APPLICATION SECURITY REPORT

XMPro

Feb 17 2022 16:02 GMT+11:00 AEDT

Subscription Manager

Static, Dynamic, and SCA

Inside This Report

EXECUTIVE SUMMARY

Application: Subscription Manager

POLICY NAME: Veracode Recommended Medium + ...

✗ Rules ✓ Scan Requirements ✗ Grace Period

OPEN FINDINGS: 27
FINDINGS IMPACTING POLICY: 0

Findings Violating Policy Rules

Remediation Status	Count
Open	**0**
Proposed Mitigations	0

Open Findings Impacting Policy

Very High	High	Medium	Low	Very Low	Informational
0	0	0	0	0	0

Top 5 CWEs Found

CWE ID	Issue
757	Selection of Less-Secure Algorithm During Negotiation ...
526	Exposure of Sensitive Information Through Environmental Variables
1174	ASP.NET Misconfiguration: Improper Model Validation
404	Improper Resource Shutdown or Release
915	Improperly Controlled Modification of ...

Scans Included in Application

Static
20220217.1 - QA
Feb 17 2022 13:13 GMT+11:00 AEDT
Score 98

Dynamic
SM QA 08/02/2022
Feb 14 2022 11:09 GMT+11:00 AEDT
Score 95

SCA Findings Summary

398 Third-Party Components

Components Impacting Policy: 4

Vulnerabilities Impacting Policy: 12

High 6 Medium 6

Third-Party Component License Risk

High	4
Medium	43
Low	360
Unassessable	5
Total	**412**

POLICY EVALUATION

Policy Name	Veracode Recommended Medium + SCA
Revision	1
Policy Status	Did Not Pass
Description	Veracode provides default policies to make it easier for organizations to begin measuring their applications against policies. Veracode Recommended Policies are available for customers as an option when they are ready to move beyond the initial bar set by the Veracode Transitional Policies. The policies are based on the Veracode Level definitions.

REMEDIATION

Flaw Severity	Grace Period	Flaws Exceeding	Policy Status
Very High	0	0	Passed
High	0	0	Passed
Medium	0	0	Passed
Low	0	0	Passed
Very Low	0	0	Passed
Informational	0	0	Passed

SCAN REQUIREMENTS

Scan Type	Frequency	Last performed	Policy Status
Static Analysis	Quarterly	Feb 17 2022 08:13 GMT+11:00 AEDT	Passed

RULES

Rule Type	Requirement	Result	Policy Status
Minimum Veracode Level	VL3 + SCA	VL3	Did not pass
Min Analysis Score	70	95	Passed
Max Severity	High	Flaws found: 0	Passed
Disallow Component Blocklist	Prevent an application from passing policy if blockliste...	0 Blocklisted	Passed
Disallow Vulnerabilities by Severity	High and Above Not Allowed	4 Components	Did not pass

SCORE

Rule Type	Grace Period	Flaws Exceeding	Policy Status
Meet Minimum Score	0	0	Passed

CHANGES FROM LAST SCAN

Latest Scan

Static Scan

Scan Name:	20220217.1 - QA
Completed:	Feb 17 2022 13:13 GMT+11:00 AEDT
Score:	98

Dynamic Scan

Scan Name:	SM QA 08/02/2022
Completed:	Feb 14 2022 11:09 GMT+11:00 AEDT
Score:	95

Prior Scan

Static Scan

Scan Name:	20220207.1 - QA
Completed:	Feb 07 2022 12:47 GMT+11:00 AEDT
Score:	98

Dynamic Scan

Scan Name:	SM Staging 02/10/21
Completed:	Oct 06 2021 16:59 GMT+11:00 AEDT
Score:	95

Changes in Scope of Static Scan

MODULES SELECTED IN SCAN

	Latest Scan	Prior Scan
Module Name	JS files within SM.zip XMIdentity.dll	JS files within SM.zip XMIdentity.dll
Total Modules Selected	2	2
Scan Size	5.7MB	5.7MB

