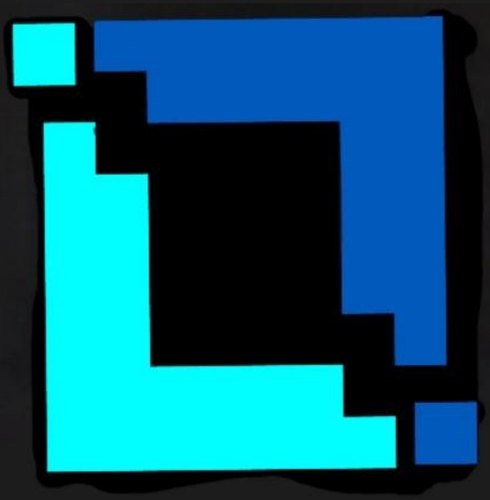
SEPTEMBER2021

AUDITEK

Financial Audit Report

Contract Security Audit

LITUNI



PRESENTED BY

Davis Thorne and Partners

Audit Details

Audited project

LITUNI

Deployer address

0x8d9bb0226aeb57366b380aea27c362c3446063ce

Client contacts:

LITUNI TEAM

Blockchain

Binance Smart Chain

Project website:

http://lituni.social

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order 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 on the basis of what it says or doesn't 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 and 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 Auditek and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Auditek) owe no duty of care towards you or any other person, nor does Auditek 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 Auditek 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, Auditek hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Auditek, 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 security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Background

Auditek was commissioned by LITUNI to perform an audit of smart contracts:
https://bscscan.com/token/0x8d9bb0226aeb57366b380aea27c362c3446063ce

The purpose of the audit was to achieve the following:

- **Ensure that the smart contract functions as intended.**
- **Identify potential security issues with the smart contract.**

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts Details

Token contract details for 27.08.2021

Contract name	LITUNI
Contract address	0x8d9bb0226aeb57366b380aea27c362c3446063ce
Total supply	20,000,000
Token ticker	LITO
Decimals	9
Token holders	5
Transactions count	5
Top 100 holders dominance	100.00%
Liquidity fee	9
Tax fee	2
Total fees	0
Compiler Version	v0.8.4+commit.c7e474f2
Contract deployer address	0x064703eb03725190a9E5f9C35A2e820A83b0113b
Contract's current owner address	0x064703eb03725190a9E5f9C35A2e820A83b0113b

LITO Token Distribution

LITO Contract Interaction Details

lyansactions Contract ® Events Analytics Comments

, Read Contract | Write Contract

© Search Source Code

Contract Source Code Verified (Exact Match)

Contract Name:	LITUNI	Optimization Enabled:	No with 200 runs
Compiler Version	v0.8.4+commit.c7e474f2	Other Settings:	default evmVersion, MIT license

0 Contract Source Code (Solidity)

More Options

LITO Top 10 Token Holders

Transfers | Holders | Info | Read Contract | Write Contract | Analytics | Comments

Token Holders Chart

A total of 5 token holders

First | Page 1 of 1 | Last

Rank	Address	Quantity	Percentage	Analytics
1	0x064703eb03725190a9e5f9c35a2e820a83b0113b	15,000,000	75.0000%	
2	0x4fe1fa4cf9bd8513db130c5aff179e4ae964a830	2,000,000	10.0000%	
3	0x3f6a1afe449d17c2c3c0e37f0b8f037c76e398ac	1,000,000	5.0000%	
4	0x18aafa3a8d0c46d92eda4511ff7ceeb852ab6329	1,000,000	5.0000%	
5	0x7c7d8b7d5e7a3d937b533d326dfbad6e1ce302dc	1,000,000	5.0000%	

[Download CSV Export]

Contract functions details

+ Context
  -[Int] _msgSender
  -[int] _msgData

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

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

+ [Lib] Address
  -[Int] isContract
  -[Int] sendValue #
  -[Int] functionCall #
  -[Int] functionCall #
  -[Int] functionCallWithValue #
  -[Int] functionCallWithValue #
  -[Prv] _functionCallWithValue #

