



# Penetration Test Results

Instructure Engineering, Security, and Operations

May 2020

# bugcrowd



CANVAS



PORTFOLIUM



BRIDGE

## Table of Contents

|                                 |    |
|---------------------------------|----|
| Executive Summary .....         | 3  |
| Reporting and Methodology ..... | 4  |
| Targets and Scope.....          | 5  |
| Findings Summary.....           | 6  |
| Findings by Severity .....      | 6  |
| Risk and Priority Key.....      | 7  |
| Findings table .....            | 8  |
| Appendix.....                   | 18 |
| Submissions Over Time .....     | 18 |
| Submissions Signal.....         | 18 |
| Bug Types Overview .....        | 19 |
| Closing Statement.....          | 20 |
| Introduction .....              | 20 |
| Ongoing Program Overview .....  | 21 |
| Testing Methods.....            | 21 |
| Summary of Findings.....        | 21 |

## Executive Summary

This is Instructure's 9<sup>th</sup> annual open security audit and once again Instructure engaged Bugcrowd, Inc. to perform an Ongoing Bounty Program, commonly known as a crowd-sourced penetration test for its Canvas LMS, Canvas Mastery, Bridge, Studio, Practice, and Portfolium products.

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 Canvas LMS, Canvas Mastery, Bridge, Studio, Practice, and Portfolium's targets during the period of: 01/01/2019 – 12/31/2019.

For this Ongoing Program, submissions were received from 60 unique researchers.

The continuation of this document summarizes the finding, analysis, and recommendations from the Ongoing Bounty Program performed by Bugcrowd for Canvas LMS, Canvas Mastery, Bridge, Studio, Practice, and Portfolium.

The full program brief can be found on [Bugcrowd's website](#). If you are interested in joining our bug bounty program as a security researcher, please contact [security@instructure.com](mailto:security@instructure.com) with your Bugcrowd username and we will get you hooked up!

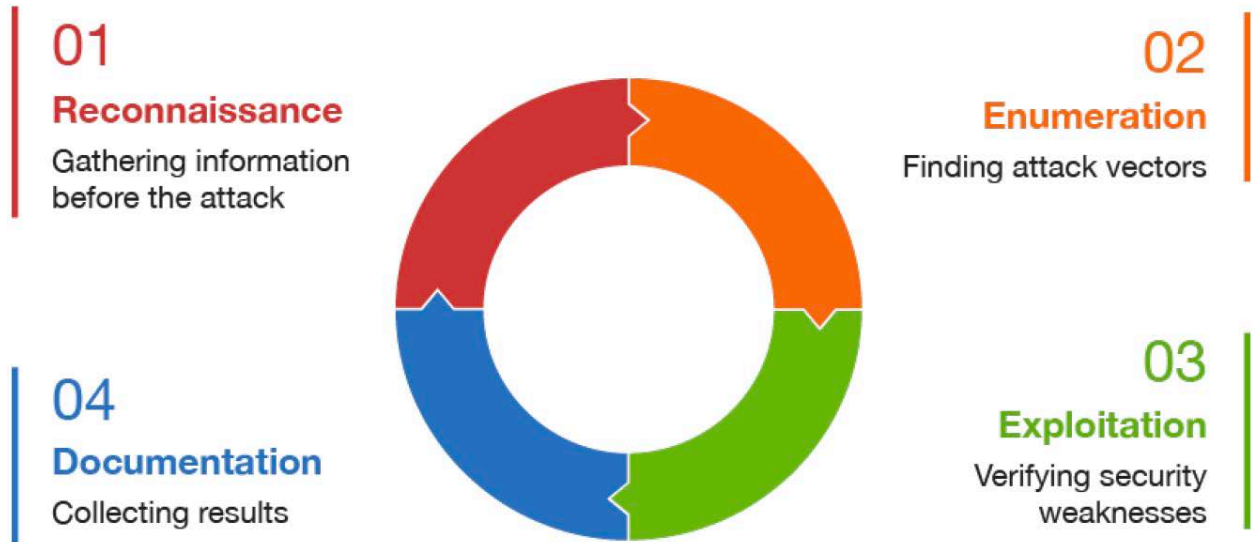
Keep learning,

*Matthew Spencer*

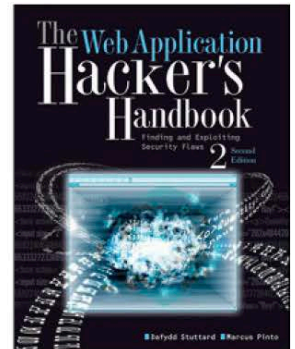
Matthew Spencer  
Senior Security Engineer  
[security@instructure.com](mailto:security@instructure.com)

## Reporting and Methodology

The strength of crowdsourced testing lies in multiple researches, 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.



