Cyberscope

Audit Report

Alpha Ape Network

February 2022

Type	BEP20
Network	BSC
Address	0xa621B4B852a4E48688924f4436f16CE47405133F
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	AANT
Compiler Version	v0.7.6+commit.7338295f
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xa621B4B852a4E48688924f4436f16CE47405133F
Symbol	AANT
Decimals	18
Total Supply	75,000,000,000,000
Source	contract.sol
Domain	alphaape.net

Audit Updates

Initial Audit	1st March 2022
Corrected	

Contract Analysis

Critical Medium Minor Pass

Severity	Code	Description
●	ST	Contract Owner is not able to stop or pause transactions
●	OCTD	Contract Owner is not able to transfer tokens from specific address
●	OTUT	Owner Transfer User's Tokens
●	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
●	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
●	MT	Contract Owner is not able to mint new tokens
●	BT	Contract Owner is not able to burn tokens from specific wallet
●	BC	Contract Owner is not able to blacklist wallets from selling

ST - Stop Transactions

Criticality	critical
Location	contract.sol#L1070

Description

The contract owner has the authority to stop selling for all users excluding the owner. The owner may take advantage of it by setting `maxSellTransactionAmount` to a very low value. This way the contract will operate as a honeypot.

```
 if(!swapping && automatedMarketMakerPairs[to] && from !=
address(uniswapV2Router) && !_isExcludedFromFees[to])
        {
            require(amount <= maxSellTransactionAmount, "Amount exceeds the
Protocol-X");
        }
```

Recommendation

The contract could embody a check for not allowing setting `maxSellTransactionAmount` less than a reasonable amount. A suggested implementation could check that the maximum amount should be more than a fixed percentage of the total supply.

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

OCTD - Owner Contract Tokens Drain

Criticality	minor
Location	contract.sol#L1278,1283

Description

The contract owner has the authority to claim all the balance of the contract. The owner may take advantage of it by calling any of the below withdraw functions.

```
    function withdrawRemainingToken(address account) public onlyOwner {
        uint256 balance = balanceOf(address(this));
        super._transfer(address(this), account, balance);
    }

    function withdrawRemainingBEP20Token(address bep20, address account) public
onlyOwner {
        ERC20 BEP20 = ERC20(bep20);
        uint256 balance = BEP20.balanceOf(address(this));
        BEP20.transfer(account, balance);
    }
```

Recommendation

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

BC - Blacklisted Contracts

Criticality	critical
Location	contract.sol#L465

Description

The contract owner has the authority to stop contracts from transactions. The owner may take advantage of it by calling the `addToBlackList` function.

```
   function transferFrom(address sender, address recipient, uint256 amount)
public virtual override returns (bool) {
        require(!_isBlacklisted[sender] && !_isBlacklisted[recipient], "This
address is BlackListed!");
        _transfer(sender, recipient, amount);
```

Recommendation

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L05	Unused State Variable
	L09	Dead Code Elimination
	L11	Unnecessary Boolean equality
	L12	Using Variables before Declaration
	L07	Missing Events Arithmetic
	L15	Local Scope Variable Shadowing
	L14	Uninitialized Variables in Local Scope
	L08	Tautology or Contradiction
	L13	Divide before Multiply Operation

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L251,255,262,268,323,328,430,434,438,450 and 30 more

Description

Public functions that are never called by the contract should be declared external to save gas.

```
process
getAccountAtIndex
distributeWBNBDividends
burnRemainingToken
withdrawRemainingBEP20Token
withdrawRemainingToken
updateETHRewardsFee
updateLiquidityFee
updateMarketingFee
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L770,774,613

Description

Constant state variables should be declared constant to save gas.

```
lastAmount
marketingDivisor
_rateLimitSeconds
```

Recommendation

Add the constant attribute to state variables that never change.

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L778,613

Description

There are segments that contain unused state variables.

```
lastAmount
DefaultLiquidityLockTime
```

Recommendation

Remove unused state variables.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L31,690,521,594,584,599,533,551,556,544 and 3 more

Description

Functions that are not used in the contract, and make the code's size bigger.

```
mul
div
trySub
tryMul
tryMod
tryDiv
tryAdd
mod
_setupDecimals
...
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L1044

Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
getAntiBotEnabled() == true
```

Recommendation

Remove the equality to the boolean constant.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract.sol#L1126

Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
iterations
claims
lastProcessedIndex
```

Recommendation

The variables should be declared before any usage of them.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L882,886,890,894,1263,1268,1273

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
liquidityFee = newFee
marketingFee = newFee
swapTokensAtAmount = amountOfTokens
maxSellTransactionAmount = maxSell * (10 ** 18)
maxBuyTranscationAmount = maxBuy * (10 ** 18)
_maxWalletToken = maxWallet * (10 ** 18)
```

Recommendation

Emit an event for critical parameter changes.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L625

Description

The are variables that are defined in the local scope containing the same name from an upper scope.

