Bugcrowd Security Program for Atlassian

Summary Report

Report created on November 11, 2024
Report date range: July 31, 2024 - September 30, 2024
Prepared by: Ben Howe, bhowe@atlassian.com

bugcrowd

Table of contents

Executive summary

Atlassian engaged Bugcrowd, Inc. to launch and implement a Security Program comprising one or more engagements on the Bugcrowd Platform.

The testing type, scope, targets, and duration of the testing done for each engagement were specified in the Engagement Brief that was created during planning.

During testing, each discovered vulnerability was validated and assigned a priority level for remediation. The complete list of vulnerabilities uncovered during the engagement(s), along with their potential impact, is shown in the Findings section.

This report shows testing for Atlassian's targets during the period between **07/31/2024 – 09/30/2024**.

For this Security Program comprising one or more engagements, submissions were received from **68** unique testers.

The continuation of this document summarizes the findings, analysis, and recommendations from the engagement(s) performed by Bugcrowd for {org-name}.
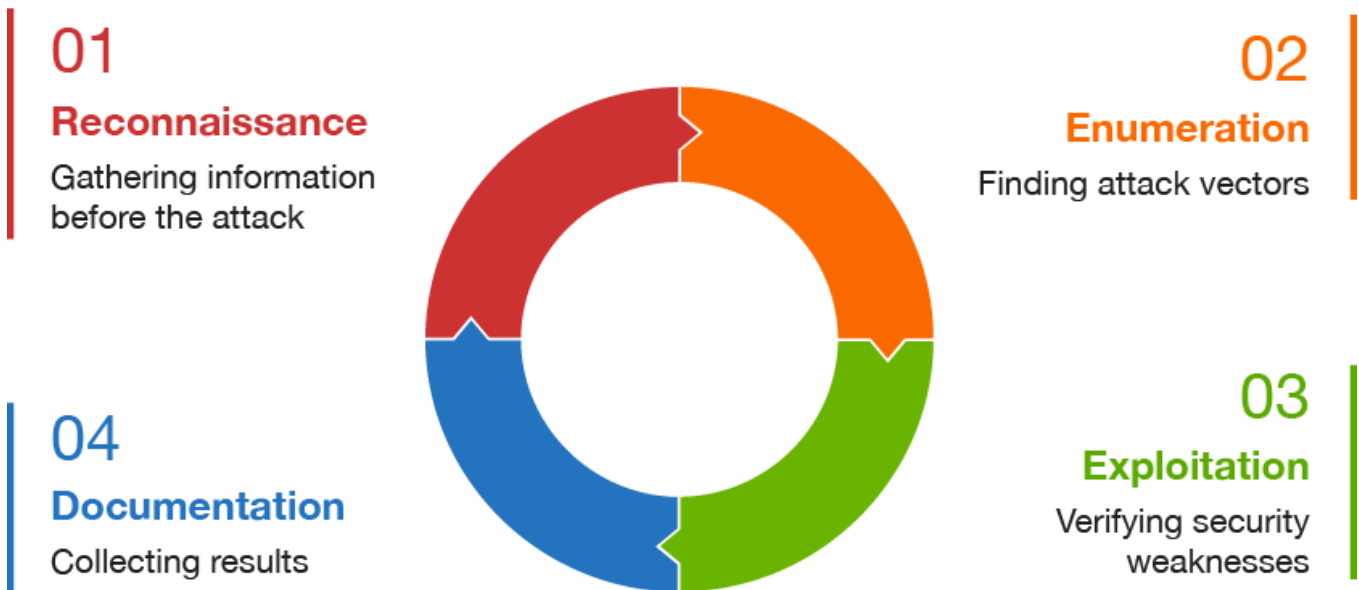
This report is a summary of the information available. All details of the engagement's findings — comments, code, and any tester provided remediation information — can be found in the Bugcrowd platform (https://tracker.bugcrowd.com)

Reporting and methodology

Background

The strength of crowdsourced testing lies in multiple researchers, the pay-for-results model, and the varied methodologies that the researchers implement. To this end, researchers are encouraged to use their own individual methodologies on various Engagements.

