RugFreeCoins Audit

ADA Boy Token Audit

Smart Contract Security Audit

August 04, 2021

Contents

Audit details

Audited project

ADA Boy Token

Contract Address

0x1E653794A6849bC8A78be50C4D48981AfAD6359d

Client contact

ADA Boy Team

Blockchain

Binance smart chain

Project website

https://adaboytoken.com/

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 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 Rugfreecoins and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Rugfreecoins) owe no duty of care towards you or any other person, nor does Rugfreecoins 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 Rugfreecoins 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, Rugfreecoins hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Rugfreecoins, 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

Rugfreecoins was commissioned by ADA Boy to perform an audit of the smart contract.

https://bscscan.com/token/0x1E653794A6849bC8A78be50C4D48981AfAD6359d

The focus of this audit is to verify that the smart contract is secure, resilient and working according to the specifications.

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

About the project

ADA Boy is a token built on the Binance Smart Chain. Each transaction, purchase incur a 12% fee, and sales incur an 13% fee.

- The **automatic ADA reward of 8%** is what ADA Boy's entire marketing strategy is based around: that tokens will be distributed among every holder proportional to how many tokens each individual holds.

- The liquidity fee of 3%, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity. This is a key element for decentralized exchanges like Pancakeswap.

- The **sustainability fee of 1% marketing** is what allows ADA Boy to hold the aforementioned promise. Tokens will be swapped into ADA and will be sent to a marketing wallet per transaction. This way, ADA Boy will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.

Tokenomics

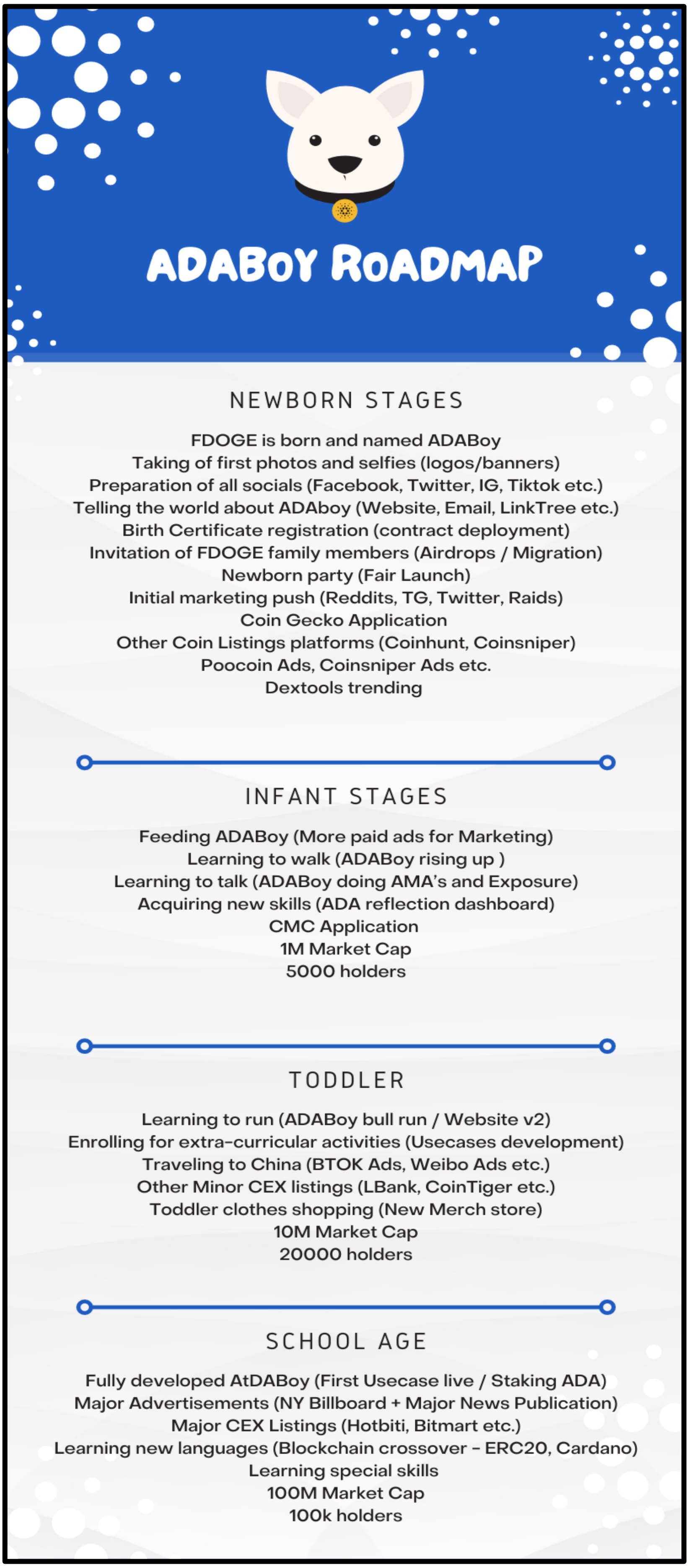
12% fee when buying

- 8% of trade goes to holders pockets in ADA.
- 3% of trade goes to the liquidity pool.
- 1% of trade goes to marketing.

