BiPOLE

Blockchain Audit Report

brise-chain

2023-02-10

BiPOLE Labs
Email: team@bipole.org

CONTENTS

Summary

This report was created for brise-chain in order to identify bugs and vulnerabilities in the contract dependencies that were not a part of an officially recognized library as well as project's source code.
Static Analysis and Manual Review approaches have been used to conduct a thorough examination.

The following factors receive extra consideration during the auditing process:
testing the smart contracts for both typical and unusual attack vectors.

* evaluating the codebase to make sure it adheres to the most recent industry standards and best practices.
* ensuring that contract logic adheres to the client's requirements and goals.
* Cross-referencing contract design and execution with related smart contracts created by top industry producers.
* Complete manual line-by-line review of the complete codebase by specialists in the field.

Findings from the security evaluation ranged from important to informative. To guarantee a high degree of security standards and industry practices, we advise taking action on these results. We offer suggestions that, from a security standpoint, could benefit the project.:

* Improve general coding techniques for improved source code organization;
* Include sufficient unit tests to cover all potential use cases.;
* For improved readability, include extra comments for each function, especially for contracts that are publicly verifiable;

* Once the protocol is active, provide additional transparency on privileged operations.

Project Introduction

Brise Chain brings EVM-compatible programmability and native cross-chain communication with Brise Chain using an innovative consensus of Proof of Authority(PoA). Brise Chain relies on a system with Proof of Authority (PoA) consensus that can support short block time and lower fees.

Audit Scope

ID	Filename	SHA256 CHECK SUM
Project	https://github.com/Bitgert/brise-chain	6b7fcd3977a606a3e9f0fa2d4078263e7e2a32da

Audit Methodology

Step	Operations	Description
1	Backgroud	Reading the descriptions, white papers, contract source code, and other relevant information the project team provides to ensure a proper understanding of project functions.
2	Automated Testing	Automated detection tools will be mainly used to scan the source code to find common potential vulnerabilities
3	Manual Review	The code will be thoroughly reviewed line by line by engineers to find potential vulnerabilities
4	Logic Proofread	The engineer will compare the understanding of the code with the information provided by the project and check whether the code implementation is in line with the white paper information.
5	Test Cases	Including test case design, test scope analysis, symbolic execution, etc.
6	Optimization	Items Review the project from the aspects of maintainability, security and operability according to the application scenarios, call methods and the latest research results.

Risk Levels

Risk Level	Issue Description
Critical	Fatal risks and hazards that need to fixed immediately.
Major	Some high risks and hazards that will lead to related problems that must be solved.
Medium	Some moderate risks and pitfalls may lead to potential risks that will eventually need to be addressed.
Minor	There are low risks and hazards, mainly details of various types of mishandling or warning messages, which can be set aside for the time being.
Informational	Some parts can be optimized, such problems can be shelved, but it is recommended that the final solution.

Vulnerability Summary

ID	Title	Category	Severity	Status
Information-01	Package Vulnerability Analysis	Passed	Informational	Acknowledged
Information-02	Static Analysis: Error unhandled	Improvement	Informational	Acknowledged
Information-03	Static Analysis: Use of week random number generator	Improvement	Informational	Acknowledged
Information-04	Transaction and block processing	Passed	Informational	Acknowledged
Information-05	RPC and Websocket interfaces	Passed	Informational	Acknowledged
Information-06	Optimizations of default parameter about memory	Improvement	Informational	Acknowledged
Information-07	Advices on a ethereum-like chain with POA consensus	Improvement	Informational	Acknowledged

Information-01 | Package Vulnerability Analysis

Category	Severity	Location	Status
Passed	Informational	go-brise/go.mod	Acknowledged

Description:

There was not found any issue in 67 packages scanned against any suspicious.

Recommendation:

Since each dependency can introduce new attack vectors and requires time and effort to monitor for security vulnerabilities, the team recommends that keep eyes on the security issues in communities of third-party packages, and upgrade packages immediately.

Information-02 | Static Analysis: Error unhandled

Category	Severity	Location	Status
Improvement	Informational	go-brise/core/state/statedb.go: #128	Acknowledged

Description:

In Go, error handling is important. The language's design and conventions encourage you to explicitly check for errors where they occur (as distinct from the convention in other languages of throwing exceptions and sometimes catching them). There are various number of unhandled errors in directory which are low level severity, not only mentioned in that Location file. The same situation also happens in Go-Ethereum, since brise-chain is forked from that, we still recommend that any error returned should be managed to reduce the risk of fail, especially in some important modules, such as StateDB, Database, Trie, etc...

Recommendation:

If not disturb current log system, use Logger (go-brise/log/logger.go) instance to display more error messages of essential operations as possible as you can.

Information-03 | Static Analysis: Use of week random number generator

Category	Severity	Location	Status
Improvement	Informational	go-brise/eth/protocols/eth/peer.go: #328	Acknowledged

Description:

There are various number of “use of week random number generator” issue in go-brise directory, not only mentioned in that Location file, which are low level severity issues but if any error returned should be managed to reduce the risk of fail.

