

Communardo

Viewtracker - Analytics for Confluence Web Application Penetration Test Report

Report creation date: **April 28, 2026**

Testing period: **April 21, 2026** to **April 24, 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

F001 - Spring Boot Actuator Management Interface Exposed to Public Internet - Unauthenticated Enumeration Possible 6

Reporting and Methodology 9

Appendix A - Risk and priority key 10

Appendix B - Vulnerability details 12

Closing Statement 13

Executive Summary

Communardo engaged Bugcrowd, Inc. to perform a Penetration Test that took place from April 21st, 2026, through April 24th, 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 this report, **1 vulnerabilities** were identified, including:

- **0 Critical**
- **0 Severe**
- **0 Moderate**
- **0 Low**
- **1 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: Viewtracker - Analytics for Confluence - Cloud - https://marketplace.atlassian.com/apps/28637/viewtracker-analytics-for-confluence?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

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

Vulnerability Rating Taxonomy (VRT)	Title	Priority
Other	F001 - Spring Boot Actuator Management Interface Exposed to Public Internet - Unauthenticated Enumeration Possible	P5

Other

P5

F001 - Spring Boot Actuator Management Interface Exposed to Public Internet - Unauthenticated Enumeration Possible

Bug URL

https://de.viewtracker.aws.bitvoodoo.cloud/actuator/heapdump
(https://de.viewtracker.aws.bitvoodoo.cloud/actuator/heapdump)

Description

The Viewtracker application exposes Spring Boot Actuator management endpoints on its public-facing hostname de.viewtracker.aws.bitvoodoo.cloud. All /actuator/* sub-endpoints are routable from the public internet and respond with 401 Unauthorized, confirming they exist and are actively processing requests.
Spring Boot Actuator is an operations and diagnostics framework that, when enabled, exposes sub-endpoints such as:

- /actuator/env - environment variables, including database URLs, credentials, and API keys injected at runtime
- /actuator/heapdump - a full JVM heap dump containing all in-memory objects, secrets, and session tokens
- /actuator/mappings - complete HTTP route table, exposing all application endpoints including undocumented ones
- /actuator/beans - full Spring bean registry with configuration details
- /actuator/loggers - runtime log-level manipulation
- /actuator/configprops - all configuration properties including potentially sensitive values

These endpoints are intended exclusively for internal monitoring and should never be reachable from the public internet. The current 401 responses indicate authentication is enforced, however this single control is insufficient given the attack history of this endpoint class. Several CVEs and misconfigurations have resulted in authentication bypass on Actuator endpoints in the past (e.g., CVE-2022-22963, Spring4Shell ecosystem, and various Spring Security filter ordering bugs).

Business Impact

Viewtracker stores Confluence user PII, account IDs, page view history, and full search queries — across multiple enterprise tenants on shared AWS infrastructure. If the `401` on Actuator endpoints is bypassed through a future Spring Security CVE or token leakage, an attacker can pull environment variables containing plaintext database credentials, AWS IAM keys, and Atlassian Connect shared secrets. A compromised shared secret allows forging valid JWTs for any connected tenant, giving full access to all customer data with no further exploitation required. The `/actuator/heapdump` endpoint

additionally exposes a full JVM memory dump in a single request, containing live session tokens and in-flight user data. Given the multi-tenant architecture, a single exploitation event constitutes a GDPR Article 33 reportable breach affecting every customer on the DE-region instance.

