



# Coverity® CVSS Report

---

## Chipset\_INF

Company	Intel
Project	Chipset_INF
Project Contact	SDL
Contact Email	sys_chipsinf@intel.com
Report Generation Date	Jun 17, 2025 7:16 PM

## 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](https://coverity.devtools.intel.com/prod20/reports#p10010) (https://coverity.devtools.intel.com/prod20/reports#p10010) in order to see source code annotated with remediation recommendations.

Lines of Code Inspected: 10405

### 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 Count	0	0	Yes
CWE/SANS Top 25 Count	0	1	No
CVSS Critical Count	0	0	Yes
CVSS High Count	0	1	No
Overall Status			No

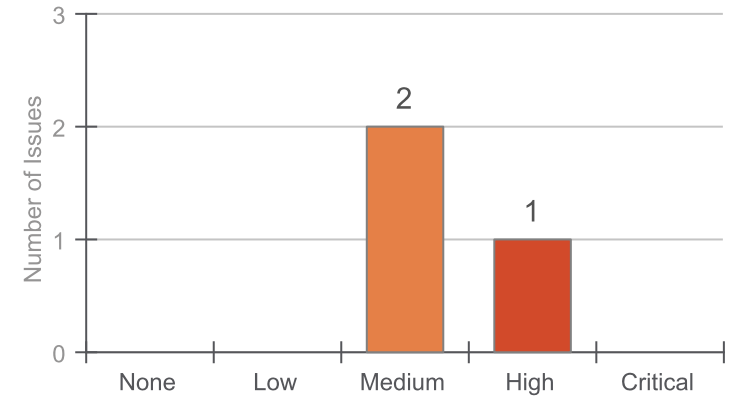
### Additional Quality Measures

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

Category	Count
Issues Marked "False Positive" or "Intentional"	0

### Issues By CVSS Severity

The chart below shows the number of issues in each of the CVSS severity categories.



---

## 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.
- There must be no CWE/SANS Top 25 issues among those found in the project. See the [CWE/SANS Top 25](#) 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.

- 1 “High” issues.

## OWASP Top 10 Remediation

The project has no issues in the OWASP Top 10.

## CWE/SANS Top 25 Remediation

Resolve 1 issues that are present in the CWE/SANS Top 25. See the [CWE/SANS Top 25](#) 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 ensuring 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® CVSS 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
Chipset_INF-master	1024243	2025-6-17 7:15:6 PM	2023.3.0

## The 2017 OWASP Top 10 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 2017](#), 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."

Each entry in the OWASP Top 10 refers to a set of CWE entries. Those entries may be individual weaknesses or families of weaknesses. See [the next section](#) for further discussion.

The table below shows the number of issues found in each category of the OWASP Top 10 for 2017.

2017 OWASP Top 10 Categories	CWE Number	Count
1. <a href="#">Injection</a>	1027	0
2. <a href="#">Broken Authentication</a>	1028	0
3. <a href="#">Sensitive Data Exposure</a>	1029	0
4. <a href="#">XML External Entities (XXE)</a>	1030	0
5. <a href="#">Broken Access Control</a>	1031	0
6. <a href="#">Security Misconfiguration</a>	1032	0
7. <a href="#">Cross-Site Scripting (XSS)</a>	1033	0
8. <a href="#">Insecure Deserialization</a>	1034	0
9. <a href="#">Using Components with Known Vulnerabilities *</a>	1035	0
10. <a href="#">Insufficient Logging &amp; Monitoring</a>	1036	0
<b>Total</b>		<b>0</b>

\* Category 9 of the OWASP Top 10 for 2017, "Using Components with Known Vulnerabilities," is not detected by Coverity Static Analysis, but is detected by BlackDuck and Protecode ES, which are other Synopsys products.

# Coverity® CVSS Report

5

## The CWE/SANS Top 25 List

The Common Weakness Enumeration is a community-developed dictionary of software weakness types. The [2019 CWE/SANS Top 25 Most Dangerous Software Errors](#) (or, "Top 25") is a list of weaknesses, taken from the CWE, that are thought to be the most widespread and critical errors that can lead to serious vulnerabilities in software.

Each category in the Top 25 List mentions one primary CWE identifier (CWE ID). Such a CWE ID can refer to an individual weakness or to a family of related weaknesses, since a given CWE ID may have children CWE IDs, which in turn may have children CWE IDs of their own. A Coverity issue corresponds to the most relevant CWE ID. A CWE/SANS Top 25 Category will consist of all of the Coverity issues that correspond to either the mentioned CWE ID or to one of its associated descendants.

