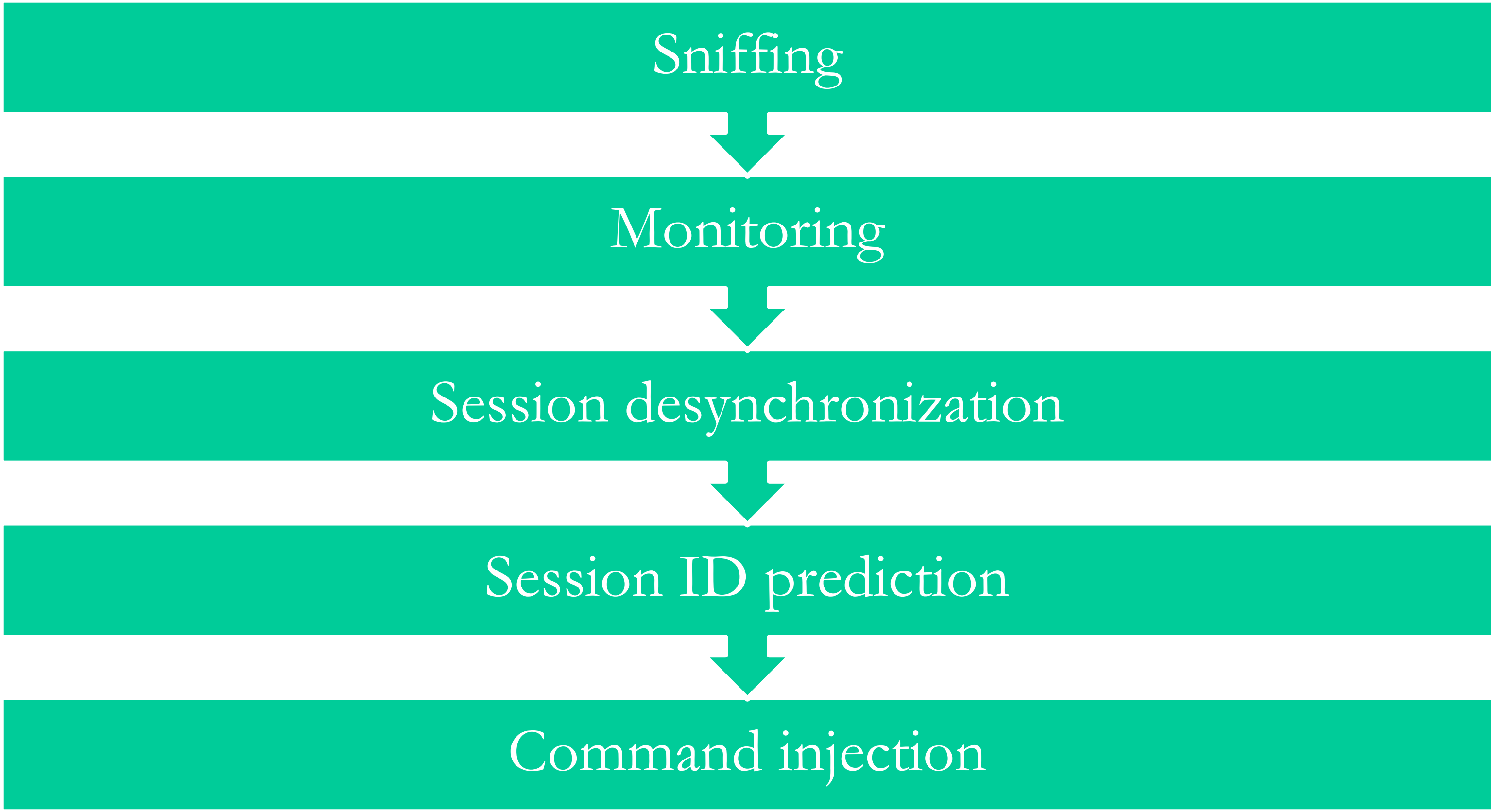
It can be used to take over authenticated sessions.

Understanding Session Hijacking

- No account lockout for invalid session IDs
- Insecure handling
- Weak session ID generation algorithm
- Indefinite session expiration time
- Cleartext transmission
- Small session IDs

Spoofing vs. Hijacking

Spoofing occurs when an attacking party impersonates an identity. In hijacking, the attacker takes over an existing active session.



