SEP 2021

AvaXLauncher Smart Contract Final Audit Report

CONTENT

Scope of Audit

The scope of this audit was to analyze and document the AvaXLauncher Token smart contract codebase for quality, security, and correctness.

Check Vulnerabilities

- Re-entrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Use of tx.origin
- Exception disorder
- Gasless send
- Balance equality
- Byte array
- Transfer forwards all gas
- ERC-20 API violation
- Malicious libraries
- Compiler version not fixed
- Redundant fallback function
- Send instead of transfer
- Style guide violation
- Unchecked external call
- Unchecked math
- Unsafe type inference
- Implicit visibility level

Techniques and Methods

Throughout the audit of smart contracts, care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments match logic and expected behaviour.
- Token distribution and calculations are as per the intended behaviour mentioned in the whitepaper.
- Implementation of BEP-20 token standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods, and tools were used to review all the smart contracts.

Structural Analysis

In this step, we have analyzed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.

Static Analysis

A static Analysis of Smart Contracts was done to identify contract vulnerabilities. In this step, a series of automated tools are used to test the security of smart contracts.

Code Review / Manual Analysis

Manual Analysis or review of code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analyzed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of the automated analysis were manually verified.

Gas Consumption

In this step, we have checked the behaviour of smart contracts in production. Checks were done to know how much gas gets consumed and the possibilities of optimization of code to reduce gas consumption.

Tools and Platforms used for Audit

Remix IDE, Truffle, Truffle Team, Solhint, Mythril, Slither, Solidity statistic analysis, Theo.

Issue Categories

High Severity Issues

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality, and we recommend these issues be fixed before moving to a live environment.

Medium Severity Issues

The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems, and they should still be fixed.

Low Severity Issues

Low-level severity issues can cause minor impact and or are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

Informational Issues

These are four severity issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Number of security issues per severity

TYPE	HIGH	MEDIUM	LOW	INFORMATIONAL
Open	0	0	1	1
Acknowledged	0	0	1	1
Closed	0	0	1	1

Introduction

During the period of August 31, 2021 to September 1, 2021 - Crypticocean Team performed a security audit for AvaXLauncher smart contracts.

Issues Found – Code Review / Manual Testing

High Severity Issues

No issues were found.

Medium Severity Issues

No issues were found.

Low Severity Issues

A.1 Local Variable Shadow

Line	Code
278	ERC20.allowance.owner (local variable @ AvaxLauncher.sol#278) shadows: - Owned.owner (state variable @ AvaxLauncher.sol#6) ERC20._approve.owner (local variable @ AvaxLauncher.sol#422) shadows: - Owned.owner (state variable @ AvaxLauncher.sol#6)

- Check: shadowing-local
- Severity: Low
- Confidence: High

Description
Detection of shadowing using local variables.

Remediation
Rename variable

Status: Acknowledged by the Auditee.

Informational Issues

A.2 Public function that could be declared external

Description

The following public functions that are never called by the contract should be declared external to save gas:

- IERC20.totalSupply (AvaxLauncher.sol#66) should be declared external
- ERC20.totalSupply (AvaxLauncher.sol#263-265) should be declared external
- IERC20.balanceOf (AvaxLauncher.sol#71) should be declared external
- ERC20.balanceOf (AvaxLauncher.sol#270-272) should be declared external
- ERC20.allowance (AvaxLauncher.sol#278-280) should be declared external
- IERC20.allowance (AvaxLauncher.sol#89) should be declared external
- ERC20.approve (AvaxLauncher.sol#289-292) should be declared external
- IERC20.approve (AvaxLauncher.sol#105) should be declared external
- ERC20.increaseAllowance (AvaxLauncher.sol#307-310) should be declared external
- ERC20.decreaseAllowance (AvaxLauncher.sol#326-329) should be declared external
- AvaxLauncher.transfer (AvaxLauncher.sol#456-464) should be declared external
- IERC20.transfer (AvaxLauncher.sol#80) should be declared external
- AvaxLauncher.transferFrom (AvaxLauncher.sol#478-483) should be declared external
- IERC20.transferFrom (AvaxLauncher.sol#116) should be declared external
- AvaxLauncher.burn (AvaxLauncher.sol#485-489) should be declared external

Remediation

Use the external attribute for functions that are never called from the contract.

