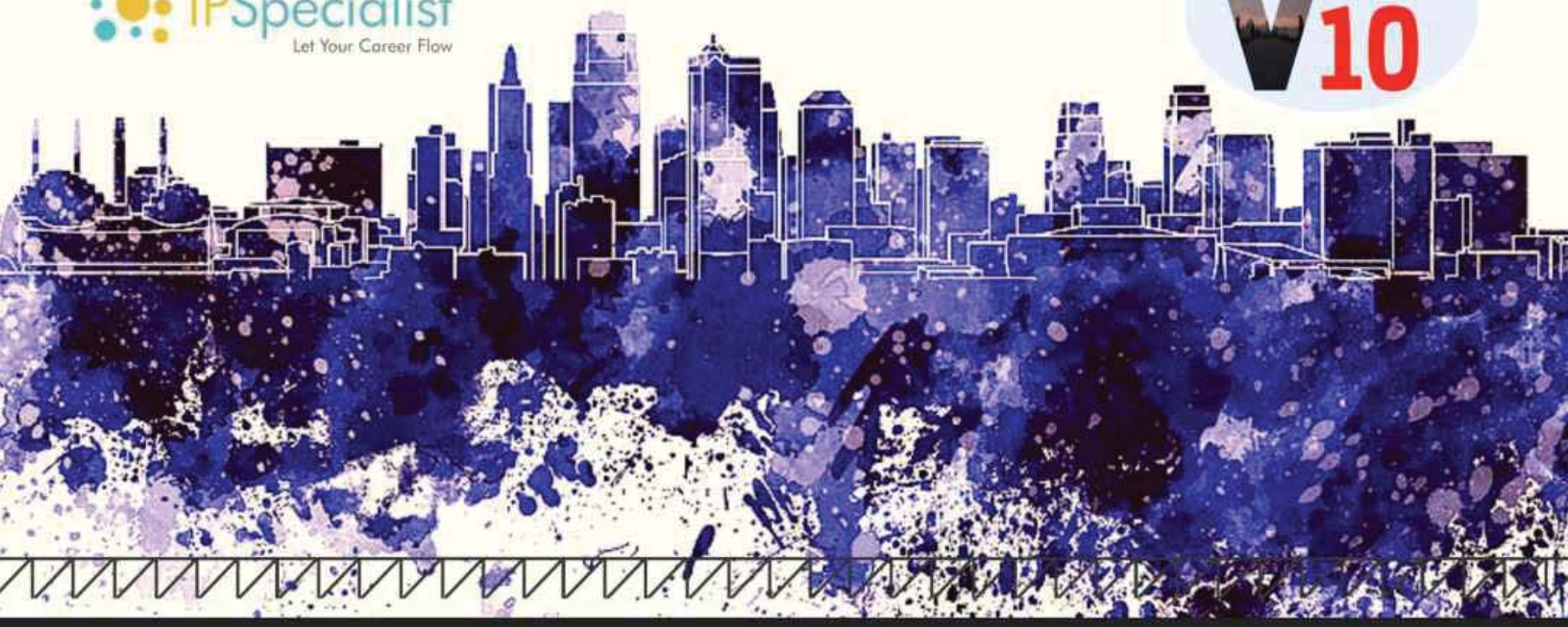
IPSpecialist
Let Your Career Flow

V10



# CEH V10 EC-COUNCIL CERTIFIED ETHICAL HACKER

## MOST DEMANDING COMPLETE HACKING GUIDE

**EXAM: 312-50**

C|EH
Certified Ethical Hacker

"To beat a hacker, you need to think like a hacker

**MOST ADVANCED HACKING COURSE**

www.ipspecialist.net

# Chapter 14: Hacking Web Applications

## Technology Brief

Significant increase in usage of Web application requires high availability and extreme performance of the application. In this modern era, the web application is popularly used in the corporate sector to perform important tasks as well as used globally for social purposes. It became a great challenge for the web server administrators and Application Server administrators to ensure security measures and eliminate vulnerabilities to provide high availability and smooth performance.

