Chapter 11: Session Hijacking

Technology Brief

The concept of session hijacking is an interesting topic among other scenarios. It is basically hijacking of sessions by intercepting the communication between hosts. The attacker usually intercepts the communication to obtain the roles of authenticated user or for the intention of Man-in-the-Middle attack.

Session Hijacking

In order to understand the session hijacking concept, assume an authenticated TCP session between two hosts. The attacker intercepts the session and takes over the legitimate authenticated session. When a session authentication process is complete, and the user is authorized to use resources such as web services, TCP communication or other, the attacker takes advantage of this authenticated session and places himself in between the authenticated user and the host. Authentication process initiates at the start of TCP session only, once the attacker successfully hijacks the authenticated TCP session, traffic can be monitored, or attacker can get the role of the legitimate authenticated user. Session hijacking becomes successful because of weak session IDs or no blocking upon receiving an invalid session ID.

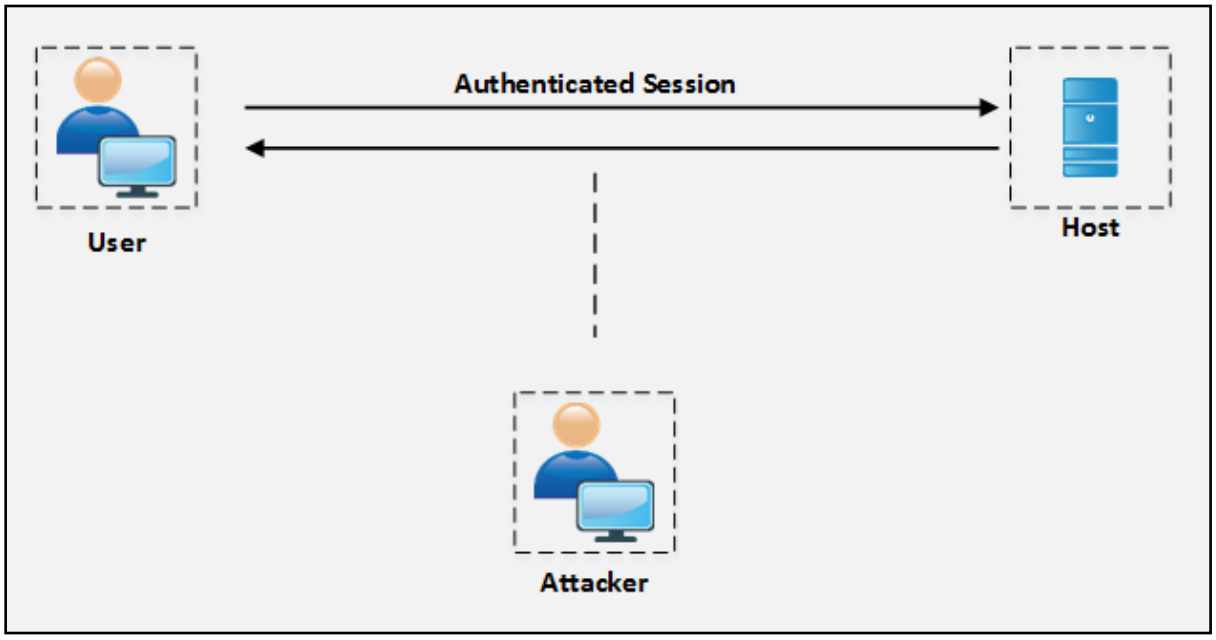


Figure 11-01 Session Hijacking

Session Hijacking Techniques

Session Hijacking process is categorized into the following three techniques:

Stealing

Stealing category includes the different technique of stealing session ID such as "Referrer attack" network sniffing, Trojans or by any other mean.

Guessing

Guessing category include tricks and techniques used to guess the session ID such as by observing the variable components of session IDs or calculating the valid session ID by figuring out the sequence etc.

Brute-Forcing

Brute-Forcing is the process of guessing every possible combination of credential. Usually, Brute-Forcing is performed when an attacker gains information about the range of Session ID.

