Attglobal-III
Audit Report

Fri Jan 10 2025

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

ScaleBit

Attglobal-III Audit Report

1 Executive Summary

1.1 Project Information

Description	This audit primarily includes a token-locking contract and a token distribution contract.
Type	DeFi
Auditors	ScaleBit
Timeline	Wed Jan 08 2025 - Fri Jan 10 2025
Languages	Solidity
Platform	Ethereum
Methods	Architecture Review, Unit Testing, Manual Review

1.2 Files in Scope

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

ID	File	SHA-1 Hash
L4PV1	AttGlobal-III/Lock4PassV1.sol	272e67159e8e70e44f0506bf984860e6bc38f197
L4WV1A	AttGlobal-III/Lock4WhitelistV1A.sol	18ad2f4224be68e177abf54e1b68e7f61d8a46da
FDV1	AttGlobal-III/FeeDistributorV1.sol	e20a29258f80216b1a035613b7f1c8afa1d9b8ca

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	13	8	5
Informational	4	3	1
Minor	6	4	2
Medium	2	1	1
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 Attglobal to identify any potential issues and vulnerabilities in the source code of the Attglobal-III 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 13 issues of varying severity, listed below.

ID	Title	Severity	Status
FDV-1	Redundant Code	Minor	Fixed
FDV-2	Use `+` Instead of `+=`	Minor	Fixed
FDV-3	MemberIds May Be Empty	Informational	Fixed
L4P-1	Incompatible With Deflationary Token	Medium	Acknowledged
L4P-2	Use `safeTransferFrom()` Instead of `transferFrom()`	Medium	Fixed
L4P-3	Incorrect Error Message	Minor	Fixed
L4P-4	Lack of Events Emit	Minor	Acknowledged
L4P-5	Redundant Parameters in lock Function	Informational	Fixed
L4P-6	Function Does Not Determine If The Value Exists	Informational	Fixed
L4W-1	Centralization Risk	Major	Acknowledged
L4W-2	`SUPPLY_LIMIT` May Not Be Enough	Minor	Acknowledged

L4W-3	Missing Parameter Checks	Minor	Fixed
L4W-4	Front-running Initialization	Informational	Acknowledged

3 Participant Process

Here are the relevant actors with their respective abilities within the Attglobal-III Smart Contract :

Owner

- **Owner** can call the `distribute` function to distribute rewards or tokens to a specified set of addresses.
- **Owner** can call the `distributeToTree` function to distribute rewards or tokens to a merkle tree structure for efficient claims.
- **Owner** can call the `burnATT` function to burn a specified amount of ATT tokens, effectively reducing the total supply.
- **Owner** can call the `addDaoMember` , `updateDaoMember` , and `removeDaoMember` functions to manage DAO members by adding, updating, or removing their addresses.
- **Owner** can call the `addCoreTeamMember` , `updateCoreTeamMember` , and `removeCoreTeamMember` functions to manage core team members by adding, updating, or removing their addresses.
- **Owner** can call the `setUnlockDuration` function to configure the duration required for unlocking tokens or assets.
- **Owner** can call the `addLockRule` , `updateLockRule` , and `removeLockRule` functions to manage lock rules, including adding new rules, updating existing ones, or removing outdated rules.

User

- **User** can call the `claim` function to claim rewards.
- **User** can call the `userClaimableAmount` function to check the amount of rewards.
- **User** can call the `lock` function to stake token.
- **User** can call the `unlock` function to unlock tokens.

4 Findings

FDV-1 Redundant Code

Severity: Minor

Status: Fixed

Code Location:

AttGlobal-III/FeeDistributorV1.sol#77

Descriptions:

In the `claim()` function, the code `MerkleProof.verify()` is executed twice.

Suggestion:

It is recommended to modify the code to avoid executing redundant codes.

Resolution:

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

FDV-2 Use `+` Instead of `+=`

Severity: Minor

Status: Fixed

Code Location:

AttGlobal-III/FeeDistributorV1.sol#92

Descriptions:

There are some functions that the user can call only once, so there's no need to use `+=` to update the variables, using `=` is enough.

```
userCurrentLockedId[msg.sender] += 1;

userClaimedAmount[distributeId][msg.sender] += claimAmount;
```

Suggestion:

It is recommended to update as suggested.

Resolution:

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

FDV-3 MemberIds May Be Empty

Severity: Informational

Status: Fixed

Code Location:

AttGlobal-III/FeeDistributorV1.sol#232

Descriptions:

The `_remove` function, although bounds-checked, can be problematic when the array is empty.

Suggestion:

It is recommended to determine if the array is 0.

Resolution:

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

L4P-1 Incompatible With Deflationary Token

Severity: Medium

Status: Acknowledged

Code Location:

AttGlobal-III/Lock4PassV1.sol;

AttGlobal-III/Lock4WhitelistV1A.sol;

AttGlobal-III/FeeDistributorV1.sol

Descriptions:

In the `Lock4WhitelistV1A`, `Lock4PassV1`, and `FeeDistributorV1` contracts, due to the unknown address of the `IERC20 public token`, when the token is deflationary, the number of tokens transferred to the contract by the user may not be accurate.