Status: Closed

Functional Tests

Function Name()	Technical Result	Logical Result	Overall Result
Read Functions()			
allowance	Pass	Pass	Pass
balanceOf	Pass	Pass	Pass
decimal	Pass	Pass	Pass
name	Pass	Pass	Pass
owner	Pass	Pass	Pass
symbol	Pass	Pass	Pass
totalSupply	Pass	Pass	Pass
Read Functions()			
approve	Pass	Pass	Pass
burn	Pass	Pass	Pass
transferOwnership	Pass	Pass	Pass
mint	Pass	Pass	Pass
transfer	Pass	Pass	Pass
transferFrom	Pass	Pass	Pass

Unit Tests

- Should correctly initialize constructor values of AVXL Token Contract (65ms)
- Should check a name of a token
- Should check a symbol of a token
- Should check a decimal of a token
- Should check a owner of a token
- Should check a balance of a token contract
- Should check a balance of a owner
- Should check the total supply of a AVXL token contract
- Should check a balance of a token contract
- Should check if contract is paused or not
- Should Not be able to pause the contract by non owner account (60ms)
- Should be able to pause the contract (70ms)
- Should check if contract is paused or not after pause
- Should Not be able to transfer AVXL token when contract is paused (57ms)
- Should Not be able to unpause the contract by non owner account (45ms)
- Should be able to unpause the contract from Owner Account (43ms)
- Should check if contract is paused or not after unpaused
- Should check approval by accounts 9 to accounts 1 to spend tokens on the behalf of accounts 0
- Should Approve accounts[1] to spend specific tokens of accounts[0]
- Should check approval by accounts 0 to accounts 1 to spend tokens on the behalf of accounts 4 (52ms)
- Should increase Approve accounts[0] to spend specific tokens of accounts[1]
- Should check approval by accounts 0 to accounts 1 to spend tokens on the behalf of accounts 0 (54ms)
- Should decrease Approve accounts[0] to spend specific tokens of accounts[1]

- ✓ Should check approval by accounts 0 to accounts 1 to spend tokens on the behalf of accounts 0 (51ms)
- ✓ Should check a owner of a token before transferring ownership
- ✓ Should not be able to transfer ownership by non ownner account (46ms)
- ✓ Should be able to transfer ownership before (42ms)
- ✓ Should be able to accept transfer ownership before (50ms)
- ✓ Should check a owner of a token after transferring ownership
- ✓ Should be able to transfer ownership again to accounts[0] (43ms)
- ✓ Should be able to accept transfer ownership before (41ms)
- ✓ Should check a owner of a token after transferring ownership
- ✓ Should check a balance of a owner before transferring tokens
- ✓ Should be able to transfer AVXL token when contract is not paused (59ms)
- ✓ Should check a balance of a owner after sending tokens
- ✓ Should check a balance of a receiver after sending tokens
- ✓ Should check the total supply of a AVXL token contract before owner burn tokens
- ✓ Should check a balance of a owner before burning tokens
- ✓ Should be able to burn AVXL token when contract is not paused (47ms)
- ✓ Should check the total supply of a AVXL token contract after owner burn tokens
- ✓ Should check a balance of a owner after burning tokens
- ✓ Should check a balance of a account[3] before burning token
- ✓ Should Not be able to burn tokens when user doesnt have tokens (53ms)

43 passing (2s)

