# Coverity® OWASP Top 10 2021 Report

SM Solutions
SW Security & Quality

## WebGoat_Checker_IDE

| | |
|---|---|
| Company | Sm-Solutions |
| Project | WebGoat_Checker_IDE |
| Project Contact | Kakao Entertainment |
| Contact Email | jhkim@sm-solutions.co.kr |
| Report Generation Date | Jan 3, 2024 2:33 PM |

# Coverity® **OWASP Top 10 2021 Report**

## Executive Summary

This report details the application security assessment activities that were carried out, providing a summary of findings, compliance against published policy requirements, and remediation actions required. Also provided is a detailed breakdown and cross reference between technical findings and Coverity analysis results.

The intended audience for this report is an application security assurance team and their clients or end users. To review detailed code-level findings, it is recommended that developers click this link to the Coverity Connect platform (http://192.168.0.28:8080/reports#p10051) in order to see source code annotated with remediation recommendations.

Lines of Code Inspected: 18144

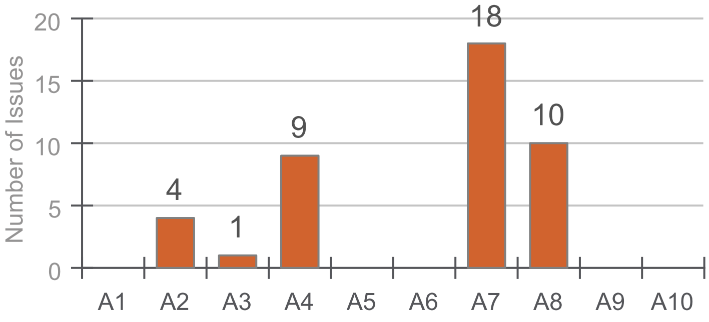
### Scorecard

The issues were evaluated according to each element of the report's policy. The results are shown in the table below. An overall status of "pass" is assigned if all the policy elements passed.

| Policy Element | Target | Value | Passed |
|---|---|---|---|
| OWASP Top 10 2021 Count | 0 | 42 | No |
| **Overall Status** | | | **No** |

### Issues By Category

The chart below shows the number of occurrences in each of the categories.



### Additional Quality Measures

This table reports the numbers of issues of various categories that were not included in this report. Although they were excluded from the report, they may nonetheless indicate the presence of significant quality or security issues.

| Category | Count |
|---|---|
| Issue Occurrences Marked "False Positive" or "Intentional" | 0 |

## Action Items

The code base was evaluated based on the policy in force. The policy has the following elements:

- There must be no issues with CVSS Severity Critical or High. See the Analysis Details section for more information.
- There must be no OWASP Top 10 issues among those found in the project. See the OWASP Top 10 section for details. Coverity recommends the following actions in order to resolve critical outstanding issues, achieve compliance with policy, and improve the overall security of the software.

### Remediation of issues with CVSS Severity Critical or High

Resolve/Remediate all issues that have a CVSS Severity of Critical or High.

### OWASP Top 10 Remediation

Resolve 42 issues that are present in the OWASP Top 10. See the OWASP Top 10 Section for a list of them.

### Recent Source Code Analysis

Regular source code analysis is key to identifying security issues in a timely manner and to that these issues are effectively eliminated, in-line with development activities.

The current results are sufficiently recent (less than 30 days old).

### Long Term and Residual Risk Management

Review and consider broader improvement to the overall security posture of the target application.
Review outstanding lesser-rated issues to ensure minimal residual risk.
Review issues marked false positive to be sure that a coding change will not eliminate them
Review any security issues marked Informational to see if some are in fact credible threats.
Review and correct non-security issues found by Coverity Analysis, in order to increase the overall quality of the code.

# Coverity® OWASP Top 10 2021 Report

## Analysis Details

A Coverity project is a collection of one or more streams containing separately-analyzed snapshots. The latest snapshot in each stream is used when reporting results for a project. This section gives details about the streams and the analysis performed for each snapshot.

| Stream Name | Snapshot ID | Analysis Date | Analysis Version |
|---|---|---|---|
| WebGoat_IDE_Stream | 10212 | 2024-1-3 2:4:39 오후 | 2023.9.0 |

## The OWASP Top 10 - 2021 List

The Open Web Application Security Project (OWASP) is an open community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted. The OWASP maintains the OWASP Top 10 List for 2021, a prioritized list of security weaknesses. OWASP says, “We can no longer afford to tolerate relatively simple security problems like those presented in this OWASP Top 10.”