+ Ownable (Context)
  -[Pub] ‹Constructor› #
  -[Pub] owner
  -[Pub] renounceOwnership #
    -modifiers: onlyOwner
  -[Pub] transferOwnership #
    -modifiers: onlyOwner
  -[Pub] getUnlockTime
  -[Pub] getTime
  -[Pub] lock #
    -modifiers: onlyOwner
  -[Pub] unlock #

+ [Int] IUniswapV2Factory
  -[Ext] feeTo
  -[Ext] feeToSetter
  -[Ext] getPair
  -[Ext] allPairs
  -[Ext] allPairsLength
  -[Ext] createPair #

- [Ext] setFeeTo #
- [Ext] setFeeToSetter #

+ [Int] IUniswapV2Pair
- [Ext] name
- [Ext] symbol
- [Ext] decimals
- [Ext] totalSupply
- [Ext] balanceOf
- [Ext] allowance
- [Ext] approve #
- [Ext] transfer #
- [Ext] transferFrom #
- [Ext] DOMAIN_SEPARATOR
- [Ext] PERMIT_TYPEHASH
- [Ext] nonces
- [Ext] permit #
- [Ext] MINIMUM_LIQUIDITY
- [Ext] factory
- [Ext] tokenO
- [Ext] tokenI
- [Ext] getReserves
- [Ext] priceOCumulativeLast
- [Ext] priceI CumulativeLast
- [Ext] kLast
- [Ext] burn #
- [Ext] swap #
- [Ext] skim #
- [Ext] sync #
- [Ext] initialize #

+ [Int] IUniswapV2RouterO1
- [Ext] factory
- [Ext] WETH
- [Ext] addLiquidity #
- [Ext] addLiquidityETH ($)
- [Ext] removeLiquidity #
- [Ext] removeLiquidityETH #
- [Ext] removeLiquidityWithPermit #
- [Ext] removeLiquidityETHWithPermit #
- [Ext] swapExactT okensForT okens #
- [Ext] swapT okensForExactT okens #
- [Ext] swapExactETHForTokens ($)
- [Ext] swapT okensForExactETH #
- [Ext] swapExactT okensForETH #
- [Ext] swapETHForExactTokens ($)
- [Ext] quote
- [Ext] getAmountOut
- [Ext] getAmountIn
- [Ext] getAmountsOut
- [Ext] getAmountsIn

+ [Int] IUniswapV2RouterO2 (IUniswapV2RouterO1)
- [Ext] removeLiquidityETHSupportingFeeOnTransferTokens #
- [Ext] removeLiquidityETHWithPermitSupportingFeeOnTransferTokens #

[Ext] swapExactT okensForT okensSupporting FeeOnT ransferT okens
[Ext] swapExactETHForTokensSupportingFeeOnTransferTokens ($)
[Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

\+ Kingwarrior (Context, IERC20, Ownable)
[Pub] ‹Constructor› #
[Pub] name
[Pub] symbol
[Pub] decimals
[Pub] totalSupply
[Pub] balanceOf
[Pub] transfer #
[Pub] allowance
[Pub] approve #
[Pub] transferFrom #
[Pub] increaseAllowance #
[Pub] decreaseAllowance #
[Pub] isExcludedFromReward
[Pub] totalFees
[Pub] minimumTokensBeforeSwapAmount
[Pub] buyBackUpperLimitAmount
[Pub] deliver #
[Pub] reflectionFromToken
[Pub] tokenFromReflection
[Pub] excludeFromReward #
-modifiers: onlyOwner
[Ext] includeInReward #
-modifiers: onlyOwner
[Prv] .approve #
[Prvj _transfer #
[Prv] swapTokens #
-modifiers: lockTheSwap
[Prv] buyBackTokens #
-modifiers: lockTheSwap
[Prv] swapT okensForEth #
[Prv] swap ETHForT okens #
[Prv] addLiquidity #
[Prv] _tokenTransfer #
[Prvj _transferStandard #
[Prv] _transferToExcluded #
[Prv] _transferFromExcluded #
[Prv] _transferBothExcluded #
[Prv] _reflectFee #
[Prv] _getValues
[Prv] _getTValues
[Prv] _getRValues
[Prv] _getRate
[Prv] _getCurrentSupply
[Prv] _takeLiquidity #
[Prv] calculateTaxFee
[Prv] calculateLiquidityFee
[Prv] removeAllFee #
[Prv] restoreAllFee #
[Pub] isExcludedFromFee
[Pub] excludeFromFee #

