Blockchain Security Audit Report

Table Of Contents

1 Executive Summary

On 2025.09.03, the SlowMist security team received the community's security audit application for Jovay, developed the audit plan according to the agreement of both parties and the characteristics of the project, and finally issued the security audit report.

The SlowMist security team adopts the strategy of "black, grey box lead, white box assists" to conduct a complete security test on the project in the way closest to the real attack.

The participants of this security audit are shown as follows:

Company	Auditor	Certification
SLOWMIST LIMITED	Jianhai Zhang	CISSP
SLOWMIST LIMITED	Changsheng Feng	CISSP

The test method information:

Test method	Description
Black box testing	Conduct security tests from an attacker's perspective externally.
Grey box testing	Conduct security testing on code modules through the scripting tool, observing the internal running status, mining weaknesses.
White box testing	Based on the open source code, non-open source code, to detect whether there are vulnerabilities in programs such as nodes, SDK, etc.

The vulnerability severity level information:

Level	Description
Critical	Critical severity vulnerabilities will have a significant impact on the security of the DeFi project, and it is strongly recommended to fix the critical vulnerabilities.
High	High severity vulnerabilities will affect the normal operation of the DeFi project. It is strongly recommended to fix high-risk vulnerabilities.
Medium	Medium severity vulnerability will affect the operation of the DeFi project. It is recommended to fix medium-risk vulnerabilities.

Level	Description
Low	Low severity vulnerabilities may affect the operation of the DeFi project in certain scenarios. It is suggested that the project party should evaluate and consider whether these vulnerabilities need to be fixed.
Informational	Features that are consistent with the design intent but important for users to be aware of.
Suggestion	There are better practices for coding or architecture.

2 Audit Methodology

2.1 Compliance

Ensure that the implementation of the blockchain is fully compliant with the "Guideline on Supervision of Stablecoin Issuers" issued by the HKMA.

2.2 Threat Modeling

Threat modeling is a core methodology for blockchain security audits. It provides guidance for subsequent security testing and the design of protective measures by systematically analyzing potential attack vectors and security threats. In the threat modeling of stablecoin blockchain, we adopt the classic CIA Triad and STRIDE model as analytical frameworks, with appropriate adjustments based on the unique characteristics of fiat-referenced stablecoins blockchain.

CIA Triad Analysis

- Confidentiality: Protect sensitive operational parameters and administrative functions from unauthorized access.
- Integrity: Ensure that critical data such as token supply and account balances cannot be tampered with.
- Availability: Guarantee the continuous operation of the smart contract under both normal and emergency conditions.

STRIDE Threat Model

- Spoofing: Prevent unauthorized users from impersonating legitimate roles.
- Tampering: Protect the integrity of smart contract code and state data.
- Repudiation: Ensure the non-repudiation of operations through event logs.
- Information Disclosure: Prevent unauthorized access to sensitive information/functions.
- Denial of Service: Ensure the smart contract's ability to withstand DoS attacks.
- Elevation of Privilege: Prevent privilege escalation attacks

2.3 Checklist

The security audit process of SlowMist security team for the chain includes two steps:

Chain codes are scanned/tested for commonly known and more specific vulnerabilities using automated analysis tools.

Manual audit of the codes for security issues. The codes are manually analyzed to look for any potential problems.

Following is the list of commonly known vulnerabilities that was considered during the audit of the chain:

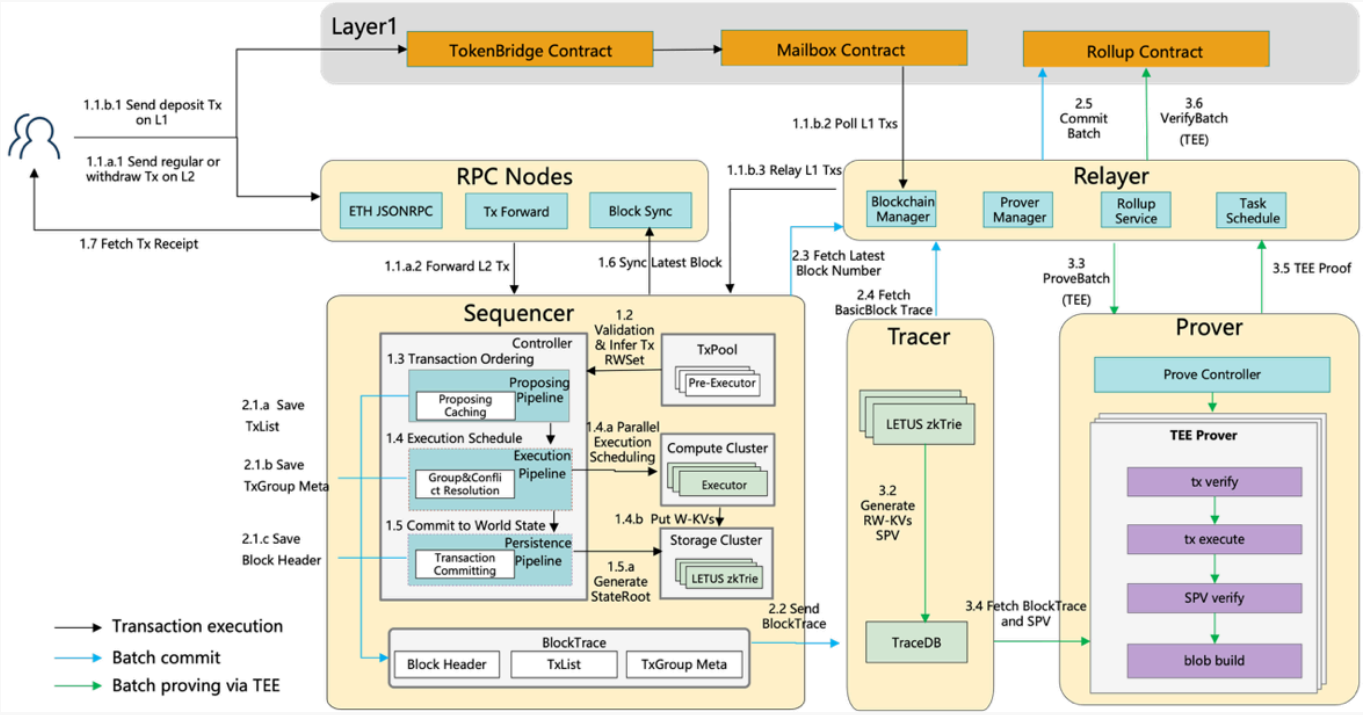
NO.	Audit Items	Result
1	Distributed Ledger Technology Evaluation	Passed
2	Security Infrastructure	Passed
3	Network Scale	Passed
4	Consensus Algorithm	Passed
5	Code Security	Some Risks
6	Historical Audit Records	Passed
7	Audit Institution Evaluation	Passed

3 Project Overview

3.1 Project Introduction

Jovay is a high-performance, user-friendly Layer 2 scaling solution that aims to break through the scalability bottleneck of blockchain through innovative technologies while maintaining compatibility with the Ethereum ecosystem.

Architecture of Jovay:



3.2 Mainnet Assessment

Distributed Ledger Technology Evaluation

(1) Robustness Assessment

Jovay is designed as a high-performance Ethereum Layer 2 network with a modular architecture that decouples key components such as execution, proof service, data availability, and settlement. This enables independent scaling and optimization, enhancing robustness against scalability constraints. The system employs a fully pipelined, parallel execution engine that breaks down transaction workflows into discrete units for concurrent processing at transaction, block, and batch levels, achieving cluster-scale throughput (current single-node testnet: 6,000–7,000 TPS; expected multi-process: 20,000–30,000 TPS; target: 100,000 TPS within the next year) with reduced latency. Robustness is further supported by dynamic load balancing, compute-storage decoupling, asynchronous multi-stage pipelining for inter-block parallelism, and adaptive DAG-based scheduling for intra-block concurrency. These features ensure semantic consistency with serial execution while handling high loads without bottlenecks. The network incorporates

a phased validity proof mechanism (initially TEE for fast finality, transitioning to ZKP for stronger guarantees), backed by on-chain attestation and third-party audits, to maintain integrity under stress. In congestion scenarios, L2 gas pricing adjusts dynamically based on L1 blob gas prices (via EIP-4844), requiring higher gas limits for transactions to succeed, preventing overload.

(2) Reliability Assessment

Jovay's testnet has been running stably with no reported downtime or synchronization issues. The mainnet is scheduled for launch on September 25, 2025, or earlier. Reliability is bolstered by a "pragmatism-first" approach, aligning technical roadmaps with on-chain asset complexity. The modular design allows granular upgrades (e.g., hot-swapping execution engines) without system disruption, and the network supports EVM compatibility for seamless developer adoption. Emergency response relies on Ant Group's security management system, including predefined plans for intrusions (e.g., keys, contracts, cloud infrastructure). Block rollback is not supported in normal operations but may occur in extreme cases for batch-level recovery. Finality is achieved through L1 (Ethereum) submission and verification, ensuring consistent state reconciliation. The Deterministic Virtual Machine (DTVM), an open-sourced tiered lazy JIT compilation engine, enhances smart contract execution reliability and is being optimized for EVM bytecode.

