LP Staking AvaXlauncher

Smart Contract Audit (Final Report)

23rd September 2021

For: AvaXLauncher
Prepared By: Entersoft Pte Ltd of 1B Trengganu Street, Singapore 058455

1

23rd September 2021 | Smart Contract Audit Report for AvaXLauncher
Entersoft Pte Ltd | 1B Terengganu Street, Singapore 058455

Contents

Revision History and Version Control

Version	Date	Author(s)	Description
1.0	September 23rd,2021	ES Auditors	Initial Draft of Final Report
1.0	September 23rd,2021	Jake Lemke	Reviewed
1.0	September 23rd,2021	Paul Kang	Released Final Report

Entersoft was commissioned by AvaXlauncher to perform source code review on their solidity smart contract. The review was conducted between September 20th 2021 to September 23nd 2021. The report is organized into the following sections.

- Executive Summary: A high-level overview of the security audit findings.
- Technical analysis: Our detailed analysis of the Smart Contract code

The information in this report should be used to understand overall code quality, security, correctness, and meaning that code will work as AvaxLauncher described in the smart contract.

1.0   Disclaimer

This is a limited audit report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to: (i) smart contract best coding practices and issues in the framework and algorithms based on white paper, code, the details of which are set out in this report, (Smart Contract audit). To get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us based on what it says or does not say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full. DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Entersoft Australia and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers, and other representatives) (Entersoft) owe no duty of care towards you or any other person, nor does Entersoft make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Entersoft hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Entersoft hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Entersoft, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the Smart contract is purely based on the smart contract code shared with us alone.

4

23rd September 2021 | Smart Contract Audit Report for AvaXLauncher
Entersoft Pte Ltd | 1B Terengganu Street, Singapore 058455

2.0 Overview

2.1 Project Overview

During the period of **September 20, 2021 to September 22, 2021** - Entersoft performed security audits for **AvaXLauncher (AVXL)** smart contracts.

2.2 Scope

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

OUT-OF-SCOPE: External contracts, External Oracles, other smart contracts in the repository or imported smart contracts.

2.3 Project Summary

Project Name	AvaXlauncher
Platform	Avalanche-Avax
Codebase	https://bscscan.com/token/0xbd29490383edfd560426c3b63d01534408bc2da6
Token Name	AVXL
Contract Name(s)	AvaXlauncher
Contract Address	https://bscscan.com/address/0xcc5bd209c5202254d826061f4e13e2c9f57fa911
Verified	Yes
Audited	Yes
Vulnerabilities / Issues	Below

2.4 Audit Summary

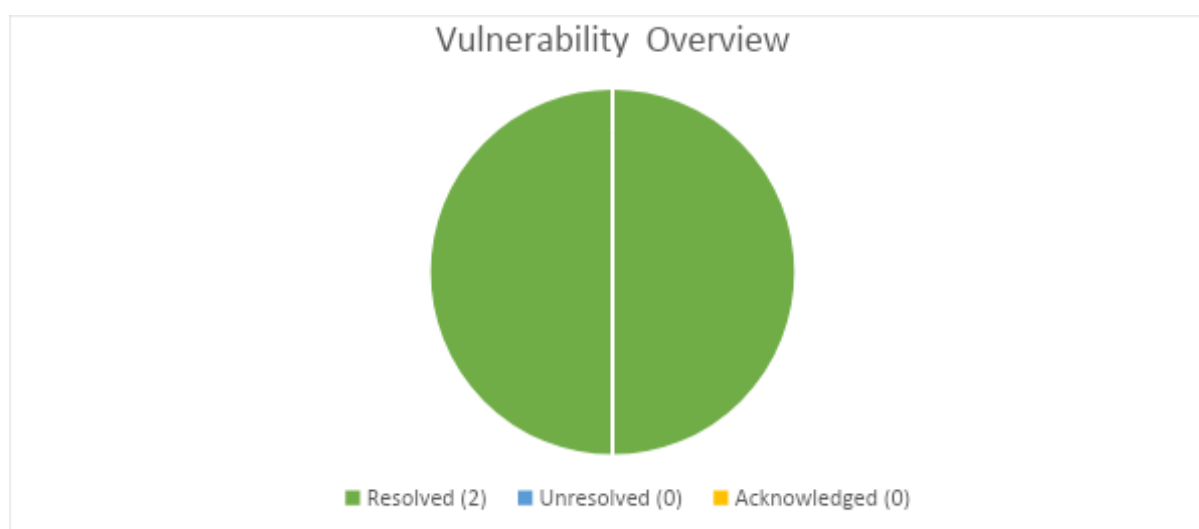|	
Delivery Date	23rd September 2021
Method of Audit	
Consultants Engaged	1

