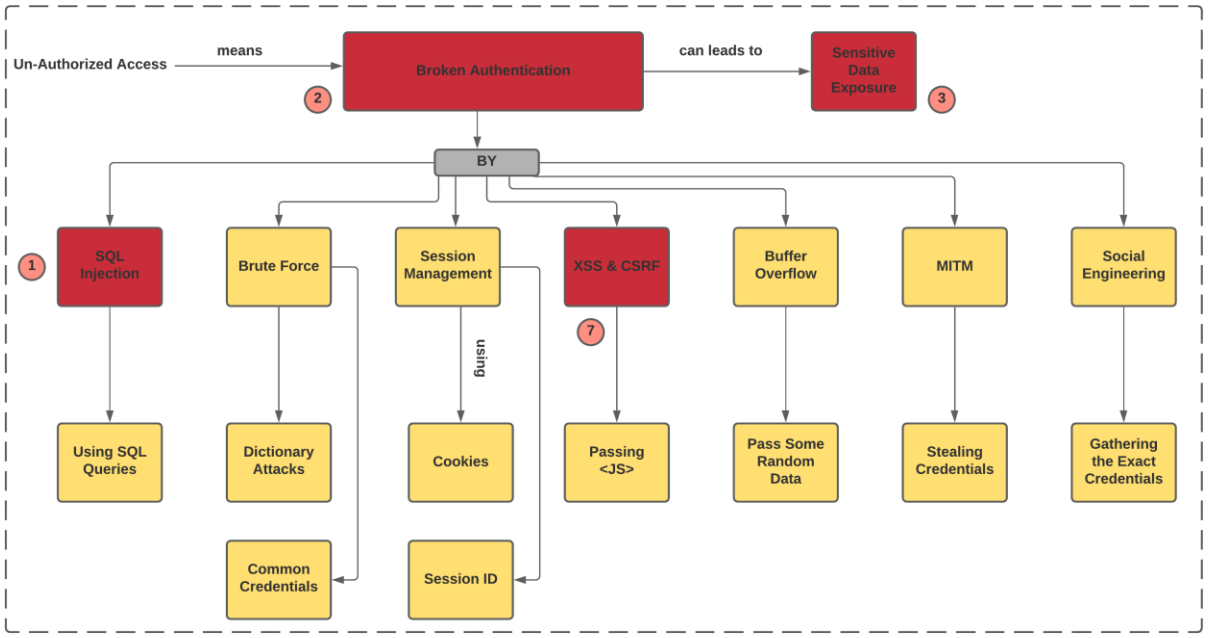
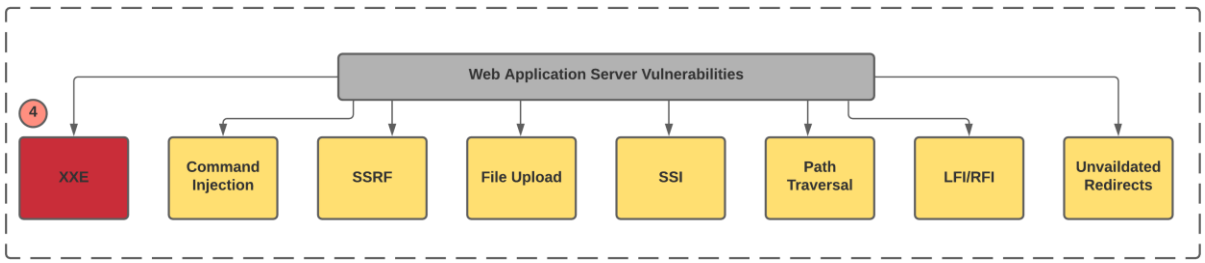
Web Application Security

