**Instructure: Canvas, Bridge, Arc, Practice, MasteryConnect, Portfolium, Mobile Apps, and Other**
Grow from the First Day of School to the Last Day of Work

# Instructure Penetration Test Results: 2019

Bugcrowd Ongoing program results

Report created on May 30, 2019

Report date range: January 01, 2018 - December 31, 2018

bugcrowd

**Prepared by**

mhillary@instructure.com

# Table of contents

## Executive summary

bugcrowd

**Instructure: Canvas, Bridge, Arc, Practice, MasteryConnect, Portfolium, Mobile Apps, and Other** 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 engagement 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 **Instructure: Canvas, Bridge, Arc, Practice, MasteryConnect, Portfolium, Mobile Apps, and Other's** targets during the period of: **01/01/2018 – 12/31/2018**.

For this Ongoing Program, submissions were received from **34** unique researchers.

The continuation of this document summarizes the findings, analysis, and recommendations from the Ongoing Bounty Program performed by Bugcrowd for **Instructure: Canvas, Bridge, Arc, Practice, MasteryConnect, Portfolium, Mobile Apps, and Other**.

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.

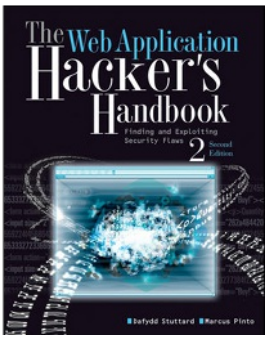
### Background

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:


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





## Scope

All details of the program scope and full program brief can be reviewed in the Program Brief.

Prior to the Ongoing program launching, Bugcrowd worked with Instructure: Canvas, Bridge, Arc, Practice, MasteryConnect, Portfolium, Mobile Apps, and Other 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:

```
https://bugcrowd-tc.instructure.com/
```

```
https://bugcrowd*.staging.bridgeapp.com
```

```
https://sectest.beta.instructuremedia.com
```

```
https://bugcrowd*.perform.stage.bridgeapp.com
```

```
https://play.google.com/store/apps/details?
id=com.instructure.candroid
```

```
https://play.google.com/store/apps/details?
id=com.instructure.teacher
```

```
https://play.google.com/store/apps/details?
id=com.instructure.parentapp
```

```
https://play.google.com/store/apps/details?
id=com.instructure.androidpolling.app
```

```
https://itunes.apple.com/us/app/canvas-student/id480883488?
mt=8
```

```
https://itunes.apple.com/us/app/canvas-parent/id1097996698?
mt=8
```

```
https://itunes.apple.com/us/app/canvas-teacher/id1257834464?
mt=8
```

```
https://itunes.apple.com/us/app/polls-for-canvas-create-take-
polls-in-canvas-by-instructure/id884329644?mt=8
```

```
*.stage.practice.xyz
```

```
app.stage.practice.xyz
```