2.5 Security Level references

Every issue in this report was assigned a severity level from the following classification table:
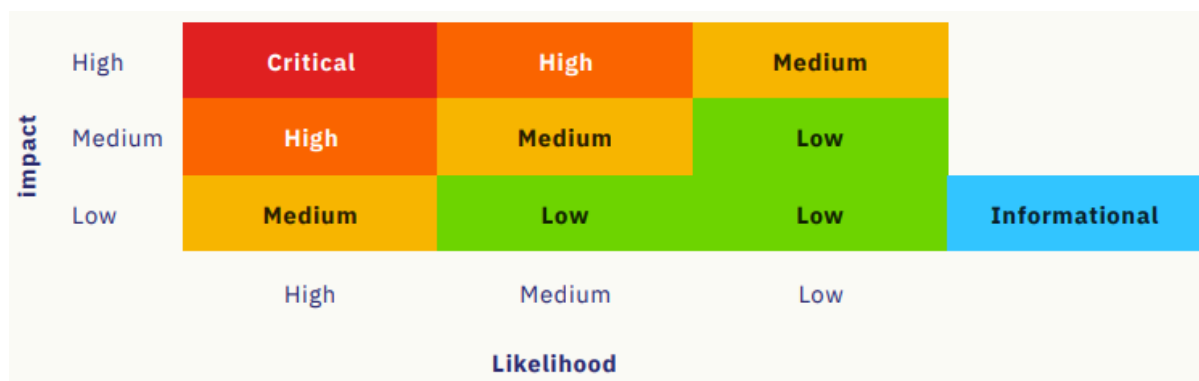
2.6 Vulnerability Summary

Total Critical	0
Total High	0
Total Medium	0
Total Low	2
Total Informational	2

2.7 Audit Results Overview

Audit Item	Audit Subclass	Audit Result
Overflow	-	Passed
Race Conditions	-	Passed
Permissions	Permission Vulnerability Audit	Passed
	Excessive Auditing Authority	
Safety Design	Zeppelin Safe Math	Passed
DDOS Attack	Call Function Security	Passed
Gas Optimization	-	Passed
Design Logic	-	Passed
Know Attacks	-	Passed
Overall Audit Result	-	Passed

3.0 Executive Summary

3.1 Files in Scope

3.2 Findings

ID	Title	Severity	Resolved
AXVL-001	Public Function Could be declared external	Informational	**RESOLVED**
AXVL-002	Reward Calculation safemath	Informational	**RESOLVED**

3.3 Comments

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.

4.0 Vulnerabilities

4.1 Public function that could be declared external

Severity	Confidence	Status
Informational	High	Resolved

Description

The following public functions that are never called by the contract should be

declared external to save gas:

- BEP20.balanceOf (LpStaking.sol#6) should be declared external
- StakeLPAvxl.contractStats (LpStaking.sol#201-206) should be declared external
- StakeLPAvxl.userTimeStats (LpStaking.sol#209-214) should be declared external
- StakeLPAvxl.userStats (LpStaking.sol#217-235) should be declared external
- StakeLPAvxl.stakeLpTokens (LpStaking.sol#238-256) should be declared external
- StakeLPAvxl.startPoolForThirtyDays (LpStaking.sol#258-264) should be declared
- external
- StakeLPAvxl.unStake (LpStaking.sol#267-287) should be declared external
- StakeLPAvxl.claimLP (LpStaking.sol#290-300) should be declared external
- StakeLPAvxl.claimReward (LpStaking.sol#303-317) should be declared external

Remediation

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

4.2 Reward Calculation Safemath

Severity	Confidence	Status
Informational	High	Resolved

Description

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

Remediation

Use safemath for every operation

9

23rd September 2021 | Smart Contract Audit Report for AvaXLauncher
Entersoft Pte Ltd | 1B Terengganu Street, Singapore 058455

5.0 AvaXLauncher Functional List

The following is the list of functions tested and checked for vulnerabilities during audit:

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
Write 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

