Satoshi Protocol

Audit Report

Mon Mar 25 2024

✉ contact@bitslab.xyz

https://twitter.com/scalebit_

ScaleBit

Satoshi Protocol Audit Report

1 Executive Summary

1.1 Project Information

Description	The Satoshi Protocol aims to provide a cornerstone for DeFi and make BTC truly spendable in daily usage by offering a CDP-style stablecoin.
Type	DeFi
Auditors	ScaleBit
Timeline	Sun Feb 25 2024 - Mon Mar 25 2024
Languages	Solidity
Platform	BEVM
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/Satoshi-Protocol/satoshi-core
Commits	17b598ea88f0057078dfb0e477785842ca1d6a2b 58831110505073f3053dbbfa6b8bf8c621b9320e

1.2 Files in Scope

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

ID	File	SHA-1 Hash
CIS	src/OSHI/CommunityIssuance.sol	3ee03f5405d20ffc7ce1bd3541bb8ba68f3a7424
OSHIT	src/OSHI/OSHIToken.sol	1d0853b3281b1f71d23e346e7fa727e1bdca4f1e
RES	src/OSHI/Reserve.sol	324e14c950e20215c18a8e413290fd8a67dfd598
IVE1	src/OSHI/InvestorVesting.sol	5bd0f18cb6c56e80fa2f5f7f1b35b8b2d073824c
VMA	src/OSHI/VestingManager.sol	a55b0978dbb21e4e883cbf877b4623323ff5a389
RMA1	src/OSHI/RewardManager.sol	a9f7249744a2b4deef361fc752eb364397e69d06
VES	src/OSHI/Vesting.sol	c7d7ea1ad93e6f4bd4e5e3e6e93d74462391fdaa
CIS	src/OSHI/CommunityIssuance.sol	58890bf91d934293324db6473a6e2b7d32512371
RES	src/OSHI/Reserve.sol	ccca89af68f16fe6f8fb903b59f4f9960560fdb8
OSHIT	src/OSHI/OSHIToken.sol	4e57420eacd6306ea510517d5b397eb284470bbb
RMA1	src/OSHI/RewardManager.sol	ca02b51815b014a23251061f9050225aa09573ff

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	6	5	1
Informational	0	0	0
Minor	4	4	0
Medium	2	1	1
Major	0	0	0
Critical	0	0	0

1.4 ScaleBit Audit Breakdown

ScaleBit 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
- Number of rounding errors
- Unchecked External Call
- Unchecked CALL Return Values
- Functionality Checks
- Reentrancy
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic issues
- Gas usage
- Fallback function usage
- tx.origin authentication
- Replay attacks
- Coding style issues

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) 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 Satoshi Protocol to identify any potential issues and vulnerabilities in the source code of the Satoshi Protocol 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 identified 6 issues of varying severity, listed below.

ID	Title	Severity	Status
OSH-1	Signature Malleability	Medium	Fixed
RES-1	Immutable Parameters	Minor	Fixed
RES-2	Unused Variable	Minor	Fixed
RMA-1	Same Collateral Token May Causes Error	Medium	Acknowledged
RMA-2	Use Memory Variables Instead of Storage Variables	Minor	Fixed
RMA-3	The Length of The `registerTroveManager` Can be Too Long	Minor	Fixed

3 Participant Process

Here are the relevant actors with their respective abilities within the Satoshi Protocol Smart Contract :

Admin

- `admin` can create a new instance through `deployNewInstance` .
- `admin` can set the reward rate through `setRewardRate` .
- `admin` can set the minNetDebt through `setMinNetDebt` .
- `admin` can set the collateral through `configureCollateral` .
- `admin` can set the troveManager useable through `enableTroveManager` .
- `admin` can set the priceFeed through `setPriceFeed` .
- `admin` can set the fee receiver through `setFeeReceiver` .
- `admin` can set the guardian through `setGuardian` .
- `admin` can set the reward manager through `setRewardManager` .
- `admin` can change the paused parameter through `setPaused` .
- `admin` can transfer ownership through `commitTransferOwnership` .
- `admin` can revoke transfer ownership through `revokeTransferOwnership` .
- `admin` can start the sunset through `startSunset` .
- `admin` can change the rewardRate through `setRewardRate` .
- `admin` can start sunsetting collateral through `startCollateralSunset` .
- `admin` can set the time when the OSHI claim starts through `setClaimStartTime` .
- `admin` can change the maxTimeThreshold through `updateMaxTimeThreshold` .

