

RISK AUDIT

for

SAFEDROP

on

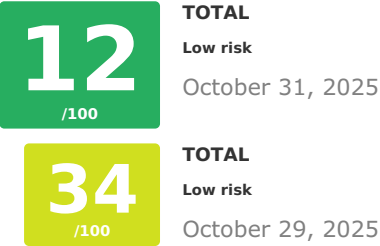
Oct 29, 2025



FIDESIUM

Executive Summary

Report



Abstract

Fidesium was asked to perform an audit on SafeDrop's codebase

Repository Link: <https://github.com/digdef/safedrop-back>

Repository Link: <https://github.com/digdef/safedrop-front>

Issue Summary



Caveats

The codebase is generally well written, but does incur a handful of flaws.

Methodology

The assessment methodology covered a range of phases and employed various tools, including but not limited to the following:

- Mapping Functionality of API
- Application Logic Flaws
- Access Handling
- Authentication/Authorization Flaws
- Brute Force Attempt
- Input Handling
- Source Code Review
- Fuzzing of all input parameter
- Dependency Analysis

Test Approach

Fidesium performed both Whitebox and Blackbox testing, as per the scope of the engagement, and relied on automated security testing.

Severity Definitions

Critical	The issue can cause large economic losses, large-scale data disorder or loss of control of authority management.
High	The issue puts users' sensitive information at risk or is likely to lead to catastrophic financial implications.
Medium	The issue puts a subset of users' sensitive information at risk, reputation damage or moderate financial impact.
Low	The risk is relatively small and could not be exploited on a recurring basis, or is low-impact to the client's business.
Informational	The issue does not pose an immediate risk but is relevant to security best practices or defence in Depth.

Risk Issues

Vulnerability	Description	Risk	Probability	Status
NextJS Authorization bypass	The version of NextJS used is vulnerable to authorization bypass	Critical	Medium	Resolved
ShaJS Input Manipulation	The version of ShaJS used allows input manipulation	Critical	Medium	Resolved
Supply Chain Injection: Potential wallet drainer	@metamask/sdk nad @metamask/sdk-communication-layer were compromised	Critical	Info	Resolved
Denial of Service	The version of multer is vulnerable to DoS	High	Medium	Resolved
Secrets Leak	Privileged API responses are logged as plaintext	High	Medium	Resolved
XSS	The version of validator.js is vulnerable to XSS	Medium	Medium	Active
Silent Failures: Partial Data	The binance API calls do not properly handle errors	Medium	Medium	Resolved
Misleading error handling	API errors are treated as succesful but empty lookups	Medium	Medium	Resolved
Credential Collection	safedrop-front collects API credentials	Medium	High	Remediated
File Upload Code Injection	The version of @nestjs/common in safedrop-back is vulnerable to code injection via file upload	Medium	Medium	Resolved
Lack of results caching could lead to duplicate payment	Results are neither stored nor cached and charged per query	Low	Medium	Active
Denial of Service: Prototype Pollution	The version of fast-redact in safedrop-front is vulnerable to prototype pollution	Low	Low	Active

Risk Overview

Team Risk

Low risk: 1

No issues found in founding team

Doxxing Status	Team Experience	Risk Summary
Public	Highly relevant	Low

Smart Contract Risks

Risk summary: 21

The contracts are well written, and secure with only a few minor issues..

Vulnerabilities **Critical**

RESOLVED: NextJS Authorization bypass

Vulnerability severity: **Critical**

Vulnerability probability: **Medium**

The version of NextJS used in safedrop-back is vulnerable to authorization bypass

Known CVEs affecting this version:

- <https://nvd.nist.gov/vuln/detail/CVE-2025-29927>
- <https://nvd.nist.gov/vuln/detail/CVE-2025-57822>
- <https://nvd.nist.gov/vuln/detail/CVE-2025-57752>
- <https://nvd.nist.gov/vuln/detail/CVE-2025-55173>

Recommendations:

Upgrade next.js to at least version 15.4.7

Action Taken:

next.js upgraded.

RESOLVED: ShaJS Input Manipulation

Vulnerability severity: **Critical**

Vulnerability probability: **Medium**

The version of ShaJS used in safedrop-front is vulnerable to input manipulation

Known CVEs affecting this version:

- <https://nvd.nist.gov/vuln/detail/CVE-2025-9288>

Recommendations:

Upgrade sha.js to at least version 2.4.12

Action Taken:

sha.js upgraded.

RESOLVED: Supply Chain Injection: Potential wallet drainer

Vulnerability severity: **Critical**

Vulnerability probability: **Info**

@metamask/sdk nad @metamask/sdk-communication-layer were compromised

Known Github advisories:

- <https://github.com/MetaMask/metamask-sdk/security/advisories/GHSA-qj3p-xc97-xw74>

Recommendations:

Verify that you are not caching (npm, pnpm, yarn, etc) debug@4.4.2 either locally or in any deployed environment. To be on the safe side, we recommend deleting node_modules as well as any node_module caches in CI, and reinstalling.