13% fee when selling

- 8% of trade goes to holders pockets in ADA.
- 4% of trade goes to the liquidity pool.
- 1% of trade goes to marketing.

Roadmap



ADABoy Roadmap

NEWBORN STAGES

FDOGE is born and named ADABoy
Taking of first photos and selfies (logos/banners)
Preparation of all socials (Facebook, Twitter, IG, Tiktok etc.)
Telling the world about ADAboy (Website, Email, LinkTree etc.)
Birth Certificate registration (contract deployment)
Invitation of FDOGE family members (Airdrops / Migration)
Newborn party (Fair Launch)
Initial marketing push (Reddits, TG, Twitter, Raids)
Coin Gecko Application
Other Coin Listings platforms (Coinhunt, Coinsniper)
Poocoin Ads, Coinsniper Ads etc.
Dextools trending

INFANT STAGES

Feeding ADABoy (More paid ads for Marketing)
Learning to walk (ADABoy rising up)
Learning to talk (ADABoy doing AMA's and Exposure)
Acquiring new skills (ADA reflection dashboard)
CMC Application
1M Market Cap
5000 holders

TODDLER

Learning to run (ADABoy bull run / Website v2)
Enrolling for extra-curricular activities (Usecases development)
Traveling to China (BTOK Ads, Weibo Ads etc.)
Other Minor CEX listings (LBank, CoinTiger etc.)
Toddler clothes shopping (New Merch store)
10M Market Cap
20000 holders

SCHOOL AGE

Fully developed AtDABoy (First Usecase live / Staking ADA)
Major Advertisements (NY Billboard + Major News Publication)
Major CEX Listings (Hotbiti, Bitmart etc.)
Learning new languages (Blockchain crossover - ERC20, Cardano)
Learning special skills
100M Market Cap
100k holders

Target market and the concept

Target market

- ❖ Anyone who's interested in Crypto space with long term investment plans.
- ❖ Anyone who's ready to earn a passive income in ADA by holding tokens.
- ❖ Anyone who's interested in trading tokens.
- ❖ All Cardano inventors and fans out there.
- ❖ Anyone who's interested in stake ADA or ADA Boy and earn rewards.
- ❖ Anyone who's interested in taking part with the future lottery and games that's going to be built by the ADA Boy team.
- ❖ Anyone who's interested in making financial transactions with any other party using ADA Boy or ADA as the currency.

Core concept

The ADA reward system

8% of each transaction gets converted to ADA, and is split amongst all holders. The rewards are sent to holders that have at least 200,000 ADA Boy tokens, holders will be eligible to receive tokens every one hour and rewards are proportional to how many tokens each individual holds.

Sustainable mechanism

The liquidity fee of 3%, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

The **fee of 1% marketing** is what allows ADA Boy to promote the token and use funds to further development of the platform. Tokens will be swapped into ADA and will be sent to a marketing wallet per transaction. This way, ADA Boy will have access to the funds without selling tokens as the traditional way, which will enable them to consume funds without hurting the project.

The future plan

Dapp site to stake ADA or ADA Boy

The ADA Boy team is working on building a DApp for the holders to stake their ADA Boy tokens or ADA for a longer period and earn rewards.

Gaming and Lottery platform

The platform will allow users to use ADABoy tokens to play planned lottery-like games and earn pools of prizes.

Potential to grow with score points

1.	Project efficiency	8/10
2.	Project uniqueness	8/10
3	Information quality	8/10
4	Service quality	810
5	System quality	7/10
6	Impact on the community	8/10
7	Impact on the business	8/10
8	Preparing for the future	8/10
Total Points		**7.88/10**

