LinkStarter
Audit Report

Thu Jun 27 2024

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

ScaleBit

LinkStarter Audit Report

1 Executive Summary

1.1 Project Information

Description	The first native launchPad on zkLink Nova. LinkStarter aims to link exponential value growth with limitless liquidity.
Type	Launchpad
Auditors	ScaleBit
Timeline	Wed Jun 26 2024 - Thu Jun 27 2024
Languages	Solidity
Platform	zkLink Nova
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/garyphy/linkstarter https://github.com/Jmagicc/hardhat-example
Commits	https://github.com/garyphy/linkstarter: 21fb78677f15f01ace284b00c7fb49dd9aee74b4 https://github.com/Jmagicc/hardhat-example: 1a537f3bfa1b7dbf05bed765e6b8c3a3f086f100

1.2 Files in Scope

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

ID	File	SHA-1 Hash
LFI	linkstarter_fixed.sol	21c86486ab174511566d6f23f992acc1b3dbb42b
LIN	linkstarter.sol	9d9c495e1a050c14253a6aa4cddb2c98d09c14d6
LFI	contracts/linkstarter_fixed.sol	8c4763afdc326eb402e51007b21c2611a0732dec

1.3 Issue Statistic

Item	Count	Fixed	Partially Fixed	Acknowledged
Total	7	5	1	1
Informational	0	0	0	0
Minor	6	5	0	1
Medium	1	0	1	0
Major	0	0	0	0
Critical	0	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 LinkStarter to identify any potential issues and vulnerabilities in the source code of the LinkStarter 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 7 issues of varying severity, listed below.

ID	Title	Severity	Status
LFI-1	Centralization Risk	Medium	Partially Fixed
LFI-2	Lack of Events Emit	Minor	Fixed
LFI-3	`setParameters` Function Missing Check	Minor	Fixed
LFI-4	`claimDt` Logic Conflict	Minor	Fixed
LFI-5	`getWhiteAccountIth` is Not Rigorous	Minor	Fixed
LFI-6	Unnecessary Calculations	Minor	Fixed
LFI-7	User Funds May Locked	Minor	Acknowledged

3 Participant Process

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

User

- User can stake platform tokens to participate in the contract through the `joinIdo` function and obtain the stake record.
- User can claim the `rewardToken` obtained after staking through the `claimToken` function.
- User can claim the excess refund through the `claimBTC` function.

Owner

- Owner can set `rewardToken` , `joinIdoPrice` , `rewardBTCAmount` through `setParameters` function.
- Owner can set `idoStart` , `startTime` through `setStart` function.
- Owner can set `idoTimeRange` , `claimDt` , `claimPeriod` through `setDt` function.
- Owner can set `idoWhiteAddr` through `setbWhiteAddr` function to control the whitelist switch.
- Owner can withdraw the contract platform token balance through `withdraw` function.
- Owner can withdraw the contract specified token balance through `withdrawToken` function.
- Owner can add whitelist through `addWhiteAccount` function.
- Owner can add whitelist in batches through `addWhiteAccount(address[] calldata accountArr)` function.
- Owner can remove whitelist through `removeWhiteAccount` function.

4 Findings

LFI-1 Centralization Risk

Severity: Medium

Status: Partially Fixed

Code Location:

linkstarter_fixed.sol#307,338,343

Descriptions:

Centralization risk was identified in the smart contract.

1. `owner` can set the `rewardToken` address before starting. Users need to pay attention to whether it is consistent with expectations. `Owner` can change the `idoStart` status at any time.
2. `Owner` can withdraw the balance in the contract to `mFundAddress` (the specified address cannot be changed), which may cause users to be unable to refund excess funds or receive reward tokens.

Suggestion:

It's recommended that measures be taken to reduce the centralization issue like using multi-sig.

Resolution:

The client replied that they will adopt multi-sig to mitigate this issue.

LFI-2 Lack of Events Emit

Severity: Minor

Status: Fixed

Code Location:

linkstarter_fixed.sol#301,313,320,331

Descriptions:

The contract lacks appropriate events for monitoring sensitive operations, which could make it difficult to track sensitive actions or detect potential issues. For example, the `setParameters` , `setStart` , `setDt` , `setbWhiteAddr` function.

Suggestion:

It's recommended to emit events for those important functions.