The table below shows the number of issues found in each category of the OWASP Top 10 for 2021.

| OWASP Top 10 - 2021 Category | CWE Number | Count |
|---|---|---|
| **A1:2021** - Broken Access Control | 1345 | 0 |
| **A2:2021** - Cryptographic Failures | 1346 | 4 |
| **A3:2021** - Injection | 1347 | 1 |
| **A4:2021** - Insecure Design | 1348 | 9 |
| **A5:2021** - Security Misconfiguration | 1349 | 0 |
| **A6:2021** - Vulnerable and Outdated Components | 1352 | 0 |
| **A7:2021** - Identification and Authentication Failures | 1353 | 18 |
| **A8:2021** - Software and Data Integrity Failures | 1354 | 10 |
| **A9:2021** - Security Logging and Monitoring Failures | 1355 | 0 |
| **A10:2021** - Server-Side Request Forgery (SSRF) | 1356 | 0 |
| **Total** | | **42** |

## OWASP Top 10 2021 Category: A2:2021 - Cryptographic Failures

### CWE 326: Inadequate Encryption Strength

## Detailed Issues Ranked By OWASP Top 10 2021 Category

This is categorized under the 2 in the OWASP Top 10 2021.

**Summary:** The software stores or transmits sensitive data using an encryption scheme that is theoretically sound, but is not strong enough for the level of protection required.

**Details:** A weak encryption scheme can be subjected to brute force attacks that have a reasonable chance of succeeding using current attack methods and resources.more details.

**Remediation:** Use a cryptographic algorithm that is currently considered to be strong by experts in the field.

| Issue ID (CID) and Issue Type | Source File and Line Number |
|---|---|
| 233266<br>Insufficient Asymmetric Key Size | /Users/KimJihyeong/Desktop/WebGoat-main/src/it/java/org/owasp/webgoat/JWTLessonIntegrationTest.jav<br>251<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233196<br>Insufficient Asymmetric Key Size | /Users/KimJihyeong/Desktop/WebGoat-main/src/test/java/org/owasp/webgoat/lessons/jwt/claimmisuse/<br>JWTHeaderJKUEndpointTest.java:51<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233188<br>Insufficient Asymmetric Key Size | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/lessons/cryptography/<br>CryptoUtil.java:38<br>First Detected: 2024-01-03T14:04:37.118+09:00 |

### CWE 327: Use of a Broken or Risky Cryptographic Algorithm

## Detailed Issues Ranked By OWASP Top 10 2021 Category

This is categorized under the 2 in the OWASP Top 10 2021.

**Summary:** The use of a broken or risky cryptographic algorithm is an unnecessary risk that may result in the exposure of sensitive information.

**Details:** The use of a non-standard algorithm is dangerous because a determined attacker may be able to break the algorithm and compromise whatever data has been protected. Well-known techniques may exist to break the algorithm.more details.

**Remediation:** When there is a need to store or transmit sensitive data, use strong, up-to-date cryptographic algorithms to encrypt that data. Select a well-vetted algorithm that is currently considered to be strong by experts in the field, and use well-tested implementations. As with all cryptographic mechanisms, the source code should be available for analysis.

| Issue ID (CID) and Issue Type | Source File and Line Number |
|---|---|
| 233195<br>Weak hash algorithm | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/lessons/cryptography/<br>HashingAssignment.java:55<br>First Detected: 2024-01-03T14:04:37.118+09:00 |

## OWASP Top 10 2021 Category: A3:2021 - Injection

### CWE 611: Improper Restriction of XML External Entity Reference ('XXE')

## Detailed Issues Ranked By OWASP Top 10 2021 Category

This is categorized under the 3 in the OWASP Top 10 2021.

**Summary:** The software processes an XML document that can contain XML entities with URIs that resolve to documents outside of the intended sphere of control, causing the product to embed incorrect documents into its output.

**Details:** XML documents optionally contain a Document Type Definition (DTD), which, among other features, enables the definition of XML entities. It is possible to define an entity by providing a substitution string in the form of a URI. The XML parser can access the contents of this URI and embed these contents back into the XML document for further processing.more details.

