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.