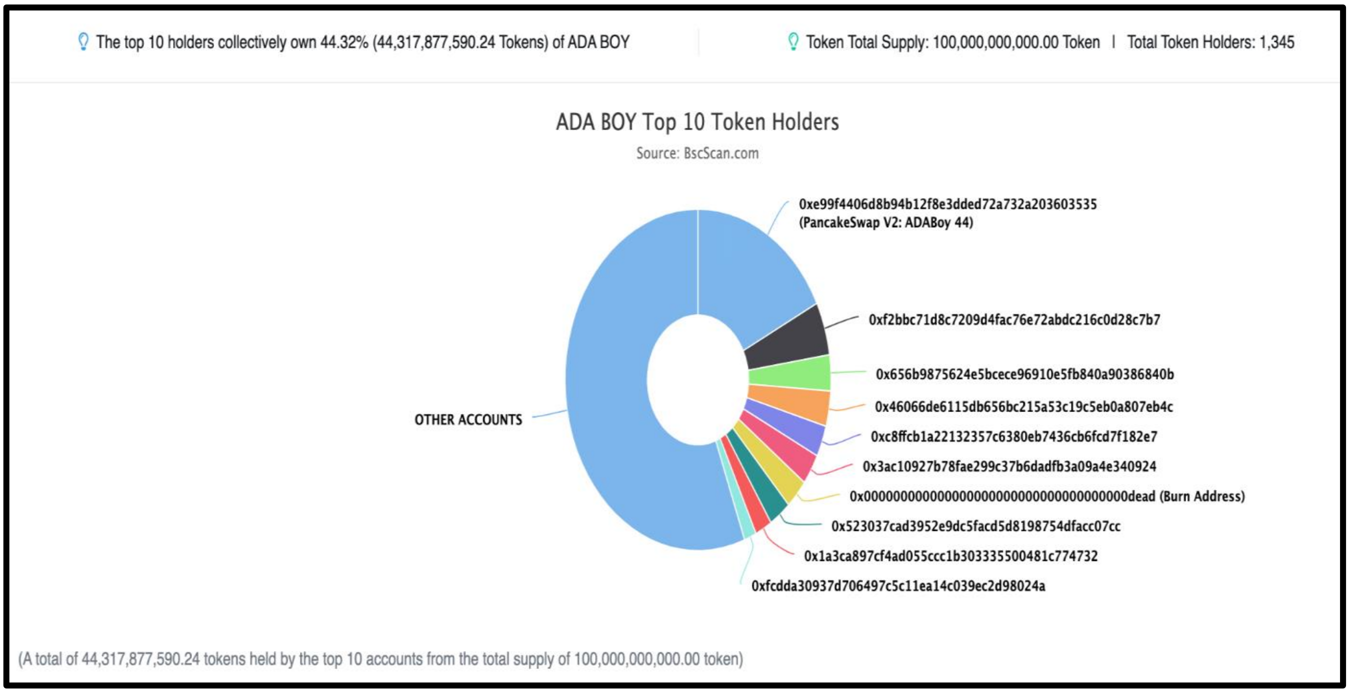
Contract details

Token contract details for 04th August 2021

Contract name	ADA Boy
Contract address	0x1E653794A6849bC8A78be50C4D48981AfAD6359d
Token supply	100,000,000,000
Token ticker	ADABoy
Decimals	18
Token holders	1,344
Transaction count	26,803
Top 100% holders dominance	86.00%
Dividend Tracker	0x82ba7c718bcd70338e33295e302aacb3598c29d0
Marketing wallet address	0xf2bbc71d8c7209d4fac76e72abdc216c0d28c7b7
Contract deployer address	0x252b4Ed3516F8921b36A71363b7047eb9cF069eF
Contract's current owner address	0x656b9875624e5bcece96910e5fb840a90386840b