*Not Included in prior scan

Changes in Scope of Dynamic Scan

NEW MODULES

Setting	Latest Scan	Prior Scan
Max Links to Crawl	5000	5000
Successful Logins	11 of 18	5 of 5
User Agent	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.0 Safari/537.36/Veracode Security Scan/support@veracode.com	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.0 Safari/537.36/Veracode Security Scan/support@veracode.com
Scan Duration	4 Hours 20 Minutes	2 Hours 26 Minutes

VERACODE'S METHODOLOGY

ABOUT VERACODE'S METHODOLOGY

The Veracode Application Security Platform uses static analysis, software composition analysis, and dynamic analysis (for web applications) to identify software security findings in your applications. Using static and dynamic analysis as well as software composition analysis helps reduce false negatives and detect a broader range of security findings. Veracode Static Analysis models the application into an intermediate representation, which is then analyzed for security flaws using a set of automated security tests. Veracode Dynamic Analysis uses an automated penetration testing technique to detect security flaws at runtime. Veracode Software Composition Analysis (SCA) helps you build an inventory of your open-source components to identify vulnerabilities. Veracode leverages a continuous improvement process and rapid updates to its SaaS-based scanning technology to ensure the lowest false-positive rates in the industry. The end result is an accurate list of security findings for the classes of automated scans applied to the application.

APPLICATION SECURITY POLICIES

The Veracode Platform allows an organization to define and enforce a uniform application security policy across all applications in its portfolio. The elements of an application security policy include the:

- Target Veracode Level (LV) for the application
- Types of flaws that should not be in the application (which may be defined by flaw severity, flaw category, CWE, or a common standard including OWASP, CWE/SANS Top 25, or PCI)
- Minimum Veracode security score
- Required scan types and frequencies
- Any customized rules around specifically blacklisted components or CVE severities
- Grace period within which any policy-relevant flaws should be fixed

Policy Constraints

Policies have three main constraints that can be applied: rules, required scans, and remediation grace periods.

Evaluating Applications Against a Policy

When an application is evaluated against a policy, it can receive one of four assessments:

Not assessed	The application has not yet had a scan published.
Passed	The application has passed all the aspects of the policy, including rules, required scans, and grace period.
Did not pass	The application has not completed all required scans, has not achieved the target Veracode Level, or has one or more policy relevant flaws that have exceeded the grace period to fix.
Conditional pass	The application has one or more policy relevant flaws that have not yet exceeded the grace period to fix.

UNDERSTAND VERACODE LEVELS

The Veracode Level (VL) achieved by an application is determined by type of testing performed on the application, and the severity and types of static, dynamic, and manual flaws detected. A minimum security score (defined below) is also required for each level.

There are five Veracode Levels, denoted as VL1, VL2, VL3, VL4, and VL5. VL1 is the lowest level and is achieved by demonstrating that security testing, automated static or dynamic, is utilized during the SDLC. VL5 is the highest level and is achieved by performing automated and manual testing and removing all significant flaws. The Veracode Levels VL2, VL3, and VL4 provide a continuum of increasing software assurance between VL1 and VL5.

For IT staff operating applications, Veracode Levels can be used to set application security policies. For deployment scenarios of different business criticality, differing VLs should be made requirements. For example, the policy for applications that manage credit card transactions and, therefore, have PCI compliance requirements, should be VL5. A medium business criticality internal application could have a policy requiring VL3.

Software developers can decide which VL they want to achieve, based on the requirements of their customers. Developers of software that is mission critical to most of their customers will want to achieve VL5. Developers of general purpose business software may want to achieve VL3 or VL4. Once the software has achieved a Veracode Level, it can be communicated to customers through a Veracode report or through the Veracode Directory on the Veracode web site.

Criteria for Achieving Veracode Levels

The following table defines the details for achieving each Veracode Level. The criteria for all columns: Flaw Severities Not Allowed, Flaw Categories not Allowed, Testing Required, and Minimum Score.

