

Secure Programming

A.A. 2022/2023

Corso di Laurea in Ingegneria delle Telecomunicazioni

J. Dynamic Security Test

Paolo Ottolino

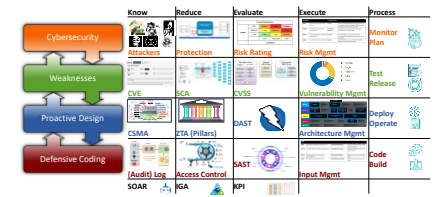
Politecnico di Bari

Secure Programming Lab: Course Program

- A. **Intro Secure Programming: «Who-What-Why-When-Where-How»**
- B. **Building Security in: Buffer Overflow, UAF, Command Injection**
- C. **SwA: Weaknesses, Vulnerabilities, Attacks**
- D. **SwA (Software Assurance): Vulnerabilities and Weaknesses (CVE, OWASP, CWE)**
- E. **Security & Protection: Objectives (CIA), Risks (Likelihood, Impact), Rating Methodologies**
- F. **Security & Protection: Security Indicators, BIA, Protection Techniques (AAA, Listing, Duplication etc.)**
- G. **Architecture and Processes: App Infrastructure, Three-Tiers, Cloud, Containers, Orchestration**
- H. **Architecture and Processes 2: Ciclo di Vita del SW (SDLC), DevSecOps (OWASP DSOMM, NIST SSDF)**
- I. **Free Security Tools: OWASP (ZAP, ESAPI, etc), NIST (SAMATE, SARD etc.)**
- J. **Dynamic Security Test: VA, PT, DAST, WebApp Sec Scan Framework (Nikto, Wapiti, OWASP ZAP)**
- K. **Operating Environment: Kali Linux on WSL**
- L. **Python: Powerful Language for easy creation of hacking tools**
- M. **Exercises: SecureFlag**



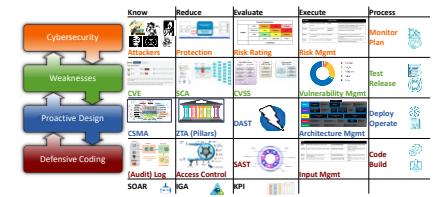
Application Security Testing



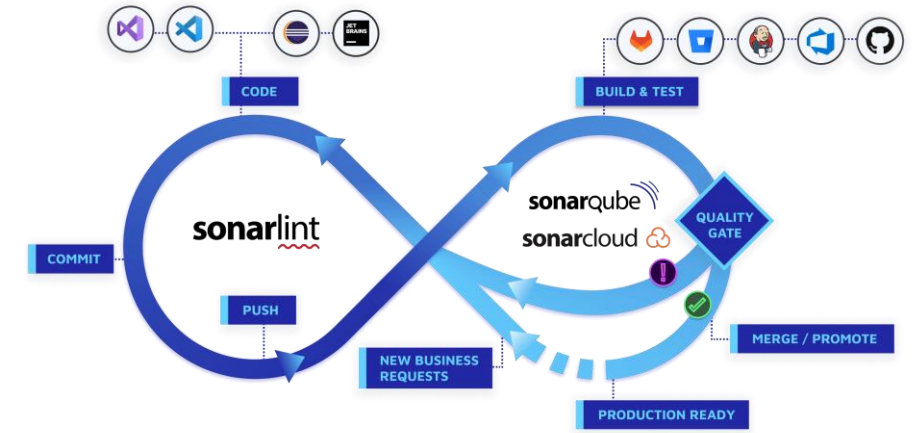
1. Dynamic vs Static Security Test
2. Kali Linux Tools by Category
3. Web Site Pentesting with Kali Linux

J.1. Security Test: Static vs Dynamic

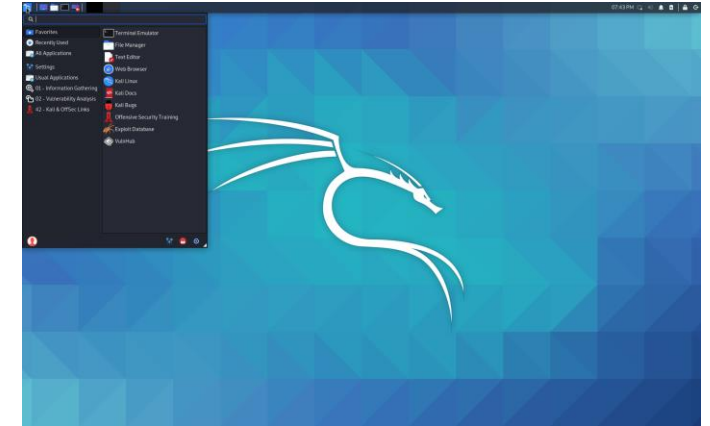
Introduction



Static analysis is performed in a non-runtime environment. Static application security testing (SAST) is a testing process that looks at the application from the inside out. This test process is performed without executing the program, but rather by examining the source code, byte code or application binaries for signs of security vulnerabilities. In the static test process, the application data and control paths are modeled and then analyzed for security weaknesses. Static analysis is a test of the internal structure of the application, rather than functional testing.

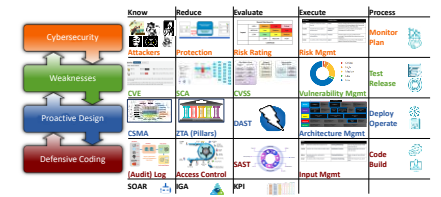


Dynamic analysis adopts the opposite approach and is executed while a program is in operation. Dynamic application security testing (DAST) looks at the application from the outside in — by examining it in its running state and trying to manipulate it in order to discover security vulnerabilities. The dynamic test simulates attacks against a web application and analyzes the application's reactions, determining whether it is vulnerable. Having originated and evolved separately, static and dynamic analysis have, at times, been mistakenly viewed in opposition. There are, however, a number of strengths and weaknesses associated with both approaches to consider.



