August 23, 2021

To whom it may concern,

As of March 30, 2021, HackerOne has completed our 2021 Penetration Test and issued a report. This report details the Methodology and Findings for the test itself, including links to the reports for vulnerabilities found.

All reports have now been remediated and closed. The links provided will take you to live Hacktivity pages outlining the remediation information. You may also find this information on our Hacktivity Page, https://hackerone.com/security/hacktivity.

Thank you for your interest in HackerOne and taking the time to review our Penetration Test.

Best Regards,

Samantha Cowan

Sam Cowan
Sr. Manager, Security Compliance

hackerone

HackerOne Pentest
Security Assessment

MARCH 30TH, 2021 • CONFIDENTIAL

Description

This document details the process and result of a penetration test performed by HackerOne between March 15th, 2021 and March 26th, 2021.

Author

Eduardo Cervantes (Security Solutions Architect, HackerOne)
eduardo@hackerone.com

Reviewers

Joaquin Silva Jr. (Security Solutions Architect, HackerOne)

Prepared for:

Table of Contents

1. Executive Summary

HackerOne performed an internal HackerOne pentest from March 15th, 2021 to March 26th, 2021. During this timeframe, 17 vulnerabilities were identified by 3 unique researchers.

During the assessment, 0 vulnerabilities were found that had a CVSS score of 7.0 or higher, rating either high or critical. Table 1 shows the in-scope assets and breakdown of findings by severity per asset. Section 2.5 contains more information on how severity is calculated.

	Critical	High	Medium	Low	None	Σ
api.hackerone.com/v1	0	0	0	0	0	0
hackerone.com/graphql	0	0	**2**	**4**	0	**6**
hackerone.com	0	0	0	**9**	**2**	**11**
66.232.20.0/23	0	0	0	0	0	0
206.166.248.0/23	0	0	0	0	0	0
	0	**0**	**2**	**13**	**2**	**17**

Table 1: Findings per asset

The security assessment was conducted using a crowdsourced penetration testing methodology. From its community of over 900,000 hackers, HackerOne curated a set of top-tier researchers to focus on identifying vulnerabilities in HackerOne's scope during the

agreed-upon testing window, while abiding by the policies set forth by HackerOne. Section 2 contains more information about the methodology.

The most common vulnerability type was Information Disclosure. The most severe vulnerability found was an Information Disclosure on HackerOne's GraphQL endpoint. This vulnerability, if exploited, could be used to access potentially sensitive objects that were soft-deleted by users.

State of Security

Below is the summary of methodologies used to assess security at the targets:

Inventory of API endpoints:

The team inventoried calls made by the application during all user activities. Requests and responses were then analyzed and observed to identify the underlying technology and any possible vulnerabilities.

Manual and automated fuzzing of web service endpoints:

Active testing then commenced by reverse-engineering the endpoints and performing modified calls using manual and automated methods.
During this phase, the team attempted the following:

- Parameter Manipulation (adding/modifying/removing parameters to perform new functions, horizontal privilege escalation, etc.);
- Code Injection (SQL Injection attempts, Template Injection, Cross-Site Scripting attacks, etc.);
- XML External Entity attacks;
- Header Manipulation;
- Path Traversal;
- Malicious file uploads;
- Etc.

As a result of the engagement, 2 Medium and 13 Low severity vulnerabilities were reported by the team.

Additionally, two very low-risk vulnerabilities were reported with a CVSS value of 0.0, which demonstrated no tangible risk to the organization or its users.

The applications seem very well hardened and developed and maintained with security in mind.

During this engagement, the most common vulnerabilities were not found, demonstrating that the applications are more secure than the average. While the team can’t say with certainty that some particular aspects of the application were more prone to issues than others, the consensus was that the GraphQL functionality should focus on the authorization process that could lead to vulnerabilities.

Maintaining a healthy security posture requires constant review and refinement of existing security processes. Running a HackerOne pentest allows HackerOne's internal security team to not only uncover specific vulnerabilities but gain a better understanding of the current security threat landscape.

Reviewing the remaining reports for a root cause analysis can further educate HackerOne's internal development and security teams and allow manual or automated procedures to be put in place to weed out entire classes of vulnerabilities in the future. This proactive approach helps contribute to future-proofing the security posture of HackerOne's assets.

Recommendations

Based on the results of this assessment, HackerOne has the following high-level key recommendations.

