APPCHECK NG

ACCURACY IS EVERYTHING

AppCheck NG vs OWASP Top Ten

Based on a broad consensus, the OWASP Top Ten defines the current most critical web application security flaws.

OWASP Top Ten Coverage

The following table outlines how AppCheck NG identifies these vulnerabilities and helps to mitigate risk.

Category	Description	AppCheck NG Coverage
A1: Injection Flaws	Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorisation.	Each target component is assessed for injection vulnerabilities. Key vulnerabilities such as SQL Injection are exploited to provide proof of concept evidence and deliver 100% detection accuracy. **Example injection vulnerabilities discovered by AppCheck NG:** • Blind SQL Injection • Error based SQL Injection • XPath injection • LDAP injection • PHP Code / Complex syntax injection • Command injection (including blind) • Path traversal • Header / Control character injection
A2: Broken Authentication and Session management	Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.	Authentication and session management components are identified during the initial discovery phase of the scan. If configured to do so, usernames gathered by passive assessment modules can then be used in dictionary based authentication attacks against the target system. **The following authentication and session management vulnerabilities are detected by AppCheck NG:** • Common username enumeration flaws (e.g. parameter cycling, registration response messages, authentication error messages, Wordpress username enumeration) • Authentication system weaknesses (e.g. verbose authentication error disclosure) • Optional brute force password guessing attacks • Default credentials check

Category	Description	AppCheck^{NG} Coverage
A3: Cross Site Scripting (XSS)	XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. 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.	By combining Classic Cross Site Scripting (XSS) detection and JavaScript runtime taint analysis, AppCheck^{NG} achieves industry leading detection rates and accuracy for one of the most prevalent vulnerabilities. AppCheck^{NG} can hook events deep within a customised web browser engine to perform analysis of client-side web application activity, allowing the detection of subtle DOM-XSS vulnerabilities that would be missed by most mainstream scanners. Adobe Flash files are decompiled and analysed using static analysis techniques to detect flash based XSS vulnerabilities. **The following Cross Site Scripting vulnerabilities are identified by AppCheck^{NG}:** • Reflected Cross Site Scripting • Adobe Flash (AS2 & AS3) XSS flaws • DOM Based Cross Site Scripting Persistent/Stored Cross Site Scripting • HTTP Header Injection • HTTP Response splitting • HTML5 postMessage calls are analysed for cross domain script injection vulnerabilities
A4: Insecure Direct Object Reference	A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorised data.	The AppCheck^{NG} passive content discovery engine attempts to identify sensitive information within the application. Any numeric identifiers passed to the affected component are then cycled in an attempt to identify direct object reference vulnerabilities. • Parameter cycling / AI based content enumeration
A5: Security Misconfiguration	Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.	A vulnerability assessment is performed against the target application and hosting environment using multiple vulnerability scanners. • Operating system, network and application level vulnerability assessment using a combination of industry leading vulnerability scanners and custom assessment modules

APPCHECK NG

ACCURACY IS EVERYTHING

Category	Description	AppCheck NG Coverage
A6: Sensitive Data Exposure	Many web applications do not properly protect sensitive data, such as credit cards, tax ID's, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.	• Browser content caching directives and HTML form autocomplete settings are assessed to ensure sensitive data is not cached by the client browser • Database access gained via SQL injection is leveraged to identify sensitive data that is unencrypted (no sensitive data is retrieved but rather proof that it could be accessed is provided e.g. table and column names) • All systems that transmit sensitive data are assessed to ensure data is encrypted using SSL3/TLS and that weak ciphers and protocols are not supported • Error handling and other application responses are carefully examined for information disclosure vulnerabilities
A9: Using components with known vulnerabilities	Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.	• Multiple vulnerability scanners are used to assess server security (e.g. patch levels) • Application libraries such as jQuery are checked against known vulnerable versions • Dedicated modules for common platforms such as WordPress are deployed to perform technology specific assessments
A10: Unvalidated Redirects and Forwards	Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorised pages.	**The following redirect and forward vulnerabilities are identified by AppCheck NG:** • URL structures are injected within each URI based parameter to identify open redirect vulnerabilities • A number of techniques are used in an attempt to bypass URL validation schemes

For more information please contact us on:

T. 01924 284 269 E. info@appcheck-ng.com W. appcheck-ng.com

AppCheck NG, Unit 1 Centre 27 Business Park, Bankwood Way, Birstall WF17 9TB