
CEH.VN
Session Hijacking
Module 10
Unmask the Invisible Hacker.
C|EH
Certified Ethical Hacker

Module Objectives

- Understanding Session Hijacking Concepts
- Understanding Application Level Session Hijacking
- Understanding Network Level Session Hijacking
- Session Hijacking Tools
- Session Hijacking Countermeasures
- Overview of Session Hijacking Penetration Testing



Module Flow

1. Session Hijacking Concepts
2. Application Level Session Hijacking
3. Network Level Session Hijacking
4. Session Hijacking Tools
5. Countermeasures
6. Penetration Testing

What is Session Hijacking?

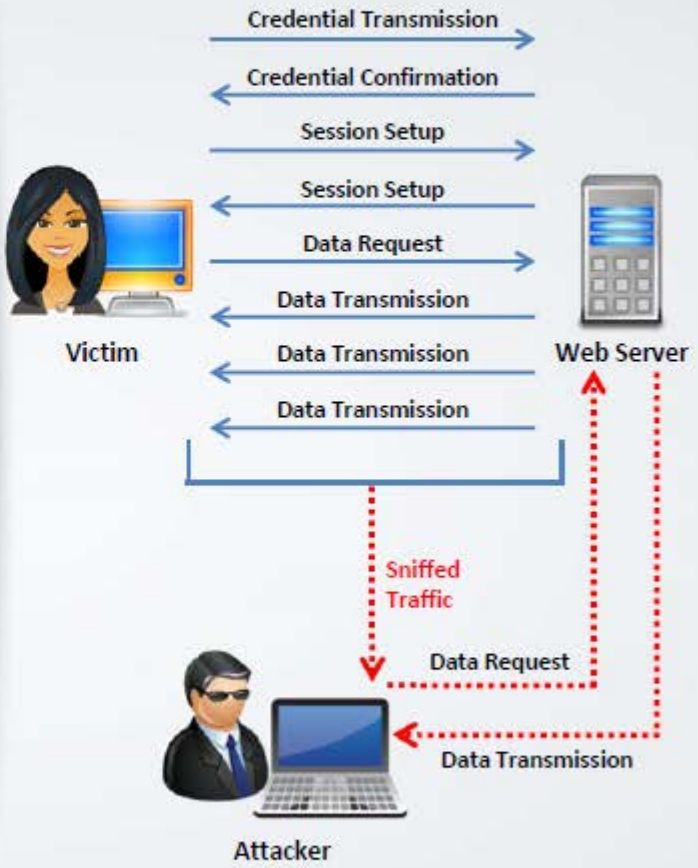
C|EH Certified Ethical Hacker

01 Session hijacking refers to an attack where an attacker takes over a **valid TCP communication session** between two computers

02 Since most **authentication only occurs at the start of a TCP session**, this allows the attacker to gain access to a machine

03 Attackers can sniff all the traffic from the established TCP sessions and perform **identity theft**, **information theft**, **fraud**, etc.

04 The attacker steals a valid session ID and use it to **authenticate himself with the server**



Why Session Hijacking is Successful?

C|EH Certified Ethical Hacker

No account lockout for invalid session IDs

Indefinite session expiration time

Weak session ID generation algorithm or small session IDs

Most computers using TCP/IP are vulnerable

Insecure handling of session IDs

Most countermeasures do not work unless you use encryption

Session Hijacking Process

C|EH Certified Ethical Hacker

Stealing

1. The attacker uses different techniques to steal session IDs

Some of the techniques used to steal session IDs:

1. Using the HTTP referrer header
2. Sniffing the network traffic
3. Using the cross-site-scripting attacks
4. Sending Trojans on client machines

Guessing

2. 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

3. 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

Stealing Session IDs

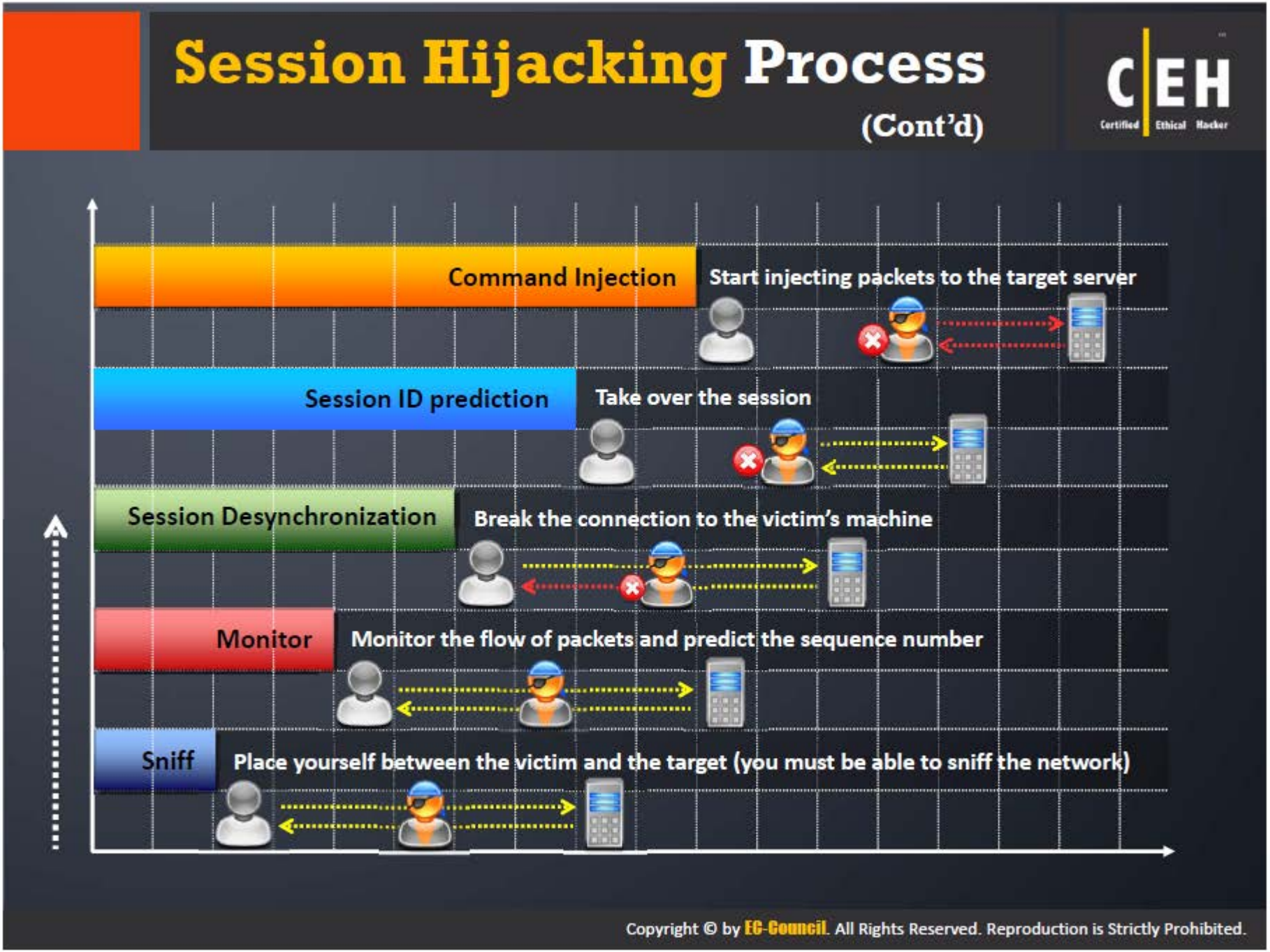
Using a "**referrer attack**," an attacker tries to lure a user to click on a link to malicious site (say www.hacksite.com)

For example, GET /index.html HTTP/1.0 Host: www.hacksite.com Referrer: www.webmail.com/viewmsg.asp?msgid=689645&SID=2556X54VA75

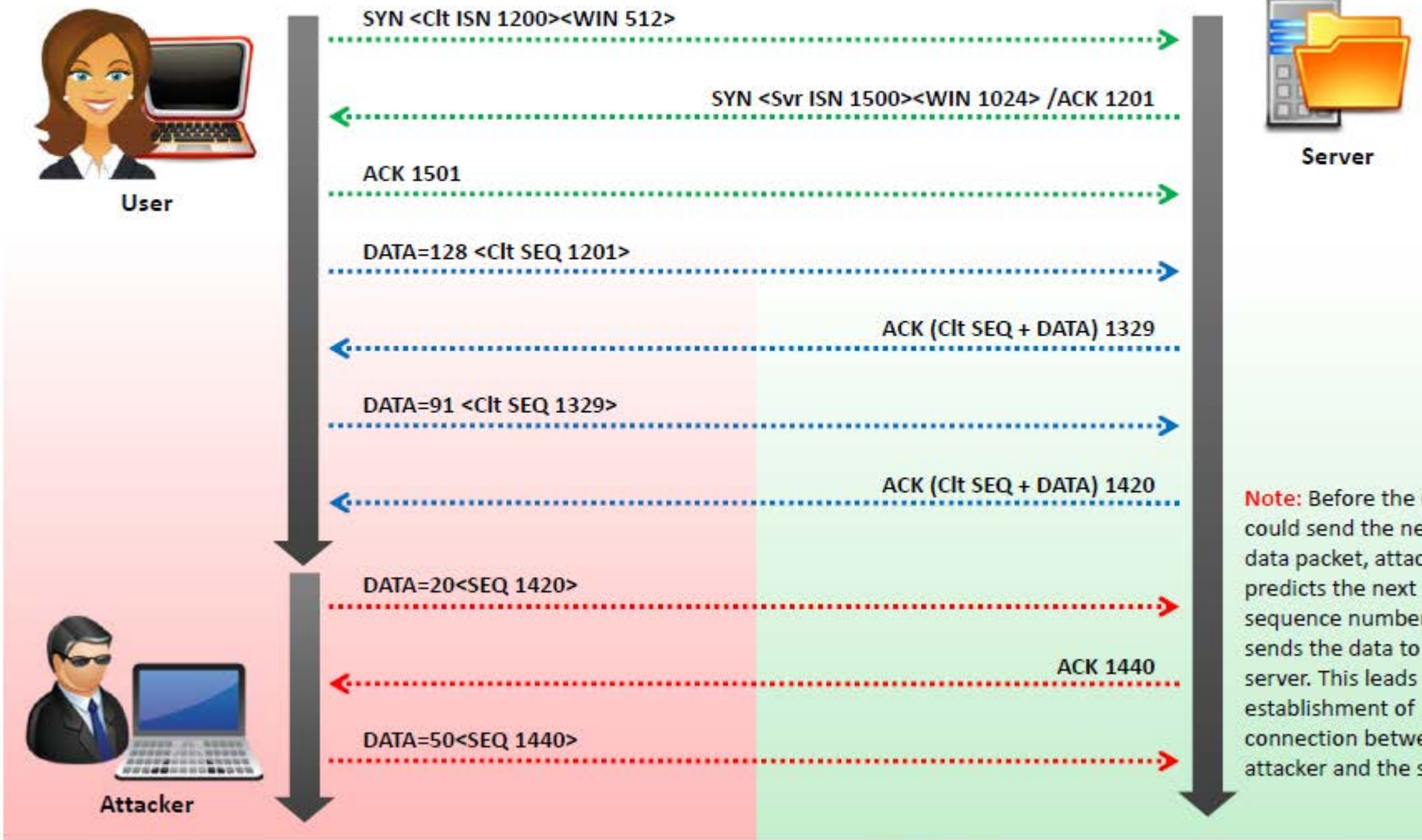
The browser directs the **referrer URL** that contains the user's session ID to the attacker's site (www.hacksite.com), and now the attacker possesses the user's session ID

