LP Staking AvaXLauncher Smart Contract Final Audit Report

SEP 2021

CONTENT

SCOPE OF AUDIT

The scope of this audit was to analyze and document the AvaXLauncher Lp-Stake 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 behavior.
- Token distribution and calculations are as per the intended behavior 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

Every issue in this report has been assigned to a severity level. There are four levels of severity, and each of them has been explained below.

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	0	2
Acknowledged	0	0	0	2
Closed	0	0	0	2

Introduction

During the period of September 12, 2021 to September 16, 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	No issues were found

Informational Issues

A.1 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:

- 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.
Status: Closed

A.2 Reward Calculation Safemath

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

Remediation
Use safemath for every operation
Status: Closed

FUNCTIONAL TESTS

Function Name()	Technical Result	Logical Result	Overall Result
Read Functions()			
contractStats	Pass	Pass	Pass
userStats	Pass	Pass	Pass
userTimeStats	Pass	Pass	Pass
totalClaimed	Pass	Pass	Pass
totalRewards	Pass	Pass	Pass
rewardStats	Pass	Pass	Pass
lpStats	Pass	Pass	Pass
Write Functions()			
stake	Pass	Pass	Pass
unStake	Pass	Pass	Pass
claimLp	Pass	Pass	Pass
claimRewards	Pass	Pass	Pass
startPool	Pass	Pass	Pass

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

AUTOMATED TESTS

Slither:

```
        - 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
INFO:Detectors:
Detected issues with version pragma in LpStaking.sol:
        - pragma solidity0.5.16 (LpStaking.sol#1): it allows old versions
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#incorrect-version-of-solidity
INFO:Detectors:
Parameter '' of BEP20.transferFrom (LpStaking.sol#5) is not in mixedCase
Parameter '_scope_0' of BEP20.transferFrom (LpStaking.sol#5) is not in mixedCase
Parameter '_scope_1' of BEP20.transferFrom (LpStaking.sol#5) is not in mixedCase
Parameter '' of BEP20.balanceOf (LpStaking.sol#6) is not in mixedCase
Parameter '' of BEP20.allowance (LpStaking.sol#7) is not in mixedCase
Parameter '_scope_0' of BEP20.allowance (LpStaking.sol#7) is not in mixedCase
Parameter '' of BEP20.transfer (LpStaking.sol#8) is not in mixedCase
Parameter '_scope_0' of BEP20.transfer (LpStaking.sol#8) is not in mixedCase
Parameter '' of BEP20.burn (LpStaking.sol#9) is not in mixedCase
Parameter '_owner' of Owned. (LpStaking.sol#21) is not in mixedCase
Parameter '_newOwner' of Owned.transferOwnership (LpStaking.sol#30) is not in mixedCase
Parameter '_lpContract' of StakeLPAvxl. (LpStaking.sol#188) is not in mixedCase
Parameter '_avxlContract' of StakeLPAvxl. (LpStaking.sol#188) is not in mixedCase
Variable 'StakeLPAvxl.StakingOpen' (LpStaking.sol#177) is not in mixedCase
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#conformance-to-solidity-naming-conventions
```

```
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)(BEP20(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)(BEP20(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
```

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:

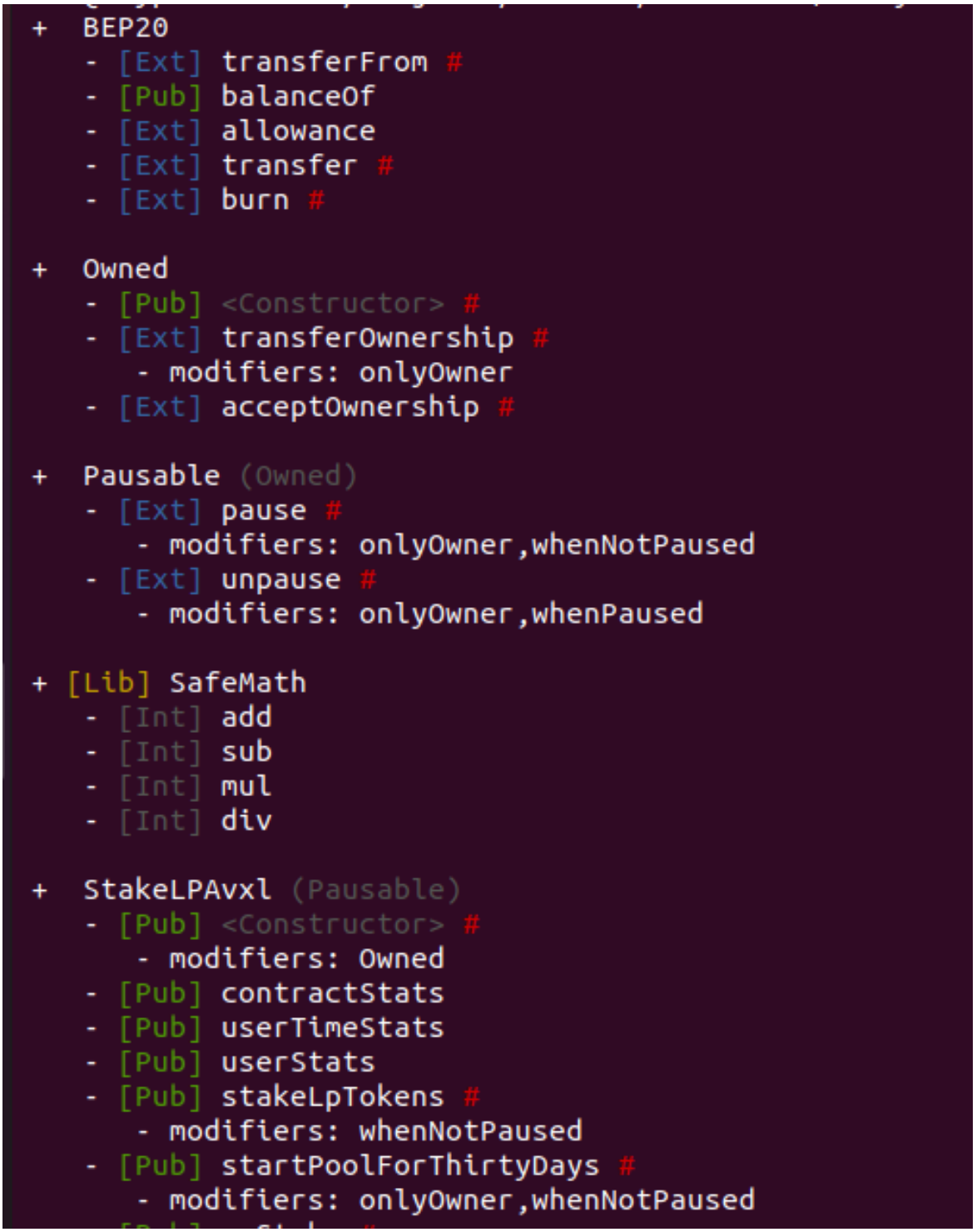
```
    - [Ext] acceptOwnership #

+   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
```


+ 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		
└	<Constructor>	Public ❗	🔴	Owned
└	contractStats	Public ❗		NO ❗
└	userTimeStats	Public ❗		NO ❗
└	userStats	Public ❗		NO ❗
└	stakeLpTokens	Public ❗	🔴	whenNotPaused
└	startPoolForThirtyDays	Public ❗	🔴	onlyOwner whenNotPaused
└	unStake	Public ❗	🔴	whenNotPaused
└	claimLP	Public ❗	🔴	whenNotPaused
└	claimReward	Public ❗	🔴	whenNotPaused

Legend

Symbol	Meaning
🔴	Function can modify state
💵	Function is payable

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

Sūrya's Description Report

Files Description Table

File Name	SHA-1 Hash
LpStaking.sol	225138c0c5c52692bb24545269c587d93ae3a879

Contracts Description Table

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

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.