Sniffing

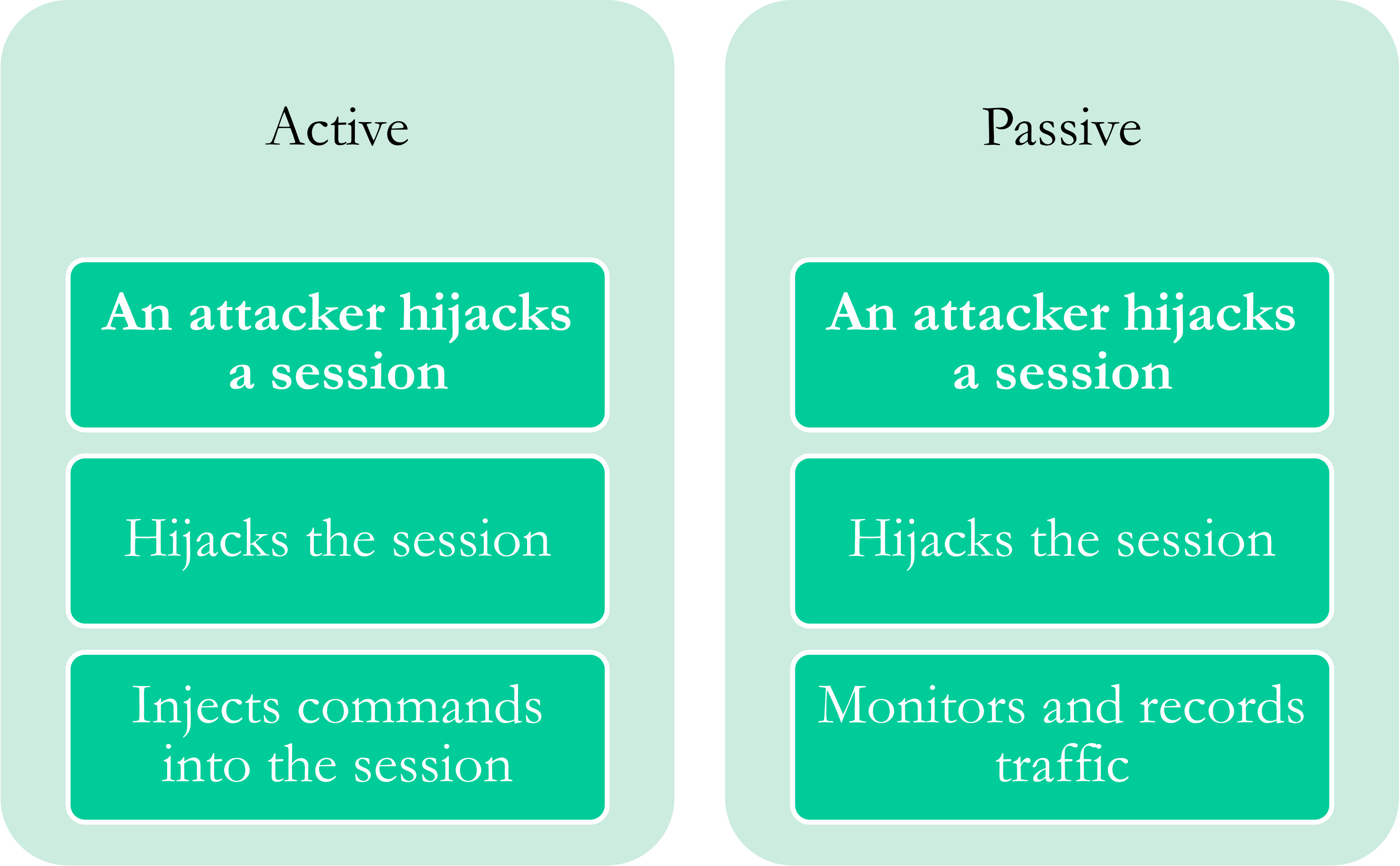
Monitoring

Session desynchronization

Session ID prediction

Command injection

Types of Session Hijacking



Active

An attacker hijacks a session

Hijacks the session

Injects commands into the session

Passive

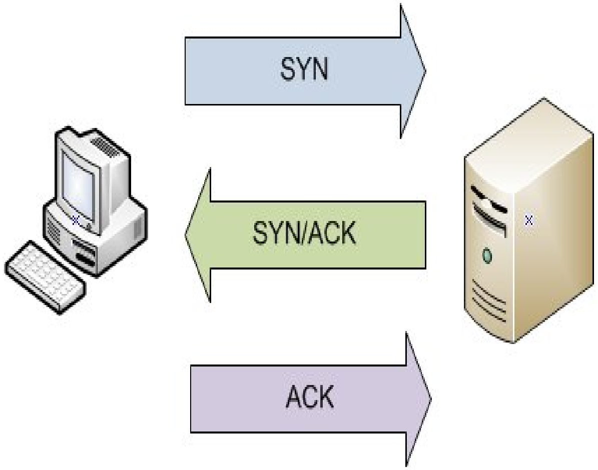
An attacker hijacks a session

Hijacks the session

Monitors and records traffic

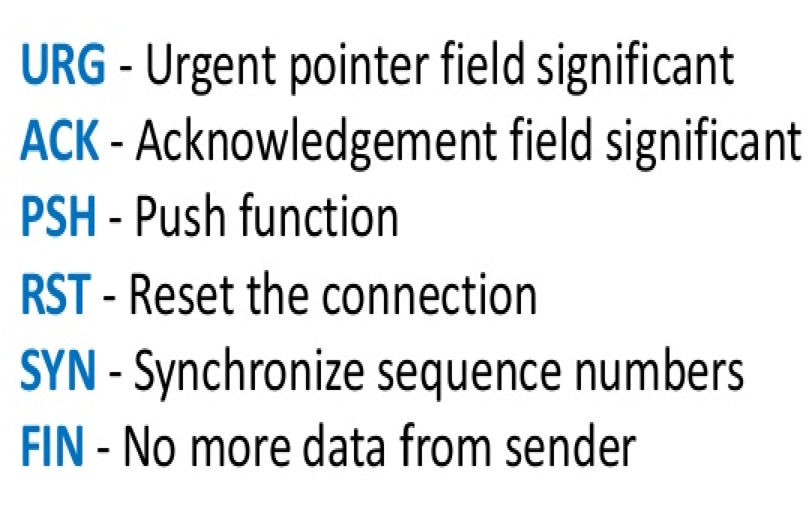
TCP and the Three-Way Handshake

TCP establishes connections and then verifies that each and every packet makes it to their destination in the right order. To accomplish this, TCP uses the three-way handshake.



- Ports can be TCP or UDP.
- TCP is a connection-oriented protocol.