J.2. Kali Linux Tools by Category

Introduction



Tools Categorization:

Kali Tools divided by categories:

https://www.tutorialspoint.com/kali_linux/index.htm

Kali Linux Tutorial

LEARN KALI LINUX
absolute beginners

Kali Linux Tutorial

- o Kali Linux - Home
- o Installation & Configuration
- o Information Gathering Tools
- o Vulnerability Analyses Tools
- o Kali Linux - Wireless Attacks
- o Website Penetration Testing

PDF Version Quick Guide Resources Job Search Discussion

Kali Linux is one of the best open-source security packages of an ethical hacker, containing a set of tools divided by categories. Kali Linux can be installed in a machine as an Operating System, which is discussed in this tutorial. Installing Kali Linux is a practical option as it provides more options to work and combine the tools.

This tutorial gives a complete understanding on Kali Linux and explains how to use it in practice.

Kali Tools: <https://www.kali.org/tools/> Detailed description of each installed (or installable) tool in the distro

Nikto

Nikto is a pluggable web server and CGI scanner written in Perl, using rfp's LibWhisker to perform fast security or informational checks.

Features:

- Easily updatable CSV-format checks database
- Output reports in plain text or HTML
- Available HTTP versions automatic switching
- Generic as well as specific server software checks
- SSL support (through libnet-ssleay-perl)
- Proxy support (with authentication)
- Cookies support

Installed size: 2.22 MB

How to install: `sudo apt install nikto`

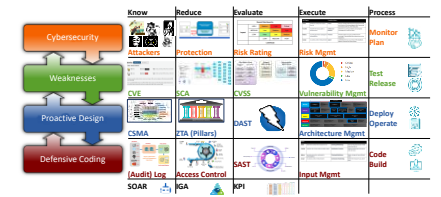
Dependencies:

nikto



J.2.a Kali Linux Tools by Category

Categories

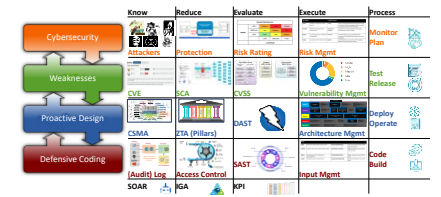


Identified Categories (https://www.tutorialspoint.com/kali_linux/index.htm)

- Information Gathering Tools (e.g. nmap for scanning the open ports)
- Vulnerability Analyses Tools (e.g. CISCO vulnerabilities)
- Kali Linux - Wireless Attacks (WiFi vulnerabilities)
- Website Penetration Testing (e.g. **nikto**, **ZAP** etc.)
- Kali Linux - Exploitation Tools (e.g. Metasploit)
- Kali Linux - Forensics Tools (e.g. p0f, pdfparser)
- Kali Linux - Social Engineering (e.g. Social Engineering Toolkit)
- Kali Linux - Stressing Tools: to create DoS attacks
- Kali Linux - Sniffing & Spoofing (simple as wiretapping)
- Kali Linux - Password Cracking Tools (e.g. John the ripper)
- Kali Linux - Maintaining Access: tools for C&C
- Kali Linux - Reverse Engineering: from binary to source code
- Kali Linux - Reporting Tools: share the result in a presentable format

J.3. Kali Linux: Website Pentesting

5 website pentesting tools



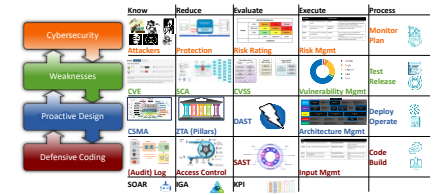
1. **Nikto**: pluggable web server and CGI scanner written in Perl (*)
2. **SkipFish**: active web application security reconnaissance tool
3. **Wapiti**: scan the web pages of the deployed web applications, looking for scripts and forms where it can inject data
4. **OWASP ZAP**: easy-to-use integrated penetration testing tool for finding vulnerabilities in web applications
5. **SQLmap**: automates the process of detecting and exploiting SQL injection flaws
6. **Sqlninja**: detecting SQLi on MS SQL server
7. **Vega**: scanner and testing platform for web application, identifying SQLi, XSS, disclosed information etc.
8. **WPscan**: WordPress CMS scanning tool
9. **JoomScan**: Joomla CMS scanning tool
10. **XSSer**: automatic -framework- to detect, exploit and report XSS vulnerabilities in web-based applications.

(*) see also (<https://www.freecodecamp.org/news/an-introduction-to-web-server-scanning-with-nikto/> <https://null-byte.wonderhowto.com/how-to/scan-for-vulnerabilities-any-website-using-nikto-0151729/>)



J.3a. DVWA

Damn Vulnerable Web Application



DVWA: Damn Vulnerable Web Application: <http://www.dvwa.co.uk/>

Also in Kali: <https://www.kali.org/tools/dvwa/>

Home
Instructions
Setup

Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored

DVWA Security
PHP Info
About
Logout

Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing **XAMPP** onto a local machine inside your LAN which is used solely for testing.

Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the persons who uploaded and installed it.

General Instructions

The help button allows you to view hints/tips for each vulnerability and for each security level on their respective page.

You have logged in as 'admin'

Username: admin
Security Level: high
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

<https://wilsonmar.github.io/owasp-testing/>

