Flawnter DAST Rules Overview

Dynamic Application Security Testing (DAST) is important process in application security testing designed to identify vulnerabilities and security flaws in running web application. Flawnter DAST rule sets are predefined sets of rules that Flawnter use to analyze and evaluate the running application for potential security issues. These rule sets are crafted to address specific types of security threats and industry best practices from OWASP, SANS/CWE, NIST, SEI CERT, PCI and others. Flawnter uses different type of rule sets to achieve the best results. There are direct rules, generic rules and predictive rules that use AI driven analysis. Additionally Flawnter has deep scan capabilities that help go beyond and potentially find more bugs. Please visit to our documentation https://www.flawnter.com/documentation that will show how organizations can use Flawnter configuration file to tailor these rule sets to their specific needs and achieve a more robust and secure software development lifecycle.

The table provided below focuses on the most commonly used security test rules. It's important to note that this list is not exhaustive, and there are other security rules that are not included in this table. Furthermore, each primary testing rule may encompass hundreds of sub-test rules. When aggregated, the total number of rules surpasses several thousand.

#	Rule	Description	CWE ID
1	Cryptography Issues	Weak algorithms, weak keys, weak hashes, weak randomness, etc.	327, 1346
2	SQL/XML/LOG Injection	Injecting untrusted input into SQL, XML, LOG.	89, 90, 91
3	Path Traversal & File Manipulation Attacks	Looks for path/directory traversal attacks that come from untrusted input.	22
4	Cross-site Scripting	Check for any cross-site scripting attacks including DOM based.	79
5	Broken/Improper Authentication	Check for possible authentication bypass and weaknesses.	287
6	Information / Sensitive Data Exposure	Information Disclosure, Sensitive Data Exposure.	200
7	Directory Listing	Disclosure of directory listings.	548
8	Insecure/Improper Authorization	Insecure/Improper access controls.	284, 285
9	Clickjacking	Check for clickjacking attacks.	1021
10	Response Splitting / Header Injections	Check for untrusted input being sent to functions that may cause response splitting and header injections.	113

11	Security Misconfiguration	Check for incorrect configurations.	16
12	Insecure Communication	Check for cleartext transmission of data. Missing encryption.	319
13	Xml External Entity	Check for Xml External Entity attacks.	611
14	Missing Http Only Cookie Attribute	Check for missing HttpOnly flag for sensitive cookies.	1004
15	Missing Secure Cookie Attribute	Check for missing Secure flag for sensitive cookies.	614
16	Debug Enabled	Check to see if debug is turned on.	489
17	External Javascript Linking	Inclusion of Web Functionality from an Untrusted source.	830
18	WebSockets Origin Header Checking	Missing Origin Validation in WebSockets.	1385