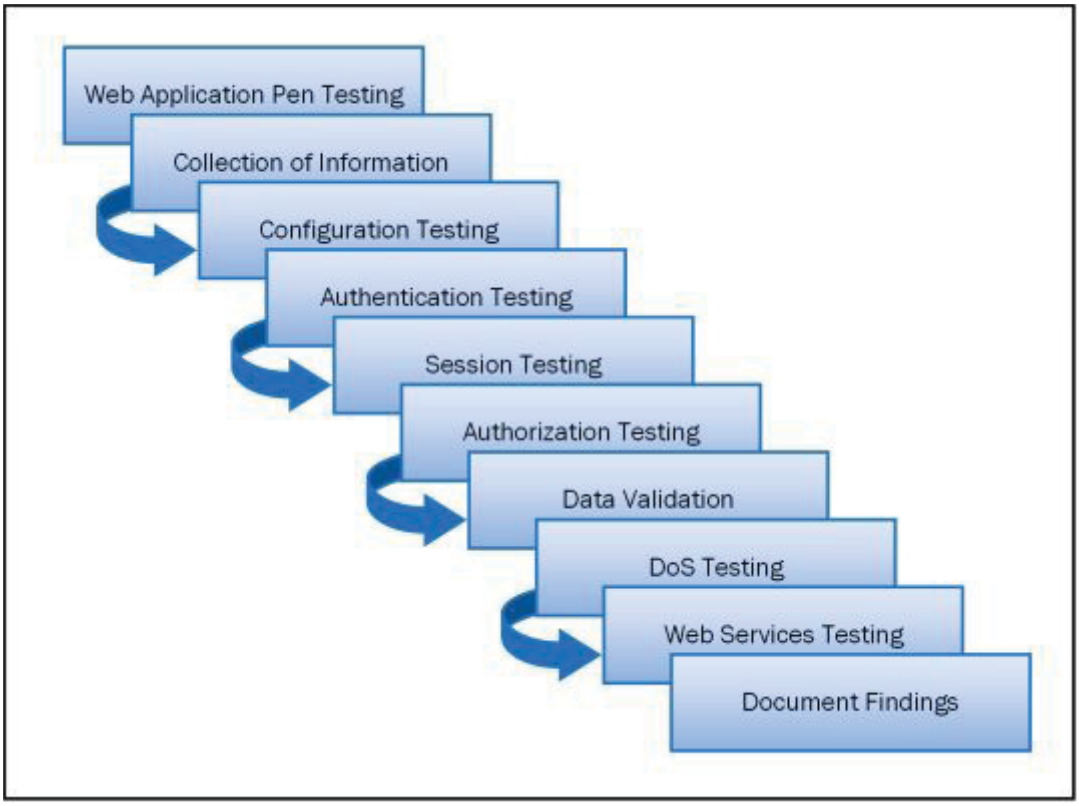



*Figure 14-01 Web Application Pentesting*

## Web Application Concepts

Web Applications are that application that is running on a remote application server and available for clients over the internet. These web applications can be available on different platforms such as Browser or Software to entertain the clients. Use of Web application has been incredibly increased in last few years. Web Application is basically depending upon Client-Server relationship. Web applications are basically providing an interface to the client to avail web services. Web pages may be generated on the server or containing scripting to be executed on the client web browser dynamically.

### Server Administrator

The server administrator is the one who took care of the web server in terms of safety, security, functioning, and performance. It is responsible for estimating security measures and deploying security models, finding and eliminating vulnerabilities.

### Application Administrator

Application Administrator is responsible for the management and configuration required for the web application. It ensures the availability and high performance of the web application.

### Client

Clients are those endpoints which interact with the web server or application server to avail the services offered by the server. These clients require a highly available service from the server at any time. While these clients are accessing the resources, they are using different web browsers which might be risky in terms of security.