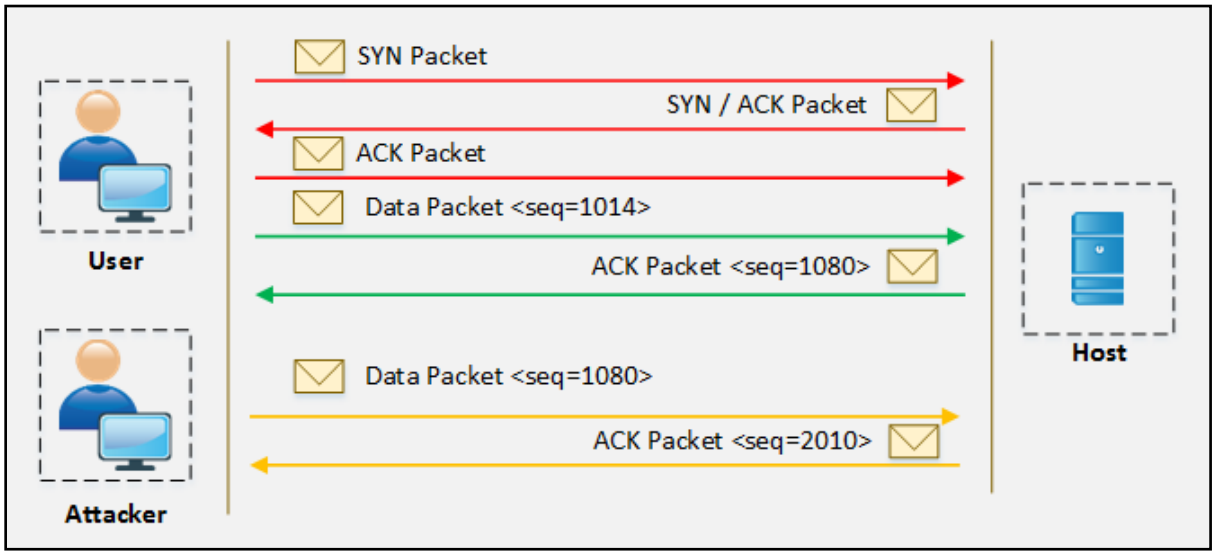


Figure 11-02 Brute-Forcing

Session Hijacking Process

The process of session hijacking involves: -

Sniffing

Attacker attempt to place himself in between victim and target in order to sniff the packet.

Monitoring

Monitor the traffic flow between victim and target.

Session Desynchronization

The process of breaking the connection between the victim and the target.

Session ID

Attacker takes control over the session by predicting the session ID.

Command Injection

After successfully taking control over the session, the attacker starts injecting the commands.

Types of Session Hijacking

Active Attack

The active attack includes interception in the active session from the attacker. An attacker may send packets to the host in the active attack. In an active attack, the attacker is manipulating the legitimate users of the connection. As the result of an active attack, the legitimate user is disconnected from the attacker.

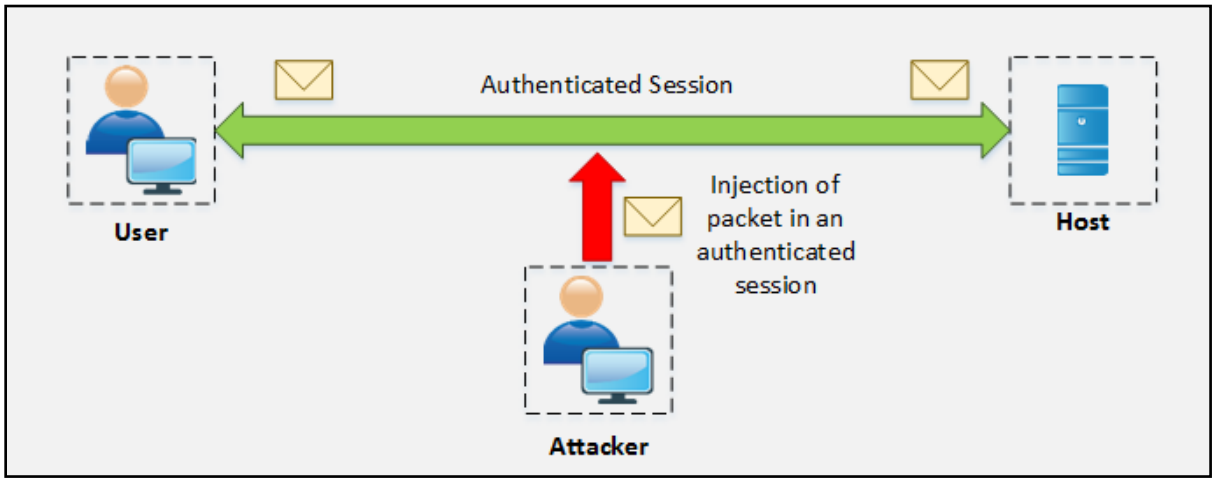


Figure 11-03 Active Attack

Passive Attack

The passive attack includes hijacking a session and monitoring the communication between hosts without sending any packet.

