Scallop

Token Contract

Smart Contract Audit Report

August 14, 2021

IMMUNE BYTES

Introduction

1. About Scallop

As the first De-Fi enabled banking platform, Scallop provides a simple and eloquent solution to this problem. The backend technology will instantly process each one of these transactions for the users, converting the cryptocurrency of their choice into fiat, immediately for vendors to receive, even if they don't accept cryptocurrencies themselves for payments. Scallop Pay is the first ever regulated solution for this increasingly painful real world problem.

Scallop will issue its users with a virtual or physical debit card which will enable these transactions over Scallop Pay, which users can use in exactly the same way that they use their current fiat cards. Scallop Pay will support the exchange of both cryptocurrency tokens and liquidity pool (LP) tokens staked in De-Fi protocols, so that wherever your tokens are in the crypto world, you still have easy access to them and the ability to use them instantaneously.

Visit https://www.scallopx.com/ to know more about.

2. About ImmuneBytes

ImmuneBytes is a security start-up to provide professional services in the blockchain space. The team has hands-on experience in conducting smart contract audits, penetration testing, and security consulting. ImmuneBytes's security auditors have worked on various A-league projects and have a great understanding of DeFi projects like AAVE, Compound, 0x Protocol, Uniswap, dydx.

The team has been able to secure 65+ blockchain projects by providing security services on different frameworks. ImmuneBytes team helps start-up with a detailed analysis of the system ensuring security and managing the overall project.

Visit http://immunebytes.com/ to know more about the services.

Documentation Details

The Scallop team has provided the following doc for the purpose of audit:

1. https://scallop-docs.s3.eu-west-1.amazonaws.com/Scallop_Litepaper.pdf

Audit Process & Methodology

ImmuneBytes team has performed thorough testing of the project starting with analyzing the code design patterns in which we reviewed the smart contract architecture to ensure it is structured and safe use of third-party smart contracts and libraries.

Our team then performed a formal line-by-line inspection of the Smart Contract in order to find any potential issues like Signature Replay Attacks, Unchecked External Calls, External Contract Referencing, Variable Shadowing, Race conditions, Transaction-ordering dependence, timestamp dependence, DoS attacks, and others.

In the Unit testing phase, we run unit tests written by the developer in order to verify the functions work as intended. In Automated Testing, we tested the Smart Contract with our in-house developed tools to identify vulnerabilities and security flaws.

The code was audited by a team of independent auditors which includes -

1. Testing the functionality of the Smart Contract to determine proper logic has been followed throughout.
2. Analyzing the complexity of the code by thorough, manual review of the code, line-by-line.
3. Deploying the code on testnet using multiple clients to run live tests.
4. Analyzing failure preparations to check how the Smart Contract performs in case of bugs and vulnerabilities.
5. Checking whether all the libraries used in the code are on the latest version.
6. Analyzing the security of the on-chain data.

Audit Details

- Project Name: Scallop
- Contracts Name: ScallopToken
- Languages: Solidity(Smart contract)
- Github commit/Smart Contract Address for audit: e1aa5529af4aa8598de42468462d38da19788c1f
- Platforms and Tools: Remix IDE, Truffle, Truffle Team, Ganache, Solhint, VScode, Contract Library, Slither, SmartCheck

Audit Goals

The focus of the audit was to verify that the smart contract system is secure, resilient, and working according to its specifications. The audit activities can be grouped into the following three categories:

1. Security: Identifying security-related issues within each contract and within the system of contracts.
2. Sound Architecture: Evaluation of the architecture of this system through the lens of established smart contract best practices and general software best practices.
3. Code Correctness and Quality: A full review of the contract source code. The primary areas of focus include:
    a. Correctness
    b. Readability
    c. Sections of code with high complexity
    d. Quantity and quality of test coverage

Security Level References

Every issue in this report was assigned a severity level from the following:

High severity issues will bring problems and should be fixed.

Medium severity issues could potentially bring problems and should eventually be fixed.

Low severity issues are minor details and warnings that can remain unfixed but would be better fixed at some point in the future.

Issues	High	Medium	Low
Open	-	-	-
Closed	-	-	-

High Severity Issues

No issues found.

Medium Severity Issues

No issues found.

Low Severity Issues

No issues found.

Recommendations

1. **NatSpec Annotations must be included**

   Description:
   The smart contracts do not include the NatSpec annotations adequately.

   Recommendation:
   Cover by NatSpec all Contract methods.

Automated Audit Result

```
Compiled with solc
Number of lines: 1486 (+ 0 in dependencies, + 0 in tests)
Number of assembly lines: 0
Number of contracts: 17 (+ 0 in dependencies, + 0 tests)

Number of optimization issues: 17
Number of informational issues: 17
Number of low issues: 2
Number of medium issues: 4
Number of high issues: 0

ERCs: ERC165, ERC20

+---------------+-------------+--------------+--------------------+--------------+----------+
|      Name     | # functions |     ERCS     |     ERC20 info     | Complex code | Features |
+---------------+-------------+--------------+--------------------+--------------+----------+
|    Strings    |      4      |              |                    |     Yes      |          |
| EnumerableSet |      20     |              |                    |      No      |          |
|  ScallopToken |      70     | ERC20,ERC165 |      Pausable      |      No      |          |
|               |             |              |     ∞ Minting      |              |          |
|               |             |              | Approve Race Cond. |              |          |
|               |             |              |                    |              |          |
+---------------+-------------+--------------+--------------------+--------------+----------+
INFO:Slither:flat.sol analyzed (17 contracts)
```

This audit does not provide a security or correctness guarantee of the audited smart contract. Securing smart contracts is a multistep process, therefore running a bug bounty program as a complement to this audit is strongly recommended.

Concluding Remarks

While conducting the audits of the Scallop smart contract, it was observed that the contract does not contain any issues.

- ***The Scallop team has followed best practices for Smart Contract development and implemented standard ERC20.***

Disclaimer

ImmuneBytes's audit does not provide a security or correctness guarantee of the audited smart contract. Securing smart contracts is a multistep process, therefore running a bug bounty program as a complement to this audit is strongly recommended.

Our team does not endorse the Scallop platform or its product nor this audit is investment advice.
Notes:

- Please make sure contracts deployed on the mainnet are the ones audited.
- Check for the code refactor by the team on critical issues.

ImmuneBytes