



# Requirement Yogi

Requirement Yogi Web Application Penetration Retest  
Report

Report creation date: **March 16, 2026**

Testing period: **January 28, 2026 to February 16, 2026**

Prepared by: **Josh Durand**



# Table of contents

|   |          |
|---|----------|
| <b>Executive Summary</b> .....                  | <b>2</b> |
| <b>Engagement details</b> .....                 | <b>4</b> |
| <b>Findings table</b> .....                     | <b>5</b> |
| <b>Reporting and Methodology</b> .....          | <b>6</b> |
| <b>Appendix A - Risk and priority key</b> ..... | <b>7</b> |
| <b>Closing Statement</b> .....                  | <b>9</b> |

# Executive Summary

**Requirement Yogi** engaged Bugcrowd, Inc. to perform a Penetration Test that took place from January 28th, 2026, through February 16th, 2026.

The purpose of this engagement was to identify security vulnerabilities in the assets listed under the [Targets and Scope](#) section of the report. Once identified, each vulnerability was rated for technical impact defined in the [Findings Summary](#).

Bugcrowd's testing methodology involves several key stages:

- Initial reconnaissance to gather critical information about the target environment,
- Enumeration to identify potential attack vectors,
- Exploitation to test the impact and feasibility of identified vulnerabilities, and
- Post-exploitation to evaluate the extent of potential compromise

The tester(s) used a number of sophisticated tools, techniques, and procedures for each of these stages. Crucially, our approach leverages human ingenuity, creative problem-solving, and critical thinking to uncover complex vulnerabilities that might otherwise go undetected from scanners and scripts alone. The insights gained from this assessment are intended to provide a comprehensive understanding of the organization's security risks, enabling informed decision-making and effective prioritization of remediation efforts.

At the time of the original report, **9 vulnerabilities** were identified. **1 Critical, 1 Low, and 1 Informational vulnerability** were retested and could no longer be reproduced with the reproduction steps. These findings are now remediated. **1 Low vulnerability** was moved to **Not Applicable** due to it impacting portion of the Atlassian infrastructure that is not something that can be remediated. The remaining vulnerabilities include:

- **0 Critical**
- **0 Severe**
- **0 Moderate**
- **0 Low**
- **5 Informational**

At this time, **Bugcrowd has rated the risk to the Requirement Yogi's assets as Low**. Our rating is based on the severity of the findings disclosed within this report.

It is recommended that Requirement Yogi focus on Critical and Severe vulnerabilities first, with Moderate, Low, and Informational findings being fixed once the most critical issues are remediated.

Bugcrowd recommends that all Critical, Severe, and Moderate severity findings are retested once remediation activities are completed.



If not already implemented, Bugcrowd recommends taking the following high-level actions to further improve the overall security posture of the organization:

- Implement a secure development lifecycle such as Microsoft Secure Development Lifecycle (MSDL).
- Implement a static code analysis (SAST) tool into the development lifecycle to minimize the introduction of vulnerabilities in code.
- Provide ongoing training to developers to ensure that they are aware of secure development practices and emerging threats.

The continuation of this report contains technical details of the specific vulnerabilities that were discovered throughout the Penetration Test.

This report is just a summary of the information available and is a 'snapshot' in time of the state for the tested environment.

All details of the engagement's findings — comments, code, and any researcher provided remediation information — can be found in the Bugcrowd Crowdcontrol platform: <https://tracker.bugcrowd.com> (<https://tracker.bugcrowd.com>)

# Engagement details

## Scope of testing

Prior to penetration test launching, Bugcrowd worked with Requirement Yogi. to define the rules of the engagement, commonly known as the engagement brief, which includes the scope of work.

The following targets were considered explicitly in-scope for testing:

- website: <https://marketplace.atlassian.com/apps/1214094/requirement-yogi-requirements-management-for-jira?hosting=cloud&tab=overview>
- website: <https://marketplace.atlassian.com/apps/1212523/requirement-yogi-requirements-management-for-confluence?hosting=cloud&tab=overview>
- website: [app.requirementyogi.com](http://app.requirementyogi.com)
- api: [api.requirementyogi.com/api/](http://api.requirementyogi.com/api/)

All details of the engagement scope and full brief can be reviewed in each of the respective Engagement Settings pages found on the Bugcrowd Crowdcontrol platform.

# Findings table

The following table lists all validated findings identified through manual testing grouped by Vulnerability Rating Taxonomy (VRT):

| Vulnerability Rating Taxonomy (VRT)     | Title  | Priority      |
|---|--|---------------|
| <b>Server Security Misconfiguration</b> | <del>F001 — Publicly Exposed Keycloak Admin Console on Master Realm</del>        | <del>P1</del> |
|   | <del>F002 — No Rate Limiting on Login Attempts Enables Brute Force Attacks</del> | <del>P4</del> |
| <b>Other</b>                            | F003 - Password Reset Abuse Leads to Account Lockout and API Token Disruption    | P5            |
| <b>Server Security Misconfiguration</b> | F004 - User Enumeration via Email Change Endpoint                                | P5            |
|   | F005 - Fingerprinting Banner Disclosure  | P5            |
|   | <del>F006 — Missing CSP Header</del>   | <del>P5</del> |
|   | F007 - Lack of Rate Limiting Allows Multiple Verification Emails to Be Sent      | P5            |
|   | F008 - Email Enumeration via Registration Response                               | P5            |

# Reporting and Methodology

By leading with a best-in-class testing approach, Bugcrowd's methodology provides enhanced risk reduction while supporting critical compliance initiatives. A review of these standards follows.

