Atlassian

Collaboration tools for teams of all sizes

Atlassian

Bugcrowd Ongoing program results

Report created on July 05, 2021

Report date range: April 01, 2021 - June 30, 2021

bugcrowd

Prepared by

bmarriott@atlassian.com

Table of contents

Executive summary

Atlassian engaged Bugcrowd, Inc. to perform an Ongoing Bounty Program, commonly known as a crowd-sourced penetration test.

An Ongoing Bounty Program is a cutting-edge approach to an application assessment or penetration test. Traditional penetration tests use only one or two personnel to test an entire scope of work, while an Ongoing Bounty leverages a crowd of security researchers. This increases the probability of discovering esoteric issues that automated testing cannot find and that traditional vulnerability assessments may miss in the same testing period.

The purpose of this program was to identify security vulnerabilities in the targets listed in the targets and scope section. Once identified, each vulnerability was rated for technical impact defined in the findings summary section of the report.

This report shows testing for **Atlassian's** targets during the period of: **04/01/2021 – 06/30/2021**.

For this Ongoing Program, submissions were received from **152** unique researchers.

The continuation of this document summarizes the findings, analysis, and recommendations from the Ongoing Bounty Program performed by Bugcrowd for **Atlassian**.

This report is just a summary of the information available.

All details of the program's findings — comments, code, and any researcher provided remediation information — can be found in the Bugcrowd Crowdcontrol platform.

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 Bugcrowd Ongoing programs.

The workflow of every penetration test can be divided into the following four phases:

01
Reconnaissance
Gathering information before the attack

02
Enumeration
Finding attack vectors

04
Documentation
Collecting results

03
Exploitation
Verifying security weaknesses

Bugcrowd researchers who perform web application testing and vulnerability assessment usually subscribe to a variety of methodologies following the highlighted workflow, including the following:
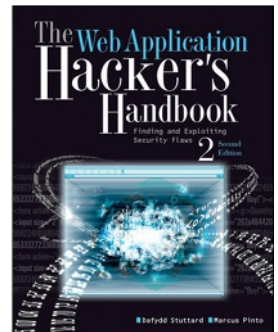
bugcrowd

Scope

Prior to the Ongoing program launching, Bugcrowd worked with Atlassian to define the Rules of Engagement, commonly known as the program brief, which includes the scope of work. The following targets were considered explicitly in scope for testing:

> All details of the program scope and full program brief can be reviewed in the [Program Brief](#).

```
Atlassian Access (https://admin.atlassian.com/atlassian-
access)
```

```
Atlassian Admin (https://admin.atlassian.com/)
```

```
Atlassian Identity (https://id.atlassian.com/login)
```

```
Atlassian Start (https://start.atlassian.com)
```

```
Bitbucket Cloud including Bitbucket Pipelines
(https://bitbucket.org)
```

```
Confluence Cloud (bugbounty-test-<bugcrowd-
name>.atlassian.net/wiki)
```

```
Confluence Cloud Premium (bugbounty-test-<bugcrowd-
name>.atlassian.net/wiki)
```

```
Confluence Cloud Mobile App for Android
```

```
Confluence Cloud Mobile App for iOS
```

```
Jira Cloud Mobile App for Android
```

```
Jira Cloud Mobile App for iOS
```

```
Jira Service Management Cloud (bugbounty-test-<bugcrowd-
name>.atlassian.net)
```

```
Jira Software Cloud (bugbounty-test-<bugcrowd-
name>.atlassian.net)
```

```
Jira Work Management Cloud formerly Jira Core (bugbounty-test-
<bugcrowd-name>.atlassian.net)
```

```
Any associated *.atlassian.com or *.atl-paas.net domain that can
be exploited DIRECTLY from the *.atlassian.net instance
```

```
Atlassian Marketplace (https://marketplace.atlassian.com)
```

```
Bitbucket Server
```

Confluence Server

Crowd

Jira Core Server

Jira Service Management Server

Jira Software Server

*.atlastunnel.com

Any other *.atlassian.com or *.atl-paas.net domain that cannot
be exploited directly from a *.atlassian.net instance

