Rock'n'Block

SMART CONTRACTS EXPERTISE — RIGHT WAY TO SUCCESS
Freezylist Token contract report

If you have any questions concerning smart contract design and audit, feel free to contact audit@rocknblock.io

Content

Description of the complex of procedures for auditing a smart contract

1. Primary architecture review

- Checking the architecture of the contract.
- Correctness of the code.
- Check for linearity, shortness and self-documentation.
- Static verification and code analysis for validity and the presence of syntactic errors.

2. Comparison of requirements and implementation

- Checking the code of the smart contract for compliance with the requirements of the customer code logic, writing algorithms, matching the initial constant values.
- Identification of potential vulnerabilities

3. Testing according to the requirements

- Control testing of a smart contract for compliance with specified customer requirements.
- Running tests of the properties of the smart contract in test net.

Terms of Reference for the creation of a smart contract

FRZL Smart Contract

Token details:

- Token Name - Freezylist Token
- Token Symbol - FRZL
- Decimals - 18
- Total Supply - 100,000,000,000 FRZL
- Token address - 0xc8433a1e284a2533def45791aae3d6dc6f7d5277
- Token Creator address - 0x1e1fedbeb8ce004a03569a3ff03a1317a6515cf1
- Token Owner address - 0x9eA57604f8EE225B07796E77051BB211c81551c7
- Type of token - ERC20
- Minted tokens:
  0x9ea57604f8ee225b07796e77051bb211c81551c7 - 99,999,900,000 FRZL
  0xa5d14439918f19232cc15d4ff61242395fd57f2f - 100,000 FRZL
- No more tokens can be minted

Token functions:

- mint
- mintandfreeze
- transfer
- transferFrom
- burn
- freezeTo
- releaseAll
- releaseOnce
- pause
- unpause
- renounceOwnership
- transferOwnership
- approve
- increaseApproval
- decreaseApproval

List of audited files

Github:

https://github.com/Rock-n-Block/AUDIT/blob/main/FRZL

The following 1 .sol, files were reviewed:

- FRZL.sol

Uploaded source code

Etherscan:

https://etherscan.io/address/0xc8433a1e284a2533def45791aae3d6dc6f7d5277#code

Review of smart contract #1

FRZL smart contract review #1

https://github.com/Rock-n-Block/AUDIT/blob/main/FRZL

1. Overview

1.1. Overall code quality is normal, but have few drawbacks, some of them are important to security.

2. Critical issues

2.1. No critical vulnerabilities were found.

3. High issues

3.1. No high vulnerabilities were found.

4. Medium issues

4.1. No medium vulnerabilities were found.

5. Low issues

5.1. The disparity of expectation in release functions: Users use releaseOnce() and releaseAll() to release their frozen tokens once the freezing period has elapsed. In the event, a user does not hold any frozen tokens eligible for release, the releaseOnce() function reverts state changes. This is not the case forreleaseAll(), which will simply do nothing. While this does not pose a significant danger for users, we recommend the inconsistency to be addressed.

5.2. Overuse of public function visibility: The reviewed token contract is assembled using a script which generates a file of constants with which the token contract will set its initial values. Because each constant is marked public, Solidity implicitly creates a publicly visible getter function with the same name. While using constants is generally efficient, excessive use of public fields:

1. Makes a contract little more expensive to deploy (longer bytecode)

2. Makes a contract little more expensive to use, as each additional function selector created by these implicit getters means more options to traverse at runtime.

Consider removing the word public from each constant unless absolutely necessary. They will be set to the default, internal, meaning they will still be accessible internally to the contract.

6. Testings

6.1. **Successful** Deployment token in test net. Open link

6.2. **Successful** Check name, symbol, decimals.

6.3. **Successful** Owner of contracts sets correctly.

6.4. **Successful** Checking the distribution function of tokens. Tokens are distributed correctly. Open link

6.5. **Successful** Transfer tokens from address to address. Open link

6.6. **Successful** The MINT function. Tokens are minted and sent to an address. Open link

6.7. **Successful** The MintAndFreeze function. Tokens are minted, frozen and sent to an address. Tokens are displayed on the address, but cannot be sent to another address until they are unfrozen by the token holder after the unfreezing date. Open link

6.8. **Successful** Pause. Token transfer function is disabled. No one can transfer tokens from address to address. Open link

6.9. **Successful** Transfer token function disabled. Transaction failed. Open link

6.10. **Successful** Unpause. Token transfer function is enabled. Any holder can transfer tokens from address to address. Open link

6.11. **Successful** Transfer token function enabled. Transaction successful. Open link

6.12. **Successful** Burn. Burning tokens from management address. Open link

6.13. **Successful** Finalize. The mintable function is disabled. No more tokens can be minted. Open link

6.14. **Successful** transferOwnership. Transfer of rights to manage the token. Open link

Results of contract audit

The information in this review is a list of recommendations on what needs to be done to ensure the quality and security of the smart contract.

The Rock`n`block experts conducted the verification of the smart contract. Based on the results of the reviewing and testing, it is established that the token smart contract complies with the specifications specified in the terms of reference.

During the reviewing and testing of the contracts, critical errors and possible vulnerabilities were not detected.

For all questions regarding the review and testing of the smart contract, we recommend contacting audit@rocknblock.io