Suggestion:

Since it's not known exactly what type of token this is, it's recommended to confirm whether such a question would conflict with the design philosophy.

L4P-2 Use `safeTransferFrom()` Instead of `transferFrom()`

Severity: Medium

Status: Fixed

Code Location:

AttGlobal-III/Lock4PassV1.sol;

AttGlobal-III/Lock4WhitelistV1A.sol;

AttGlobal-III/FeeDistributorV1.sol

Descriptions:

The `ERC20.transfer()` and `ERC20.transferFrom()` functions return a boolean value indicating success. This parameter needs to be checked for success.

Some ERC20 tokens that are not compliant with the specification could return false from the transfer and transferFrom functions call to indicate that the transfer fails, but the calling contract would not notice the failure if the return value is not checked.The EIP-20 specification requires to check the return value. Also due to the unknown custodian type, it can cause reentrant issue.

The protocol uses `transferFrom()` to transfer funds from users, but it does not perform any checks on the return value.

Suggestion:

It is recommend to use OpenZeppelin's SafeERC20 versions with the safeTransfer and safeTransferFrom functions that handle the return value check as well as non-standard-compliant tokens.

Resolution:

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

L4P-3 Incorrect Error Message

Severity: Minor

Status: Fixed

Code Location:

AttGlobal-III/Lock4PassV1.sol#89

Descriptions:

There is a mismatch between the error message and the condition judged by the `require` statement.

Suggestion:

It is recommended to update the error message to accurately describe the condition, such as "User has already staked".

Resolution:

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

L4P-4 Lack of Events Emit

Severity: Minor

Status: Acknowledged

Code Location:

AttGlobal-III/Lock4PassV1.sol#181,185,189,193,201,205,209;

AttGlobal-III/Lock4WhitelistV1A.sol#110,118,122,126,130,134,138;

AttGlobal-III/FeeDistributorV1.sol#211,215,223

Descriptions:

The contract lacks appropriate events for some key functions. The lack of event records for these functions may cause inconvenience in the subsequent tracking and contract status changes.

Suggestion:

It is recommended that events for these functions be emitted.

L4P-5 Redundant Parameters in lock Function

Severity: Informational

Status: Fixed

Code Location:

AttGlobal-III/Lock4PassV1.sol#87

Descriptions:

The `ttAmount` and `attAmount` parameters in the `lock` function are redundant because they are required to match the values defined in the `lockRules`.

Suggestion:

It is recommended that the code logic be simplified.

Resolution:

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

L4P-6 Function Does Not Determine If The Value Exists

Severity: Informational

Status: Fixed

Code Location:

AttGlobal-III/Lock4PassV1.sol#177-203

Descriptions:

Functions that update global information do not determine if the field corresponding to the id exists, such as the `updateCoreTeamMember` and `removeCoreTeamMember` functions.

Suggestion:

It is recommended to determine the exist of the value before removing or editing it.

Resolution:

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

L4W-1 Centralization Risk

Severity: Major

Status: Acknowledged

Code Location:

AttGlobal-III/Lock4WhitelistV1A.sol#146

Descriptions:

Centralization risk was identified in the smart contract:

- Administrators can call `transferToken()` to transfer tokens to any address.
- Administrators can call `transferNFT()` to transfer NFT to any address.
- Administrators can update the configs of the contract.

Suggestion:

It is recommended that measures be taken to reduce the risk of centralization, such as a multi-signature mechanism.

L4W-2 `SUPPLY_LIMIT` May Not Be Enough

Severity: Minor

Status: Acknowledged

Code Location:

AttGlobal-III/Lock4WhitelistV1A.sol#83;

AttGlobal-III/Lock4PassV1.sol#87

Descriptions:

The lock function does not determine the relationship between the number of `REWARD_WHITELIST_AMOUNT` and the number of remaining `SUPPLY_LIMITs` when updating the `SUPPLY_LIMIT`, which may result in the `SUPPLY_LIMIT` underflowing and thus the function call failing. The lock function has the same problem.

Suggestion:

It is recommended to determine if `SUPPLY_LIMIT` is sufficient at the beginning of the function.

L4W-3 Missing Parameter Checks

Severity: Minor

Status: Fixed

Code Location:

AttGlobal-III/Lock4WhitelistV1A.sol#118

Descriptions:

When initializing or setting fee rates, the rate values should be range-checked to avoid calculation errors caused by unexpected settings.

```
function setFEE_RATE(uint256 rate) public onlyOwner {
        FEE_RATE = rate;
    }
```

Suggestion:

It is recommended to add a range-check.

Resolution:

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

L4W-4 Front-running Initialization

Severity: Informational

Status: Acknowledged

Code Location:

AttGlobal-III/Lock4WhitelistV1A.sol#23;

AttGlobal-III/FeeDistributorV1.sol#21

Descriptions:

The initialization function lacks permission control, and the initialization of the function may be preempted, resulting in incorrect initialization data.

Suggestion:

It is recommended to initialize it immediately after creating the contract.

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.