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

Session Hijacking Process (Cont'd)



Packet Analysis of a Local Session Hijack



Types of Session Hijacking

C|EH Certified Ethical Hacker

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

Session Hijacking in OSI Model

C|EH Certified Ethical Hacker

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

Spoofing vs. Hijacking

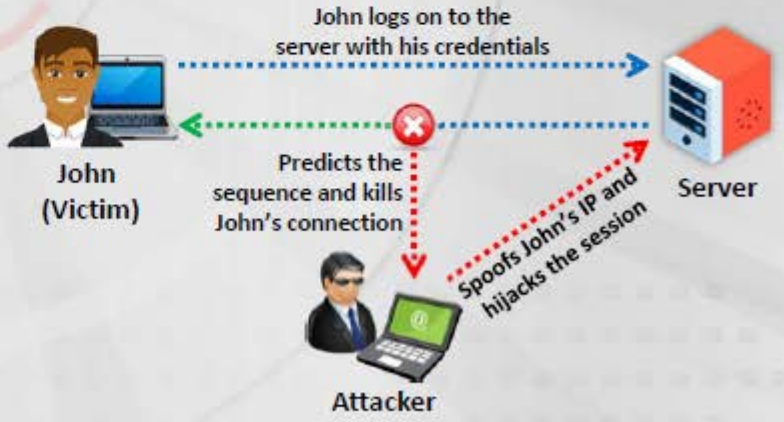
C|EH Certified Ethical Hacker

Spoofing Attack

- Attacker **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



Module Flow

1. Session Hijacking Concepts
2. Application Level Session Hijacking
3. Network Level Session Hijacking
4. Session Hijacking Tools
5. Countermeasures
6. Penetration Testing

Application Level Session Hijacking

In a session hijacking attack, a session token is stolen or a valid session token is predicted to gain unauthorized access to the web server

A session token can be compromised in various ways

1. Session sniffing
2. Predictable session token
3. Man-in-the-middle attack
4. Man-in-the-browser attack
5. Cross-site script attack
6. Cross-site request forgery attack
7. Session replay attack
8. Session fixation

Compromising Session IDs by Predicting Session Token

01 Attackers can **predict session IDs** generated by weak algorithms and **impersonate a web site user**

02 Attackers perform analysis of variable section of session IDs to **determine the existence of a pattern**

03 The analysis is performed **manually** or by **using various cryptanalytic tools**

04 Attackers **collect a high number of simultaneous session IDs** in order to gather samples in the same time window and keep the variable constant

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/JBEX21022014152820
http://www.juggyboy.com/view/JBEX21022014153020
http://www.juggyboy.com/view/JBEX21022014160020
http://www.juggyboy.com/view/JBEX21022014164020
                            Constant  Date  Time
```

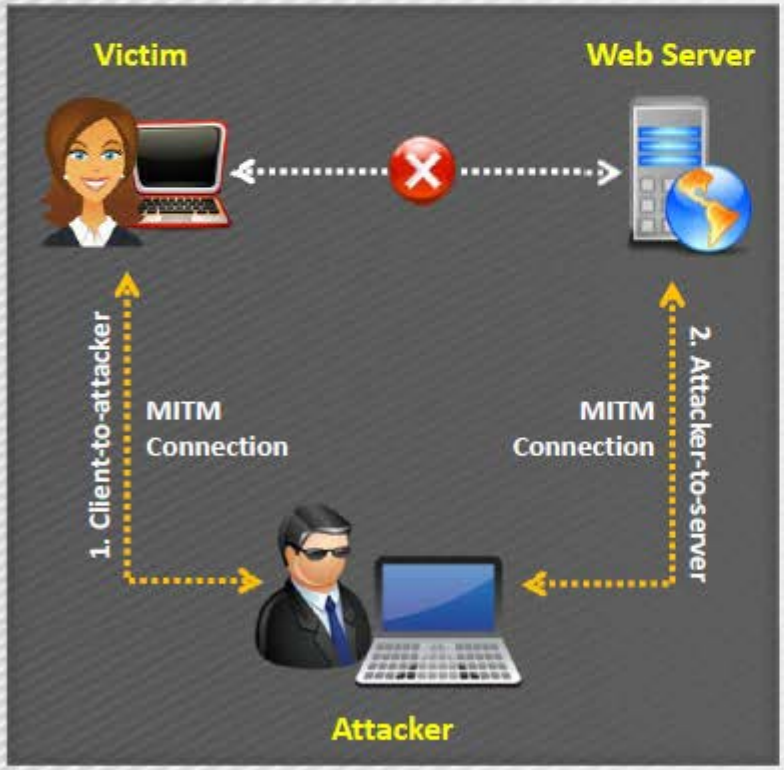
Predicts

At 16:25:55 on Feb-25, 2014, the attacker can successfully predict the session ID to be

