

Communardo

User Profiles for Jira Web Application Penetration Test Report

Report creation date: **May 12, 2026**

Testing period: **May 04, 2026** to **May 08, 2026**

Prepared by: **Josh Durand**

CONFIDENTIAL INFORMATION • Do not share outside of organization

bugcrowd

Table of contents

Executive Summary 2
Engagement details 4
Findings table 5
Reporting and Methodology 6
Appendix A - Risk and priority key 7
Appendix B - Vulnerability details 9
Closing Statement 10

Executive Summary

Communardo engaged Bugcrowd, Inc. to perform a Penetration Test that took place from May 4th, 2026, through May 8th, 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 assessment leveraged a combination of industry-standard automated tools and in-depth manual testing techniques to analyze how the application handles externally sourced identity data, processes administrator and end-user input, enforces authorization boundaries across Jira project and global permission scopes, and manages errors. Manual testing was performed using an intercepting proxy to inspect and manipulate HTTP requests and responses associated with the OAuth/Entra ID synchronization flow, attribute mapping configuration endpoints, user profile rendering APIs, directory query parameters, and tenant-scoped data retrieval paths. Targeted payloads were used to assess the application for injection-based vulnerabilities such as Cross-Site Scripting (XSS), including stored XSS through poisoned Entra ID attributes (display name, position, department) that flow into rendered Jira profile views, HTML injection, parameter tampering, open redirects via Teams deep-links, SSRF in the directory connector, and improper input validation. Authorization controls were evaluated to ensure that configuration screens were restricted to Jira administrators, that synchronized user data respected the source tenant's visibility rules, and that no cross-tenant data leakage was possible within Atlassian's multi-tenant Cloud environment.

Testing covered all available logic paths, including the Entra ID directory connection and token lifecycle, attribute mapping and synchronization workflows, user profile display within Jira issue and project contexts, directory filtering by position, department, and location, Microsoft Teams presence and link handling, header customization options, and administrator configuration screens, in alignment with the defined testing methodology and the documented functionality of the application.

Throughout the engagement, no security vulnerabilities were identified. All tested inputs, including attribute values originating from the external Entra ID directory, were handled with appropriate server-side validation and context-aware output encoding when rendered into Jira views. OAuth token handling and the directory synchronization flow were observed to follow secure practices, with no token leakage, replay, or scope-escalation conditions identified. Access control mechanisms consistently enforced expected Jira administrator and tenant-isolation boundaries, and no unauthorized data access or cross-tenant leakage was observed. Error handling behavior was found to be secure and did not expose

sensitive system information, stack traces, internal Entra ID identifiers, or implementation details. Defensive controls effectively mitigated common web application attack techniques as well as identity-integration-specific risks.

At the time of this report, **0 vulnerabilities** were identified, including:

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

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

It is recommended that Communardo 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 Communardo. 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/1212508/user-profiles-for-jira-microsoft-attributes-sync?hosting=cloud&tab=overview

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

No data available

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
P1 **Critical** 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.	• Remote Code Execution • Vertical Authentication Bypass • XML External Entities Injection • SQL Injection • Insecure Direct Object Reference for a critical function
P2 **Severe** 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.	• Lateral authentication bypass • Stored Cross-Site Scripting • Cross-Site Request Forgery for a critical function • Insecure Direct Object Reference for an important function • Internal Server-Side Request Forgery
P3 **Moderate** 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.	• Reflected Cross-Site Scripting with limited impact • Cross-Site Request Forgery for an important function • Insecure Direct Object Reference for an unimportant function
P4 **Low** 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.	• Cross-Site Scripting with limited impact • Cross-Site Request Forgery for an unimportant function • External Server-Side Request Forgery
P5 **Informational** 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.	• Lack of code obfuscation • Autocomplete enabled • Non-exploitable SSL issues

VRT

Bugcrowd's Vulnerability Rating Taxonomy

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

Appendix B - Vulnerability details

This section outlines the full submission data for each valid finding. These findings are unaltered from their original state from the researcher. Due to the competitive nature and gamification of crowd-sourced security assessments, some typos or grammar errors may occur. Each finding is headlined with the submission title and priority followed by more detailed vulnerability information based on the type of finding submitted. Several other fields may appear based on the context and VRT classification selected by a researcher.

Such details may include the following:

Title

A brief summary of the vulnerability.

Bug URL

This is the full URL/URI of where the vulnerability took place

Description

This section appears above the "Reference Number" as a free form area for the researcher to describe the context of the submission. The full HTTP(S) request that triggered the vulnerability is normally in this field, including all its associated headers and cookie information.

Target

The target that is vulnerable.

Remediation advice

Steps to mitigate or compensate for the vulnerability.

Vulnerability Rating Taxonomy (VRT)

The Vulnerability Rating Taxonomy (https://bugcrowd.com/vrt) is the baseline guide used for classifying technical severity.

Jira ID

This is the ID to a customer Jira ticket associated with a vulnerability submission.

Custom fields

Some customers add extra attributes to vulnerability submissions, for example, "Product Start Version" or "Product End Version".

Submission ID

Submission ID is visible to customers and researchers.

bugcrowd

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

12 May 2026

Closing Statement

Introduction

This report reflects testing of the Communardo assets targets between the dates of **May 4th, 2026** to **May 8th, 2026** . The purpose of this assessment was to identify security issues that could adversely affect the integrity of Communardo. The assessment was performed under the guidelines provided in the statement of work between Communardo 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 Findings

The summary of Bugcrowd's findings are as follows:

Severity	Number of findings
Critical	0
Severe	0
Moderate	0
Low	0
Informational	0