```
_symbol
_name
```

Recommendation

The local variables should have different names from the upper scoped variables.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L745,1126

Description

The are variables that are defined in the local scope and are not initialized.

```
lastProcessedIndex
claims
iterations
i
```

Recommendation

All the local scoped variables should be initialized.

L08 - Tautology or Contradiction

Criticality	minor
Location	contract.sol#L1263,1268,1273

Description

Detects expressions that are tautologies or contradictions. For instance, an uint variable will always be greater than or equal to zero.

```
require(bool,string)(newFee >= 0 && newFee <= 15,AANT: ETH/BNB Rewards tax must be between 0 and 15)
require(bool,string)(newFee >= 0 && newFee <= 10,AANT: Liquidity tax must be between 0 and 10)
require(bool,string)(newFee >= 0 && newFee <= 10,AANT: Expenses tax must be between 0 and 10)
```

Recommendation

Fix the incorrect comparison by changing the value type or the comparison.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L1044

Description

Performing divisions before multiplications may cause lose of prediction.

```
fees = amount.mul(totalFees).div(100)
transferToMarketingWallet(address(marketingWallet),address(this).balance.div(10
** 2).mul(marketingDivisor))
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-

	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Pair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-

	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IterableMapping	Library			
	get	Public		-
	getIndexOfKey	Public		-
	getKeyAtIndex	Public		-
	size	Public		-
	set	Public	✓	-
	remove	Public	✓	-
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner

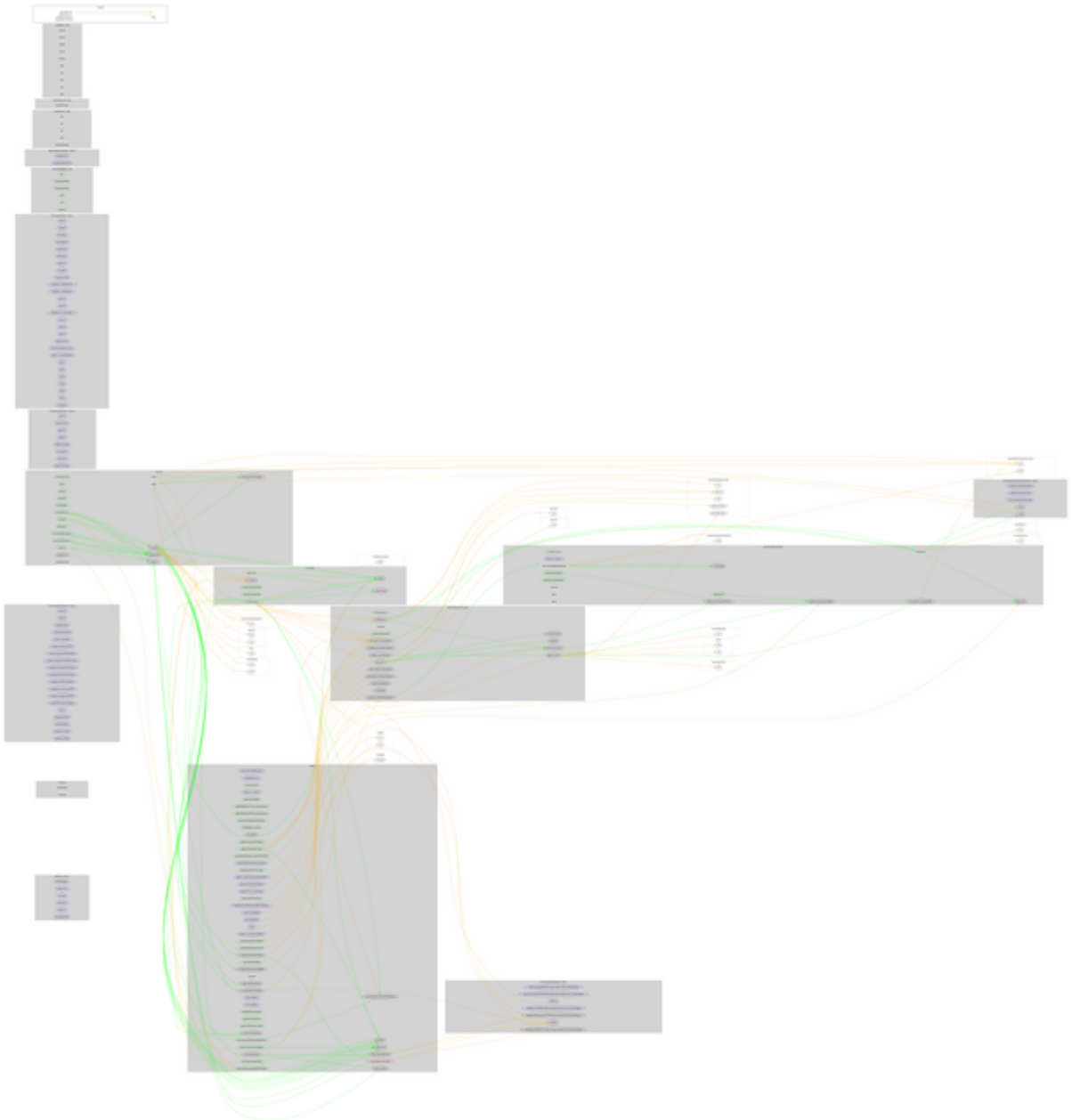
IDividendPayingTokenOptional	Interface			
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	External		-
IDividendPayingToken	Interface			
	dividendOf	External		-
	withdrawDividend	External	✓	-
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	toUint256Safe	Internal		
SafeMathUint	Library			
	toInt256Safe	Internal		
ERC20	Implementation	Context, IERC20		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	