KEY RECOMMENDATION 1	
Key Issue	The most severe vulnerabilities found were: • #1139535 - Changing the 2FA secret key and backup codes without knowing the 2FA OTP; and • #1132606 - Attachment object in GraphQL continues to grant access to files, even if they are removed from rendering.
Recommendation	It is recommended to never allow changes on Two-Factor Authentication (2FA) configuration without confirming that the user knows the 2FA One Time Password (OTP). Attackers can exploit this vulnerability together with Clickjacking, Cross-Site Scripting, Cross-Site Request Forgery, or if they have physical access to the victim's device. • **Attachment object in GraphQL continues to grant access to files, even if they are soft-deleted** • **Changing the 2FA secret key and backup codes without knowing the 2FA OTP** Review the relationship to deleting files for the policy page, or change access to the Attachment object in GraphQL if rendering was removed.
Resources	• https://medium.com/@iSecMax/two-factor-authentication-security-testing-and-possible-bypasses-f65650412b35#a12c • https://hackerone.com/reports/1139535 • https://hackerone.com/reports/1132606

KEY RECOMMENDATION 2	
Key Issue	The application is particularly vulnerable to different types of `Security Misconfiguration`. Most of them are considered as missing security best practices and are not high-impact vulnerabilities.
Recommendation	Focus on implementing missing security best practices and other low impact vulnerabilities to prevent bug chaining. Bug chaining is the idea of combining different bugs (sometimes of low severity) to create a vulnerability of very high severity.
Resources	• https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration • https://owasp.org/www-pdf-archive//Application-Bug-Chaining-Live.pdf

2. Methodology

HackerOne engaged HackerOne to perform a HackerOne pentest. The following sections cover how the engagement was put together and executed.

2.1 Preparation phase

HackerOne worked internally to identify the types of vulnerabilities most important to the platform and understand the goal of this assessment. This collaborative process was used to:

- develop a scope for the engagement;
- determine what user permissions levels exist and which ones are in scope;
- select a sufficient testing window;
- identify the areas of HackerOne's scope that researchers should pay special attention to;
- and what types of vulnerabilities HackerOne is most interested in testing for.

All of this information was then placed into a "Security Page", also known as the rules of engagement. From its community of over 900,000 hackers, HackerOne curated a set of top-tier researchers to focus on identifying vulnerabilities in HackerOne's scope during the agreed-upon testing window, while following the guidelines and instructions from the Security Page. The hand-chosen researchers were tailored based on the size of the scope and the types of assets that were in scope to ensure broad coverage of skill and experience.

During the preparation phase a testing window from March 15, 2021 to March 26, 2021, was agreed-upon.

The contents of the Security Page were approved by HackerOne before moving to the testing phase.

2.1.1 Scope

During the preparation phase the following scope for the engagement was agreed-upon:

IN SCOPE ASSETS
api.hackerone.com/v1
hackerone.com/graphql
hackerone.com
66.232.20.0/23
206.166.248.0/23

Table 2: in scope assets

2.1.2 Test plan

The team used the following tools:

- Burp Suite Pro;
- Burp Suite Pro Extensions;
- Metasploit;
- Nmap; and
- Ffuf.

2.1.3 Engagement restrictions

The engagement execution happened under a truncated testing window of 11 days instead of the standard 14. The allocated hours of testing remained the same.

2.2 Testing phase

2.2.1 Information gathering & reconnaissance

The information gathering and reconnaissance step is the critical starting point for every researcher. This step is used to explore the boundaries of the targets in scope and develop a plan of attack. Each member of the security research team is encouraged to be creative in uncovering what may have been missed with conventional reconnaissance steps and tools, using unique methodologies and techniques. This includes but is not limited to:

- Conventional port and banner scanning using tools such as nmap and masscan
- DNS discovery and subdomain enumeration
- Reviewing certificate transparency records
- Exploration of Shodan and Censys public data
- Enumeration of possible hidden web directories
- Content spidering and crawling using tools such as Burp Suite

HackerOne further facilitates this testing by providing the testing team with useful documentation and guides to allow hackers to consume the service in the same manner used by a typical customer.

2.2.2 Penetration testing & exploitation

Upon starting the testing phase, all eligible researchers selected in the preparation phase were invited to participate in the engagement. A list of researchers that participated is available in Appendix A. The testing period ran from March 15, 2021 until March 26, 2021.

HackerOne's methodology encourages the use of individual tools and methods by each researcher. This ensures diversity in the testing and realistically simulates real-world attacks while also putting emphasis on vulnerabilities that are exploitable and have significant impact. It also ensures that new tools and techniques can be used in the testing. While individuality in testing methodology is encouraged, researchers ascribe to OWASP's (Open Web Application Security Project) standard testing techniques to uncover issues (e.g. OWASP Top 10) within HackerOne's scope. HackerOne also actively encourages creative

thinking by its researchers to combine potentially low-severity vulnerabilities into greater bugs that can have more impact, also known as "chaining".