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



## Targets and Scope

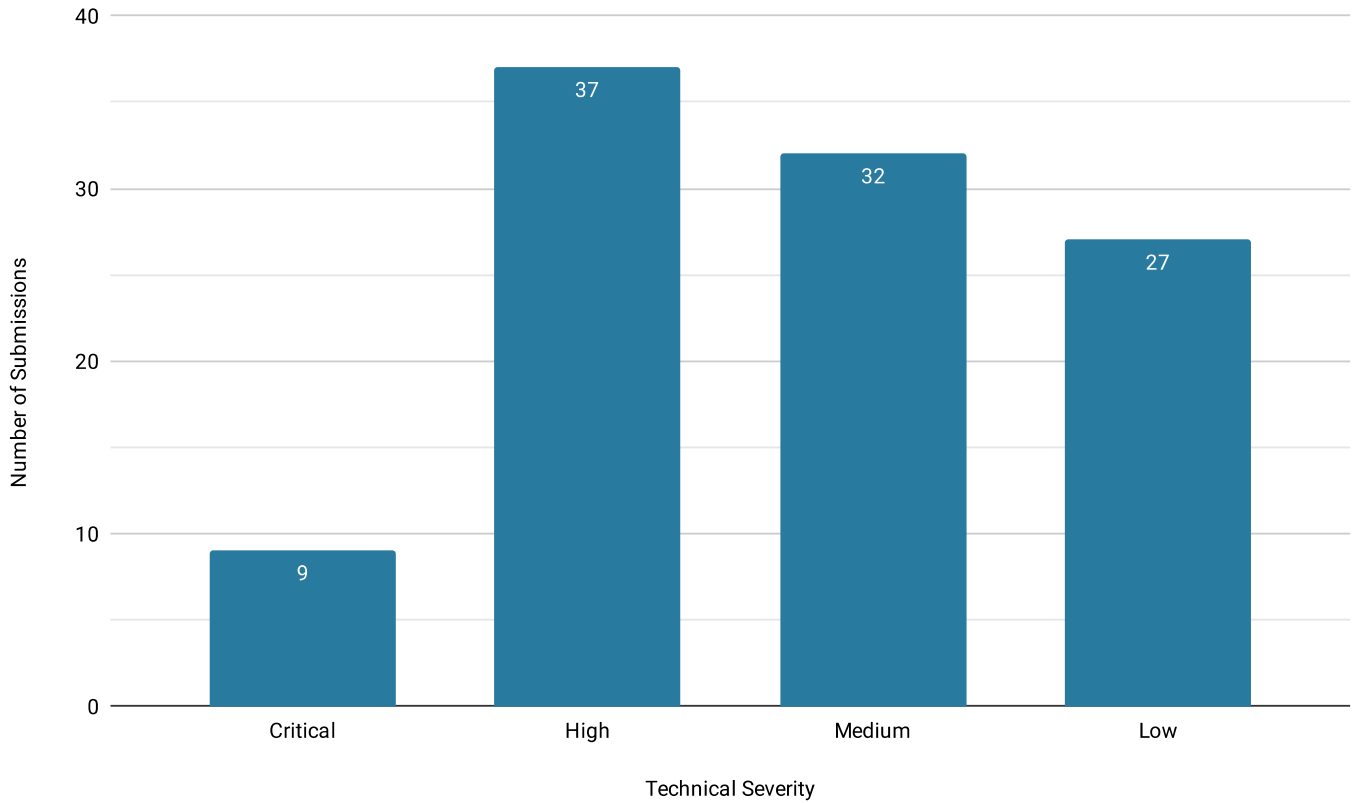
Prior to the Ongoing program launching, Bugcrowd worked with Instructure to define the Rules of Engagement, commonly known as the program brief, which includes the scope of work. The following targets were considered explicitly in scope for testing:

```
https://bugcrowd-tc.instructure.com
https://bugcrowd*.staging.bridgeapp.com
https://secttest.beta.instructuremedia.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/poll-for-canvas-create-take-polls-in-canvas-by-
instructure/id884329644?mt=8
https://*.stage.practice.xyz
https://app.stage.practice.xyz
https://*.suite.staging.bridgeapp.com
https://catalog-bugcrowd.insclooudgate.net
https://bugcrowd.suite.staging.bridgeapp.com/connect
https://*qa.portfolium.com
https://commons-pdx-edge.inseng.net
```