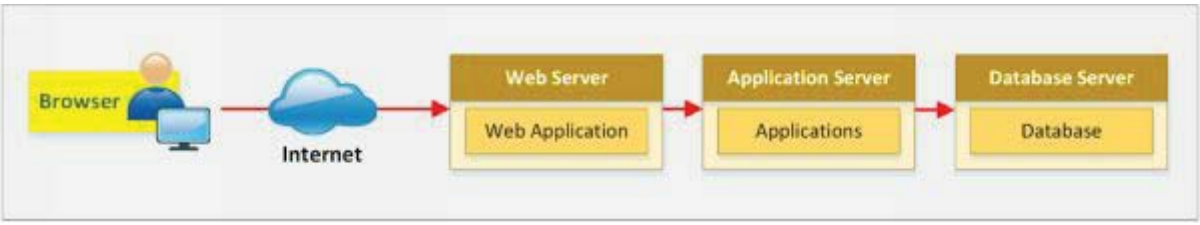



*Figure 14-02 Web Application Architecture*

### How do Web Applications works?

A Web Application functions in two steps, i.e., Front-end and Back-end. Users requests are handled by front-end where the user is interacting with the

web pages. Services are communicated to the user from the server through the button and other controls of the web page. All processing was controlled and processed on the back-end.

Server-side languages include: -

- Ruby on Rails
- PHP
- C#
- Java
- Python
- JavaScript

Client-side languages include: -

- CSS
- JavaScript
- HTML

The web application is basically working on the following layers: -

- ***Presentation Layer:*** Presentation Layer Responsible for displaying and presenting the information to the user on the client end.
- ***Logic Layer:*** Logic Layer Used to transform, query, edit, and otherwise manipulate information to and from the forms.
- ***Data Layer:*** Data Layer Responsible for holding the data and information for the application as a whole.

**Web 2.0**

Web 2.0 is the generation of world wide web websites that provide dynamic and flexible user interaction. It provides ease of use, interoperability between other products, systems, and devices. Web 2.0 allows the users to interact and collaborate with social platforms such as social media site and social networking sites. Prior generation, i.e., web 1.0 in which users are limited to passive viewing to static content. Web 2.0 offers almost all users the same freedom to contribute. the characteristics of Web 2.0 are rich user experience, user participation, dynamic content, metadata, Web standards, and scalability.

**Web App Threats**

The threat to Web Application are: -

- Cookie Poisoning

- Insecure Storage
- Information Leakage
- Directory Traversal
- Parameter/Form Tampering
- DOS Attack
- Buffer Overflow
- Log tampering
- SQL Injection
- Cross-Site (XSS)
- Cross-Site Request Forgery
- Security Misconfiguration
- Broken Session Management
- DMZ attack
- Session Hijacking
- Network Access Attacks

***Unvalidated Inputs***

Unvalidated Input refers to the processing of non-validated input from the client to the web application or backend servers. This is a vulnerability that can be exploited to perform XSS, buffer overflow, and injection attacks.

***Parameter / Form Tampering***

Parameter tampering refers to the attack in which parameters are manipulated while client and server are communicating with each other. Parameters such as Inform Resource Locator (URL) or web page form fields are modified. By this way, a user may either redirected to another website that may exactly look like the legitimate site or modifies the field such as cookies, form fields, HTTP Headers.

***Injection Flaws***

Injection attacks work with the support of web application vulnerabilities if a web application is vulnerable that it allows untrusted input to be executed. Malicious code injection, file injection or malicious SQL injection will result in the exploit. Injection flaws include the following:

- SQL Injection
- Command Injection
- LDAP Injection

***SQL Injection:***

SQL Injection is basically the injection of malicious SQL queries. Using SQL queries, unauthorized user interrupts the processes, manipulate the database and execute the commands and queries by injection results in data leakage or loss. These vulnerabilities can be detected by using application vulnerability scanners. SQL injection is often executed using address bar. Attacker bypasses the vulnerable application's security and extracts the valuable information from its database using SQL injection

***Command Injection:***

Command injection can be done by any of the following methods:

- Shell Injection
- File Injection
- HTML Embedding

***LDAP Injection***

LDAP injection is a technique that also takes advantage of non-validated input vulnerability. An attacker may access the database using LDAP filter to search the information.