*Dynamic is only an option for web applications.

Veracode Level	Flaw Severities Not Allowed	Testing Required*	Minimum Score
VL5	V.High, High, Medium	Static AND Manual	90
VL4	V.High, High, Medium	Static	80
VL3	V.High, High	Static	70
VL2	V.High	Static OR Dynamic OR Manual	60
VL1		Static OR Dynamic OR Manual	

When multiple testing techniques are used, it is likely that not all testing will be performed on the exact same build. If that is the case, the latest test results from a specific technique will be used to calculate the current Veracode Level. After six months, test results are deemed out of date and will no longer be used to calculate the current Veracode Level.

BUSINESS CRITICALITY

The foundation of the Veracode rating system is the concept that more critical applications require higher security quality scores to be acceptable risks. Less business-critical applications can tolerate lower security quality. The business criticality is dictated by the typical deployed environment and the value of data used by the application. Factors that determine business criticality are: reputation damage, financial loss, operational risk, sensitive information disclosure, personal safety, and legal

violations.

US. Govt. OMB Memorandum M-04-04; NIST FIPS Pub. 199

Business Criticality	Description
Very High	Mission critical for business/safety of life and limb on the line
High	Exploitation causes serious brand damage and financial loss with long term business impact
Medium	Applications connected to the internet that process financial or private customer information
Low	Typically internal applications with non-critical business impact
Very Low	Applications with no material business impact

Business Criticality Definitions

Very High (BC5) This criticality is typically an application where the safety of life or limb is dependent on the system. It is mission critical that the application maintain 100% availability for the long-term viability of the project or business. Examples are control software for industrial, transportation or medical equipment or critical business systems such as financial trading systems.

High (BC4) This criticality is typically an important multi-user business application reachable from the internet and it is critical that the application maintain high availability to accomplish its mission. Exploitation of high criticality applications cause serious brand damage, and business/financial loss, and could lead to long-term business impact.

Medium (BC3) This criticality is typically a multi-user application connected to the internet or any system that processes financial or private customer information. Exploitation of medium-criticality applications typically result in material business impact resulting in some financial loss, brand damage or business liability. An example is a financial services company's internal 401K management system.

Low (BC2) This criticality is typically an internal-only application that requires low levels of application security such as authentication to protect access to non-critical business information and prevent IT disruptions. Exploitation of low criticality applications may lead to minor levels of inconvenience, distress, or IT disruption. An example internal system is a conference room reservation or business card order system.

Very Low (BC1) Applications that have no material business impact if its confidentiality, data integrity and availability are affected. Code security analysis is not required for applications at this business criticality, and security spending should be directed to other higher criticality applications.

SCORING METHODOLOGY

The Veracode scoring system, Security Quality Score, is built on the foundation of two industry standards: the Common Weakness Enumeration (CWE) and Common Vulnerability Scoring System (CVSS v3.0). CWE provides the dictionary of security flaws and CVSS v3.0 provides the foundation for computing severity, based on the potential Confidentiality, Integrity, and Availability impact of a flaw, if exploited.

The Security Quality Score is a single score from 0 to 100, where 0 is the most insecure application and 100 is an application with no detectable security flaws. The score calculation includes non-linear factors

so that, for instance, a single Severity 5 flaw is weighted more heavily than five Severity 1 flaws. Thus, each additional flaw at a given severity contributes progressively less to the score.

Veracode assigns a severity level to each flaw type based on three foundational application security requirements: Confidentiality, Integrity, and Availability. Each of the severity levels reflects the potential business impact if a security breach occurs across one or more of these security dimensions.

Confidentiality Impact

According to CVSS v3.0, this metric measures the impact on confidentiality if an exploit should occur using the vulnerability on the target system. At the weakness level, the scope of the Confidentiality in this model is within an application and is measured at three levels of impact: None, Partial, and Complete.

