hackerone

HackerOne Pentest
Security Assessment

MARCH 31ST, 2020 • CONFIDENTIAL

Description

This document details the process and result of a penetration test performed by HackerOne between February 24th, 2020 and March 9th, 2020.

Author

Antoine (Technical Program Manager, HackerOne)

antoine@hackerone.com

Reviewers

Thaddeus (Technical Program Manager, HackerOne)

thaddeus@hackerone.com

Prepared for:

HackerOne

Table of Contents

hackerone

1. Executive Summary

HackerOne (Customer) performed a HackerOne Pentest from February 24th, 2020 to March 9th, 2020 on its own applications. During this timeframe, 8 vulnerabilities were identified by 2 unique researchers.

During the assessment, 1 vulnerability was found that had a CVSS score of 7.0 or higher, rating either high or critical. These vulnerabilities represent the greatest immediate risk to HackerOne (Customer) and should be prioritized for remediation. 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
https://hackerone.com	0	**1**	**1**	**4**	**2**	**8**
https://api.hackerone.com	0	0	0	0	0	0
https://hackerone-us-west-2-production-attachments.s3-us-west-2.amazonaws.com/	0	0	0	0	0	0
	0	**1**	**1**	**4**	**2**	**8**

Table 1: Findings per asset

The security assessment was conducted using a crowd-sourced penetration testing methodology. From its community of over 700,000 hackers, HackerOne curated a set of top-tier researchers to focus on identifying vulnerabilities in HackerOne's (Customer) scope during the agreed-upon testing window, while abiding by the policies set forth by HackerOne (Customer). Chapter 2 contains more information about the methodology.

The most common vulnerability type was Business Logic Errors. The most severe vulnerability found was an Information Disclosure in https://hackerone.com. This vulnerability could have been used by program team members to reveal the personal emails of HackerOne users they invite to their program.

State of Security

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

Reviewing the remaining resolved 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 (Customer) assets.

Recommendations

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

KEY RECOMMENDATION	
Key Issue	HackerOne (Customer) has multiple Access Control vulnerabilities that allow users with less privileged roles to view/edit resources. Certain sensitive user and program information were retrievable via GraphQL.
Recommendation	For each new endpoint developed and published in the GraphQL schema, make sure to have an access control matrix of what each role should be able to access and have QA automated tests running for each new deployment. Since it can be hard doing this for each new functionality, it could make sense to just make the endpoint available to one particular rule and by default make it completely unavailable (both read/write) to all the other possible roles.

2. Methodology

HackerOne (Customer) performed a HackerOne Pentest. The following sections cover how the engagement was put together and performed.

2.1 Preparation phase

HackerOne (Customer) identified the types of vulnerabilities most important to them and understood 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;
- determine a sufficient testing window;
- identify the areas of HackerOne's (Customer) scope that researchers should pay special attention to;
- and what types of vulnerabilities HackerOne (Customer) 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 700,000 hackers, HackerOne curated a set of top-tier researchers to focus on identifying vulnerabilities in HackerOne's (Customer) 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 February 24th, 2020 to March 9th, 2020 was agreed-upon.

The contents of the Security Page were approved by HackerOne (Customer) 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
https://hackerone.com
https://api.hackerone.com
https://hackerone-us-west-2-production-attachments.s3-us-west-2.amazonaws.com/

Table 2: In-scope assets

2.1.2 Test plan

The selected researchers were able to create and use their own accounts in order to test for vulnerabilities within the agreed-upon scope. The researchers were given access to programs that represented a variety of product editions and had various features enabled. All testing took place in production environments.

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 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 February 24th, 2020 through March 9th, 2020.

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 great 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 (Customer) 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 (Customer) 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.

2.3 Retesting phase

While HackerOne (Customer) worked to resolve any identified vulnerabilities, HackerOne kicked off a retest of those findings to ensure they are no longer reproducible.

HackerOne believes in the power of many and uses its community of researchers to ensure vulnerabilities are not only fixed but are fixed thoroughly and a mitigation can't be bypassed. Each retest was completed individually by multiple researchers for increased confidence the deployed mitigation is functioning as intended.

2.4 Reporting phase

At the conclusion of the engagement, HackerOne worked with HackerOne (Customer) to analyze the results of the testing phase and identify any potential trends in vulnerabilities found across HackerOne's (Customer) 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 (Customer) 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 (Customer).

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:

- **Antoine, Technical Program Manager**
    - antoine@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

During the engagement, 3 hand-picked researchers participated in this assessment. The first vulnerability was identified on February 28, 2020. Hackers from 3 different countries participated.

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

3. Findings

This chapter 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, 8 unique vulnerabilities were found across 5 different vulnerability categories (CWE). The most common vulnerability type was Business Logic Errors with 3 unique reports. Vulnerabilities of the following kinds were identified:

- Business Logic Errors (CWE-840)
- Improper Authorization (CWE-285)
- Information Disclosure (CWE-200)
- Incorrect Authorization (CWE-863)
- Modification of Assumed-Immutable Data (MAID) (CWE-471)

Table 3 shows a visualization of how HackerOne's assets performed against the most common types of vulnerabilities as defined by the OWASP Top 10.

