[Skip to main content](https://lms.alnafi.com/xblock/block-v1:alnafi+DCCS102+2025_DCCS+type@vertical+block@facfe1bac0c440fe908d7888af537580?exam_access=&recheck_access=1&show_bookmark=0&show_title=0&view=student_view#main)

ref <https://hackr.io/blog/top-10-open-source-security-testing-tools-for-web-applications>

**Top 10 Open Source Security Testing Tools for Web Applications**

 Posted in  [Information Security & Ethical Hacking](https://hackr.io/blog/category/information-security-ethical-hacking)

       [Youssef Nader](https://hackr.io/blog/author/youssefnader)

Last Updated 23 Feb, 2021

Table of Contents

The Internet has grown, but so have hacking activities. Every now and then  there is some news regarding a website being hacked or a [data breach](https://en.wikipedia.org/wiki/Data_breach). Technology has come a long way, but so does hacking. Just like the  digital world, hacking techniques and tools have also become more  sophisticated and also threatening.

Better late than sorry! It’s  important to keep your website or web applications foolproof against  malicious activities. What you need to do is to use some security  testing tools to identify and measure the extent of security issues with your web application(s).

The primary function of security  testing is to perform functional testing of a web application under  observance and find as many security issues as possible that could  potentially lead to hacking. All of this is done without the need to  access the source code.

Before delving into some of the best  open-source security testing tools to test your web application, let’s  first acquaint ourselves with definition, intent, and need for security  testing.

**Security Testing**

**The Definition** – In  order to assure that data within some information system stays secure  and not accessible by unapproved users, we use security testing.  Successful security testing protects web applications against severe  malware and other malicious threats that might lead it to crash or give  out unexpected behavior.

Security testing helps in figuring  out various loopholes and flaws of a web application in the initial  stage. Furthermore, it also helps in testing whether an application has  successfully encoded security code or not. Primary areas covered by  security testing are:

* Authentication
* Authorization
* Availability
* Confidentiality
* Integrity
* Non-repudiation

**The Intent** –  Security testing is used by organizations and professionals throughout  the world to ensure their web applications and information systems  remain secure. Chief purposes of deploying security testing are:

* To help improve the security and shelf-life of a product
* To identify as well as fix various security issues in the initial stage of development
* To rate the stability in the present state

**The Need** – Why do we need security testing? Well, there are a number of reasons, ranging  from analyzing the degree of security to the prevention of unexpected  breakdowns in the future. Some of the most important reasons are:

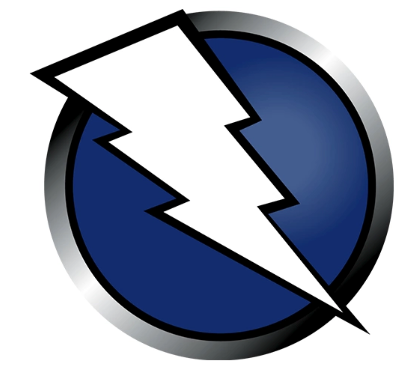
* Avert inconsistent performance
* Avoid losing customer trust
* Avoid losing important information in the form of security leaks
* Prevent information theft by unidentified users
* Save from unexpected breakdown
* Save additional costs required for fixing security issues

There are several free, paid, and  open-source tools available to check the vulnerabilities and flaws in  your web applications. The best thing about open-source tools, besides  being free, is that you can customize them to match your specific  requirements.

So, here is the list of 11 open source security testing tools for checking how secure your website or web application is:

**Top 10 Open Source Security Testing Tools**

**1. Zed Attack Proxy (ZAP)**

Developed by OWASP (Open Web Application Security Project), [ZAP or Zed Attack Proxy](https://www.zaproxy.org/) is a multi-platform, open-source web application security testing tool. ZAP is used for finding a number of security vulnerabilities in a web  app during the development as well as the testing phase. Thanks to its  intuitive GUI, Zed Attach Proxy can be used with equal ease by newbies  as that by experts. The security testing tool supports command-line  access for advanced users. In addition to being one of the most famous OWASP projects, it is awarded the flagship status. ZAP is written in Java. Other than  its use as a scanner, ZAP can also be used to intercept a proxy for  manually testing a webpage. ZAP exposes:

* Application error disclosure
* Cookie not HttpOnly flag
* Missing anti-CSRF tokens and security headers
* Private IP disclosure
* Session ID in URL rewrite
* SQL injection
* XSS injection

**Key highlights:**

* Automatic scanning
* Easy to use
* Multi-platform
* Rest-based API
* Support for authentication
* Uses traditional and powerful AJAX spiders

Download the [Zed Attack Proxy (ZAP) source code](https://github.com/zaproxy).