- The three-way handshake is used to establish a connection.
- The completion of the three-way handshake is used before sending packets.
- The three-way handshake does not handle security.
- TCP also provides sequence numbers for reassembly of data.

TCP Flags



URG - Urgent pointer field significant
ACK - Acknowledgement field significant
PSH - Push function
RST - Reset the connection
SYN - Synchronize sequence numbers
FIN - No more data from sender

SYN: Used to initiate a connection between two different hosts in order to facilitate communications.

ACK: Used to acknowledge the receipt of a packet of information.

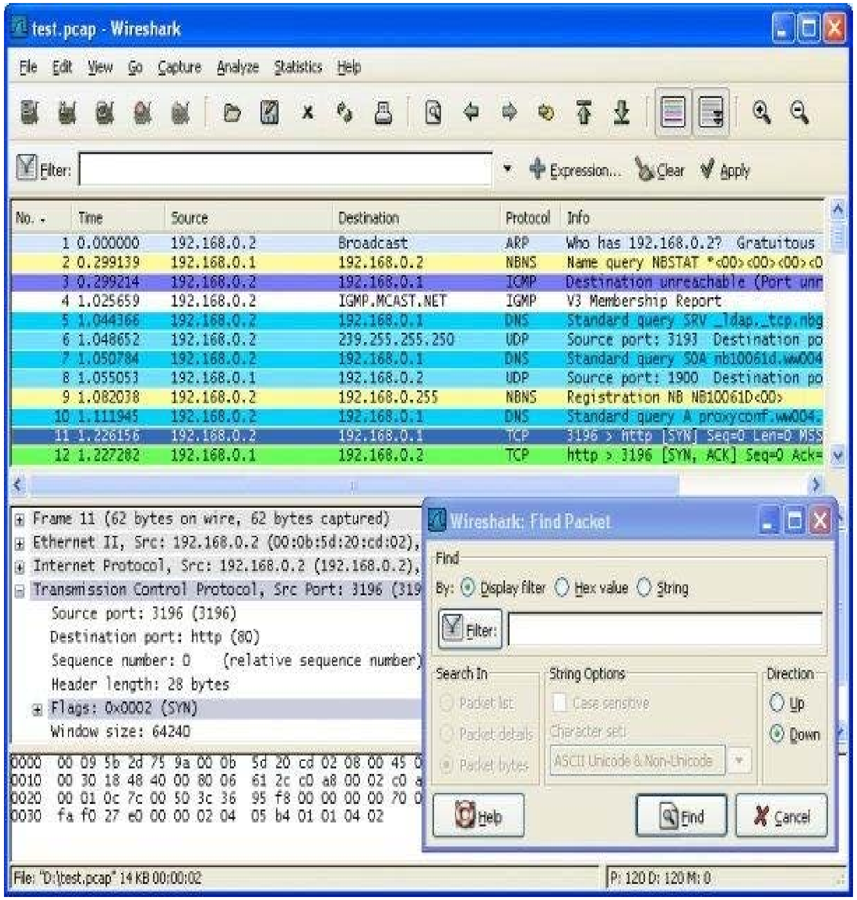
URG: States that the data contained in the packet should be processed immediately.

