



AUDITEK SECURITY

COMPLETE AUDIT REPORT

AOK Chain
Security Assessment

AUGUST 21TH, 2021

Prepared by
ELLEN DOWNING

Approved by
LILLIAN PRATT

Audit Details



Audited project

AOK Chain



Description

An experimental digital Proof-of-Stake currency



Client contacts:

AOK CHAIN TEAM



Platform

Proprietary; C++



Project website:

<https://aok.network/>



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 AOK CHAIN to perform an audit of smart contracts:

<https://bscscan.com/token/0xbbca21d9a94c53ea105cee8941e04880b257b8d4>

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 20.08.2021

Contract name	AOK CHAIN
Contract address	0xbbca21d9a94c53ea105cee8941e04880b257b8d4
Total supply	100,000,000,000,000,000
Token ticker	AOK TOKEN
Decimals	12
Token holders	12
Transactions count	23
Top 100 holders dominance	100.00%
Liquidity fee	4
Tax fee	8
Total fees	12
Uniswap V2 pair	0x93d94fcb0dcc8a88257b2d2eec7a2615ebedb542
Contract deployer address	0x21C14b3361383e7e56d034E98eDa2Ae350570937
Contract's current owner address	0x21C14b3361383e7e56d034E98eDa2Ae350570937

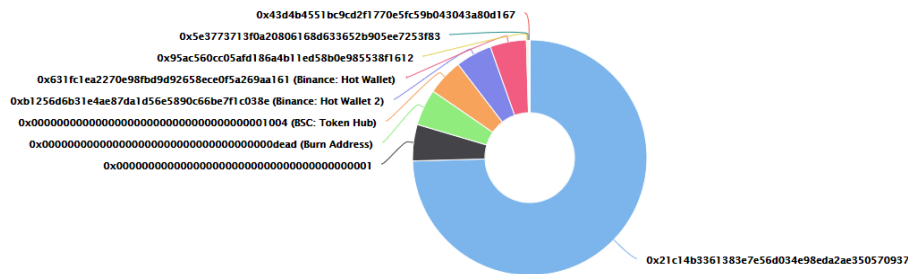
AOK CHAIN Token Distribution

💡 The top 100 holders collectively own 100.00% (99,999,999,999,999,900.00 Tokens) of AOK TOKEN.Finance

💡 Token Total Supply: 100,000,000,000,000.00 Token | Total Token Holders: 15

AOK TOKEN.Finance Top 100 Token Holders

Source: BscScan.com



(A total of 99,999,999,999,900.00 tokens held by the top 100 accounts from the total supply of 100,000,000,000,000.00 token)

AOK CHAIN Contract Interaction Details

Transactions **Contract** Events Analytics Comments

Code Read Contract Write Contract

Search Source Code

✔ **Contract Source Code Verified** (Exact Match)

Contract Name: AOKTOKEN

Optimization Enabled: Yes with 200 runs

Compiler Version v0.5.17+commit.d19bba13

Other Settings: **default** evmVersion, **GNU GPLv2** license

 Contract Source Code (Solidity)


Outline ▾ More Options ▾

AOK CHAIN Top 10 Token Holders

[Token Holders Chart](#)

A total of 15 token holders

First < Page 1 of 1 > Last

Rank	Address	Quantity	Percentage	Analytics
1	0x21c14b3361383e7e56d034e98eda2ae350570937	74,553,619,529,080,600.0856138855	74.5536%	📊
2	0x0000000000000000000000000000000000000001	5,000,000,000,000,000	5.0000%	📊
3	Burn Address	5,000,000,000,000,000	5.0000%	📊
4	 BSC: Token Hub	5,000,000,000,000,000	5.0000%	📊
5	Binance: Hot Wallet 2	5,000,000,000,000,000	5.0000%	📊
6	Binance: Hot Wallet	5,000,000,000,000,000	5.0000%	📊
7	0x95ac560cc05afd186a4b11ed58b0e985538f1612	149,327,092,450,561.130544563595	0.1493%	📊
8	0x5e3773713f0a20806168d633652b905ee7253f83	148,732,904,497,095.235080417328	0.1487%	📊
9	0x43d4b4551bc9cd2f1770e5fc59b043043a80d167	148,142,257,472,541.539024128512	0.1481%	📊
10	0xd34af3981db2b1fdcf6e0416cdc354428e0952c	149,014,967,478.87112432261	0.0001%	📊

A total of 10 holders

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] token0
- [Ext] token1
- [Ext] getReserves
- [Ext] price0CumulativeLast
- [Ext] price1CumulativeLast
- [Ext] kLast
- [Ext] burn #
- [Ext] swap #
- [Ext] skim #
- [Ext] sync #
- [Ext] initialize #

