Pausable UDT
Audit Report

Wed Dec 18 2024

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

ScaleBit

Pausable UDT Audit Report

1 Executive Summary

1.1 Project Information

Description	This is a SSRI-compliant smart contract that implements a pausable UDT (User-Defined Token) with the SSRI protocol. By maintaining and referring to the `UDTPausableData` , transactions involving UDT would in effect pause minting, transferring, and burning behaviors when any paused lock hash is involved in the transaction. Moreover, the pause list would be publicly accessible and readable to the general public for any decentralized financial arrangements.
Type	Token
Auditors	ScaleBit
Timeline	Mon Dec 16 2024 - Wed Dec 18 2024
Languages	Rust
Platform	CKB
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/ckb-devrel/pausable-udt
Commits	dc6c164b8956072180b97e0e3ce225c21aff70a9

1.2 Files in Scope

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

ID	File	SHA-1 Hash
MAI	contracts/pausable-udt/src/main.rs	690d59d67a071f35212c65f0638ddb54849024e3
ERR	contracts/pausable-udt/src/error.rs	5e348b873be53be556c402c1916e75e466b0903b
FAL	contracts/pausable-udt/src/fallback.rs	2d16fc6496c295d40d4702ce2fae79c54048a0ff
MOD	contracts/pausable-udt/src/modules.rs	5507e0cfd43e9b3cabe388c96569281444f6c5e0
UTI	contracts/pausable-udt/src/utils.rs	a97a44cff60ea3dd408790b22d3ab769b7efe6dd

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	1	0	1
Informational	1	0	1
Minor	0	0	0
Medium	0	0	0
Major	0	0	0
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 CALL Return Values
- Functionality Checks
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic issues
- Replay attacks
- Coding style issues

1.5 Methodology

The security team adopted the **"Testing and Automated Analysis"**, **"Code Review"** 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 nervos to identify any potential issues and vulnerabilities in the source code of the Pausable UDT 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
MAI-1	Centralization Risk	Informational	Acknowledged

3 Participant Process

Here are the relevant actors with their respective abilities within the Pausable UDT Smart Contract :
Script `<pausable-udt>`

- This project would only introduce one new `Script` as the asset type script.
- To be compatible with those UDT issuance that would take place before Script `<pausable-udt>` and scheduled to upgrade when it becomes available, we would use the same rule for args definition as what sUDT/xUDT requires: if at least one input cell in the transaction uses owner lock specified by the `pausable-udt` as its cell lock, it enters governance operation and minting would be allowed.
- By default, the contract code itself maintains a pause list of lock hashes that can only be updated by upgrading; if necessary, we can also maintain external lists of lock hashes in extra cell with Type ID implementation and point to them at `UDTPausableData.next_type_script` in a chained pattern.

4 Findings

MAI-1 Centralization Risk

Severity: Informational

Status: Acknowledged

Code Location:

contracts/pausable-udt/src/main.rs

Descriptions:

There is a risk of centralization in the token protocol, where administrators can hack users so that they can't trade tokens, and can mint tokens at will.

Suggestion:

Administrators and token owners can use multi-signature account management.

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.