6.0 Unit Tests

✔ Should correctly initialize constructor values of AvaxLancher token contract (116ms)
✔ Should correctly initialize constructor values of LP Token Contract (216ms)
✔ Should correctly initialize constructor values of Staking contract (202ms)
✔ Should check LP token Contract address
✔ Should check avxl Contract address
✔ Should check stake contract stats
✔ Should check avxl total staked (64ms)
✔ Should check avxl no of stakers
✔ Should check total pool rewards available
✔ Should check total rewards Distributed
✔ Should check lp Withdraw Successfully
✔ Should check Staking Open or not
✔ Should check Staking pool start or not
✔ Should Not be able to pause the contract by non owner account (94ms)
✔ Should be able to pause the contract (78ms)
✔ Should check if contract is paused or not after pause
✔ Should Not be able to unpause the contract by non owner account (49ms)
✔ Should be able to unpause the contract from Owner Account (51ms)
✔ Should be able to transfer LP to Accounts[1] contract (71ms)
✔ Should check a AVXL balance of a Contract address LP Staking
✔ Should be able to transfer AVXL to staking contract that will be rewarded (60ms)
✔ Should check a AVXL balance of a Contract address LP Staking after
✔ Should check approval by accounts 0 to Staking contract to spend tokens on the
✔ behalf of staking (38ms)
✔ Should Approve staking to spend specific tokens of accounts[1] (47ms)
✔ Should check approval by accounts 0 to Staking contract to spend tokens on the
✔ behalf of staking after
✔ should Stake 100 LP tokens by account[1] (63ms)
✔ Should check a AVXL balance of a account[1] after stake
✔ Should check stake contract stats after staking
✔ Should check avxl total staked after staked 1st time (38ms)
✔ Should check avxl no of stakers after 1st stake
✔ Should check total pool rewards available after 1st stake
✔ Should check total rewards Distributed before claiming
✔ Should check user stats after user staked tokens (38ms)
✔ Should UnStake 100 LP tokens before vesting time start
✔ Should check a AVXL balance of a account after unstake (204ms)
✔ Should check user stats after unstaked
✔ Should check stake contract stats after unstaked by user 1 (53ms)

- ✔ Should be able to transfer LP to Accounts[1] contract again (183ms)
- ✔ Should check a AVXL balance of a account
- ✔ Should check approval by accounts 0 to Staking contract to spend tokens on the
- ✔ behalf of staking (53ms)
- ✔ Should Approve Stake to spend specific tokens of accounts[1]
- ✔ Should check approval by accounts 0 to Staking contract to spend tokens on the
- ✔ behalf of staking after (120ms)
- ✔ Should check if staking open or not before staking (66ms)
- ✔ Should Stake 100 LP tokens again after unstake when time not started yet (215ms)
- ✔ Should check a AVXL balance of a account after unstake (41ms)
- ✔ Should check stake contract stats (61ms)
- ✔ Should check avxl total staked (55ms)
- ✔ Should check avxl no of stakers (53ms)
- ✔ Should check total pool rewards available (66ms)
- ✔ Should check total total rewards Distributed (69ms)
- ✔ Should check user stats (43ms)
- ✔ Should UnStake 100 LP tokens (144ms)
- ✔ Should check a LP balance of a account after unstake (59ms)
- ✔ Should check user stats (55ms)
- ✔ Should check stake contract stats (53ms)
- ✔ Should be able to transfer LP to Accounts[4] contract (131ms)
- ✔ Should be able to transfer LP to Accounts[2] contract (76ms)
- ✔ Should be able to transfer LP to Accounts[3] contract (198ms)
- ✔ Should check a LP balance of account 2 (38ms)
- ✔ Should check a LP balance of account 3 (142ms)
- ✔ Should check a LP balance of account 4 (63ms)
- ✔ Should check Staking Open or not before staking (90ms)
- ✔ Should check started or not staking pool (69ms)
- ✔ Should check started or not staking pool (77ms)
- ✔ Should check user stats
- ✔ Should check stake contract time stats of user (127ms)
- ✔ Should check stake contract stats (155ms)
- ✔ Should not be able to invest if you dont have LP tokens (344ms)
- ✔ Should not be able to invest if you dont allow lp to withdraw (438ms)
- ✔ Should check a AVXL balance of a Contract address LP Staking after (77ms)
- ✔ Should check approval by accounts 2 to Staking contract to spend tokens on the
- ✔ behalf of staking (92ms)
- ✔ Should Approve staking to spend specific tokens of accounts[1] (124ms)
- ✔ Should check approval by accounts 0 to Staking contract to spend tokens on the
- ✔ behalf of staking after (46ms)
- ✔ Should check if staking open or not before staking (50ms)
- ✔ Should Stake 200 LP tokens (227ms)
- ✔ Should check user stats after account 2 staked tokens (113ms)
- ✔ Should Not be able to stake again (160ms)

