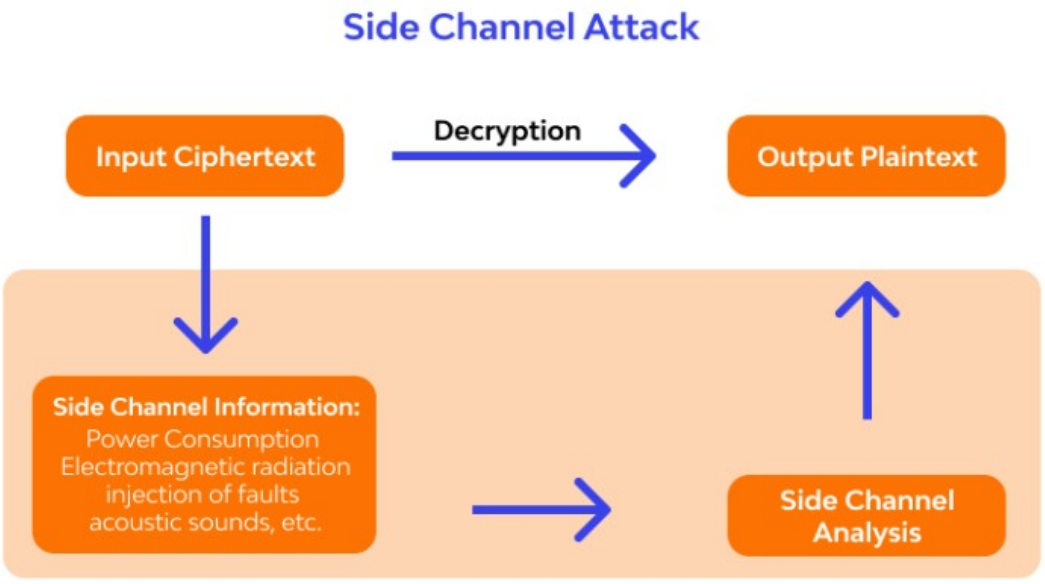
Cyber QB for ISE:

Q1) What are side channel attacks and give example for the same.

Ans: Side-channel attacks exploit information leaked from a system's physical implementation, such as power consumption, timing, or electromagnetic emissions, rather than directly targeting its code. These attacks analyze "side channels" to extract sensitive data like cryptographic keys. Examples include timing attacks, power analysis, and electromagnetic attacks, which can be used to compromise everything from microchips to web browsers.



Examples of side-channel attacks (any 2-3)

Timing Attacks: The attacker measures the time it takes for a system to perform operations. By analyzing variations in execution time, they can infer information about the data being processed, such as a secret key.

Power Analysis Attacks: An attacker monitors the power consumption of a device. Since the power usage changes depending on the operation, it can reveal secrets. A specific type called {Differential Power Analysis} (DPA) uses statistical methods to analyze many power traces to find patterns and extract information, even from a black-box system.

Electromagnetic Attacks: Similar to power analysis, this attack involves measuring the electromagnetic fields that a device emits during its operation. These emissions can be captured and analyzed to reveal sensitive data.