**Remediation:** Many XML parsers and validators can be configured to disable external entity expansion.

| Issue ID (CID) and Issue Type | Source File and Line Number |
|---|---|
| 233255<br>XML external entity processing enabled | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/lessons/xxe/CommentsCache.java:98<br>First Detected: 2024-01-03T14:04:37.118+09:00 |

## OWASP Top 10 2021 Category: A4:2021 - Insecure Design

### CWE 1023: Incomplete Comparison with Missing Factors

## Detailed Issues Ranked By OWASP Top 10 2021 Category

This is categorized under the 4 in the OWASP Top 10 2021.

**Summary:** The software performs a comparison between entities that must consider multiple factors or characteristics of each entity, but the comparison does not include one or more of these factors. This can lead to resultant weaknesses, e.g. by operating on the wrong object.

**Remediation:** Thoroughly test the comparison scheme before deploying code into production. Perform positive testing as well as negative testing.

| Issue ID (CID) and Issue Type | Source File and Line Number |
|---|---|
| 233220<br>Unsafe dependence on database key | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/container/users/LessonTracker.java:54<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233190<br>Unsafe dependence on database key | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/container/lessons/Assignment.java:44<br>First Detected: 2024-01-03T14:04:37.118+09:00 |

### CWE 253: Incorrect Check of Function Return Value

## Detailed Issues Ranked By OWASP Top 10 2021 Category

This is categorized under the 4 in the OWASP Top 10 2021.

**Summary:** The software incorrectly checks a return value from a function, which prevents the software from detecting errors or exceptional conditions.

**Details:** Important and common functions will return some value about the success of its actions. This will alert the program whether or not to handle any errors caused by that function.more details.

**Remediation:** Use a language or compiler that uses exceptions and requires the catching of those exceptions.

| Issue ID (CID) and Issue Type | Source File and Line Number |
|---|---|
| 233285<br>RV: Bad use of return value | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/webwolf/FileServer.java:10<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233245<br>RV: Bad use of return value | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/lessons/pathtraversal/ProfileUploadBase.java:43<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233237<br>RV: Bad use of return value | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/webwolf/MvcConfiguration.java:72<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233234<br>RV: Bad use of return value | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/lessons/xxe/BlindSendFileAssignment.java:76<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233233<br>RV: Bad use of return value | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/webwolf/FileServer.java:88<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233211<br>RV: Bad use of return value | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/lessons/clientsidefiltering/Salaries.java:62<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233182<br>RV: Bad use of return value | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/lessons/pathtraversal/ProfileUploadRetrieval.java:48<br>First Detected: 2024-01-03T14:04:37.118+09:00 |

## OWASP Top 10 2021 Category: A7:2021 - Identification and Authentication Failures

### CWE 613: Insufficient Session Expiration

## Detailed Issues Ranked By OWASP Top 10 2021 Category

This is categorized under the 7 in the OWASP Top 10 2021.

**Summary:** According to WASC, "Insufficient Session Expiration is when a web site permits an attacker to reuse old session credentials or session IDs for authorization."

**Remediation:** Set sessions/credentials expiration date.

| Issue ID (CID) and Issue Type | Source File and Line Number |
|---|---|
| 233287<br>JSON Web Token without expiration time | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/lessons/jwt/JWTRefreshEndpoint.java:86<br>First Detected: 2024-01-03T14:04:37.118+09:00 |

### CWE 613: Insufficient Session Expiration

## Detailed Issues Ranked By OWASP Top 10 2021 Category

This is categorized under the 7 in the OWASP Top 10 2021.

**Summary:** According to WASC, "Insufficient Session Expiration is when a web site permits an attacker to reuse old session credentials or session IDs for authorization."

