Project Name
WebGoat (Demo)

Last Analysis Date
2020-03-07

OWASP Vulnerabilities
121

OWASP Risk Factor
8.3%

OWASP Rating
E

OWASP Vulnerabilities Density
40.6%

OWASP Technical Debt
3d 2h

OWASP TOP 10 Application Security Risks

		Category Rating	Vulnerabilities					HotSpots to Review
A1	**Injection** Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query.	E	12	0	0	0	0	18
A2	**Broken Authentication** Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.	D	0	6	0	0	0	16
A3	**Sensitive Data Exposure** Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes.Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.	B	0	0	0	14	0	14
A4	**XML External Entities (XXE)** External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.	E	1	0	0	0	0	0
A5	**Broken Access Control** Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.	D	0	1	1	0	0	4
A6	**Security Misconfiguration** This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information.	E	20	5	0	0	0	4
A7	**Cross-Site Scripting (XSS)** XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.	E	1	0	0	0	0	2
A8	**Insecure Deserialization** Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.	A	0	0	0	0	0	2
A9	**Components with Known Vulnerabilities** Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.	A	0	0	0	0	0	0
A10	**Insufficient Logging & Monitoring** Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data.	A	0	0	0	0	0	0

Project Name
WebGoat (Demo)

Last Analysis Date
2020-03-07

OWASP TOP 10 Vulnerabilities Breakdown

Security-related issues which represents a backdoor for attackers

Severity	Rule	OWASP Category	# Vulnerabilities
	"@RequestMapping" methods should specify HTTP method	A6	20
	Database queries should not be vulnerable to injection attacks	A1	12
	XML parsers should not be vulnerable to XXE attacks	A4	1
	Endpoints should not be vulnerable to reflected cross-site scripting (XSS) attac...	A7	1
	HTTP referers should not be relied on	A2	4
	"Random" objects should be reused	A6	3
	Authentication should not rely on insecure "PasswordEncoder"	A2 A3 A6	2
	Persistent entities should not be used as arguments of "@RequestMapping" methods	A5	1
	Server-side requests should not be vulnerable to forging attacks	A5	1
	Throwable.printStackTrace(...) should not be called	A3	14

OWASP TOP 10 Security HotSpots to Review

Security Hotspots highlight security-sensitive pieces of code that need to be manually reviewed to ensure the sensitive piece of code is being used in the safest manner. Upon review, you'll either find a Vulnerability that needs to be fixed or that there is no threat.

Severity	Rule	OWASP Category	# Hotspots
	Hard-coded credentials are security-sensitive	A2	14
	Formatting SQL queries is security-sensitive	A1	9
	Using pseudorandom number generators (PRNGs) is security-sensitive	A3	8
	Using regular expressions is security-sensitive	A1	5
	Controlling permissions is security-sensitive	A5	4
	Using command line arguments is security-sensitive	A1	4
	Hashing data is security-sensitive	A3 A6	3
	Disabling Spring Security's CSRF protection is security-sensitive	A6	2
	Deserializing objects from an untrusted source is security-sensitive	A8	2
	Creating cookies without the "HttpOnly" flag is security-sensitive	A7	2
	Creating cookies without the "secure" flag is security-sensitive	A3	2
	Writing cookies is security-sensitive	A3	1