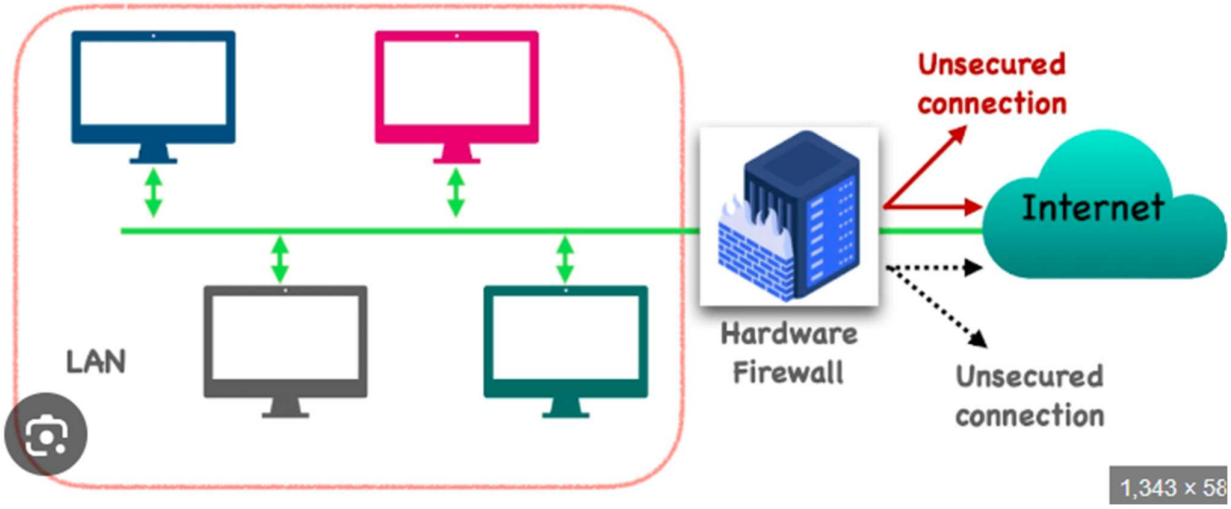
Cache Attacks: These attacks exploit the way a computer's cache memory works. By monitoring how the cache is accessed, an attacker can deduce what memory locations other processes are using, which can leak sensitive information. An example is the Prime+Probe attack.

Acoustic Attacks: In this type of attack, the sound produced by a device's components is analyzed. For example, a sensitive microphone could be used to listen to the sounds of a hard drive or a CPU to determine the data being accessed, much like a digital safecracker listening to the tumblers of a lock.

Specific Vulnerabilities: Some attacks target specific hardware vulnerabilities. For example, the {iLeakage attack} was an Apple-specific attack that gathered information from a user's Safari browser by exploiting a technique called speculative execution to extract browsing history and credentials.

Q3) What is Firewall? Explain it's types and how it is different from IDS.

Ans: A firewall is a network security device or software that monitors and controls incoming and outgoing network traffic based on predefined security rules. Its main purpose is to create a barrier between a trusted internal network and untrusted external networks (like the Internet) to prevent unauthorized access and cyber-attacks.



Types:

Packet-Filtering Firewall: Checks packets' IP addresses and ports; allows or blocks them based on rules.

Stateful Inspection Firewall: Monitors active connections and makes decisions based on the state of the traffic.

Proxy (Application-Level) Firewall: Acts as an intermediary between users and the internet; inspects data at the application layer.

Next-Generation Firewall (NGFW): Combines traditional firewall features with intrusion prevention and deep packet inspection.

Cloud Firewall (Firewall-as-a-Service): Cloud-based firewall that protects distributed networks and cloud environments.

⚔️ Firewall vs IDS (Intrusion Detection System)

Feature	Firewall	IDS (Intrusion Detection System)
Primary Function	Blocks or allows traffic based on rules	Detects and alerts about suspicious activity
Action Type	Preventive (proactive)	Detective (reactive)
Placement	At the network perimeter (between internal and external networks)	Inside the network or alongside the firewall
Traffic Control	Controls traffic flow	Monitors traffic and generates alerts
Response	Can block or permit traffic	Only alerts; cannot block traffic (unless combined with IPS)
Layer of Operation	Network, Transport, and sometimes Application layers	Network and Application layers
Example	Cisco ASA, FortiGate, Palo Alto	Snort, Suricata, OSSEC

Q6) short note on Clickjacking, cross-site scripting and significance of Honeypot.

Ans:

Clickjacking

- **Clickjacking, also known as a "UI redress attack"**, is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the top level page. Thus, **the attacker is "hijacking" clicks** meant for their page and routing them to another page, most likely owned by another application, domain, or both.

- Using a similar technique, **keystrokes can also be hijacked.** With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker.

- Typically, clickjacking is performed by displaying an invisible page or HTML element, inside an iframe, on top of the page the user sees. The user believes they are clicking the visible page but in fact they are clicking an invisible element in the additional page transposed on top of it.

