Module 13
Session Hijacking

Ansh Bhawnani

Basics Concepts

Module 13

1. Introduction

Module 13

Basics Concepts

- Session hijacking refers to an attack where an attacker takes over a valid TCP communication session between two computers.
- Since most authentication only occurs at the start of a TCP session, this allows the attacker to gain access to a machine.
- Attackers can sniff all the traffic from the established TCP sessions and perform identity theft, information theft, fraud, etc.
- The attacker steals a valid session ID and use it to authenticate himself with the server.

2. Why Session Hijacking is Successful?

Basics Concepts

- No account lockout for invalid session IDs.
- Weak session ID generation algorithm or small session IDs.
- Insecure handling of session IDs.
  - DNS poisoning, XSS, exploiting a bug in browser
- Indefinite session expiration time.
- Most computers using TCP/IP are vulnerable.
- Most countermeasures do not work unless you use encryption.

3. Session Hijacking Process

Basics Concepts

- **Stealing**: The attacker uses different techniques to steal session IDs.
  - Some of the techniques used to steal session IDs:
    - Using the HTTP referrer header.
    - Sniffing the network traffic.
    - Using the cross-site-scripting attacks.
    - Sending Trojans on client machines.

Basics Concepts

- **Guessing**: The attacker tries to guess the session IDs by observing variable parts of the session IDs.
  - http://www.hacksite.com/view/VW48266762824302
  - http://www.hacksite.com/view/VW48266762826502
  - http://www.hacksite.com/view/VW48266762828902
- **Brute Forcing**: The attacker attempts different IDs until he succeeds.
  - Using brute force attacks, an attacker tries to guess a session ID until he finds the correct session ID.

Basics Concepts

Stealing Session IDs:

- Using a "referrer attack," an attacker tries to lure a user to click on a link to malicious site (say www.hacksite.com)
- For example, GET /index.html HTTP/1.0 Host: www.hacksite.com Referrer: www.webmail.com/viewmsg.asp?msgid=689645&SID=2556X54VA75
- The browser directs the referrer URL that contains the user's session ID to the attacker's site (www.hacksite.com), and now the attacker possesses the user's session ID.

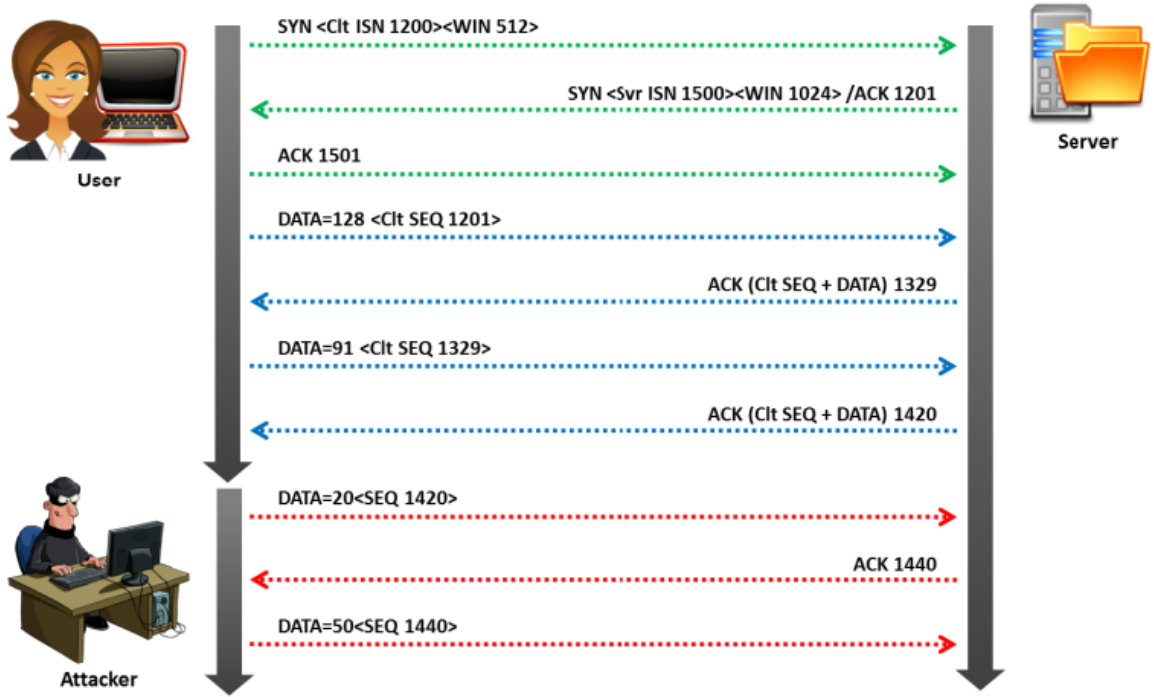
Basics Concepts

- **Note**: Session ID brute forcing attack is known as session prediction attack if the predicted range of values for a session ID is very small.
- **Command Injection**: Start injecting packets to the target server.
- **Session ID prediction**: Take over the session.
- **Session Desynchronization**: Break the connection to the victim's machine.
- **Monitor**: Monitor the flow of packets and predict the sequence number.
- **Sniff**: Place yourself between the victim and the target (you must be able to sniff the network).

4. Packet Analysis of a Local Session Hijack

Module 13

Basics Concepts



Basics Concepts

- According to the diagram, the next expected sequence number would be 1420. If you can transmit that packet sequence number before the user does, you can desynchronize the connection between the user and the server.
- After establishing the connection between the attacker and the server, though the user sends the data with the correct sequence number, the server drops the data considering it as a resent packet.

5. Types of Session Hijacking

Basics Concepts

- **Active Attack**: In an active attack, an attacker finds an active session and takes over.
- **Passive Attack**: With a passive attack, an attacker hijacks a session but sits back and watches and records all the traffic that is being sent forth.

6. Session Hijacking in OSI Model

Module 13

Basics Concepts

- **Network Level Hijacking**: Network level hijacking can be defined as the interception of the packets during the transmission between the client and the server in a TCP and UDP session.
- **Application Level Hijacking**: Application level hijacking is about gaining control over the HTTP's user session by obtaining the session IDs.

