zksync-contract-v2

Audit Report

Thu Feb 22 2024

contact@scalebit.xyz https://twitter.com/scalebit_

ScaleBit

zksync-contract-v2 Audit Report

1 Executive Summary

1.1 Project Information

Description	SpaceFi is the DeFi hub on zk Rollups with DEX+NFT+Spacebase+Launchpad, exploring the Layer2 ecosystem.
Type	Staking
Auditors	ScaleBit
Timeline	Sat Feb 10 2024 - Mon Feb 12 2024
Languages	Solidity
Platform	zkSync Era
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/SpaceFinance/zksync-contract-v2
Commits	cd05b2f28a15a1cd01bad6f27cf38e731b84d0db 3e1491819cbaa8e59a68ca4b059a484732650862

1.2 Files in Scope

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

ID	File	SHA-1 Hash
SFA	StarFarm.sol	57dd6711a631109accb37b3fc3fcd7c49026e3d1
SFL	StarFarmLib.sol	7ae949e361413bbe466fe62021d03c0f0685ea94
SFP	StarFarmPending.sol	bb1684f1b3aff1ebabe7ab3daa186e0c9ed7ec72

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	5	4	1
Informational	2	2	0
Minor	1	1	0
Medium	1	1	0
Major	1	0	1
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 Space Finance to identify any potential issues and vulnerabilities in the source code of the zksync-contract-v2 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 5 issues of varying severity, listed below.

ID	Title	Severity	Status
SFA-1	Potential Reentrancy Risk	Medium	Fixed
SFA-2	Unused Interface	Informational	Fixed
SFL-1	Centralization Risk	Major	Acknowledged
SFL-2	Lack of Events Emit	Minor	Fixed
SFL-3	Lack of Validation for Zero Address	Informational	Fixed

3 Participant Process

Here are the relevant actors with their respective abilities within the zksync-contract-v2 Smart Contract :

Admin

- The Admin can set the `starNode` / `FarmLib` / `Pending` / `StarNFT` / `NFTLogic` / `Airdrop` address in the `MasterChef` contract through `setNode()` / `setFarmLib()` / `setPending()` / `setStarNFT()` / `setNFTLogic()` / `setAirdrop()` .
- The Admin can set the deposit status through `setDeposit()` .
- The Admin can set the pool as `LP` or not through `setIsNotLp()` .
- The Admin can update the `lpAddr` and `lpPerBlock` through `updatelpPerBlock()` / `updatelpTwoPerBlock()` / `updatelpThrPerBlock()` .
- The Admin can set the migrator contract through `setMigrator()` .
- The Admin can migrate lp token to another lp contract through `migrate()` .
- The Admin can update `Multiplier` through `updateMultiplier()` .
- The Admin can add a new lp to the pool through `add()` .
- The Admin can update the given pool's `STAR` allocation point through `set()` .
- The Admin can update the `boost` / `isSingle` / `poolPrice` / `poolLpInfo` through `setBoost()` / `setSingle()` / `setPoolPrice()` / `setLpInfo()` / `setLpTwoInfo()` / `setLpThrInfo()` .
- The Admin can delete the specified pool through `delPool()` .
- The Admin can add the specified preset `starPerBlock` through `addBlockReward()` .
- The Admin can set the specified preset `starPerBlock` through `setBlockReward()` .
- The Admin can delete the specified preset `starPerBlock` through `delBlockReward()` .
- The Admin can set the specified dividend address and proportion through `setAllocationInfo()` .
- The Admin can set the `StarFarm` / `starToken` / `bonus` / `starNode` / `Pending` address in the `FarmLib` contract through `setStarFarm()` / `setToken()` / `setBonus()` / `setNode()` / `setPending()` .

- The Admin can set the `starNode` / `StarFarm` / `FarmLib` / `StarNFT` / `NFTLogic` address in the `FarmPending` contract through `setNode()` / `setStarFarm()` / `setFarmLib()` / `setStarNFT()` / `setNFTLogic()`.
- The Admin can update the `allMultiplier` / `nftBoost` / `Gradient` / `Multipliers` / `XGradient` / `XMultipliers` / `PairUSDC` / `lpPair` / `Bonus` / `EqualSinglePrice` through `setAllMultiplier()` / `setNFTBoost()` / `setGradientAndMultipliers()` / `setXGradientAndMultipliers()` / `setPairUSDC()` / `setPair()` / `setBonus()` / `setEqualSinglePrice()`.
- The Admin can add `Node` user through `regNodeUser()`.
- The Admin can harvest tokens through `harvest()`.
- The Admin can harvest `Lp` tokens through `harvestLp()`.
- The Admin can update the `StartBlock` through `updateStartBlock()`.
- The Admin can update the `lp supply` through `setLpSupply()`.
- The Admin can update the size through `setSize()`.
- The Admin can update the `Blp` through `updateBlp()`.
- The Admin can update the `userBlp` through `setUserBlp()`.
- The Admin can update the `allUser` through `setAllUser()`.

User

- The User can add redeems through `addRedeens()`.
- The User can delete redeems through `delRedeens()`.
- The User can withdraw redeems through `withdrawRedeens()`.
- The User can deposit `LP` tokens to `MasterChef` for `STAR` allocation through `deposit()`.
- The User can withdraw `LP` tokens from `MasterChef` through `withdraw()`.
- The User can stake `Star NFT` to `MasterChef` through `enterStakingNFT()`.
- The User can withdraw `Star NFT` from `STAKING` through `leaveStakingNFT()`.
- The User can withdraw without caring about rewards when `EMERGENCY ONLY` through `emergencyWithdraw()`.

4 Findings

SFA-1 Potential Reentrancy Risk

Severity: Medium

Status: Fixed

Code Location:

StarFarm.sol#283,313,381,418

Descriptions:

The code segment in question is susceptible to reentrancy risk due to its reliance on external calls.

Suggestion:

It is recommended to adopt `ReentrancyGuard` to fix this issue.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

SFA-2 Unused Interface

Severity: Informational

Status: Fixed

Code Location:

StarFarm.sol#66-71

Descriptions:

The aforementioned interface is not used throughout the entire contract. If there are no plans for future utilization, it is advisable to remove it. Removing redundant code can significantly enhance code readability.

Suggestion:

It is recommended to remove the unused interface if there's no further design.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

SFL-1 Centralization Risk

Severity: Major

Status: Acknowledged

Code Location:

StarFarmLib.sol#691,697

Descriptions:

Centralization risk was identified in the smart contract.

- The privileged `admin` can set the migrator contract and migrate the lp token to another lp contract.

Any potential leaks or malicious manipulation could lead to serious issues.

Suggestion:

It is recommended to confirm if it aligns with the design.

Resolution:

The client replied that Admin address will use multi-signature management.

SFL-2 Lack of Events Emit

Severity: Minor

Status: Fixed

Code Location:

StarFarmLib.sol#538,542,546,550,557,564,585,595,603,642,649...

Descriptions:

The smart contract lacks appropriate events for monitoring sensitive operations, which could make it difficult to track sensitive actions or detect potential issues.

Suggestion:

It is recommended to emit events for those sensitive functions.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

SFL-3 Lack of Validation for Zero Address

Severity: Informational

Status: Fixed

Code Location:

StarFarmLib.sol#512

Descriptions:

There is no check for the zero address.

Suggestion:

It is recommended to add a check for the zero address.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

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.