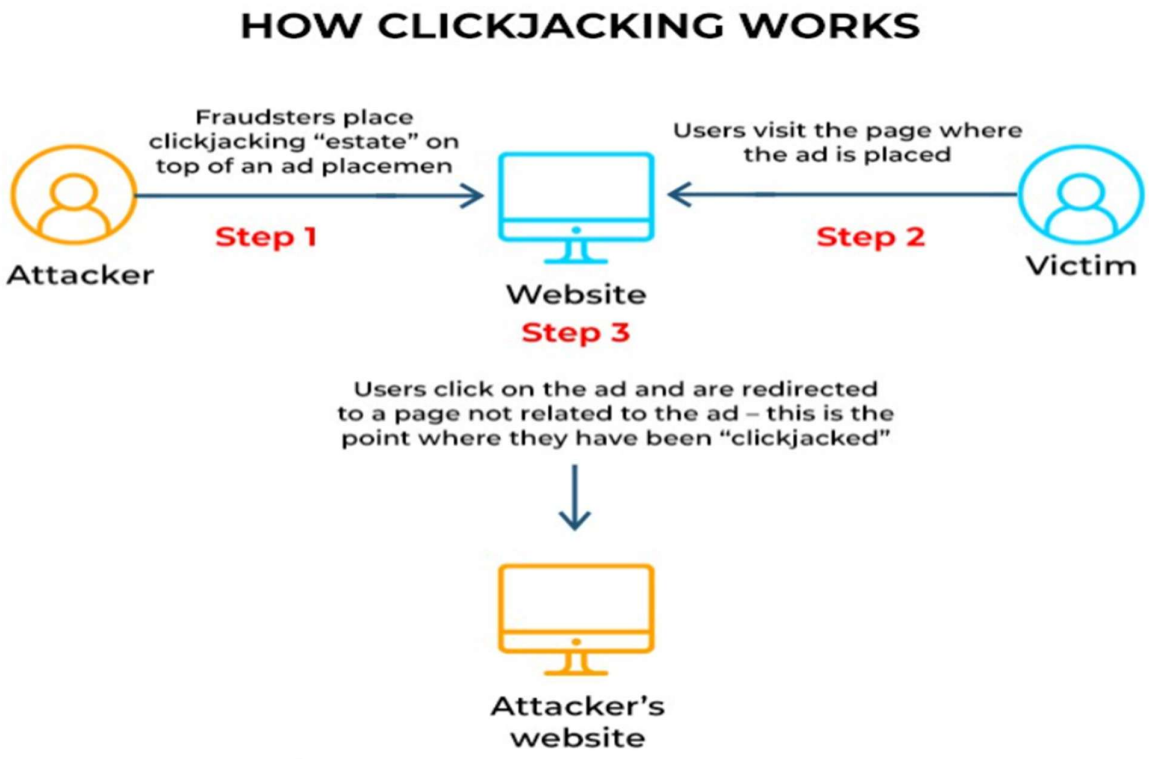
How does clickjacking work?

