BlockBoltPay

Audit Report

contact@movebit.xyz

https://twitter.com/movebit_

Thu Aug 03 2023

BlockBoltPay Audit Report

1 Executive Summary

1.1 Project Information

Description	Simple, Secure and Fast Decentralized Borderless Payment Protocol，along with a 60-line code implementation.
Type	DeFi
Auditors	MoveBit
Timeline	Fri Jul 28 2023 – Fri Jul 28 2023
Languages	Move
Platform	Sui
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/blockboltpay/boltpay-sui-contract
Commits	aac2a9e79888314134cdb525f46bbf916500a369

1.2 Files in Scope

The following are the SHA1 hashes of the original reviewed files.

ID	File	SHA-1 Hash
MOV	Move.toml	cad19f6be364c4cfd99ffaef9933068e14442362
ID	sources/id.move	454944a3266475374ea6732e3e953e69e9440b12

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	1	0	1
Informational	1	0	1
Minor	0	0	0
Medium	0	0	0
Major	0	0	0
Critical	0	0	0

1.4 MoveBit Audit Breakdown

MoveBit aims to assess repositories for security-related issues, code quality, and compliance with specifications and best practices. Possible issues our team looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Integer overflow/underflow by bit operations
- Number of rounding errors
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting
- Unchecked CALL Return Values
- The flow of capability
- Witness Type

1.5 Methodology

The security team adopted the **"Testing and Automated Analysis"**, **"Code Review"** and **"Formal Verification"** strategy to perform a complete security test on the code in a way that is closest to the real attack. The main entrance and scope of security testing are stated in the conventions in the "Audit Objective", which can expand to contexts beyond the scope according to the actual testing needs. The main types of this security audit include:

(1) Testing and Automated Analysis

Items to check: state consistency / failure rollback / unit testing / value overflows / parameter verification / unhandled errors / boundary checking / coding specifications.

(2) Code Review

The code scope is illustrated in section 1.2.

(3) Formal Verification

Perform formal verification for key functions with the Move Prover.

(4) Audit Process

- Carry out relevant security tests on the testnet or the mainnet;
- If there are any questions during the audit process, communicate with the code owner in time. The code owners should actively cooperate (this might include providing the latest stable source code, relevant deployment scripts or methods, transaction signature scripts, exchange docking schemes, etc.);
- The necessary information during the audit process will be well documented for both the audit team and the code owner in a timely manner.

2 Summary

This report has been commissioned by BlockBolt to identify any potential issues and vulnerabilities in the source code of the BoltPay smart contract, as well as any contract dependencies that were not part of an officially recognized library. In this audit, we have utilized various techniques, including manual code review and static analysis, to identify potential vulnerabilities and security issues.

During the audit, we have identified 1 issues of varying severity, listed below.

ID	Title	Severity	Status
ID-1	Off-chain Indexing Recommendation	Informational	Acknowledged

3 Participant Process

Here are the relevant actors with their respective abilities within the BoltPay Smart Contract:
User

- User can generates a unique transaction id (MerchantDetails), emits a MerchantDetailsEvent, and transfers the new MerchantDetails to the user by `transaction_identifier` function.

4 Findings

ID-1 Off-chain Indexing Recommendation

Severity: Informational

Status: Acknowledged

Code Location:

sources/id.move#L32-36

Descriptions:

The contract contains an issue where the event `MerchantDetailsEvent` is not properly indexed. Indexing event fields can enhance the accessibility and efficiency of off-chain tools that parse events, especially when filtering based on specific addresses. However, in the provided contract, the event fields lack the necessary indexing, potentially leading to performance bottlenecks and delays when parsing and filtering events off-chain.

Suggestion:

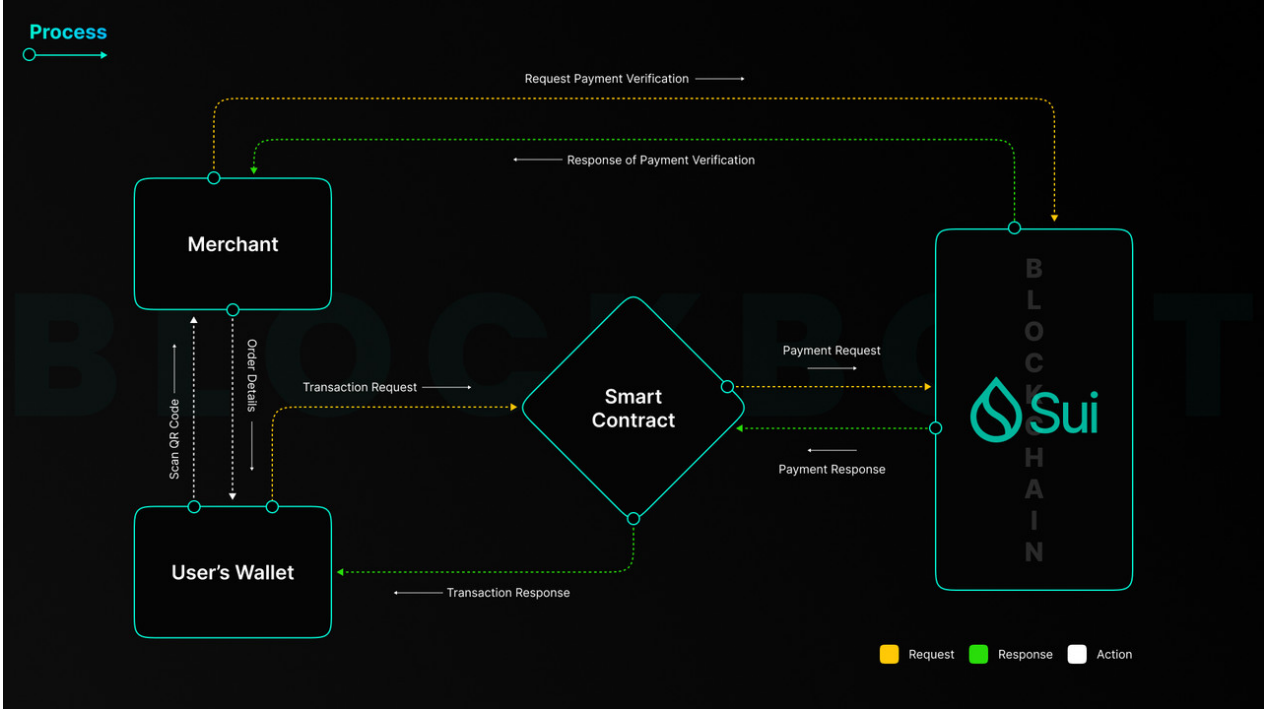
To improve the contract's efficiency and performance, it is recommended to add indexing to the fields of the `MerchantDetailsEvent` event.By adding indexing, event filtering and querying will be significantly expedited, but it is important to exercise caution to avoid unnecessary high gas costs. Depending on the specific usage scenario and gas cost considerations, the number of indexed fields can be adjusted accordingly to ensure optimal contract performance. Fixing this issue will enhance the contract's security, improve its efficiency, and result in better user experience by reducing delays in parsing and filtering events off-chain.

Resolution:

Feedback from the BlockBolt project team:

Our payment system is completely decentralized and consists of two secure and verified segments: Merchants (Clients) and Wallets. These two entities operate independently and are not interconnected in any way. Each has a decentralized process for retrieving information, making payments, and verifying.

For a better understanding, please take a look at the following flow chart.



Our BlockBolt payment protocol is entirely decentralized and does not interact with off–chain data. We provide transaction data for merchant websites to store on their servers for off–chain data. The filtering and parsing of this data are solely dependent on the merchant's side. However, our client SDK provides blockchain transaction data which can be utilized for off–chain data storage on the merchant's website. By following our Boltpay Client SDK guidelines, merchants can easily streamline the storage process on their websites. Merchants can streamline the analysis of on–chain events by implementing an indexing solution. This approach accelerates the retrieval of data, while also saving time and computational resources required for parsing and filtering on–chain data.

Appendix 1

Issue Level

- **Informational** issues are often recommendations to improve the style of the code or to optimize code that does not affect the overall functionality.
- **Minor** issues are general suggestions relevant to best practices and readability. They don't post any direct risk. Developers are encouraged to fix them.
- **Medium** issues are non-exploitable problems and not security vulnerabilities. They should be fixed unless there is a specific reason not to.
- **Major** issues are security vulnerabilities. They put a portion of users' sensitive information at risk, and often are not directly exploitable. All major issues should be fixed.
- **Critical** issues are directly exploitable security vulnerabilities. They put users' sensitive information at risk. All critical issues should be fixed.

Issue Status

- **Fixed:** The issue has been resolved.
- **Partially Fixed:** The issue has been partially resolved.
- **Acknowledged:** The issue has been acknowledged by the code owner, and the code owner confirms it's as designed, and decides to keep it.

Appendix 2

Disclaimer

This report is based on the scope of materials and documents provided, with a limited review at the time provided. Results may not be complete and do not include all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your own risk. A report does not imply an endorsement of any particular project or team, nor does it guarantee its security. These reports should not be relied upon in any way by any third party, including for the purpose of making any decision to buy or sell products, services, or any other assets. TO THE FULLEST EXTENT PERMITTED BY LAW, WE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, IN CONNECTION WITH THIS REPORT, ITS CONTENT, RELATED SERVICES AND PRODUCTS, AND YOUR USE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NOT INFRINGEMENT.