```
http://www.juggyboy.com/view/JBEX25022014162555
                            Constant  Date  Time
```

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

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

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

C|EH Certified Ethical Hacker

01

Man-in-the-browser attack **uses a Trojan Horse** to intercept the calls between the browser and its security mechanisms or libraries

02

It works with an already installed Trojan horse and acts between the **browser and its security mechanisms**

03

Its main objective is to cause financial deceptions by manipulating transactions of **Internet Banking systems**

Steps to Perform Man-in-the-Browser Attack

C|EH Certified Ethical Hacker

01 The Trojan first infects the computer's software (OS or application)

02 The Trojan installs malicious code (extension files) and saves it into the browser configuration

03 After the user restarts the browser, the malicious code in the form of extension files is loaded

04 The extension files register a handler for every visit to the webpage

05 When the page is loaded, the extension uses the URL and matches it with a list of known sites targeted for attack

06 The user logs in securely to the website

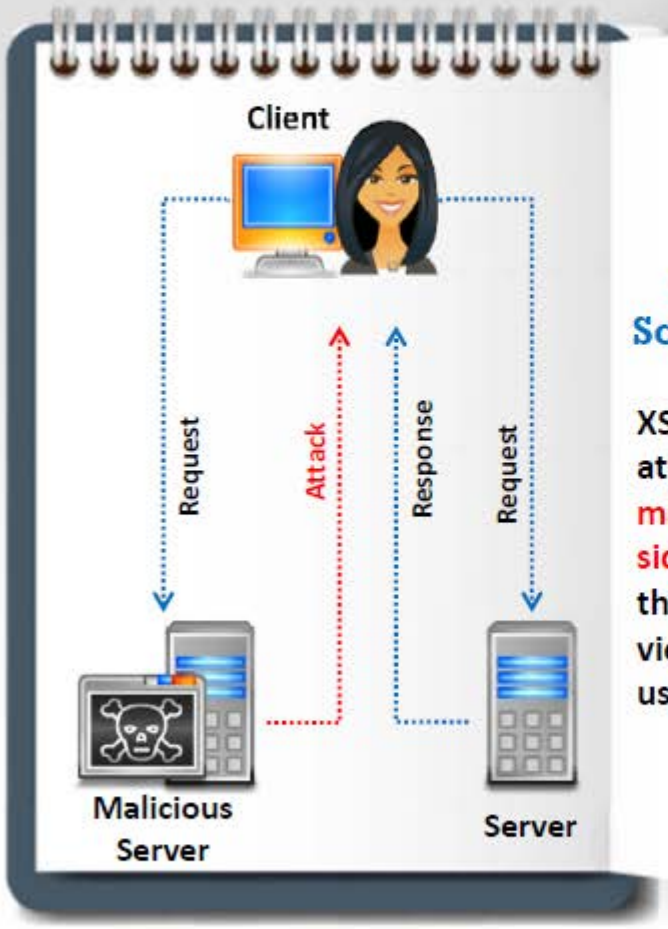
07 It registers a button event handler when a specific page load is detected for a specific pattern and compares it with its targeted list

08 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

Steps to Perform Man-in-the-Browser Attack (Cont'd)

09	The browser sends the form and modified values to the server
10	The server receives the modified values but cannot distinguish between the original and the modified values
11	After the server performs the transaction, a receipt is generated
12	Now, the browser receives the receipt for the modified transaction
13	The browser displays the receipt with the original details
14	The user thinks that the original transaction was received by the server without any interceptions

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

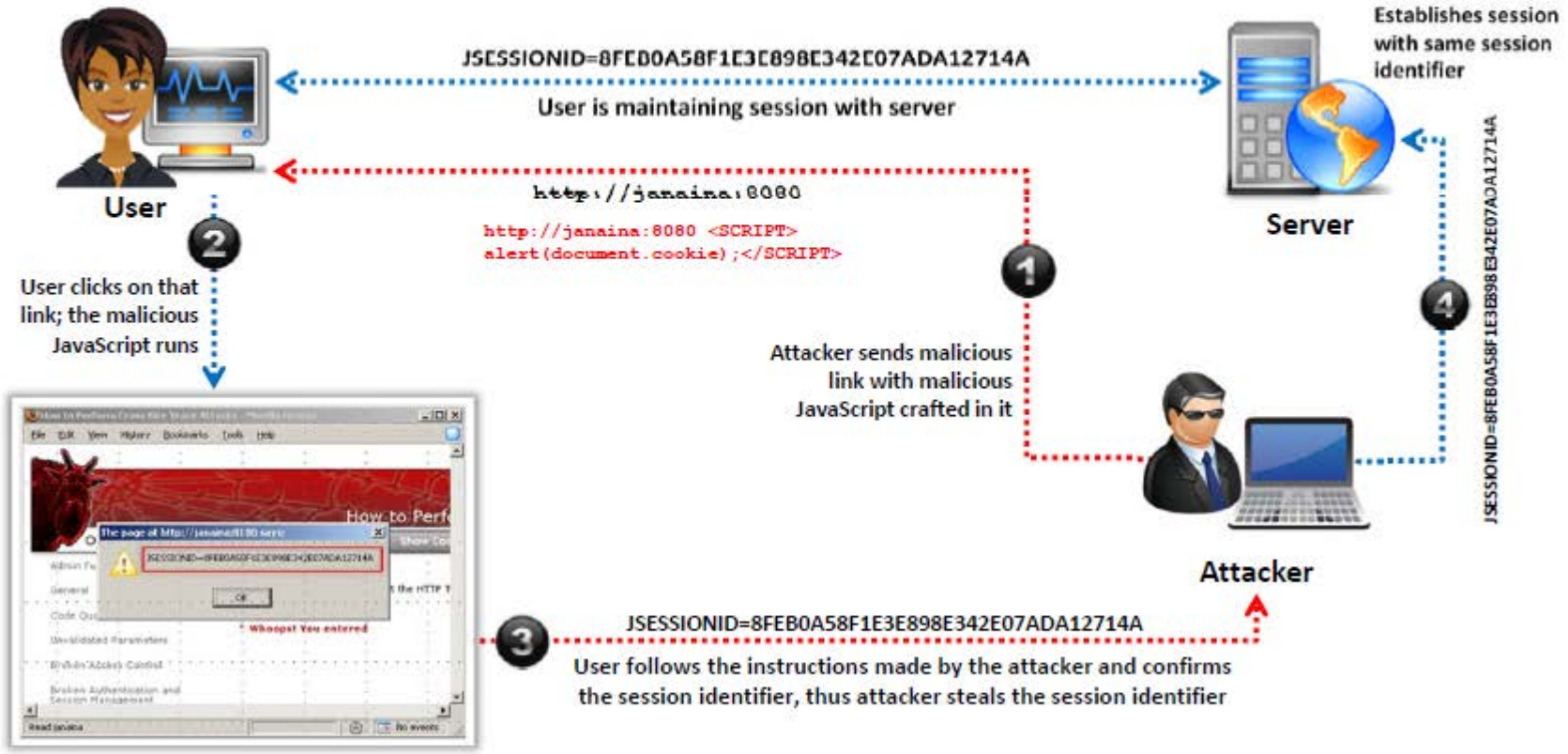