Bamboo

Confluence Companion App for macOS and Windows

Confluence Server Mobile App for Android

Confluence Server Mobile App for iOS

Crucible

FishEye

Jira Server Mobile App for Android

Jira Server Mobile App for iOS

Sourcetree for macOS and Windows
(https://www.sourcetreeapp.com/)

Other - (all other Atlassian targets)

Findings summary

Findings by severity

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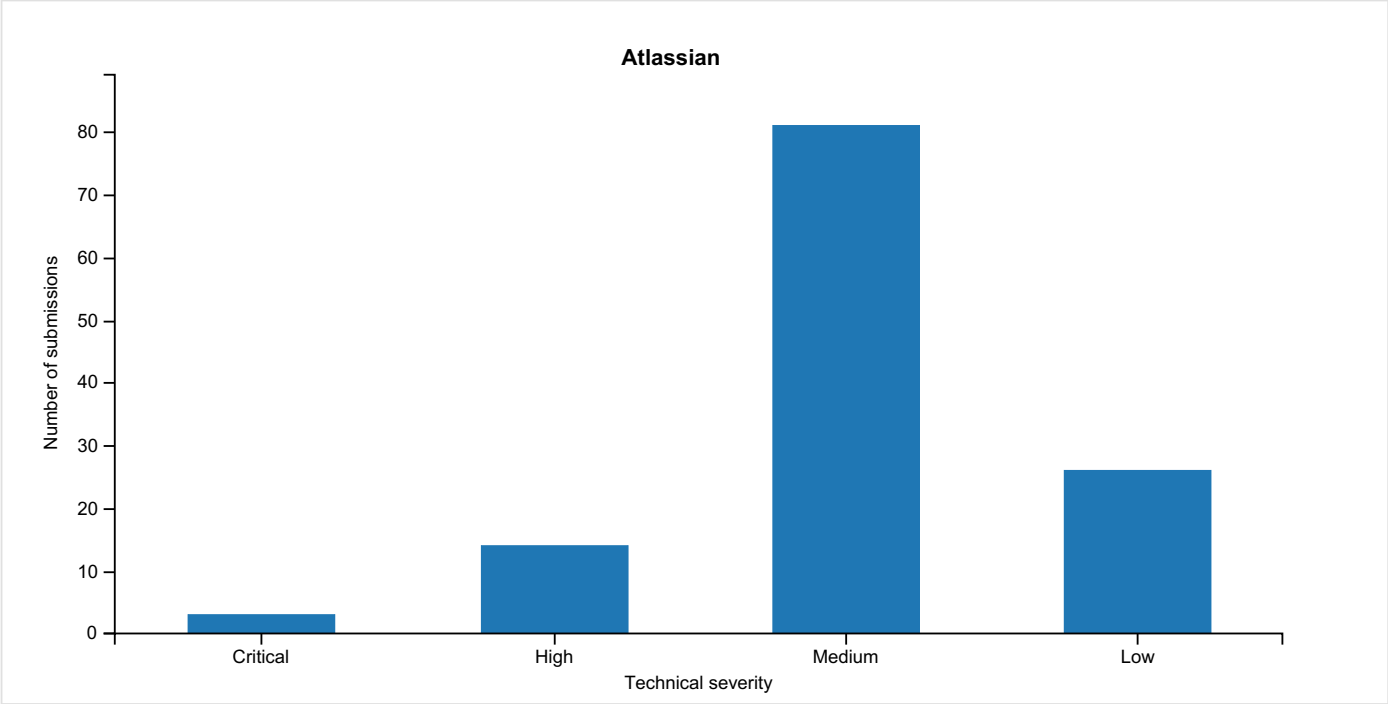
Risk and priority key

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

Critical

Critical severity submissions (also known as "P1" or "Priority 1") are submissions that are escalated to **Atlassian** 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.

High

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.

Medium

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.

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.

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.