***Denial-of-Service DoS Attack***

An attacker may perform a DoS attack in the following ways: -

1. **User Registration DoS**

   An attacker may automate the process to keep registering with fake accounts.

2. **Login DoS**

   Attacker attempt to send login requests repeatedly.

3. **User Enumeration**

   An attacker may attempt to try different username password combinations from a dictionary file.

4. **Account Lockout**

   An attacker is attempting to lock the legitimate account by attempting invalid passwords.

# Web App Hacking Methodology

## Analyze Web Applications

Analyzing Web application includes observing the functionality and other parameters to identify the vulnerabilities, entry points and server technologies that can be exploited. HTTP requests and HTTP fingerprinting techniques are used to diagnose these parameters.

## Attack Authentication Mechanism

By exploiting the authentication mechanism using different techniques, an attacker may bypass the authentication or steal information. Attacking on authentication mechanism includes: -

- Username Enumeration
- Cookie Exploitation
- Session Attacks
- Password Attacks

## Authorization Attack Schemes

Attacker by accessing the web application using low privilege account, escalate the privileges to access sensitive information. Different techniques are used such as URL, POST data, Query string, cookies, parameter tampering, HTTP header, etc. to escalate privileges.

## Session Management Attack

As defined earlier, Session management attack is performed by bypassing the authentication in order to impersonate a legitimate authorized user. This can be done using different session hijacking techniques such as: -

- Session Token Prediction
- Session Token Tampering
- Man-in-the-Middle Attack
- Session Replay

## Perform Injection Attacks

Injection attack is basically an injection of malicious code, commands, and file by exploiting the vulnerabilities in a web application. Injection attack may be performed in a different form such as: -

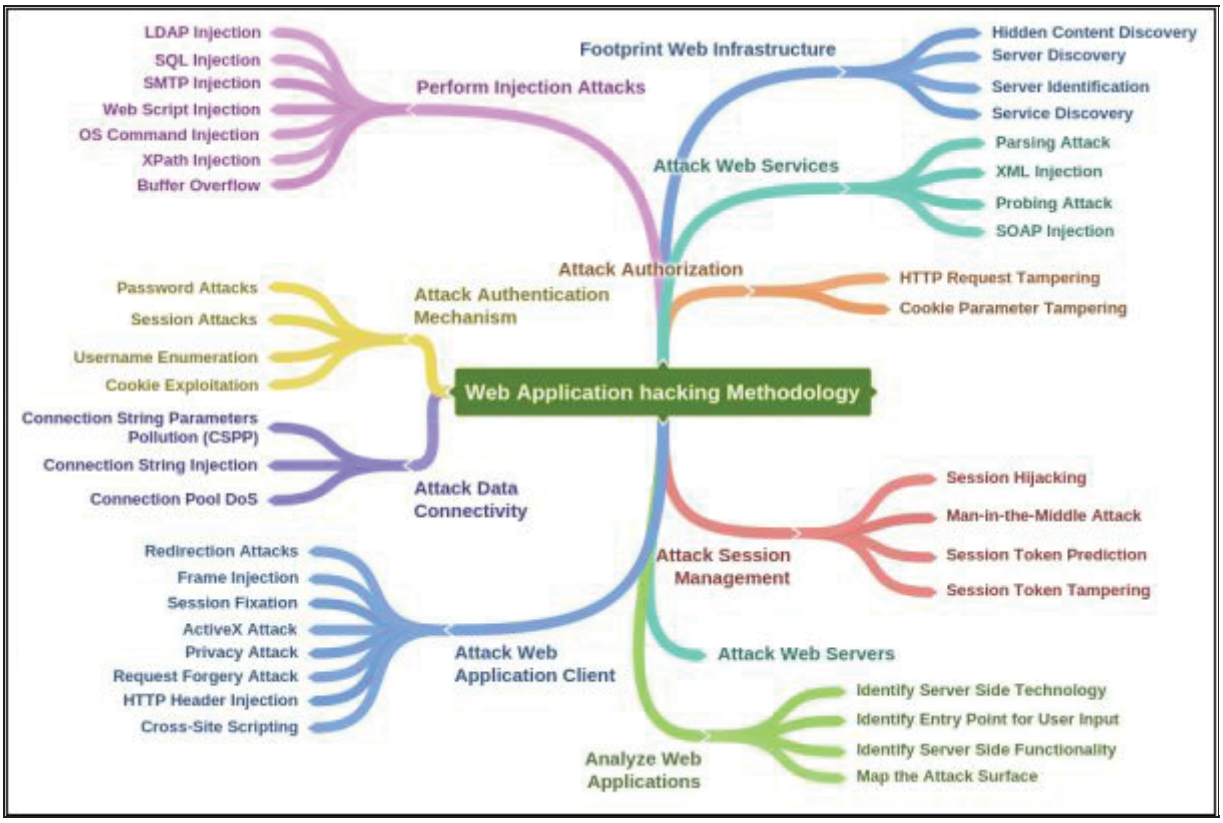
- Web Script Injection
- OS Command Injection
- SMTP Injection
- SQL Injection
- LDAP Injection
- XPath Injection
- Buffer Overflow
- Canonicalization

