AWT-IDO-Claim Smart Contract **Audit Report**

07/11/2023

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

ScaleBit

AWT-IDO-Claim Audit Report

1 Executive Summary

1.1 Project Information

Description	A token distribution smart contract.
Type	DeFi
Auditors	ScaleBit
Timeline	July 10, 2023 – July 11, 2023
Languages	Solidity
Platform	Polygon
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/Metagame-Industries/AWT-IDO-Claim
Commits	f53d7d5d5462fb795538e740ceb4ebacd498b448 2e4f89f6dfe659c34d1c3d1255a6905c46153ec0

1.2 Files in Scope

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

ID	Files	SHA-1 Hash
IDO	IDO_TokenDistribution.sol	9db26af1dad67b9ee81627ecb608e2f8e250ec39

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	5	3	2
Informational			
Minor	4	3	1
Medium			
Major	1		1
Critical			

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 **Metagame Industries** to identify any potential issues and vulnerabilities in the source code of the **AWT-IDO-Claim** 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
IDO-01	Declaring `immutable` for Variables	Minor	Fixed
IDO-02	Missing Zero Address Validation	Minor	Fixed
IDO-03	Missing Events	Minor	Acknowledged
IDO-04	Unchecked Return Value	Minor	Fixed
IDO-05	Centralization Risk	Major	Acknowledged

3 Participant Process

Here are the relevant actors with their respective abilities within the **AWT-IDO-Claim** Smart Contract:

Admin

- Admin can update the token distribution frequency through `setMonth()`.
- Admin can update the Merkle Root through `setMerkleRoot()`.
- Admin can update the distributed token address through `setTokenAddress()`.
- Admin can withdraw all the tokens from the contract through `withdrawTokens()`.
- Admin can transfer ownership through `transferAdmin()`.

User

- User with signature can claim tokens a maximum of `month` times through `claimTokens()`.

4 Findings

IDO-01 Declaring `immutable` for Variables

Severity: Minor

Status: Fixed

Location: IDO_TokenDistribution.sol#L8

Description:

The `admin` is not updated following deployment and should be declared immutable to save gas.

Suggestion: It is recommended to declare the `admin` as `immutable`.

Resolution: The client adds a `transferAdmin()` function so it is not an issue.

IDO-02 Missing Zero Address Validation

Severity: Minor

Status: Acknowledged

Location: IDO_TokenDistribution.sol#L49

Description:

Addresses should be validated to ensure they are not zero, as otherwise, it may lead to issues.

Suggestion: It is recommended to add zero address validation.

Resolution: The client has followed our suggestion and fixed this issue.

IDO-03 Missing Events

Severity: Minor

Status: Acknowledged

Location: IDO_TokenDistribution.sol#L21, L45, L49, L53

Description:

Emitting events for critical operations in the contract is essential as it enables monitoring of the contract's operational state and facilitates prompt identification of operational risks.

Suggestion: It is recommended to emit events for these critical operations.

IDO-04 Unchecked Return Value

Severity: Minor

Status: Acknowledged

Location: IDO_TokenDistribution.sol#L34, L56

Description:

For `ERC20.transfer()`, it is usually good to add a require-statement that checks the return value or to use something like `safeTransfer`; unless one is sure the given token reverts in case of a failure.

Suggestion: It is recommended to add validation for the return value.

Resolution: The client has followed our suggestion and fixed this issue.

IDO-05 Centralization Risk

Severity: Major

Status: Acknowledged

Location: IDO_TokenDistribution.sol#L21, L45, L49, L53

Description:

The contract exhibits a level of centralization that poses potential risks to the overall security, trust, and resilience of the system.

Admin

- Admin can update the token distribution frequency through `setMonth()`.
- Admin can update the Merkle Root through `setMerkleRoot()`.
- Admin can update the distributed token address through `setTokenAddress()`.
- Admin can withdraw all the tokens from the contract through `withdrawTokens()`.
- Admin can transfer ownership through `transferAdmin()`.

Suggestion: It is recommended to take measures to mitigate 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.