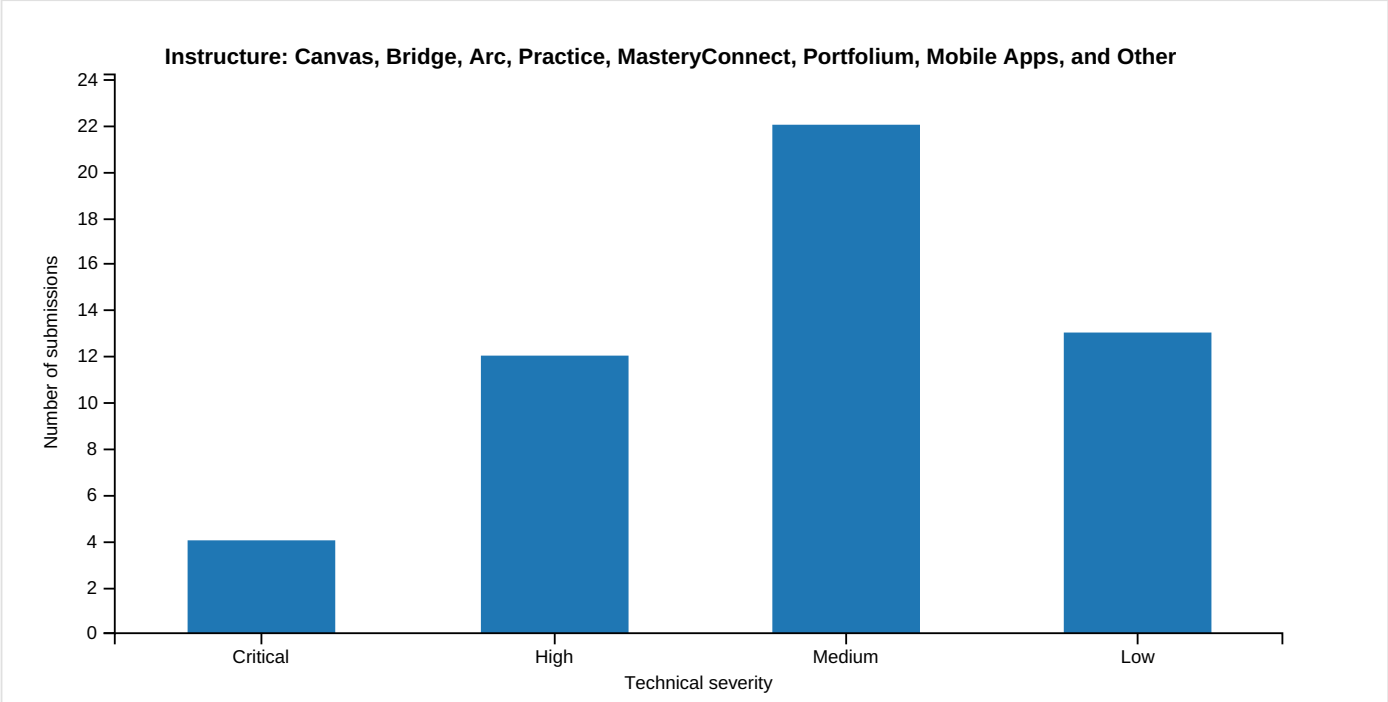
```
*.suite.staging.bridgeapp.com
```

```
https://catalog-bugcrowd.inscloudgate.net
```

```
https://bugcrowd.suite.staging.bridgeapp.com/connect
```

### 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 | EXAMPLE VULNERABILITY TYPES |
|---|---|
| **Critical**<br><br>Critical severity submissions (also known as "P1" or "Priority 1") are submissions that are escalated to **Instructure: Canvas, Bridge, Arc, Practice, MasteryConnect, Portfolium, Mobile Apps, and Other** 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<br>• Vertical Authentication Bypass<br>• XML External Entities Injection<br>• SQL Injection<br>• Insecure Direct Object Reference for a critical function |
| **High**<br><br>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<br>• Stored Cross-Site Scripting<br>• Cross-Site Request Forgery for a critical function<br>• Insecure Direct Object Reference for a important function<br>• Internal Server-Side Request Forgery |
| **Medium**<br><br>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<br>• Cross-Site Request Forgery for a important function<br>• Insecure Direct Object Reference for an unimportant function |
| **Low**<br><br>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<br>• Cross-Site Request Forgery for an unimportant function<br>• External Server-Side Request Forgery |
| **Informational**<br><br>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<br>• Autocomplete enabled<br>• Non-exploitable SSL issues |


**Bugcrowd's Vulnerability Rating Taxonomy**

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

### Findings table

The following table lists all valid assessment findings from the program:

| TITLE | VRT | DUPLICATES | PRIORITY | STATE | LINK |
|---|---|---|---|---|---|
| LFI and SSRF on "canvasdocs.instructure.com" file preview functionality via submitting HTML document with PDF extension - dumped AWS keys that have privileges | Server-Side Injection | - | P1 | RESOLVED | |
| XXE! | Server-Side Injection | - | P1 | RESOLVED | |
| Leaked Slack-webhook used for Public Canvas LMS exposes internal users/channels for instructure.slack.com and allows posting to any user/channel | Sensitive Data Exposure | - | P1 | RESOLVED | |
| SSRF to Aws credentials via API(upload via url)! | Sensitive Data Exposure | 1 | P1 | RESOLVED | |
| Listing/Fetching/Modifying/Deleting any uploaded file in the instructure-signal-beta bucket due to flawed multipart_signature-logic | Server Security Misconfiguration | - | P2 | RESOLVED | |
| XSS from Author to Admin via URI XSS via `original` parameter on https://bugcrowd201710257.staging.bridgeapp.com | Cross-Site Scripting (XSS) | - | P2 | RESOLVED | |
| Listing any uploaded file in the instructure-signal-beta bucket due to flawed parsing in s3_url_signature.json | Server Security Misconfiguration | - | P2 | RESOLVED | |
| Stored URI XSS via `author/training` from author to admin (everyone) via `meeting_url` parameter | Cross-Site Scripting (XSS) | - | P2 | RESOLVED | |
| Stored XSS for any user on sectest.beta.instructuremedia.com due to Movie Name on public page | Cross-Site Scripting (XSS) | - | P2 | RESOLVED | |
| Session Hijacking in Canvas | Cross-Site Scripting (XSS) | - | P2 | RESOLVED | |