Top token holders

Top 10 Token Holders



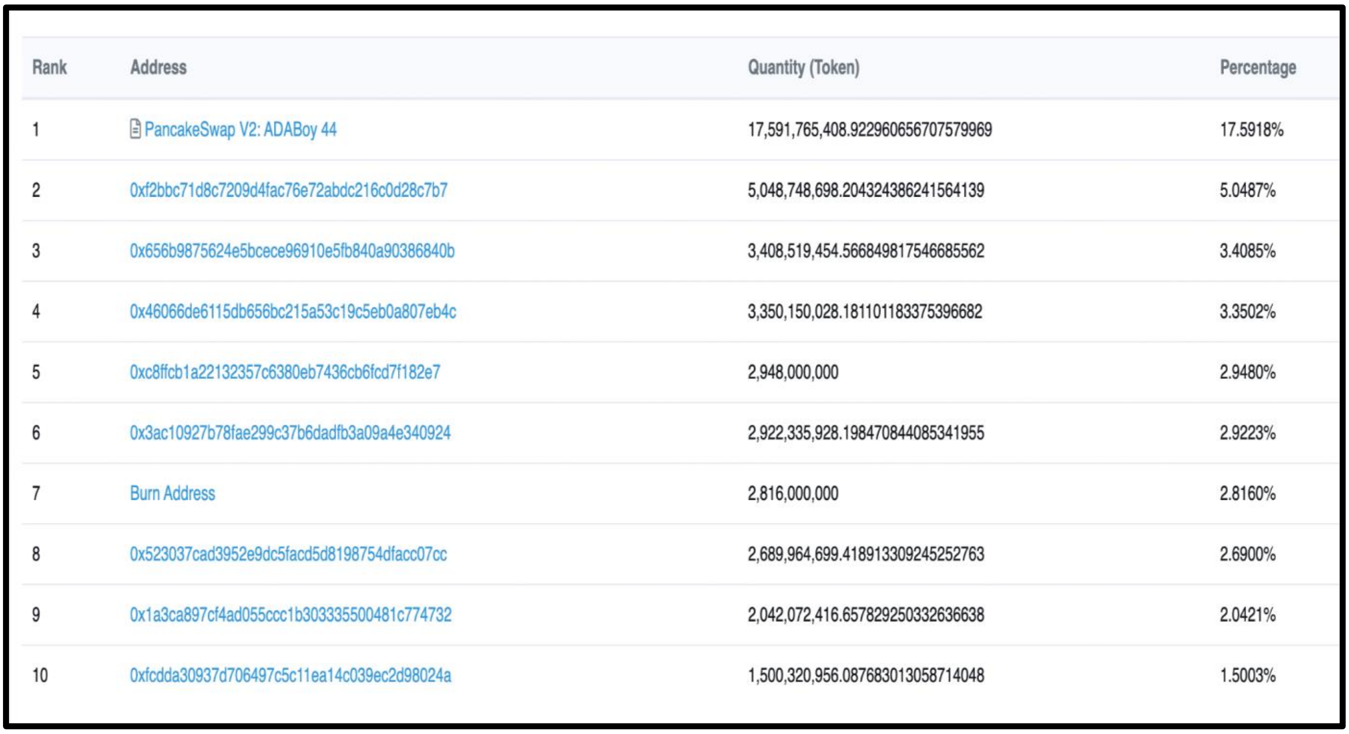
The top 10 holders collectively own 44.32% (44,317,877,590.24 Tokens) of ADA BOY

Token Total Supply: 100,000,000,000.00 Token | Total Token Holders: 1,345

ADA BOY Top 10 Token Holders

Source: BscScan.com

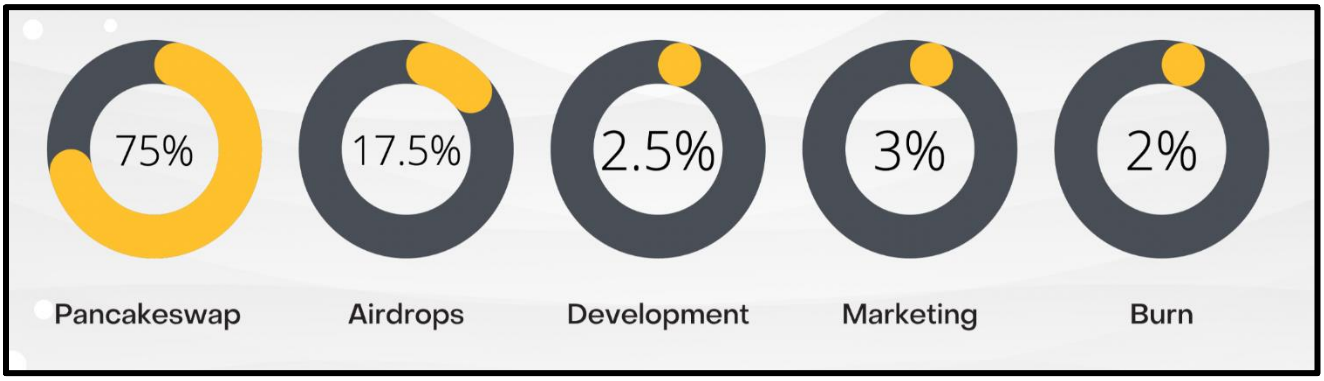
(A total of 44,317,877,590.24 tokens held by the top 10 accounts from the total supply of 100,000,000,000.00 token)