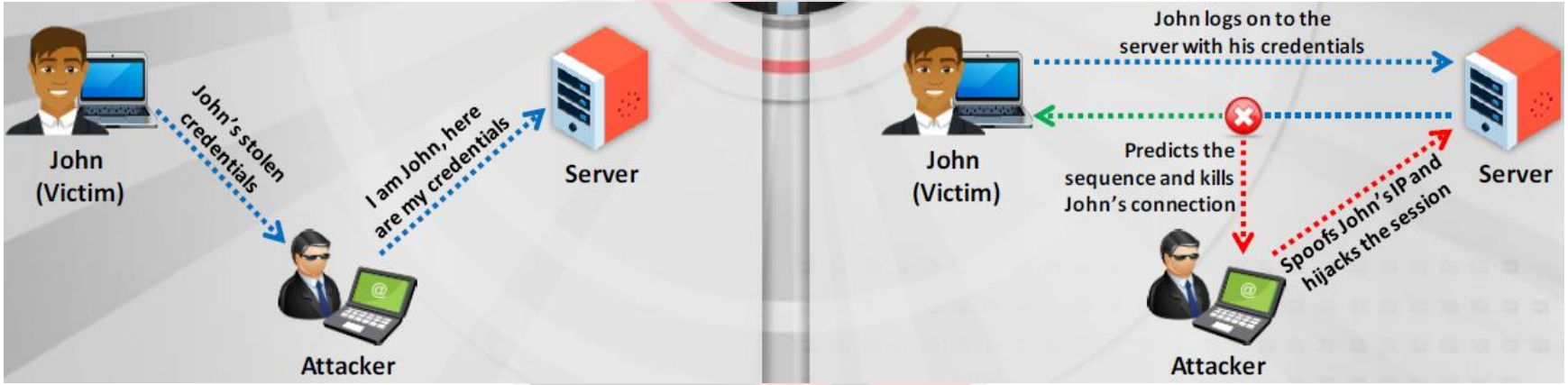
7. Spoofing vs Hijacking

Module 13

Basics Concepts

- **Spoofing** Attack:
  - Attack pretends to be another user or machine (victim) to gain access.
  - Attacker does not take over an existing active session. Instead he initiates a new session using the victim's stolen credentials.
- **Hijacking**:
  - Session hijacking is the process of taking over an existing active session.
  - Attacker relies on the legitimate user to make a connection and authenticate.

Basics Concepts



Application Level Session Hijacking

Module 13

Application Level Session Hijacking

- A session token is stolen or valid session token is predicted to gain unauthorized access to the web server.
- A session token can be compromised in various ways:
  - Session sniffing, Session replay attack, Session fixation
  - Predictable session token
  - Man-in-the-middle attack
  - Man-in-the-browser attack
  - Cross-site script attack
  - Cross-site request forgery attack (CSRF)

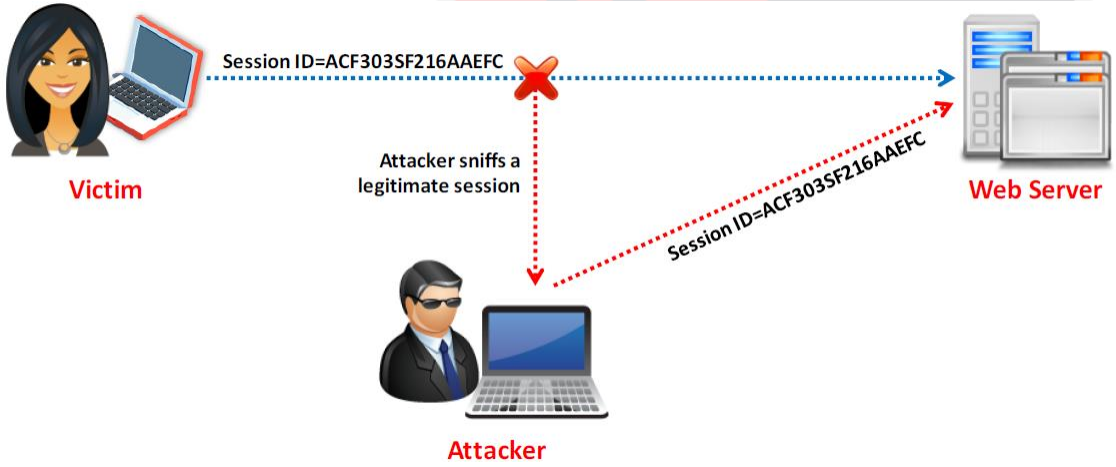
1. Sniffing

Module 13

Application Level Session Hijacking

Compromising Sessions IDs using Sniffing

- Attacker uses a sniffer to capture a valid session token or session ID.
- Attacker then uses the valid token session to gain unauthorized access to the web server.



2. Predicting

Module 13

Application Level Session Hijacking

Compromising Session IDs by Predicting Session Token

- Attacker can predict session IDs generated by weak algorithms and impersonate a web site user.
- Attackers perform analysis of variable section of session IDs to determine the existence of a pattern.
- The analysis is performed manually or by using various cryptanalytic tools.
- Attackers collect a high number of simultaneous session IDs in order to gather samples in the same time window and keep the variable constant.

Application Level Session Hijacking

How to Predict a Session Token

- Most of the web servers use custom algorithms or a predefined pattern to generate sessions IDs.
- Attacker guess the unique session value or deduce the session ID to hijack the sessions.
- **Captures**: Attacker captures several session IDs and analyzes the pattern.
  - http://www.juggyboy.com/view/JBEX250220141528 20
  - http://www.juggyboy.com/view/JBEX250220141530 20
  - http://www.juggyboy.com/view/JBEX250220141600 20
  - http://www.juggyboy.com/view/JBEX250220141640 20

Application Level Session Hijacking

- **Predicts**: At 16:25:55 on Feb-25, 2014, the attacker can successfully predict the session ID to be http://www.juggyboy.com/view/JBEX25022014162555
  - **JBEX**: Constant
  - **25022014**: Date
  - **162555**: Time

