Fireblocks APIs and SDK Black Box Review

2023-12-15

By CoinFabrik

v202311

Executive Summary

CoinFabrik performed a black box review of some parts of the Fireblocks APIs and SDK. We focused our analysis on the sandbox:

https://sandbox-api.fireblocks.io

During this review we found 2 high issues, 1 medium issues and 1 minor issues. Also, several enhancements were proposed.

Scope

The scope for this review includes and is limited to the following endpoint categories:

- Vaults
- Network Ids
- Internal Wallets
- External Wallets
- NFTs

These represent a small subset of all the endpoints exposed for the API version 1.6.2 in: https://docs.fireblocks.com/api/swagger-ui/

Methodology

CoinFabrik made a black box analysis of the Fireblocks API and SDK and general documentation about the project. Our auditors spent one week reviewing the API and SDK, which includes understanding the context of use, analyzing the boundaries of the expected behavior of each endpoint and identifying possible situations in which the code allows the caller to reach a state that exposes some vulnerability. Without being limited to them, the review process included the following analyses:

- Poor or nonexistent error handling
- Insufficient validation of the input parameters
- Incorrect handling of cryptographic signatures

Findings

In the following table we summarize the security issues we found in this review. The severity classification criteria and the status meaning are explained below. This table does not include the enhancements we suggest to implement, which are described in a specific section after the security issues.

ID	Title	Severity	Status
HI-01	Arbitrary API Call in `get_transaction_with_pay_info()`	High	Unresolved
HI-02	Improper Input Validation in `create_vault_account()`	High	Unresolved
ME-01	Server Error in `create_network_id()`	Medium	Unresolved
MI-01	Possible Data Leak in `get_network_id()`	Minor	Unresolved

Severity Classification

Security risks are classified as follows:

- **Critical:** These are issues that we manage to exploit. They compromise the system seriously. Blocking bugs are also included in this category. They must be fixed **immediately**.
- **High:** These refer to a vulnerability that, if exploited, could have a substantial impact, but requires a more extensive setup or effort compared to critical issues. These pose a significant risk and **demand immediate attention**.
- **Medium:** These are potentially exploitable issues. Even though we did not manage to exploit them or their impact is not clear, they might represent a security risk in the near future. We suggest fixing them **as soon as possible**.
- **Minor:** These issues represent problems that are relatively small or difficult to take advantage of, but might be exploited in combination with other issues. These kinds of issues do not block deployments in production environments. They should be taken into account and be fixed **when possible**.

Issues Status

An issue detected by this audit has one of the following statuses:

- **Unresolved**: The issue has not been resolved.
- **Acknowledged**: The issue remains in the code, but is a result of an intentional decision. The reported risk is accepted by the development team.
- **Resolved**: Adjusted program implementation to eliminate the risk.
- **Partially resolved**: Adjusted program implementation to eliminate part of the risk. The other part remains in the code, but is a result of an intentional decision.
- **Mitigated**: Implemented actions to minimize the impact or likelihood of the risk.

High Severity Issues

HI-01 Arbitrary API Call in get_transaction_with_pay_info()

Location:

- `fireblocks.get_transaction_with_pay_info() API`
- `sdk.py: 862`

Classification:

- CWE-227: API Abuse

The `get_transactions_with_page_info()` API has several parameters. The parameter called "`next_or_previous_path`" has the following description:

"`get transactions matching the path, provided from pageDetails`"

And the code processing it is the following:

```
def get_transactions_with_page_info(
        self ,
        ... ,
        next_or_previous_path=None
):
        index = next_or_previous_path.index("/v1/")
        length = len(next_or_previous_path) - 1
        suffix_path = next_or_previous_path[index:length]
        return self._get_request(suffix_path, True)
```

This code allows calling any API by simply specifying the path inside the "`next_or_previous_path`" parameter. For example, this code will indirectly call the `get_staking_positions_summary()` API, by abusing the `get_transaction_with_page_info()` call:

```
b=fireblocks.get_transactions_with_page_info(after=XSS,before=XSS,txhash=XSS*10,assets=XSS*10,source_id=XSS, next_or_previous_path="/v1/staking/positions/summaryy")
```

Note that the "sumaryy" parameter string is not a typo as the call requires an additional character at the end. In this way, if a malicious user can control the `next_or_previous_path` parameter, it could potentially call any API inside the application.