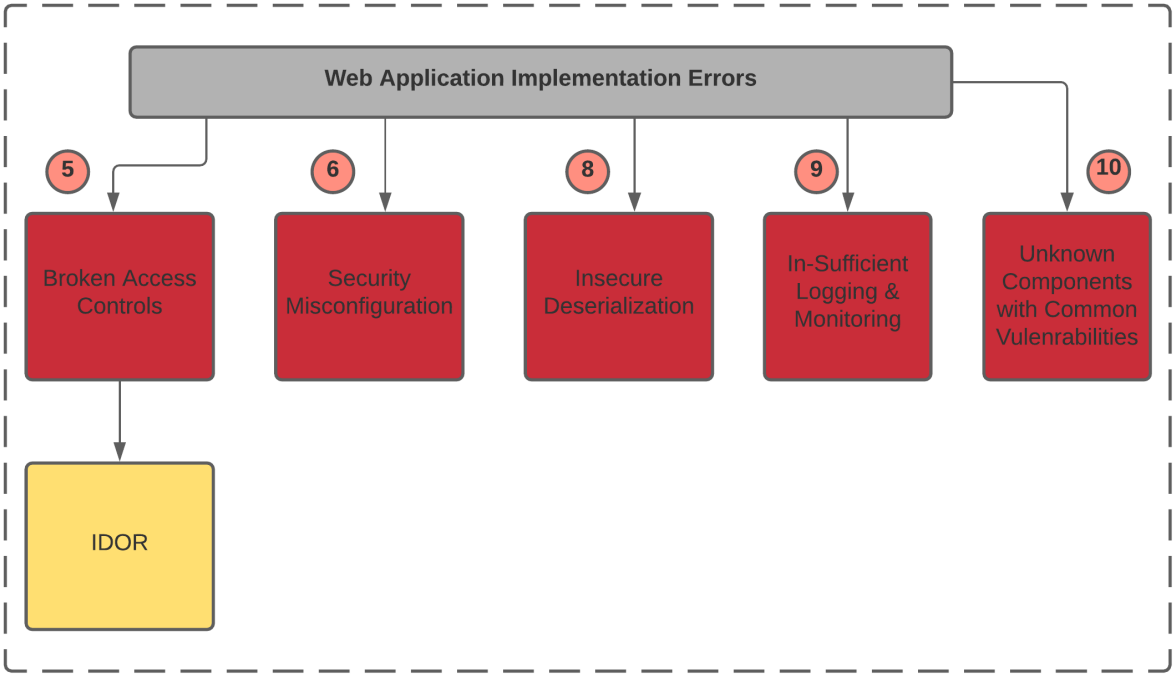
OWASP TOP 10

To ensure security, there should be no un-authorized access and OWASP all top 10 attacks and vulnerabilities ensure this thing.

Following is the flow of operations showing how and where OWASP is responsible:







Abbreviations:

XXE: XLM External Entity

SSRF: Server-Side Request Forgery

SSI: Server-Side Include

RFI: Remote File Inclusion

LFI: Local File Inclusion

IDOR: Insecure direct object references

Tools Based on Vulnerabilities

1 SQL Injection

A SQL injection attack consists of insertion or “injection” of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system.

1.1 TOOLS

#	Tools	Platform
1	SQLMap	Python
2	SQLninja	Perl – implementation by C

2 Brute Force

Brute-force attacks are often used for attacking authentication and discovering hidden content/pages within a web application. These attacks are usually sent via GET and POST requests to the server.

2.1 TOOLS

#	Tools	Platform
1	John The Ripper	Shell
2	Hydra	C

3 Session Management

The only way to maintain a session is when some unique information about session (session_id, cookies) is passed between server & client in every requests and response. The session management may be due to following causes:

- User Authentication
- HTML Hidden Field
- Cookies
- URL Rewriting
- Session Management API

Session management is a complete process of involving the above factors however, tools like:

BurpSuite, OpenVas, Zap etc. can be used to check for vulnerbility. Moreover, as it is a complete process of involving many factors which is why have to go one by one from the above.

4 XSS + CSRF

Cross-Site Scripting (XSS) attacks occur when:

1. Data enters a Web application through an untrusted source, most frequently a web request.
2. The data is included in dynamic content that is sent to a web user without being validated for malicious content.

The malicious content sent to the web browser often takes the form of a segment of JavaScript, but may also include HTML, Flash, or any other type of code that the browser may execute. The variety of attacks based on XSS is almost limitless, but they commonly include transmitting private data, like cookies or other session information, to the attacker, redirecting the victim to web content controlled by the attacker, or performing other malicious operations on the user's machine under the guise of the vulnerable site.