A malicious script can be embedded in a web page that does not generate any warning but it captures session tokens in the background and send it to the attacker

Trojans

A Trojan horse can change the proxy settings in user's browser to send all the sessions through the attackers machine

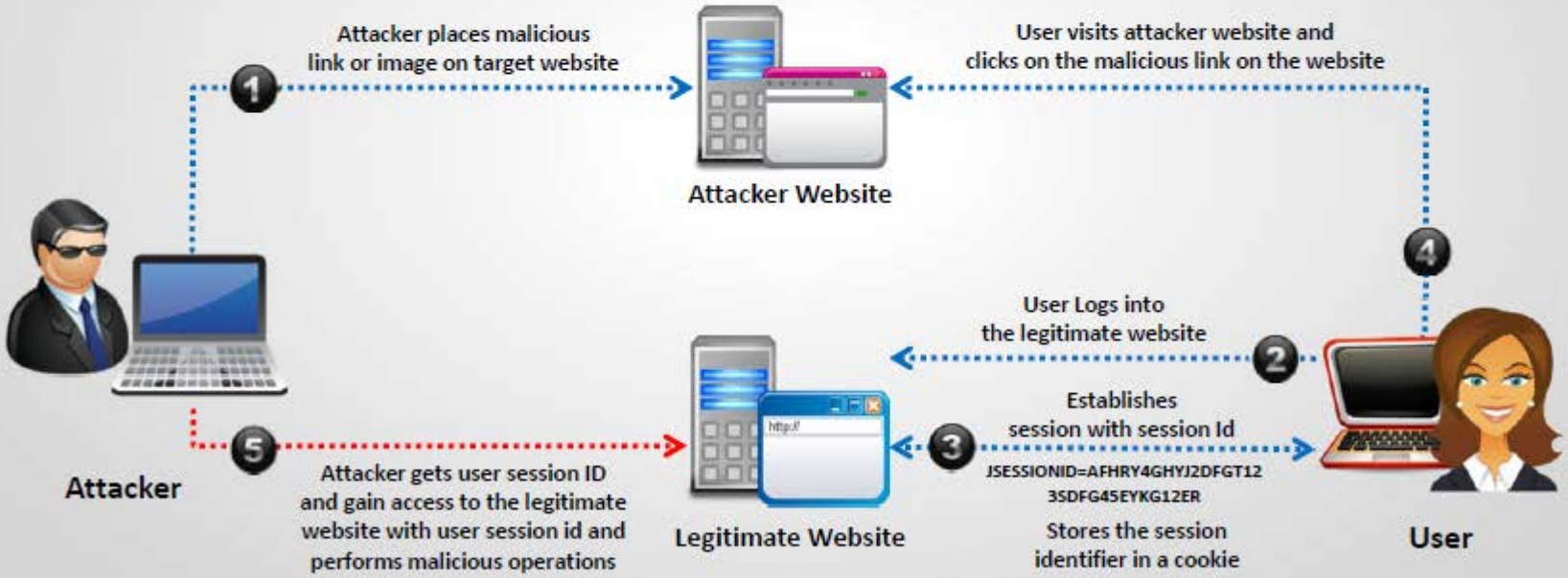
Compromising Session IDs Using Client-side Attacks: Cross-site Script Attack

- If an attacker sends a **crafted link** to the victim with the **malicious JavaScript**, when the victim clicks on the link, the JavaScript will run and complete the instructions made by the attacker



Compromising Session IDs Using Client-side Attacks: Cross-site Request Forgery Attack

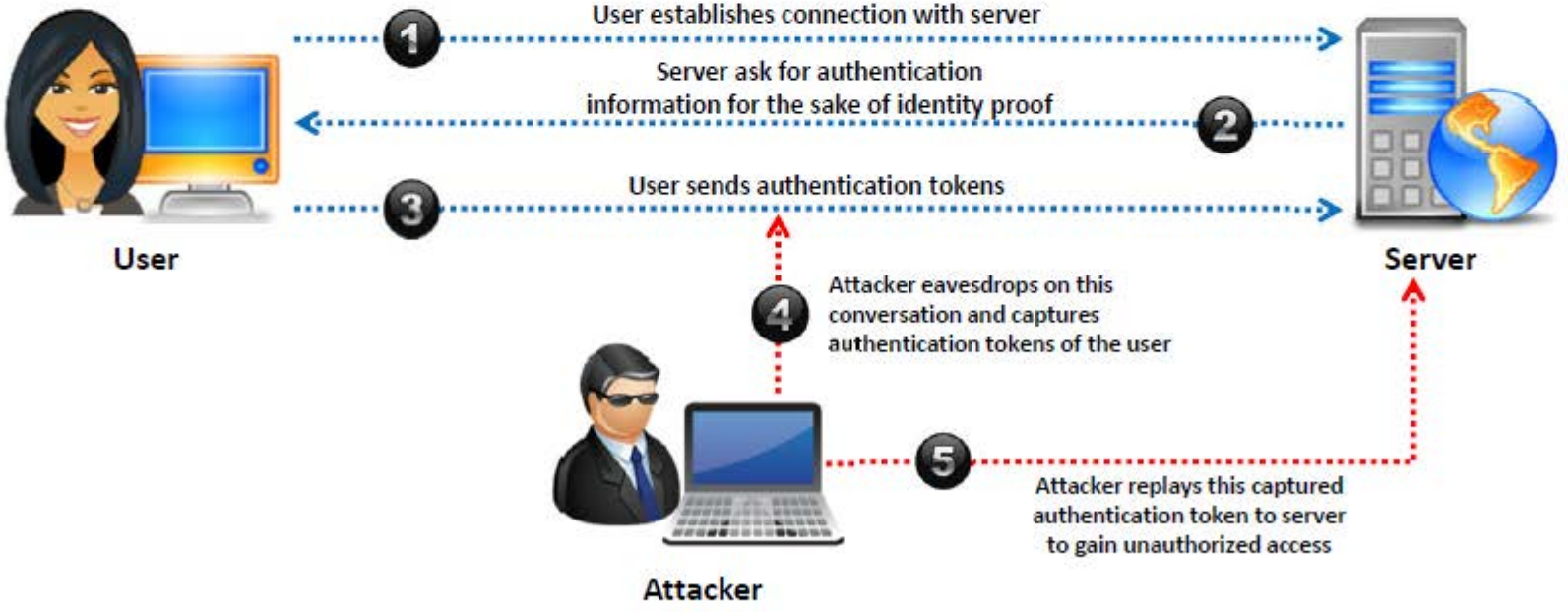
Cross-Site Request Forgery (CSRF) attack **exploits victim's active session** with a trusted site in order to perform malicious activities



Compromising Session IDs Using Session Replay Attack

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

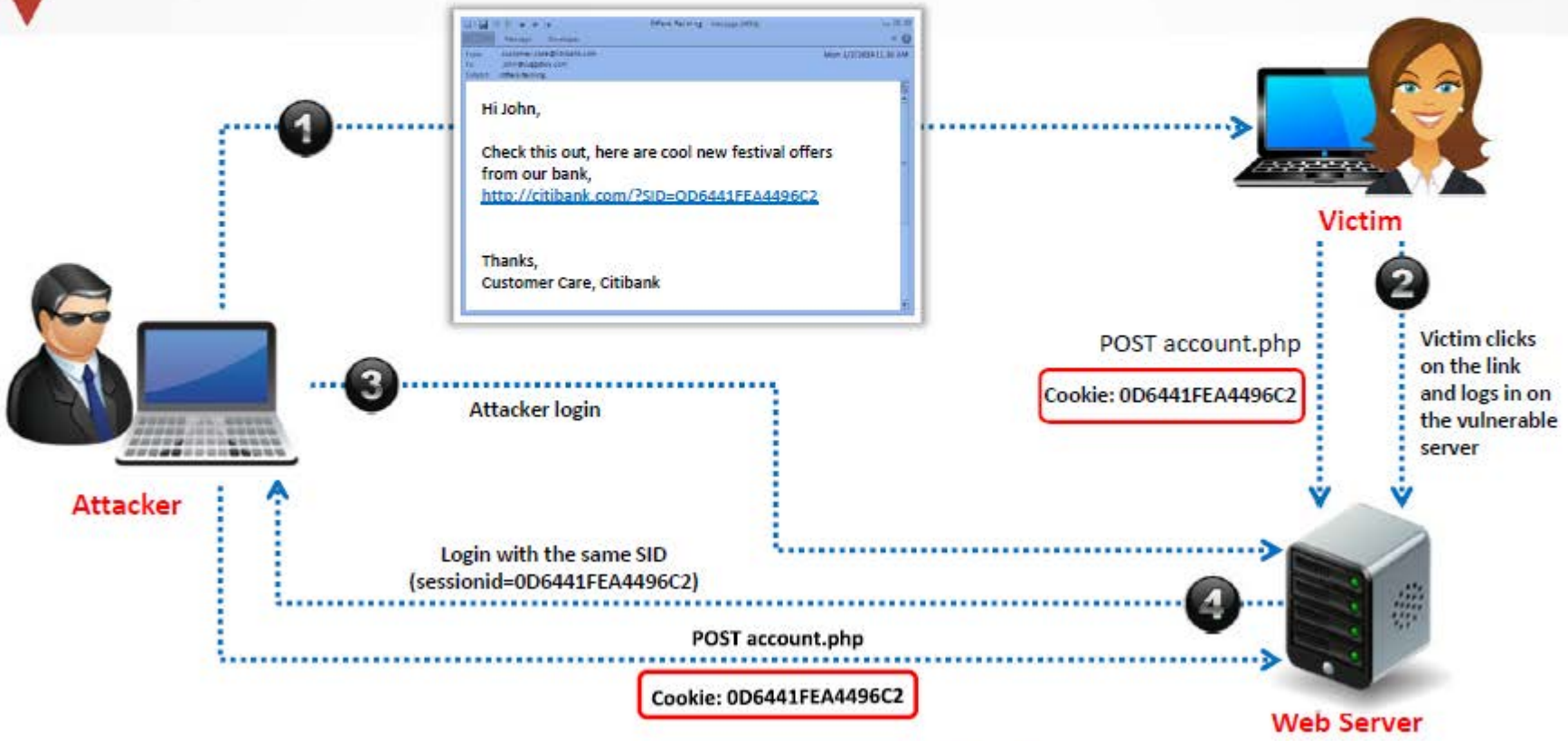


Compromising Session IDs Using Session Fixation

- Session fixation is an attack that allows an attacker to hijack a **valid user session**
- 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**

Session Fixation Attack

- Attacker exploits the **vulnerability of a server** which allows a user to use fixed SID
- Attacker provides a **valid SID** to a victim and lures him to **authenticate himself** using that SID



Module Flow

1. Session Hijacking Concepts
2. Application Level Session Hijacking
3. Network Level Session Hijacking
4. Session Hijacking Tools
5. Countermeasures
6. Penetration Testing

Network-level Session Hijacking

C|EH Certified Ethical Hacker

Session Hijacking

- The network-level hijacking relies on hijacking **transport** and **Internet protocols** used by web applications in the application layer
- By attacking the network-level sessions, the attacker gathers some **critical information** which is used to **attack the application level sessions**

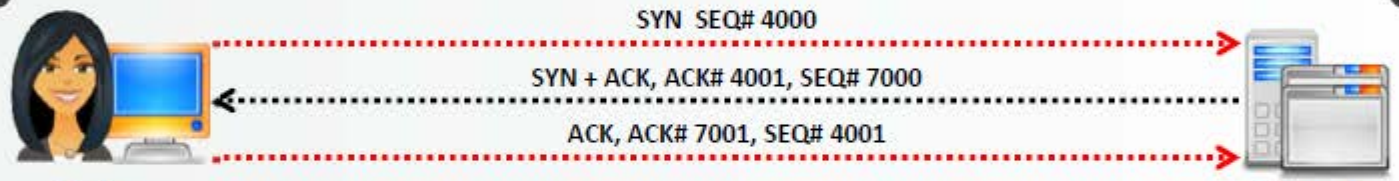
Network-level hijacking includes:

- Blind Hijacking
- UDP Hijacking
- TCP/IP Hijacking
- RST Hijacking
- Man-in-the-Middle: Packet Sniffer
- IP Spoofing: Source Routed Packets

The 3-Way Handshake