PSH: Instructs the sending system to send all buffered data immediately.

FIN: Tells the remote system that no more information will be sent. In essence this is gracefully closing a connection.

RST: A reset packet that is used to reset a connection.

TCP Sequence Numbers


test.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

```
No. -  Time       Source         Destination      Protocol  Info
    1 0.000000   192.168.0.2    Broadcast        ARP       Who has 192.168.0.2?  Gratuitous
    2 0.299139   192.168.0.1    192.168.0.2      NBNS      Name query NBSTAT *<00><00><00><0
    3 0.299214   192.168.0.2    192.168.0.1      ICMP      Destination unreachable (Port unr
    4 1.025659   192.168.0.2    IGMP.MCAST.NET   IGMP      V3 Membership Report
    5 1.044366   192.168.0.2    192.168.0.1      DNS       Standard query SRV _ldap._tcp.nbg
    6 1.048652   192.168.0.2    239.255.255.250  UDP       Source port: 3193  Destination po
    7 1.050784   192.168.0.2    192.168.0.1      DNS       Standard query SOA nb10061d.ww004
    8 1.055053   192.168.0.1    192.168.0.2      UDP       Source port: 1900  Destination po
    9 1.082038   192.168.0.2    192.168.0.255    NBNS      Registration NB NB10061D<00>
   10 1.111945   192.168.0.2    192.168.0.1      DNS       Standard query A proxyconf.ww004.
   11 1.226156   192.168.0.2    192.168.0.1      TCP       3196 > http [SYN] Seq=0 Len=0 MSS
   12 1.227282   192.168.0.1    192.168.0.2      TCP       http > 3196 [SYN, ACK] Seq=0 Ack=
```

```
⊞ Frame 11 (62 bytes on wire, 62 bytes captured)
⊞ Ethernet II, Src: 192.168.0.2 (00:0b:5d:20:cd:02),
⊞ Internet Protocol, Src: 192.168.0.2 (192.168.0.2),
⊟ Transmission Control Protocol, Src Port: 3196 (319
    Source port: 3196 (3196)
    Destination port: http (80)
    Sequence number: 0    (relative sequence number)
    Header length: 28 bytes
  ⊞ Flags: 0x0002 (SYN)
    Window size: 64240
```

```
0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 0
0010  00 30 18 48 40 00 80 06  61 2c c0 a8 00 02 c0 a
0020  00 01 0c 7c 00 50 3c 36  95 f8 00 00 00 00 70 0
0030  fa f0 27 e0 00 00 02 04  05 b4 01 01 04 02
```

Wireshark: Find Packet

Find

By: ● Display filter ○ Hex value ○ String

Filter:

Search In: ○ Packet list ○ Packet details ● Packet bytes

String Options: □ Case sensitive; Character set: ASCII Unicode & Non-Unicode

Direction: ○ Up ● Down

File: "D:\test.pcap" 14 KB 00:00:02

P: 120 D: 120 M: 0

Sequence number describes order of packets

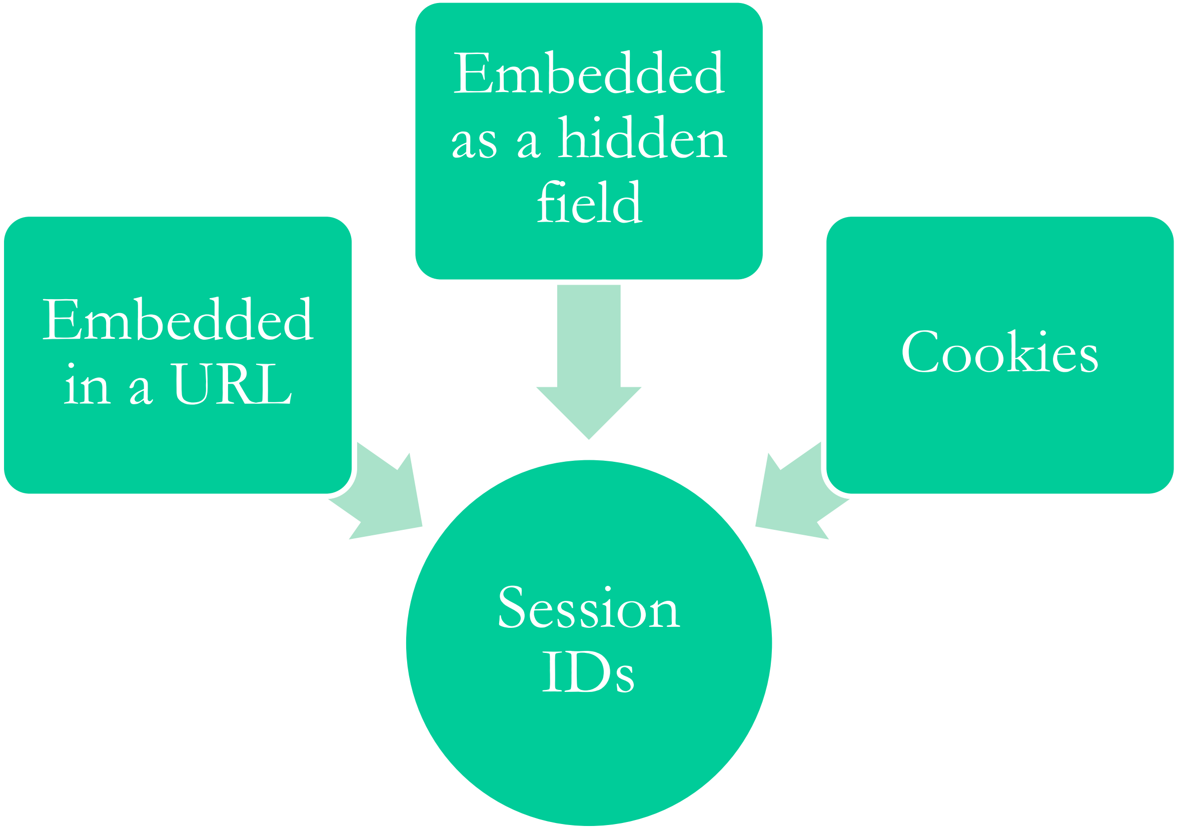
Incremented during transmission of each packet

Starts from an initial sequence number (ISN)

32-bit number

Session Hijacking and Web Applications

Session hijacking at the application level focuses on gaining access to a host by obtaining legitimate session IDs from the victim.



Application-Level Hijacking

Session sniffing

Predicting session tokens

- /app/spo22022005131020
- /app/spo22022005141520
- /app/spo22022005171126
- /app/spo22022005213111

Man-in-the-middle attack

Man-in-the-browser attack

- Browser helper objects
- Extensions
- API hooking
- JavaScript

Cross-Site Scripting

Cross-site scripting (XSS) is a type of attack that can occur in many forms, but in general it occurs when data of some type enters a web application through an untrusted source.

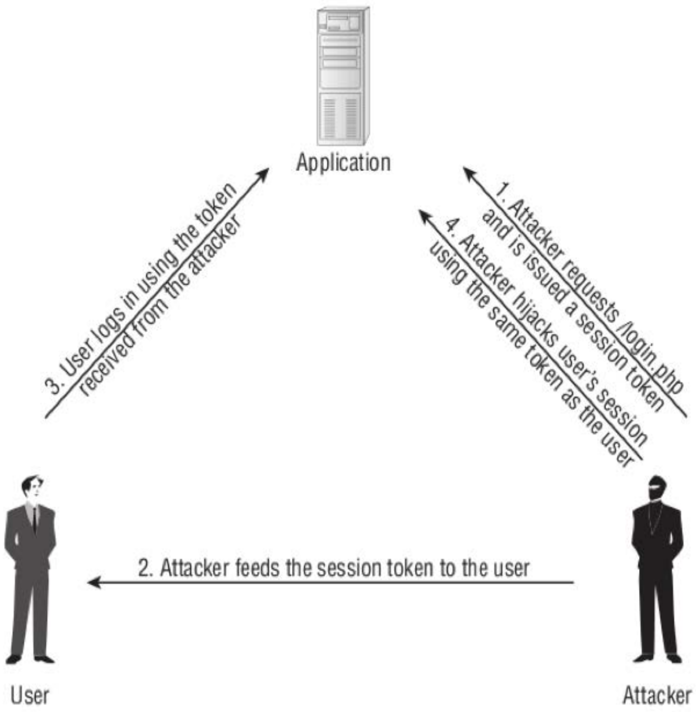
Stored attacks

- The attacker stores malicious code in the vulnerable page.
- The user authenticates in the application.
- The user visits a vulnerable page.
- Malicious code is executed by the user's browser.

