



Case Study: Heroku Data Breach

By Vikrant Singh Chauhan
For Coursera Peer Graded Assignment
12th June, 2022

Table of Content

Account Takeover Attack

Data Breach

Timeline

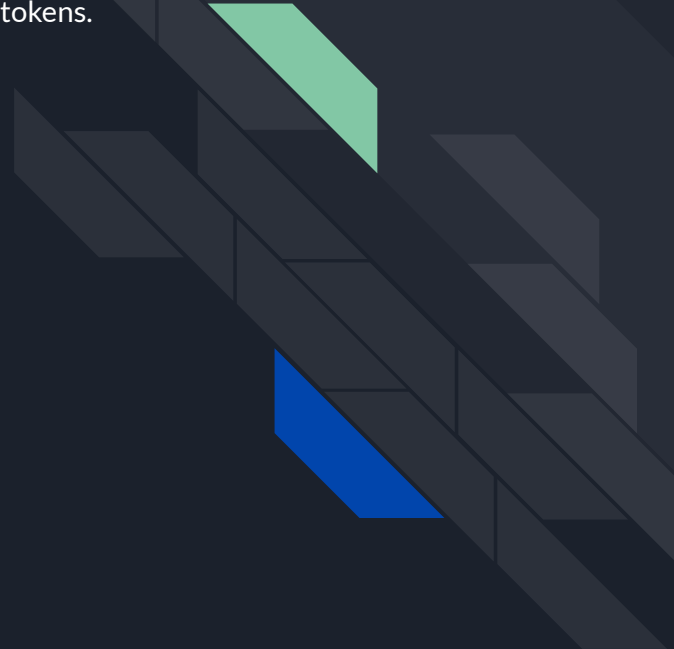
Vulnerabilities

Cost & Preventions

References

Summary

This Case Study looks at the recent security incident at Heroku that resulted in a major leak of private source code repositories via GitHub account takeovers via OAuth tokens.



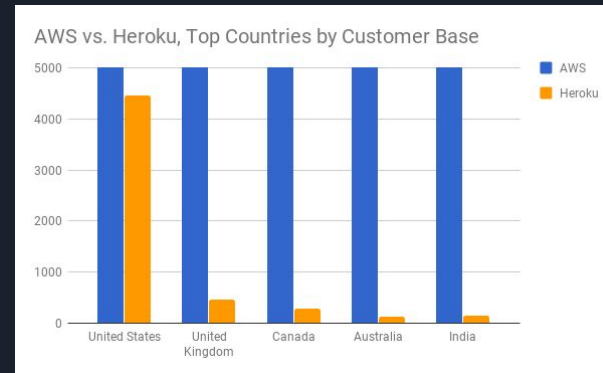
Account takeover

Account takeover is a form of identity theft and fraud, where a malicious third party successfully gains access to a user's account credentials. By posing as the real user, cyber-criminals can change account details, send out phishing emails, steal financial information or sensitive data, or use any stolen information to access further accounts within the organization.

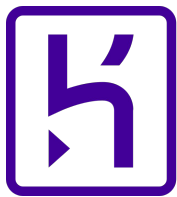
Once an attacker have control of an account, attackers can launch a variety of attacks, such as Internal phishing, Supply-chain phishing, Business Email Compromise attacks, Data exfiltration, Financial fraud etc.

Company Overview

Heroku is a container-based cloud Platform as a Service (PaaS) owned by Salesforce. It allows developers to deploy, manage, and scale modern apps. It is one of the best PaaS providers in the industry directly competing with the AWS (which is an Infrastructure as a Service Provider) in the US economy as per enlyft's sales and marketing intelligence report.



Source: <https://enlyft.com/resources/aws-vs-heroku-worldwide-market-share-compared>



HEROKU



GitHub

Data Breach

On April 13, 2022 Salesforce (Parent company of Heroku) received a report from GitHub that a subset of Heroku's GitHub private repositories, including some source code, were downloaded by a threat actor on April 9, 2022. After 2 days, Heroku notified its users publicly about the incident and starts the investigation.

The data breach at Heroku caused a major leak of GitHub OAuth tokens. These tokens were later used to enumerate GitHub users to download the private git repositories of the compromised accounts.



Timeline

- April 7, 2022 A threat actor obtained access to a Heroku database and downloaded stored customer GitHub integration OAuth tokens. On that same day, the threat actor downloaded data from another database that stores pipeline-level config vars for Review Apps and Heroku CI. Additionally, another small subset of Heroku users had their Heroku tokens exposed in a config var for a pipeline. Access to the environment was gained by leveraging a compromised token for a Heroku machine account.
- April 12, 2022 GitHub investigates the malicious activity and informs Heroku about the data breach the next day.
- April 15, 2022 Heroku publicly notifies the users about the incident and starts an investigation. Same day, Heroku releases a prevention advisory to its users to disable the Github integration.
- April 16, 2022 Heroku started revoking the existing OAuth tokens and by 5:00 p.m. PT, Heroku revoked all tokens. Heroku also disabled many features to ensure safety of its users.
- May 5, 2022 The investigation revealed that the same compromised token was leveraged to gain access to a database and exfiltrate the hashed and salted passwords for customers' user accounts. To mitigate, Heroku resets the user passwords.



Timeline


- | | |
|-----------------|---|
| May 7,
2022 | The investigation revealed that a small subset of user application related secrets stored in environment variables were leaked but the attacker didn't gain access to the encryption key necessary to decrypt the encrypted environment variables. Any users affected by these issues were notified directly and provided with additional guidance. |
| May 25,
2022 | Heroku enables all of the features again. |
| May 31,
2022 | Heroku concludes its investigation. Complete details of the attacker's actions is scheduled to be published on June 13, 2022. |



Vulnerabilities

Multiple Account Takeovers

While the complete details of the attack are still unreleased, there are hints from which we can guess that the incident started with some kind of social engineering attack. The attacker gained access to a machine through some sort of API by leveraging the token he compromised. After gaining the access to the machine, he pivoted to database servers and accessed all the stored OAuth tokens of GitHub and encrypted config vars of some applications hosted on Heroku. Then he started enumerating Github private repositories and downloaded source code and metadata related to those private repositories.





Cost

- The attack was severe and Critical Information Infrastructure (CII) was heavily damaged.
- The attacker downloaded private repositories which might include source code of many critical software.
- The attacker is now capable of finding more vulnerabilities with whitebox testing and do damage to those applications.
- Further, the attacker also accessed hashes of salted passwords and config vars.

Preventions

- The attack was mitigated by revoking the leaked tokens.
- The passwords were reset and salt was changed by Heroku to ensure that leaked passwords are not usable.
- The users were advised to keep track of security logs on GitHub and Heroku.



References

- Heroku Security Notification <https://status.heroku.com/incidents/2413>
- Heroku resets user passwords after concluding April cyber-attack ran deep <https://portswigger.net/daily-swig/heroku-resets-user-passwords-after-concluding-april-cyber-attack-ran-deep>
- Heroku admits that customer credentials were stolen in cyberattack <https://www.bleepingcomputer.com/news/security/heroku-admits-that-customer-credentials-were-stolen-in-cyberattack/>
- GitHub hacked, npm data stolen after OAuth tokens stolen in upstream breach <https://thestack.technology/github-hacked-npm-data-downloaded-in-an-evolving-supply-chain-attack/>
- So, what happened with GitHub, Heroku, and those raided private repos? <https://www.theregister.com/2022/04/21/github-stolen-oauth-tokens-used-in-breaches/>