EXAMPLE VULNERABILITY TYPES

- Remote Code Execution
- Vertical Authentication Bypass
- XML External Entities Injection
- SQL Injection
- Insecure Direct Object Reference for a critical function

- 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

- Reflected Cross-Site Scripting with limited impact
- Cross-Site Request Forgery for an important function
- Insecure Direct Object Reference for an unimportant function

- Cross-Site Scripting with limited impact
- Cross-Site Request Forgery for an unimportant function
- External Server-Side Request Forgery

- 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

Appendix

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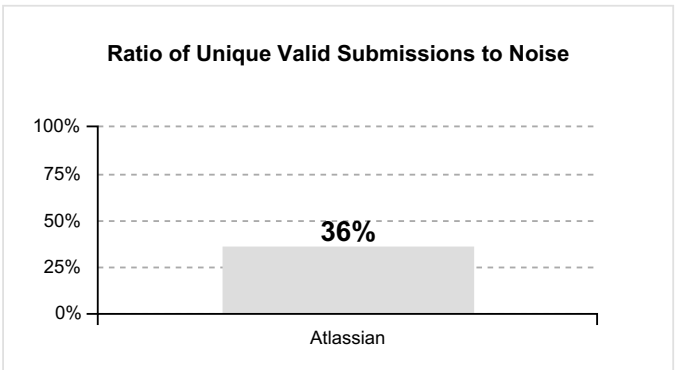
Submissions over time

