**Web Application** **Vulnerability Assessment**

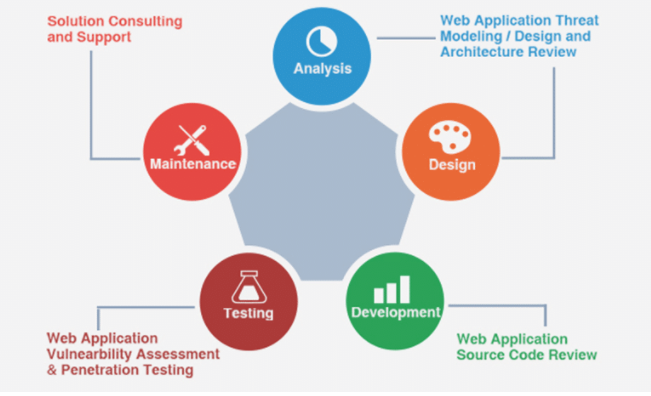
**Web application assessments** are performed to identify potential or realized weaknesses as a result of inadvertent mis-configuration, weak authentication, insufficient error handling, sensitive information leakage, etc. With a **Web Application** Vulnerability **Assessment** or **Application** Penetrating Testing, you will understand your corporate security posture and receive actionable recommendations on how to perform remediation of the vulnerabilities discovered in your environment, including potentially required patches, code changes, access.



**Why to perform Web Application Vulnerability Assessment**

**Web application security assessment** combines information security best practices and technologies specifically designed to test websites, web-based services, and web applications.Web application security assessment can be performed manually or automatically, and should continue throughout the software development lifecycle (SDLC).

* It will typically include safety protocols, security checks, and regular assessments, as well as safe coding practices, secure firewalls, vulnerability testing, and the installation of protocols that will ensure safe operation.
* Identifying Vulnerabilities and Impact. Identify vulnerabilities and the potential impact at the infrastructure, **application** and operational levels using testing standards such as OWASP, CIS, SANS or NIST.
* Determine Business. Risks.



**14 best open-source web application vulnerability scanners**

A web application security scanner is a software program which performs automatic black-box testing on a web application and identifies security vulnerabilities. Scanners do not access the source code; they only perform functional testing and try to find security vulnerabilities. Various paid and free web application vulnerability scanners are available.

These are the best open-source web application penetration testing tools.

## 1. Grabber

Grabber is a web application scanner which can detect many security vulnerabilities in web applications. It performs scans and tells where the vulnerability exists.

## 2. Vega

Vega is another free open-source web vulnerability scanner and testing platform. With this tool, you can perform security testing of a web application. It can be used to find SQL injection, header injection, directory listing, shell injection, cross-site scripting, file inclusion and other web application vulnerabilities. This tool can also be extended using a powerful API written in JavaScript.

## 3. Zed Attack Proxy

Zed Attack Proxy is also known as ZAP. This tool is open-source and is developed by OWASP. It is available for Windows, Unix/Linux and Macintosh platforms.

## 4. Wapiti

Wapiti is a web vulnerability scanner which lets you audit the security of your web applications. It performs black-box testing by scanning web pages and injecting data. It tries to inject payloads and see if a script is vulnerable. It supports both GET and POSTHTTP attacks and detects multiple vulnerabilities.

[GET PRICING](https://www.infosecinstitute.com/form/boot-camp-pricing/?utm_source=resources&utm_medium=infosec%20network&utm_campaign=boot%20camp%20discount&utm_content=sidebar%20bottom)

## 5. W3af

W3af is a popular web application attack and audit framework. This framework aims to provide a better web application penetration testing platform. It was developed using Python. It comes with a graphical and console interface.

## 6. WebScarab

WebScarab is a Java-based security framework for analyzing web applications using HTTP or HTTPS protocol. With available plugins, you can extend the functionality of the tool.

## 7. Skipfish

Skipfish is another nice web application security tool. It crawls the website and then checks each page for various security threats. At the end, it prepares the final report.

## 8. Ratproxy

Ratproxy is an open-source web application security audit tool which can be used to find security vulnerabilities in web applications. This tool is designed to overcome the problems users usually face while using other proxy tools for security audits.

## 9. SQLMap

SQLMap is another popular open-source penetration testing tool. It automates the process of finding and exploiting SQL injection vulnerabilities in a website’s database. It has a powerful detection engine and many useful features. This way, a penetration tester can easily perform an SQL injection check on a website.

## 10. Wfuzz

Wfuzz is another freely available open-source tool for web application penetration testing. It can be used to brute-force GET and POST parameters for testing against various kinds of injections like SQL, XSS, LDAP and many others.

## 11. Grendel-Scan

Grendel-Scan is another nice open-source web application security tool. This is an automatic tool for finding security vulnerabilities in web applications. Many features are also available for manual penetration testing.

## 12. Watcher

Watcher is a passive web security scanner. It does not attack with loads of requests or crawl the target website

## 13. X5S

X5S is also a Fiddler add-on intended to provide a way to find cross-site scripting vulnerabilities. This is not an automatic tool, so you need to understand how encoding issues can lead to XSS before using it.

## 14. Arachni

Arachni is an open-source tool developed for providing a penetration testing environment. This tool can detect various web application security vulnerabilities. It can detect various vulnerabilities like SQL injection, XSS, local file inclusion, remote file inclusion, unvalidated redirect and many others.

**Steps to web application security testing with documented results will help keep your organization's applications free of flaws:**

1. What needs to be tested? ...
2. What tools are best suited for the task? ...
3. Vulnerability scanning. ...
4. Scanner validation and additional manual checks.



**Application Security Audit Checklist**

* Make sure the **application's** authentication system is up-to-date.
* Create model of **application**.
* Approval: **Application** model
* Restrict access to **application** directories and files.
* Implement session expiration timeout.
* Forbid multiple concurrent sessions.
* Provide least privilege to **application** users.