The table below lists all the entries of the Top 25 and shows how many Coverity issues in the current project were found to be members of the Top 25.

2019 CWE/SANS Top 25 Categories	CWE Number	Count
1. Improper Restriction of Operations within the Bounds of a Memory Buffer	<a href="#">CWE-119</a>	0
2. Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	<a href="#">CWE-79</a>	0
3. Improper Input Validation	<a href="#">CWE-20</a>	0
4. Information Exposure	<a href="#">CWE-200</a>	0
5. Out-of-bounds Read	<a href="#">CWE-125</a>	0
6. Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	<a href="#">CWE-89</a>	0
7. Use After Free	<a href="#">CWE-416</a>	0
8. Integer Overflow or Wraparound	<a href="#">CWE-190</a>	0
9. Cross-Site Request Forgery (CSRF)	<a href="#">CWE-352</a>	0
10. Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	<a href="#">CWE-22</a>	0
11. Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	<a href="#">CWE-78</a>	0
12. Out-of-bounds Write	<a href="#">CWE-787</a>	0
13. Improper Authentication	<a href="#">CWE-287</a>	0
14. NULL Pointer Dereference	<a href="#">CWE-476</a>	1
15. Incorrect Permission Assignment for Critical Resource	<a href="#">CWE-732</a>	0
16. Unrestricted Upload of File with Dangerous Type	<a href="#">CWE-434</a>	0
17. Improper Restriction of XML External Entity Reference ('XXE')	<a href="#">CWE-611</a>	0
18. Improper Control of Generation of Code ('Code Injection')	<a href="#">CWE-94</a>	0
19. Use of Hard-coded Credentials	<a href="#">CWE-798</a>	0
20. Uncontrolled Resource Consumption ('Resource Exhaustion')	<a href="#">CWE-400</a>	0
21. Missing Release of Resource after Effective Lifetime	<a href="#">CWE-772</a>	0
22. Untrusted Search Path	<a href="#">CWE-426</a>	0
23. Deserialization of Untrusted Data	<a href="#">CWE-502</a>	0
24. Improper Privilege Management	<a href="#">CWE-269</a>	0
25. Improper Certificate Validation	<a href="#">CWE-295</a>	0
<b>Total</b>		<b>1</b>

## Detailed Issues Ranked By CVSS Severity

### CVSS Severity: High

#### CWE 476: NULL Pointer Dereference

This is categorized under the 14 in the [CWE/SANS Top 25](#).

**Summary:** A NULL pointer dereference occurs when the application dereferences a pointer that it expects to be valid, but is NULL, typically causing a crash or exit.

**Details:** NULL pointer dereference issues can occur through a number of flaws, including race conditions, and simple programming omissions. [more details](#).

**Remediation:** If all pointers that could have been modified are sanity-checked previous to use, nearly all NULL pointer dereferences can be prevented.

**NOTE:** A value of "No" for "CVSS\_Audited" indicates that the CVSS Vector, Score and Severity are based on static analysis data and that the issues with this designation have not been audited for accuracy by an application security professional or equivalent.

Issue ID (CID) and Issue Type	Source File and Line Number
<a href="#">1536103</a> Dereference after null check	/github_runner/intel/001/_work/Chipset_Repo/Chipset_Repo/chipset_inf_installer/Tools/MSBuildTasks/Common/CreateMup/CreateMup.cs:281 First Detected: Sep 5, 2023 10:03 AM CVSS Severity: High CVSS Score: 7.39 CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:L/I:L/A:L CVSS Audited: No NOTE: This issue has not been audited for accuracy by an application security professional or equivalent.

### CVSS Severity: Medium

#### CWE 366: Race Condition within a Thread

**Summary:** If two threads of execution use a resource simultaneously, there exists the possibility that resources may be used while invalid, in turn making the state of execution undefined.

**Remediation:** Use locking functionality. This is the recommended solution. Implement some form of locking mechanism around code which alters or reads persistent data in a multithreaded environment.

**NOTE:** A value of "No" for "CVSS\_Audited" indicates that the CVSS Vector, Score and Severity are based on static analysis data and that the issues with this designation have not been audited for accuracy by an application security professional or equivalent.

Issue ID (CID) and Issue Type	Source File and Line Number
<a href="#">1536104</a> Data race condition	/github_runner/intel/001/_work/Chipset_Repo/Chipset_Repo/chipset_inf_installer/Common/Bootstrapper/ViewModels/_MainViewModel.cs:416 First Detected: Sep 5, 2023 10:03 AM CVSS Severity: Medium CVSS Score: 4.91 CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:N/I:L/A:N CVSS Audited: No NOTE: This issue has not been audited for accuracy by an application security professional or equivalent.