0 Failed

Automated Tests

Slither:

```
INFO:Detectors:
Parameter '_owner' of Owned. (AvaxLauncher.sol#11) is not in mixedCase
Parameter '_newOwner' of Owned.transferOwnership (AvaxLauncher.sol#20) is not in mixedCase
Function 'ERC20._transfer' (AvaxLauncher.sol#345-352) is not in mixedCase
Function 'ERC20._mint' (AvaxLauncher.sol#363-369) is not in mixedCase
Function 'ERC20._burn' (AvaxLauncher.sol#382-388) is not in mixedCase
Function 'ERC20._transferFrom' (AvaxLauncher.sol#402-406) is not in mixedCase
Function 'ERC20._approve' (AvaxLauncher.sol#422-428) is not in mixedCase
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#conformance-to-solidity-naming-conventions
INFO:Slither:AvaxLauncher.sol analyzed (6 contracts), 25 result(s) found
```

```
INFO:Detectors:
ERC20.allowance.owner (local variable @ AvaxLauncher.sol#278) shadows:
        - Owned.owner (state variable @ AvaxLauncher.sol#6)
ERC20._approve.owner (local variable @ AvaxLauncher.sol#422) shadows:
        - Owned.owner (state variable @ AvaxLauncher.sol#6)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#local-variable-shadowing
INFO:Detectors:
ERC20.totalSupply (AvaxLauncher.sol#263-265) should be declared external
IERC20.totalSupply (AvaxLauncher.sol#66) should be declared external
IERC20.balanceOf (AvaxLauncher.sol#71) should be declared external
ERC20.balanceOf (AvaxLauncher.sol#270-272) should be declared external
IERC20.allowance (AvaxLauncher.sol#89) should be declared external
ERC20.allowance (AvaxLauncher.sol#278-280) should be declared external
IERC20.approve (AvaxLauncher.sol#105) should be declared external
ERC20.approve (AvaxLauncher.sol#289-292) should be declared external
ERC20.increaseAllowance (AvaxLauncher.sol#307-310) should be declared external
ERC20.decreaseAllowance (AvaxLauncher.sol#326-329) should be declared external
IERC20.transfer (AvaxLauncher.sol#80) should be declared external
AvaxLauncher.transfer (AvaxLauncher.sol#456-464) should be declared external
IERC20.transferFrom (AvaxLauncher.sol#116) should be declared external
AvaxLauncher.transferFrom (AvaxLauncher.sol#478-483) should be declared external
AvaxLauncher.burn (AvaxLauncher.sol#485-489) should be declared external
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#public-function-that-could-be-declared-as-external
INFO:Detectors:
Detected issues with version pragma in AvaxLauncher.sol:
        - pragma solidity0.5.16 (AvaxLauncher.sol#1): it allows old versions
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#incorrect-version-of-solidity
INFO:Detectors:
Parameter '_owner' of Owned. (AvaxLauncher.sol#11) is not in mixedCase
Parameter '_newOwner' of Owned.transferOwnership (AvaxLauncher.sol#20) is not in mixedCase
Function 'ERC20._transfer' (AvaxLauncher.sol#345-352) is not in mixedCase
Function 'ERC20._mint' (AvaxLauncher.sol#363-369) is not in mixedCase
Function 'ERC20._burn' (AvaxLauncher.sol#382-388) is not in mixedCase
Function 'ERC20._transferFrom' (AvaxLauncher.sol#402-406) is not in mixedCase
Function 'ERC20._approve' (AvaxLauncher.sol#422-428) is not in mixedCase
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#conformance-to-solidity-naming-conventions
```

Result:

No major issues were found. Some false positive errors were reported by the tool. All the other issues have been categorized above according to their level of severity.

Surya:

```
+ [Lib] SafeMath
    - [Int] add
    - [Int] sub
    - [Int] mul
    - [Int] div

+  ERC20 (IERC20, Pausable)
    - [Pub] totalSupply
    - [Pub] balanceOf
    - [Pub] allowance
    - [Pub] approve #
    - [Pub] increaseAllowance #
    - [Pub] decreaseAllowance #
    - [Int] _transfer #
    - [Int] _mint #
    - [Int] _burn #
    - [Int] _transferFrom #
       - modifiers: whenNotPaused
    - [Int] _approve #

+  AvaxLauncher (ERC20)
    - [Pub] <Constructor> #
       - modifiers: Owned
    - [Ext] transfer #
       - modifiers: whenNotPaused
    - [Ext] transferFrom #
       - modifiers: whenNotPaused
    - [Ext] burn #
       - modifiers: whenNotPaused


($) = payable function
# = non-constant function
```

```
+  Owned
   - [Pub] <Constructor> #
   - [Ext] transferOwnership #
      - modifiers: onlyOwner
   - [Ext] acceptOwnership #

+  Pausable (Owned)
   - [Ext] pause #
      - modifiers: onlyOwner,whenNotPaused
   - [Ext] unpause #
      - modifiers: onlyOwner,whenPaused

+ [Int] IERC20
   - [Ext] totalSupply
   - [Ext] balanceOf
   - [Ext] transfer #
   - [Ext] allowance
   - [Ext] approve #
   - [Ext] transferFrom #

+ [Lib] SafeMath
   - [Int] add
   - [Int] sub
   - [Int] mul
   - [Int] div

+  ERC20 (IERC20, Pausable)
   - [Pub] totalSupply
   - [Pub] balanceOf
   - [Pub] allowance
   - [Pub] approve #
   - [Pub] increaseAllowance #
   - [Pub] decreaseAllowance #
   - [Int] _transfer #
```

