



> Smart
Contract

Audit #

ALGOBLOCKS

Feb 06
2022



TABLE OF CONTENTS

Table of contents.....	3
Methodology	4
Structure of contact Algotblocks.sol	5
Structure of contact Wallet.sol	8
Verification check sums	12

METHODOLOGY

MAIN TESTS LIST:

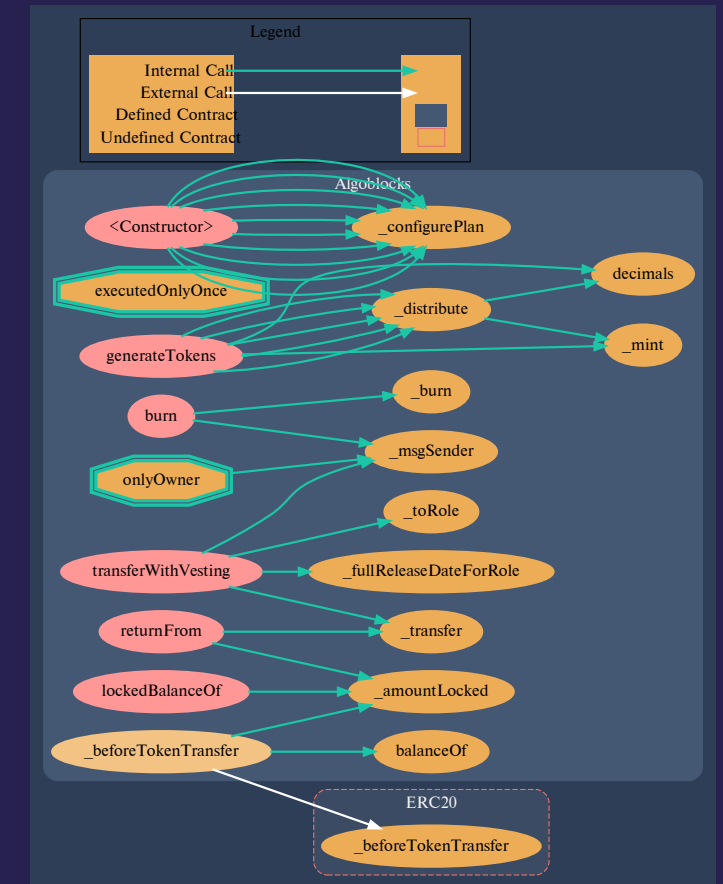
- ◆ Best code practices
- ◆ ERC20/BEP20 compliance (if applicable)
- ◆ FA2 compliance (if applicable)
- ◆ Logical bugs
- ◆ General Denial Of Service(DOS)
- ◆ Locked ether
- ◆ Private data leaks
- ◆ Using components with known vulns
- ◆ Weak PRNG
- ◆ Unused vars
- ◆ Unchecked call return method
- ◆ Code with no effects
- ◆ Pool Asset Security (backdoors in the underlying ERC-20)
- ◆ Function visibility
- ◆ Use of deprecated functions
- ◆ Authorization issues
- ◆ Re-entrancy
- ◆ Arithmetic Over/Under Flows
- ◆ Hidden Malicious Code
- ◆ External Contract Referencing
- ◆ Short Address/ Parameter Attack
- ◆ Race Conditions / Front Running
- ◆ Uninitialized Storage Pointers
- ◆ Floating Points and Precision
- ◆ Signatures Replay

STRUCTURE OF CONTRACT

ALGOBLOCKS.SOL

CONTRACT METHODS ANALYSIS:

- ◆ function generateTokens(
 - address[] memory teamAddresses,
 - uint256[] memory teamAmounts,
 - address[] memory advisorsAddresses,
 - uint256[] memory advisorsAmounts,
 - address[] memory privateInvestorsAddresses,
 - uint256[] memory privateInvestorsAmounts,
 - address[] memory strategicInvestorsAddresses,
 - uint256[] memory strategicInvestorsAmounts,
 - address[] memory idInvestorsAddresses,
 - uint256[] memory idInvestorsAmounts
 -)
- Vulnerabilities not detected



Pic. 1.1
Algotricks.sol

- ◆ `burn(uint256 amount)`
Vulnerabilities not detected
- ◆ `transferWithVesting(address to, uint256 amount, uint role)`
Vulnerabilities not detected
- ◆ `_beforeTokenTransfer(address from, address to, uint256 amount)`
Vulnerabilities not detected
- ◆ `_toRole(uint roleId)`
Vulnerabilities not detected
- ◆ `_configurePlan(Role role, uint cliff, uint vestingMonths, uint dayOneRelease)`
Vulnerabilities not detected
- ◆ `lockedBalanceOf(address account)`
Vulnerabilities not detected
- ◆ `returnFrom(address account)`
Vulnerabilities not detected
- ◆ `_amountLocked(address account)`
Vulnerabilities not detected