Integrity Impact

This metric measures the potential impact on integrity of the application being analyzed. Integrity refers to the trustworthiness and guaranteed veracity of information within the application. Integrity measures are designed to protect data from unauthorized modification. When the integrity of a system is sound, it is fully protected from unauthorized modification of its contents.

Availability Impact

This metric measures the potential impact on availability if a successful exploit of the vulnerability is performed on a target application. Availability refers to the accessibility of information resources. Almost exclusive to this domain are denial-of- service vulnerabilities. Attacks that compromise authentication and authorization for application access, application memory, and administrative privileges are examples of impact on the availability of an application.

SECURITY QUALITY SCORE CALCULATION

The overall Security Quality Score is computed by aggregating impact levels of weaknesses from static, dynamic, and manual test results within an application and representing the score on a 100-point scale. This score does not predict vulnerability potential as much as it enumerates the security from those weaknesses and their impact levels within the application code.

The Raw Score formula puts weights on each static, dynamic, or manual flaw based on its impact level. These weights are exponential and determined by empirical analysis by Veracode's application security experts, with validation from industry experts. The score is normalized to a scale of 0 to 100, where a score of 100 is an application with 0 detected flaws using the analysis technique selected for the application's business criticality.

UNDERSTAND SEVERITY, EXPLOITABILITY, AND REMEDIATION EFFORT

Severity and exploitability are two different measures of the seriousness of a flaw. Severity is defined in terms of the potential impact to confidentiality, integrity, and availability of the application as defined in the CVSS v3.0, and exploitability is defined in terms of the likelihood or ease with which a flaw can be exploited. A high-severity flaw with a high likelihood of being exploited by an attacker is potentially more dangerous than a high severity flaw with a low likelihood of being exploited.

Remediation effort, also called Complexity of Fix, is a measure of the likely effort required to fix a flaw. Together with severity, the remediation effort is used to give Fix First guidance to the developer.

VERACODE SEVERITIES

Veracode severities are defined in the following table.

Severity	Description
Very High	The offending line or lines of code pose a very serious weakness and are easy targets for an attacker. The code should be modified immediately to avoid potential attacks.
High	The offending line or lines of code have significant weakness, and the code should be modified immediately to avoid potential attacks.
Medium	A weakness of average severity. These should be fixed in high assurance software. A fix for this weakness should be considered after fixing the very high and high severities for medium assurance software.
Low	This is a low-priority weakness that will have a small impact on the security of the software. Fixing should be considered for high-assurance software. Medium and low-assurance software can ignore these flaws.
Very Low	Minor problems that some high-assurance software may want to be aware of. These flaws can be safely ignored in medium and low-assurance software.
Informational	Issues that have no impact on the security quality of the application but which may be of interest to the reviewer.

Informational Findings

Informational severity findings are items observed in the analysis of the application that have no impact on the security quality of the application but may be interesting to the reviewer for other reasons. These findings may include code quality issues, API usage, and other factors.

Informational severity findings have no impact on the security quality score of the application and are not included in the summary tables of flaws for the application.

EXPLOITABILITY

Each flaw instance in a static scan may receive an exploitability rating. The rating is an indication of the intrinsic likelihood that the flaw may be exploited by an attacker. Veracode recommends that the exploitability rating be used to prioritize flaw remediation within a particular group of flaws with the same severity and difficulty of fix classification.

The possible exploitability ratings include:

Exploitability	Description
V. Unlikely	Very unlikely to be exploited
Unlikely	Unlikely to be exploited
Neutral	Neither likely nor unlikely to be exploited.
Likely	Likely to be exploited
V. Likely	Very likely to be exploited

Note: All reported flaws found via Veracode Dynamic Analysis are assumed to be exploitable because the dynamic scan actually executes the attack in question and verifies that it is valid.