- ✔ Should check a LP balance of a account after unstake (71ms)
- ✔ Should check a LP balance of a contract staking after stake
- ✔ Should check stake contract stats (49ms)
- ✔ Should check Staking Open or not before staking by account 3
- ✔ Should check started or not staking pool before staking by account 3 (57ms)
- ✔ Should check started or not staking pool by account 3 (42ms)
- ✔ Should check a AVXL balance of a Contract address LP Staking after (58ms)
- ✔ Should check approval by accounts 3 to Staking contract to spend tokens on the
- ✔ behalf of staking (46ms)
- ✔ Should Approve staking to spend specific tokens of accounts[3] (231ms)
- ✔ Should check approval by accounts 0 to Staking contract to spend tokens on the
- ✔ behalf of staking after (42ms)
- ✔ Should check if staking open or not before staking (117ms)
- ✔ Should Stake 200 LP tokens by account 3 (128ms)
- ✔ Should check stake contract stats after 3rd stake (110ms)
- ✔ Should check user stats after account 3 staked tokens (40ms)
- ✔ Should Not be able to stake again accounts[3] (146ms)
- ✔ Should check Staking Open or not before closing
- ✔ Should check Staking pool start or not before starting (59ms)
- ✔ Should not be able to close the staking and start pool time by non owner (71ms)
- ✔ Should be able to close the staking and start pool time by owner (95ms)
- ✔ Should check Staking Open or not before closing
- ✔ Should check Staking pool start or not before starting
- ✔ Should check approval by accounts 4 to Staking contract to spend tokens on the
- ✔ behalf of staking (38ms)
- ✔ Should Approve staking to spend specific tokens of accounts[4] (102ms)
- ✔ Should check approval by accounts 0 to Staking contract to spend tokens on the
- ✔ behalf of staking after
- ✔ Should check if staking open or not before staking
- ✔ Should Stake 200 LP tokens by account 4 (149ms)
- ✔ Should check user stats after user staked tokens for accounts 2
- ✔ Should not be able to claim LP tokens by account 4 (86ms)
- ✔ Should not be able to claim Rewards tokens by account 2 (65ms)
- ✔ Should be able to increase time to get 30 days
- ✔ Should check a LP balance of a staking contract
- ✔ Should check a LP balance of a account LP claim
- ✔ should be able to claim LP tokens by account 2 (61ms)
- ✔ Should check a LP balance of a account after LP claim
- ✔ should not be able to claim LP tokens by account 2 again (49ms)
- ✔ Should check a LP balance of a staking contract
- ✔ should not be able to claim Rewards tokens by account 2 (77ms)
- ✔ Should check stake contract stats after 3rd stake
- ✔ Should check user stats after user staked tokens for accounts 2 after claim
- ✔ Should check a LP balance of a staking contract (39ms)

- ✔ Should check a LP balance of accounts 3 LP claim
- ✔ Should be able to claim LP tokens by account 3 (95ms)
- ✔ Should check a LP balance of accounts after LP claim (38ms)
- ✔ should not be able to claim LP tokens by account 3 again (59ms)
- ✔ Should check a LP balance of a staking contract
- ✔ should not be able to claim Rewards tokens by account 3 (86ms)
- ✔ Should check stake contract stats after 3rd stake
- ✔ Should check user stats after user staked tokens for accounts 3 after claim

117 passing (10s)

0 Failed

14

23rd September 2021 | Smart Contract Audit Report for AvaXLauncher
Entersoft Pte Ltd | 1B Terengganu Street, Singapore 058455

7.0 Automated Testing

Automated testing is carried out with the following tools:
- Slither
- Surya

7.1 Slither

Slither is an open-source Solidity static analysis framework. This tool provides rich information about Ethereum smart contracts and has the critical properties. It runs a suite of vulnerability detectors, prints visual information about contract details, and provides an API to easily write custom analyses.