Action Taken:

Safedrop team verified their deployed caches

Vulnerabilities **High**

RESOLVED: Secrets Leak

Vulnerability severity: **High**

Vulnerability probability: **Medium**

Privileged API responses are logged as plaintext

`this.fileLoggerService.log` is used throughout safedrop-back to log API responses as is, with no validation or sanitization.

Recommendations:

Do not log these responses

Action taken:

Safedrop team removed response logging

RESOLVED: Denial of Service

Vulnerability severity: **High**

Vulnerability probability: **Medium**

The version of multer in safedrop-back is vulnerable to DoS

Known CVEs affecting this version:

- <https://nvd.nist.gov/vuln/detail/CVE-2025-48997>
- <https://nvd.nist.gov/vuln/detail/CVE-2025-47944>

Recommendations:

Upgrade to at least 2.0.2

Action Taken:

multiparty has been upgraded

Vulnerabilities **Medium**

RESOLVED: Silent Failures: Partial Data

Vulnerability severity: **Medium**

Vulnerability probability: **Medium**

The binance API calls do not properly handle errors

```
catch (error) {  
  this.logger.error(error);  
}
```

Execution continues, and failed API calls are silently ignored

Recommendations:

Add error handling.

Action Taken:

Error handling was added

RESOLVED: Misleading error handling

Vulnerability severity: **Medium**

Vulnerability probability: **Medium**

API errors are treated as succesful but empty lookups

```
return { found: false };
```

Recommendations:

Distinguish API errors from wallet absence

Action Taken:

API errors got separated out

Vulnerabilities **Medium**

XSS

Vulnerability severity: **Medium**

Vulnerability probability: **Medium**

The version of validator.js in safedrop-back is vulnerable to XSS

Known CVEs affecting this version:

- <https://nvd.nist.gov/vuln/detail/CVE-2025-56200>

Recommendations:

Upgrade validator.js to at least 13.15.20

RESOLVED: File Upload Code Injection

Vulnerability severity: **Medium**

Vulnerability probability: **Medium**

The version of @nestjsjs/common in safedrop-back is vulnerable to code injection via file upload

Known CVEs affecting this version:

- <https://nvd.nist.gov/vuln/detail/CVE-2025-29409>

Recommendations:

Upgrade @nestjsjs/common to at least 10.4.16

Action Taken:

@nestjsjs/common was upgraded

REMEDIATED: Credential Collection

Vulnerability severity: **Medium**

Vulnerability probability: **Medium**

safedrop-front collects API credentials

While nothing is necessarily done with these credentials, this looks very suspicious, and opens up a potential vulnerability if an attacker can listen to these requests or interrogate the deployed containers.

Recommendations:

- Clearly explain why you need these keys
- Clearly define the minimum permissions you need these keys to have, and provide guidance on configuring these permissions
- Where possible, use your own predefined API keys and avoid collecting user keys entirely

Action Taken:

Remediated: Explanation is provided to customers and UI includes instructions on least privilege

Vulnerabilities **Low**

Lack of results caching could lead to duplicate payment

Vulnerability severity: **Low**

Vulnerability probability: **Low**

Results are neither stored nor cached and charged per query

This can mean that customers may double pay for the same information.

Recommendations:

Allow for timestamped retrieval of past results, and potentially an interrogation of the fact that "anything" has changed before paying for newer data.

Denial of Service: Prototype Pollution

Vulnerability severity: **Low**

Vulnerability probability: **Low**

The version of fast-redact in safedrop-front is vulnerable to prototype pollution

Known CVEs affecting this version:

- <https://nvd.nist.gov/vuln/detail/CVE-2025-57319>

Recommendations:

As a safe version of fast-redact does not exist, replace the library with slow-redact. The performance implication for real world loads is measure in microseconds.

Vulnerabilities Info

Disclaimer

Disclaimer

This report is governed by the Fidesium terms and conditions.

This report does not constitute an endorsement or disapproval of any project or team, nor does it reflect the economic value or potential of any related product or asset. It is not investment advice and should not be used as the basis for investment decisions. Instead, this report provides an assessment intended to improve code quality and mitigate risks inherent in cryptographic tokens and blockchain technology.

Fidesium does not guarantee the absence of bugs or vulnerabilities in the technology assessed, nor does it comment on the business practices, models, or regulatory compliance of its creators. All services, reports, and materials are provided "as is" and "as available," without warranties of any kind, including but not limited to merchantability, fitness for a particular purpose, or non-infringement.

Cryptographic assets and blockchain technologies are novel and carry inherent technical risks, uncertainties, and the possibility of unpredictable outcomes. Assessment results may contain inaccuracies or depend on third-party systems, and reliance on them is solely at the Customer's risk.

Fidesium assumes no liability for content inaccuracies, personal injuries, property damages, or losses related to the use of its services, reports, or materials. Third-party components are provided "as is," and any warranties are strictly between the Customer and the third-party provider.

These services and materials are intended solely for the Customer's use and benefit. No third party or their representatives may claim rights to or rely on these services, reports, or materials under any circumstances.