EFFORT/COMPLEXITY OF FIX

Each flaw instance receives an effort/complexity of fix rating based on the classification of the flaw. The effort/complexity of fix rating is given on a scale of 1 to 5, as follows:

Effort/Complexity of Fix	Description
5	Complex design error. Requires significant redesign.
4	Simple design error. Requires redesign and up to 5 days to fix.
3	Complex implementation error. Fix is approx. 51-500 lines of code. Up to 5 days to fix.
2	Implementation error. Fix is approx. 6-50 lines of code. 1 day to fix.
1	Trivial implementation error. Fix is up to 5 lines of code. One hour or less to fix.

Note: All reported flaws found via Veracode Dynamic Analysis are assumed to be exploitable because the dynamic scan actually executes the attack in question and verifies that it is valid.

COMMON WEAKNESS ENUMERATION (CWE)

The Common Weakness Enumeration (CWE) is an industry-standard classification of types of software weaknesses or flaws, that can lead to security problems. CWE is widely used to provide a standard taxonomy of software errors. Every flaw in a Veracode report is classified according to a standard CWE identifier.

More guidance and background about the CWE is available at http://cwe.mitre.org/data/index.html.

COMMON VULNERABILITIES AND EXPOSURES (CVE)

Common Vulnerabilities and Exposures (CVE) is a catalog of known security threats. CVE identifiers are unique identifiers for publicly known security vulnerabilities. CVEs are found in third-party components by Veracode Software Composition Analysis.

More guidance and background about CVE is available at https://cve.mitre.org/about/.

ABOUT MANUAL ASSESSMENTS

All Veracode Manual Penetration Testing is performed according to industry-standard testing methodologies, where applicable. The following table describes what testing methodology is used by test type and vulnerability types for manual penetration tests.

Test Type	Methodology	Vulnerabilities
Web Application/API	OWASP Testing Guide	OWASP Top 10/SANS Top 25
Mobile Application	OWASP Mobile Security Testing Guide	OWASP Mobile Top 10
Desktop or Thick-Client Application	OWASP recommended testing guidance and best practices	Application Logic
		Code Injection
		Local Storage
		Binary Exploitation and Reverse Engineering
		Excessive Privileges
		Unencrypted Storage of Sensitive Information
		Unencrypted Transmission of Sensitive Information
		Weak Encryption Implementations

Test Type	Methodology	Vulnerabilities
		Weak Assembly Controls
		Weak GUI Controls
		Weak or Default Passwords
Internet of Things (IoT) and Embedded Systems	OWASP IoT Testing Guide and other industry best practices	OWASP IoT Top 10
Infrastructure and Operations (DevOps Penetration Testing)	PTES (Penetration Testing Execution Standard), NIST SP 800-115, PCI DSS 11.3 (for PCI engagements)	Can vary depending on scope and rules of engagement

For more information on Veracode Manual Penetration Testing, refer to the Veracode Help Center.

Time-Boxed Testing

Veracode's Time-Boxed Manual Penetration Testing service tests as much of the application as possible within the number of days purchased. Veracode's Penetration Tester's follows testing methodologies noted in the table above. Vulnerabilities from Veracode Static and/or Veracode Dynamic Analysis scans that have been previously reported in the Veracode Platform are leveraged during Manual Penetration Testing, if available to the Penetration Tester.

In cases where time is overly constrained Veracode focuses on providing the most value for the time allotted. For smaller applications, less time may be needed to cover a majority of vulnerabilities, while other larger applications may require additional time. For this reason, Veracode Penetration Testers may choose to tailor the methodology to focus on higher priority, business relevant flaws. For example, if a three-day penetration test is purchased for a 500+ page application with complex business logic, the tester may choose to focus more on finding representative examples of higher risk flaws such as injection, authentication, and authorization flaws. In contrast, if this were a 10-day engagement, the tester would be able to cover the entire methodology in more adequate depth.

Traditional Scoped Manual Penetration Testing

For customers who have purchased MPT, which is scoped through a scoping questionnaire and scoping call, Veracode performs testing according to the scoped number of days using information provided from the customer from the scoping questionnaire and scoping call.

TERMS OF USE