## 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 Instructure 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
  
- Lateral Authentication Bypass
- Stored Cross-Site Scripting
- Cross-Site Request Forgery for a critical 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 a 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



### 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    |
|--|----------------------------------|------------|----------|----------|
| Invalid Access Controls/Privilege Escalation – Teacher can gain School Administrator, and change schools without approval of the target district. [MasteryConnect] | Broken Access Control (BAC)      | -          | P1       | Resolved |
| SSRF to RCE [Portfolio]  | Broken Access Control (BAC)      | 1          | P1       | Resolved |
| Public Google Calendar exposing Passwords, Conference call data, and critical PII info   | Sensitive Data Exposure          | 3          | P1       | Resolved |
| Lead of Authorization Token via Bypass of Validation Functionality for External Tools [Canvas iOS]   | Sensitive Data Exposure          | -          | P1       | Resolved |
| Misconfigured COORS and CSRF to Steal user's all files [Portfolio]   | Server Security Misconfiguration | 2          | P1       | Resolved |
| ePortfolio export will bypass all access controls for files [Canvas]   | Broken Access Control            | -          | P1       | Resolved |
| Unauthenticated RCE on /mstrit [MasteryConnect]  | Server-Side Injection            | -          | P1       | Resolved |
| Upload content to any Portfolio s3 bucket/ replace/delete website/user content [Portfolio]   | Broken Authentication            | 1          | P1       | Resolved |



| Title  | VRT                         | Duplicates | Priority  | State           |
|--|-----------------------------|------------|-----------|-----------------|
|  | and Session Management      |            |           |                 |
| Leak of app Authorization token via Image with data-api-endpoint [Canvas iOS]                    | Sensitive Data Exposure     | -          | <b>P1</b> | <b>Resolved</b> |
| Stored XSS [MasteryConnect]  | Cross-Site Scripting (XSS)  | -          | <b>P2</b> | <b>Resolved</b> |
| [Stored] XSS via exploit elementToggler.js [Canvas]  | Cross-Site Scripting (XSS)  | 0          | <b>P2</b> | <b>Resolved</b> |
| [Stored] XSS in Group Wiki Pages via Prerequisites lookup exploit [Canvas]                       | Cross-Site Scripting (XSS)  | -          | <b>P2</b> | <b>Resolved</b> |
| [Stored] XSS via `data-item-href` in Wiki Pages [Canvas]   | Cross-Site Scripting (XSS)  | -          | <b>P2</b> | <b>Resolved</b> |
| [Stored] XSS via Exploiting jQuery Selector issue via `data-focus-returns-to` attribute [Canvas] | Cross-Site Scripting (XSS)  | -          | <b>P2</b> | <b>Resolved</b> |
| CSRF + BAC – Able to add email to any user account without authorization [Canvas]                | Broken Access Control (BAC) | -          | <b>P2</b> | <b>Resolved</b> |
| [Stored] XSS in KyleMenu (global widget) via kyleMenuOptions [Canvas]                            | Cross-Site Scripting (XSS)  | -          | <b>P2</b> | <b>Resolved</b> |
| [Stored] XSS in Arc via exploit of TrackFormatParser.dfxp.parse method [Studio]                  | Cross-Site Scripting (XSS)  | -          | <b>P2</b> | <b>Resolved</b> |
| [Stored] XSS in ePortfolios via `data-popup-within` & `al-trigger` class [Canvas]                | Cross-Site Scripting (XSS)  | -          | <b>P2</b> | <b>Resolved</b> |
| [Stored] XSS via Bypass of sanitizeUrl functionality [Canvas]                                    | Cross-Site Scripting (XSS)  | -          | <b>P2</b> | <b>Resolved</b> |

