Cyberscope

Audit Report

NDAPP

May 2022

Type	BEP20
Network	BSC
Address	0xAe488aac5f6F42cbFB058a625Cb78D0e1d9caFeC
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	NDAPProtocol
Compiler Version	v0.7.6+commit.7338295f
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xae488aac5f6f42cbfb058a625cb78d0e1d9cafec
Symbol	NDAPP
Decimals	18
Total Supply	5,000,000,000
Domain	ndapp.finance

Source Files

Filename	**SHA256**
contract.sol	a6436f1294a4618bab991f44cbf86dd25c2b50e41217236db2e862569a3aa4f5

Audit Updates

Initial Audit	3rd June 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#L511,518,522

Description

The contract owner has the authority to stop the sales for all users excluding the owner. The owner may take advantage of it by setting the `TwentyFourhours` to a high value and the `maxSellAmount` to 1. As a result, some users will not be able to sell more than 1% of their holdings and some others will not be able to sell completely.

```
uint256 sellPercent = balanceOf(sender).mul(maxSellAmount).div(100); //Should
use variable
require(amount <= sellPercent, "ERR: Can't sell more than maxSellAmount
percent");

if(blkTime > tradeData[sender].lastTradeTime + TwentyFourhours) {
    tradeData[sender].lastTradeTime = blkTime;
    tradeData[sender].tradeAmount = amount;
}
else if( (blkTime < tradeData[sender].lastTradeTime + TwentyFourhours) && ((
blkTime > tradeData[sender].lastTradeTime)) ){
    require(tradeData[sender].tradeAmount + amount <= sellPercent, "ERR: Can't
sell more than maxSellAmount percent in TwentyFourhours");
    tradeData[sender].tradeAmount = tradeData[sender].tradeAmount + amount;
}
```

The contract owner has the authority to stop transactions for all users excluding the owner. The owner may take advantage of it by setting the `initialDistributionFinished` to false or the maxSellTransactionAmount to a very low value.

```
require(initialDistributionFinished || excludedAccount, "Trading not started");
///
require(amount <= maxSellTransactionAmount, "Error amount");
```

Recommendation

The contract should not allow manipulating name sensitive variables like `TwentyFourhours`.

The contract could embody a check for not allowing setting the `maxSellTransactionAmount` and `maxSellAmount` less than a reasonable amount. Additionally, the `initialDistributionFinished` should not be able to be disabled after the initial mount. 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.

Contract Diagnostics

🔴 Critical 🟠 Medium ⚪ Minor

Severity	Code	Description
🔴	MTS	Manipulate Total Supply
⚪	L01	Public Function could be Declared External
⚪	L02	State Variables could be Declared Constant
⚪	L04	Conformance to Solidity Naming Conventions
⚪	L05	Unused State Variable
⚪	L07	Missing Events Arithmetic
⚪	L09	Dead Code Elimination
⚪	L13	Divide before Multiply Operation

MTS - Manipulate Total Supply

Criticality	critical
Location	contract.sol#L730

Description

Owner is able to aggressively manipulate total supply. This change will have a direct impact on the token price and Market Cap. The owner may take advantage of it by manipulating the rewardYield and rewardYieldDenominator properties. Additionally, if the `rewardYieldDenominator` set to zero, the transactions will revert.

```
function _rebase() private {
    if(!inSwap) {
        uint256 circulatingSupply = getCirculatingSupply().add(balanceOf(DEAD));
        int256 supplyDelta =
int256(circulatingSupply.mul(rewardYield).div(rewardYieldDenominator));

        coreRebase(supplyDelta);
    }
}
```

Recommendation

The contract owner should carefully manage the adjustment of the circulating supply (increases or decreases), according to the token’s price fluctuations.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L154,158,162,249,258,263,290,294,298

Description

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

```
renounceWhitelisted
removeWhitelisted
addWhitelisted
transferOwnership
renounceOwnership
owner
decimals
symbol
name
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L342,343,348,361

Description

Constant state variables should be declared constant to save gas.