Recommendation:

It is recommended to use crypto/rand instead of math/rand.

Information-04 | Transaction and block processing

Category	Severity	Location	Status
Passed	Informational	go-brise/...	Acknowledged

Description:

On a running private node, the team reviewed the transaction and block processing parts. And we made temporary pressure test in four regular situations simulating user's operation: transfer, contract storages written in db (SSTORE Opcode), contract storages read from db (SLOAD Opcode) and arithmetic operator executed in contract. The metrics about CPU processing, opmemory allocation, I/O operations and goroutine leaks showed steady

Recommendation:

Even though the benchtest in manual passed successfully, the team recommends that use a high-performance server to running a validator node, in case of some contract executing some opcodes consuming large gas cost, which will have impact on physical hardware.

Information-05 | RPC and Websocket interfaces

Category	Severity	Location	Status
Passed	Informational	go-brise/rpc/...	Acknowledged

Description:

In order for a software application to interact with the blockchain - either by reading blockchain data or sending transactions to the network - it must connect to an node. For this purpose, every client implements a JSON-RPC specification, so there are a uniform set of methods that applications can rely on regardless of the specific node or client implementation. On a running private node, the team requested common rpc methods listed on https://ethereum.org/en/developers/docs/apis/json-rpc , by different rpc clients, such as go-ethereum, web3j and web3.js. All requests could be handled successfully.

Recommendation:

In case of heavy load of requests, it is necessary to use distributed servers to handle requests. And it is better to corporate with the third-party service, like Ankr.

Information-06 | Optimizations of default parameter about memory

Category	Severity	Location	Status
Improvement	Informational	go-brise/core/blockchain.go: #93	Acknowledged

Description:

Based on previous experience, a full node without archive mode will flush matured singleton nodes to disk when exceeding the memory allowance. So it will disturb developers calling requests in recent blocks when old states are flushed already.

Recommendation:

It is not a must option. If considered, set a larger number, like 1024, to keep old states kept (in memory OR on disk) before pruning.

Information-07 | Advices on a ethereum-like chain with POA consensus

Category	Severity	Location	Status
Improvement	Informational		Acknowledged

Description:

Different with POW or POS, validator nodes running with POA consensus have to take more responsibilities on keeping the data of whole chain by malicious attacks.

Recommendation:

It is better to use high security solutions to protect the private keys of validator nodes from leaking.

Disclaimer

1 Only the audit types mentioned in the final report published are the subject of this audit report. This audit does not cover any other undiscovered security flaws, and we disclaim all liability for them. ii. A report on an audit may only be based on an attack or vulnerability that existed or had already taken place at the time the report was issued.br/>

2 We are not liable for any new attacks or vulnerabilities that may be launched or arise in the future, and we are unable to predict their likely effects on the security posture of our projects.

3 Prior to the publication of the audit report, the Project Party gave us with certain papers and materials, including but not limited to contract codes, on which we based our security audit analysis and other audit report components. Such documents and materials shall not be false, inaccurate, uninformative, changed, deleted, or concealed, and if the Project Party's documents and materials are false, inaccurate, uninformative, changed, deleted, or concealed, or if the Project Party's documents and materials are untrue, inaccurate, uninformative, altered, deleted, or concealed, or if the Project Party's documents and materials. We shall not be responsible for any loss or negative consequences resulting from any discrepancy between the reflected and actual conditions if the records and information provided by the Project Party are false, inaccurate, uninformative, altered, deleted, or concealed, or if changes are made to such documents and information after the audit report is issued.

4 The Project Parties are aware that our audit report depends on currently available technology and is based on information and documents supplied by the Project Parties. However, there is a chance that our audit report might not fully identify all hazards due to the technical limits of any business. The project development team and any other interested parties are urged to carry out further testing and audits of the project by our audit team.

5 The project owner guarantees that the project is legitimate, compliant, and does not break any laws in the country where the audit or testing is being performed. The audit report is just for the project owner's reference; it should not be used for investment, tax, legal, regulatory, or advisory reasons of any sort, and we will not be held responsible for the contents,

method of acquisition, use, or any services or resources included in the audit report. Without our prior written consent, the Project Party shall not make any references to, quotations from, displays of, or transmissions of the Audit Report, in whole or in part, to any third party. Any damage or liabilities caused by that location is the responsibility of the project party. We disclaim all liability for any reliance or use of the audit report, regardless of its intended use.

6 The compiler of the contract or any other topics outside of the Smart Contract's programming language are not covered in this audit report. The project party is solely responsible for the risk and liability of the audited Smart Contract resulting from references to off-chain data or resources.

About BiPOLE Labs

Through the provision of market-leading smart contract auditing services, BiPOLE Labs, a leading blockchain security company, aims to conduct security and vulnerability research on current blockchain ecosystems. Please contact us for more information at (www.bipole.org) or Email (team@bipole.org)