| TITLE | VRT | DUPLICATES | PRIORITY | STATE | LINK |
|---|---|---|---|---|---|
| XSS from Author to Admin via URI XSS in `img href` on https://bugcrowd201710257.staging.bridgeapp.com | Cross-Site Scripting (XSS) | - | P2 | RESOLVED | |
| XSS from `author` to `admin` by force-using disabled applications via the `url` parameter on `bugcrowd201710257.staging.bridgeapp.com` | Cross-Site Scripting (XSS) | - | P2 | RESOLVED | |
| `holly.inscloudgate.net` allows registration to anyone for employee-only social media, stored XSS via `resource[url]` parameter at `/resources` | Cross-Site Scripting (XSS) | - | P2 | RESOLVED | |
| [Stored] XSS & Leak of Authenticity Token via JavaScript event handler for `data-method`. | Cross-Site Scripting (XSS) | - | P2 | RESOLVED | |
| Subdomain Takeover Via unclaimed Heroku Instance bbb03.instructure.com | Server Security Misconfiguration | - | P2 | RESOLVED | |
| Stored XSS via embedded SWF | Server-Side Injection | - | P2 | RESOLVED | |
| [Reflected] XSS via `return_to` parameter | Cross-Site Scripting (XSS) | - | P3 | RESOLVED | |
| The user with manager role can modify due date of learners who in another teams in the checkpoints | Broken Access Control (BAC) | - | P3 | RESOLVED | |
| The user with manager role can modify expire day of learners who in another teams in the Programs | Broken Access Control (BAC) | - | P3 | RESOLVED | |
| The user with learner can modify groups in Programs | Broken Access Control (BAC) | - | P3 | RESOLVED | |
| The user with learner role can delete group in Courses | Broken Access Control (BAC) | 1 | P3 | RESOLVED | |

| TITLE | VRT | DUPLICATES | PRIORITY | STATE | LINK |
|---|---|---|---|---|---|
| The user can add note in agenda on behalf of another user | Broken Access Control (BAC) | 1 | P3 | RESOLVED | |
| The user can edits tasks of another users | Broken Access Control (BAC) | 1 | P3 | RESOLVED | |
| Privilege Escalation - Organization Takeover | Broken Authentication and Session Management | - | P3 | RESOLVED | |
| The user with manager role can delete team goal of another teams | Broken Access Control (BAC) | - | P3 | RESOLVED | |
| The user with manager role can add learners who in another teams in the Session of Live training | Broken Access Control (BAC) | - | P3 | RESOLVED | |
| The user with manager role can enable Requires Approval in Checkpoints of another team and add him as Approver | Broken Access Control (BAC) | - | P3 | RESOLVED | |
| The user with manager role can make complete progress of learners who in another teams in the program | Broken Access Control (BAC) | - | P3 | RESOLVED | |
| Plaintext login credentials for Zendesk in Canvas app source code. | Sensitive Data Exposure | - | P3 | RESOLVED | |
| postMessage XSS in Arc due to cross-origin listener without origin-check | Cross-Site Scripting (XSS) | - | P3 | RESOLVED | |
| The learner user can post any notes to Private Notes of account_admin | Broken Access Control (BAC) | 1 | P3 | RESOLVED | |
| The use can delete goals of another users | Broken Access Control (BAC) | 1 | P3 | RESOLVED | |