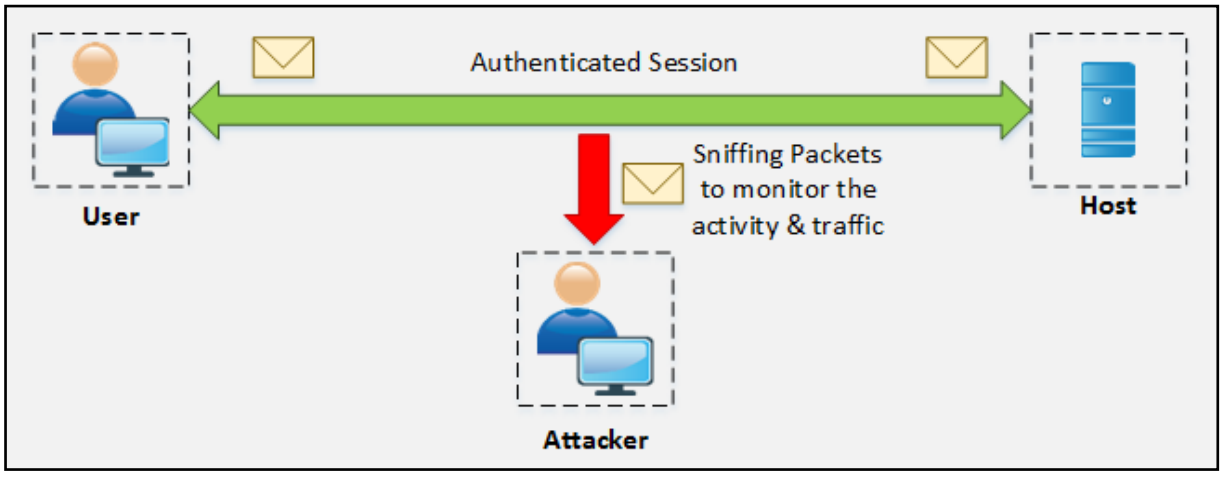


Figure 11-04 Passive Attack

Session Hijacking in OSI Model

Network Level Hijacking

Network level hijacking includes hijacking of a network layer session such as TCP or UDP session.

Application Level Hijacking

Application level hijacking includes hijacking of Application layer such as hijacking HTTPS session.

Network-Level Hijacking and Application-Level Hijacking are discussed in detail later in this chapter.

Spoofing vs. Hijacking

The major difference between Spoofing and Hijacking is of the active session. In a spoofing attack, the attacker is pretending to be another user by impersonating to gain access. The attacker does not have any active session; it initiates a new session with the target with the help of stolen information.

Hijacking is basically the process of taking control over an existing active session between an authenticated user and a target host. The attacker uses the authenticated session of a legitimate user without initiating a new session with the target.

Application Level Session Hijacking

Application-Level Hijacking Concept

Session hijacking as defined focuses on the application layer of the OSI model. In the application layer hijacking process, the attacker is looking for a legitimate session ID from the victim in order to gain access to an authenticated session which allows the attacker to avail web resources. For example, attacker, with an application layer hijacking can access the website resources secured for authenticated users only. The web server may assume that the incoming request forms the known host whereas an attacker has been hijacked the session by predicting the session ID.

Compromising Session IDs using Sniffing

Session sniffing is another flavor of sniffing in which an attacker is looking for the session ID / Session Token. Once the attacker has the found the session ID, it can gain access to the resources.

Compromising Session IDs by Predicting Session Token

Predicting the session ID is the process of observing the currently occupied session IDs by the client. By observing the common and variable part of the session key, an attacker can guess the next session key.

How to Predict a Session Token?

Web servers normally use random session ID generation to prevent prediction however some web servers use customer defined algorithms to assign session ID. For example, as shown below:

```
http://www.example.com/ABCD01012017191710
http://www.example.com/ABCD01012017191750
http://www.example.com/ABCD01012017191820
http://www.example.com/ABCD01012017192010
```

After observing the above session IDs, you can easily identify the constant part and other variable parts. In the above example, **ABCD** is the constant part, **01012017** is a date. and the last section is the time. An attacker may attempt with the following session ID at 19:25:10

```
http://www.example.com/ABCD01012017192510
```

Compromising Session IDs Using Man-in-the-Middle Attack

The process of compromising the session ID using Man-in-the-Middle attack requires splitting of the connection between Victim and Web server into two connections, one of them between Victim-to-Attacker and another between Attacker-to-Server.