| Title   | VRT                               | Duplicates | Priority | State    |
|---|-----------------------------------|------------|----------|----------|
| [Stored] XSS via a.file_preview_link [Canvas]   | Cross-Site Scripting (XSS)        | -          | P2       | Resolved |
| Subdomain Takeover of *masteryconnect-staging.com subdomains [MasteryConnect]                       | Server Security Misconfiguration  | -          | P2       | Resolved |
| Access to submission/comments media! [Canvas]   | Broken Access Control (BAC)       | -          | P2       | Resolved |
| XSS from student to anyone in ePortfolios [Canvas]  | Cross-Site Scripting (XSS)        | -          | P2       | Resolved |
| [Stored] XSS via Flash Message exploit [Canvas]   | Cross-Site Scripting (XSS)        | -          | P2       | Resolved |
| [Stored] XSS by editor_button.icon_url [Canvas]   | Cross-Site Scripting (XSS)        | -          | P2       | Resolved |
| Student can view course's unpublished/not allowed for students pages via atom feed [Canvas]         | Broken Access Control             | -          | P2       | Resolved |
| CSRF on various endpoints (Following/Subscribing, Blocking, and Pin Liking) [MasteryConnect]        | Cross-Site Request Forgery (CSRF) | -          | P2       | Resolved |
| [Stored] XSS via Malicious Lang Name inn CC [Studio]  | Cross-Site Scripting (XSS)        | -          | P2       | Resolved |
| CRLF injection leading to installation of a service worker on cdn.inst-fs-iad-prod.inscloudgate.net | Other                             | -          | P2       | Resolved |
| Student can access all unpublished/restricted course files! [Canvas]                                | Broken Access Control (BAC)       | -          | P2       | Resolved |
| Stored XSS via math equation editor [Canvas]  | Cross-Site Scripting (XSS)        | -          | P2       | Resolved |

| Title  | VRT                               | Duplicates | Priority | State    |
|--|-----------------------------------|------------|----------|----------|
| Wormable Stored XSS! [Portfolium]  | Cross-Site Scripting (XSS)        | -          | P2       | Resolved |
| [Stored] XSS via `data-turn-into-dialog` behavior [Canvas]   | Cross-Site Scripting (XSS)        | -          | P2       | Resolved |
| HTML5 AppCache can be used to intercept and modify file downloads on cdn.inst-fs-iad-prod.inscloudgate.net | Server Security Misconfiguration  | -          | P2       | Resolved |
| CSRF [MasteryConnect]  | Cross-Site Request Forgery (CSRF) | -          | P2       | Resolved |
| [Stored] XSS in Learner Submission via Video Response download link [Practice]                             | Cross-Site Scripting (XSS)        | -          | P2       | Resolved |
| [Stored] XSS in Calendar via `data-mathml` attribute [Canvas]  | Cross-Site Scripting (XSS)        | -          | P2       | Resolved |
| Chat module see other people's chat, delete messages and impersonate users. [Canvas]                       | Broken Access Control             | -          | P2       | Resolved |
| [Stored] XSS via Replacing Server Response for GET `media_objects/:media_object_id/info` [Canvas]          | Cross-Site Scripting (XSS)        | -          | P2       | Resolved |
| Subdomain takeover of *masteryconnect.com subdomains [MasteryConnect]                                      | Server Security Misconfiguration  | -          | P2       | Resolved |
| [Stored] XSS in ePortfolios [Canvas]   | Cross-Site Scripting (XSS)        | -          | P2       | Resolved |
| Subdomain takeover via unclaimed Heroku Instance bridge-registration.bridgeapp.com [Bridge]                | Server Security Misconfiguration  | -          | P2       | Resolved |
| Multiple XSS on /app/:id [eduappcenter.com]  | Cross-Site Scripting (XSS)        | -          | P2       | Resolved |