Steps to Reproduce

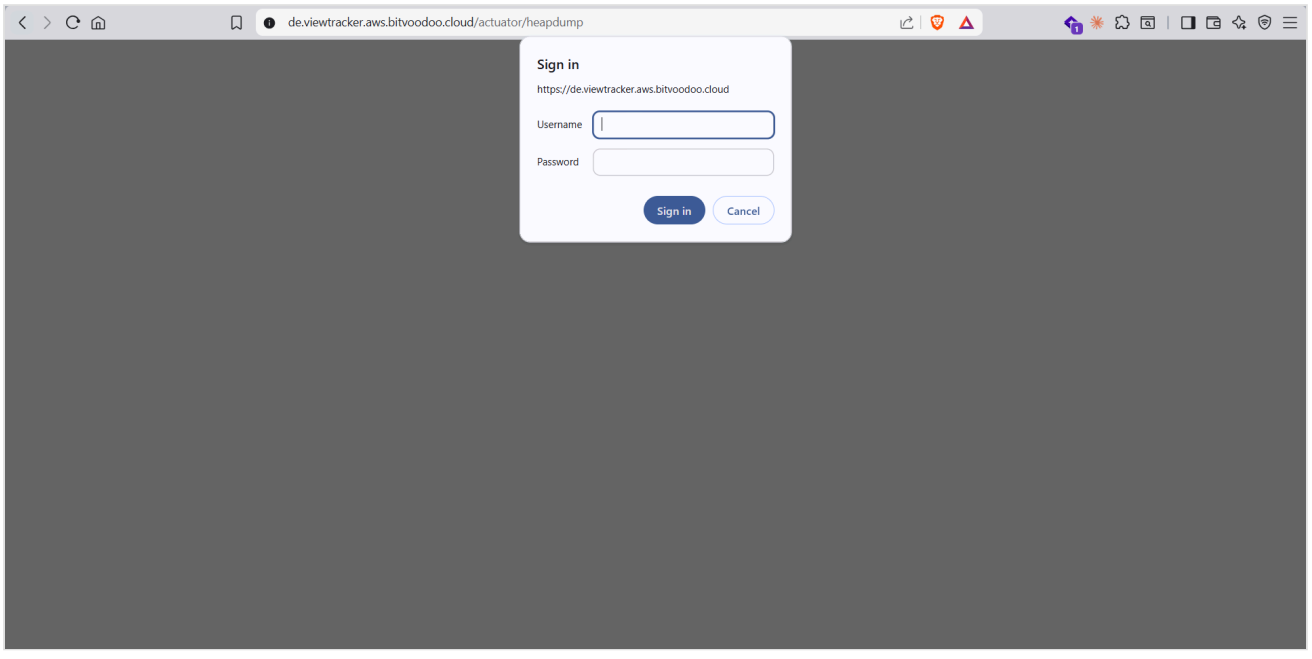
1. Open any browser (no authentication, no tools required).
2. Navigate to `https://de.viewtracker.aws.bitvoodoo.cloud/actuator/heapdump` the browser immediately presents a native Sign in dialog with Username and Password fields for https://de.viewtracker.aws.bitvoodoo.cloud, confirming the endpoint is live and publicly accessible from the internet.
3. Click Cancel to dismiss the dialog. Repeat the same for each sensitive sub-endpoint to confirm all are reachable:

```
https://de.viewtracker.aws.bitvoodoo.cloud/actuator/env
https://de.viewtracker.aws.bitvoodoo.cloud/actuator/beans
https://de.viewtracker.aws.bitvoodoo.cloud/actuator/mappings
https://de.viewtracker.aws.bitvoodoo.cloud/actuator/configprops
https://de.viewtracker.aws.bitvoodoo.cloud/actuator/shutdown
```

4. Observe that every endpoint above presents the same login prompt, none return a 404 or are blocked at the network perimeter. The management interface is fully internet-routable and actively soliciting credentials from any unauthenticated user on the internet.

Proof of Concept (PoC)

The screenshot(s) below demonstrate(s) the vulnerability:



Remediations

- Bind the actuator server to the loopback interface, preventing all external access regardless of authentication state. Add to application.properties: management.server.address=127.0.0.1 management.server.port=8090
- Restrict which actuator endpoints are exposed. Only health and info should ever be enabled in production: management.endpoints.web.exposure.include=health,info management.endpoint.health.show-details=never management.endpoint.heapdump.enabled=false management.endpoint.env.enabled=false management.endpoint.shutdown.enabled=false
- Place the actuator management interface behind a network-level control (VPC security group, WAF rule, or internal load balancer) so it is only reachable from your monitoring infrastructure (Prometheus, Datadog, etc.) and not from the public internet under any circumstances, even in the event of a Spring Security misconfiguration.

Target
Viewtracker - Analytics for Confluence - Cloud - https://marketplace.atlassian.com/apps/28637/viewtracker-analytics-for-confluence?hosting=cloud&tab=overview

Vulnerability Rating Taxonomy (VRT)
Other

Submission ID
3a932558-06f1-4a1c-ac06-9406a136f69c

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

28 Apr 2026

Closing Statement

Introduction

This report reflects testing of the Communardo assets targets between the dates of **April 21st, 2026** to **April 24th, 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	1