4.1 TOOLS

#	Tools	Platform
1	Zap	Java
2	Wfuzz	Python
3	Wapiti	Python
4	Arachini	Ruby

5 Buffer Overflows

Buffer overflow errors are characterized by the overwriting of memory fragments of the process, which should have never been modified intentionally or unintentionally. Overwriting values of the IP (Instruction Pointer), BP (Base Pointer) and other registers causes exceptions, segmentation faults, and other errors to occur.

- To check or generate this error, passing random values in the code or during execution.
- To prevent the issue, it is required to automate the process of development and deploying security patches.

There is no specific tools but a process as ***fuzz*** testing for that ***SFuzz*** can be used based on several directories.

6 MITM

A man-in-the-middle (MITM) attack is when an attacker intercepts communications between two parties either to secretly eavesdrop or modify traffic traveling between the two. Attackers might use MitM attacks to steal login credentials or personal information, spy on the victim, or sabotage communications or corrupt data.

6.1 TOOLS

#	Tools	Platform
1	Arpspoof	Python
2	TcpDump	C
3	Ettercap	C
4	Wireshark	C
5	Dsniff	Package of tools (c, shell, python)
6	Cain & Abel	Python

7 Social Engineering

Social Engineer Toolkit is an open source tool to perform online social engineering attacks. The tool can be used for various attack scenarios including spear phishing and website attack vectors. Social Engineer Toolkit works in an integrated manner with Metasploit. It enables the execution of client-side attacks and seamless harvesting of credentials. With Social Engineer Toolkit, one can backdoor an executable and send it to the victim.

7.1 TOOLS

#	Tools	Platform
1	Metasploit	Ruby
2	OWASP Maryam	Python
3	Search Engines	Shodan, Google, Fofa
4	theHarvester	Python
5	Maltego	Java
6	FOCA	.net, c++, sql server

8 Security Misconfiguration

Security misconfiguration happens when the responsible party fails to follow best practices when configuring an asset. This asset can be an operating system, a web server, software running on a machine, etc. Security misconfiguration don’t affect web assets only. Any component which

requires a configuration is subject to this vulnerability. This means that network devices, hardware, email services, etc. can suffer from this vulnerability. Following are some of the causes:

- Unnecessary services.
- Using default accounts.
- Using default configuration.

There are automated scanners like Zap, BurpSuite and OpenVas which can be used to find out vulnerabilities in this area.

9 Broken Access Control

Exploitation of access control is a core skill of attackers. Access control is detectable using manual means, or possibly through automation for the absence of access controls in certain frameworks. The vulnerabilities could be in the form of IDOR (Insecure direct object references) & CSRF etc. These can be check using BurpSuite (community edition) and TestIDOR (github). However, there two formats of testing broken access control as DAST (Dynamic Application Security Testing) and SAST (Source Code Analysis Tools).

9.1 TOOLS for SAST

#	Tools	Platform
1	Bandit	Python
2	Brokenman	Ruby

9.2 TOOLS for DAST

#	Tools	Platform
1	Arachini	Ruby
2	Grabber	Python
3	Wapiti	Python

10 Insufficient Logging & Monitoring

Exploitation of insufficient logging and monitoring is the bedrock of nearly every major incident. Attackers rely on the lack of monitoring and timely response to achieve their goals without being detected.

To prevent form this scenario:

- Testers action should be recorder.
- Examine the logs.

11 Insecure De-serialization

According to OWASP, applications and APIs will be vulnerable if they de-serialize hostile or tampered objects supplied by an attacker. This form of vulnerability is mostly found in the code as in Java or php code. This can result in two primary types of attacks:

- Object and data structure related attacks where the attacker modifies application logic or achieves arbitrary remote code execution if there are classes available to the application that can change behaviour during or after de-serialization.
- Typical data tampering attacks, such as access-control-related attacks, where existing data structures are used but the content is changed.

The tools which can used to test are BurpSuite, Zap, OpenVas etc.

Summarizing the major causes as:

- PHP serialization format.
- Java serialization format.
- Modification of the object attributes.
- Modifying data types.