| Title  | VRT  | Duplicates | Priority | State    |
|--|--|------------|----------|----------|
| Insufficient Access Controls [eduappcenter.com]  | Broken Access Control (BAC)                  | -          | P2       | Resolved |
| [Stored] XSS via `data-tooltip` HTML exploit   | Cross-Site Scripting (XSS)                   | -          | P2       | Resolved |
| [Stored] XSS in Media Comments via Subtitles (similar to Arc issue) [Studio]                               | Cross-Site Scripting (XSS)                   | -          | P2       | Resolved |
| Insecure access control handling on what-if requests for muted Quizzes allows for hidden scores to be read | Broken Access Control (BAC)                  | -          | P3       | Resolved |
| Subdomain takeover via unclaimed acquia.com domain for ondeckdev.blog.instructure.com                      | Server Security Misconfiguration             | -          | P3       | Resolved |
| Stealing private information with XSS – Reflected XSS – Bypass WAF   | Cross-Site Scripting (XSS)                   | 2          | P3       | Resolved |
| Social media account compromise via instructure.com/about/blog   | Server Security Misconfiguration             | -          | P3       | Resolved |
| Access to Kibana instance  | Broken Authentication and Session Management | -          | P3       | Resolved |
| DOM XSS in both “t” and “s” parameter at recorded-webinar.html page [masteryconnect-staging.com]           | Cross-Site Scripting (XSS)                   | -          | P3       | Resolved |
| Access to internal excel document, leaking finance data and vendor sheet and cost agreements, etc.         | Sensitive Data Exposure                      | -          | P3       | Resolved |
| [Stored] XSS in vdd_tooltip_link (exploit jQuery selector)   | Cross-Site Scripting (XSS)                   | -          | P3       | Resolved |
| [Stored] CSS in Canvas Quizzes via malicious response from server  | Cross-Site Scripting (XSS)                   | -          | P3       | Resolved |

| Title  | VRT                                       | Duplicates | Priority | State    |
|--|---|------------|----------|----------|
| [Stored] XSS in Courses (by combining two issues)  | Cross-Site Scripting (XSS)                | -          | P3       | Resolved |
| Student can access files with restricted access with link  | Broken Access Control (BAC)               | -          | P3       | Resolved |
| Blind SSRF due to misconfiguration   | Broken Access Control (BAC)               | -          | P3       | Resolved |
| Stored XSS via SVG file on Portfolium  | Cross-Site Scripting (XSS)                | 4          | P3       | Resolved |
| [masteryconnect-staging.com] CVE-2018-6389 – Application level DoS in loading scripts function       | Application-Level Denial-of-Service (DoS) | -          | P3       | Resolved |
| Unauthenticated user can see events and locations of public courses                                  | Broken Access Control (BAC)               | -          | P3       | Resolved |
| Stored XSS via iframe  | Cross-Site Scripting (XSS)                | 1          | P3       | Resolved |
| Stored XSS via rubric in SpeedGrader option  | Cross-Site Scripting (XSS)                | 1          | P3       | Resolved |
| Users can see automatically graded scores on muted or hidden assignments                             | Server Security Misconfiguration          | 1          | P3       | Resolved |
| Application level DoS due to video title lack of content length limit                                | Application-Level Denial-of-Service       | -          | P3       | Resolved |
| IDOR – Access other account user’s PII information (email, role, full name, job, hire_date) [Bridge] | Broken Access Control                     |            | P3       | Resolved |
| Stored XSS on Upload Gradebook   | Cross-Site Scripting (XSS)                | 1          | P3       | Resolved |

| Title   | VRT                                 | Duplicates | Priority | State    |
|---|-------------------------------------|------------|----------|----------|
| Application level DoS on project comment_list due to lists are not sanitized  | Application-Level Denial-of-Service | -          | P3       | Resolved |
| No rate limit leads to send thousands of email to hundreds of users at once   | Server Security Misconfiguration    | -          | P3       | Resolved |
| Enumeration of names of users   | Server Security Misconfiguration    | -          | P3       | Resolved |
| SSRF in add app by URL function   | Broken Access Control (BAC)         | -          | P3       | Resolved |
| Stored XSS in Markdown link feature at project's description can be use against anyone (even non-Portfolium users) – bypass WAF | Cross-Site Scripting (XSS)          | -          | P3       | Resolved |
| BAC – Able to attach my project to any other user's courses (and other items)   | Broken Access Control (BAC)         | -          | P3       | Resolved |
| IDOR – Create/Delete question bank and questions in other accounts  | Broken Access Control (BAC)         | -          | P3       | Resolved |
| IDOR – A user is possible to read all to-do info of other account users with same ID – information leakage                      | Broken Access Control (BAC)         | 1          | P3       | Resolved |
| Exif data not stripped [MasteryConnect]   | Sensitive Data Exposure             | -          | P3       | Resolved |
| Reflected CSS via successFlash parameter  | Cross-Site Scripting (XSS)          | -          | P3       | Resolved |
| Stored XSS  | Cross-Site Scripting (XSS)          | -          | P3       | Resolved |