Additionally, HackerOne's team of security analysts validated each vulnerability as they were reported throughout the testing phase. They also categorized all identified vulnerabilities against the CWE (Common Weakness Enumeration) standard, as well as assigned a severity rating based on the CVSS v3.0 (Common Vulnerability Scoring System) standard, providing consistent, easy to understand guidelines on the severity of each finding. Each finding was made available immediately to HackerOne through HackerOne's vulnerability management platform.

Throughout the testing phase, HackerOne continuously managed the engagement to maximize output and ensure the focus areas of the engagement are thoroughly covered.

2.4 Reporting phase

At the conclusion of the engagement, HackerOne's security team worked to analyze the results of the testing phase and identify any potential trends in vulnerabilities found across HackerOne's assets and key recommendations. The results of the engagement and post-engagement analysis were then summarized in this report. The final report was discussed with and approved by HackerOne during an engagement wrap-up meeting.

Any identified vulnerabilities were made available immediately through HackerOne's vulnerability management platform to ensure quick action can be taken by HackerOne.

2.5 Vulnerability classification and severity

To categorize vulnerabilities according to a commonly understood vulnerability taxonomy, HackerOne uses the industry standard Common Weakness Enumeration (CWE). CWE is a community-developed taxonomy of common software security weaknesses. It serves as a common language, a measuring stick for software security tools, and as a baseline for weakness identification, mitigation, and prevention efforts.

To rate the severity of vulnerabilities, HackerOne uses the industry-standard Common Vulnerability Scoring System (CVSS) to calculate severity for each identified security vulnerability. CVSS provides a way to capture the principal characteristics of a vulnerability, and produce a numerical score reflecting its severity, as well as a textual representation of that score.

To help prioritize vulnerabilities and assist vulnerability management processes, HackerOne translates the numerical CVSS rating to a qualitative representation (such as low, medium, high and critical):

- \\\\ **Critical:** CVSS rating 9.0 - 10
- \\\\ **High:** CVSS rating 7.0 - 8.9
- \\\\ **Medium:** CVSS rating 4.0 - 6.9
- \\\\ **Low:** CVSS rating 0.1 - 3.9
- \\\\ **None:** CVSS rating 0.0

More information about CWE can be found on MITRE's website: https://cwe.mitre.org/.

More information about CVSS can be found on the Forum for Incident Response and Security Teams' (FIRST) website: https://www.first.org/cvss.

2.6 HackerOne staff

The following individual at HackerOne managed this engagement and produced this report:

- **Eduardo Cervantes, Security Solutions Architect**
    - eduardo@hackerone.com

Please feel free to contact this individual with any questions or concerns you have around the engagement or this document.

2.7 HackerOne security testing team

3	119	757
HackerOne Security Researchers	Total HackerOne Customer Engagements Worked On	Total Vulnerabilities Found for HackerOne Customers

During the engagement, 3 hand-picked researchers participated in this assessment. 3 of the participating hackers submitted a valid vulnerability report. The first vulnerability was identified on March 16, 2021. Hackers from 4 different countries participated.

A full list of researchers that participated can be found in Appendix A.

3. Findings

This section contains the results of the security assessment. Findings are sorted by their severity and grouped by the asset and CWE classification. Each asset section will contain a summary. Table 1 in the executive summary contains the total number of identified security vulnerabilities per asset per risk indication. All findings were entered in the HackerOne platform, which is the authoritative source for the information on the vulnerabilities and can be referred to for details about each finding using the stated reference number in the asset vulnerability summary.

3.1 Findings Overview

During the engagement, 17 unique vulnerabilities were found across 10 different vulnerability categories (CWE). The most common vulnerability type was Information Disclosure with 6 unique reports. Vulnerabilities of the following kinds were identified:

- Information Disclosure
- Modification of Assumed-Immutable Data (MAID)
- Business Logic Errors

- Misconfiguration
- Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')
- Phishing
- Violation of Secure Design Principles
- Open Redirect
- Denial of Service
- Cross-Site Request Forgery (CSRF)

Exploring the findings further by their actual vulnerability type as defined by CWE, Table 3 shows the number of individual findings and its distribution of severity.