The workflow of every Bugcrowd Security Program Engagement can be divided into the following four phases:

01 Reconnaissance — Gathering information before the attack

02 Enumeration — Finding attack vectors

03 Exploitation — Verifying security weaknesses

04 Documentation — Collecting results

Depending on the type of engagement, testers may have adopted one or more methodologies to guide their testing, including:
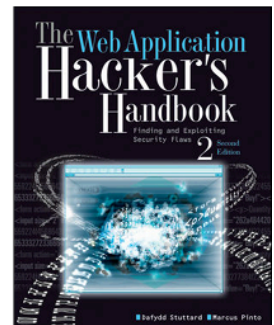
Engagement details

This report incorporates following engagements within the **Atlassian** Security Program as an aggregate:

- Trello

Targets and scope

Bugcrowd worked with **Atlassian** to define the Rules of Engagement, targets, duration, and scope for each engagement prior to launch. The following targets were considered explicitly in scope for testing:

- Calendar Power-Up
- Voting Power-Up
- Trello Mobile App for Android
- api.trello.com
- Trello Desktop Client
- Trello Mobile App for iOS
- *.trello.services
- Card Aging Power-Up
- Butler for Trello
- trello.com
- Microsoft Teams Integration
- List Limits Power-Up
- Trello Third Party Powerups

All details including the scope can be reviewed in the settings of each engagement.

Findings Summary

The following chart shows all valid assessment findings from the engagement by technical severity.
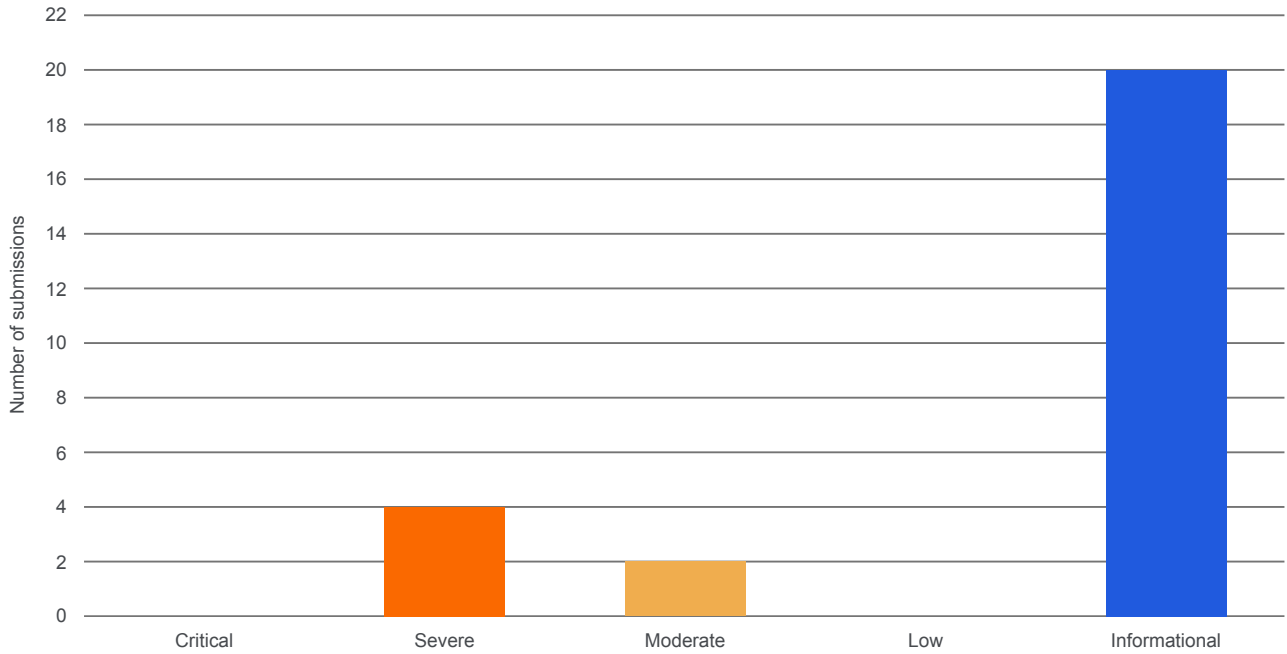
Risk and priority key

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** |

Critical **P1**

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

Severe **P2**

High 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 High 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