C|EH Certified Ethical Hacker

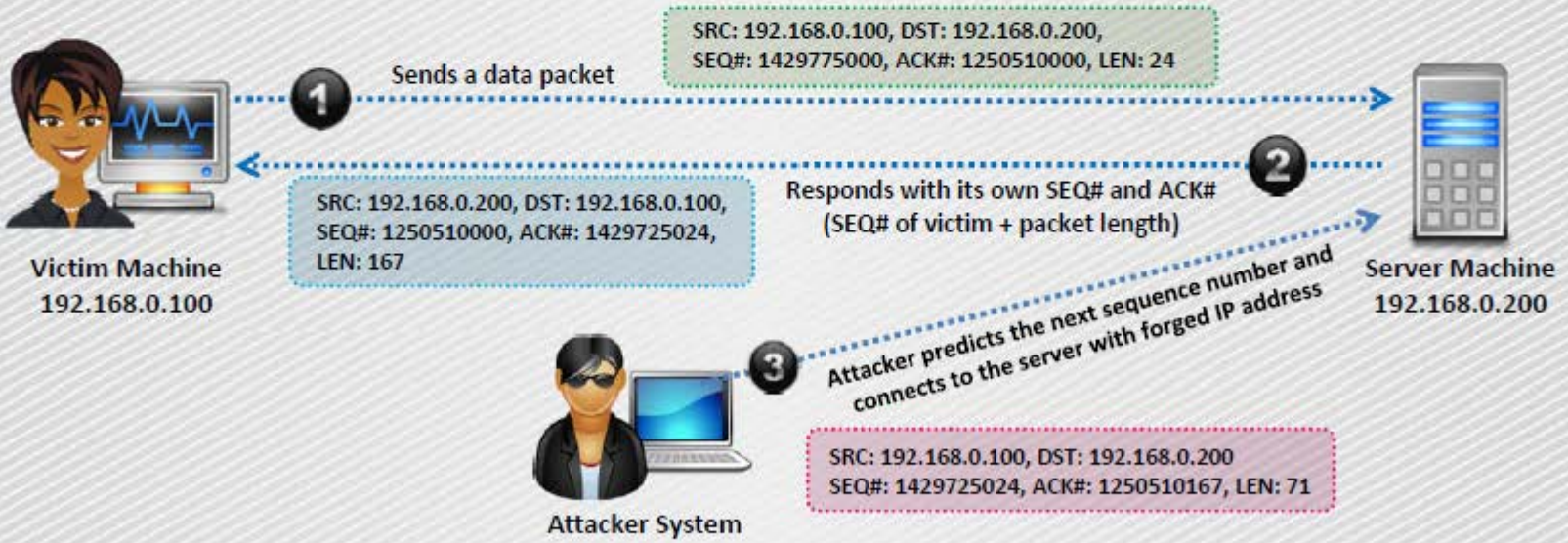
If the attacker can anticipate the **next sequence** and **ACK number** that Bob will send, he/she will spoof Bob's address and start a communication with the server

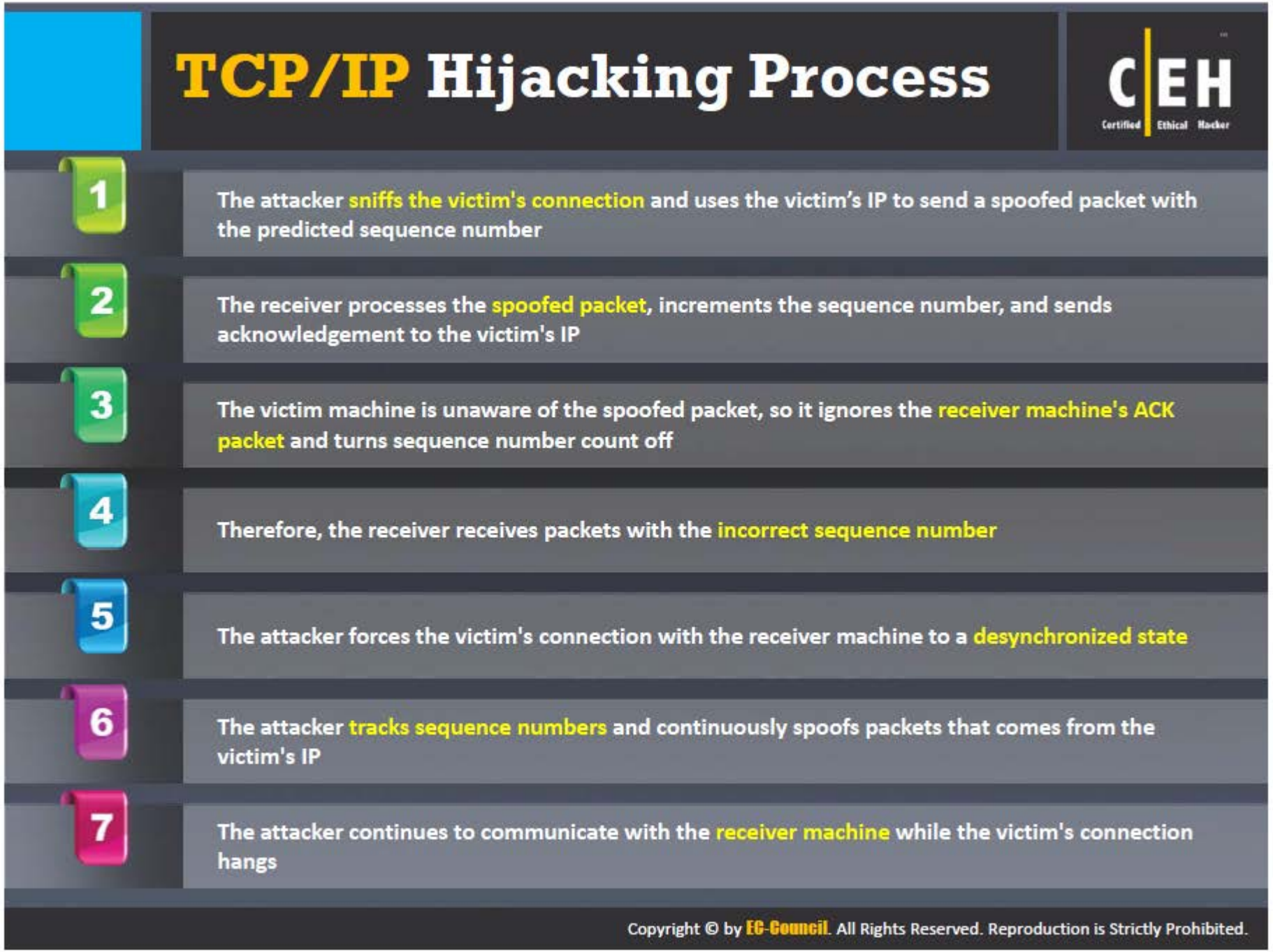


1. Bob initiates a connection with the server and sends a packet to the server with the **SYN flag set**
2. The server receives this packet and sends back a packet with the **SYN + ACK flag** and an **ISN (Initial Sequence Number)** for the server
3. Bob sets the **ACK flag** acknowledging the receipt of the packet and increments the sequence number by 1
4. Now, the two machines successfully **established a session**

TCP/IP 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





IP Spoofing: Source Routed Packets

C|EH Certified Ethical Hacker

01 Packet source routing technique is used for **gaining unauthorized access** to a computer with the help of a trusted host's IP address

02 The attackers spoofs the host's IP address so that the server **managing a session** with the host, accepts the packets from the attacker

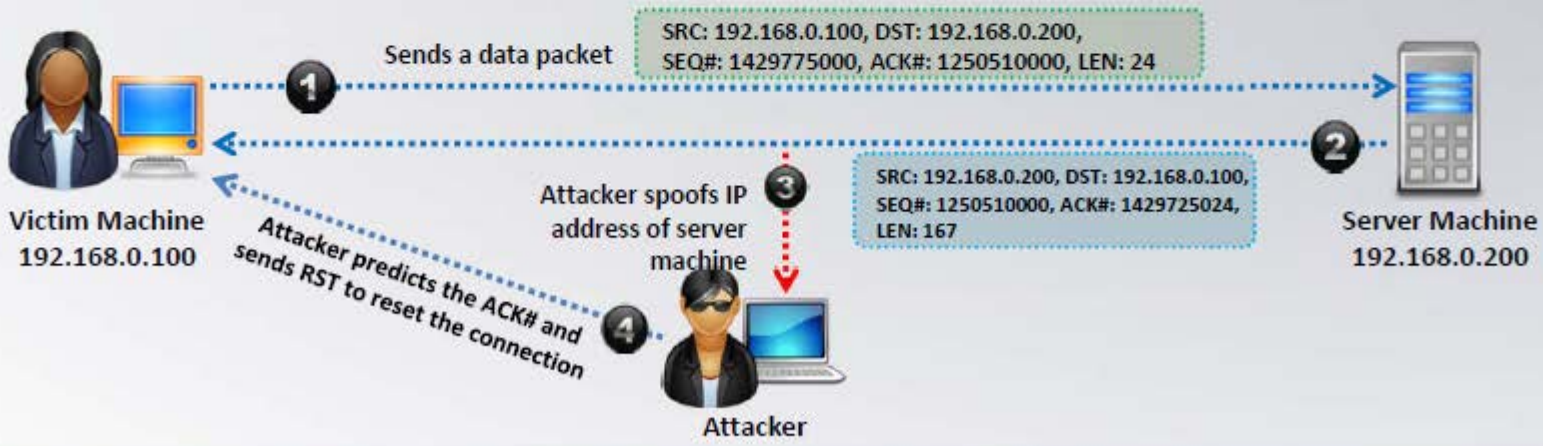
03 When the session is established, the attacker **injects forged packets** before the host responds to the server

04 The original packet from the host is lost as the server gets the packet with a **sequence number** already used by the attacker

05 The packets from attacker are source-routed through the host with the **destination IP** specified by the attacker

RST Hijacking

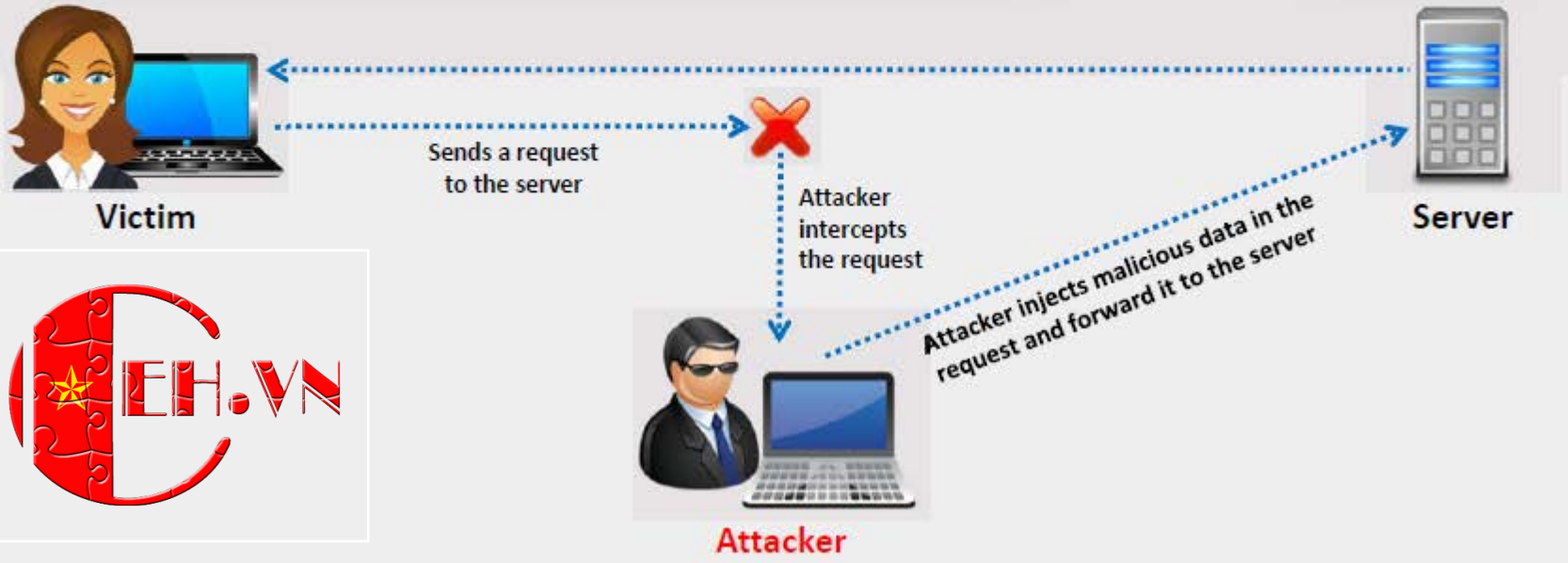
- RST hijacking involves injecting an **authentic-looking reset (RST) packet** using spoofed source address and predicting the acknowledgment number
- The hacker can reset the victim's connection if it uses an **accurate acknowledgment number**
- The victim believes that the source actually sent the **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



Blind Hijacking

01 The attacker can inject the **malicious data or commands** into the intercepted communications in the TCP session even if the source-routing is disabled

02 The attacker can send the data or comments but has no **access to see the response**