	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_setupDecimals	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
SafeMath	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
	div	Internal		
	mod	Internal		
DividendPayingToken	Implementation	ERC20, IDividendPayingToken, IDividendPayingTokenOptional		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	_distributeWBNBDividends	Internal	✓	
	withdrawDividend	Public	✓	-
	_withdrawDividendOfUser	Internal	✓	
	dividendOf	Public		-
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	_transfer	Internal	✓	

	_mint	Internal	✓	
	_burn	Internal	✓	
	_setBalance	Internal	✓	
AANT	Implementation	ERC20, Ownable		
	removeFromBlackList	External	✓	onlyOwner
	addToBlackList	External	✓	onlyOwner
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	updateMaxWallet	Public	✓	onlyOwner
	updateMaxBuyTranscationAmount	Public	✓	onlyOwner
	updateMaxSellTransactionAmount	Public	✓	onlyOwner
	updateSwapTokensAtAmount	Public	✓	onlyOwner
	multiWalletTransfer	Public	✓	onlyOwner
	updateDividendTracker	Public	✓	onlyOwner
	updateUniswapV2Router	Public	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	excludeMultipleAccountsFromFees	Public	✓	onlyOwner
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	updateGasForProcessing	Public	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getClaimWait	External		-
	getTotalDividendsDistributed	External		-
	isExcludedFromFees	Public		-
	withdrawableDividendOf	Public		-
	dividendTokenBalanceOf	Public		-
	getAccountDividendsInfo	External		-
	getAccountDividendsInfoAtIndex	External		-
	processDividendTracker	External	✓	-
	claim	External	✓	-
	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	setAntiBotEnabled	Public	✓	onlyOwner
	getAntiBotEnabled	Public		-

	_transfer	Internal	✓	
	swapAndLiquify	Private	✓	
	swapTokensForEth	Private	✓	
	swapTokensForAANT	Private	✓	
	addLiquidity	Private	✓	
	swapAndSendDividends	Private	✓	
	removeAllFee	External	✓	onlyOwner
	restoreAllFee	External	✓	onlyOwner
	updateMarketingFee	Public	✓	onlyOwner
	updateLiquidityFee	Public	✓	onlyOwner
	updateETHRewardsFee	Public	✓	onlyOwner
	withdrawRemainingToken	Public	✓	onlyOwner
	withdrawRemainingBEP20Token	Public	✓	onlyOwner
	burnRemainingToken	Public	✓	onlyOwner
	checkBot	Private	✓	
	swapTokensForBNB	Private	✓	
	transferToMarketingWallet	Private	✓	
AANTDividend Tracker	Implementation	DividendPayingToken, Ownable		
	<Constructor>	Public	✓	DividendPayingToken
	distributeWBNBDividends	Public	✓	onlyOwner
	_transfer	Internal		
	withdrawDividend	Public		-
	excludeFromDividends	External	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getLastProcessedIndex	External		-
	getNumberOfTokenHolders	External		-
	getAccount	Public		-
	getAccountAtIndex	Public		-
	canAutoClaim	Private		
	setBalance	External	✓	onlyOwner
	process	Public	✓	-
	processAccount	Public	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	alphaape.net
Registry Domain ID	2666060254_DOMAIN_NET-VRSN
Creation Date	2022-01-04T19:58:00.00Z
Updated Date	2022-01-11T13:50:36.00Z
Registry Expiry Date	
Registrar WHOIS Server	WHOIS.ENOM.COM
Registrar URL	WWW.ENOM.COM
Registrar	ENOM, INC.
Registrar IANA ID	48

The domain has been created about 2 months before the creation of the audit.

There is no public billing information, the creator is protected by the privacy settings.

Summary

AANT is the native token for the Alpha Ape Network. The Project has a friendly and growing community. There are some functions that can be abused by the owner, like mass blacklisting users and transferring funds to the team's wallet. The contract might also be converted into a honeypot and prevent users from selling if the owner abuses the admin ownership. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provides all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io