## CWE 404: Improper Resource Shutdown or Release

**Summary:** The program does not release or incorrectly releases a resource before it is made available for re-use.

**Details:** When a resource is created or allocated, the developer is responsible for properly releasing the resource as well as accounting for all potential paths of expiration or invalidation, such as a set period of time or revocation.[more details](#).

**Remediation:** Use a language that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.

**NOTE:** A value of "No" for "CVSS\_Audited" indicates that the CVSS Vector, Score and Severity are based on static analysis data and that the issues with this designation have not been audited for accuracy by an application security professional or equivalent.

Issue ID (CID) and Issue Type	Source File and Line Number
<a href="#">1536102</a> Resource leak on an exceptional path	/github_runner/intel/001/_work/Chipset_Repo/Chipset_Repo/chipset_inf_installer/Tools/DeviceManager/ DeviceQuery.cs:51 First Detected: Sep 5, 2023 10:03 AM CVSS Severity: Medium CVSS Score: 5.35 CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:L CVSS Audited: No NOTE:This issue has not been audited for accuracy by an application security professional or equivalent.

## 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.

### About CVSS

The Common Vulnerability Scoring System([CVSS](#)) is an open framework for communicating the characteristics and severity of software vulnerabilities. CVSS consists of three metric groups: Base, Temporal, and Environmental. The Base group represents the intrinsic qualities of a vulnerability, the Temporal group reflects the characteristics of a vulnerability that change over time, and the Environmental group represents the characteristics of a vulnerability that are unique to a user's environment. The Base metrics produce a score ranging from 0.0 to 10.0, which can then be modified by scoring the Temporal and Environmental metrics. A CVSS score is also represented as a vector string, a compressed textual representation of the values used to derive the score. This document provides a collection of examples of vulnerabilities scored using CVSS v3.0.

### About FIRST and CVSS-SIG

CVSS is owned and managed by FIRST.Org, Inc.([FIRST](#)), a US-based non-profit organization, whose mission is to help computer security incident response teams across the world. FIRST reserves the right to update CVSS and this document periodically at its sole discretion. While FIRST owns all right and interest in CVSS, it licenses it to the public freely for use, subject to the conditions below. Membership in FIRST is not required to use or implement CVSS. FIRST does, however, require that any individual or entity using CVSS give proper attribution, where applicable, that CVSS is owned by FIRST and used by permission. Further, FIRST requires as a condition of use that any individual or entity which publishes scores conforms to the guidelines described in this document and provides both the score and the scoring vector so others can understand how the score was derived.

### CVSS 3.0 Base Vector Calculations and Static Analysis

Coverity CVSS base vector scores take input from a development team-provided configuration file to provide certain application-specific base vector components (AV ,AC, PR, UI) and in addition determine the remaining base vector components (S, C, I, A) based on static analysis data for each individual software weakness (per CWE definitions). This provides the initial CVSS base vector, score and severity for each potential vulnerability identified by Coverity static analysis. Coverity static analysis results know a lot about a potential vulnerability type, and can provide, with proper per-application configuration file, a good initial CVSS base score, which in many cases will be acceptable. However, these initial CVSS base vectors and scores may not provide full and complete accuracy in all cases. Synopsys recommends that all Coverity static analysis based CVSS base vectors and scores require human review by a security professional or development team application expert responsible for the application code in question to ensure the best possible accuracy for the CVSS vector and score for each vulnerability identified. Issues identified by Coverity static analysis that do not have a CWE number receive a CVSS base vector that equates to a CVSS score of 0.

### 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 2017](#). The OWASP Top 10 List is referenced by many standards including MITRE, PCI DSS, DISA, and the FTC.



---

## The CWE/SANS Top 25 List

The SANS Institute is a cooperative research and education organization made up security experts from around the world. SANS is a major source of information on computer security and makes available an extensive collection of research documentation. It also operates the Internet's early security vulnerability warning system, the Internet Storm Center. The [2019 CWE/SANS Top 25 Most Dangerous Software Errors](#) is a list of the most common and critical errors that can lead to software vulnerabilities as published by this organization.

## About Synopsys Software Integrity Group

Synopsys Software Integrity Group ([Synopsys](#)) is a leading provider of quality and security testing solutions. The 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, also maintains a free, cloud based analysis platform, called Scan, for the Open Source Community.