Secure Programming

A.A. 2022/2023

Corso di Laurea in Ingegneria delle Telecomnicazioni

J. Dynamic Security Test

Paolo Ottolino

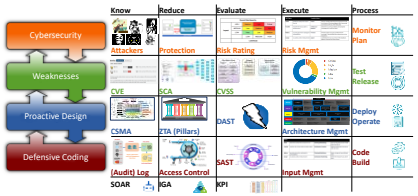
Politecnico di Bari

DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELL'INFORMAZIONE

Secure Programming Lab: Course Program

A. Intro Secure Programming: «Who-What-Why-When-Where-How»
B. Building Security in: Buffer Overflow, UAF, Command Inection
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

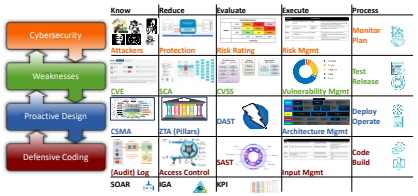
Application Security Testing



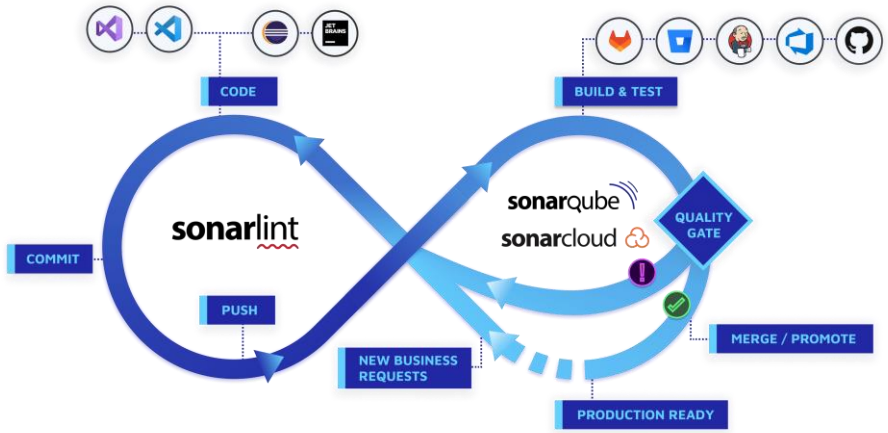
1. **Dynamic vs Static Security Test**
2. **Kali Linux Tools by Category**
3. **Web Site Pentesting with Kali Linux**

J.1. Security Test: Static vs Dynamic

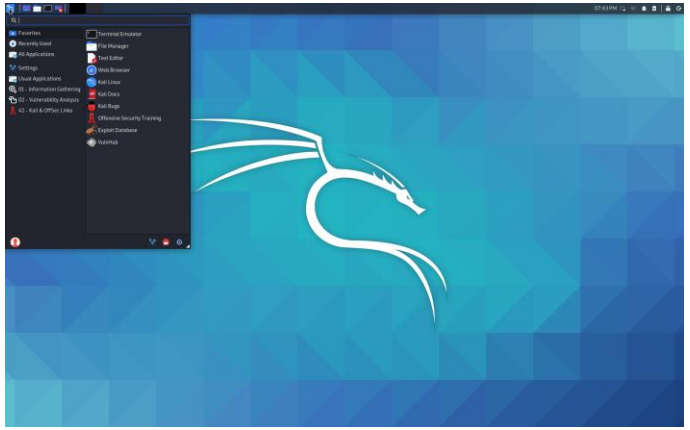
Introduction



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.

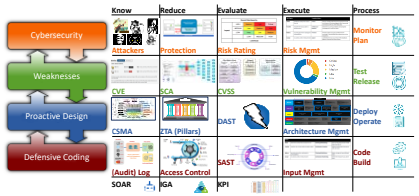


Dynamic analysis adopts the opposite approach and is executed while a program is in operation. Dynamic application security testing (DAST) looks at the application from the outside in — by examining it in its running state and trying to manipulate it in order to discover security vulnerabilities. The 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.