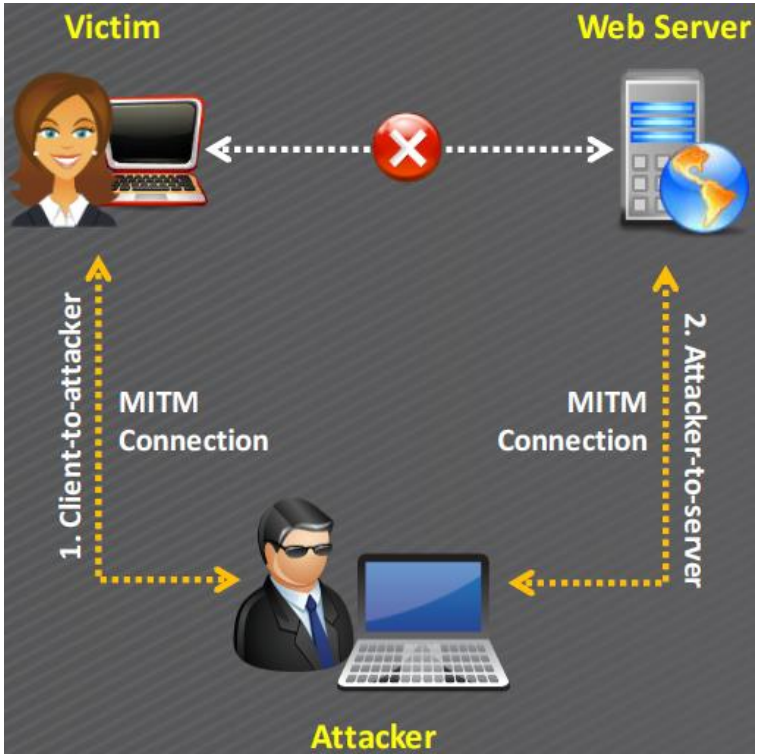
3. Man in the Middle

Module 13

Application Level Session Hijacking

- The man-in-the-middle attack is used to intrude into an existing connection between systems and to intercept messages being exchanged.
- Attackers use different techniques and split the TCP connection into two connections.
  - Client-to-attacker connection
  - Attacker-to-server connection
- After the successful interception of TCP connection, an attacker can read, modify, and insert fraudulent data into the intercepted communication.
- In the case of an http transaction, the TCP connection between the client and the server becomes the target.

Application Level Session Hijacking



4. Man in the Browser

Module 13

Application Level Session Hijacking

- Man-in-the-browser attack uses a Trojan Horse to intercept the calls between the browser and its security mechanisms or libraries.
- It works with an already installed Trojan horse and acts between the browser and its security mechanisms.
- Its main objective is to cause financial deceptions by manipulating transactions of Internet Banking systems.

Application Level Session Hijacking

Steps to Perform Man-in-the-Browser Attack

- The Trojan first infects the computer's software (OS or application).
- The Trojan installs malicious code (extension files) and saves it into the browser configuration.
- After the user restarts the browser, the malicious code in the form of extension files is loaded.
- The extension files register a handler for every visit to the webpage.
- When the page is loaded, the extension uses the URL and matches it with a list of known sites targeted for attack.

Application Level Session Hijacking

- The user logs in securely to the website.
- It registers a button event handler when a specific page load is detected for a specific pattern and compares it with its targeted list.
- When the user clicks on the button, the extension uses DOM interface and extracts all the data from all form fields and modifies the values.
- The browser sends the form and modified values to the server.

Application Level Session Hijacking

- The server receives the modified values but cannot distinguish between the original and the modified values.
- After the server performs the transaction, a receipt is generated.
- Now, the browser receives the receipt for the modified transaction.
- The browser displays the receipt with the original details.
- The user thinks that the original transaction was received by the server without any interceptions.

5. Client Side Attacks

Module 13

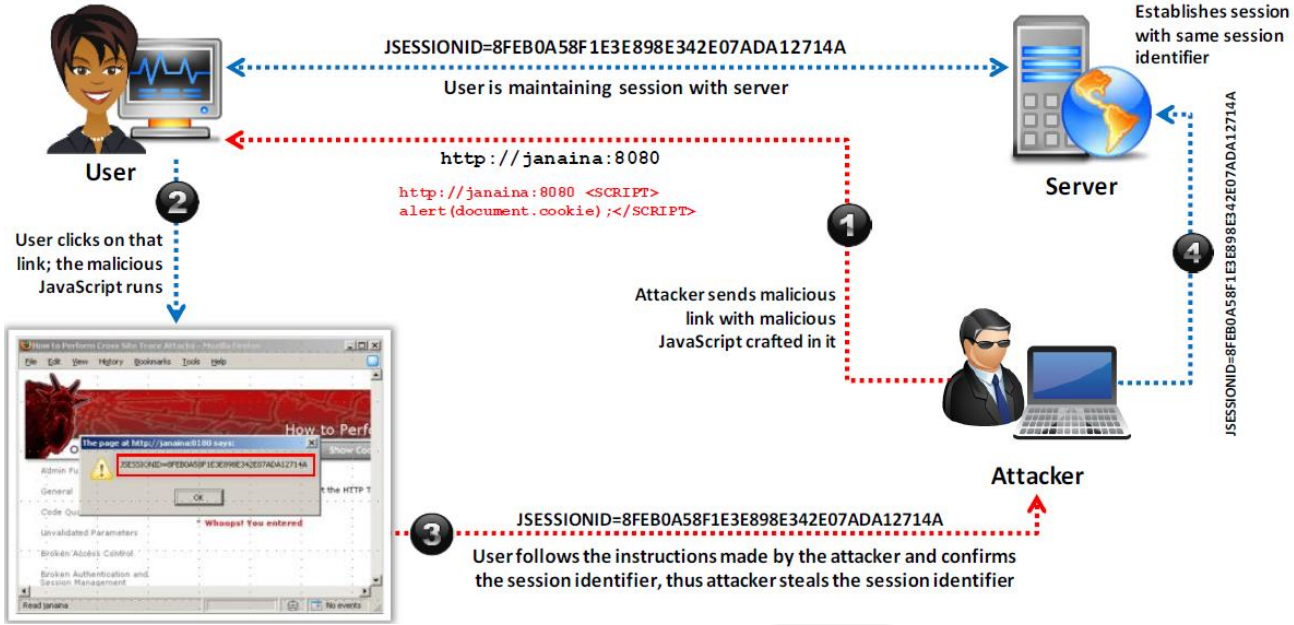
Application Level Session Hijacking

Compromising Session IDs Using Client-side Attacks

- **Cross-Site Scripting (XSS)**: XSS enables attackers to inject malicious client side scripts into the web pages viewed by other users.
- **Malicious JavaScript Codes**: A malicious script can be embedded in a web page that does not generate any warning but it captures session tokens in the background and send it to the attacker.
- **Trojans**: A Trojan horse can change the proxy settings in user's browser to send all the sessions through the attackers machine.