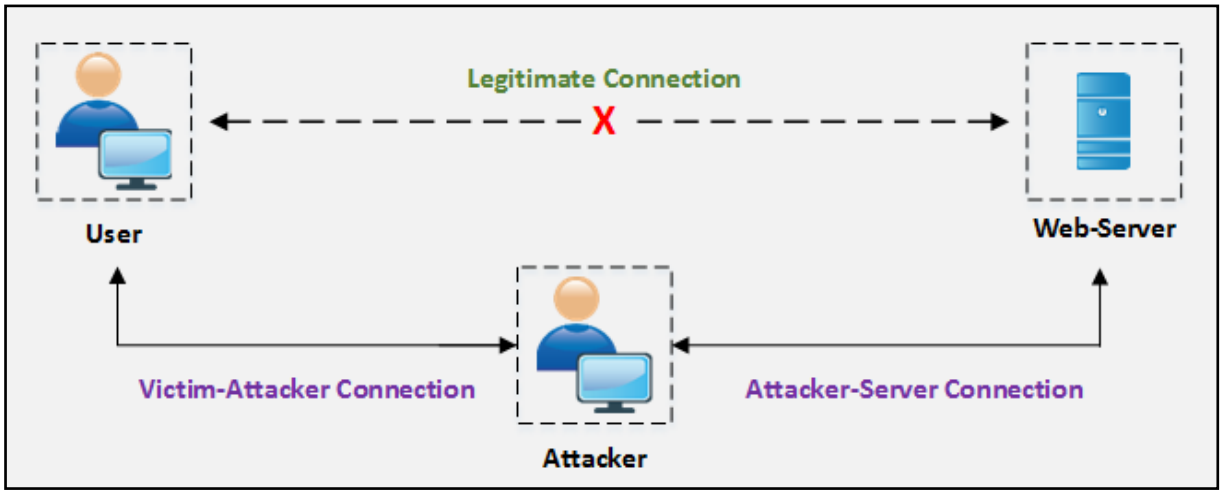


Figure 11-05 MITM Process

Compromising Session IDs Using Man-in-the-Browser Attack

Compromising Session ID using Man-in-the-Browser attack requires a Trojan, already deployed on the target machine. The trojan can either change the proxy settings, redirecting all traffic through the attacker whereas another technique using Trojan is that intercept the process between the browser and its security mechanism.

Steps to Perform Man-in-the-Browser Attack

To launch Man-in-the-Browser attack; the attacker first infected the victim's machine using a Trojan. Trojan installs malicious code in the form of an extension on the victim's machine and which modifies the browser's configuration upon boot. When a user logged into the site, URL is checked against a known list of the targeted website; the Event handler will register the event when it is detected. Using DOM interface attacker can extract and modify the values when the user clicks the button. The browser will send the form with modified entries to the web server. As the browser shows original transaction details, the user could not identify any interception.

Compromising Session IDs Using Client-side Attacks

Session IDs can be compromised easily by using Client-side attacks such as: -

1. Cross-Site Scripting (XSS)
2. Malicious JavaScript Code
3. Trojans

Cross-site Script Attack

Cross-site Scripting attack is performed by an attacker by sending a crafted link with a malicious script. When the user clicks this malicious link, the script will be executed. This script may be coded to extract the Session IDs and send it to the attacker.

Cross-site Request Forgery Attack

Cross-Site Request Forgery (CSRF) attack is the process of obtaining the session ID of a legitimate user and exploiting the active session with the trusted website in order to perform malicious activities.

Session Replay Attack

Another technique for session hijacking is Session Replay Attack. Attacker captures the authentication token from user intended for the server and replays the request to the server resulting in unauthorized access to the server.

Session Fixation

Session Fixation is an attack permitting the attacker to hijack the session. The attacker has to provide valid session ID and make the victim's browser to use it. It can be done y the following technique

1. Session Token in URL argument

2. Session Token in hidden form
3. Session ID in a cookie

To understand the Session Fixation attack, assume an attacker, victim, and the web server. The atttacker initiates a legitimate connection with the web server, issues a session ID or uses a new session ID. The attacker then sends the link to the victim with the established session ID for bypassing the authentication. When the user clicks the link and attempts to log into the website, web server continues the session as it is already established, and authentication is performed. Now, the attacker already has the session ID information will continue using a legitimate user account.