	Critical	High	Medium	Low	None
Information Disclosure	0	0	**1**	**4**	**1**
Modification of Assumed-Immutable Data (MAID)	0	0	**1**	**1**	0
Business Logic Errors	0	0	0	**2**	0
Misconfiguration	0	0	0	**1**	0
Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')	0	0	0	**1**	0
Phishing	0	0	0	**1**	0
Violation of Secure Design Principles	0	0	0	0	**1**
Open Redirect	0	0	0	**1**	0
Denial of Service	0	0	0	**1**	0

Cross-Site Request Forgery (CSRF)	0	0	0	**1**	0

Table 3: severity distribution across vulnerability types

Vulnerabilities were found in the following assets:

- hackerone.com
- hackerone.com/graphql

There were no vulnerabilities found in the following assets:

- api.hackerone.com/v1
- 66.232.20.0/23
- 206.166.248.0/23

3.2 Asset: hackerone.com

3.2.1 Asset Summary

This asset was the most vulnerable (based on quantity). Multiple different types of vulnerabilities were found, but only low severity ones.

3.2.2 Vulnerability Summary

During the security assessment, 11 security vulnerabilities were identified in this asset.

VULNERABILITY TITLE	SEVERITY	CWE
#1139541 Enumerating HackerOne Pentests	Low (3.7)	Misconfiguration
#1131473 CSRF allows to test email forwarding	Low (3.1)	Cross-Site Request Forgery (CSRF)

#1127453 Mapping victim's report ID's via time based XS leak attack	Low (3.1)	Information Disclosure
#1139520 Bypassing the External Link Warning	Low (3.1)	Open Redirect
#1131306 User's who are banned from program can still be invited to the new reports as collaborators	Low (3.1)	Business Logic Errors
#1133536 Temporary banned user (from platform) is able to make submissions via embedded submission forms	Low (3.1)	Business Logic Errors
#1132171 Race condition allows to send multiple times feedback for the hacker	Low (2.7)	Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')
#1127455 Hackers can reveal the names of private programs that have an external link	Low (2.6)	Information Disclosure
#1130235 Hackers can reveal the names of private programs that have an external link and Enterprise Product Edition	Low (2.6)	Information Disclosure
#1131887 CSV injection in the credentials export	None (0.0)	Violation of Secure Design Principles
#1128358 Used email confirmation link reveals the email address which is tied to it	None (0.0)	Information Disclosure

Table 4: findings in hackerone.com

3.2.3 Findings Summary

HackerOne Web Security Checklist METHODOLOGY	TEST RESULT	FINDINGS
Injection	✘	1 finding
Broken Authentication	✔	
Sensitive Data Exposure	✘	5 findings
XML External Entities (XXE)	✔	
Broken Access Control	✔	
Security Misconfiguration	✘	3 findings
Cross-Site Scripting (XSS)	✔	
Cross-Site Request Forgery (CSRF)	✘	1 finding
Insecure Deserialization	✔	
Using Components with Known Vulnerabilities	✔	
Unvalidated Redirects and Forwards	✘	1 finding

3.3 Asset: hackerone.com/graphql

3.3.1 Asset Summary

The most severe vulnerabilities (based on impact) were found on this asset and a few other low-severity logic issues were found as well.

3.3.2 Vulnerability Summary

During the security assessment, 6 security vulnerabilities were identified in this asset.

VULNERABILITY TITLE	SEVERITY	CWE
#1132606 Attachment object in GraphQL continues to grant access to files, even if they are removed from rendering	Medium (5.3)	Information Disclosure
#1139535 Changing the 2FA secret key and backup codes without knowing the 2FA OTP	Medium (4.6)	Modification of Assumed-Immutable Data (MAID)
#1128701 Lack warning label when receiving a letter	Low (3.1)	Phishing
#1138668 The possibility of disrupting the normal operation of frontend using markdown	Low (3.1)	Denial of Service
#1139528 Editing Pentest Summary Report Answers After Submitting Them	Low (3.1)	Modification of Assumed-Immutable Data (MAID)
#1129649 Hackers can find out the ID of private programs	Low (2.6)	Information Disclosure

Table 5: findings in hackerone.com/graphql

3.3.3 Findings Summary

HackerOne Web Security Checklist METHODOLOGY	TEST RESULT	FINDINGS
Injection	✔	
Broken Authentication	✔	
Sensitive Data Exposure	✘	2 findings
XML External Entities (XXE)	✔	
Broken Access Control	✘	2 findings
Security Misconfiguration	✘	2 findings
Cross-Site Scripting (XSS)	✔	
Cross-Site Request Forgery (CSRF)	✔	
Insecure Deserialization	✔	
Using Components with Known Vulnerabilities	✔	
Unvalidated Redirects and Forwards	✔	

Appendix A

HackerOne researchers

The following individuals were curated to participate in this pentest from HackerOne's community of over 900,000 hackers:

Username	Member Since	Reputation	# Of Lifetime Findings	# Of Programs Participated
muon4	March 2016	9,924	412	67
haxta4ok00	January 2016	3,774	226	33
whhackersbr	October 2014	1,564	119	28

End of Summary Report