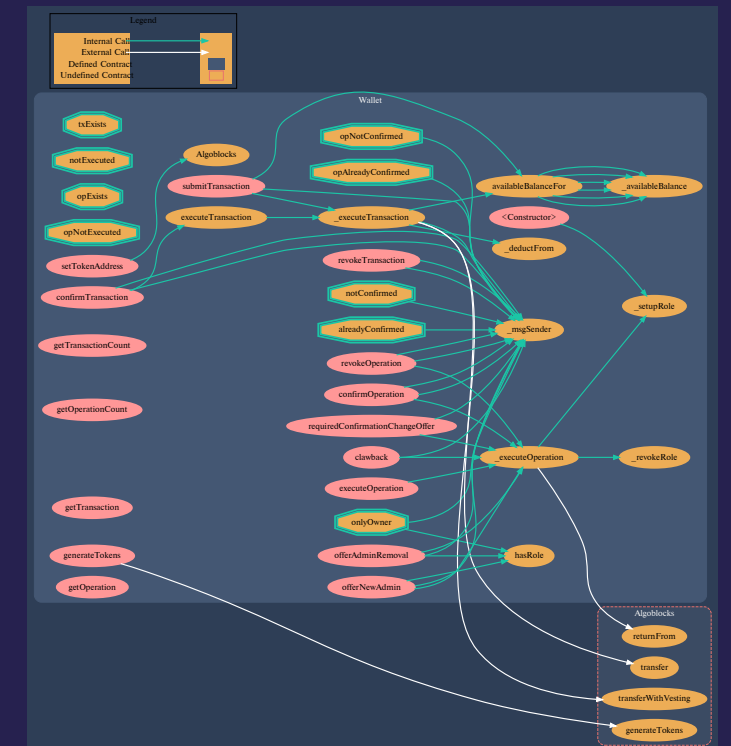
- ◆ `_distribute(Role role, address[] memory addresses, uint256[] memory amounts)`
Vulnerabilities not detected
- ◆ `_fullReleaseDateForRole(Role role, address account)`
Vulnerabilities not detected

STRUCTURE OF CONTRACT

WALLET.SOL

CONTRACT METHODS ANALYSIS:

- ◆ `setTokenAddress(address tokenAddress)`
Vulnerabilities not detected



Pic. 1.2
Wallet.sol

- ◆ generateTokens(
 address[] memory teamAddresses,
 uint256[] memory teamAmounts,
 address[] memory advisorAddresses,
 uint256[] memory advisorAmounts,
 address[] memory
privateInvestorAddresses,
 uint256[] memory privateInvestorAmounts,
 address[] memory
strategicInvestorAddresses,
 uint256[] memory
strategicInvestorAmounts,
 address[] memory idoInvestorAddresses,
 uint256[] memory idoInvestorAmounts)
Vulnerabilities not detected
- ◆ getTransactionCount()
Vulnerabilities not detected
- ◆ getOperationCount()
Vulnerabilities not detected
- ◆ getTransaction(uint _txIndex)
Vulnerabilities not detected
- ◆ submitTransaction(address to, uint value,
Pool pool, uint role)
Vulnerabilities not detected
- ◆ confirmTransaction(uint txIndex)
Vulnerabilities not detected
- ◆ revokeTransaction(uint txIndex)
Vulnerabilities not detected

- ◆ `executeTransaction(uint txIndex)`
Vulnerabilities not detected
- ◆ `_executeTransaction(uint txIndex)`
Vulnerabilities not detected
- ◆ `getOperation(uint _opIndex)`
Vulnerabilities not detected
- ◆ `offerAdminRemoval(address admin)`
Vulnerabilities not detected
- ◆ `offerNewAdmin(address newAdmin)`
Vulnerabilities not detected
- ◆ `requiredConfirmationChangeOffer(uint newRequiredConfirmationCount)`
Vulnerabilities not detected
- ◆ `clawback(address account)`
Vulnerabilities not detected
- ◆ `confirmOperation(uint opIndex)`
Vulnerabilities not detected
- ◆ `revokeOperation(uint opIndex)`
Vulnerabilities not detected
- ◆ `executeOperation(uint opIndex)`
Vulnerabilities not detected

- ◆ `availableBalanceFor(Pool pool)`
Vulnerabilities not detected
- ◆ `_deductFrom(Pool pool, uint256 amount)`
Vulnerabilities not detected
- ◆ `_availableBalance(
 uint256 total,
 uint256 remaining,
 uint cliffPeriod,
 uint vestingCount,
 uint vestingUnit
)`
Vulnerabilities not detected

VERIFICATION CHECK SUMS

Contract Name	Solc version	Optimisation	Bytecode hash (SHA 256)
Algoblocks.sol	0.8.0	200	29f0ee2f1a3de7fb5d01b7cd 8e1903ca4dea09cb7128abd 8b8220c5eba1ec5a3
Wallet.sol	0.8.0	200	2b67b9ff0ae0d3deaac3473 46337706684ba607194df4 8eb2dcf1b17dc34374f

EXECUTIVE SUMMARY

Based on our understanding contract, as well as the nature of the vulnerabilities discovered, their exploitability, and the potential impact we have assessed the level of risk for your organisation to be Low.

Overall security rating: **High**



Get In Touch

info@smartstate.tech

smartstate.tech

