Author

CAST



**OWASP 2017 TOP 10**

**Detailed Report**

Application Name –

Version –

CAST AIP -

|  |
| --- |
|  |
|  |

Monday, xx July 2012

My Application Name

Version Number

My CAST Version

# Table of Content

Table of Content

1. Introduction

1.1. Application Characteristics

2. Security Violation Overview

2.1. OWASP -2017 Top 10 vulnerabilities

2.2. OWASP -2017 A1 - Injection

2.3. OWASP -2017 A2 – Broken Authentication

2.4. OWASP -2017 A3 – Sensitive Data Exposure

2.5. OWASP -2017 A4 – XML External Entities (XXE)

2.6. OWASP -2017 A5 – Broken Access Control

2.7. OWASP -2017 A6 – Security Misconfiguration

2.8. OWASP -2017 A7 – Cross-Site Scripting (XSS)

2.9. OWASP -2017 A8 – Insecure Deserialization

2.10. OWASP -2017 A9 – Using Components with known Vulnerabilities

2.11. OWASP -2017 A10 – Insufficient Logging & Monitoring

3. Security Violation Details

3.1. OWASP -2017 A1 - Injection

3.2. OWASP -2017 A2 – Broken Authentication

3.3. OWASP -2017 A3 – Sensitive Data Exposure

3.4. OWASP -2017 A4 – XML External Entities (XXE)

3.5. OWASP -2017 A5 – Broken Access Control

3.6. OWASP -2017 A6 – Security Misconfiguration

3.7. OWASP -2017 A7 – Cross-Site Scripting (XSS)

3.8. OWASP -2017 A8 – Insecure Deserialization

3.9. OWASP -2017 A9 – Using Components with known Vulnerabilities

3.10. OWASP -2017 A10 – Insufficient Logging & Monitoring

4. Appendix

4.1. About CAST Software Intelligence

4.2. About CAST Security

# Introduction

This assessment is an effort to determine the security health of the application and identify some of the root causes of current Security concerns, as well as any risks of future degradation. This assessment uses the CAST Application Intelligence Platform (AIP) to automatically scan the implementation of these applications to review the architecture, design, and code against OWASP standards.

CAST AIP adapts the quality rules from best-in-class industry standards (OWASP, CWE, CISQ). With its unique ability to perform dataflow and system-level analysis (From Presentation layer to Database layer), CAST provides the most accurate security findings, reducing a lot of false positives.

## Application Characteristics

This assessment is focused solely on the technical implementation of the said application (user interface to database), with no investigation of the functionality.

|  |  |
| --- | --- |
| Name | Value |
| kLoC | 504 |
| Files | 6,586 |
| Classes | 593 |
| SQL Art. | 0 |
| Tables | 119 |

*Fig 1: Application Technology characteristics Table 1: Application characteristics*

# Security Violation Overview

This section provides a summary of the most severe security vulnerability identified in the structural quality analysis and measurement by CAST AIP against the OWASP 2017 standard. Details about OWASP Security Standard can be found [here](https://www.owasp.org/index.php/Top_10-2017_Top_10).

## OWASP -2017 Top 10 vulnerabilities

The [OWASP Top 10](https://www.owasp.org/index.php/Top_10-2017_Top_10)focuses on identifying the most serious web application security risks for a broad array of organizations.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OWASP-2017 | Exploitability | Weakness Prevalence | Weakness Detectability | Technical Impact |
| A1 - Injection | Easy | Common | Easy | Severe |
| A2 - Broken Authentication | Easy | Common | Average | Severe |
| A3 - Sensitive Data Exposure | Average | Widespread | Average | Severe |
| A4 - XML External Entities (XXE) | Average | Common | Easy | Severe |
| A5 - Broken Access Control | Average | Common | Average | Severe |
| A6 - Security Misconfiguration | Easy | Widespread | Easy | Moderate |
| A7 - Cross-Site Scripting (XSS) | Easy | Widespread | Easy | Moderate |
| A8 - Insecure Deserialization | Difficult | Common | Average | Severe |
| A9 - Using Components with Known Vulnerabilities | Average | Widespread | Average | Moderate |
| A10 - Insufficient Logging & Monitoring | Average | Widespread | Difficult | Moderate |

List of OWASP -2017 rules that had any findings in this application.

|  |  |  |  |
| --- | --- | --- | --- |
| OWASP-2017 | Total Vulnerabilities | Added Vulnerabilities | Removed Vulnerabilities |
| A1 | 0 | 0 | 0 |
| A3 | 0 | 0 | 0 |
| A6 | 0 | 0 | 0 |
| A7 | 0 | 0 | 0 |
| A… | 0 | 0 | 0 |