-modifiers: onlyOwner -[Pub] includeInFee #
-modifiers: onlyOwner
-[Ext] setT axFeePercent #
-modifiers: onlyOwner
-[Ext] setLiquidityFeePercent #
-modifiers: onlyOwner
-[Ext] setMaxT xAmount #
-modifiers: onlyOwner
-[Ext] setMarketingDivisor #
-modifiers: onlyOwner
-[Ext] setNumTokensSellToAddToLiquidity # -
modifiers: onlyOwner
-[Ext] setBuybackUpperLimit #
- m o d i f i e r s : o n l y O w n e r - [E x t]
setMarketingAddress #
-modifiers: onlyOwner
-[Pub] setSwapAndLiquifyEnabled #
-modifiers: onlyOwner
-[Pub] setBuyBackEnabled #
-modifiers: onlyOwner
-[Ext] prepareForPreSale #
-modifiers: onlyOwner
-[Ext] afterPreSale #
-modifiers: onlyOwner
-[Prv] transferT oAddressETH #
-[Ext] <Fallback> ($)

($) = payable function
= non-constant function

Issues Checking Status

	Issue description	Checking status
1.	Compiler errors.	Passed
2.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3.	Possible delays in data delivery.	Passed
4.	Oracle calls.	Passed
5.	Front running.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow.	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Low issues
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	Passed
12.	The impact of the exchange rate on the logic.	Passed
13.	Private user data leaks.	Passed
14.	Malicious Event log.	Passed
15.	Scoping and Declarations.	Passed
16.	Uninitialized storage pointers.	Passed
17.	Arithmetic accuracy.	Passed
18.	Design Logic.	Passed
19.	Cross-function race conditions.	Passed
20.	Safe Open Zeppelin contracts implementation and usage	Passed
21.	Fallback function security.	Passed

Security Issues

High Severity Issues

No high severity issues found.

Medium Severity Issues

No medium severity issues found.

Low Severity Issues

1. Out of gas

Issue:

- **The function includeInReward() uses the loop to find and remove addresses from the .excluded list. Function will be aborted with OUT_OF_GAS exception if there will be a long excluded addresses list.**

```
function includeInReward(address account!) external onlyOwner()
  { require(_isExciudecj[accountt], "Account is already excluded"); for (uint256 i = 0;
  i < ^excluded.length; i++) {
        if (Lexcludecf[i] = account!) { texclude^[i] = Lexciuded(Lexctuded.length - 1];
         ^tOwned[account t ] = 0; tisExciuded[account t ] = false; Lexclude&pop();
         break;
       }
  ?
```

- **The function _getCurrentSupply also uses the loop for evaluating total supply. It also could be aborted with OUT_oF_GAS exception if there will be a long excluded addresses list.**

```
function _getCurrentSupply() private view returns (uint256, uint256) { uint256
  rSupply = i  rTota:
      uint256 tSupply          LtTotali;    ....
      for (uint256 i = 0; i < ^excluded.length; i++) {
      if ( ......
      ...
              LrOwned [^excluded[i]] > rSupply 11
      )return (rTotal, _tTotal);      LtOwned[_excluded[i]] > tSupply
...
        rSupply = rSupply.sub (trOwned[_exctudeS[i]]); tSupply =
      tSupply.sub (ttOwned[^exciuded|[i]]);
  } “
..
..
      if (rSupply v _rTotal.div(_tTotal)) return (_rTotal, _tTotal); return (rSupply,
  tSupply);
}
```

Recommendation:

Check that the excluded array length is not too big.