**Remediation:** Set sessions/credentials expiration date.

| Issue ID (CID) and Issue Type | Source File and Line Number |
|---|---|
| 233284<br>JSON Web Token not before time ignored | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/webwolf/jwt/JWTToken.jav<br>124<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233281<br>JSON Web Token without expiration time | /Users/KimJihyeong/Desktop/WebGoat-main/src/test/java/org/owasp/webgoat/lessons/jwt/JWTVotesEndpointTest.java:228<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233279<br>JSON Web Token without expiration time | /Users/KimJihyeong/Desktop/WebGoat-main/src/test/java/org/owasp/webgoat/lessons/jwt/TokenTest.java:<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233272<br>JSON Web Token without expiration time | /Users/KimJihyeong/Desktop/WebGoat-main/src/test/java/org/owasp/webgoat/lessons/jwt/JWTSecretKeyEndpointTest.java:94<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233249<br>JSON Web Token expiration time ignored | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/webwolf/jwt/JWTToken.jav<br>124<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233240<br>JSON Web Token without expiration time | /Users/KimJihyeong/Desktop/WebGoat-main/src/test/java/org/owasp/webgoat/lessons/jwt/claimmisuse/JWTHeaderJKUEndpointTest.java:81<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233238<br>JSON Web Token without expiration time | /Users/KimJihyeong/Desktop/WebGoat-main/src/test/java/org/owasp/webgoat/lessons/jwt/claimmisuse/JWTHeaderKIDEndpointTest.java:38<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233224<br>JSON Web Token without expiration time | /Users/KimJihyeong/Desktop/WebGoat-main/src/test/java/org/owasp/webgoat/lessons/jwt/JWTSecretKeyEndpointTest.java:122<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233216<br>JSON Web Token without expiration time | /Users/KimJihyeong/Desktop/WebGoat-main/src/test/java/org/owasp/webgoat/lessons/jwt/JWTVotesEndpointTest.java:81<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233209<br>JSON Web Token without expiration time | /Users/KimJihyeong/Desktop/WebGoat-main/src/test/java/org/owasp/webgoat/lessons/jwt/JWTRefreshEndpointTest.java:102<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233205<br>JSON Web Token without expiration time | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/lessons/jwt/JWTVotesEndpoint.java:126<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233202<br>JSON Web Token without expiration time | /Users/KimJihyeong/Desktop/WebGoat-main/src/test/java/org/owasp/webgoat/lessons/jwt/JWTSecretKeyEndpointTest.java:82<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233201<br>JSON Web Token without expiration time | /Users/KimJihyeong/Desktop/WebGoat-main/src/test/java/org/owasp/webgoat/lessons/jwt/JWTVotesEndpointTest.java:63<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233198<br>JSON Web Token without expiration time | /Users/KimJihyeong/Desktop/WebGoat-main/src/test/java/org/owasp/webgoat/lessons/jwt/JWTVotesEndpointTest.java:246<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233180<br>JSON Web Token without expiration time | /Users/KimJihyeong/Desktop/WebGoat-main/src/test/java/org/owasp/webgoat/lessons/jwt/JWTSecretKeyEndpointTest.java:134<br>First Detected: 2024-01-03T14:04:37.118+09:00 |

### CWE 613: Insufficient Session Expiration

## Detailed Issues Ranked By OWASP Top 10 2021 Category

This is categorized under the 7 in the OWASP Top 10 2021.

**Summary:** According to WASC, "Insufficient Session Expiration is when a web site permits an attacker to reuse old session credentials or session IDs for authorization."

**Remediation:** Set sessions/credentials expiration date.

| Issue ID (CID) and Issue Type | Source File and Line Number |
|---|---|
| 233176<br>JSON Web Token without expiration time | /Users/KimJihyeong/Desktop/WebGoat-main/src/test/java/org/owasp/webgoat/lessons/jwt/JWTSecretKeyEndpointTest.java:107<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233170<br>JSON Web Token without expiration time | /Users/KimJihyeong/Desktop/WebGoat-main/src/test/java/org/owasp/webgoat/lessons/jwt/JWTSecretKeyEndpointTest.java:71<br>First Detected: 2024-01-03T14:04:37.118+09:00 |

## OWASP Top 10 2021 Category: A8:2021 - Software and Data Integrity Failures

### CWE 261: Weak Cryptography for Passwords

## Detailed Issues Ranked By OWASP Top 10 2021 Category

This is categorized under the 8 in the OWASP Top 10 2021.

**Summary:** Obscuring a password with a trivial encoding does not protect the password.

**Remediation:** Passwords should be encrypted with keys that are at least 128 bits in length for adequate security.

| Issue ID (CID) and Issue Type | Source File and Line Number |
|---|---|
| 233213<br>Weak password hash algorithm | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/container/WebSecurityConfig.java:112<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233207<br>Weak password hash algorithm | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/webwolf/WebSecurityConfig.java:103<br>First Detected: 2024-01-03T14:04:37.118+09:00 |

### CWE 345: Insufficient Verification of Data Authenticity

## Detailed Issues Ranked By OWASP Top 10 2021 Category

This is categorized under the 8 in the OWASP Top 10 2021.