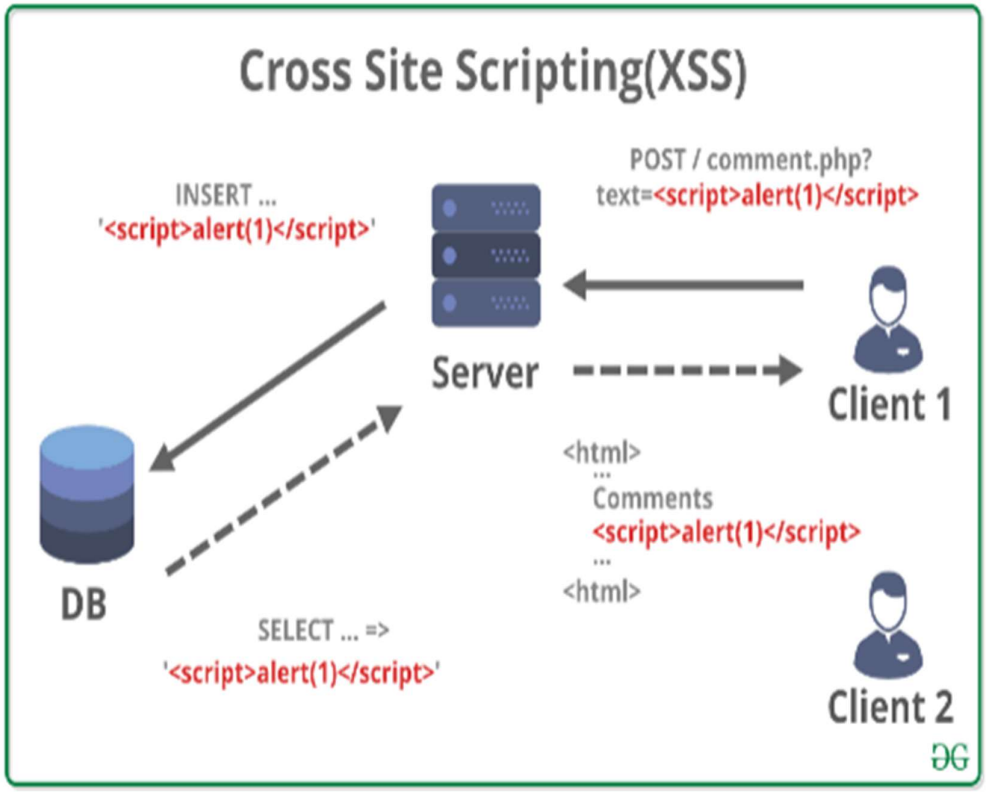
- A frame inside a frame is known as an iframe. **iFrames** allow you to incorporate material from other websites into your own. When you go to a website and see an integrated YouTube video, the video is shown in an iframe.

Clickjacking is made possible because of HTML frames or iframes

If a web page can be displayed within a frame, attackers can cover the original web page with a hidden, transparent layer with its own JavaScript and UI elements. The outward appearance of the web page remains unchanged, which means users have no reason to suspect anything might be amiss.

Users then navigate the web page, expecting links and buttons to work normally. But the hidden UI means the attacker's script works instead. The attacker's script can work behind the scenes to make it appear as though nothing is wrong. This makes a range of malicious actions possible, including: Authorizing money transfers, Identifying your location, Stealing credentials etc.


HOW CLICKJACKING WORKS
Fraudsters place clickjacking "estate" on top of an ad placemen
Step 1
Attacker
Website
Step 3
Users visit the page where the ad is placed
Step 2
Victim
Users click on the ad and are redirected to a page not related to the ad – this is the point where they have been "clickjacked"
Attacker's website


Cross Site Scripting(XSS)
INSERT ...
'<script>alert(1)</script>'
POST / comment.php?
text=<script>alert(1)</script>
Server
Client 1
DB
<html>
...
Comments
<script>alert(1)</script>
...
<html>
SELECT ... =>
'<script>alert(1)</script>'
Client 2

Cross-Site Scripting (XSS) Overview

- **What it is:**
  XSS is a security vulnerability where attackers inject malicious scripts (usually JavaScript) into web pages viewed by other users. The script runs in the victim's browser, appearing as if it's from the trusted website.
- **Why it matters:**
  It can lead to **account compromise, data theft, session hijacking, malware spread**, and more.
- **Origin:**
  Initially called **CSS (Cross Site Scripting)** but renamed to **XSS** to avoid confusion with Cascading Style Sheets (CSS).
- **How it happens:**
  Occurs when user input is not properly sanitized and is embedded in the webpage response, allowing malicious code execution.

Types of XSS

1. Reflected XSS

- Occurs when the malicious script is reflected off a web server in an error message, search result, or any other response that includes some or all of the input sent to the server as part of the request.
- The payload is delivered via a link, and the victim has to click it for the attack to trigger.
- Example: A search field that takes user input and reflects it back on the page without proper sanitization.