12 Components with Known-Vulnerabilities

Known vulnerabilities are vulnerabilities that were discovered in open source components. To prevent from this issue, following are some preventive measures or daily practices:

- ***Remove:*** useless dependencies, unnecessary features, components, files and documentation.
- Continuously inventory versions of both client-side & server0side components. ***Using tools: versions, DependencyCheck, retire.js.***
- Continuously monitor resources like CVE & NVD for vulnerabilities in the components.
- Monitor the libraries components & do not create security patches for the older versions.

Standard Operating Procedure

It is observed from the research that one of the main things that are required is a well-defined standard operating procedure which would be very helpful to show users a road map by which they could perform a penetration test. In the following section, a standard operating procedure was designed that help in performing a penetration test, comprised of the six steps and their description is also given. The list of six steps are given below:

- Information gathering
- Scanning
- Enumeration
- Vulnerability identification
- Exploitation
- Reporting

1. Information gathering

Information about the target will be provided in this section which will provide a standard road map to the penetration tester to perform test on the target. We will use this information in out next coming steps which will help in proceeding our attacks on the target and hence gathering of information in this step is very useful.

The two types of information gathering are called active and passive. In passive, the target is unaware of the information being acquired from him but in the active information gathering if the target has adopted security measures than the information being gathered from the target is also known to him.

Table 1 Exploring Internet

Input	Name of the target.
Utilities	Searching on engines like Google, Yahoo, Bing, Duckduckgoo.
Actions	1 Type the name of the target in the search bar and get information about it. 2 Perform analysis of the target's site and try to gather as much information as could from it.
Outcome	The data or material of any type regarding the target on the internet including its location, email, phone numbers, fax also including information on its hardware or software the company is using.

Table 2 Whois

Input	IP of the target.
Utilities	1 Terminal 2 Whois (Whois, n.d.)
Actions	1 Run the terminal. 2 Enter command: "whois <target-ip>". 3 Save outcome in the file.
Outcome	The data including emails, phone numbers, fax, admin details, server details and any sort of administrative or technical details regarding target.

Table 3 Search for DNS

Input	IP of the target.
Utilities	1 Terminal 2 Dig (nixCraft, 2020)
Actions	1 Run the terminal.

	2 Enter command: “dig <target-ip>”. 3 Save outcome in the file.
Outcome	The details of the nameservers (DNS), hos address, mail exchange server or any useful information.

Table 4 Search for Sub-domain

Input	URL of the target.
Utilities	GoBuster (Reeves, 2020).
Actions	1 Using the command apt-get install gobuster, install it on the Linux machine. 2 Enter command: “gobuster dir -u <target-url> -w <wordlist>”. 3 Save outcome in the file.
Outcome	The details regarding IP, web server, admin, and domain being used by them and the OS.

Table 5 Active OS fingerprinting

Input	Target’s IP address.
Utilities	1 Terminal 2 Xprobe2 (OCCUPYTHEWEB, 2013).
Actions	1 Run the terminal. 2 Enter the command: “xprobe2 <domain OR target-ip>”. 3 Save outcome in the file.
Outcome	The details regarding OS and its version number running on the system.

2. Scanning

The services running on the target machine could be detected during the scanning process, also the port running them and if there is any port or ports which are open and not using any service, which could be exploited. Moreover, the version numbers of the services could be detected. Also, if possible, the server information and its version number could be founded. These results obtained are then documented and thus vulnerabilities that are found are exploited afterwards.

Table 6 Searching for live servers and services using Nmap

Input	IP of the target.
Utilities	1 Terminal 2 Nmap (Lyon, 1997).
Actions	1 Run the terminal. 2 Enter command on terminal: “nmap <parameter> <target-ip>”. 3 Save outcome in the file.
Outcome	The details of the open ports, services running on them, state of the ports whether open or not also about the OS running.

Table 7 Searching for live servers and services using Nikto

Input	IP of the target.
Utilities	1 Terminal 2 Nikto (Sullo, 2014).
Actions	1 Run the terminal. 2 Enter the command: “Nikto -host <target-ip>”. 3 Save outcome in the file.
Outcome	The details of the open ports, services running on them, state of the ports whether open or not also about the allowed HTTP methods.