Security Infrastructure

(1) Integrity Mechanism (Integrity)

Jovay employs robust integrity mechanisms to ensure data consistency and verifiability. It uses a hash chain structure with zkTrie (a zero-knowledge-friendly Merkle Tree variant, replacing Ethereum's MPT) for account and storage state management, providing verifiable consistency and completeness. Digital signatures are handled via ECDSA (secp256k1) for transaction verification, maintaining EVM compatibility. Hashing algorithms include Keccak-256 for transaction, block, and receipt hashes (EVM-compatible), and Poseidon for state trees to optimize proof performance. Finality is enforced through Rollup submissions to Ethereum L1, where validity proofs (via TEE initially, transitioning to ZKP) confirm L2 state transitions. On-chain attestation protocols, including node registration, initialization verification, enrollment, and runtime checks, form a structured trust chain. Critical components like system contracts and RPC interfaces have undergone audits by Antgroup Skyward Lab, with ongoing cycles for subsystems (sequencer, prover, relayer) and a full-chain evaluation planned before mainnet.

(2) Protection Capabilities

Protection is multi-layered, including DDoS defenses via node-level rate limiting, load balancing on RPC endpoints, and transaction filters (e.g., minimum gas price, chainId/nonce/fee checks) to mitigate abuse. Contracts operate in an EVM-based sandbox with gas limits to prevent infinite loops or denial-of-service attacks, storage isolation, permission boundaries, and call stack/memory separation adhering to Ethereum semantics. Emergency response leverages Ant Group's frameworks for rapid containment and recovery from intrusions. The hybrid TEE-ZK proof system provides verifiable computation, with TEE enabling high-throughput, low-latency proofs and ZKP planned for trust-minimized security. Anti-replay protections include global nonce constraints and Mailbox contract verifications to prevent fake deposits. MEV risks are addressed through centralized sequencing (with no current punishments), and the network supports proxy-based contract upgrades. Privacy features include planned ZKP support (zk-SNARK/STARK), though quantum-resistant algorithms (e.g., BLS) are not yet integrated to prioritize EVM compatibility. The testnet has participated in crowd-sourced testing via Ant Security Response Center, and a custom ZK proof system is under development for release by year-end.

Network Scale

(1) Geographic Distribution

Nodes are deployed across multiple availability zones on Alibaba Cloud, with cross-regional backups to ensure redundancy and fault tolerance. Specific geographic details (e.g., North America, Europe, Asia) are not fully disclosed, but the setup emphasizes distributed resilience within cloud infrastructure.

(2) Hosting Provider Distribution

The network primarily relies on Alibaba Cloud for hosting, with nodes distributed across its availability zones. No details on other providers (e.g., AWS, Hetzner, OVH) or self-built servers are provided, indicating a concentrated but redundantly configured hosting strategy.

(3) Centralization Degree

Jovay exhibits a degree of centralization in its current sequencer design, which uses a centralized sequencing and block production method without a distributed consensus algorithm, punishments, or election mechanisms. Operational permissions, such as multi-signature wallets and KMS for system contract upgrades, Rollup contracts, and L2 coinbase withdrawals, are controlled by the operational team. Only the operational team can run Relayers

currently. Mainnet node counts (execution and consensus layers) and total distribution are pending, as the mainnet is not yet live (testnet is operational). This setup inherits Ethereum's security for finality but relies on centralized components for L2 execution, with plans for decentralization through clustered node expansion and ZKP transitions.

Consensus Algorithm

(1) Consensus Algorithm Features

Jovay does not employ a traditional distributed consensus algorithm like PoW, PoS, DPoS, or BFT. Instead, it uses a centralized sequencer for transaction ordering and block production, batching transactions off-chain and submitting them to Ethereum L1 for data availability and finality. This inherits Ethereum's security while enabling high performance. Key features include no punishment or election mechanisms, with security focused on validity proofs rather than consensus. The architecture supports parallel execution and modular scaling, but consensus is effectively offloaded to L1.