**Summary:** The software does not sufficiently verify the origin or authenticity of data, in a way that causes it to accept invalid data.

| Issue ID (CID) and Issue Type | Source File and Line Number |
|---|---|
| 233197<br>JSON Web Token decoded without claims verification | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/webwolf/jwt/JWTToken.jav<br>124<br>First Detected: 2024-01-03T14:04:37.118+09:00 |

### CWE 347: Improper Verification of Cryptographic Signature

## Detailed Issues Ranked By OWASP Top 10 2021 Category

This is categorized under the 8 in the OWASP Top 10 2021.

**Summary:** The software does not verify, or incorrectly verifies, the cryptographic signature for data.

| Issue ID (CID) and Issue Type | Source File and Line Number |
|---|---|
| 233276<br>JSON Web Token decoded without signature verification | /Users/KimJihyeong/Desktop/WebGoat-main/src/it/java/org/owasp/webgoat/JWTLessonIntegrationTest.jav<br>75<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233268<br>JSON Web Token decoded without signature verification | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/lessons/jwt/<br>JWTVotesEndpoint.java:180<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233263<br>JSON Web Token decoded without signature verification | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/lessons/jwt/<br>JWTVotesEndpoint.java:203<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233230<br>JSON Web Token decoded without signature verification | /Users/KimJihyeong/Desktop/WebGoat-main/src/test/java/org/owasp/webgoat/lessons/jwt/TokenTest.java:<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233219<br>JSON Web Token decoded without signature verification | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/lessons/jwt/<br>JWTVotesEndpoint.java:155<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233189<br>JSON Web Token decoded without signature verification | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/lessons/jwt/<br>JWTRefreshEndpoint.java:137<br>First Detected: 2024-01-03T14:04:37.118+09:00 |
| 233169<br>JSON Web Token decoded without signature verification | /Users/KimJihyeong/Desktop/WebGoat-main/src/main/java/org/owasp/webgoat/lessons/jwt/<br>JWTRefreshEndpoint.java:107<br>First Detected: 2024-01-03T14:04:37.118+09:00 |

## Methodology

### Introduction

This report is a distillation of the output of the Coverity Code Advisor used on a particular code source base. Coverity Code Advisor is a static analysis tool that is capable of finding quality defects, security vulnerabilities, and test violations through the process of scanning the output of a specially-compiled code base. The information in this report is specific to security vulnerabilities detected by Coverity Code Advisor and their categorization in the OWASP and CWE/SANS ranking systems.

### About Static Analysis

Static analysis is the analysis of software code without executing the compiled program, for the purpose of finding logic errors or security vulnerabilities. Coverity's static analysis tools integrate with all major build systems and generate a high fidelity representation of source code to provide full code path coverage, ensuring that every line of code and execution path is analyzed. Code Advisor supports the market leading compilers for C, C++, Java, C#, Objective C, and Javascript.

### About CWE

CWE (Common Weakness Enumeration) is a software community project that is responsible for creating a catalog of software weaknesses and vulnerabilities and is sponsored by the office of Cybersecurity and Communications at the U.S. Department of Homeland Security. The Common Weakness Scoring System (CWSS) provides a method by which to identify and compare weaknesses.

CWE is used by vulnerability-listing efforts such as CWE/SANS Top 25 and OWASP Top 10, among others, to create generalized lists of ranked vulnerabilities. Some, but not all, of the issues reported by Coverity are mapped to CWE-listed vulnerabilities.

### The OWASP Top 10 List

The OWASP (Open Web Application Security Project) Foundation is an international organization whose mission is to advance the cause of secure software. As part of its activities, OWASP publishes a report of the most critical web application security flaws in rank order based on the input of a worldwide group of security experts. The most recent version of this list and accompanying report is the OWASP Top 10 List for 2021, The OWSAP Top 10 List is referenced by many standards including MITRE, PCI DSS, DISA, and the FTC.

### About Synopsys Software Integrity Group

Synopsys Software Integrity Group (Synopsys) is a leading provider of quality and security testing solutions. Synopsys provides an array of tools that assist developers in addressing critical quality and security issues early in the development cycle, thus saving development organizations from remediating issues late in the development cycle or after release when they are much more costly. Many major software development organizations, including 8 of the top 10 global brands and 9 of the top 10 top software companies, deploy Coverity analysis tools. Synopsys Software Integrity Group also maintains a free, cloud based analysis platform, called Scan, for the Open Source Community.