Summary

On Wednesday Sept 27 we were notified about an Immunefi bug bounty report regarding the Governance Portal. The report described an exploit taking advantage of an issue with the gray-matter markdown library which allowed execution of arbitrary javascript code. During our investigation we concluded that the vulnerability was classified as “Critical” severity. While critical, the bug could not lead to any loss of user funds, therefore it was eligible for a bug bounty payout of 50,000 DAI.

Our team patched the issue with the gray-matter library and took other steps to protect the site. We investigated, but did not find any evidence that this bug was ever exploited in a real world context outside of the bug report.

Background

On September 25, a report was submitted by a whitehat hacker to the Immunefi bug bounty platform. The report went through initial screening by the Immunefi team. On September 27 a member of the Immunefi team escalated the report and contacted Jetstream to look into it. That day members of Jetstream begin investigating and trying to reproduce the issue. We were able to reproduce and took steps to patch the issue immediately.

The next day, upon further review, we realized additional avenues for the attack existed. We involved the TechOps team in our investigations and took further action to close off these additional avenues. After internal discussion we decided the bounty report should be classified as “critical”, but it could not directly lead to loss of user funds.

Detailed Analysis

The exploit as reported worked like this (simplified a little bit for clarity):

- on the voting portal, when connected to goerli, a bad actor could create a new poll. For the poll’s url

field, they could add a link to a markdown file hosted on a server that they control.

- While our production databases have a whitelist which only allows polls created by authorized Maker governance addresses, the goerli testing databases do not have a whitelist. Therefore polls created by any user can be scraped and added to the goerli testing database.
- In the markdown file, the bad actor can include a javascript block to execute arbitrary javascript code. This code would run on the server and could expose secret environmental variables.
- The malicious user would then visit the all-polls

endpoint on our frontend. The javascript block of malicious code would execute and reveal any secret environmental variables, such as passwords, api keys, or connection strings.

During our evaluation of this bounty report we reviewed the security keys that would have potentially been accessible to an attacker. Most of these are out of scope or not critical. For example API keys to Alchemy endpoints and connection to the comments database, both of which are out of scope for Immunefi bug bounties. In our assessment, none of the exposed environmental vars could have lead to a loss of funds.

Mitigation steps

To test the viability of this exploit we began to try and reproduce it ourselves. While working on our local dev environments we were able to verify that exposing environmental variables was indeed possible. Then we tested it in a staging environment by creating a poll which linked to a mock malicious markdown file. We discovered that the vulnerability was reproducible in the remote environments.

Upon discovering that the bug report was valid we immediately deployed a hotfix to the gov portal patching the issue with the gray-matter library. This removed the ability to execute any javascript code, rendering the attack impossible. We also worked closely with TechOps to include a whitelist on our goerli databases to prevent unauthorized users from adding polls. Then, we updated all our current environmental variables to new values, including passwords and connection strings, so any previously exposed variables would no longer work. Finally, we went through all the existing polls in the goerli database to verify that no other instances of this vulnerability were ever exploited, and confirmed that there were none.