(2) Consensus Implementation Security

Security is ensured through a heterogeneous TEE-ZK hybrid proof system rather than consensus-specific mechanisms. TEE provides initial fast finality and scalable proofs, with on-chain verification contracts checking attestation quotes, measurements (mrsigner/mrenclave), and materials from Intel PCS. A phased transition to ZKP aims to reduce trust assumptions, supported by research on zk-SNARK security under the Generic Group Model (accepted at Asiacrypt 2024). Audits cover critical components, and relayers validate L1 finality before L2 submissions. Without distributed consensus, risks like slashing are absent, but centralization introduces single points of failure, mitigated by emergency plans and no routine rollback support.

Code Security

The SlowMist security team conducted a black box and gray box audit on the key modules of Jovay, using automated tools and manual reviews, focusing mainly on the following areas:

- Static code analysis
- Encrypted and signature security audit
- Account and transaction security audit
- RPC Security audit
- P2P Security audit

- Consensus Security audit

Black-box / Gray-box audit entries include:

- Insufficient entropy of private key random numbers
- Precision loss in private key seed conversion
- Theoretical reliability assessment of symmetric encryption algorithms
- Theoretical reliability assessment of hash algorithms
- Theoretical reliability assessment of signature algorithms
- Supply chain security of symmetric crypto algorithm reference libraries
- Keystore encryption strength detection
- Hash algorithm length extension attack
- secp256k1 k-value randomness security
- secp256k1 r-value reuse private key extraction attack
- ECC signature malleability attack
- ed25519 private key extraction attack
- Schnorr private key extraction attack
- ECC twist attack
- Merkle-tree Malleability attack (CVE-2012-2459)
- Native characteristic false recharge
- Contract call-based false recharge
- Native chain transaction replay attack
- Cross-chain transaction replay attack
- Transaction lock attack
- Transaction fees not dynamically adjusted
- RPC remote key theft attack
- RPC open cross-domain vulnerability to local phishing attacks
- RPC malformed packet denial-of-service attack
- RPC database injection

- RPC communication encryption
- Excessive administrator privileges
- Non-privacy/Non-dark Coin Audit
- Insufficient number of core nodes
- Excessive concentration of core node physical locations
- P2P node maximum connection limit
- P2P node independent IP connection limit
- P2P inbound/outbound connection limit
- P2P shapeshift attack
- P2P communication encryption
- Consensus algorithm potential risk assessment
- Block time offset attack
- PoS/BFT final confirmation conditions
- PoS/BFT double-signing penalty
- PoS/BFT block refusal penalty

481 issues were found through source code scanning tools such as Qodana/cppcheck/semgrep during the audit process, and have been submitted to the project for fixing.

Different levels of safety risks were identified through manual assessment, details please refer to the findings section of this report.

Historical Audit Records

Jovay has undergone several security audits and testing processes to ensure robustness, particularly for critical components. Critical system contracts, RPC interfaces, Rollup contracts, and TEE Verifier have been audited by internal security departments and professional security firms. Specifically, audits were conducted by Antgroup Skyward Lab, with reports covering key subsystems.

The testnet has participated in crowd-sourced vulnerability testing through the Ant Security Response Center, a platform for bug bounties and security crowdsourcing. Details are available at: https://security.alipay.com/ and referenced in a public announcement: https://x.com/Jovay_zh/status/1960953828670353874.

Audit Institution Evaluation

The primary audit institution involved is Antgroup Skyward Lab (part of Ant Group), which has a strong reputation in blockchain and cybersecurity, particularly within the Asian market. Ant Group, the parent company, is renowned for its expertise in financial technology security, powering platforms like Alipay, and has a track record of conducting high-profile audits for blockchain projects. Skyward Lab specializes in security research and auditing for distributed systems, with contributions to open-source security tools (e.g., https://github.com/antgroup-skyward).

Additionally, the Ant Security Response Center facilitates crowd-sourced testing, leveraging a community-driven approach similar to bug bounty programs, which enhances detection of vulnerabilities through diverse tester input.