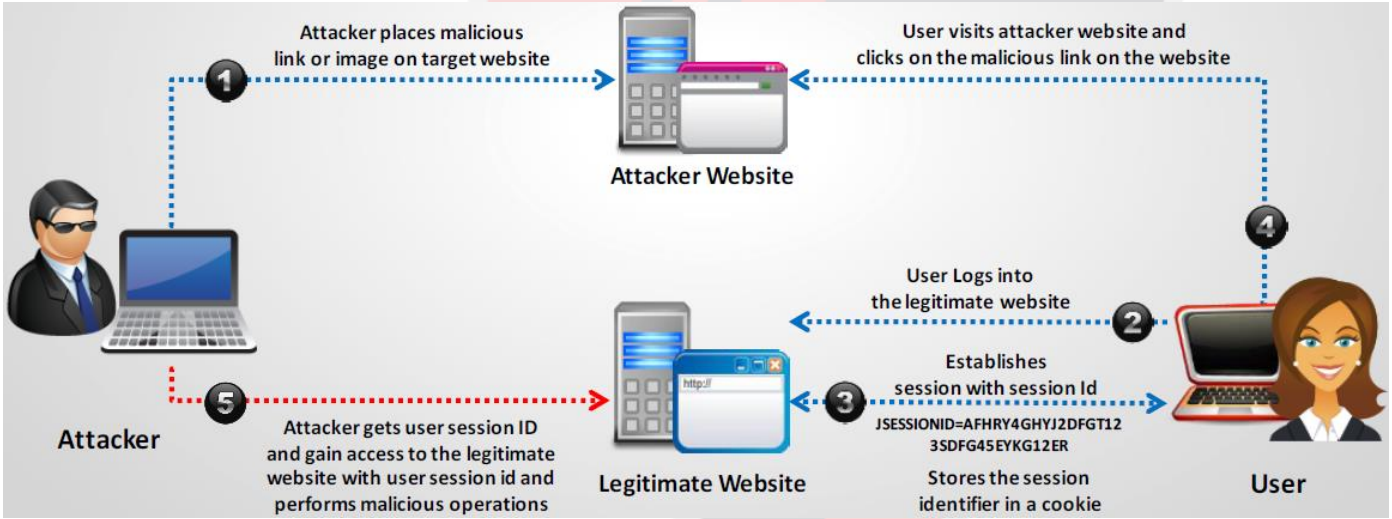
Application Level Session Hijacking

Cross-site Script Attack



Application Level Session Hijacking

Cross-site Request Forgery Attack

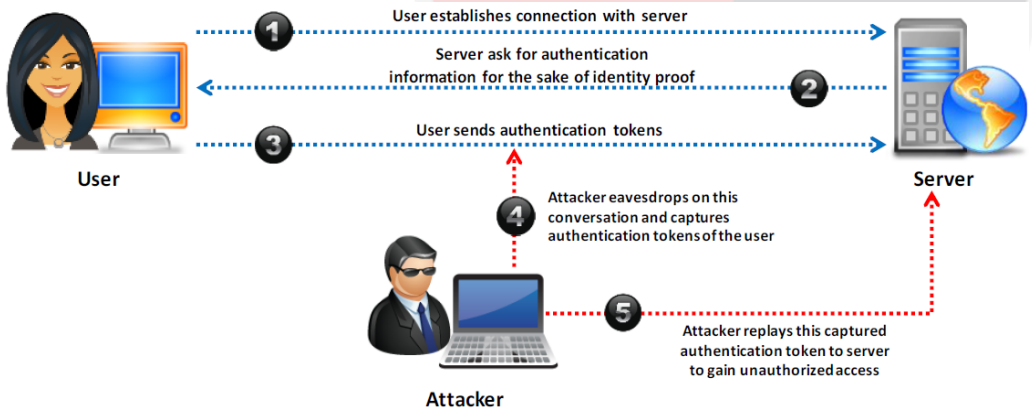


6. Replay Attacks

Module 13

Application Level Session Hijacking

- In a session replay attack, the attacker listens to the conversation between the user and the server and captures the authentication token of the user.
- Once the authentication token is captured, the attacker replays the request to the server with the captured authentication token and gains unauthorized access to the server.