Network-level Session Hijacking

Network-Level hijacking is focused on Transport layer and Internet layer protocols used by the application layer. Network level attack results in extracting information which might be helpful for application layer session.

There are several types of network level hijacking including: -

- Blind Hijacking
- UDP Hijacking
- TCP/IP Hijacking
- RST Hijacking
- MITM
- IP Spoofing

The 3-Way Handshake

TCP communication initiates with the 3-way handshaking between requesting host and target host. In this handshaking Synchronization (SYN) packets and Acknowledgment (ACK) packets are communicated between them. To understand the flow of 3-way handshaking observe the following diagram.

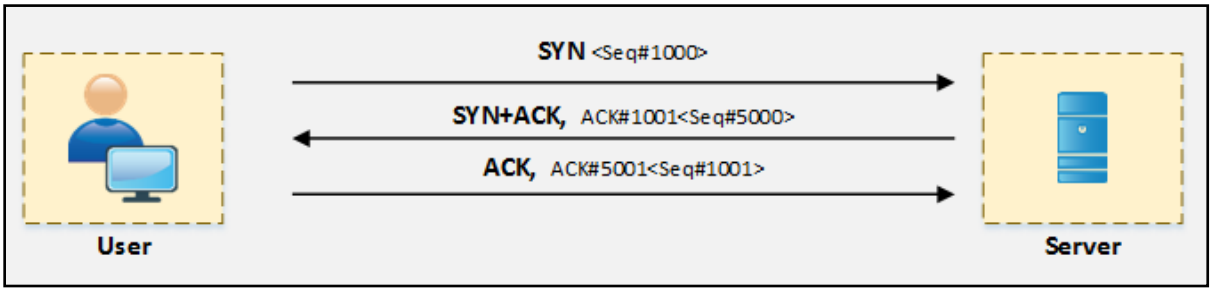


Figure 11-06 3-way Handshaking

TCP/IP Hijacking

TCP/IP hijacking process is the network level attack on a TCP session in which an attacker predicts the sequence number of a packet flowing between victim and host. To

perform TCP/IP attack, the attacker must be on the same network with the victim. Usually, the attacker uses sniffing tools to capture the packets and extract the sequence number. By injecting the spoofed packet session can be interrupted. Communication from the legitimate user can be disrupted by a Denial-of-Service attack or Reset connection.

Source Routing

Source routing is a technique of sending the packet via selected route. In session hijacking, this technique is used to attempt IP spoofing as a legitimate host with the help of Source routing to direct the traffic through the path identical to the victim's path.

RST Hijacking

RST hijacking is the process of sending Reset (RST) packet to the victim with the spoofed source address. Acknowledgment number used in this Reset packet is also predicted. When the victim receives this packet, it could not identify that the packet is spoofed believing the actual source has sent the packet resulting in resetting the connection. RST packet can be crafted using packet crafting tools.

Blind Hijacking

Blind Hijacking is the technique in which attacker is not able to capture the return traffic. In Blind hijacking, attacker captures the packet coming from victim destined towards the server, inject malicious packet and forward to the target server.

Forged ICMP and ARP Spoofing

A man-in-the-middle attack can also be performed by using Forged ICMP packet and ARP spoofing techniques. Forged ICMP packets such as Destination unavailable or high latency message are sent to fool the victim.

UDP Hijacking

UDP Session Hijacking process is quite simpler than TCP session hijacking. Since the UDP is a connectionless protocol, it does not require any sequence packet between requesting client and host. UDP session hijacking is all about sending the response packet before a destination server responds. There are several techniques to intercept the coming traffic from the destination server

Countermeasures

Session Hijacking Countermeasures

Mitigation of Session Hijacking attacks includes several detection techniques and countermeasures that can be implemented including manual and automated processes. Deployment of Defense-in-depth technology, Network monitoring devices such as

Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) are categorized as automated detection process. There are several Packet sniffing tools available which can be used for manual detection.

Furthermore, encrypted session and communication using Secure Shell (SSH), using HTTPS instead of HTTP, using Random and lengthy string for Session ID, session timeout, and strong authentication like Kerberos can be helpful to prevent and mitigate session hijacking. Using IPsec and SSL can provide stronger protection against hijacking.

IPSec

IPSec stands for IP security. As the name suggests, it is used for the security of general IP traffic. The power of IPsec lies in its ability to support multiple protocols and algorithms. It also incorporates new advancements in encryption and hashing protocols. The main objective of IPSec is to provide CIA (confidentiality, integrity, and authentication) for virtual networks used in current networking environments. IPSec makes sure the above objectives are in action by the time packet enters a VPN tunnel until it reaches the other end of the tunnel.