MiTM Attack Using Forged ICMP and ARP Spoofing

CEH Certified Ethical Hacker

- In this attack, the 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)

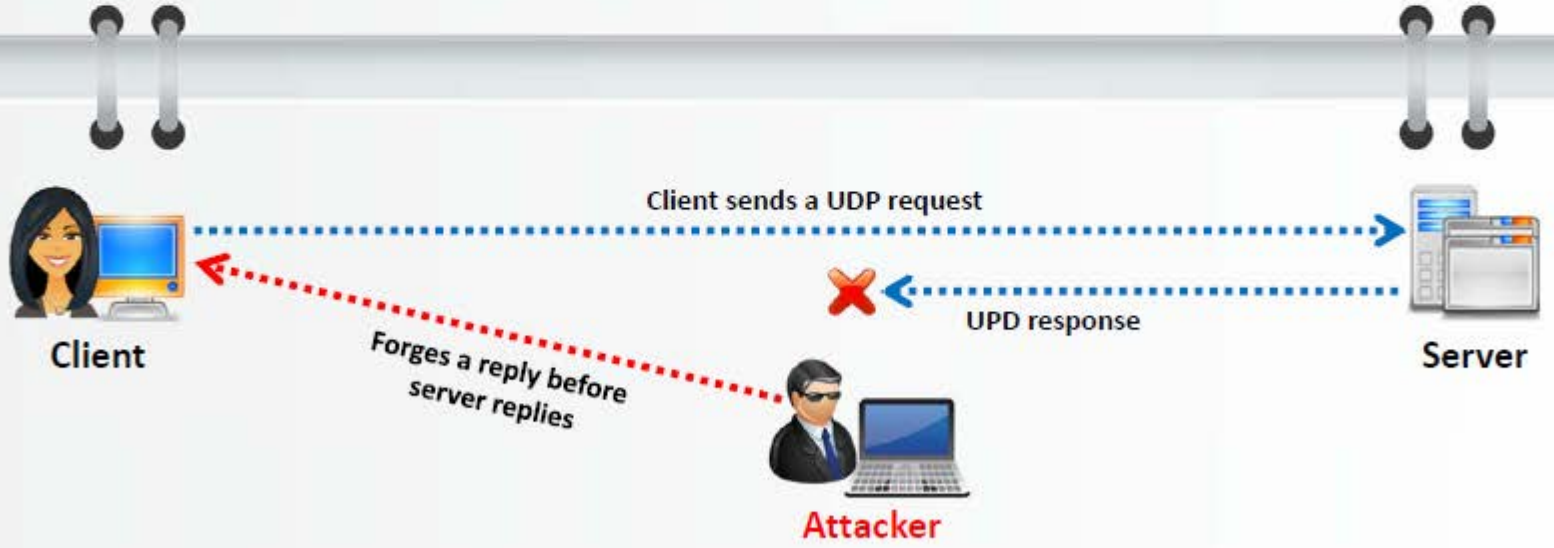
It is an extension of IP to **send error messages** where the attacker can send messages **to fool the client and the server**

Using Address Resolution Protocol (ARP) Spoofing

ARP is used to map the **network layer addresses** (IP address) to **link layer addresses** (MAC address)

UDP 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



Module Flow

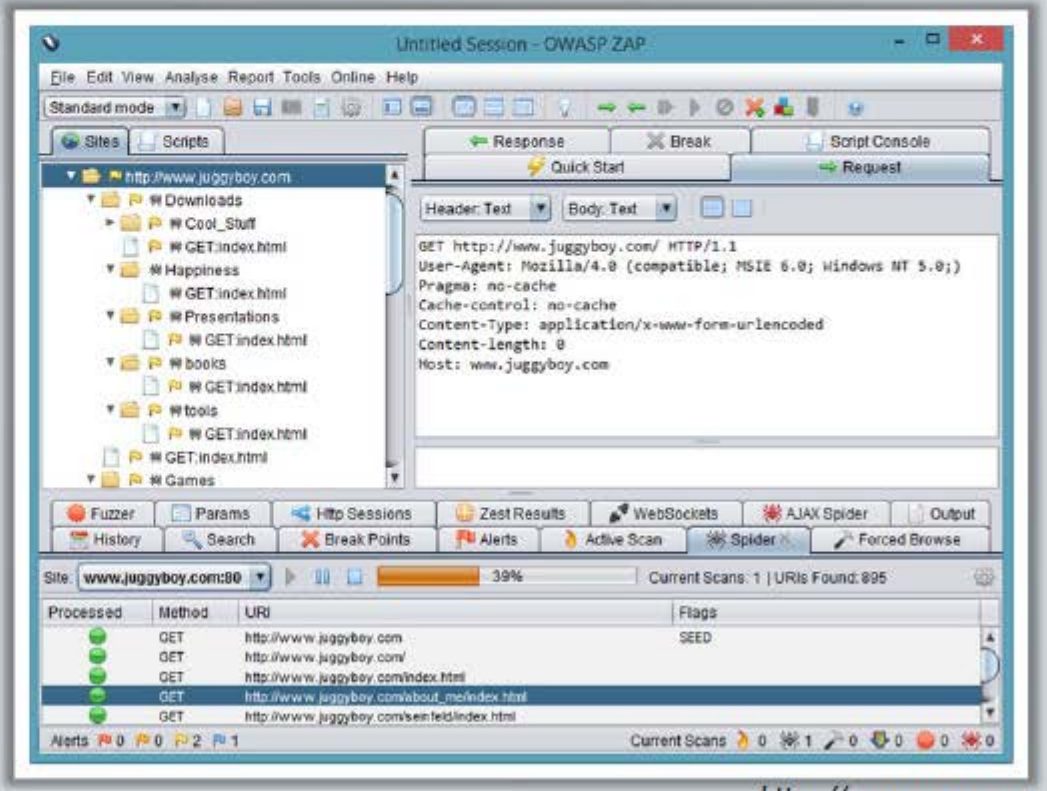
1. Session Hijacking Concepts
2. Application Level Session Hijacking
3. Network Level Session Hijacking
4. Session Hijacking Tools
5. Countermeasures
6. Penetration Testing

Session Hijacking Tool: Zaproxy

The OWASP Zed Attack Proxy (ZAP) is an integrated penetration testing tool for **finding vulnerabilities in web applications**

Features

- Intercepting proxy
- Active scanner
- Passive scanner
- Brute force scanner
- Spider and fuzzer
- Port scanner
- Dynamic SSL certificates
- API
- Beanshell integration



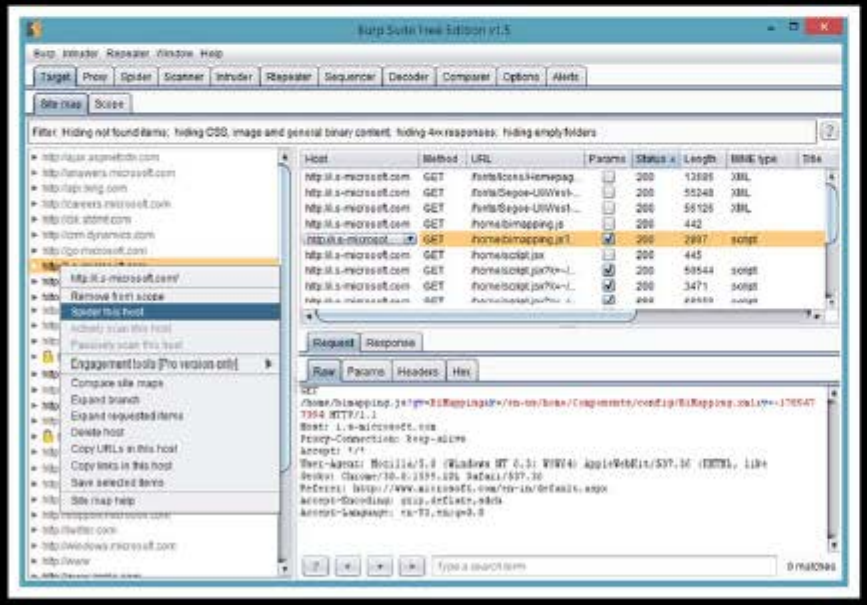
https://www.owasp.org

Session Hijacking Tools: Burp Suite and JHijack

C|EH Certified Ethical Hacker

Burp Suite

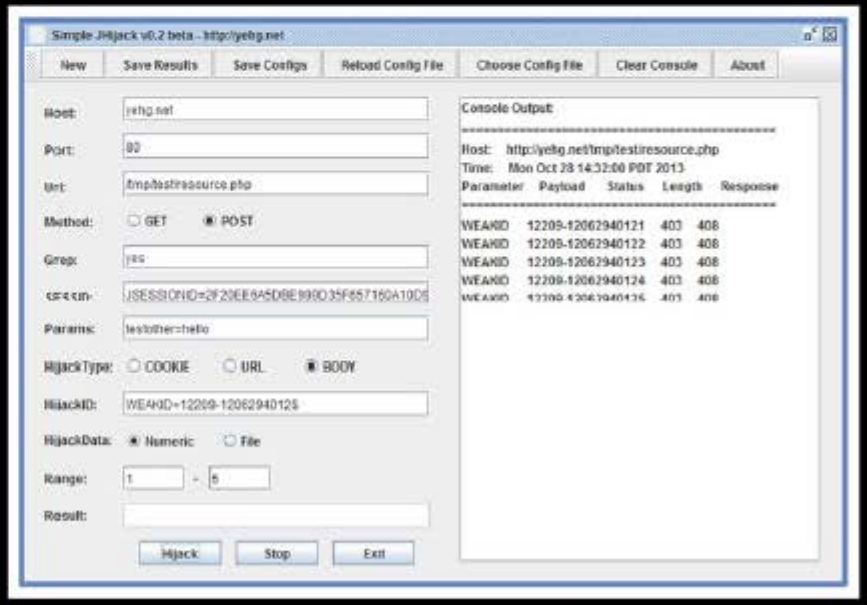
- Burp suite allows the attacker to **inspect and modify traffic** between the browser and the target application
- It **analyzes all kinds of content**, with automatic colorizing of request and response syntax



http://portswigger.net

JHijack

- A Java hijacking tool for **web application session security assessment**
- A simple Java Fuzzer mainly used for **numeric session hijacking** and **parameter enumeration**



http://jhijack.sourceforge.net

Session Hijacking Tools

C|EH Certified Ethical Hacker

Surf Jack
https://code.google.com

Ettercap
http://ettercap.github.io

TamperIE
http://www.bayden.com

PerJack
http://packetstormsecurity.org

WhatsUp Gold Engineer's Toolkit
http://www.whatsupgold.com

Cookie Cadger
https://www.cookiecadger.com

Firesheep
http://codebutler.github.io

CookieCatcher
https://github.com

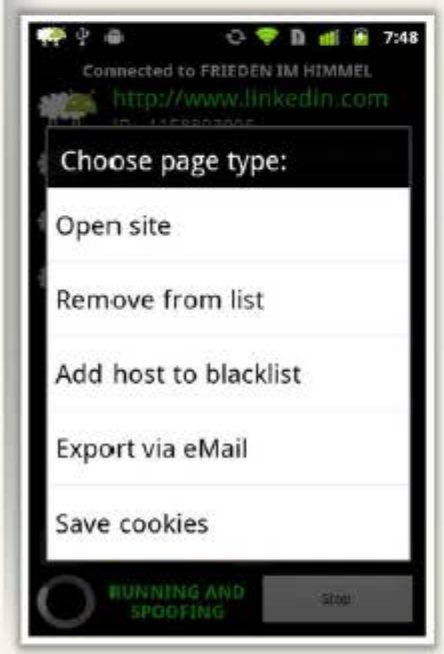
T-sight