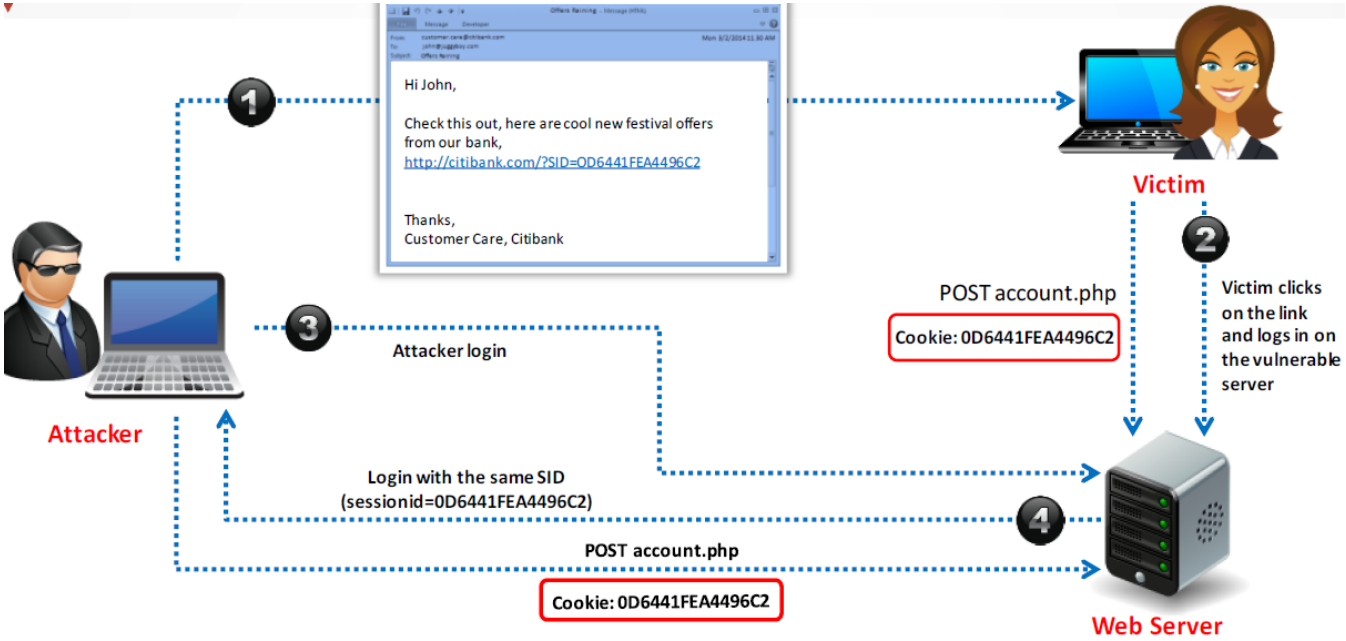
7. Session Fixation

Module 13

Application Level Session Hijacking

- The attack tries to lure a user to authenticate himself with a known session ID and then hijacks the user-validated session by the knowledge of the used session ID.
- The attacker has to provide a legitimate web application session ID and try to lure victim browser to use it.
- Several techniques to execute Session Fixation attack are:
  - Session token in the URL argument
  - Session token in a hidden form field
  - Session ID in a cookie

Application Level Session Hijacking



8. Proxy Servers

Module 13

Application Level Session Hijacking

- Attacker lure victim to click on bogus link which looks legitimate but redirect user to attacker server.
- Attacker forwards request to the legitimate server on behalf of victim and serve as a proxy for the entire transaction.
- Attacker then captures the sessions information during interaction of legitimate server and user.

Network Level Session Hijacking

Module 13

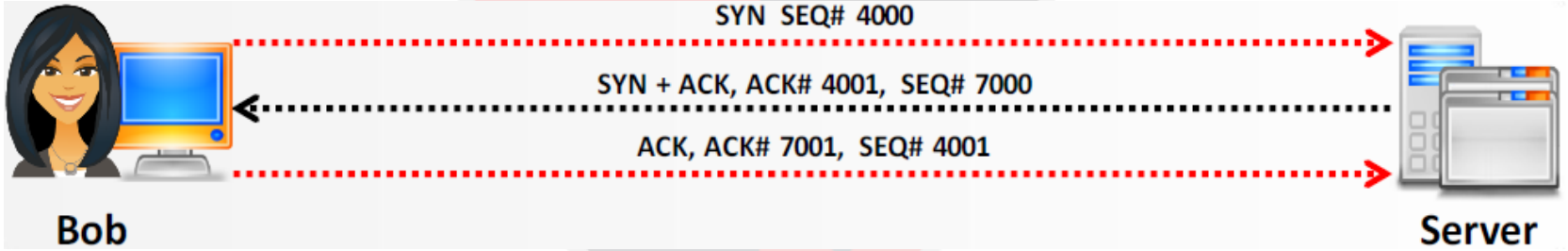
Network Level Session Hijacking

- The network-level hijacking relies on hijacking transport and Internet protocols used by web applications in the application layer.
- Attacker gathers some critical information used to attack application level.
- Network-level hijacking includes:
  - Blind Hijacking
  - TCP/IP Hijacking, UDP Hijacking
  - RST Hijacking
  - Man-in-the-Middle: Packet Sniffer
  - IP Spoofing: Source Routed Packets

Network Level Session Hijacking

The 3-Way Handshake

- If the attacker can anticipate the next sequence and ACK number that Bob will send, he/she will spoof Bob's address and start a communication with the server.



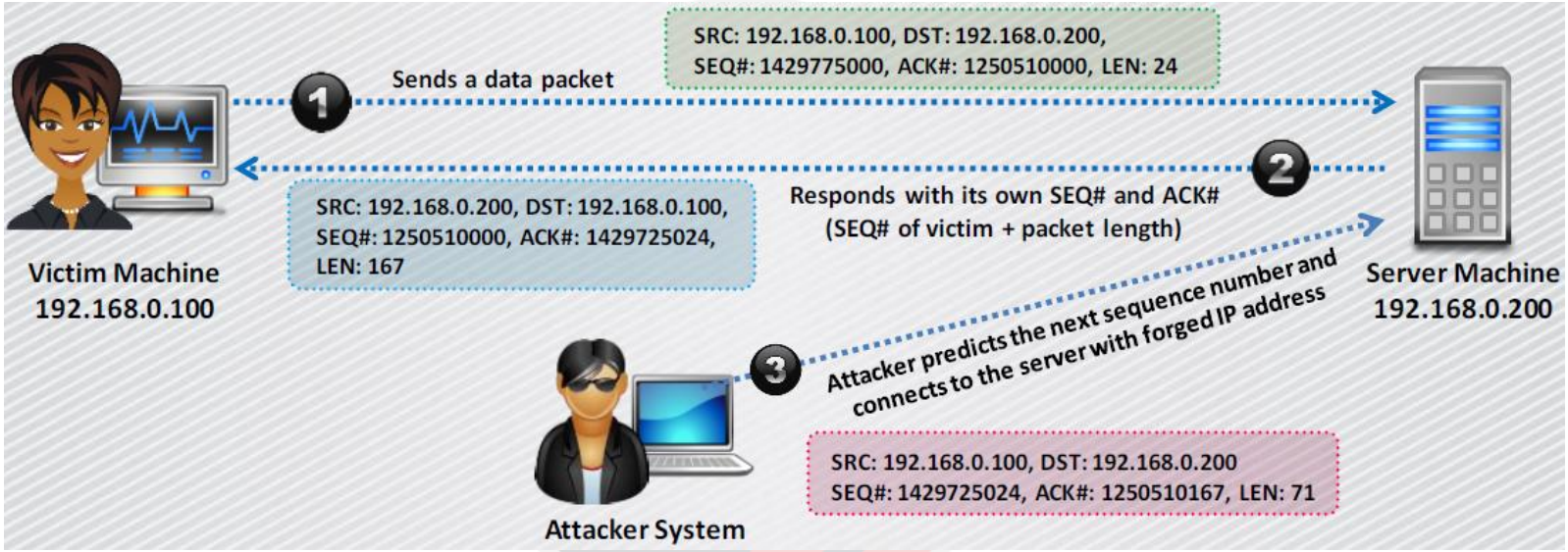
1. TCP/IP Hijacking

Module 13

Network Level Session Hijacking

- TCP/IP hijacking is a hacking technique that uses spoofed packets to take over a connection between a victim and a target machine.
- The victim's connection hangs and the attacker is then able to communicate with the host's machine as if the attacker is the victim.
- To launch a TCP/IP hijacking attack, the attacker must be on the same network as the victim.
- The target and the victim machines can be anywhere.

