Agora ausd Smart Contract

Audit Report

contact@bitslab.xyz

https://twitter.com/movebit_

Tue Sep 03 2024

Agora ausd Smart Contract Audit Report

1 Executive Summary

1.1 Project Information

Description	AUSD is a digital dollar issued by Agora.
Type	Token
Auditors	MoveBit
Timeline	Mon Aug 26 2024 - Mon Aug 26 2024
Languages	Move
Platform	Sui
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/MystenLabs/ausd
Commits	469648c5fbcaa397e34c0a2410c44a0bef1a44dc 51673cb744cddf21fe277f01b8182f67127645dc

1.2 Files in Scope

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

ID	File	SHA-1 Hash
MOV	packages/Move.toml	cbdae49512bb271e2166e3980bcf9706a7671b85
CON	packages/sources/constants.move	857e418f13545c0aa0cea6392a2d6f6550318fbf
AUS	packages/sources/ausd.move	55c198bae0ba0ce736bb4c0d275a4e75a79f4eb3
ADM	packages/sources/admin/admin.move	840ad4593066214303d7aa5e1640a70da040b4f9
PRO	packages/sources/admin/proposals.move	e8efd4af036f5708e1429ccdbbdcfd93f33c410b
FRE	packages/sources/roles/freezer.move	fc0169b7dbaa287861633fc31a6277527b63c6b2
PAU	packages/sources/roles/pauser.move	0cb8a1b9e231b05820f9198caa81fa92cee567c6
BUR	packages/sources/roles/burner.move	512b0af12722f1abd6e9dc11f88aabe3ae992631
MIN	packages/sources/roles/minter.move	0fb9cb90d37d630d884c7ea855283243a96379f9
ROL	packages/sources/roles.move	0e2889d0d2e98fca22f43685db5abede39fea077
SET	packages/sources/setup.move	b2115ed0e6c5c2fea06d30ff5e9783373859cf77

TRE	packages/sources/treasury.move	e5645a9a969834370088a5c5a020ebc4247a2698

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	1	1	0
Informational	0	0	0
Minor	1	1	0
Medium	0	0	0
Major	0	0	0
Critical	0	0	0

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 by bit operations
- Number of rounding errors
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting
- Unchecked CALL Return Values
- The flow of capability
- Witness Type

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) Formal Verification

Perform formal verification for key functions with the Move Prover.

(4) 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 Agora to identify any potential issues and vulnerabilities in the source code of the Agora ausd 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 1 issues of varying severity, listed below.

ID	Title	Severity	Status
ADM-1	Lack of Events Emit	Minor	Fixed

3 Participant Process

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

Sudo Admin

- Sudo admin can reject a proposal through `reject_proposal()`.
- Sudo admin can set the contract version through `set_version()`.
- Sudo admin can authorize admin through `authorize_admin()`.
- Sudo admin can authorize minter through `authorize_minter()`.
- Sudo admin can authorize freezer through `authorize_freezer()`.
- Sudo admin can authorize burner through `authorize_burner()`.
- Sudo admin can authorize pauser through `authorize_pauser()`.
- Sudo admin can deauthorize a role through `deauthorize()`.

Time-locked Admin

- Time-locked admin can authorize admin after waiting a specified locking period through `authorize_admin()`.
- Time-locked admin can authorize a minter after waiting a specified locking period through `authorize_minter()`.
- Time-locked admin can authorize a freezer after waiting a specified locking period through `authorize_freezer()`.
- Time-locked admin can authorize a burner after waiting a specified locking period through `authorize_burner()`.
- Time-locked admin can authorize a pauser after waiting a specified locking period through `authorize_pauser()`.
- Time-locked admin can deauthorize a role after waiting a specified locking period through `deauthorize()`.

Minter

- Minter can mint `AUSD` tokens and transfer them to the recipient through `mint()`.

Burner

- Burner can burn the `AUSD` tokens through `burn`.

Pauser

- Pauser can pause various functionality of the contract through `pause`.
- Pauser can resume various functionality of the contract through `resume`.

Freezer

- Freezer can add the address in the denylist which freezes its transaction of `AUSD` through `freeze_address()`.
- Freezer can remove the address from the denylist which unfreezes its transaction of `AUSD` through `unfreeze_address()`.

Proposer

- Proposer can execute proposal through `execute_proposal`.
- Proposer can reject proposal through `reject_proposal`.

4 Findings

ADM-1 Lack of Events Emit

Severity: Minor

Status: Fixed

Code Location:

packages/sources/admin/admin.move#103,120,138,153,168,184;

packages/sources/roles/freezer.move#9,20;

packages/sources/roles/pauser.move#8,13

Descriptions:

The contract lacks appropriate events for monitoring operations, such as `authorize_admin`, `authorize_minter`, `pause`, `resume`, and so on, which could make it difficult to track sensitive actions or detect potential issues.

Suggestion:

It is recommended to emit events for the function.

Resolution:

The client fixed this issue and adopted the suggestion.

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.