2. Stored XSS (Persistent XSS)

- The malicious script is stored permanently on the target server, such as in a database, comment section, or message board.
- Every time a user accesses the stored content, the script runs.
- More dangerous because it doesn't require user interaction beyond simply visiting the affected page.

3. DOM-Based XSS

- Happens entirely on the client side — when client-side scripts write user input directly to the Document Object Model (DOM) without proper sanitization.
- Can be either reflected or stored in nature.
- The server is unaware since the vulnerability happens in browser-side code.

A **honeypot** is a **decoy system or server** designed to attract attackers and study their behavior without exposing real systems to risk. It plays a crucial role in **cybersecurity defense and research**.

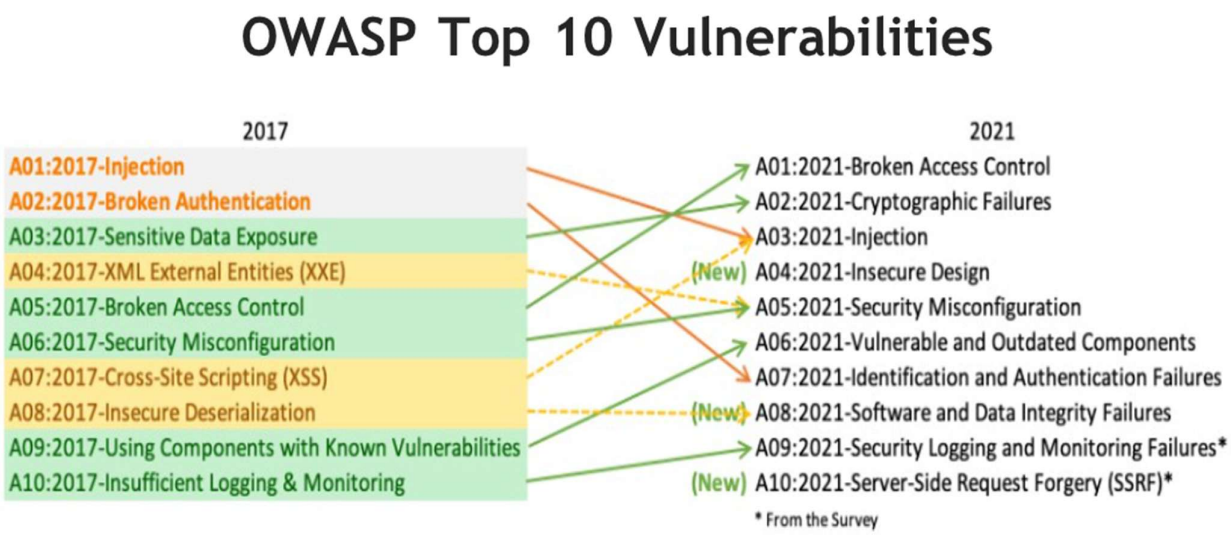
Here's the **significance** of honeypots 👇

- **Detects Intrusions:** Identifies unauthorized access attempts and new attack patterns.
- **Distracts Attackers:** Diverts attackers away from critical systems, buying time for defense.
- **Analyzes Attack Methods:** Helps security experts study hacker tools, tactics, and motives.

- **Improves Security Measures:** Provides insights to strengthen network defenses and update security policies.
- **Early Threat Warning:** Acts as an early alert system for potential attacks or vulnerabilities.
- **Low False Alarms:** Since legitimate users don't access honeypots, any activity is likely malicious.
- **Supports Forensic Investigation:** Captures data useful for tracing attackers and understanding breaches.

Q8) Outline the top 10 security projects in OWASP with their analysis.

Ans:



- **Green arrows** are vulnerabilities that were promoted in importance
- **Orange arrows** are vulnerabilities that were demoted in importance
- **Yellow broken line arrows** are vulnerabilities removed and merged into other categories.

Q11) Difference between IDS and IPS.