Sūrya's Description Report

Files Description Table

File Name	SHA-1 Hash
AvaxLauncher.sol	effa53faacce8b3f6a953ab6fb296f41599d71f9

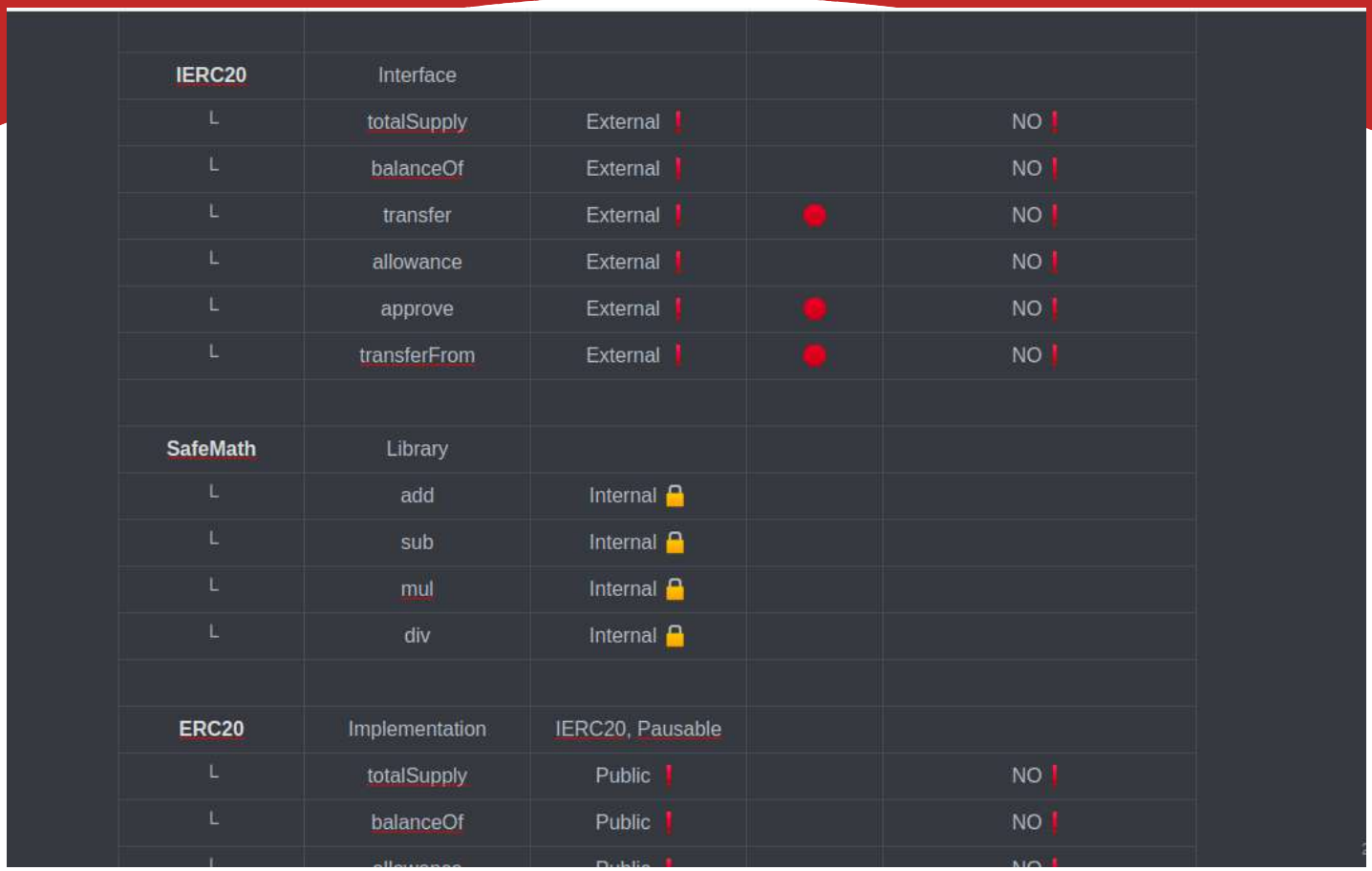
Contracts Description Table

Contract	Type	Bases		
└	Function Name	Visibility	Mutability	Modifiers
Owned	Implementation			
└	<Constructor>	Public ❗	🛑	NO ❗
└	transferOwnership	External ❗	🛑	onlyOwner
└	acceptOwnership	External ❗	🛑	NO ❗
Pausable	Implementation	Owned		
└	pause	External ❗	🛑	onlyOwner whenNotPaused
└	unpause	External ❗	🛑	onlyOwner whenPaused