15

23rd September 2021 | Smart Contract Audit Report for AvaXLauncher
Entersoft Pte Ltd | 1B Terengganu Street, Singapore 058455

```
StakeLPAvxl.claimLP (LpStaking.sol#290-300) uses a dangerous strict equality:
        - require(bool,string)(withdrawLpTime[msg.sender] == 0,Withdrawn Lp already)
StakeLPAvxl.claimLP (LpStaking.sol#290-300) uses a dangerous strict equality:
        - require(bool,string)(withdrawLpTime[msg.sender] == 0,Withdrawn Lp already)
StakeLPAvxl.claimReward (LpStaking.sol#303-317) uses a dangerous strict equality:
        - require(bool,string)(withdrawRewardTime[msg.sender] == 0,already withdrawn rewads)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#dangerous-strict-equalities
INFO:Detectors:
Reentrancy in StakeLPAvxl.stakeLpTokens (LpStaking.sol#238-256):
        External calls:
        - require(bool,string)(BEP28(lpContract).balanceOf(msg.sender) >= amount,balance of a user is less then value) (LpStaking.sol#241)
        State variables written after the call(s):
        - userStaking (LpStaking.sol#245)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#reentrancy-vulnerabilities-1
INFO:Detectors:
Reentrancy in StakeLPAvxl.stakeLpTokens (LpStaking.sol#238-256):
        External calls:
        - require(bool,string)(BEP28(lpContract).balanceOf(msg.sender) >= amount,balance of a user is less then value) (LpStaking.sol#241)
        State variables written after the call(s):
        - stakedOn (LpStaking.sol#246)
        - totalStaked (LpStaking.sol#247)
        - totalStakers (LpStaking.sol#248)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
BEP20.balanceOf (LpStaking.sol#6) should be declared external
StakeLPAvxl.contractStats (LpStaking.sol#201-206) should be declared external
StakeLPAvxl.userTimeStats (LpStaking.sol#209-214) should be declared external
StakeLPAvxl.userStats (LpStaking.sol#217-235) should be declared external
StakeLPAvxl.stakeLpTokens (LpStaking.sol#238-256) should be declared external
StakeLPAvxl.startPoolForThirtyDays (LpStaking.sol#258-264) should be declared external
StakeLPAvxl.unStake (LpStaking.sol#267-287) should be declared external
StakeLPAvxl.claimLP (LpStaking.sol#290-300) should be declared external
StakeLPAvxl.claimReward (LpStaking.sol#303-317) should be declared external
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#public-function-that-could-be-declared-as-external
```

Results

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.

7.2 Surya

```
    - [Ext] acceptOwnership #

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

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

+  StakeLPAvxl (Pausable)
    - [Pub] <Constructor> #
        - modifiers: Owned
    - [Pub] contractStats
    - [Pub] userTimeStats
    - [Pub] userStats
    - [Pub] stakeLpTokens #
        - modifiers: whenNotPaused
    - [Pub] startPoolForThirtyDays #
        - modifiers: onlyOwner,whenNotPaused
    - [Pub] unStake #
        - modifiers: whenNotPaused
    - [Pub] claimLP #
        - modifiers: whenNotPaused
    - [Pub] claimReward #
        - modifiers: whenNotPaused


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

17

23rd September 2021 | Smart Contract Audit Report for AvaXLauncher
Entersoft Pte Ltd | 1B Terengganu Street, Singapore 058455

```
+  BEP20
   - [Ext] transferFrom #
   - [Pub] balanceOf
   - [Ext] allowance
   - [Ext] transfer #
   - [Ext] burn #