Ans:

Feature	IDS (Intrusion Detection System)	IPS (Intrusion Prevention System)
Main Function	Detects and alerts about suspicious activity	Detects **and prevents** suspicious activity
Action Type	**Passive** – monitors traffic and sends alerts	**Active** – monitors and blocks malicious traffic
Response	Only **alerts** administrators	**Blocks, drops, or modifies** malicious packets automatically
Placement	Usually **outside** the main traffic path (monitoring mode)	Placed **inline** with network traffic (in the data path)
Impact on Traffic	No impact on network flow	Can affect traffic flow and latency slightly
Use Case	Ideal for monitoring and analysis	Ideal for real-time threat prevention
Example Tools	Snort (IDS mode), Suricata, OSSEC	Snort (IPS mode), Cisco Firepower, Palo Alto IPS

Q12) Explain steps involved for execution of XSS refractive on DVWA.

Ans:

Steps : (refer XSS diagram above)

1] Lab setup & authorization: Run DVWA locally (or an authorized test environment) and set appropriate security level—only test where you have permission.

2] Identify input point: Locate a page/parameter that echoes user input back in the HTTP response.

3] Confirm reflection: Send a harmless unique token and inspect the response (browser DevTools / proxy) to verify the input appears.

4] Context analysis & PoC: Determine the reflection context (HTML, attribute, JS) and create a safe, non-destructive proof-of-concept in the lab to show script execution.

5] Document & remediate: Record request/response, screenshots, and recommend fixes (output encoding, input validation, CSP, HttpOnly cookies) and report responsibly.

Q13) Difference between XSS and cross-site request froggery citing with example.

Ans:

Aspect	XSS (Cross-Site Scripting)	CSRF (Cross-Site Request Forgery)
Meaning	Attacker injects **malicious scripts** into a trusted website, which run in a user's browser.	Attacker tricks a **logged-in user** into unintentionally sending a request to a trusted site.
Goal	Steal user data, session cookies, or deface web pages.	Perform unauthorized actions (like money transfer, password change) on behalf of the user.
Vulnerability Location	Exists in the **web application's code** that fails to sanitize user input.	Exploits the **trust a site has in a user's browser session**.
User Interaction	User views a malicious page or input that executes code in their browser.	User unknowingly clicks a link or loads a page that sends a forged request.
Example	A comment box allows `<script>alert('Hacked')</script>` to execute in other users' browsers.	A malicious website makes a user's browser send a hidden request to `bank.com/transfer?amount=1000` while the user is logged in.

Q14) Difference between backdoor and trapdoor.

Ans:

Feature	Backdoor	Trapdoor
Definition	A secret method used to gain unauthorized access to a system or software, bypassing normal authentication mechanisms.	A hidden entry point intentionally built into a program or system to allow access under specific conditions, often for testing or maintenance.
Purpose	Usually **malicious** — used by attackers to exploit a system after compromising it.	Often **intentional and legitimate** — created by developers for debugging, testing, or maintenance (though it can be misused).
Created By	Hackers or malware developers.	Original software developers.
Visibility	Not officially documented; designed to remain hidden.	May be known internally (to developers), though hidden from users.
Example	A hacker installs a hidden program that lets them log in later without authentication.	A programmer adds a special password that allows bypassing login during testing.
Risk	High — enables persistent unauthorized access.	Moderate — can become a vulnerability if discovered or abused.

Q15) Explain session hijacking and its management.

Ans:

Session Hijacking and Its Management

Session hijacking is a cyber attack in which an attacker takes control of a valid user session by stealing or predicting session IDs or tokens, allowing unauthorized access without the user's credentials.
Common methods include session sniffing, XSS, session fixation, and man-in-the-middle attacks.

Consequences: The attacker can impersonate the user, steal data, or perform malicious transactions.

Management / Prevention:

- Use HTTPS (TLS) to encrypt communication.
- Set cookies with Secure, HttpOnly, and SameSite attributes.
- Regenerate session IDs after login to avoid fixation.
- Implement short session timeouts and automatic logout.
- Protect against XSS and CSRF attacks.
- Use Multi-Factor Authentication (MFA) and monitor abnormal session activity.