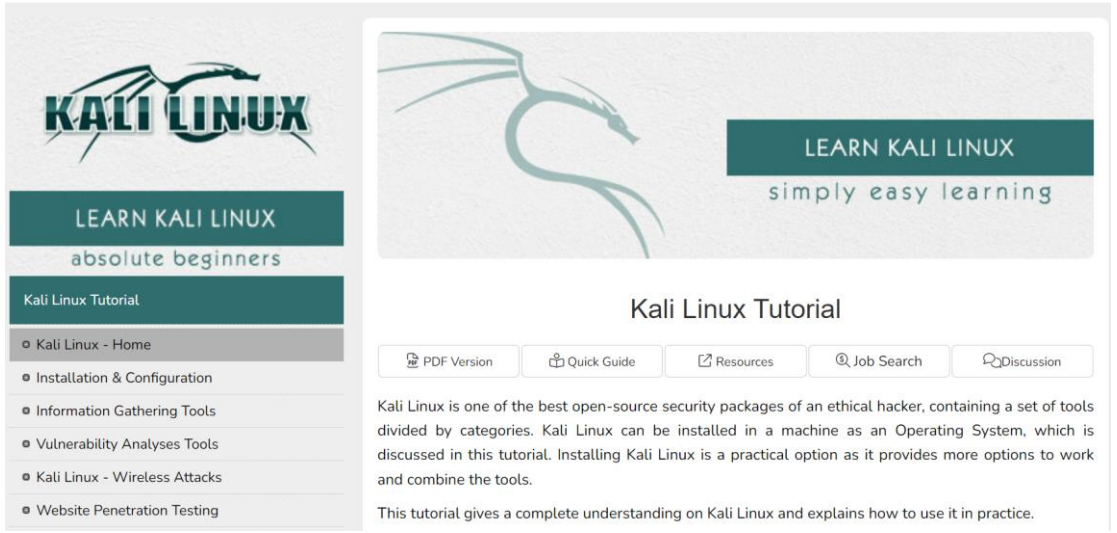
J.2. Kali Linux Tools by Category

Introduction

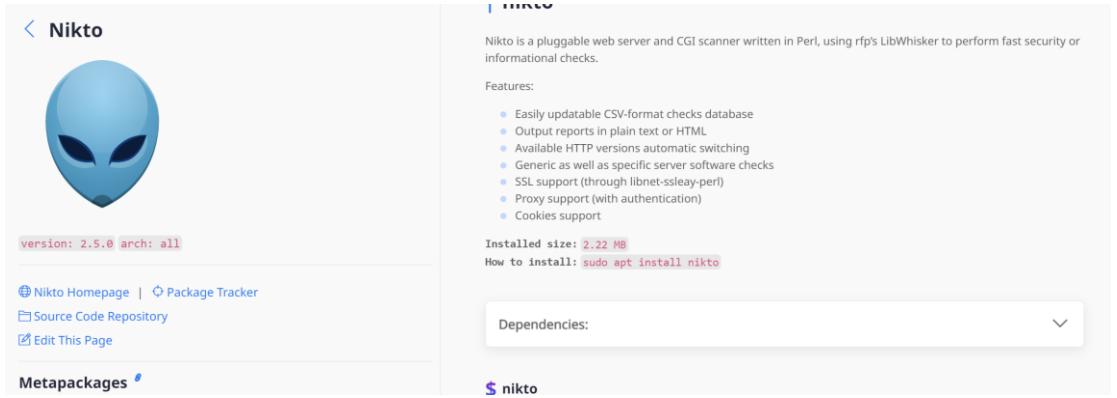


Tools Categorization:

Kali Tools divided by categories: https://www.tutorialspoint.com/kali_linux/index.htm

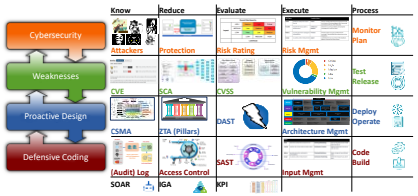


Kali Tools: https://www.kali.org/tools/ Detailed description of each installed (or installable) tool in the distro



J.2.a Kali Linux Tools by Category

Categories

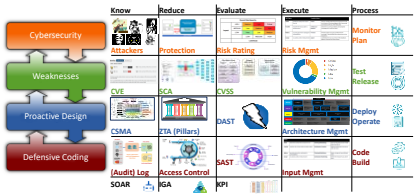


Identified Categories (https://www.tutorialspoint.com/kali_linux/index.htm)

- 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

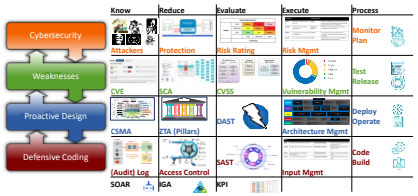


1. **Nikto**: pluggable web server and CGI scanner written in Perl (*)
2. **SkipFish**: active web application security reconnaissance tool
3. **Wapiti**: scan the web pages of the deployed web applications, looking for scripts and forms where it can inject data
4. **OWASP ZAP**: easy-to-use integrated 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/

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