Reflected attacks

- It is in the form of an email or via a different web server.
- It occurs when a party injects executable code within an HTTP response.
- The code is not persistent and is not stored.
- It leverages JavaScript, VBScript, or other scripting languages where appropriate.

Session Fixation


Application
1. Attacker requests /login.php and is issued a session token
4. Attacker hijacks user's session using the same token as the user
3. User logs in using the token received from the attacker
2. Attacker feeds the session token to the user
User
Attacker

A session ID is sent to a victim in a malicious hyperlink for the victim to click.

The victim is tricked into authenticating to a target using an attacker-created login form.

The attacker uses injection to insert malicious code in the hyperlink.

The HTTP header response uses the server to fix the session ID in the victim's browser.

Key Concepts

- Blind hijacking
- IP spoofing
- Source routing
- DNS spoofing
- ARP cache poisoning

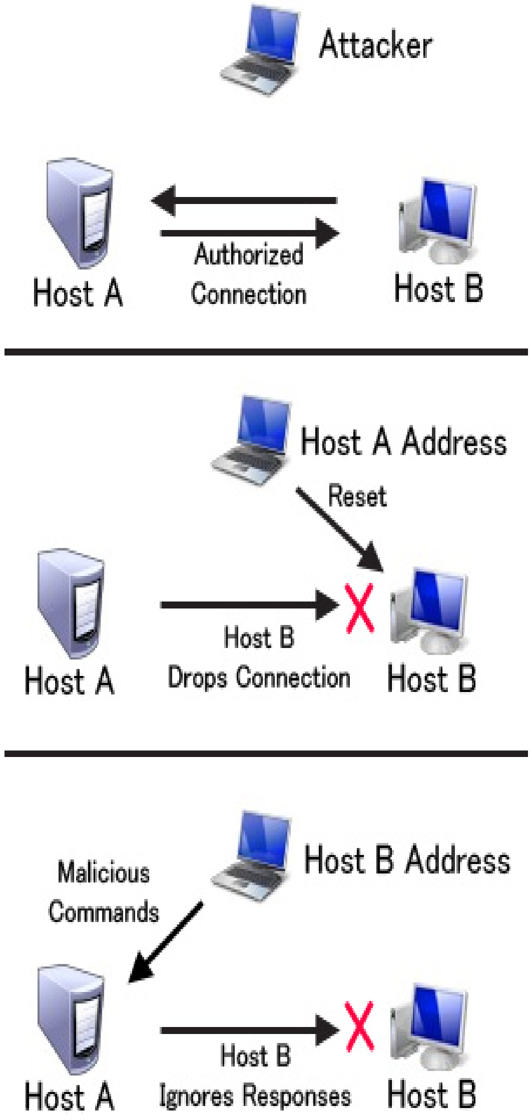
Network Session Hijacking

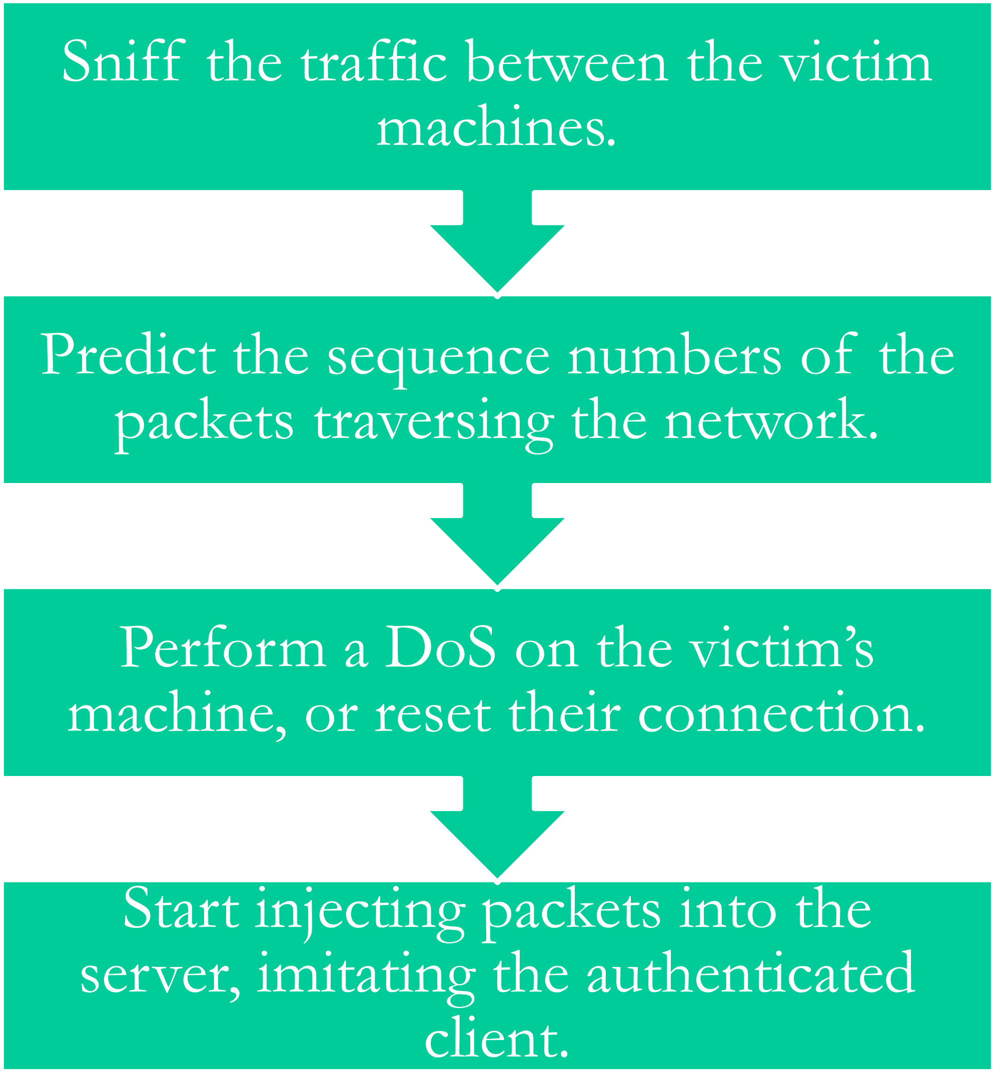
- Blind hijacking
- IP spoofing
- Source routing
- DNS spoofing
- ARP cache poisoning
- Desynchronizing the connection

Network Session Hijacking

- Blind hijacking
- IP spoofing
- Source routing
- DNS spoofing
- ARP cache poisoning
- Desynchronizing the connection

TCP Session Hijacking





Sniff the traffic between the victim machines.

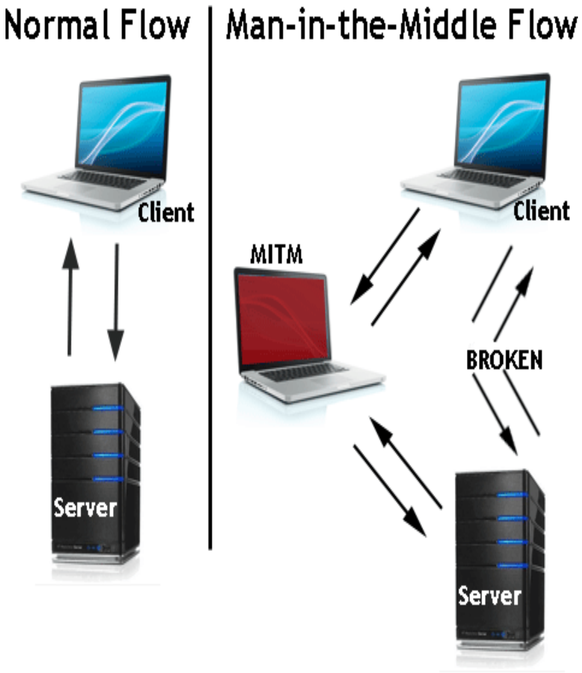
Predict the sequence numbers of the packets traversing the network.

Perform a DoS on the victim’s machine, or reset their connection.

Start injecting packets into the server, imitating the authenticated client.

Man-in-the-Middle (MitM)

Once attackers are in the middle of the connection via a technique such as ARP poisoning, they can monitor or manipulate traffic.