| Title  | VRT  | Duplicates | Priority | State      |
|--|--|------------|----------|------------|
| Open Redirect  | Unvalidated Redirects and Forwards           | -          | P4       | Resolved   |
| IDOR Delete planner_notes of another user  | Broken Access Control (BAC)                  | -          | P4       | Resolved   |
| Unvalidated redirect at login page [return_url param]  | Unvalidated Redirects and Forwards           | -          | P4       | Resolved   |
| Click-jacking to deactivate admin account via user's account                                       | Server Security Misconfiguration             | -          | P4       | Resolved   |
| Course members without permission to view grades can see grades via the Student Interaction Report | Broken Authentication and Session Management | -          | P4       | Resolved   |
| Course members without permission to view or manage grades can download all quiz submissions       | Broken Authentication and Session Management | -          | P4       | Resolved   |
| Unauthenticated users can send arbitrary HTTP requests to remote hosts                             | Broken Access Control (BAC)                  | 1          | P4       | Unresolved |
| Open redirect – “location” param at [/sso/authmonger/login]  | Unvalidated Redirects and Forwards           | -          | P4       | Resolved   |
| GET-based open redirection   | Unvalidated Redirects and Forwards           | 1          | P4       | Unresolved |
| SSRF on “change picture”   | Broken Access Control (BAC)                  | -          | P4       | Resolved   |

| Title  | VRT  | Duplicates | Priority | State      |
|--|--|------------|----------|------------|
| Unvalidated GET-based redirect on oath.staging.bridgeapp.com   | Unvalidated Redirects and Forwards           | -          | P4       | Resolved   |
| No rate limiting on email triggering function of “forgot password” feature                                       | Server Security Misconfiguration             | -          | P4       | Resolved   |
| Lack of rate limiting in promotion code for paid courses [Catalog]   | Server Security Misconfiguration             | -          | P4       | Resolved   |
| Internal data exposure   | Server Security Misconfiguration             | -          | P4       | Resolved   |
| Failure to invalidate session on password change   | Broken Authentication and Session Management | -          | P4       | Unresolved |
| Enumeration of course names  | Other  | -          | P4       | Resolved   |
| Course members without permission to see submissions can receive email notifications with submission information | Broken Access Control (BAC)                  | -          | P4       | Resolved   |
| No rate limit on login form  | Server Security Misconfiguration             | 3          | P4       | Resolved   |
| Open redirect in course assignment   | Unvalidated Redirects and Forwards           | -          | P4       | Resolved   |
| Broken access control on private comment portfolio   | Broken Access Control (BAC)                  | -          | P4       | Resolved   |
| IDOR edit group sets of other user’s courses   | Broken Access Control (BAC)                  | -          | P4       | Unresolved |



