Asigna Audit

January 2024

By CoinFabrik

v202311

Executive Summary

CoinFabrik was asked to audit the multisig creation scripts for the Asigna project. The auditors were provided with the source file `generateWshMultisigAddress.ts` with checksum sha256:`594ca12d55f26209052a7aedae9e0717b7db4a5d6016d98ee2a67d55185cd127`.

During this audit we found 1 critical issue, and one enhancement was proposed. All issues were resolved.

Scope

The audited files were provided by the client. No github repository or commit hashes were provided.

The scope for this audit includes and is limited to the following files:

- `generateWshMultisigAddress.ts`: Multisig creation for the Bitcoin blockchain

No other files in this repository were audited. Its dependencies are assumed to work according to their documentation. Also, no tests were reviewed for this audit.

Methodology

CoinFabrik was provided with the source code. Our auditors spent one week auditing the source code provided, which includes understanding the context of use, analyzing the boundaries of the expected behavior of each contract and function, understanding the implementation by the development team (including dependencies beyond the scope to be audited) 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 audit process included the following analyses.

- Arithmetic errors
- Race conditions
- Misuse of block timestamps
- Denial of service attacks
- Needlessly complex code and contract interactions
- Poor or nonexistent error handling
- Insufficient validation of the input parameters
- Incorrect handling of cryptographic signatures

After delivering a report with our findings, the development team had the opportunity to comment on every finding and fix the issues they considered convenient. Once fixed and/or commented, our team ran a second review process to verify that the changes to the code effectively solve the issues found and do not unintentionally add new ones. This report includes the final status after the second review.

Findings

In the following table we summarize the security issues we found in this audit. 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
CR-01	Missing Input Validation On Multisig Threshold	Critical	Resolved

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.

Critical Severity Issues

CR-01 Missing Input Validation On Multisig Threshold

Location:

- `generateWshMultisigAddress.ts:11`

Classification:

- CWE-20: Improper Input Validation[1]

The function `generateWshMultisig()` uses the threshold argument to set the signature threshold of the multisig. The function validates that this threshold is not zero, and that is less than the amount of public keys provided, however, a negative threshold is still allowed.

As the library function `bitcoinjs-lib/payments/p2ms.ts` also fails to check for negative values, an invalid/corrupted multisig script will be generated if a negative threshold is allowed.

Recommendation

Check that no negative threshold is allowed while generating the multisig address. Also if possible, establish and check for a maximum amount of public keys allowed.

Status

Resolved. Added additional checks on code.

[1] https://cwe.mitre.org/data/definitions/20.html

High Severity Issues

No high severity issues found.

Medium Severity Issues

No medium severity issues found.

Minor Severity Issues

No minor severity issues found.

Enhancements

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

ID	Title	Status
EN-01	Use Bitcoinjs Standard Libs	Implemented

EN-01 Use Bitcoinjs Standard Libs

Location:

- `generateWshMultisigAddress.ts:4`

The script uses `@bitcoinerlab/secp256k1` as the `secp256k1` ecc library, and while this is a recommended alternative, the bitcoinjs project states that there is no guarantee that this lib will keep up with any interface change in the future (see https://github.com/bitcoinjs/tiny-secp256k1?tab=readme-ov-file#alternatives).

Recommendation

If possible, use the standard `tiny-secp256k1` lib from the bitcoinjs project by adding this line:

```
import * as ecc from 'tiny-secp256k1';
```

Status

Implemented.

Changelog

- 2024-02-05 – Initial report.
- 2024-02-05 – Fixes delivered and checked

Disclaimer: This audit report is not a security warranty, investment advice, or an approval of the Asigna project since CoinFabrik has not reviewed its platform. Moreover, it does not provide a smart contract code faultlessness guarantee.