IDOcontract Audit Report

Tue Mar 26 2024

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

ScaleBit

IDOcontract Audit Report

1 Executive Summary

1.1 Project Information

Description	Stake a variety of tokens and get a variety of rewards, time-locked projects
Type	Staking
Auditors	ScaleBit
Timeline	Mon Mar 18 2024 - Mon Mar 18 2024
Languages	Solidity
Platform	Merlin Chain
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/Merlinstarter/IDOContract
Commits	e5b62c8b27d9b07376f1cb7ff658b35bc5bd5a28 0403fcbb4609124d90172973362ed07e0bc883c3 11db43db5bb4e1c1cf47d49d7fb483433721ecf5 65ddcd0d601ca6f97d14a538cd323ffec48531e8

1.2 Files in Scope

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

ID	File	SHA-1 Hash
MSPSE2	MerlinStarterPublicSaleErc20.sol	f7d719bd60a5da270e8b06598e1d165abd8ca805
MSIDO	MerlinStarterIDO.sol	4173af93226804633e7bee39222089c374509e85
MSPS	MerlinStarterPublicSale.sol	3716fa0da302b29d38c4c8b0e551ea790c2d6ff8
MSIDOE2	MerlinStarterIDOErc20.sol	d607da64f1c33e62e3f18643f79dcaf87100280a

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	5	3	2
Informational	1	0	1
Minor	1	1	0
Medium	1	0	1
Major	1	1	0
Critical	1	1	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 Merlin Starter to identify any potential issues and vulnerabilities in the source code of the IDOcontract 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
MSI-1	Excess Funds Not Returned	Major	Fixed
MSI-2	Centralization Risk	Medium	Acknowledged
MSI-3	Lack of Events Emit	Informational	Acknowledged
MSP1-1	Incorrect Use `safetransferfrom` Function	Critical	Fixed
MSP1-2	Redundant Code	Minor	Fixed

3 Participant Process

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

Owner

- `Owner` can set the addresses of `rewardToken` and `oriToken` through `setParameters()` .
- `Owner` can set `mbStart` and `startTime` through `setStart()` or `setStartStageBeta()` .
- `Owner` can set `mbWhiteAddr` through `setbWhiteAddr()` .
- `Owner` can withdraw the platform currency to a specific `mFundAddress` in the contract through `withdraw()` .
- `Owner` can withdraw the Token in the contract through `withdrawToken()` .
- `Owner` can add a whitelist through `addWhiteAccount()` .
- `Owner` can remove the whitelist through `removeWhiteAccount()` .

User

- `User` can transfer assets or officially designated ERC20 tokens through `joinIdo()` to participate in the project.
- `User` can obtain various time-based rewards through `claimToken()` .

4 Findings

MSI-1 Excess Funds Not Returned

Severity: Major

Status: Fixed

Code Location:

MerlinStarterIDO.sol#273

Descriptions:

The excess funds invested by the user into the contract were not returned, and this event was recorded as `joinIdoPrice`.

```
    require(joinIdoPrice <= msg.value, "MerlinStarterIDO:value sent is not correct");

    _bAlreadyJoinIdoArr[_msgSender()]=true;

    _sumCount = _sumCount.add(1);
    _joinIdoPropertys[_sumCount].addr = _msgSender();
    _joinIdoPropertys[_sumCount].joinIdoAmount = joinIdoPrice;
    _joinIdoPropertys[_sumCount].time = block.timestamp;

    emit JoinIdoCoins(msg.sender, joinIdoPrice, _sumCount);
```

Suggestion:

It is recommended to return excess funds.

Resolution:

The client updated the code to refund excess funds to users and fixed this issue.

MSI-2 Centralization Risk

Severity: Medium

Status: Acknowledged

Code Location:

MerlinStarterIDO.sol#324;

MerlinStarterIDOErc20.sol#327;

MerlinStarterPublicSale.sol#380;

MerlinStarterPublicSaleErc20.sol#383

Descriptions:

Centralization risk was identified in the smart contract.

- The owner can set valut address to any address.
- The owner can set the distribution quantity of any reward and give it to any address.
- The owner can withdraw assets to a specific `mFundAddress` from the smart contract.
- The owner can add/remove whitelist addresses.
- The owner can update some parameters, such as: `oriToken` , `rewardToken` , `joinIdoPrice` , `stakeAmount` and `rewardAmount` .
- The owner can change the IDO status, such as: the start time, claim time, and whitelist status.

Suggestion:

It is recommended to take measures to mitigate this issue.

MSI-3 Lack of Events Emit

Severity: Informational

Status: Acknowledged

Code Location:

MerlinStarterIDO.sol#325,328,341,349,354,361;

MerlinStarterIDOErc20.sol#327,331,345,353,358,365;

MerlinStarterPublicSale.sol#380,383,387,399,404,411;

MerlinStarterPublicSaleErc20.sol#383,387,391,403,408,415

Descriptions:

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

```
function setParameters(address rewardTokenAddr) external onlyOwner {
    rewardToken = IERC20(rewardTokenAddr);
}
function setStart(bool bstart) external onlyOwner{
    mbStart = bstart;
    startTime = block.timestamp;
}
function setbWhiteAddr(bool bWhiteAddr) external onlyOwner{
    mbWhiteAddr = bWhiteAddr;
}
```

Suggestion:

It is recommended to emit events for those sensitive functions.

MSP1-1 Incorrect Use `safetransferfrom` Function

Severity: Critical

Status: Fixed

Code Location:

MerlinStarterPublicSaleErc20.sol#366,397;

MerlinStarterIDO.sol#321;

MerlinStarterPublicSale.sol#364;

MerlinStarterIDOErc20.sol#324

Descriptions:

The `safeTransferFrom` function requires the contract to call the approve function on msg.sender, and does not check whether the current contract rewardtoken balance is sufficient for transfer, which will cause the transfer to fail directly.

```
if(expectedAmount>0)rewardToken.safeTransferFrom(address(this),
_msgSender(),expectedAmount);
```

Suggestion:

It is recommended to change it to transfer function.

Resolution:

```
 rewardToken.safeTransfer(_msgSender(), amount);
```

MSP1-2 Redundant Code

Severity: Minor

Status: Fixed

Code Location:

MerlinStarterPublicSaleErc20.sol#139,190;

MerlinStarterPublicSale.sol#139,189;

MerlinStarterIDOErc20.sol#139,183;

MerlinStarterIDO.sol#139,181

Descriptions:

In the `withdraw_contribution` function, the code at line 921 is redundant as the `OnGoing` resource is already created when the user contributes.

```
    function name() public view returns (string memory) {
        return _name;
    }
    function symbol() public view returns (string memory) {
        return _symbol;
    }

address public constant DEAD_ADDRESS =
0x000000000000000000000000000000000000dEaD;
```

Suggestion:

It is recommended to remove the redundant code.

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