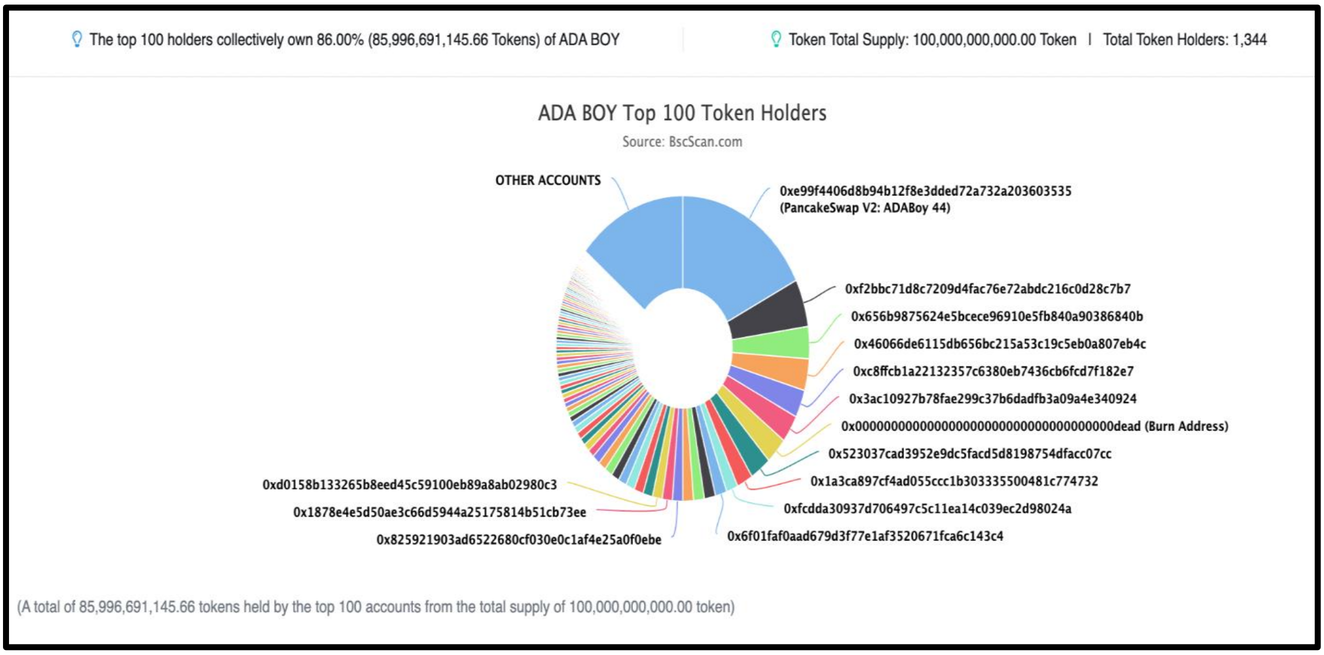
Rank	Address	Quantity (Token)	Percentage
1	PancakeSwap V2: ADABoy 44	17,591,765,408.922960656707579969	17.5918%
2	0xf2bbc71d8c7209d4fac76e72abdc216c0d28c7b7	5,048,748,698.204324386241564139	5.0487%
3	0x656b9875624e5bcece96910e5fb840a90386840b	3,408,519,454.566849817546685562	3.4085%
4	0x46066de6115db656bc215a53c19c5eb0a807eb4c	3,350,150,028.181101183375396682	3.3502%
5	0xc8ffcb1a22132357c6380eb7436cb6fcd7f182e7	2,948,000,000	2.9480%
6	0x3ac10927b78fae299c37b6dadfb3a09a4e340924	2,922,335,928.198470844085341955	2.9223%
7	Burn Address	2,816,000,000	2.8160%
8	0x523037cad3952e9dc5facd5d8198754dfacc07cc	2,689,964,699.418913309245252763	2.6900%
9	0x1a3ca897cf4ad055ccc1b303335500481c774732	2,042,072,416.657829250332636638	2.0421%
10	0xfcdda30937d706497c5c11ea14c039ec2d98024a	1,500,320,956.087683013058714048	1.5003%

Token distribution

Tokens are distributed as follows:



Top 100 Token Holders



The top 100 holders collectively own 86.00% (85,996,691,145.66 Tokens) of ADA BOY