Network Level Session Hijacking



Network Level Session Hijacking

- The attacker sniffs the victim's connection and uses the victim's IP to send a spoofed packet with the predicted sequence number.
- The receiver processes the spoofed packet, increments the sequence number, and sends acknowledgement to the victim's IP.
- The victim machine is unaware of the spoofed packet, so it ignores the receiver machine's ACK packet and turns sequence number count off.
- Therefore, the receiver receives packets with the incorrect sequence number.

Network Level Session Hijacking

- The attacker forces the victim's connection with the receiver machine to a desynchronized state.
- The attacker tracks sequence numbers and continuously spoofs packets that comes from the victim's IP.
- The attacker continues to communicate with the receiver machine while the victim's connection hangs.

2. IP Spoofing

Network Level Session Hijacking

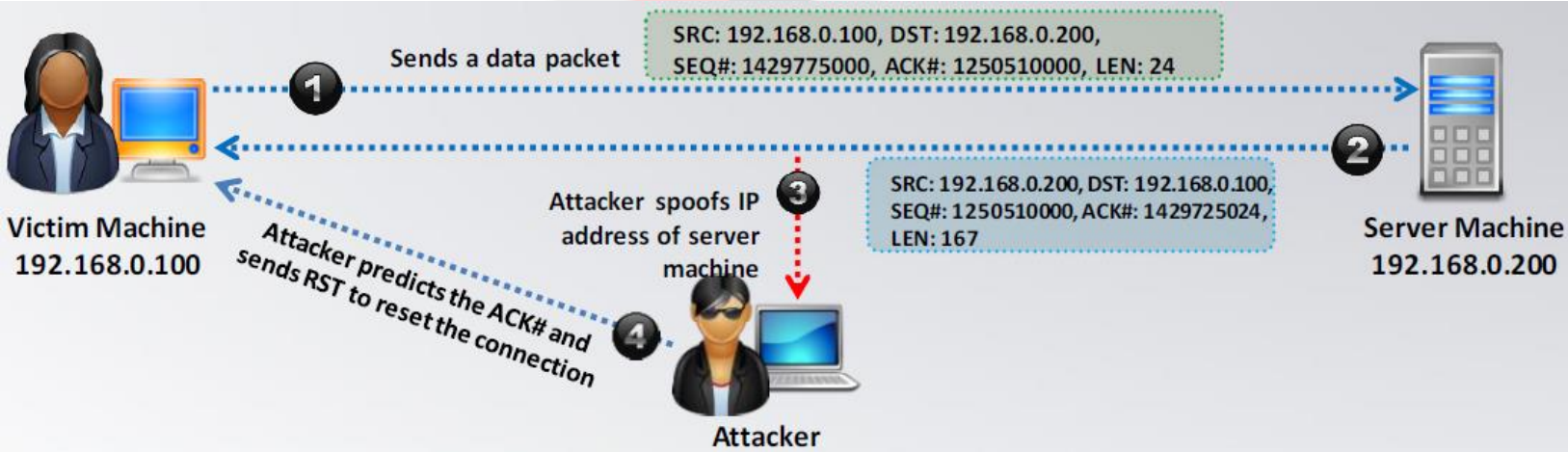
- Packet source routing technique is used for gaining unauthorized access to a computer with the help of a trusted host's IP address.
- The attackers spoofs the host's IP address so that the server managing a session with the host, accepts the packets from the attacker.
- When the session is established, the attacker injects forged packets before the host responds to the server.
- The original packet from the host is lost as the server gets the packet with a sequence number already used by the attacker.
- The packets are source-routed where the path to the destination IP can be specified by the attacker.

3. RST Hijacking

Network Level Session Hijacking

- RST hijacking involves injecting an authentic-looking reset (RST) packet using spoofed source address and predicting the acknowledgment number.
- The hacker can reset the victim's connection if it uses an accurate acknowledgement number.
- The victim believes that the source actually sent the reset packet and resets the connection.
- RST Hijacking can be carried out using a packet crafting tool such as Colasoft's Packet Builder and TCP/IP analysis tool such as tcpdump.

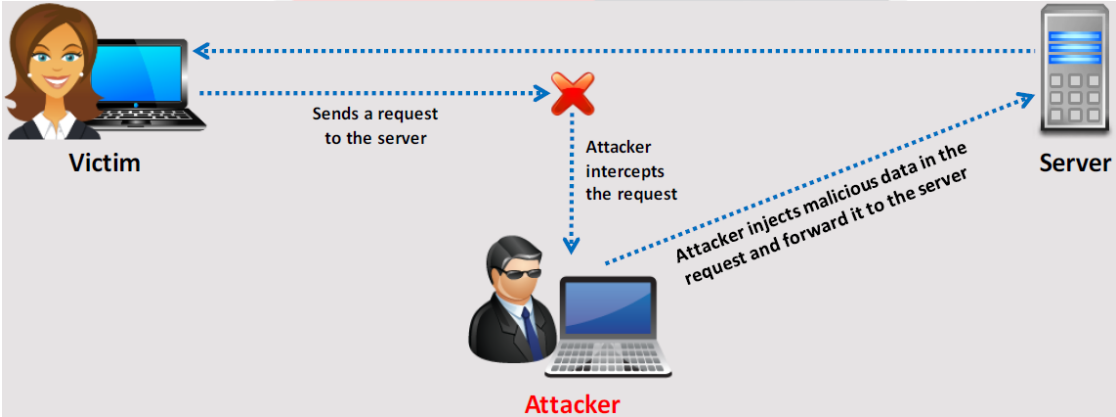
Network Level Session Hijacking



4. Blind Hijacking

Network Level Session Hijacking

- The attacker can inject the malicious data or commands into the intercepted communications in the TCP session even if the source-routing is disabled.
- The attacker can send the data or commands but has no access to see the response.