User

- `user` can go to flash loan through `flashLoan` .
- `user` can approve others to use tokens through `approve` .
- `user` can send collateral to a trove through `addColl` .
- `user` can withdraw collateral through `withdrawColl` .

- `user` can withdraw debt tokens from a trove through `withdrawDebt`.
- `user` can repay Debt tokens to a Trove through `repayDebt`.

4 Findings

OSH-1 Signature Malleability

Severity: Medium

Status: Fixed

Code Location:

src/OSHI/OSHIToken.sol#86

Descriptions:

The elliptic curve used in Ethereum for signatures is symmetrical, hence for every `v,r,s` there exists another `v,r,s` that returns the same valid result. Therefore two valid signatures exist which allows attackers to compute a valid signature without knowing the signer's private key. `ecrecover()` is vulnerable to signature malleability [1, 2] so it can be dangerous to use it directly. An attacker can compute another corresponding `v,r,s` that will make this check pass due to the symmetrical nature of the elliptic curve.

Suggestion:

It is recommended to use OpenZeppelin’s ECDSA.sol library and reading the comments above ECDSA's `tryRecover()` function provides very useful information on correctly implementing signature checks to prevent signature malleability vulnerabilities. When using OpenZeppelin's ECDSA library, special care must be taken to use version 4.7.3 or greater, since previous versions contained a signature malleability bug.

Resolution:

The client has used the ECDSA library to resolve this issue.

RES-1 Immutable Parameters

Severity: Minor

Status: Fixed

Code Location:

src/OSHI/Reserve.sol#26,27

Descriptions:

The `_totalAmount` and `_eachPeriodReleasedAmount` parameter is defined in the contract. This parameter will only be initialized in the `constructor` and will not be changed subsequently.

Suggestion:

It is recommended to change this parameter to an `immutable` type.

Resolution:

The client added the immutable keyword to solve this issue.

RES-2 Unused Variable

Severity: Minor

Status: Fixed

Code Location:

src/OSHI/Reserve.sol#24

Descriptions:

The `_1_MILLION` variable is only assigned a value when it is defined and has no subsequent use.

Suggestion:

It is recommended to confirm whether this design conforms to the design concept. If not needed, you can remove it.

Resolution:

The client has deleted unused variables.

RMA-1 Same Collateral Token May Causes Error

Severity: Medium

Status: Acknowledged

Code Location:

src/OSHI/RewardManager.sol#273-282

Descriptions:

The `registerTroveManager` function is used to register `TroveManager` information. The `collTokenIndex` array obtains the `F_COLL` and `collForFeeReceiver` data corresponding to `TroveManager` through the `collateralToken` address. However, if two `TroveManagers` use the same `collateral` token when the second `TroveManager` is registered, the first one will be overwritten, information of a `TroveManager`, so that all the data of the first `TroveManager` is reset to zero, including `F_COLL` and `collForFeeReceiver`. This seems unreasonable, and the contract does not restrict different `TroveManagers` from using the same `collateral` token.

Suggestion:

It is recommended to confirm whether this conflicts with the design concept.

Resolution:

The client says there won't be two trove managers with the same collateral.

RMA-2 Use Memory Variables Instead of Storage Variables

Severity: Minor

Status: Fixed

Code Location:

src/OSHI/RewardManager.sol#229,284,331,373

Descriptions:

In some for loops of the contract, comparisons such as `i < collToken.length` are used. The collToken here is a storage variable, which consumes a lot of gas.

Suggestion:

It is recommended to store the storage variable as a memory variable and then read it.

Resolution:

The client has already modified this issue based on our suggestions.

RMA-3 The Length of The `registerTroveManager` Can be Too Long

Severity: Minor

Status: Fixed

Code Location:

src/OSHI/RewardManager.sol#283-291

Descriptions:

The `removeTroveManager` function uses the `delete` keyword to clear the data at a certain position in the `registeredTroveManagers` array. The `delete` keyword will only reset the data to 0 and will not delete the position. The `registerTroveManager` function uses the `push` operator to add array elements, which will cause the array to get longer and longer, and there are many unavailable positions in the array. In some functions, the `registeredTroveManagers` array will be looped, consuming a lot of gas. If the array length is too long, it will even cause the call to fail.

Suggestion:

It is recommended to exchange the data to be removed with the end data and pop it out.

Resolution:

The client uses mapping to solve this issue.

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.