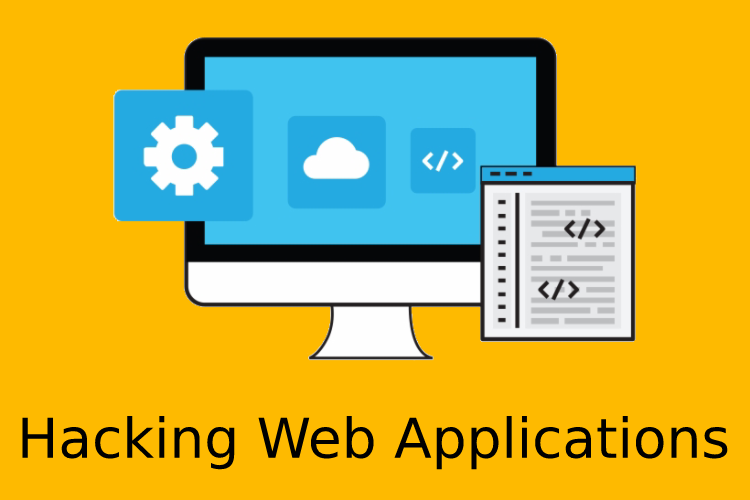
**CTF Challenge: Hacking Web Applications**

**Hacking Web Applications?**

****

"Hacking web applications" involves exploiting vulnerabilities within web applications to gain unauthorized access, manipulate data, or disrupt services. Here are some key points about hacking web applications:

### **Common Techniques for Hacking Web Applications**

1. **SQL Injection (SQLi)**: Injecting malicious SQL queries through input fields to manipulate the database and gain unauthorized access to data.
2. **Cross-Site Scripting (XSS)**: Injecting malicious scripts into web pages that are executed by the user's browser, leading to session hijacking, defacement, or redirecting users to malicious sites.
3. **Cross-Site Request Forgery (CSRF)**: Forcing a user to execute unwanted actions on a web application in which they are authenticated.
4. **Remote Code Execution (RCE)**: Executing arbitrary code on a server due to vulnerabilities in the web application.
5. **File Inclusion (LFI/RFI)**: Exploiting vulnerabilities to include and execute files on the server (Local File Inclusion and Remote File Inclusion).
6. **Directory Traversal**: Accessing files and directories outside the web root folder by manipulating URL paths.
7. **Command Injection**: Executing arbitrary commands on the server through vulnerable web applications.
8. **Authentication Bypass**: Exploiting flaws in authentication mechanisms to gain unauthorized access.
9. **Insecure Direct Object References (IDOR)**: Accessing resources without proper authorization by manipulating references to these resources.
10. **Session Hijacking**: Taking over a user session by stealing session cookies or using other methods to impersonate the user.

### 

### **Tools for Hacking Web Applications**



* **Burp Suite**: A comprehensive tool for web application security testing, including automated and manual vulnerability scanning.
* **OWASP ZAP (Zed Attack Proxy)**: An open-source web application security scanner.
* **SQLMap**: An automated tool for detecting and exploiting SQL injection flaws.
* **Nikto**: A web server scanner that checks for outdated versions, dangerous files, and configuration issues.
* **Metasploit**: A framework for developing, testing, and executing exploits against various systems.
* **Wfuzz**: A tool for brute-forcing web applications to find hidden resources and vulnerabilities.
* **Acunetix**: An automated web application security scanner that detects a wide range of vulnerabilities.
* **Nmap**: A network scanner that can identify open ports and services, often a precursor to web application testing.

# **Capture the Flag (CTF) Challenges**

**Flag 1. Which technique involves injecting malicious SQL queries to manipulate a database?**

Answer: SQL Injection

**Flag 2. What is the name of the tool used for web application security testing that includes automated and manual vulnerability scanning?**

Answer: Burp Suite

**Flag 3. Which method forces a user to execute unwanted actions on a web application in which they are authenticated?**

Answer: CSRF (Cross-Site Request Forgery)

**Flag 4. What vulnerability involves accessing files and directories outside the web root folder by manipulating URL paths?**

Answer: Directory Traversal

**Flag 5. Which tool is an open-source web application security scanner maintained by OWASP?**

Answer: ZAP (Zed Attack Proxy)