Table 8 Banner grabbing

Input	IP and port number of the target.
Utilities	1 Terminal 2 Telnet (Digital Guide IONOS, 2019).
Actions	1 Run the terminal. 2 Enter the command: “telnet <target>” or “telnet <ip-address> <port-number>” from step 1.

	3 Type the HEAD HTTP request. 4 Save outcome in the file.
Outcome	The server's details including its version which is very useful in finding vulnerabilities in it which afterwards could be exploited.

3. Enumeration

Under this section, the gathered information is organized in a methodical manner, resulting in an organized structure that is easy to understand the working of the target system and knowing vulnerabilities which later are exploited.

Table 9 Data compilation

Input	The data which is obtained from the previous steps are used as an input.
Utilities	Microsoft Excel.
Actions	1 Run excel sheet. 2 Insert a table with columns having details of all the servers found and then fill them with the following details: • Server's IP. • Name of the server's domain. • List of the open ports. • Ports running services on them. • Details about the OS of the server. 3 Save outcome in the file.
Outcome	This information is very valuable in making the pen tester to plan the attacks and to find vulnerabilities on it with the above information.

4. Vulnerability analysis

Vulnerabilities are exploited using automated and manual vulnerability methods by initiating suitable attacks on the target.

Table 10 Searching know vulnerabilities

Input	Keywords related to target.
Utilities	1 Web browser 2 Databases: • CVE Details. • CERT.
Actions	1 Enter https://www.cvedetails.com/ on the browser. 2 With help of keywords regarding target find information related to servers, services and OS used by them. 3 Save outcome in the file. 4 Search for www.kb.cert.org/vuls on the browser. 5 Enter keywords that relates to the target's services, server and OS. 6 Save outcome in the file.
Outcome	It is expected of getting server's information related to the target's version number. This information later is used to exploit known vulnerabilities.

Table 11 OpenVAS

Input	IP of the target.
Utilities	1 Web browser 2 OpenVAS (OpenVAS, 2005).
Actions	1 Run the terminal. 2 Open OpenVAS software. 3 Enter https://127.0.0.1:9392 in the browser to open the access page for OpenVAS. 4 Put the credentials. 5 Enter target's IP to run the scan. 6 Details of the vulnerabilities would be generated as a report and can be downloaded easily.
Outcom	The report on the vulnerabilities is generated but these results should be verified

e	as automated results are not completely reliable, so the report need to be investigated to remove any false results.

Table 12 Manual vulnerability analysis

Input	Results of the file from the enumeration phase.
Utilities	Excel sheet.
Actions	1 Search for the vulnerabilities online and their exploits related to the ports, services, and the OS running on the target. 2 Save outcome in the file.
Outcome	The vulnerabilities which are out of the scope of the automated tools are added in this part as automated tools are not fully trusted.

Table 13 Reviewing the vulnerabilities

Input	Report on the vulnerabilities detected.
Utilities	Excel sheet.
Actions	1 Analyze the detected vulnerabilities list. 2 Give them score on the basis of their severity and impact. 3 Categorize according to the score.
Outcome	The final list is based on the basis of scores given to vulnerabilities.

5. Exploitation

The vulnerabilities list obtained in the last section is very useful in executing several attacks on the targeted machine. These attacks could alter the services or to maneuver or attain unwarranted access to the data stored on targeted machine. The following tools are used to conduct the pen test and their procedure is written as follow:

Table 14 Organization

Input	The list from the last section.
Utilities	Excel sheet.
Actions	1 Establish targets for each vulnerability. 2 Resources are assigned to the vulnerabilities and time is allocated to each of the resources.
Outcome	The timeline which will guide the penetration testing and also shows which resource is tackling with what type of problem.

Table 15 Metasploit

Input	Use the file generated by the automated software i.e. XML.
Utilities	Metasploit (Metasploit.com)
Actions	1 Run Metasploit. 2 The downloaded XML file is loaded/imported. 3 Initialize the parameters to set up exploits for the vulnerabilities found. 4 Run the exploits.
Outcome	If the exploits are implemented successfully, the penetration test is successful. Afterwards, write the mitigation report and give recommendations and if unsuccessful try the new exploit.

Table 16 Directory scan

Input	IP and port number of the target.
Utilities	Dirbuster (OWASP, n.d.).
Actions	1 Search Dirbuster and then run it. 2 In the target's URL filed, enter IP along with the port number of the target..