Owner privileges (In the period when the owner is not renounced)

- **Owner can change tax and liquidity fees.**

```
ftrace | funcSig
function setTaxFeePercent(uint256 taxFeet) external onlyOwner() {
    ttaxFee = taxFeet;
}.....

ftrace | funcSig
function setLiquidityFeePercent(uint256 liquidityFeef) external onlyOwner() { tliquidityFee =
    liquidityFeef;
}
```

- **Owner can change maximum transaction amount.**

```
ftrace | funcSig
function setMaxTxAmount(uint256 maxTxAmountt) external onlyOwner() { _maxTxAmount|
    = maxTxAmountt;
}
```

- **Owner can exclude from the fee.**

```
function excludeFromFee(address accounts) public onlyOwner
    { LisExcludedFromFee[account t ] = true;
}
```

- **Owner can change marketingDivisor.**

```
ftrace | funcSig
function setMarketingDivisor(uint256 divisorf) external onlyOwner() { marketingDivisot =
    divisor!;
}
```

- **Owner can change minimum number of tokens to add to liquidity.**

```
ftrace | funcSig
function setNumTokensSellToAddToLiquidity(uint256 _minimumTokensBeforeSwapt) external onlyOwner()

  { minimumTokensBeforeSwa^ = _minimumTokensBeforeSwap o    ;
}
```

- **Owner can change buyBackUpperLimit.**

```
ftrace | funcSig
function setBuybackUpperLimit(uint256 buyBackLimitf) external onlyOwner() { buyBackUpperLimit =
    buyBackLimitt * 10**18;
}
```

- **Owner can change marketing address.**

```
ftrace | funcSig
function setMarketingAddress(address _marketingAddresst) external onlyOwner() { marketingAddress
    = payable(_marketingAddresst);
}
```

- **Owner can enable and disable buyBack.**

```
ftrace | funcSig
function setBuyBackEnabled(bool _enabledt) public onlyOwner
    { buyBackEnabled = _enabledt;
      emit BuyBackEnabledIlpdated (_enabled t);
}
```

- **Owner can enable before and after presale modes.**

```
ftrace | funcSig
function prepareForPreSale() external onlyOwner {
    setSwapAndLiquifyEnabled(false); _taxFee
    = 0;
     ^liquidityFee = 0; _maxTxAmount =
20,000,000 * 10**6 }
```

```
* 10**9;
```

```
ftrace | funcSig
function afterPreSale() external onlyOwner
    { setSwapAndLiquifyEnabled(t rue);
     LtaxFee = 2;
                    j = 9;
     :maxTxAmount  =3000000 * 10**6 * 10**9;
}
```

- **Owner can lock and unlock. By the way, using these functions the owner could retake privileges even after the ownership was renounced.**

```
//Locks the contract for owner for the amount of time provided function lock(uint256 time) public
virtual onlyOwner {
    _previousOwner = _owner;
    __owner = address(0);
    _lockTime = now + time;
    emit OwnershipTransferred(.owner, address(0));
            }

//Unlocks the contract for owner when _lockTime is exceeds function unlock() public virtual {
    require(_previousOwner == msg.sender, "You don't have permission to unlock"); require(now >
    _lockTime , "Contract is locked until 7 days"); emit Ownership!ransferred(_owner,
    _previousOwner);
    _owner = _previousOwner;
            }
```

Conclusion

Smart contracts contain low severity issues! Liquidity pair contract's security is not checked due to out of scope. One third of the liquidity goes to marketing address.

Liquidity locking details NOT provided by the team.

Auditek note:

Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.