**2. Wfuzz**

Developed in Python, [Wfuzz](http://www.edge-security.com/wfuzz.php) is popularly used for brute-forcing web applications. The open-source  security testing tool has no GUI interface and is usable only via  command line. Vulnerabilities exposed by Wfuzz are:

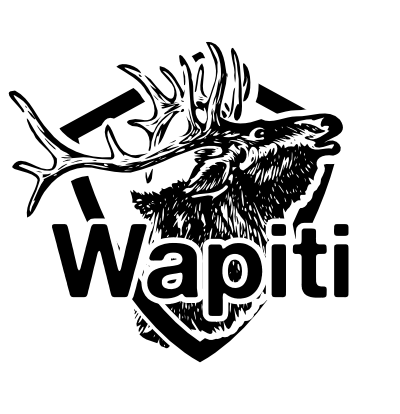
* LDAP injection
* SQL injection
* XSS injection

**Key highlights:**

* Authentication support
* Cookies fuzzing
* Multi-threading
* Multiple injection points
* Support for proxy and SOCK

Download [Wfuzz source code](https://github.com/xmendez/wfuzz).

**3. Wapiti**

One of the leading web application security testing tools, [Wapiti](http://wapiti.sourceforge.net/) is a free of cost, open source project from SourceForge and devloop. In order to check web applications for security vulnerabilities, Wapiti  performs black box testing. As it is a command-line application, it is  important to have a knowledge of various commands used by Wapiti. Wapiti is easy to use for the seasoned but testing for newcomers. But don’t  worry, you can find all the Wapiti instructions on the official  documentation. For checking whether a script is vulnerable or not,  Wapiti injects payloads. The open source security testing tool provides  support for both GET and POSTHTTP attack methods. Vulnerabilities  exposed by Wapiti are:

* Command Execution detection
* CRLF injection
* Database injection
* File disclosure
* Shellshock or Bash bug
* SSRF (Server Side Request Forgery)
* Weak .htaccess configurations that can be bypassed
* XSS injection
* XXE injection

**Key highlights:**

* Allows authentication via different methods, including Kerberos and NTLM
* Comes with a buster module, allowing brute force directories and files names on the targeted web server
* Operates like a [fuzzer](https://en.wikipedia.org/wiki/Fuzzinghttps:/en.wikipedia.org/wiki/Fuzzing)
* Supports both GET and POSTHTTP methods for attacks

Download [Wapiti source code](https://github.com/mbarbon/wapiti).

**4. W3af**

One of the most popular web application security testing frameworks that are also developed using Python is [W3af](http://w3af.org/). The tool allows testers to find over 200 types of security issues in web applications, including:

* Blind SQL injection
* Buffer overflow
* Cross-site scripting
* CSRF
* Insecure DAV configurations

Key highlights:

* Authentication support
* Easy to get started with
* Offers intuitive GUI interface
* Output can be logged into a console, a file or email

Download [W3af source code](https://github.com/andresriancho/w3af).

**5. SQLMap**

Allowing automating the process of detecting and utilizing SQL injection vulnerability in a website’s database, [SQLMap](http://sqlmap.org/) is entirely free to use. The security testing tool comes with a  powerful testing engine, capable of supporting 6 types of SQL injection  techniques:

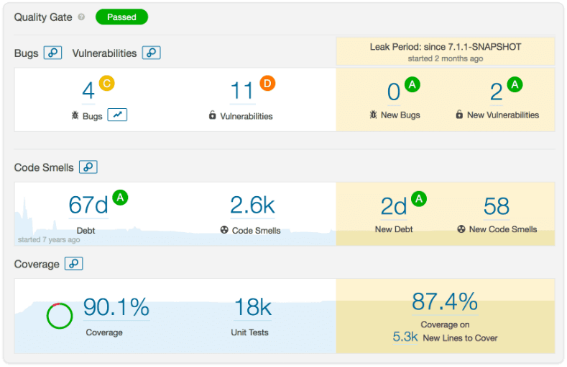
* Boolean-based blind
* Error-based
* Out-of-band
* Stacked queries
* Time-based blind
* UNION query

**Key highlights:**

* Automates the process of finding SQL injection vulnerabilities
* Can also be used for security testing a website
* Robust detection engine
* Supports a range of databases, including MySQL, Oracle, and PostgreSQL

Download [SQLMap source code](https://github.com/sqlmapproject/sqlmap).