- **Confidentiality**. IPSec uses encryption protocols namely AES, DES, and 3DES for providing confidentiality.
- **Integrity.** IPSec uses hashing protocols (MD5 and SHA) for providing integrity. Hashed Message Authentication (HMAC) can also be used for checking the data integrity.
- **Authentication algorithms.** RSA digital signatures and pre-shared keys (PSK) are two methods used for authentication purposes.

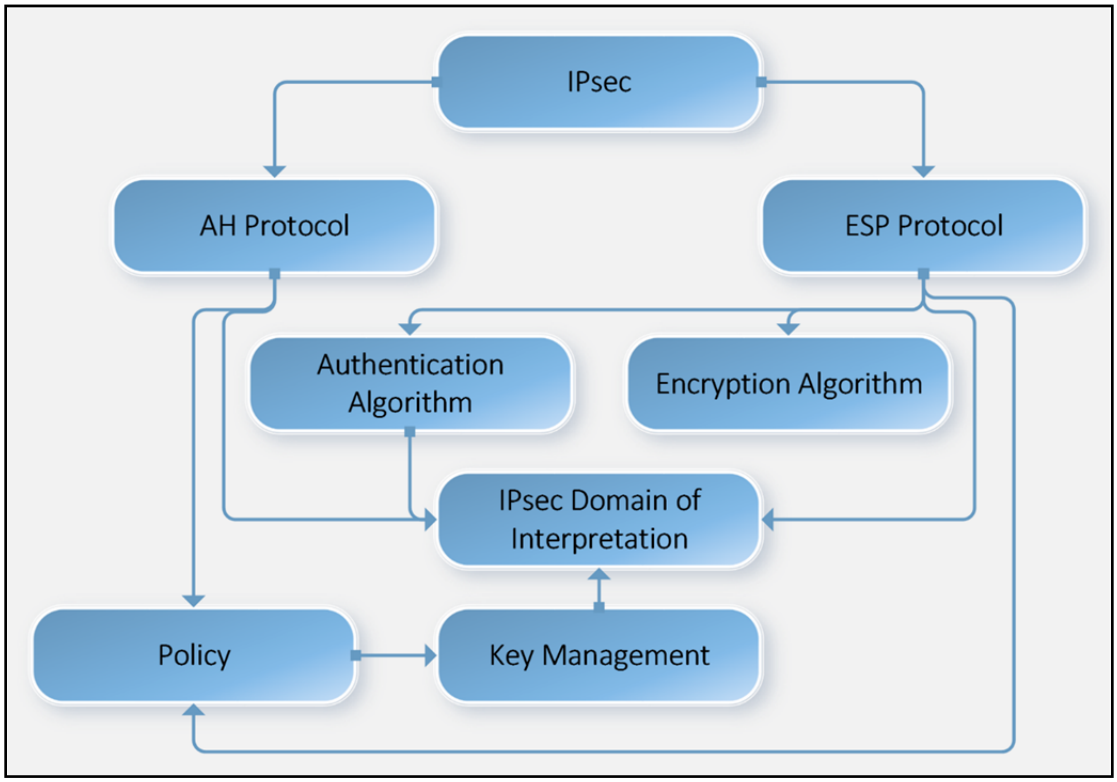


Figure 11-07 IPsec Architecture

Components of IPsec

Components of IPsec includes: -

- Components of IPsec
- IPsec Drivers
- Internet Key Exchange (IKE)
- Internet Security Association Key Management Protocol
- Oakley
- IPsec Policy Agent

Modes of IPsec

There are two working modes of IPSec namely *tunnel* and *transport mode*. Each has its features and implementation procedure.

IPSec Tunnel Mode

Being the default mode set in Cisco devices, tunnel mode protects the entire IP packet from originating device. It means for every original packet; another packet is generated with new IP header and send over the untrusted network to the VPN peer located on another end of the logical connection. Tunnel mode is commonly used in case of Site-to-

Site VPN where two secure IPSec gateways are connected over public internet using IPSec VPN connection. Consider the following diagram:

This shows IPSec Tunnel Mode with ESP header:

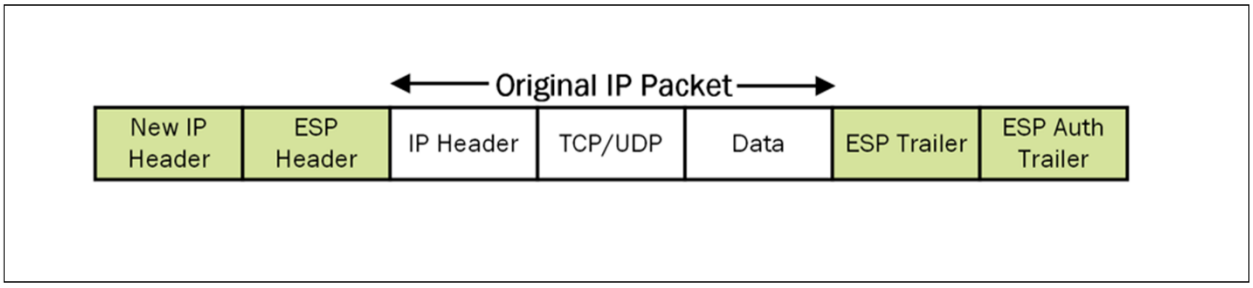


Figure 11-08 IPsec Tunnel Mode with ESP header

Similarly, when AH is used; new IP Packet format will be:

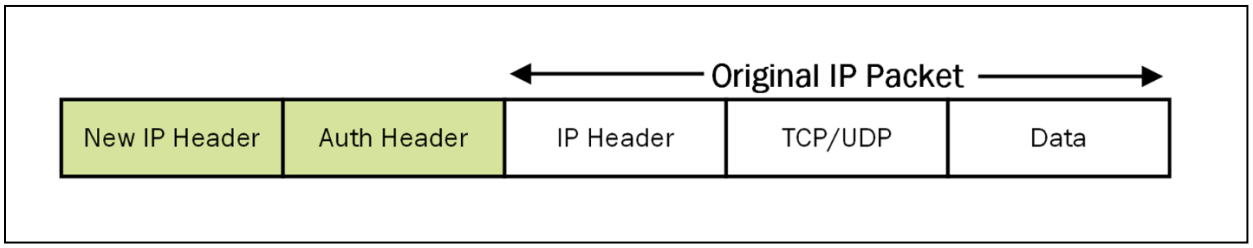


Figure 11-09 IPsec Tunnel Mode with AH header

IPsec Transport Mode

In transport mode, IPsec VPN secures the data field or payload of originating IP traffic by using encryption, hashing or both. New IPsec headers encapsulate only payload field while the original IP headers remain unchanged. Tunnel mode is used when original IP packets are the source and destination address of secure IPsec peers. For example, securing the management traffic of router is a perfect example of IPsec VPN implementation using transport mode. From a configuration point of view, both tunnel and transport modes are defined in the configuration of *transform set*. It will be covered in the Lab scenario of this section.

This diagram shows IPsec Transport Mode with ESP header:

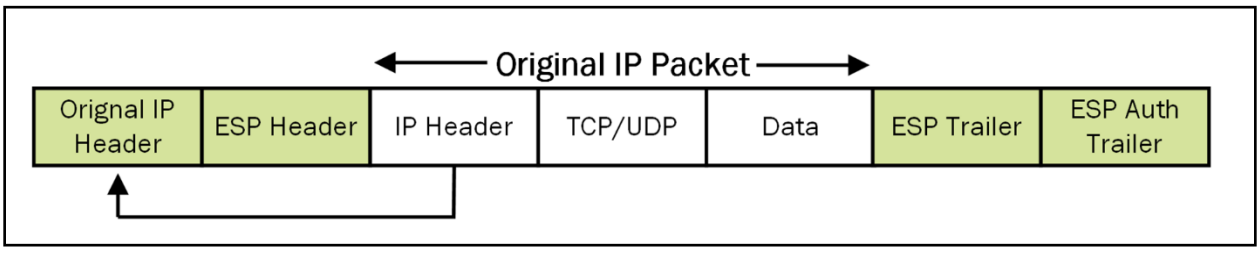


Figure 11-10 IPsec Transport Mode with ESP header

Similarly, in case of AH:

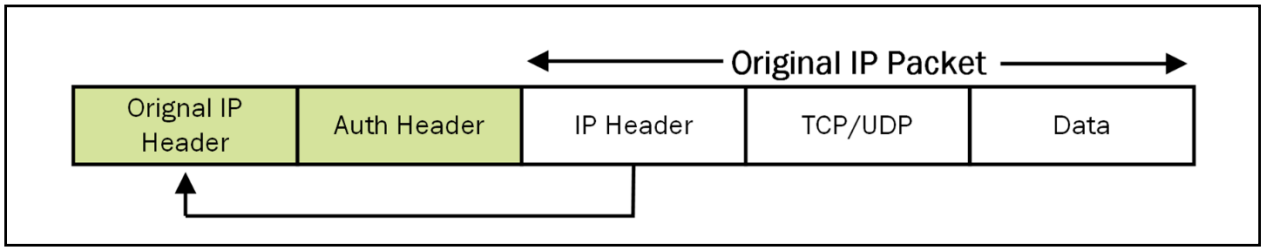
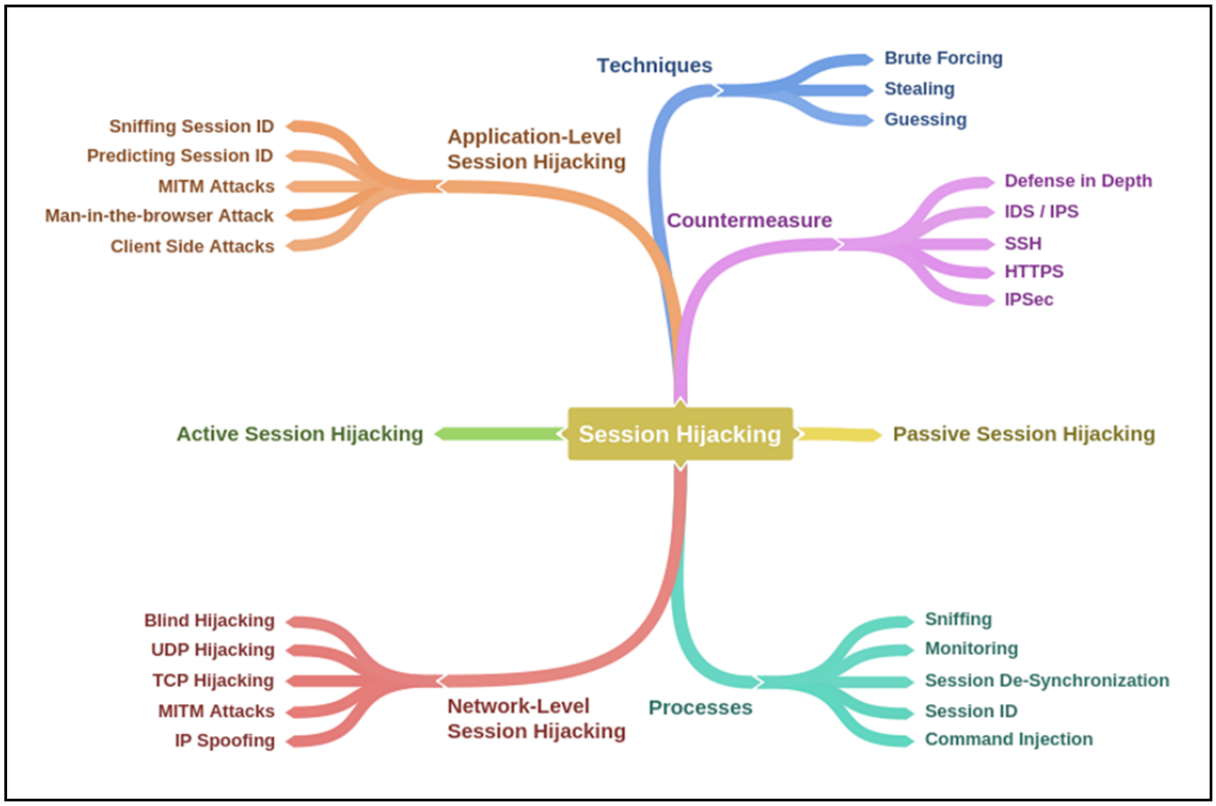


Figure 11-11 IPsec Transport Mode with AH header

Mind Map


Techniques
Brute Forcing
Stealing
Guessing
Sniffing Session ID
Predicting Session ID
MITM Attacks
Man-in-the-browser Attack
Client Side Attacks
Application-Level
Session Hijacking
Countermeasure
Defense in Depth
IDS / IPS
SSH
HTTPS
IPSec
Active Session Hijacking
Session Hijacking
Passive Session Hijacking
Blind Hijacking
UDP Hijacking
TCP Hijacking
MITM Attacks
IP Spoofing
Network-Level
Session Hijacking
Processes
Sniffing
Monitoring
Session De-Synchronization
Session ID
Command Injection