**Attack Data Connectivity**

Database connectivity attack is focused on exploiting the data connectivity between application and its database. Database connection requires connection string to initiate a connection to the database. Data connectivity attack includes: -

1. Connection String Injection
2. Connection String Parameters Pollution (CSPP)
3. Connection Pool DoS

**Mind Map**


Web Application hacking Methodology
Perform Injection Attacks
LDAP Injection
SQL Injection
SMTP Injection
Web Script Injection
OS Command Injection
XPath Injection
Buffer Overflow
Footprint Web Infrastructure
Hidden Content Discovery
Server Discovery
Server Identification
Service Discovery
Attack Web Services
Parsing Attack
XML Injection
Probing Attack
SOAP Injection
Attack Authorization
HTTP Request Tampering
Cookie Parameter Tampering
Attack Authentication Mechanism
Password Attacks
Session Attacks
Username Enumeration
Cookie Exploitation
Attack Data Connectivity
Connection String Parameters Pollution (CSPP)
Connection String Injection
Connection Pool DoS
Attack Session Management
Session Hijacking
Man-in-the-Middle Attack
Session Token Prediction
Session Token Tampering
Attack Web Application Client
Redirection Attacks
Frame Injection
Session Fixation
ActiveX Attack
Privacy Attack
Request Forgery Attack
HTTP Header Injection
Cross-Site Scripting
Attack Web Servers
Analyze Web Applications
Identify Server Side Technology
Identify Entry Point for User Input
Identify Server Side Functionality
Map the Attack Surface

## Countermeasures

### Encoding Schemes

Web Applications uses different encoding schemes for securing their data. These encoding schemes are categorized into the two categories.

***URL Encoding***

URL Encoding is the encoding technique for secure handling of URL. In URL Encoding, URL is converted into an ASCII Format for secure transportation over HTTP. Unusual ASCII characters are replaced by ASCII code, a "%" followed by two hexadecimal digits. The default character-set in HTML5 is UTF-8. Following chart is showing some symbols and their codes.

| Character | From Windows-1252 | From UTF-8 |
|---|---|---|
| space | %20 | %20 |
| ! | %21 | %21 |
| " | %22 | %22 |
| # | %23 | %23 |
| $ | %24 | %24 |
| % | %25 | %25 |
| & | %26 | %26 |

*Table 14-01 Encoding Schemes*

***HTML Encoding***

Similar to URL Encoding, HTML encoding is a technique to represent unusual characters with an HTML code. ASCII was the first character encoding standard which supports 128 different alphanumeric characters. Other techniques such as ANSI and ISO-8859-1 support 256, UTF-8 (Unicode) covers almost every character and Symbol.