LFI-3 `setParameters` Function Missing Check

Severity: Minor

Status: Fixed

Code Location:

linkstarter_fixed.sol#301

Descriptions:

1. In the `setParameters` function, it needs to be set before startup, but when setting the startup status, it will not check whether `rewardTokenAddr` is set. If it needs to be set before startup in the normal call logic, the operator may set it to 0 because `joinIdoPrice` and `rewardBTCAmount` already have default values, or the initial setting may overwrite the default value.
2. On the other hand, notice that in the variable comment there is a maximum value max for rewardBTCAmount, but there is no maximum limit in the function.

```
function setParameters(
    address rewardTokenAddr,
    uint256 joinIdoPrice0,
    uint256 rewardBTCAmount0
) external onlyOwner {
    require(!idoStart, "LinkStarterPublicSale: already Start!");
    rewardToken = IERC20(rewardTokenAddr);

    joinIdoPrice = joinIdoPrice0;
    rewardBTCAmount = rewardBTCAmount0;
}
```

Suggestion:

It's recommended to check the `rewardToken` setting before turning it on, and check that the parameter is not 0.

Resolution:

Added checks.

```
    require(!idoStart, "LinkStarterPublicSale: already Start!");
    require(rewardTokenAddr != address(0), "LinkStarterPublicSale: rewardTokenAddr is
```

```
zero address");
        require(joinIdoPrice0 > 0, "LinkStarterPublicSale: joinIdoPrice0 is zero");
        require(rewardBTCAmount0 > 0, "LinkStarterPublicSale: rewardBTCAmount0 is
zero");
```

LFI-4 `claimDt` Logic Conflict

Severity: Minor

Status: Fixed

Code Location:

linkstarter_fixed.sol#326,237

Descriptions:

In the `claimToken` function, check whether the current time is greater than `startTime + idoTimeRange` and `startTime + claimDt` respectively. Note that `startTime + idoTimeRange` mostly participates in time limit and calculation at the same time. If `claimDt < idoTimeRange`, there seems to be no point in checking. If `claimDt > idoTimeRange`, then `idoTimeRange` seems to have no meaning because it can be merged with `startTime`.

Suggestion:

It's recommended to confirm the business logic. The simplest check is to check `claimDt > idoTimeRange` in the `setDt` function.

Resolution:

The client adopted our suggestions.

LFI-5 `getWhiteAccountIth` is Not Rigorous

Severity: Minor

Status: Fixed

Code Location:

linkstarter_fixed.sol#138

Descriptions:

The alarm information and the return value are inconsistent. The function of this function is to obtain the whitelist address by index, not to check whether it is in the whitelist in the alarm.

```
function getWhiteAccountIth(uint256 ith)
    public
    view
    returns (address WhiteAddress)
{
    require(
        ith < _WhiteAddrArr.length,
        "LinkStarterPublicSale: not in White Adress"
    );
    return _WhiteAddrArr[ith];
}
```

Suggestion:

It's recommended to confirming the business logic and modifying it.

Resolution:

The client has modified the error message.

```
    require(
        ith < _WhiteAddrArr.length,
        "LinkStarterPublicSale: exceeded the length index of the whitelist collection"
    );
```

LFI-6 Unnecessary Calculations

Severity: Minor

Status: Fixed

Code Location:

linkstarter_fixed.sol#231

Descriptions:

In the `getIDOUnlockRatio` function, when `period` is 120, the `unlockRatio` is calculated to be 8000, but in fact the return value is still 10000, which seems to be an unnecessary calculation.

```
if (period > 120) return 10000;
uint256 unlockRatio = 8000 * period / 120;
return 2000 + unlockRatio;
```

Suggestion:

It's recommended to changing the check to `period >= 120` .

LFI-7 User Funds May Locked

Severity: Minor

Status: Acknowledged

Code Location:

linkstarter_fixed.sol#155

Descriptions:

1. Calling the `claimBTC` function before the `rewardBTCAmount` amount in the contract is reached will not return the funds, if the user's share has excess funds in the calculation and is about to be exceeded.
2. If the owner modifies the time, a new user participates and reaches the maximum limit, if the user has previously called the `claimBTC` function, his funds will be locked in the contract.

Suggestion:

It's recommended to take measures to avoid this issue.

Resolution:

The client reply contract is expected to be used only once and there will be no errors caused by changing the settings.

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.