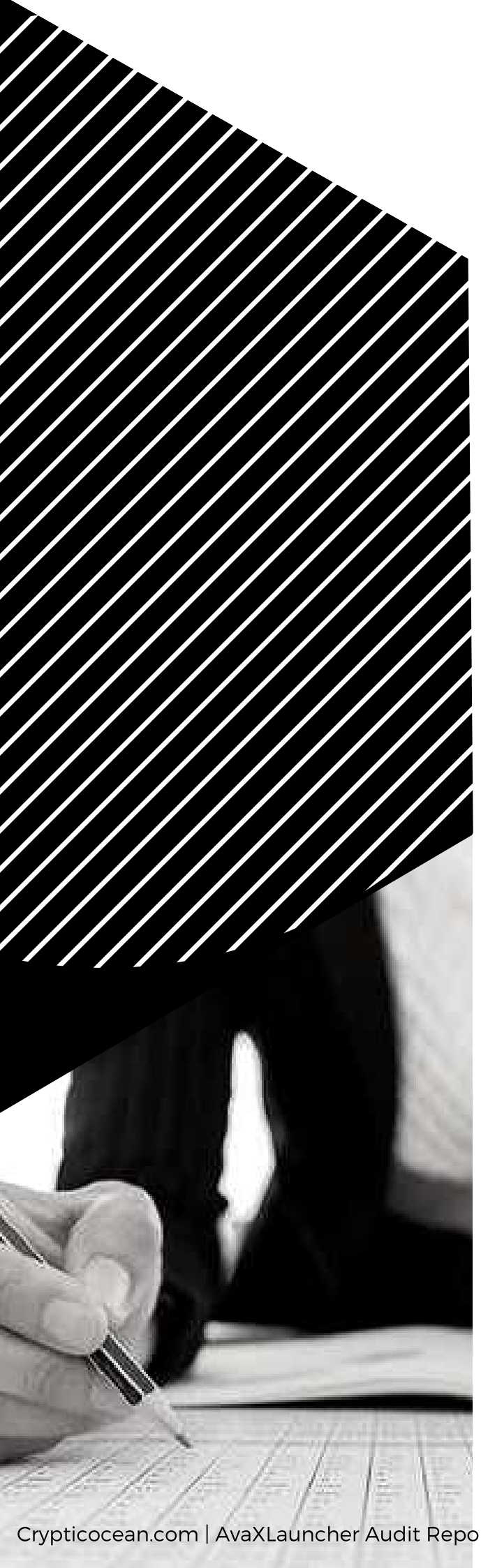
└	decreaseAllowance	Public ❗	🛑	NO ❗
└	_transfer	Internal 🔒	🛑	
└	_mint	Internal 🔒	🛑	
└	_burn	Internal 🔒	🛑	
└	_transferFrom	Internal 🔒	🛑	whenNotPaused
└	_approve	Internal 🔒	🛑	
AvaxLauncher	Implementation	ERC20		
└	<Constructor>	Public ❗	🛑	Owned
└	transfer	External ❗	🛑	whenNotPaused
└	transferFrom	External ❗	🛑	whenNotPaused
└	burn	External ❗	🛑	whenNotPaused

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable



IERC20	Interface			
L	totalSupply	External !		NO !
L	balanceOf	External !		NO !
L	transfer	External !	🔴	NO !
L	allowance	External !		NO !
L	approve	External !	🔴	NO !
L	transferFrom	External !	🔴	NO !
SafeMath	Library			
L	add	Internal 🔒		
L	sub	Internal 🔒		
L	mul	Internal 🔒		
L	div	Internal 🔒		
ERC20	Implementation	IERC20, Pausable		
L	totalSupply	Public !		NO !
L	balanceOf	Public !		NO !
L	allowance	Public !		NO !



Closing Summary

Overall, smart contracts are very well written and adhere to guidelines.

No instances of Integer Overflow and Underflow vulnerabilities or Back-Door Entry were found in the contract, but relying on other contracts might cause Reentrancy Vulnerability.

Some low severity issues were detected; it is recommended to fix them.

Disclaimer

Cryptiocean audit is not a security warranty, investment advice, or an endorsement of the AvaXLauncher platform. This audit does not provide a security or correctness guarantee of the audited smart contracts. The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them. Securing smart contracts is a multistep process. One audit cannot be considered enough. We recommend that the AvaXLauncher Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.