Reviewed Organizational Methodology Standards:

- PCI DSS Requirement 11.2, 11.3.1, 11.3.3, 11.3.4
- NIST 800-115 - Technical Guide to Information Security Testing and Assessment 2.1 "Information Security Assessment Methodology"
- Open Source Security Testing Methodology Manual (OSSTMM)
- Penetration Testing Execution Standard (PTES)

In order to create a complete testing methodology, Bugcrowd has pulled from the following industry standard operational methodologies:

- OWASP Testing Guide (OTG)
- Web Application Hacker Handbook Methodology (WAHHM)
- Others where applicable (SANS Top 25, CREST, WASC, PTES)



# Appendix A - Risk and priority key

The following key is used to explain how Bugcrowd rates valid vulnerability submissions and their technical severity. As a trusted advisor Bugcrowd also provides common "next steps" for program owners per severity category.

| Technical severity  | Example vulnerability types  |
|---|--|
| <p><b>P1</b> <b>Critical</b></p> <p>Critical severity submissions (also known as "P1" or "Priority 1") are submissions that are escalated to Bugcrowd as soon as they are validated. These issues warrant the highest security consideration and should be addressed immediately. Commonly, submissions marked as Critical can cause financial theft, unavailability of services, large-scale account compromise, etc.</p>                  | <ul style="list-style-type: none"> <li>• Remote Code Execution</li> <li>• Vertical Authentication Bypass</li> <li>• XML External Entities Injection</li> <li>• SQL Injection</li> <li>• Insecure Direct Object Reference for a critical function</li> </ul>  |
| <p><b>P2</b> <b>Severe</b></p> <p>Severe severity submissions (also known as "P2" or "Priority 2") are vulnerability submissions that should be slated for fix in the very near future. These issues still warrant prudent consideration but are often not availability or "breach level" submissions. Commonly, submissions marked as Severe can cause account compromise (with user interaction), sensitive information leakage, etc.</p> | <ul style="list-style-type: none"> <li>• Lateral authentication bypass</li> <li>• Stored Cross-Site Scripting</li> <li>• Cross-Site Request Forgery for a critical function</li> <li>• Insecure Direct Object Reference for an important function</li> <li>• Internal Server-Side Request Forgery</li> </ul> |
| <p><b>P3</b> <b>Moderate</b></p> <p>Moderate severity submissions (also known as "P3" or "Priority 3") are vulnerability submissions that should be slated for fix in the major release cycle. These vulnerabilities can commonly impact single users but require user interaction to trigger or only disclose moderately sensitive information.</p>  | <ul style="list-style-type: none"> <li>• Reflected Cross-Site Scripting with limited impact</li> <li>• Cross-Site Request Forgery for an important function</li> <li>• Insecure Direct Object Reference for an unimportant function</li> </ul>   |
| <p><b>P4</b> <b>Low</b></p> <p>Low severity submissions (also known as "P4" or "Priority 4") are vulnerability submissions that should be considered for fix within the next six months. These vulnerabilities represent the least danger to confidentiality, integrity, and availability.</p>  | <ul style="list-style-type: none"> <li>• Cross-Site Scripting with limited impact</li> <li>• Cross-Site Request Forgery for an unimportant function</li> <li>• External Server-Side Request Forgery</li> </ul>   |
| <p><b>P5</b> <b>Informational</b></p> <p>Informational submissions (also known as "P5" or "Priority 5") are vulnerability submissions that are valid but out-of-scope or are "won't fix" issues, such as best practices.</p>  | <ul style="list-style-type: none"> <li>• Lack of code obfuscation</li> <li>• Autocomplete enabled</li> <li>• Non-exploitable SSL issues</li> </ul>   |



## Bugcrowd's Vulnerability Rating Taxonomy



More detailed information regarding our vulnerability classification can be found at: <https://bugcrowd.com/vrt>  
(<https://bugcrowd.com/vrt>)



Bugcrowd Inc.  
300 California St  
Suite 220  
San Francisco, CA 94104  
(888)361-9734

**16 Mar 2026**

# Closing Statement

## Introduction

This report reflects retesting of the Requirement Yogi assets originally tested between the dates of **January 28th, 2026** to **February 16th, 2026**. The purpose of this assessment was to identify security issues that could adversely affect the integrity of Requirement Yogi. The assessment was performed under the guidelines provided in the statement of work between Requirement Yogi and Bugcrowd. This document provides a high-level overview of the testing performed and the test results.

## Pen Test Portfolio Overview

Our suite of Pen Test Products are powered by the Bugcrowd Platform, enabling rapid setup, launch, and real-time results.

While Bugcrowd offers both continuous and on-demand penetration testing options, it is important to note that this document represents a point-in-time evaluation of security posture. Security threats and attacker techniques evolve rapidly, and the results of this assessment are not intended to represent an endorsement of the adequacy of current security measures against future threats. This document contains information in summary form and is therefore intended for general guidance only; it is not intended as a substitute for detailed research or the exercise of professional judgment. The information presented here should not be construed as professional advice or service.

## Testing Methods

This security assessment leveraged researchers that used a combination of proprietary, public, automated, and manual test techniques throughout the assessment. Commonly tested vulnerabilities include code injection, cross-site request forgery, cross-site scripting, insecure storage of sensitive data, authorization/authentication vulnerabilities, business logic vulnerabilities, and more.

## Summary of Retest Findings

The summary of Bugcrowd's retest findings are as follows:

| Severity      | Number of findings |
|---------------|--------------------|
| Critical      | 0                  |
| Severe        | 0                  |
| Moderate      | 0                  |
| Low           | 0                  |
| Informational | 5                  |