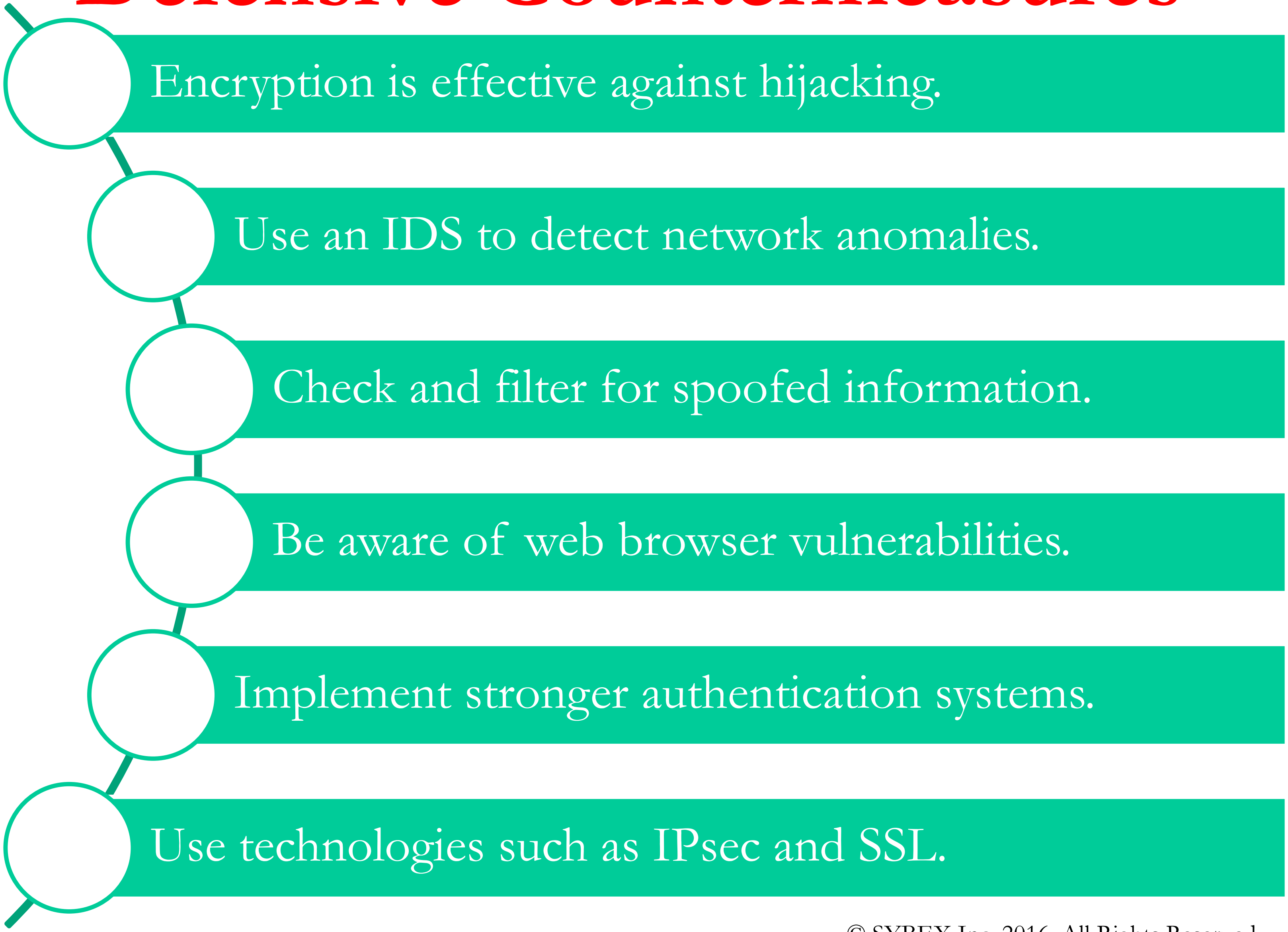
Network Level

- PacketCreator
- Ettercap
- Dsniff
- Cain & Abel

Application Level

- OWASP WebScarab
- Paros Proxy
- Burp Suite
- ProxyFuzz
- Odysseus Proxy
- Fiddler

Defensive Countermeasures



- Encryption is effective against hijacking.
- Use an IDS to detect network anomalies.
- Check and filter for spoofed information.
- Be aware of web browser vulnerabilities.
- Implement stronger authentication systems.
- Use technologies such as IPsec and SSL.

Summary

- What is TCP or session hijacking?
- How is session hijacking performed?
- Different formats of session hijacking
- Active or passive session hijacks
- Results of a successful attack