| TITLE | VRT | DUPLICATES | PRIORITY | STATE | LINK |
|---|---|---|---|---|---|
| The user can create a new assessment to another users | Broken Access Control (BAC) | 1 | P3 | RESOLVED | |
| The user can add subtask to goals of another users | Broken Access Control (BAC) | - | P3 | RESOLVED | |
| Assign 1on1 task as a other user | Broken Authentication and Session Management | 4 | P3 | RESOLVED | |
| Persistent XSS : LDAP Authentication | Cross-Site Scripting (XSS) | - | P3 | RESOLVED | |
| The user with manager role can add learners who in another teams in the program | Broken Access Control (BAC) | - | P3 | RESOLVED | |
| The user can add tasks to others in another teams | Broken Access Control (BAC) | - | P3 | RESOLVED | |
| Email bombing to any user | Server Security Misconfiguration | - | P4 | UNRESOLVED | |
| Rte limiting on forgot password | Server Security Misconfiguration | - | P4 | UNRESOLVED | |
| No Rate Limiting on Form https://app.stage.practice.xyz/login | Server Security Misconfiguration | 1 | P4 | UNRESOLVED | |
| Blind SSRF at `bugcrowd201710257.staging.bridgeapp.com/api/config/sub_account/apps` via `subdomain` parameter allows for internal port scanning | Broken Access Control (BAC) | - | P4 | RESOLVED | |
| Vulnerability: Backup mode enabled | Sensitive Data Exposure | - | P4 | RESOLVED | |

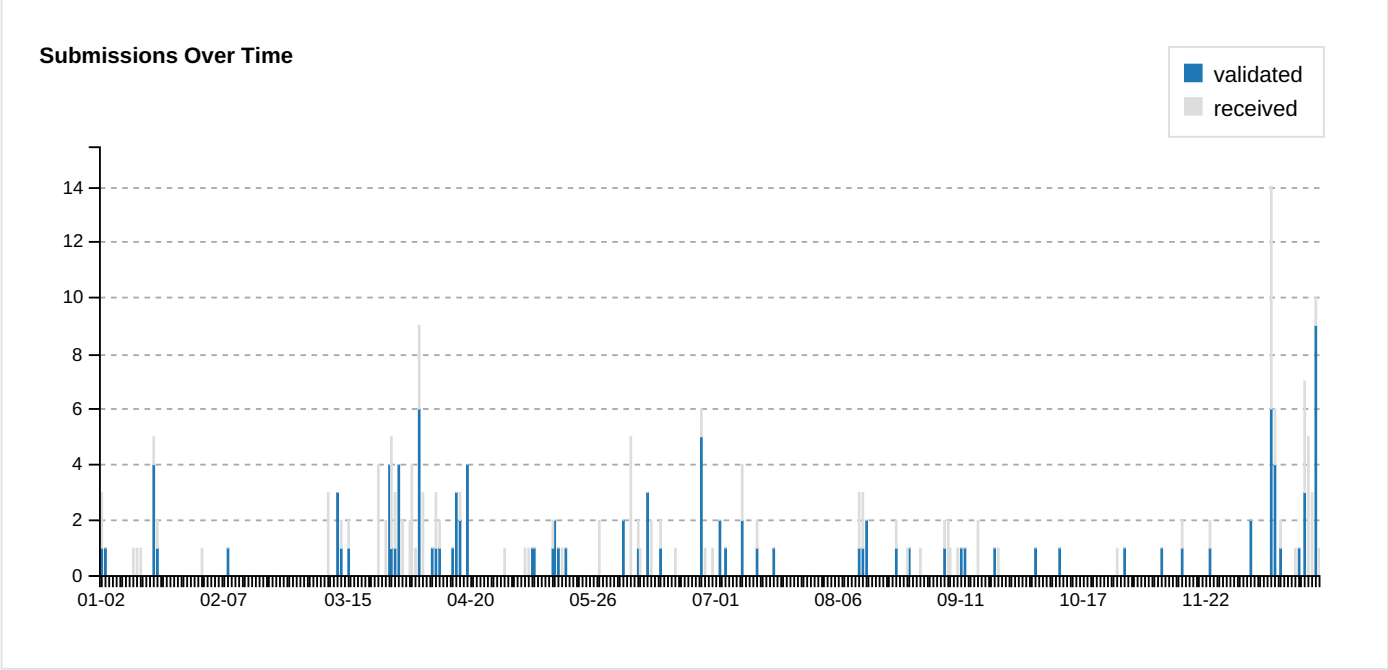
| TITLE | VRT | DUPLICATES | PRIORITY | STATE | LINK |
|---|---|---|---|---|---|
| Blind SSRF and Malicious Link Injection affecting `account_admin` from `author` (or anyone who can import CSVs) via `/api/admin/users/import` on `bugcrowd201710257.staging.bridgeapp.com` – | Server Security Misconfiguration | - | P4 | RESOLVED | |
| External Auth Injection in Older Browsers/Image from Student to Teacher via `submission[media_comment_id]` Parameter | Server-Side Injection | - | P4 | UNRESOLVED | |
| Vulnerability: Backup mode enabled | Sensitive Data Exposure | - | P4 | RESOLVED | |
| Complete Trial Takeover on *.acme.instructure.com via IDOR in /demos/{id}/loggin_as/admin Leads to Sub-domain Takeover of acme.instructure.com | Broken Access Control (BAC) | - | P4 | RESOLVED | |
| Vulnerability: Backup mode enabled | Sensitive Data Exposure | - | P4 | RESOLVED | |
| Lack of rate limiting - resend invitation feature | Server Security Misconfiguration | - | P4 | UNRESOLVED | |
| Vulnerability: Backup mode enabled | Sensitive Data Exposure | - | P4 | RESOLVED | |
| JIRA account misconfig causes internal info leak | Server Security Misconfiguration | - | P4 | RESOLVED | |
| Attackers can abuse zendesk's CC Feature to signup for other accounts with the support+id email | Broken Authentication and Session Management | - | P5 | UNRESOLVED | |