*Table 2: OWASP 2017 Top 10 Rules*

## OWASP -2017 A1 - Injection

This category of rules primarily deals with issues such as - 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. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

List of A1-Injection Vulnerabilities that had any findings in this application.

|  |  |  |  |
| --- | --- | --- | --- |
| CAST Rules | Total Vulnerabilities | Added Vulnerabilities | Removed Vulnerabilities |
| Rule 1 | 0 | 0 | 0 |
| Rule 2 | 0 | 0 | 0 |
| Rule 3 | 0 | 0 | 0 |
| Rule 4 | 0 | 0 | 0 |
| Rule 5 | 0 | 0 | 0 |

*Table 3: A1 - Injection Vulnerabilities*

## OWASP -2017 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.

List of A2-Broken Authentication vulnerabilities that had any findings in this application.

|  |  |  |  |
| --- | --- | --- | --- |
| CAST Rules | Total Vulnerabilities | Added Vulnerabilities | Removed Vulnerabilities |
| Rule 1 | 0 | 0 | 0 |
| Rule 2 | 0 | 0 | 0 |
| Rule 3 | 0 | 0 | 0 |
| Rule 4 | 0 | 0 | 0 |
| Rule 5 | 0 | 0 | 0 |

*Table 4: A2- Broken Authentication Vulnerabilities*

## OWASP -2017 A3 – Sensitive Data Exposure

Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. 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.

List of A3-Sensitive Data Exposure vulnerabilities that had any findings in this application.

|  |  |  |  |
| --- | --- | --- | --- |
| CAST Rules | Total Vulnerabilities | Added Vulnerabilities | Removed Vulnerabilities |
| Rule 1 | 0 | 0 | 0 |
| Rule 2 | 0 | 0 | 0 |
| Rule 3 | 0 | 0 | 0 |
| Rule 4 | 0 | 0 | 0 |
| Rule 5 | 0 | 0 | 0 |

*Table 4: A3- Sensitive Data Exposure Vulnerabilities*

## OWASP -2017 A4 – XML External Entities (XXE)

Many older or poorly configured XML processors evaluate external entity references within XML documents. 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.

List of A4 – XML External Entities vulnerabilities that had any findings in this application.

|  |  |  |  |
| --- | --- | --- | --- |
| CAST Rules | Total Vulnerabilities | Added Vulnerabilities | Removed Vulnerabilities |
| Rule 1 | 0 | 0 | 0 |
| Rule 2 | 0 | 0 | 0 |
| Rule 3 | 0 | 0 | 0 |
| Rule 4 | 0 | 0 | 0 |
| Rule 5 | 0 | 0 | 0 |

*Table 6: A4 – XML External Entities vulnerabilities*

## OWASP -2017 A5 – Broken Access Control

Restrictions on what authenticated users are allowed to do are often not properly enforced. 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.

List of A5 – Broken Access Control vulnerabilities that had any findings in this application.

|  |  |  |  |
| --- | --- | --- | --- |
| CAST Rules | Total Vulnerabilities | Added Vulnerabilities | Removed Vulnerabilities |
| Rule 1 | 0 | 0 | 0 |
| Rule 2 | 0 | 0 | 0 |
| Rule 3 | 0 | 0 | 0 |
| Rule 4 | 0 | 0 | 0 |
| Rule 5 | 0 | 0 | 0 |

*Table 7: A5 – Broken Access Control vulnerabilities*

## OWASP -2017 A6 – Security Misconfiguration

Security misconfiguration is the most commonly seen issue. 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. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.

List of A6-Security Misconfiguration Vulnerabilities that had any findings in this application.

|  |  |  |  |
| --- | --- | --- | --- |
| CAST Rules | Total Vulnerabilities | Added Vulnerabilities | Removed Vulnerabilities |
| Rule 1 | 0 | 0 | 0 |
| Rule 2 | 0 | 0 | 0 |
| Rule 3 | 0 | 0 | 0 |
| Rule 4 | 0 | 0 | 0 |
| Rule 5 | 0 | 0 | 0 |

*Table 8: A6- Security Misconfiguration Vulnerabilities*

## OWASP -2017 A7 – Cross-Site Scripting (XSS)

XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. 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.