4 Findings

4.1 Coverage

Audit scope:

Jovay Mainnet/Testnet

Testing version:

https://github.com/jovaynetwork/jovay

`commit: e822dfaffd958d9f94ad65cc36c4d2a7da2964af`

https://github.com/jovaynetwork/jovay-net

`commit: fb53d11a453339811b5c984e3e78495d6adedeee`

https://github.com/jovaynetwork/jovay-sequencer

`commit: a2e01279e3e01a7cef4cf98a50e62db4b3347583`

https://github.com/jovaynetwork/jovay-consensus

`commit: 12b1d39bbcc1f062a55e1810bdbec95ad318fa47`

https://github.com/jovaynetwork/jovay-relayer

`commit: 25d46a24502088f98eb6f156c26681e0d1f71fe7`

https://github.com/jovaynetwork/jovay-storage

`commit: 4101d870083e00cf111ca7cac32ef01a189661ef`

4.2 Vulnerability Information

The following is the status of the vulnerabilities found in this audit:

NO	Title	Category	Level	Status
N1	Malformed packets can cause node crashes	Code Security	Medium	Fixed
N2	Lack of trustlessness	Code Security	Low	Acknowledged
N3	False top-up risks of EVM	Code Security	Information	Acknowledged
N4	Default RPC Lack of HTTPS Encryption	Code Security	Information	Acknowledged
N5	ECDSA quantum computing risks	Code Security	Information	Acknowledged
N6	EVM smart contract attack risks	Code Security	Information	Acknowledged
N7	Sequencer MEV attack risks	Code Security	Information	Acknowledged
N8	Improving code quality	Code Security	Suggestion	Acknowledged

4.3 Vulnerability Summary

[N1] [Medium] Malformed packets can cause node crashes

Category: Code Security

Content

Constructing malformed data for testing node RPCs:

```
data = '{"' + '}' * 0x101000 + '":' + '{"x":' * 0x10000 + '"}'
print(post(posturl, data))
```

Return:

```
http.client.RemoteDisconnected: Remote end closed connection without response
```

Solution

Note the limitation of the JSON request data package length and recursion depth.

Status

Fixed

[N2] [Low] Lack of trustlessness

Category: Code Security

Content

Blockchain systems aim to create a "trustless" environment through decentralization, encryption, and consensus mechanisms, allowing users to rely on no single intermediary. However, Jovay has some centralized permissions, requiring users to have a certain level of trust in it.

(1). Risk of centralized design in the sequencer

Jovay's sequencer subsystem is responsible for transaction processing, batching, block generation, and returning execution receipts. Currently, the sequencer uses a centralized ordering mechanism for block production, without a distributed consensus algorithm (such as PoS or BFT), nor any penalty or election mechanisms. This means that block generation and transaction ordering heavily rely on a single or a few operating nodes, which may lead to centralized control.

(2). Risk of special permissions and contract upgrade by the operator

Jovay has built-in special permissions for the operator on-chain, including the ability to manage contract upgrades for system contracts and Rollup contracts, as well as the withdrawal permissions for L2 coinbase accounts via multi-signature wallets and KMS (Key Management System). These permissions allow the operator to unilaterally modify core logic or withdraw funds.

(3). Risk of exclusive operation of the relayer

The relayer is responsible for cross-layer data anchoring and asset transfers (such as bridging from L1 to L2). Currently, only the operator can run the relayer. This makes cross-chain operations (such as deposits and

withdrawals) dependent on components controlled by the operator.

(4). Risk of centralized control in block rollback and emergency response

Jovay does not support regular block rollbacks, but in extreme disaster scenarios, it may perform consistency recovery, including the rollback of a single batch. This depends on the operator's emergency plan.

(5). Overall centralized risk

Nodes are deployed in Alibaba Cloud multi-availability zones with backups. However, these still rely on centralized infrastructure (such as Alibaba Cloud) and the operator.

Solution

N/A

Status

Acknowledged

[N3] [Information] False top-up risks of EVM

Category: Code Security

Content

The false top-up vulnerability refers to attackers forging transaction records or event logs to deceive target systems (such as exchanges, DeFi protocols, or cross-chain bridges) into believing that funds have been successfully deposited. This allows attackers to withdraw real assets without any actual fund transfer.

One of the attack methods is to use revert() to interrupt the transaction in an inline transfer call, but the status of the entire transaction remains successful. If the recharge target does not strictly verify the event of the recharge transaction, it is very likely to lead to the success of a fake recharge attack.

Reference: https://mp.weixin.qq.com/s/3cMbE6p_4qCdVLa4FNA5-A

Solution

Target systems (such as exchanges, DeFi protocols, or cross-chain bridges) need to pay attention to the risk of fake top-up and strictly check deposit transactions.

Status

Acknowledged

[N4] [Information] Default RPC Lack of HTTPS Encryption

Category: Code Security

Content

The JSON-RPC interface of Jovay node is default configured as HTTP (non-encrypted), not using HTTPS. This means all data sent through RPC (such as querying balance, submitting transactions or calling smart contracts) is transmitted in plain text, making it easy to be sniffed or intercepted by the network.