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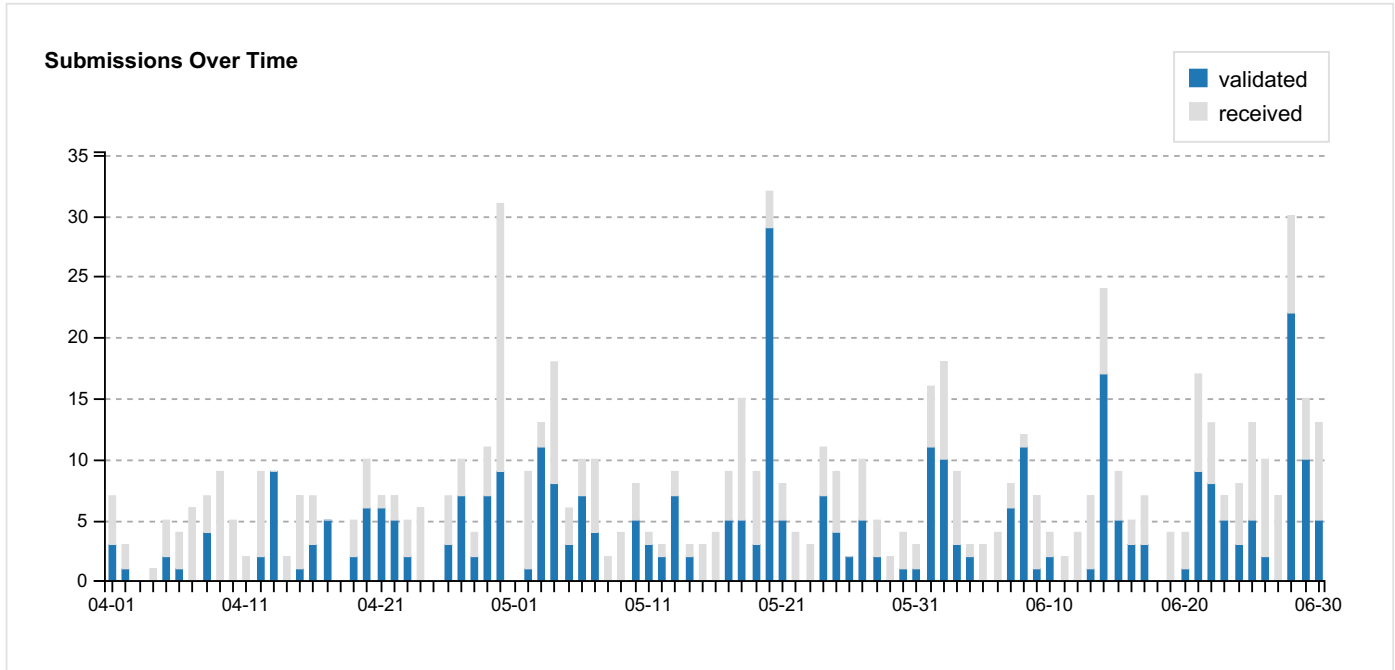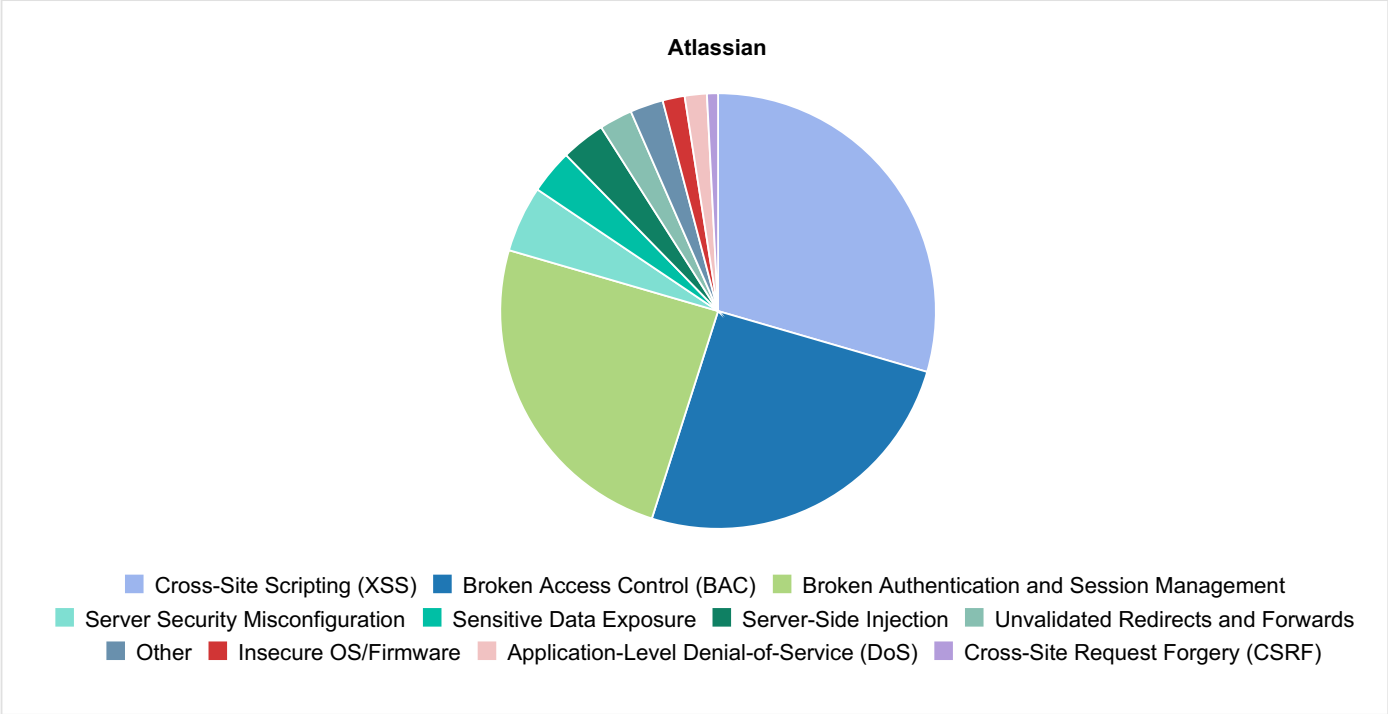

Submissions signal

A total of **366** submissions were received, with **128** unique valid issues discovered. Bugcrowd identified **26** informational submissions, **74** duplicate submissions, removed **131** invalid submissions, and is processing **7** submissions. The ratio of unique valid submissions to noise was **36%**.