```
feeDenominator
busdToken
ZERO
DEAD
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L170,375,440,776,797,802,807,812,821,826,832,855,864,869,874,879,884,889,893,330,331,342,343,381

Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
TwentyFourhours
ZERO
DEAD
_markerPairs
_isFeeExempt
_maxTxn
_nextRebase
_value
_enabled
...
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L7

Description

There are segments that contain unused state variables.

```
MAX_INT256
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L807,812,816,821,832,869,874,889,893

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.

```
maxSellTransactionAmount = _maxTxn
nextRebase = _nextRebase
rewardYield = _rewardYield
rebaseFrequency = _rebaseFrequency
liquidityFee = _liquidityFee
gonSwapThreshold = TOTAL_GONS.div(_denom).mul(_num)
targetLiquidity = target
TwentyFourhours = _time
maxSellAmount = _maxSellAmount
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L35

Description

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

```
abs
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L656,821

Description

Performing divisions before multiplications may cause lose of prediction.

```
gonSwapThreshold = TOTAL_GONS.div(_denom).mul(_num)
contractTokenBalance = _gonBalances[address(this)].div(_gonsPerFragment)
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

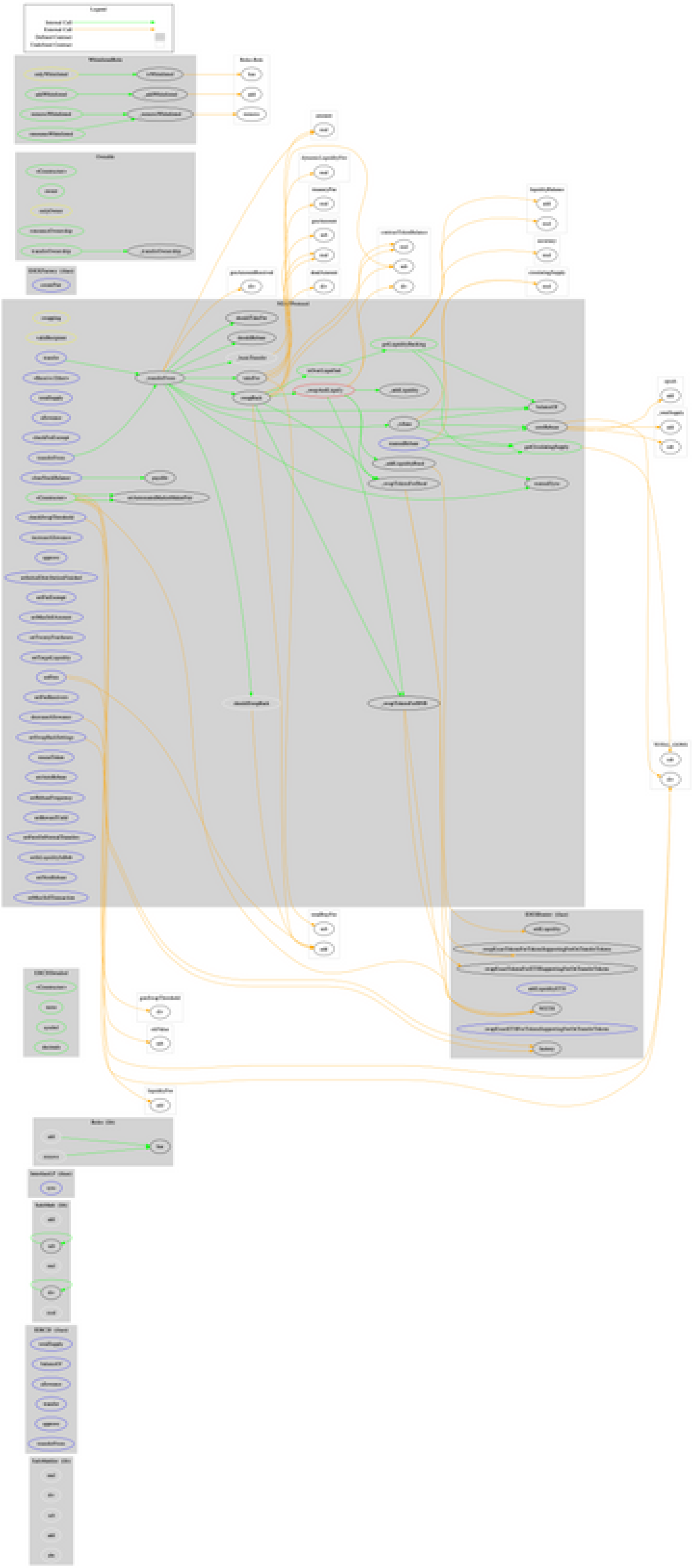
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	transfer	External	✓	-
	approve	External	✓	-
	transferFrom	External	✓	-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
InterfaceLP	Interface			
	sync	External	✓	-
Roles	Library			
	add	Internal	✓	