Moderate **P3**

Medium 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

Low **P4**

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

Informational **P5**

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/vulnerability-rating-taxonomy
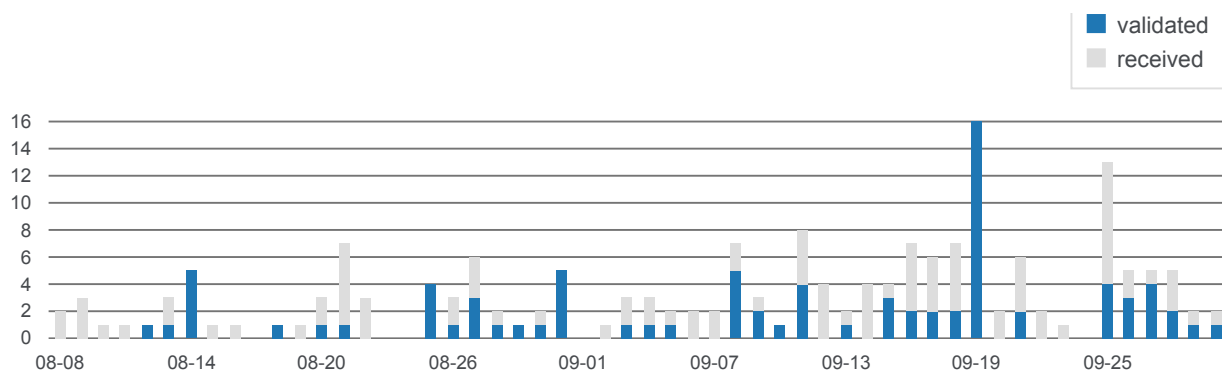

Appendix

Included in this appendix are auxiliary metrics and insights into the engagement(s). This includes information regarding submissions over time, payouts and prevalent issue types.
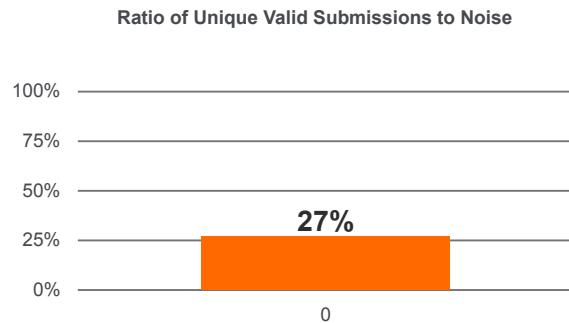
Submissions over time

The timeline below shows submissions received and validated by the Bugcrowd team:

Submissions signal

A total of **97** submissions were received, with **26** unique valid issues discovered. Bugcrowd identified **19** informational submissions, **13** duplicate submissions, removed **58** invalid submissions, and is processing **0** submissions. The ratio of unique valid submissions to noise was **27%**.

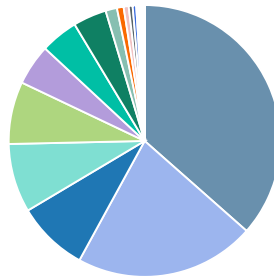| Submission Outcome	Count
Valid	26
Informational	19
Invalid	58
Duplicate	13
Processing	0
Total	**97**

Ratio of Unique Valid Submissions to Noise

Bug types overview

This distribution across bug types for the engagement(s) only includes unique and valid submissions.

Average On-Demand Program

- ■ Other
- ■ Cross-Site Scripting (XSS)
- ■ Broken Access Control (BAC)
- ■ Server Security Misconfiguration
- ■ Broken Authentication and Session Management
- ■ Cross-Site Request Forgery (CSRF)
- ■ Sensitive Data Exposure
- ■ Server-Side Injection
- ■ Unvalidated Redirects and Forwards
- ■ Mobile Security Misconfiguration
- ■ Application-Level Denial-of-Service (DoS)
- ■ Insufficient Security Configurability
- ■ Insecure Direct Object References (IDOR)
- ■ Missing Function Level Access Control
- ■ Insecure OS/Firmware
- ■ Insecure Data Storage
- ■ Using Components with Known Vulnerabilities
- ■ Automotive Security Misconfiguration
- ■ Broken Cryptography
- ■ Client-Side Injection
- ■ External Behavior
- ■ Privacy Concerns
- ■ Lack of Binary Hardening
- ■ Insecure Data Transport
- ■ Network Security Misconfiguration
- ■ Cryptographic Weakness
- ■ Indicators of Compromise

bugcrowd

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

November 11 2024

Closing Statement

Introduction

This report shows testing of Atlassian between the dates of **07/31/2024 and 09/30/2024**. During this time, **68** testers from Bugcrowd submitted a total of **97** vulnerability submissions against Atlassian's targets. The purpose of this testing was to identify security issues that could adversely affect the integrity of Atlassian. Testing focused on the following:

Engagements overview

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 engagement(s) selected and activated testers who 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.

During the Engagement, Bugcrowd discovered the following:

Technical Severity	Count
Critical vulnerabilities	0
Severe vulnerabilities	4
Moderate vulnerabilities	2
Low vulnerabilities	0
Informational vulnerabilities	20