Submission Outcome	Count
Valid	128
Informational	26
Invalid	131
Duplicate	74
Processing	7
Total	**366**

Ratio of Unique Valid Submissions to Noise

36%

Bug types overview

This distribution across bug types for the Ongoing program only includes unique and valid submissions.

Atlassian

- Cross-Site Scripting (XSS)
- Broken Access Control (BAC)
- Broken Authentication and Session Management
- Server Security Misconfiguration
- Sensitive Data Exposure
- Server-Side Injection
- Unvalidated Redirects and Forwards
- Other
- Insecure OS/Firmware
- Application-Level Denial-of-Service (DoS)
- Cross-Site Request Forgery (CSRF)

Closing statement

July 05, 2021

Bugcrowd Inc.
921 Front St
Suite 100
San Francisco, CA 94111

Introduction

This report shows testing of **Atlassian** between the dates of **04/01/2021 - 06/30/2021.** During this time, **152** researchers from Bugcrowd submitted a total of **366** vulnerability submissions against **Atlassian's** targets. The purpose of this assessment was to identify security issues that could adversely affect the integrity of Atlassian. Testing focused on the following:

1. **Atlassian Access (https://admin.atlassian.com/atlassian-access)**
2. **Atlassian Admin (https://admin.atlassian.com/)**
3. **Atlassian Identity (https://id.atlassian.com/login)**
4. **Atlassian Start (https://start.atlassian.com)**
5. **Bitbucket Cloud including Bitbucket Pipelines (https://bitbucket.org)**
6. **Confluence Cloud (bugbounty-test-<bugcrowd-name>.atlassian.net/wiki)**
7. **Confluence Cloud Premium (bugbounty-test-<bugcrowd-name>.atlassian.net/wiki)**
8. **Confluence Cloud Mobile App for Android**
9. **Confluence Cloud Mobile App for iOS**
10. **Jira Cloud Mobile App for Android**
11. **Jira Cloud Mobile App for iOS**
12. **Jira Service Management Cloud (bugbounty-test-<bugcrowd-name>.atlassian.net)**
13. **Jira Software Cloud (bugbounty-test-<bugcrowd-name>.atlassian.net)**
14. **Jira Work Management Cloud formerly Jira Core (bugbounty-test-<bugcrowd-name>.atlassian.net)**
15. **Any associated *.atlassian.com or *.atl-paas.net domain that can be exploited DIRECTLY from the *.atlassian.net instance**
16. **Atlassian Marketplace (https://marketplace.atlassian.com)**
17. **Bitbucket Server**
18. **Confluence Server**
19. **Crowd**
20. **Jira Core Server**
21. **Jira Service Management Server**
22. **Jira Software Server**
23. ***.atlastunnel.com**
24. **Any other *.atlassian.com or *.atl-paas.net domain that cannot be exploited directly from a *.atlassian.net instance**
25. **Bamboo**

26. **Confluence Companion App for macOS and Windows**
27. **Confluence Server Mobile App for Android**
28. **Confluence Server Mobile App for iOS**
29. **Crucible**
30. **FishEye**
31. **Jira Server Mobile App for Android**
32. **Jira Server Mobile App for iOS**
33. **Sourcetree for macOS and Windows (https://www.sourcetreeapp.com/)**
34. **Other - (all other Atlassian targets)**

The assessment was performed under the guidelines provided in the statement of work between **Atlassian** and Bugcrowd. This letter provides a high-level overview of the testing performed, and the result of that testing.

Ongoing Program Overview

An Ongoing Program is a novel approach to a penetration test. Traditional penetration tests use only one or two researchers to test an entire scope of work, while an Ongoing Program leverages a crowd of security researchers. This increases the probability of discovering esoteric issues that automated testing cannot find and that traditional vulnerability assessments may miss, in the same testing period.

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

During the program, Bugcrowd discovered the following:

Count	Technical Severity	
3	Critical	vulnerabilities
14	High	vulnerabilities
81	Medium	vulnerabilities
26	Low	vulnerabilities

Count	Technical Severity
26	Informational findings