	remove	Internal	✓	
	has	Internal		
ERC20Detailed	Implementation	IERC20		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
IDEXRouter	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IDEXFactory	Interface			
	createPair	External	✓	-
Ownable	Implementation			
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
WhitelistedRole	Implementation	Ownable		
	isWhitelisted	Public		-
	addWhitelisted	Public	✓	onlyOwner
	removeWhitelisted	Public	✓	onlyOwner
	renounceWhitelisted	Public	✓	-

	_addWhitelisted	Internal	✓	
	_removeWhitelisted	Internal	✓	
NDAPProtocol	Implementation	ERC20Detailed, Ownable, WhitelistedRole		
	<Constructor>	Public	✓	ERC20Detailed
	<Receive Ether>	External	Payable	-
	totalSupply	External		-
	allowance	External		-
	balanceOf	Public		-
	checkFeeExempt	External		-
	checkSwapThreshold	External		-
	shouldRebase	Internal		
	shouldTakeFee	Internal		
	shouldSwapBack	Internal		
	getCirculatingSupply	Public		-
	getLiquidityBacking	Public		-
	isOverLiquified	Public		-
	manualSync	Public	✓	-
	transfer	External	✓	validRecipient
	_basicTransfer	Internal	✓	
	_transferFrom	Internal	✓	
	transferFrom	External	✓	validRecipient
	_swapAndLiquify	Private	✓	
	_addLiquidity	Private	✓	
	_addLiquidityBusd	Private	✓	
	_swapTokensForBNB	Private	✓	
	_swapTokensForBusd	Private	✓	
	swapBack	Internal	✓	swapping
	takeFee	Internal	✓	
	decreaseAllowance	External	✓	-
	increaseAllowance	External	✓	-
	approve	External	✓	-
	_rebase	Private	✓	

	coreRebase	Private	✓	
	manualRebase	External	✓	onlyWhitelisted
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	setInitialDistributionFinished	External	✓	onlyOwner
	setFeeExempt	External	✓	onlyOwner
	setMaxSellAmount	External	✓	onlyOwner
	setTwentyFourhours	External	✓	onlyOwner
	setTargetLiquidity	External	✓	onlyOwner
	setSwapBackSettings	External	✓	onlyOwner
	setFeeReceivers	External	✓	onlyOwner
	setFees	External	✓	onlyOwner
	clearStuckBalance	External	✓	onlyOwner
	rescueToken	External	✓	onlyOwner
	setAutoRebase	External	✓	onlyOwner
	setRebaseFrequency	External	✓	onlyOwner
	setRewardYield	External	✓	onlyOwner
	setFeesOnNormalTransfers	External	✓	onlyOwner
	setIsLiquidityInBnb	External	✓	onlyOwner
	setNextRebase	External	✓	onlyOwner
	setMaxSellTransaction	External	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	ndapp.finance
Registry Domain ID	42813cbd16994aa3b1e4fa8a705bbcc7-DONUTS
Creation Date	2022-05-25T16:10:31Z
Updated Date	2022-06-02T16:37:12Z
Registry Expiry Date	2023-05-25T16:10:31Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created 9 days before the creation of the audit. It will expire in 12 months.

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

Summary

The Smart Contract analysis reported some critical severity issues. The contract owner has the authority to stop transactions. The contract can be converted into a honeypot and prevent users from selling if the owner abuses the admin functions. Additionally, the contract owner has the authority to manipulate the total supply. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. There is also a limit of max 25% fees.

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 provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io