List of A7 – Cross- Site Scripting rules that had any findings in this application

|  |  |  |  |
| --- | --- | --- | --- |
| CAST Rules | Total Vulnerabilities | Added Vulnerabilities | Removed Vulnerabilities |
| Rule 1 | 0 | 0 | 0 |
| Rule 2 | 0 | 0 | 0 |
| Rule 3 | 0 | 0 | 0 |
| Rule 4 | 0 | 0 | 0 |
| Rule 5 | 0 | 0 | 0 |

*Table 9: A7- Cross-Site Scripting Vulnerabilities*

## OWASP -2017 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.

List of A8 – Insecure Deserialization rules that had any findings in this application

|  |  |  |  |
| --- | --- | --- | --- |
| CAST Rules | Total Vulnerabilities | Added Vulnerabilities | Removed Vulnerabilities |
| Rule 1 | 0 | 0 | 0 |
| Rule 2 | 0 | 0 | 0 |
| Rule 3 | 0 | 0 | 0 |
| Rule 4 | 0 | 0 | 0 |
| Rule 5 | 0 | 0 | 0 |

*Table 10: A8- Insecure Deserialization vulnerabilities*

## OWASP -2017 A9 – Using Components with known Vulnerabilities

Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

List of A9 – Using Components with known vulnerabilities rules that had any findings in this application -

|  |  |  |  |
| --- | --- | --- | --- |
| CAST Rules | Total Vulnerabilities | Added Vulnerabilities | Removed Vulnerabilities |
| Rule 1 | 0 | 0 | 0 |
| Rule 2 | 0 | 0 | 0 |
| Rule 3 | 0 | 0 | 0 |
| Rule 4 | 0 | 0 | 0 |
| Rule 5 | 0 | 0 | 0 |

*Table 10: A9 – Using Components with known vulnerabilities*

## OWASP -2017 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. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

List of A10 – Insufficient Logging & Monitoring rules that had any findings in this application -

|  |  |  |  |
| --- | --- | --- | --- |
| CAST Rules | Total Vulnerabilities | Added Vulnerabilities | Removed Vulnerabilities |
| Rule 1 | 0 | 0 | 0 |
| Rule 2 | 0 | 0 | 0 |
| Rule 3 | 0 | 0 | 0 |
| Rule 4 | 0 | 0 | 0 |
| Rule 5 | 0 | 0 | 0 |

*Table 12: A10 – Insufficient Logging & Monitoring vulnerabilities*

# Security Violation Details

## OWASP -2017 A1 - Injection

|  |
| --- |
| Object name |
| Violation #1 |
| …. |
|  |

## OWASP -2017 A2 – Broken Authentication

|  |
| --- |
| Object name |
| Violation #1 |
| …. |

## OWASP -2017 A3 – Sensitive Data Exposure

|  |
| --- |
| Object name |
| Violation #1 |
| …. |
|  |
|  |

## OWASP -2017 A4 – XML External Entities (XXE)

|  |
| --- |
| Object name |
| Violation #1 |
| …. |
|  |

## OWASP -2017 A5 – Broken Access Control

|  |
| --- |
| Object name |
| Violation #1 |
| …. |
|  |

## OWASP -2017 A6 – Security Misconfiguration

|  |
| --- |
| Object name |
| Violation #1 |
| …. |

## OWASP -2017 A7 – Cross-Site Scripting (XSS)

|  |
| --- |
| Object name |
| Violation #1 |
| …. |

## OWASP -2017 A8 – Insecure Deserialization

|  |
| --- |
| Object name |
| Violation #1 |
| …. |

## OWASP -2017 A9 – Using Components with known Vulnerabilities

|  |
| --- |
| Object name |
| Violation #1 |
| …. |

## OWASP -2017 A10 – Insufficient Logging & Monitoring

|  |
| --- |
| Object name |
| Violation #1 |
| …. |

# Appendix

## About CAST Software Intelligence

Software Intelligence creates understanding into software architecture, end to end transaction flows, data access patterns and more, helping teams work confidently and faster. Hundreds of companies rely on CAST Software Intelligence to improve end-user satisfaction and time-to-market, prevent business disruption and reduce cost, enabling them to move past today’s obstacles and to tackle the next wave of innovation.

[Click here](https://www.castsoftware.com/software-intelligence) for more information about CAST Software Intelligence.

## About CAST Security

Cyber risk and application security require a proactive and intelligence-driven approach. CAST Software Intelligence shifts insight into security strategy blind spots before development starts. With its unique ability to do dataflow and system-level analysis, CAST provides the most accurate security findings, reducing a lot of false positives. CAST Security rules are adapted from best-in-class industry standards – CISQ, CWE, and OWASP.

To find out more about CAST Security, [click here](https://www.castsoftware.com/use-cases/application-security).