**6. SonarQube**

 Another opportune open source security testing tool is [SonarQube](https://www.sonarqube.org/). In addition to exposing vulnerabilities, it is used to measure the  source code quality of a web application. Despite being written in Java, SonarQube is able to carry out analysis of over 20 programming  languages. Furthermore, it gets easily integrated with continuous  integration tools to the likes of Jenkins. Issues found by SonarQube are highlighted in either green or red light. While  the former represent low-risk vulnerabilities and issues, the latter  corresponds to severe ones. For advanced users, access via command  prompt is available. An interactive GUI is in place for those relatively new to testing. Some of the vulnerabilities exposed by SonarQube  include:

* Cross-site scripting
* Denial of Service (DoS) attacks
* HTTP response splitting
* Memory corruption
* SQL injection

**Key highlights:**

* Detects tricky issues
* DevOps integration
* Set up analysis of pull requests
* Supports quality tracking of both short-lived and long-lived code branches
* Offers [Quality Gate](https://docs.sonarqube.org/7.4/user-guide/quality-gates/)
* Visualize history of a project

Download [SonarQube source code](https://github.com/SonarSource/sonarqube).

**7. Nogotofail**

A network traffic security testing tool from Google, [Nogotofail](https://security.googleblog.com/2014/11/introducing-nogotofaila-network-traffic.html) is a lightweight application that is able to detect TLS/SSL  vulnerabilities and misconfigurations. Vulnerabilities exposed by  Nogotofail are:

* MiTM attacks
* SSL certificate verification issues
* SSL injection
* TLS injection

**Key highlights:**

* Easy to use
* Lightweight
* Readily deployable
* Supports setting up as a router, proxy or VPN server

Download [Nogotofail source code](https://github.com/google/nogotofail).

**8. Iron Wasp**

An open-source, powerful scanning tool, [Iron Wasp](https://ironwasp.org/) is able to uncover over 25 types of web application vulnerabilities.  Additionally, it can also detect false positives and false negatives.  Iron Wasp assists in exposing a wide variety of vulnerabilities,  including:

* Broken authentication
* Cross-site scripting
* CSRF
* Hidden parameters
* Privilege escalation

**Key highlights:**

* Extensible via plugins or modules are written in C#, Python, Ruby, or VB.NET
* GUI-based
* Report generation in HTML and RTF formats

Download [Iron Wasp source code](https://github.com/Lavakumar/IronWASP).

**9. Grabber**

The portable [Grabber](https://tools.kali.org/web-applications/grabber) is designed to scan small web applications, including forums and  personal websites. The lightweight security testing tool has no GUI  interface and is written in Python. Vulnerabilities uncovered by Grabber includes:

* Backup files verification
* Cross-site scripting
* File inclusion
* Simple AJAX verification
* SQL injection

**Key highlights:**

* Generates a stats analysis file
* Simple and portable
* Supports JS code analysis

Download [Grabber source code](https://github.com/amoldp/Grabber-Security-and-Vulnerability-Analysis-).

**10. Arachni**

Apt for both penetration testers and admins, [Arachni](http://www.arachni-scanner.com/) is designed to identify security issues within a web application. The  open-source security testing tool is capable of uncovering a number of  vulnerabilities, including:

* Invalidated redirect
* Local and remote file inclusion
* SQL injection
* XSS injection

**Key highlights:**

* Instantly deployable
* Modular, high-performance Ruby framework
* Multi-platform support

Download [Arachni source code](https://github.com/Arachni/arachni).

**Conclusion**

**This sums up the list of top  10 open source testing tools for web applications. Which is your  favourite application security testing tool? Tell us in the comments.  All the best for your Ethical Hacking journey!**

If you are new to hacking then [Learn Ethical Hacking From Scratch](https://click.linksynergy.com/deeplink?id=jU79Zysihs4&mid=39197&murl=https://www.udemy.com/course/learn-ethical-hacking-from-scratch/) course would be a great starting point.

If you want to dig deeper into information security then you can check out community-recommended best [Information Security](https://hackr.io/tutorials/learn-information-security-ethical-hacking?ref=blog)