Token Total Supply: 100,000,000,000.00 Token | Total Token Holders: 1,344

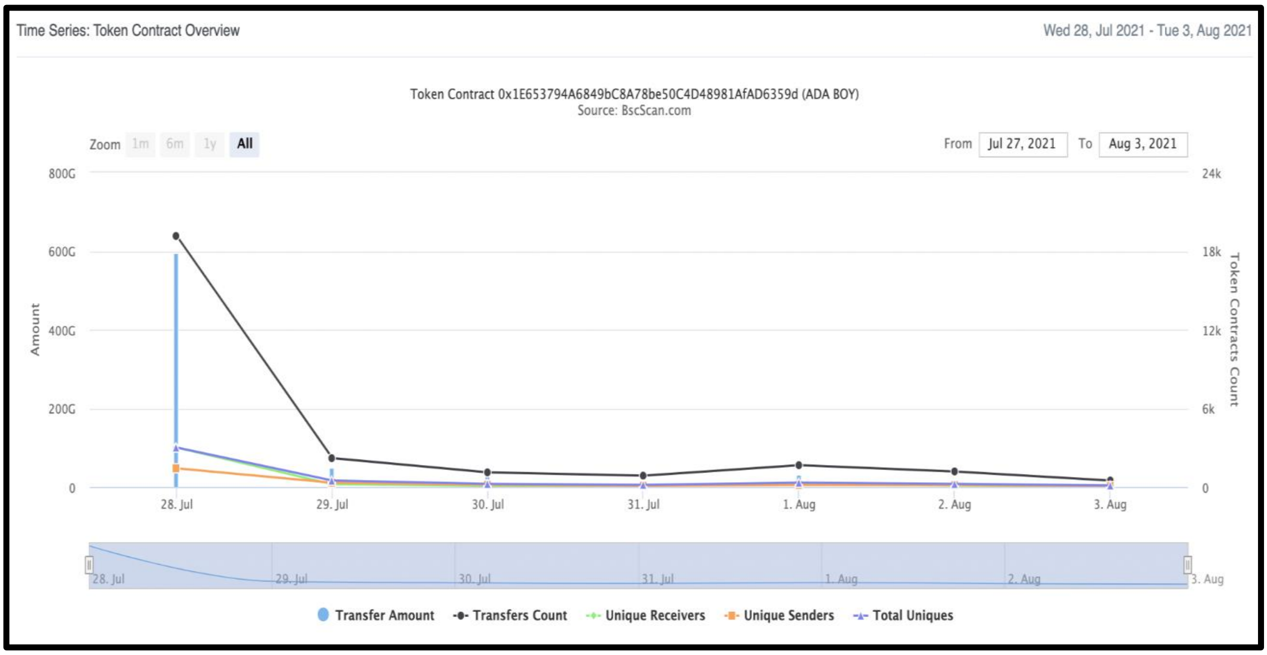
ADA BOY Top 100 Token Holders

Source: BscScan.com

OTHER ACCOUNTS
0xe99f4406d8b94b12f8e3dded72a732a203603535 (PancakeSwap V2: ADABoy 44)
0xf2bbc71d8c7209d4fac76e72abdc216c0d28c7b7
0x656b9875624e5bcece96910e5fb840a90386840b
0x46066de6115db656bc215a53c19c5eb0a807eb4c
0xc8ffcb1a22132357c6380eb7436cb6fcd7f182e7
0x3ac10927b78fae299c37b6dadfb3a09a4e340924
0x000000000000000000000000000000000000dead (Burn Address)
0x523037cad3952e9dc5facd5d8198754dfacc07cc
0x1a3ca897cf4ad055ccc1b303335500481c774732
0xfcdda30937d706497c5c11ea14c039ec2d98024a
0x6f01faf0aad679d3f77e1af3520671fca6c143c4
0x825921903ad6522680cf030e0c1af4e25a0f0ebe
0x1878e4e5d50ae3c66d5944a25175814b51cb73ee
0xd0158b133265b8eed45c59100eb89a8ab02980c3

(A total of 85,996,691,145.66 tokens held by the top 100 accounts from the total supply of 100,000,000,000.00 token)

Contract interaction details



Time Series: Token Contract Overview

Wed 28, Jul 2021 - Tue 3, Aug 2021

Token Contract 0x1E653794A6849bC8A78be50C4D48981AfAD6359d (ADA BOY)

Source: BscScan.com

From Jul 27, 2021 To Aug 3, 2021

Contract code function details

No	Category	Item	Result
1	Coding conventions	BRC20 Token standards	pass
		compile errors	pass
		Compiler version security	pass
		visibility specifiers	pass
		Gas consumption	pass
		SafeMath features	pass
		Fallback usage	pass
		tx.origin usage	pass
		deprecated items	pass
		Redundant code	pass
		Overriding variables	pass
2	Function call audit	Authorization of function call	pass
		Low level function (call/delegate call) security	pass
		Returned value security	pass
		Selfdestruct function security	pass
3	Business security	Access control of owners	pass
		Business logics	pass
		Business implementations	pass
4	Integer overflow/underflow		pass
5	Reentrancy		pass
6	Exceptional reachable state		pass
7	Transaction ordering dependence		pass
8	Block properties dependence		pass
9	Pseudo random number generator (PRNG)		pass
10	DoS (Denial of Service)		pass
11	Token vesting implementation		pass
12	Fake deposit		pass
13	Event security		pass

Contract description table

Below table represents the summary of the contracts and methods in the token contract. We scanned the whole contract and listed down all the Interfaces, functions and implementations with its visibility and mutability.