+ [Int] IUniswapV2Router01

- [Ext] factory
- [Ext] WETH
- [Ext] addLiquidity #
- [Ext] addLiquidityETH (\$)
- [Ext] removeLiquidity #
- [Ext] removeLiquidityETH #
- [Ext] removeLiquidityWithPermit #
- [Ext] removeLiquidityETHWithPermit #
- [Ext] swapExactTokensForTokens #
- [Ext] swapTokensForExactTokens #
- [Ext] swapExactETHForTokens (\$)
- [Ext] swapTokensForExactETH #
- [Ext] swapExactTokensForETH #
- [Ext] swapETHForExactTokens (\$)
- [Ext] quote
- [Ext] getAmountOut
- [Ext] getAmountIn
- [Ext] getAmountsOut
- [Ext] getAmountsIn

+ [Int] IUniswapV2Router02 (IUniswapV2Router01)

- [Ext] removeLiquidityETHSupportingFeeOnTransferTokens #
- [Ext] removeLiquidityETHWithPermitSupportingFeeOnTransferTokens #

- [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens #
- [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 #
 - [Prv] _transfer #
 - [Prv] swapTokens #
 - modifiers: lockTheSwap
 - [Prv] buyBackTokens #
 - modifiers: lockTheSwap
 - [Prv] swapTokensForEth #
 - [Prv] swapETHForTokens #
 - [Prv] addLiquidity #
 - [Prv] _tokenTransfer #
 - [Prv] _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]** setTaxFeePercent **#**
 - modifiers: onlyOwner
- **[Ext]** setLiquidityFeePercent **#**
 - modifiers: onlyOwner
- **[Ext]** setMaxTxAmount **#**
 - modifiers: onlyOwner
- **[Ext]** setMarketingDivisor **#**
 - modifiers: onlyOwner
- **[Ext]** setNumTokensSellToAddToLiquidity **#**
 - modifiers: onlyOwner
- **[Ext]** setBuybackUpperLimit **#**
 - modifiers: onlyOwner
- **[Ext]** setMarketingAddress **#**
 - modifiers: onlyOwner
- **[Pub]** setSwapAndLiquifyEnabled **#**
 - modifiers: onlyOwner
- **[Pub]** setBuyBackEnabled **#**
 - modifiers: onlyOwner
- **[Ext]** prepareForPreSale **#**
 - modifiers: onlyOwner
- **[Ext]** afterPreSale **#**
 - modifiers: onlyOwner
- **[Prv]** transferToAddressETH **#**
- **[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 setBTCBRewardsFee(uint256 value) external onlyOwner{
    BTCBRewardsFee = value;
    totalFees = BTCBRewardsFee.add(liquidityFee);
}

function setLiquidityFee(uint256 value) external onlyOwner{
    liquidityFee = value;
    totalFees = BTCBRewardsFee.add(liquidityFee);
}

function setAutomatedMarketMakerPair(address pair, bool value) public onlyOwner {
    require(pair != uniswapV2Pair, "DRAGONDOGE: The PancakeSwap pair cannot be removed from automatedMarketMakerPairs");
```

- 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 = _rTotal;
    uint256 tSupply = _tTotal;
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (
            _rOwned[_excluded[i]] > rSupply ||
            _tOwned[_excluded[i]] > tSupply
        ) return (_rTotal, _tTotal);
        rSupply = rSupply.sub(_rOwned[_excluded[i]]);
        tSupply = tSupply.sub(_tOwned[_excluded[i]]);
    }
    if (rSupply < _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.

```
address public immutable BTCB = address(0x7130d2A12B9BCbFAe4f2634d864A1Ee1Ce3Ead9c);

uint256 public swapTokensAtAmount = 2000000 * (10**18);

mapping(address => bool) public _isBlacklisted;

uint256 public BTCBRewardsFee = 8;
uint256 public liquidityFee = 4;

uint256 public totalFees = BTCBRewardsFee.add(liquidityFee);
```

- Owner can change maximum transaction amount.

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

- Owner can exclude from the fee.

```
function excludeFromFee(address account↑) public onlyOwner {
    _isExcludedFromFee[account↑] = true;
}
```

- Owner can change marketingDivisor.

```
ftrace | funcSig
function setMarketingDivisor(uint256 divisor↑) external onlyOwner() {
    marketingDivisor = divisor↑;
}
```

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

```
ftrace | funcSig
function setNumTokensSellToAddToLiquidity(uint256 _minimumTokensBeforeSwap↑) external onlyOwner() {
    minimumTokensBeforeSwap = _minimumTokensBeforeSwap↑;
}
```

- Owner can change buyBackUpperLimit.

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

- Owner can change marketing address.

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

- Owner can enable and disable buyBack.

```
ftrace | funcSig
function setBuyBackEnabled(bool _enabled↑) public onlyOwner {
    buyBackEnabled = _enabled↑;
    emit BuyBackEnabledUpdated(_enabled↑);
}
```

- 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 OwnershipTransferred(_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.

Contact:

Website: Auditek.org

Twitter: @auditekofficial

Telegram channel: @auditekofficial

Marketing: @auditek01