Recommendation

Validate the "`next_or_previous_path`" parameter of the `get_transaction_with_page_info()` function in the API SDK, or update the documentation accordingly.

Status

Unresolved

HI-02 Improper Input Validation in create_vault_account()

Location:

- `fireblocks.create_vault_account() API`
- `sdk.py: 1210`

Classification:

- CWE-20: Improper Input Validation

The API call `create_vault_account()` has a single parameter "`name`" whose length is not limited. By creating a vault with a huge length, the web page fails to render properly. This effectively prevents a vault created in this way to be manually removed via the web interface. To reproduce this error just execute this code after authentication:

```
vault_account = fireblocks.create_vault_account(name = "Quick'';!--\"<XSS>=&{()}art_Vault"*1000)
```

Recommendation

Limit the vault name length to a reasonable value, in the API call and also at the web application level.

Status

Unresolved

Medium Severity Issues

ME-01 Server Error in create_network_id()

Location:

- `fireblocks.create_network_id() API`
- `sdk.py: 655`

Classification:

- CWE-392: Missing Report of Error Condition
- CWE-248: Uncaught Exception

When calling the API `create_network_id()` with an arbitrary name like in the following code:

```
fireblocks = FireblocksSDK(api_secret, api_key, api_base_url=api_url)
b=fireblocks.create_network_id("AAA")
```

It receives an error 500 with the message `"Request failed with status code 500"`, instead of a success or failure message. This usually means there is an internal server error.

Recommendation

Catch the exception and return a descriptive error message.

Status

Unresolved

Minor Severity Issues

M1-01 Possible Data Leak in get_network_id()

Location:

- `fireblocks.get_network_id() API`
- `sdk.py: 666`

Classification:

- CWE-1390: Weak Authentication

The API call `get_network_id()` allows to get any defined network information, for example this call:

```
id="f857c2f0-48f1-4e80-8e85-87bec6c17cb0"
fireblocks.get_network_id(id)
```

Will reveal the network name as "`Mastercard -`" with the private user name. It is not clear if this is the intended behavior of the sandbox or a data leak.

Recommendation

Check if `get_network_id()` returns sensitive data by just having the Id, a parameter that is easily enumerable/discoverable using other API calls.

Status

Unresolved

Enhancements

These items do not represent a security risk. They are best practices that we suggest implementing.

Table

ID	Title	Status
EN-01	Double Identification Factor for Sandbox	Not implemented
EN-02	Unclear Onboarding Procedure	Not implemented

Details

EN-01 Double Identification Factor for Sandbox

There is some instability in the 2-factor authentication procedure. Every time we tried to get into the console, 2FA failed and eventually worked.

Recommendation

As the Sandbox environment is for non production purposes, there is no need for a 2FA implementation for login.

Status
Not implemented.

EN-02 Unclear Onboarding Procedure

When you start following onboarding instructions, accessing the Developer Sandbox Quickstart Guide (link here), there is a mention of getting the API key and the Private key. This process is not documented, and as a result, a significant barrier is placed in front of new users.

After several attempts we discovered that specific email accounts are needed (for example, not gmail accounts), you need to fill the Sandbox Sign Up Form (link here), validate your email address, receive the welcome email (seems to be a manual approval process because takes some time) and, finally, do a 2-factor authentication validation that fails repeatedly but eventually works.

Recommendation
Create a step-by-step procedure to get API keys and check if 2FA procedures have flaws or see if accessing a Developer Sandbox really needs a 2FA.

Status
Not implemented.

Other Considerations

The considerations stated in this section are not right or wrong. We do not suggest any action to fix them. But we consider that they may be of interest for other stakeholders of the project, including users of the audited contracts, owners or project investors.

WAF

While testing the APIs, a web-application-firewall was detected that prevented some calls involving SQL-type strings. But it failed to stop any call involving malformed input with invalid characters, or overtly long parameters, for example:

```
vault_account = fireblocks.create_vault_account(name = "Quick'';!--\"<XSS>=&{()}art_Vault"*1000)
```

This overtly long API call containing HTML tags was not detected by the WAF, this allowed the call to be accepted and cause an issue (HI-02).

Changelog

- 2023-12-15 – Initial report based on API version 1.6.2.

Disclaimer: This review report is not a security warranty, investment advice, or an approval of the Fireblocks project since CoinFabrik has not reviewed its platform. Moreover, it does not provide a smart contract code and/or back-end faultlessness guarantee.