Contract	**Type**	**Bases**		
└	Function Name	Visibility	Mutability	Modifiers
ADABOYTOKEN	**Implementation**	**ERC20, Ownable**		
└		Public ❗️	🛑	ERC20
└		External ❗️	💵	NO❗️
└	updateDividendTracker	Public ❗️	🛑	onlyOwner
└	updateUniswapV2Router	Public ❗️	🛑	onlyOwner
└	excludeFromFees	Public ❗️	🛑	onlyOwner
└	excludeMultipleAccountsFromFees	Public ❗️	🛑	onlyOwner
└	setMarketingWallet	External ❗️	🛑	onlyOwner
└	setADARewardsFee	External ❗️	🛑	onlyOwner
└	setLiquiditFee	External ❗️	🛑	onlyOwner
└	setMarketingFee	External ❗️	🛑	onlyOwner
└	startTrading	External ❗️	🛑	onlyOwner
└	setAutomatedMarketMakerPair	Public ❗️	🛑	onlyOwner
└	blacklistAddress	External ❗️	🛑	onlyOwner
└	_setAutomatedMarketMakerPair	Private 🔐	🛑	
└	updateGasForProcessing	Public ❗️	🛑	onlyOwner

└	updateClaimWait	External ❗️	🛑	onlyOwner
└	getClaimWait	External ❗️		NO❗️
└	getTotalDividendsDistributed	External ❗️		NO❗️
└	isExcludedFromFees	Public ❗️		NO❗️
└	withdrawableDividendOf	Public ❗️		NO❗️
└	dividendTokenBalanceOf	Public ❗️		NO❗️
└	excludeFromDividends	External ❗️	🛑	onlyOwner
└	getAccountDividendsInfo	External ❗️		NO❗️
└	getAccountDividendsInfoAtIndex	External ❗️		NO❗️
└	processDividendTracker	External ❗️	🛑	NO❗️
└	claim	External ❗️	🛑	NO❗️
└	getLastProcessedIndex	External ❗️		NO❗️
└	getNumberOfDividendTokenHolders	External ❗️		NO❗️
└	_transfer	Internal 🔒	🛑	
└	swapAndSendToFee	Private 🔐	🛑	
└	swapAndLiquify	Private 🔐	🛑	
└	swapTokensForEth	Private 🔐	🛑	
└	swapTokensForADA	Private 🔐	🛑	
└	addLiquidity	Private 🔐	🛑	
└	swapAndSendDividends	Private 🔐	🛑	

ADABoyDividendTracker	**Implementation**	**Ownable, DividendPayingToken**		
└		Public ❗️	🛑	DividendPayingToken
└	_transfer	Internal 🔒	🛑	
└	withdrawDividend	Public ❗️	🛑	NO❗️
└	excludeFromDividends	External ❗️	🛑	onlyOwner
└	updateClaimWait	External ❗️	🛑	onlyOwner
└	getLastProcessedIndex	External ❗️		NO❗️
└	getNumberOfTokenHolders	External ❗️		NO❗️
└	getAccount	Public ❗️		NO❗️
└	getAccountAtIndex	Public ❗️		NO❗️
└	canAutoClaim	Private 🔐		
└	setBalance	External ❗️	🛑	onlyOwner
└	process	Public ❗️	🛑	NO❗️
└	processAccount	Public ❗️	🛑	onlyOwner
DividendPayingToken	**Implementation**	**ERC20, Ownable, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface**		
└		Public ❗️	🛑	ERC20
└	distributeADADividends	Public ❗️	🛑	onlyOwner
└	withdrawDividend	Public ❗️	🛑	NO❗️

└	_withdrawDividendOfUser	Internal 🔒	🛑	
└	dividendOf	Public ❗		NO❗
└	withdrawableDividendOf	Public ❗		NO❗
└	withdrawnDividendOf	Public ❗		NO❗
└	accumulativeDividendOf	Public ❗		NO❗
└	_transfer	Internal 🔒	🛑	
└	_mint	Internal 🔒	🛑	
└	_burn	Internal 🔒	🛑	
└	_setBalance	Internal 🔒	🛑	
ERC20	**Implementation**	**Context, IERC20, IERC20Metadata**		
└		Public ❗	🛑	NO❗
└	name	Public ❗		NO❗
└	symbol	Public ❗		NO❗
└	decimals	Public ❗		NO❗
└	totalSupply	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	transfer	Public ❗	🛑	NO❗
└	allowance	Public ❗		NO❗
└	approve	Public ❗	🛑	NO❗
└	transferFrom	Public ❗	🛑	NO❗
└	increaseAllowance	Public ❗	🛑	NO❗
└	decreaseAllowance	Public ❗	🛑	NO❗