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

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

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

+  StakeLPAvxl (Pausable)
   - [Pub] <Constructor> #
      - modifiers: Owned
   - [Pub] contractStats
   - [Pub] userTimeStats
   - [Pub] userStats
   - [Pub] stakeLpTokens #
      - modifiers: whenNotPaused
   - [Pub] startPoolForThirtyDays #
      - modifiers: onlyOwner,whenNotPaused
```

	div	Internal 🔒		
StakeLPAvxl	Implementation	Pausable		
L	<Constructor>	Public ❗	🔴	Owned
L	contractStats	Public ❗		NO ❗
L	userTimeStats	Public ❗		NO ❗
L	userStats	Public ❗		NO ❗
L	stakeLpTokens	Public ❗	🔴	whenNotPaused
L	startPoolForThirtyDays	Public ❗	🔴	onlyOwner whenNotPaused
L	unStake	Public ❗	🔴	whenNotPaused
L	claimLP	Public ❗	🔴	whenNotPaused
L	claimReward	Public ❗	🔴	whenNotPaused

Legend

Symbol	Meaning
🔴	Function can modify state
🟢	Function is payable

Owned	Implementation			
L	<Constructor>	Public ❗	🔴	NO ❗
L	transferOwnership	External ❗	🔴	onlyOwner
L	acceptOwnership	External ❗	🔴	NO ❗
Pausable	Implementation	Owned		
L	pause	External ❗	🔴	onlyOwner whenNotPaused
L	unpause	External ❗	🔴	onlyOwner whenPaused
SafeMath	Library			
L	add	Internal 🔒		
L	sub	Internal 🔒		
L	mul	Internal 🔒		
L	div	Internal 🔒		
StakeLPAvxl	Implementation	Pausable		
L	<Constructor>	Public ❗	🔴	Owned
L	contractStats	Public ❗		NO ❗

Sūrya's Description Report

Files Description Table

File Name	SHA-1 Hash
LpStaking.sol	225138c0c5c52692bb24545269c587d93ae3a879

Contracts Description Table

Contract	Type	Bases		
└	Function Name	Visibility	Mutability	Modifiers
BEP20	Implementation			
└	transferFrom	External ❗	🔴	NO ❗
└	balanceOf	Public ❗		NO ❗
└	allowance	External ❗		NO ❗
└	transfer	External ❗	🔴	NO ❗
└	burn	External ❗	🔴	NO ❗
Owned	Implementation			

20

23rd September 2021 | Smart Contract Audit Report for AvaXLauncher
Entersoft Pte Ltd | 1B Terengganu Street, Singapore 058455

8.0 Auditing Approach and Methodologies applied

Throughout the audit of **AvaXLauncher** smart contract care was taken to ensure:

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

A combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy regarding the scope of the smart contract audit. While manual testing is recommended to uncover flaws in logic, process, and implementation; automated testing techniques help enhance coverage of smart contracts and can quickly identify items that do not follow security best practices. The following phases and associated tools were used throughout the term of the audit:

8.1 Structural Analysis

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

8.2 Static Analysis

Static Analysis of smart contracts was done to identify contract vulnerabilities. In this step series of automated tools are used to test security of smart contracts.

8.3 Code Review / Manual Analysis

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

8.4 Gas Consumption

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

8.5 Tools and Platforms used for Audit

VSCode, Remix IDE, Truffle, Truffle Team, Ganache, Solhint, Mythril, Manticore, Slither.

8.6 Checked Vulnerabilities

We have scanned The People Reserves smart contracts for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that we considered:

- Re-entrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with (Unexpected) Throw
- DoS with (Unexpected) revert
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Use of tx.origin
- Exception disorder
- Gasless send
- Balance equality
- Byte array
- Transfer forwards all gas
- ERC20 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
- Address hardcoded
- Using delete for arrays
- Integer overflow/underflow
- Locked money
- Private modifier
- Revert/require functions
- Using var
- Visibility
- Using blockhash
- Using SHA3
- Using suicide
- Using throw
- Using inline assembly

9.0 Limitations on Disclosure and Use of this Report

This report contains information concerning potential details of AvaXLauncher and methods for exploiting them. Entersoft recommends that special precautions be taken to protect the confidentiality of both this document and the information contained herein. Security Assessment is an uncertain process, based on past experiences, currently available information, and known threats. All information security systems, which by their nature are dependent on human beings, are vulnerable to some degree. Therefore, while Entersoft considers the major security vulnerabilities of the analyzed systems to have been identified, there can be no assurance that any exercise of this nature will identify all possible vulnerabilities or propose exhaustive and operationally viable recommendations to mitigate those exposures. In addition, the analysis set forth herein is based on the technologies and known threats as of the date of this report. As technologies and risks change over time, the vulnerabilities associated with the operation of the AvaXLauncher Smart Contract described in this report, as well as the actions necessary to reduce the exposure to such vulnerabilities will also change. Entersoft makes no undertaking to supplement or update this report based on changed circumstances or facts of which Entersoft becomes aware after the date hereof, absent a specific written agreement to perform the supplemental or updated analysis. This report may recommend that Entersoft use certain software or hardware products manufactured or maintained by other vendors. Entersoft bases these recommendations upon its prior experience with the capabilities of those products. Nonetheless, Entersoft does not and cannot warrant that a particular product will work as advertised by the vendor, nor that it will operate in the manner intended. This report was prepared by Entersoft for the exclusive benefit of AvaXLauncher and is proprietary information. The Non-Disclosure Agreement (NDA) in effect between Entersoft and AvaXLauncher govern the disclosure of this report to all other parties including product vendors and suppliers.