# 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

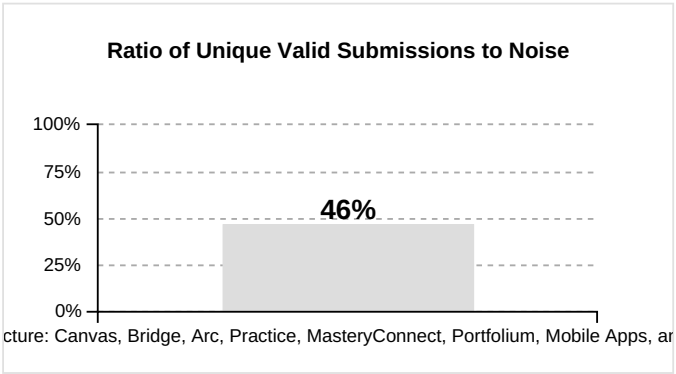
The timeline below shows submissions received and validated by the Bugcrowd team:




### Submissions signal

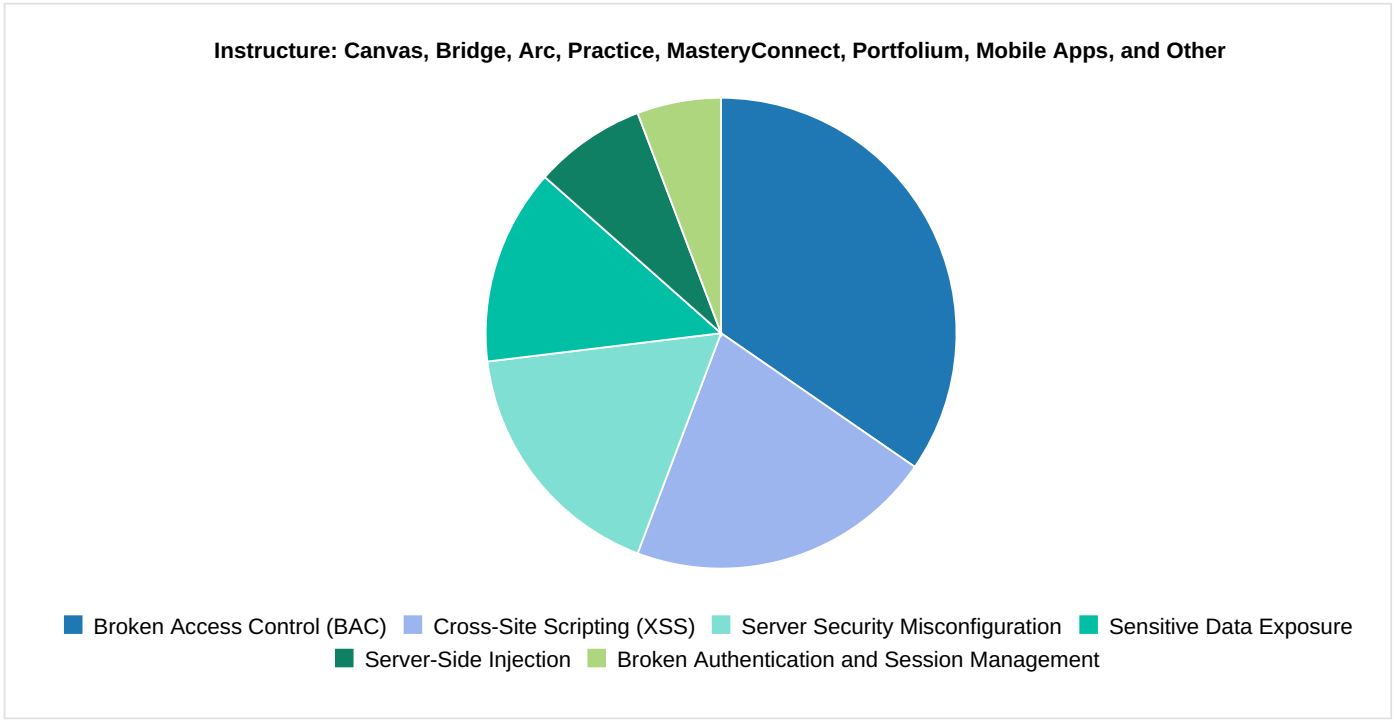
A total of **112** submissions were received, with **52** unique valid issues discovered. Bugcrowd identified **36** duplicate submissions, removed **24** invalid submissions, and is processing **0** submissions. The ratio of unique valid submissions to noise was **46%**.