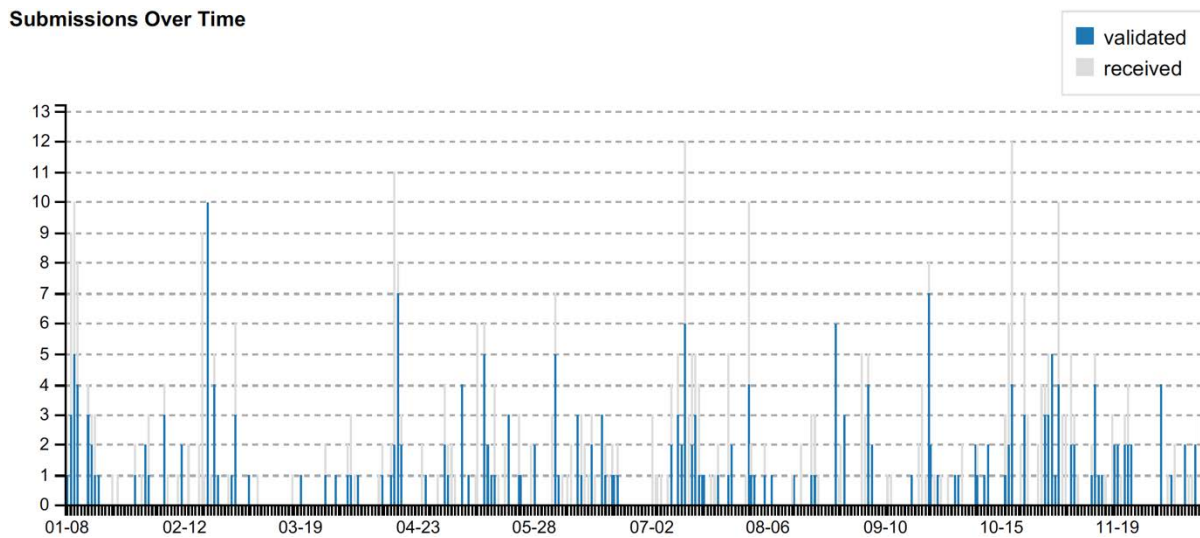
| Title  | VRT  | Duplicates | Priority | State    |
|--|--|------------|----------|----------|
| Course members without permission to view or manage grades can view complete submitted quizzes                 | Broken Authentication and Session Management | -          | P4       | Resolved |
| Unvalidated redirect in any URL path [blog.masteryconnect-staging.com]   | Unvalidated Redirects and Forwards           | -          | P4       | Resolved |
| Publicly exposed production statuses of all hosts – prod / staging / beta of instructure.com and bridgeapp.com | Server Security Misconfiguration             | -          | P4       | Resolved |
| Course members without permission to see submissions can receive email notifications with submission comments  | Broken Access Control (BAC)                  | -          | P4       | Resolved |
| Unvalidated redirect in OAuth endpoint [state parameter]   | Unvalidated Redirects and Forwards           | -          | P4       | Resolved |
| Other active sessions are not invalidated after a user completes the password reset process                    | Broken Authentication and Session Management | -          | P4       | Resolved |

## 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

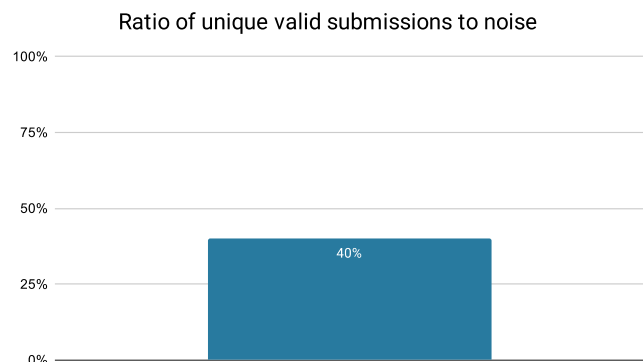
The timeline below shows submissions received and validated by the Bugcrowd team:



### Submissions Signal

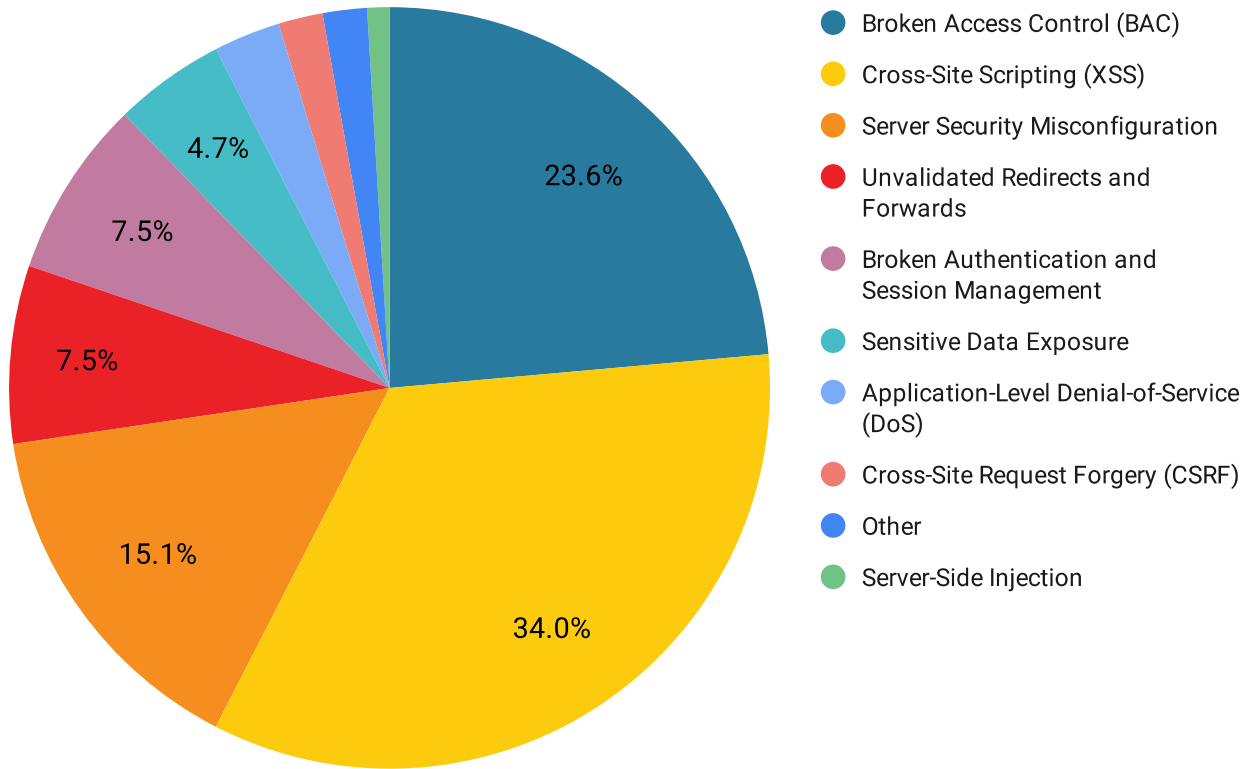
A total of **265** submissions were received, with **105** unique, valid issues discovered. Bugcrowd identified **54** duplicate submissions, removed **106** invalid submissions, and is processing **0** submissions. The ratio of unique, valid submissions to noise was **40%**.

| Submission Outcome | Count      |
|--------------------|------------|
| Valid              | 105        |
| Invalid            | 106        |
| Duplicate          | 54         |
| Processing         | 0          |
| <b>Total</b>       | <b>265</b> |



## Bug Types Overview

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



## Closing Statement

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

## Introduction

This report shows testing of Canvas LMS, Bridge, Studio, Practice, Portfolium, and Canvas Mastery between the dates of 01/01/2019 – 12/31/2019. During this time, 60 researchers from Bugcrowd submitted a total of 265 vulnerability submissions against Instructure’s targets. The purpose of this assessment was to identify security issues that could adversely affect the integrity of Instructure’s products. Testing focused on the following:

```
https://bugcrowd-tc.instructure.com
https://bugcrowd*.staging.bridgeapp.com
https://secttest.beta.instructuremedia.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/poll-for-canvas-create-take-polls-in-canvas-by-
instructure/id884329644?mt=8
https://*.stage.practice.xyz
https://app.stage.practice.xyz
https://*.suite.staging.bridgeapp.com
https://catalog-bugcrowd.inscloudgate.net
https://bugcrowd.suite.staging.bridgeapp.com/connect
https://*qa.portfolium.com
https://commons-pdx-edge.inseng.net
```

The assessment was performed under the guidelines provided in the statement of work between Instructure and Bugcrowd. This letter provides a high-level overview of the testing performed, and the results 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              |
|-------|---------------------------------|
| 9     | <b>Critical</b> vulnerabilities |
| 37    | <b>High</b> vulnerabilities     |
| 32    | <b>Medium</b> vulnerabilities   |
| 27    | <b>Low</b> vulnerabilities      |
| 0     | <b>Informational</b> findings   |