http://www.engarde.com

sslstrip
https://pypi.python.org

Session Hijacking Tools for Mobile: DroidSheep and DroidSniff

DroidSheep

- DroidSheep is a simple Android tool for web session hijacking (**sidejacking**)
- It **listens for HTTP packets** sent via a wireless (802.11) network connection and **extracts the session IDs** from these packets



http://droidsheep.de

DroidSniff

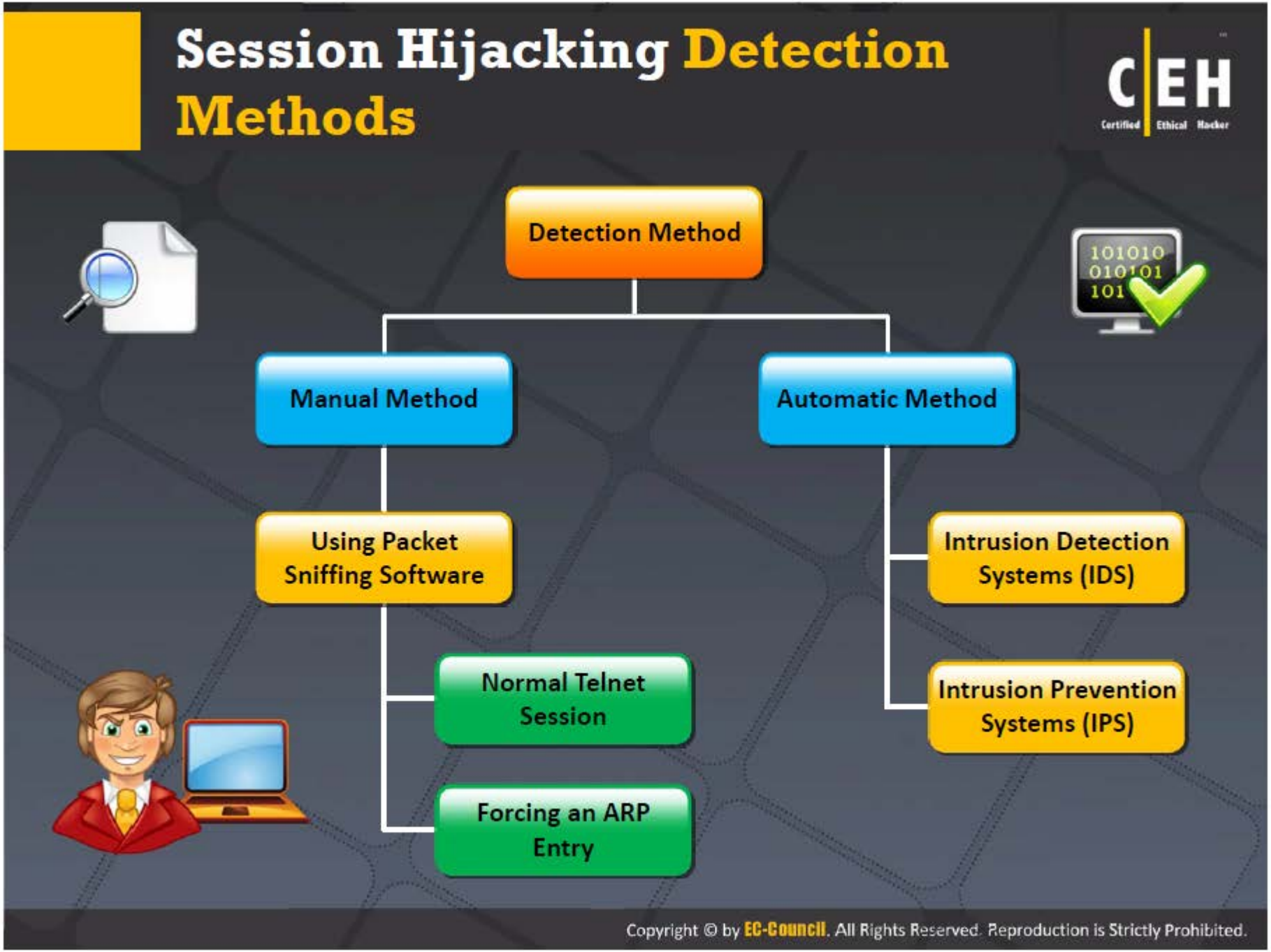
- DroidSniff is an Android app for security analysis in wireless networks and **capturing Facebook, Twitter, Linkedin, and other accounts**

https://github.com

Module Flow

1. Session Hijacking Concepts
2. Application Level Session Hijacking
3. Network Level Session Hijacking
4. Session Hijacking Tools
5. Countermeasures
6. Penetration Testing

Session Hijacking Detection Methods



Protecting against Session Hijacking

C|EH Certified Ethical Hacker

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 session IDs generated by server only
Ensure data in transit is encrypted and implement defense-in-depth mechanism	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	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

Methods to Prevent Session Hijacking: To be Followed by 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

Methods to Prevent Session Hijacking: To be Followed by Web Users

C|EH Certified Ethical Hacker

1. Do not click on the links that are received through **mails or IMs**
2. Use firewalls to prevent the **malicious content** from entering the network
3. Use firewall and browser settings to **restrict cookies**
4. Make sure that the website is certified by the **certifying authorities**
5. Make sure you clear **history**, **offline content**, and **cookies** from your browser after every confidential and sensitive transaction
6. Prefer https, a secure transmission, rather than http when transmitting **sensitive and confidential data**
7. Logout from the browser by **clicking on logout** button instead of closing the browser

Approaches Vulnerable to Session Hijacking and their Preventative Solutions

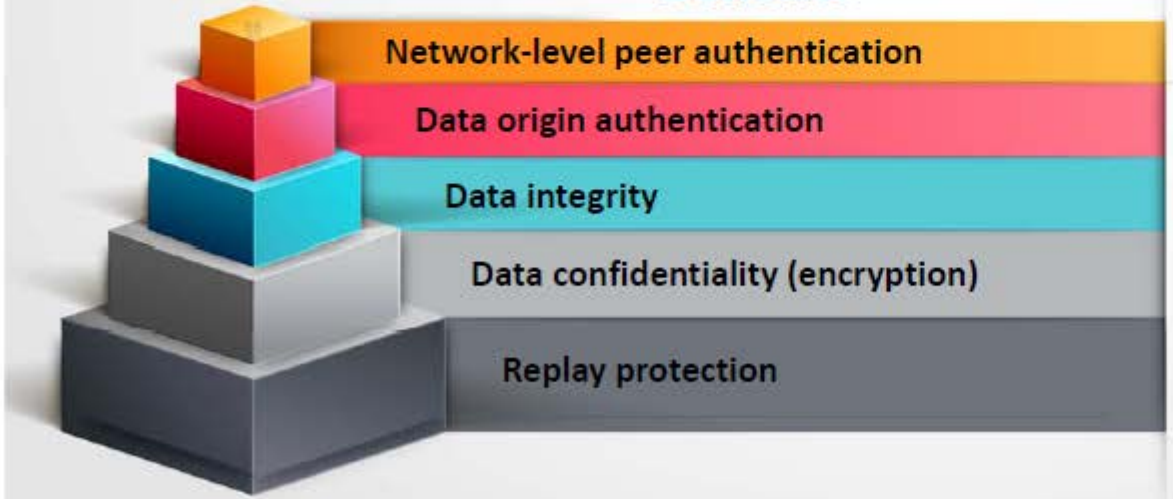
Issue	Solution	Notes
