Abyss World Token Smart Contract

Audit Report

MOVEBIT

contact@movebit.xyz

https://twitter.com/movebit_

06/05/2023

Abyss World Token Smart Contract Audit Report

1 Executive Summary

1.1 Project Information

Description	An ERC20 token.
Type	ERC20
Auditors	MoveBit
Timeline	June 5, 2023
Languages	Solidity
Platform	Polygon
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/Metagame-Industries/AbyssWorld-Token
Commits	8f5c824c1d740567b2668eddf46882119084753c

1.2 Files in Scope

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

ID	Files	SHA-1 Hash
AWT	solidity/AWT.sol	2dbce7a498687e5a6f16938cefaa3297bef5fa68

1.3 Issue Statistic

Item	Count	Fixed	Partially Fixed

Total	1		1
Informational			
Minor			
Medium	1		1
Major			
Critical			

1.4 MoveBit Audit BreakDown

MoveBit 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 **Abyss World Token** 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 have identified **1** issues of varying severity, listed below.

ID	Title	Severity	Status
AWT-01	Centralization Risk in Token Distribution	Medium	Partially Fixed

3 Participant Process

Here are the relevant actors with their respective abilities within the `Abyss World Token` Smart Contract:

Deployer

- Deployer gets `10000000000 * 10 ** 18` tokens from the deployment.

Owner

- Owner can take a snapshot of the token balances at the current block number through `snapshot()`.
- Owner can transfer the ownership through `transferOwnership()`.
- Owner can renounce the ownership through `renounceOwnership()`.

User

- User can transfer the token through `transfer()` and `transferFrom()`.
- User can approve itself token to others through `approve()`.
- User can update the allowance through `increaseAllownce()` and `decreaseAllownce()`.
- User can burn itself token through `burn()` and `burnFrom()`.

4 Findings

AWT-01 Centralization Risk in Token Distribution

Severity: Medium

Status: Partially Fixed

Code Location: solidity/AWT.sol#L11

Descriptions: A total amount of `10000000000 * 10 ** 18` tokens are sent to the deployer in the deployment. This distribution mechanism raises concerns regarding the concentration of token ownership and the associated centralization of power.

Suggestion: It is recommended to implement a fair and transparent token distribution mechanism to ensure broader participation and decentralized ownership.

Resolution: The client responds that they will use a 3/4 Safeheron Multi-sig.

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.