5. Forged ICMP and ARP Spoofing

Module 13

Network Level Session Hijacking

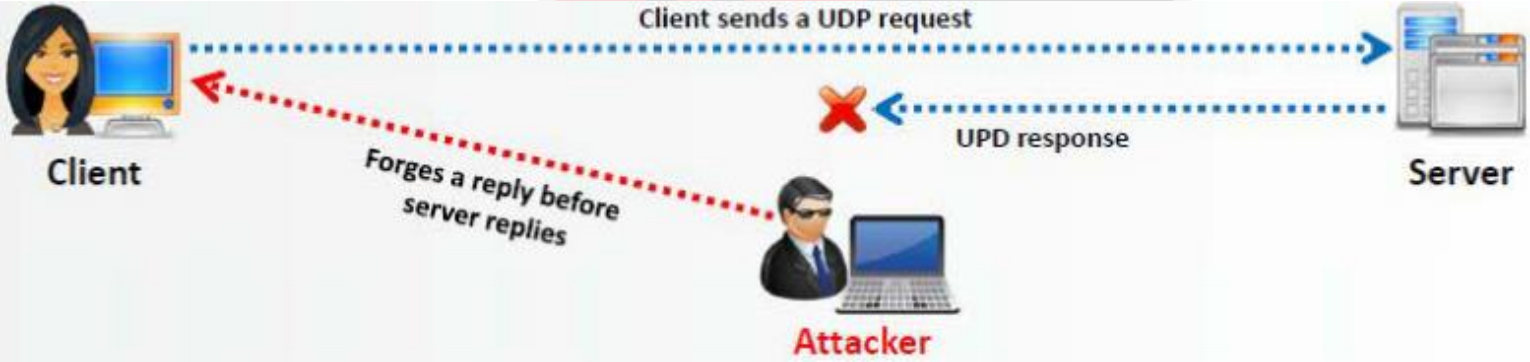
- Packet sniffer is used as an interface between the client and the server.
- ARP spoofing involves fooling the host by broadcasting the ARP request and changing its ARP tables by sending the forged ARP replies.
- The packets between the client and the server are routed through the hijacker's host by using two techniques:
  - Using **Forged Internet Control Message Protocol** (**ICMP**): It is an extension of IP to send error messages where the attacker can send messages to fool the client and the server.
  - **Using Address Resolution Protocol (ARP) Spoofing**: ARP is used to map the network layer address (IP address) to link layer addresses (MAC address).

6. UDP Hijacking

Module 13

Network Level Session Hijacking

- A network-level session hijacking where the attacker sends forged server reply to a victim's UDP request before the intended server replies to it.
- The attacker uses man-in-the-middle attack to intercept server's response to the client and sends its own forged reply.



Countermeasures

Module 13

1. Detection

Module 13

Countermeasures

- **Detection Method**
  - **Manual** Method
    - Using Packet Sniffing Software
      - Normal Telnet Session
      - Forcing an ARP Entry
  - **Automatic** Method
    - Intrusion Detection Systems (IDS)
    - Intrusion Prevention Systems (IPS)

2. Protection

Module 13

Countermeasures

- Use Secure Shell (SSH) to create a secure communication channel.
- Pass the authentication cookies over HTTPS connection.
- Implement the log-out functionality for user to end the session.
- Generate the session ID after successful login and accept sessions IDs generated by server only.
- Ensure data in transit is encrypted and implement defense-in-depth mechanism.

Countermeasures

- Use string or long random number as a session key.
- Use different user name and passwords for different accounts.
- Educate the employees and minimize remote access.
- Implement timeout() to destroy the session when expired.
- Do not transport session ID in query string.
- Use switches rather than hubs and limit incoming connections.

Countermeasures

- Ensure client-side and server-side protection software are in active state and up to date.
- Use strong authentication (like Kerberos) or peer-to-peer VPN's.
- Configure the appropriate internal and external spoof rules on gateways.
- Use IDS products or ARPwatch for monitoring ARP cache poisoning.
- Use encrypted protocols that are available at OpenSSH suite.

Countermeasures

For Web Developers:

- Create session keys with lengthy strings or random number so that it is difficult for an attacker to guess a valid session key.
- Regenerate the session ID after a successful login to prevent session fixation attack.
- Encrypt the data and session key that is transferred between the user and the web servers.
- Expire the session as soon as the user logs out.
- Prevent Eavesdropping within the network.
- Reduce the life span of a session or a cookie.

Countermeasures

For Web Users:

- Do not click on the links that are received through mails or IMs.
- Use Firewalls to prevent the malicious content from entering the network.
- Use firewall and browser settings to restrict cookies.
- Make sure that the website is certified by the certifying authorities.
- Make sure you clear history, offline content, and cookies from your browser after every confidential and sensitive transaction.
- Prefer https, a secure transmission, rather than http
- Logout from the browser instead of closing the browser.

Countermeasures

Issue	Solution	Notes
Telnet, rlogin	OpenSSH or ssh (Secure Shell)	It sends encrypted data and makes it difficult for attacker to send the correctly encrypted data if session is hijacked
FTP	sFTP	It reduces the chances of successful hijacking
HTTP	SSL (Secure Socket Layer)	It reduces the chances of successful hijacking
IP	IPSec	It prevents hijacking by securing IP communications
Any Remote Connection	VPN	Implementing encrypted VPN such as PPTP, L2PT, IPSec, etc. for remote connection prevents session hijacking
SMB (Server Message Block)	SMB signing	It improves the security of the SMB protocol and reduces the chances of session hijacking
Hub Network	Switch Network	It mitigates the risk of ARP spoofing and other session hijacking attacks