Telnet, rlogin	OpenSSH or ssh (Secure Shell)	It sends encrypted data and makes it difficult for attacker to send the correctly encrypted data if session is hijacked
FTP	sFTP	It reduces the chances of successful hijacking
HTTP	SSL (Secure Socket Layer)	It reduces the chances of successful 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

IPSec

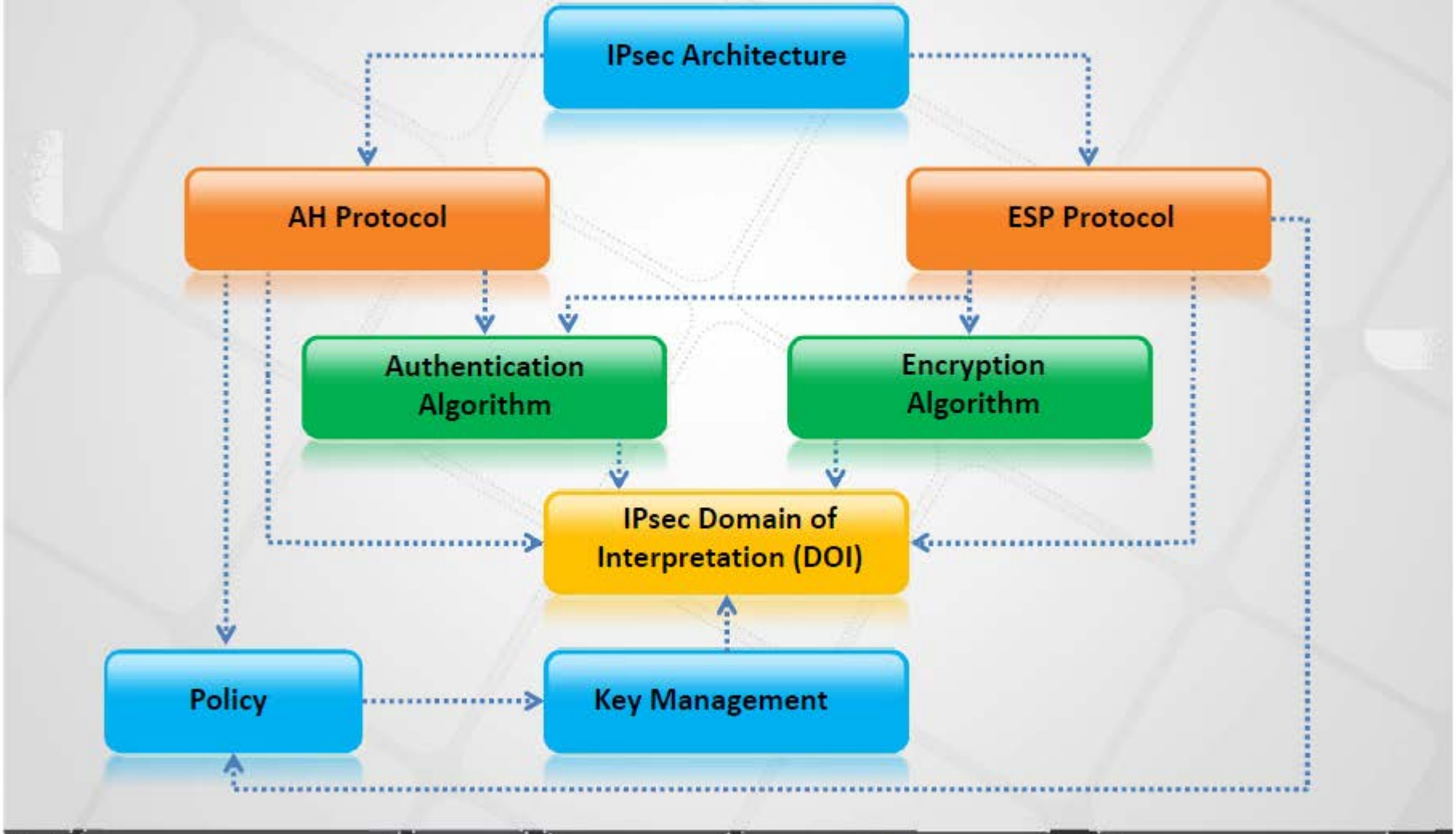
IPSec is a protocol suite developed by the IETF for **securing IP communications** by **authenticating** and **encrypting** each IP packet of a communication session

It is deployed widely to implement **virtual private networks (VPNs)** and for **remote user access** through dial-up connection to private networks

Benefits



IPsec Architecture



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

Oakley

A protocol, which uses the Diffie-Hellman algorithm to create master key, and a key that is specific to each session in IPsec data transfer

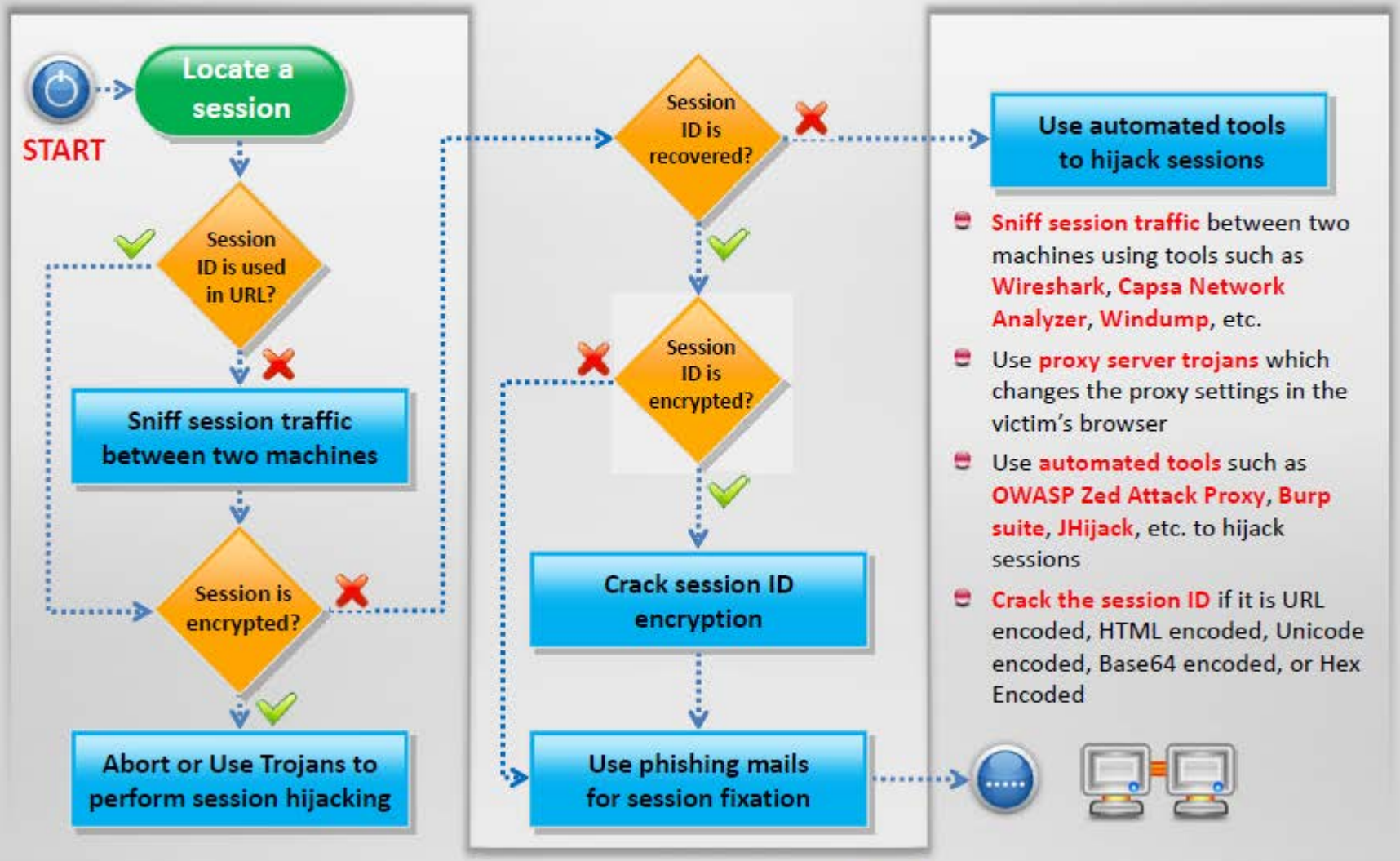
IPsec Policy Agent

A service of the Windows 2000, collects IPsec policy settings from the active directory and sets the configuration to the system at start up

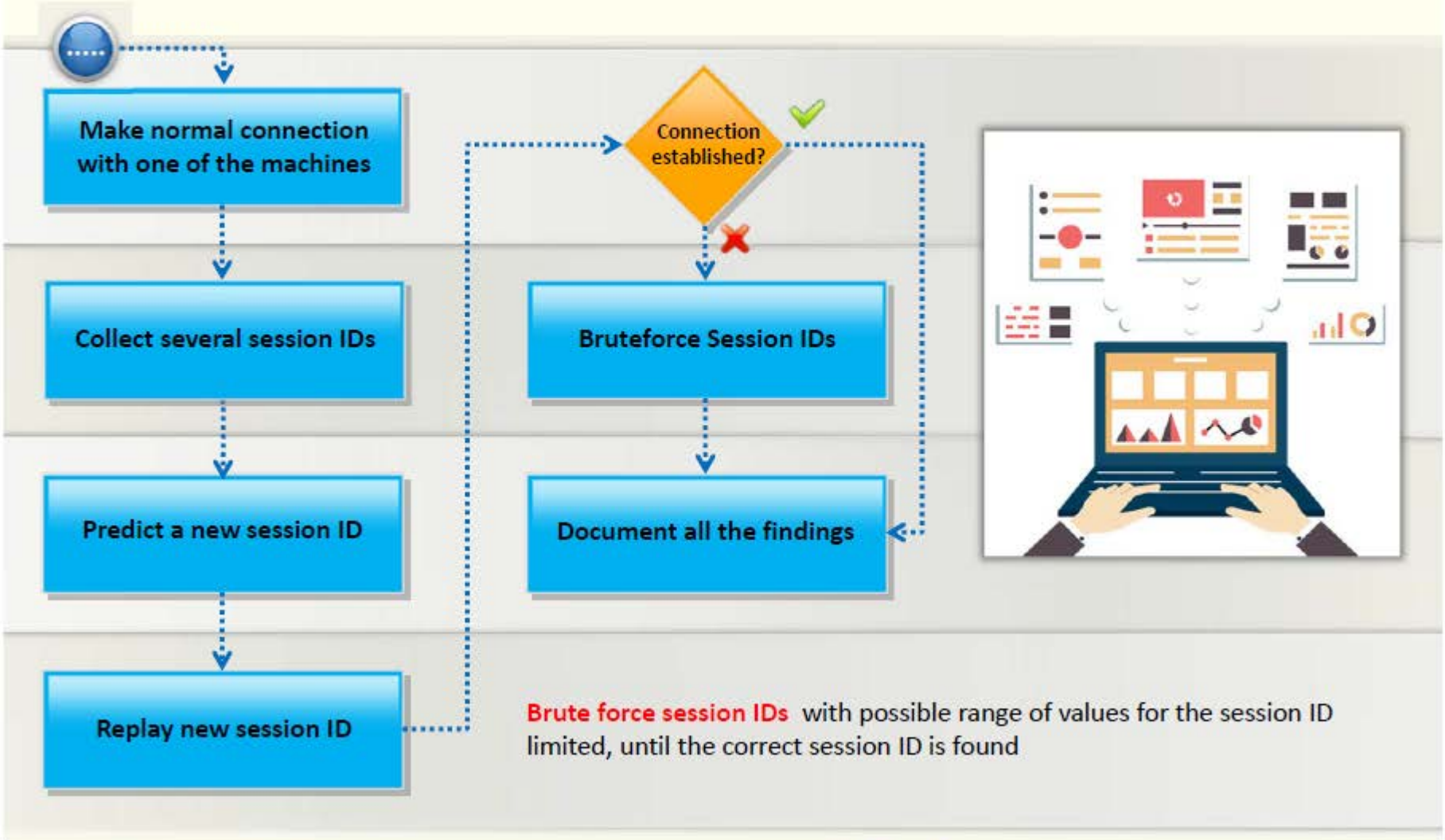
Module Flow

1. Session Hijacking Concepts
2. Application Level Session Hijacking
3. Network Level Session Hijacking
4. Session Hijacking Tools
5. Countermeasures
6. Penetration Testing

Session Hijacking Pen Testing



Session Hijacking Pen Testing (Cont'd)



Module Summary

- In session hijacking, an attacker relies on the legitimate user to connect and authenticate, and will then take over the session
- In a spoofing attack, the attacker pretends to be another user or machine to gain access
- Successful session hijacking is difficult and is only possible when a number of factors are under the attacker's control
- Session hijacking can be active or passive in nature depending on the degree of involvement of the attacker
- By attacking the network-level sessions, the attacker gathers some critical information that is used to attack the application-level sessions
- A variety of tools exist to aid the attacker in perpetrating a session hijack
- Session hijacking could be dangerous, and therefore, there is a need for implementing strict countermeasures