	3 Select the world list from usr/share/Dirbuster folder. 4 Set the number of threads. 5 Initiate the scan by clicking start. 6 Save outcome in the file.
Outcome	The results would be the list of directories that will help to discover further exploits on the targeted machine.

Table 17 SQLi attack

Input	URL of the target.
Utilities	Sqlmap (sqlmap, n.d.).
Actions	1 Run the terminal. 2 Run the Sqlmap on the system and if it is not installed than install by entering command: “https://github.com/sqlmapproject/sqlmap.git”. 3 Run Sqlmap. 4 Enter command to get details regarding databases: “Sqlmap -u <target-url> --dbs”. 5 Enter command for getting details of the tables: “Sqlmap -u <target-url> --tables”.
Outcome	The successful exploit will reveal the confidential information from the database. Provide recommendations for its mitigation in the report otherwise run new exploits.

Table 18 Brute force attack

Input	IP of the target and passwords list.
Utilities	Medusa (Rubens, n.d.).
Actions	1 Run the terminal. 2 Search and download common list of used passwords from the web search. 3 Enter the command: “medusa -h <users-list OR target-ip> -u <user-name> -P <wordlist> -n <port-number>.
Outcome	The password is obtained for the user if the exploit is successful. Afterwards, it should be mentioned in the report with providing recommendations.

Table 19 Denial of service

Input	IP of the target.
Utilities	LOIC (Oliveira, 2019) or slowloris (Yaltirakli, 2020).
Actions	1 Download LOIC or slowloris form GitHub. 2 Run terminal in folder containing LOIC or slowloris. 3 For running LOIC type and enter ./loic.sh and for slowloris type "./slowloris.py <target-ip>". 4 The slowloris results will start coming and if using LOIC, enter the IP or the URL in the specified bar and then press the button named: "lock on". 5 Press the button named: "IMMA CHARGIN MAH LAZER" to run the exploit.
Outcome	The flooding will likely be started and will flood the services if appropriate security measures are not taken. Mention in the report and provide recommendations and if unsuccessful try another exploit.

Table 20 Man-in-the-middle

Input	IP of the target.
Utilities	1 Arpspoof. 2 Tcpdump (Hat), 2018).
Actions	1 Run the terminal. 2 Install the arpspoof if not already installed by entering the command: "apt-get install dsniff". 3 Execute "arpspoof [-I *interface*] [-t *target's ip*] host". 4 Launch second terminal and execute "arpspoof [-I *interface*] [-t router's ip] host". 5 Enter command "apt-get install tcpdump" to install tcpdump if not already installed. 6 Make sure that the IP forwarding is not disabled, open third terminal, and execute: "echo 1 > /proc/sys/net/ipv4/ip_forward". 7 Now execute: "tcpdump [-I *interface*] [-n *port*] [*target's* port] *and* host [target's ip]. 8 Capturing traffic will begin now.

Outcome	If the above-mentioned procedure is executed successfully then the flow of traffic from the target can be seen which means man-in-the-middle is successful. Afterwards mention it in the report and provide recommendations to mitigate it.

Table 21 Fuzzing

Input	IP of the target.
Utilities	Sfuzz (Conole, n.d.).
Actions	1 Run the terminal. 2 If not installed, then type and enter: “sudo apt-get install sfuzz”. 3 Go into the directory with the list of fuzzing scripts by entering “cd /usr/share/sfuzz-db”. 4 To execute the script, type: “sfuzz -S [target’s ip] [-p *port*] -T [-f *config-file*] [-L *output-file*] and press enter. 5 Analyze the results.
Outcome	To check for the memory leaks and any vulnerability that could be exploited, the target is tested by giving random scripts. If it is showing errors for such commands, it means the system is secure otherwise the founded vulnerability should be mentioned along with the mitigation strategy in the report.

6. Post exploitation

The post exploitation is based on the report providing a detailed analysis of the penetration testing activities. It is explained that how the vulnerabilities are found and how they are exploited. The consequences of these vulnerabilities are also explained in the report. At the end, recommendations are provided for mitigating or removing the risk to make the system secure.