Solution

Use self-signed or CA certificate to protect RPC interfaces; for example, adding SSL via nginx reverse proxy.

Status

Acknowledged

[N5] [Information] ECDSA quantum computing risks

Category: Code Security

Content

Quantum computing poses a significant threat to Ethereum's ECDSA (Elliptic Curve Digital Signature Algorithm), which underpins account security, transaction signing, and wallet integrity. Advanced quantum algorithms like Shor's could efficiently factor large numbers, potentially cracking ECDSA private keys and enabling unauthorized access to funds or smart contracts. As per the referenced article from Safeheron, this vulnerability could emerge with scalable quantum computers, compromising the network's cryptographic foundation. While not an immediate risk, it undermines long-term security.

Reference: https://mp.weixin.qq.com/s/YxdxdVuFxtT1SaFR1cjkvg

Solution

Mitigation involves transitioning to post-quantum cryptography (e.g., BLS signatures or quantum-resistant algorithms) via protocol upgrades like those planned in Ethereum's roadmap.

Status

Acknowledged

[N6] [Information] EVM smart contract attack risks

Category: Code Security

Content

Evm smart contracts are susceptible to various attack vectors due to immutable code deployment and complex interactions, potentially leading to fund theft, denial-of-service, or unauthorized access. Common exploits include reentrancy attacks (e.g., The DAO hack in 2016, draining millions in ETH), integer overflows/underflows, access control failures, and oracle manipulation in DeFi protocols. These vulnerabilities arise from coding errors, insufficient testing, or flawed logic, often resulting in significant financial losses (e.g., over $3 billion stolen since 2016).

Reference: https://hacked.slowmist.io/statistics/?c=ETH&d=all

Solution

Mitigation requires rigorous audits, secure coding patterns (e.g., Checks-Effects-Interactions), formal verification, and upgradable proxies to patch issues post-deployment. Developers should follow best practices from the Ethereum Security Checklist to enhance resilience.

Status

Acknowledged

[N7] [Information] Sequencer MEV attack risks

Category: Code Security

Content

Maximal Extractable Value (MEV) attack risk arises from Sequencer (validators or previously miners) exploiting their ability to reorder, insert, or censor transactions within blocks to extract additional value, often at users' expense. Common exploits include front-running (e.g., bidding higher gas to preempt trades), sandwich attacks (surrounding a user's transaction to profit from price slippage), and arbitrage manipulation in DeFi protocols. This stems from the transparent mempool and block production incentives, potentially leading to unfair advantages and economic inefficiencies. Post-Merge, MEV has persisted via tools like MEV-Boost, with billions in value extracted annually.

Solution

Mitigation strategies include Proposer-Builder Separation (PBS) for fairer auctions, encrypted mempools, and user-side protections like private RPCs or Flashbots to minimize exposure.

Status

Acknowledged

[N8] [Suggestion] Improving code quality

Category: Code Security

Content

481 issues were found through source code scanning tools such as Qodana/cppcheck/semgrep during the audit process, and have been submitted to the dev team for fixing.

Solution

Optimize code quality to enhance security.

Status

Acknowledged

5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002509230001	SlowMist Security Team	2025.09.03 - 2025.09.23	Low Risk

Summary conclusion: The SlowMist security team use a manual and SlowMist team's analysis tool to audit the project, during the audit work we found 1 medium risk, 1 low risk, 1 suggestion vulnerabilities. All the findings were fixed. The code was not deployed to the mainnet.

6 Statement

SlowMist issues this report with reference to the facts that have occurred or existed before the issuance of this report, and only assumes corresponding responsibility based on these.

For the facts that occurred or existed after the issuance, SlowMist is not able to judge the security status of this project, and is not responsible for them. The security audit analysis and other contents of this report are based on the documents and materials provided to SlowMist by the information provider till the date of the insurance report (referred to as "provided information"). SlowMist assumes: The information provided is not missing, tampered with, deleted or concealed. If the information provided is missing, tampered with, deleted, concealed, or inconsistent with the actual situation, the SlowMist shall not be liable for any loss or adverse effect resulting therefrom. SlowMist only conducts the agreed security audit on the security situation of the project and issues this report. SlowMist is not responsible for the background and other conditions of the project.

Official Website

www.slowmist.com

E-mail

team@slowmist.com

Twitter

@SlowMist_Team

Github

https://github.com/slowmist