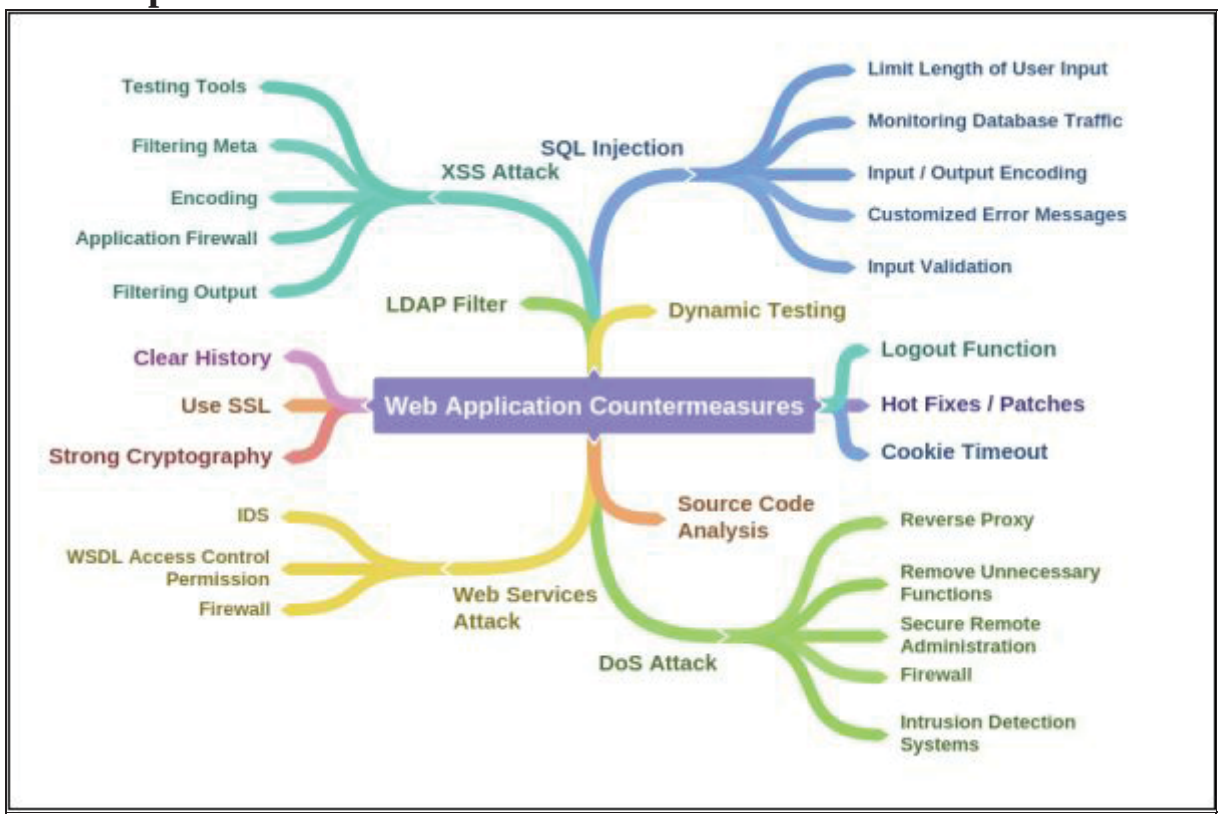
**For HTML4:**

```
<meta http-equiv="Content-Type" content="text/html;charset=ISO-8859-1">
```

**For HTML5:**

```
<meta charset="UTF-8">
```

## Mind Map


Web Application Countermeasures
Testing Tools
Filtering Meta
Encoding
Application Firewall
Filtering Output
XSS Attack
SQL Injection
Limit Length of User Input
Monitoring Database Traffic
Input / Output Encoding
Customized Error Messages
Input Validation
LDAP Filter
Dynamic Testing
Clear History
Use SSL
Strong Cryptography
Logout Function
Hot Fixes / Patches
Cookie Timeout
IDS
WSDL Access Control Permission
Firewall
Web Services Attack
Source Code Analysis
DoS Attack
Reverse Proxy
Remove Unnecessary Functions
Secure Remote Administration
Firewall
Intrusion Detection Systems