3. IPSec

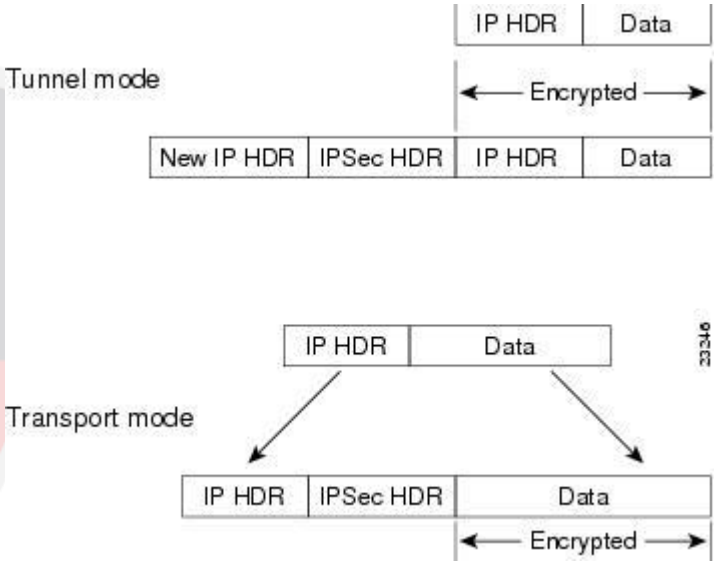
Module 13

Countermeasures

- IPSec is a protocol suite developed by the IETF for securing IP communications by authenticating and encrypting each IP packet of a communication session.
- It is deployed widely to implement virtual private networks (VPNs) and for remote user access through dial-up connection to private networks.
- **Benefits**:
  - Network-level peer authentication
  - Data origin authentication
  - Data integrity
  - Data confidentiality (encryption)
  - Replay protection

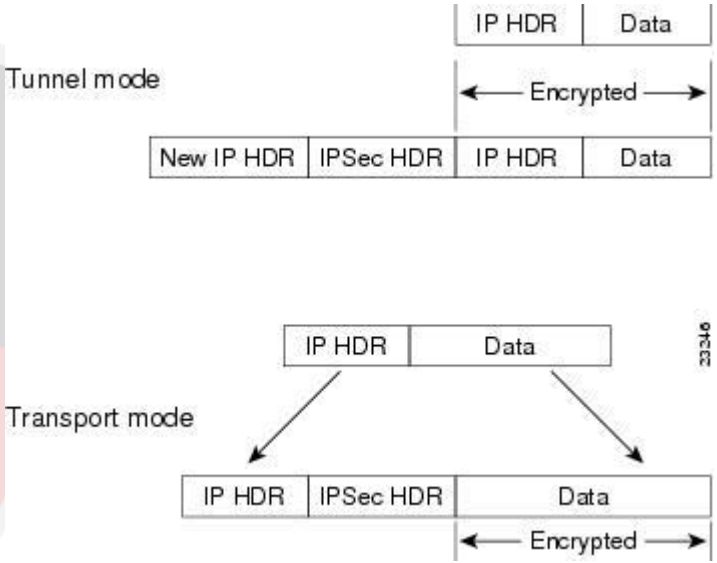
Countermeasures

- **Transport Mode**:
  - Authenticates two connected computers
  - Has an option to encrypt data transfer
  - Compatible with NAT
  - Encrypts only IP data



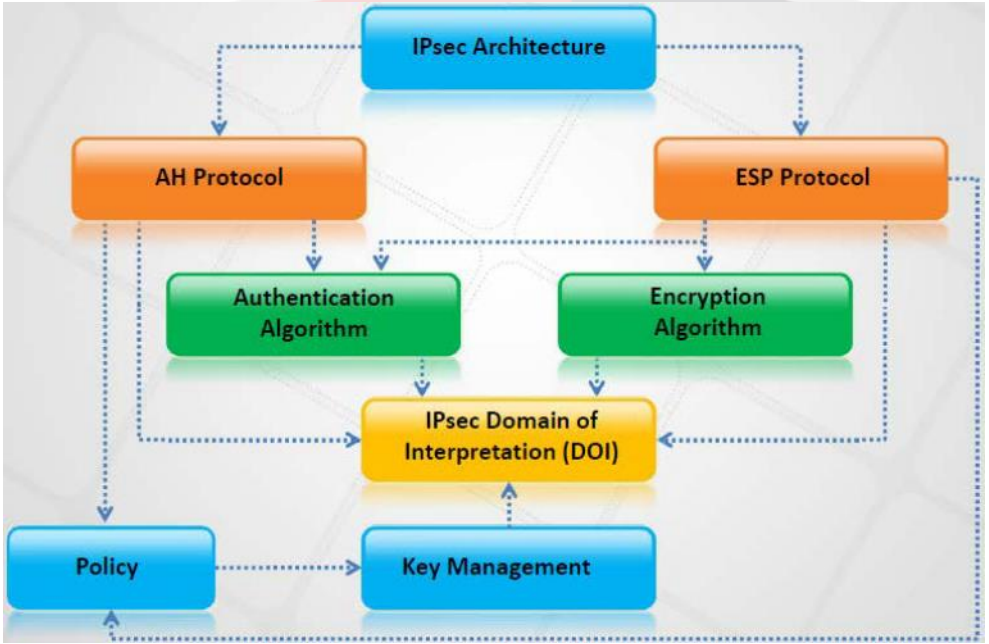
Countermeasures

- **Tunnel Mode**:
  - Encapsulates packets being transferred
  - Has an option to encrypt data transfer
  - Encrypts TCP Data + Header
  - Not compatible with NAT



Countermeasures

IPSec Architecture



Countermeasures

- IPsec uses two different security services for authentication and confidentiality:
  - **Authentication Header** (AH): Provide data authentication of the sender.
  - **Encapsulation Security Payload** (ESP): Provides both data authentication and encryption (confidentiality) of the sender.

Countermeasures

Components of IPsec (?)

- **IPsec driver**: A software, that performs protocol-level functions that are required to encrypt and decrypt the packets.
- **Internet Key Exchange (IKE)**: IPsec protocol that produces security keys for IPsec and other protocols.
- **Internet Security Association Key Management Protocol**: Software that allows two computers to communicate by encrypting the data that is exchanged between them.

Countermeasures

Components of IPsec (?)

- **Oakley**: A protocol, which uses the Diffie-Heilman algorithm to create master key, and a key that is specific to each session in IPsec data transfer.
- **IPsec Policy Agent**: A service of the Windows 2000, collects IPsec policy settings from the active directory and sets the configuration to the system at start up.

HACKING

Is an art, practised through a creative mind.