OWASP TOP 10 CATEGORY	TEST RESULT	FINDINGS
Injection	✔	Nothing Significant Discovered

Broken Authentication	✔	Nothing Significant Discovered
Sensitive Data Exposure	✘	2 Findings Report #812138 Report #807448
XML External Entities (XXE)	✔	Nothing Significant Discovered
Broken Access Control	✘	3 Findings Report #811138 Report #815467 Report #816143
Security Misconfiguration	✘	3 Findings Report #808975 Report #813300 Report #808755
Cross-Site Scripting (XSS)	✔	Nothing Significant Discovered
Cross-Site Request Forgery (CSRF)	✔	Nothing Significant Discovered
Insecure Deserialization	✔	Nothing Significant Discovered
Using Components with Known Vulnerabilities	✔	Nothing Significant Discovered
Unvalidated Redirects and Forwards	✔	Nothing Significant Discovered

Table 3: Vulnerabilities by OWASP Top 10 category

Exploring the findings further by their actual vulnerability type as defined by CWE, Table 4 shows the number of individual findings and its distribution of severity.

	Critical	High	Medium	Low	None
Business Logic Errors	0	0	0	**3**	0
Information Disclosure	0	**1**	**1**	0	0
Incorrect Authorization	0	0	0	0	**1**
Improper Authorization	0	0	0	0	**1**
Modification of Assumed-Immutable Data (MAID)	0	0	0	**1**	0

Table 4: Severity distribution across vulnerability types

Vulnerabilities were found in the following assets:

- https://hackerone.com

There were no vulnerabilities found in the following assets:

- https://api.hackerone.com
- https://hackerone-us-west-2-production-attachments.s3-us-west-2.amazonaws.com

3.2 Asset: https://hackerone.com

3.2.1 Asset Summary

This is the main application that handles the majority of user action. The asset allows the user to create and modify programs and personal profiles. It also handles vulnerability submissions. For this engagement, the main functionality tested related to programs and their relation to hackers and other programs.

3.2.2 Vulnerability Summary

During the security assessment, 8 security vulnerabilities were identified in this asset.

VULNERABILITY TITLE	SEVERITY	CWE
#807448 Customer private program can disclose email any users through invited via username	High (7.5)	Information Disclosure
#812138 Getting information about an endpoint `/sfdc_agile_accelerator_settings` via GraphQL who have permission `read-only`	Medium (4.4)	Information Disclosure
#808975 Rounding errors on rewarding a bounty leads to bypassing the 20% H1 commission fee	Low (3.5)	Business Logic Errors
#808755 Mismatch between frontend and backend validation via `ban_researcher` leads to H1 support and hackers email spam	Low (3.5)	Business Logic Errors
#813300 Changes to data in a CVE request after draft via GraphQL query	Low (2.6)	Modification of Assumed-Immutable Data (MAID)
#816143 A team member of the program with Report rights can ban the Admin	Low (2.0)	Business Logic Errors
#811138 Program owners are able to bypass hacker's invite preference by using username@wearehackerone.com	None (0.0)	Incorrect Authorization
#815467 Disclosure of private handles that conducted checks/discover	None (0.0)	Improper Authorization

Table 5: Findings in https://hackerone.com

4. Remediation Status

HackerOne (Customer) engaged HackerOne to retest the findings made during the assessment to ensure vulnerabilities were patched properly. HackerOne believes in the power of many and uses its community of researchers to ensure vulnerabilities are not only fixed but are fixed thoroughly and a mitigation can't be bypassed. Each retest was completed individually by multiple researchers for increased confidence the deployed mitigation is functioning as intended. Table 6 shows the remediation status of each finding.

VULNERABILITY TITLE	SEVERITY	REMEDIATION STATUS
#807448 Customer private program can disclose email any users through invited via username	High (7.5)	Fixed (Mar 27, 2020)
#812138 Getting information about an endpoint `/sfdc_agile_accelerator_settings` via GraphQL who have permission `read-only`	Medium (4.4)	Fixed (Mar 17, 2020)
#808755 Mismatch between frontend and backend validation via `ban_researcher` leads to H1 support and hackers email spam	Low (3.5)	Fixed (Mar 16, 2020)
#816143 A team member of the program with Report rights can ban the Admin	Low (2.0)	Fixed (Mar 16, 2020)
#808975 Rounding errors on rewarding a bounty leads to bypassing the 20% H1 commission fee	Low (3.5)	Fixed (Mar 20, 2020)
#813300 Changes to data in a CVE request after draft via GraphQL query	Low (2.6)	Fixed (Mar 17, 2020)
#815467 Disclosure of private handles that conducted checks/discover	None (0.0)	Fixed (Mar 30, 2020)
#811138 Program owners are able to bypass hacker's invite preference by using username@wearehackerone.com	None (0.0)	Fixed (Mar 17, 2020)

Table 6: Summary of findings and status of remediation

Appendix A

HackerOne researchers

The following individuals were curated to participate in this pentest from HackerOne's community of over 700,000 hackers:

Username	Member Since	Reputation	# Of Lifetime Findings	# Of Programs Participated
nahamsec	January 2014	17,245	668	108
haxta4ok00	January 2016	2,734	149	31
fisher	February 2015	2,648	122	31

End of Summary Report