| SUBMISSION OUTCOME | COUNT |
|---|---|
| Valid | 52 |
| Invalid | 24 |
| Duplicate | 36 |
| Processing | 0 |
| **Total** | **112** |



## Bug types overview

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



## Closing statement

bugcrowd

May 30, 2019

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

### Introduction

This report shows testing of **Instructure: Canvas, Bridge, Arc, Practice, MasteryConnect, Portfolium, Mobile Apps, and Other** between the dates of **01/01/2018 - 12/31/2018.** During this time, **34** researchers from Bugcrowd submitted a total of **112** vulnerability submissions against **Instructure: Canvas, Bridge, Arc, Practice, MasteryConnect, Portfolium, Mobile Apps, and Other's** targets. The purpose of this assessment was to identify security issues that could adversely affect the integrity of Instructure: Canvas, Bridge, Arc, Practice, MasteryConnect, Portfolium, Mobile Apps, and Other. Testing focused on the following:

1. **https://bugcrowd-tc.instructure.com/**
2. **https://bugcrowd*.staging.bridgeapp.com**
3. **https://sectest.beta.instructuremedia.com**
4. **https://bugcrowd*.perform.stage.bridgeapp.com**
5. **https://play.google.com/store/apps/details?id=com.instructure.candroid**
6. **https://play.google.com/store/apps/details?id=com.instructure.teacher**
7. **https://play.google.com/store/apps/details?id=com.instructure.parentapp**
8. **https://play.google.com/store/apps/details?id=com.instructure.androidpolling.app**
9. **https://itunes.apple.com/us/app/canvas-student/id480883488?mt=8**
10. **https://itunes.apple.com/us/app/canvas-parent/id1097996698?mt=8**
11. **https://itunes.apple.com/us/app/canvas-teacher/id1257834464?mt=8**
12. **https://itunes.apple.com/us/app/polls-for-canvas-create-take-polls-in-canvas-by-instructure/id884329644?mt=8**
13. ***.stage.practice.xyz**
14. **app.stage.practice.xyz**
15. ***.suite.staging.bridgeapp.com**
16. **https://catalog-bugcrowd.inscloudgate.net**
17. **https://bugcrowd.suite.staging.bridgeapp.com/connect**

The assessment was performed under the guidelines provided in the statement of work between **Instructure: Canvas, Bridge, Arc, Practice, MasteryConnect, Portfolium, Mobile Apps, and Other** 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 engagement, Bugcrowd discovered the following:

| COUNT | TECHNICAL SEVERITY |
|---|---|
| **4** | **Critical** vulnerabilities |
| **12** | **High** vulnerabilities |
| **22** | **Medium** vulnerabilities |
| **13** | **Low** vulnerabilities |
| **1** | **Informational** finding |