└	_transfer	Internal 🔒	🛑	
└	_mint	Internal 🔒	🛑	
└	_burn	Internal 🔒	🛑	
└	_approve	Internal 🔒	🛑	
└	_beforeTokenTransfer	Internal 🔒	🛑	
IERC20	**Interface**			
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	transfer	External ❗	🛑	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
IERC20Metadata	**Interface**	**IERC20**		
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
Context	**Implementation**			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
SafeMath	**Library**			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	sub	Internal 🔒		

└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	mod	Internal 🔒		
SafeMathUint	**Library**			
└	toInt256Safe	Internal 🔒		
SafeMathInt	**Library**			
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	sub	Internal 🔒		
└	add	Internal 🔒		
└	abs	Internal 🔒		
└	toUint256Safe	Internal 🔒		
DividendPayingTokenInterface	**Interface**			
└	dividendOf	External ❗		NO❗
└	withdrawDividend	External ❗	🛑	NO❗
DividendPayingTokenOptionalInterface	**Interface**			
└	withdrawableDividendOf	External ❗		NO❗
└	withdrawnDividendOf	External ❗		NO❗
└	accumulativeDividendOf	External ❗		NO❗

Ownable	**Implementation**	**Context**		
└		Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
IterableMapping	**Library**			
└	get	Public ❗		NO❗
└	getIndexOfKey	Public ❗		NO❗
└	getKeyAtIndex	Public ❗		NO❗
└	size	Public ❗		NO❗
└	set	Public ❗	🛑	NO❗
└	remove	Public ❗	🛑	NO❗
IUniswapV2Pair	**Interface**			
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transfer	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
└	DOMAIN_SEPARATOR	External ❗		NO❗

└	PERMIT_TYPEHASH	External ❗		NO❗
└	nonces	External ❗		NO❗
└	permit	External ❗	🛑	NO❗
└	MINIMUM_LIQUIDITY	External ❗		NO❗
└	factory	External ❗		NO❗
└	token0	External ❗		NO❗
└	token1	External ❗		NO❗
└	getReserves	External ❗		NO❗
└	price0CumulativeLast	External ❗		NO❗
└	price1CumulativeLast	External ❗		NO❗
└	kLast	External ❗		NO❗
└	mint	External ❗	🛑	NO❗
└	burn	External ❗	🛑	NO❗
└	swap	External ❗	🛑	NO❗
└	skim	External ❗	🛑	NO❗
└	sync	External ❗	🛑	NO❗
└	initialize	External ❗	🛑	NO❗
IUniswapV2Factory	**Interface**			
└	feeTo	External ❗		NO❗
└	feeToSetter	External ❗		NO❗
└	getPair	External ❗		NO❗
└	allPairs	External ❗		NO❗
└	allPairsLength	External ❗		NO❗
└	createPair	External ❗	🛑	NO❗

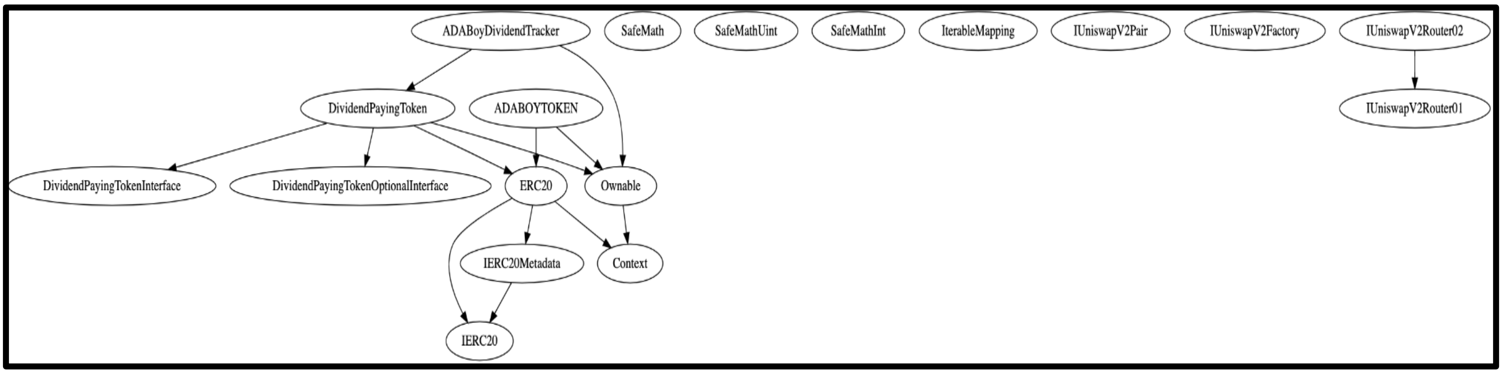
└	setFeeTo	External ❗️	🛑	NO❗️
└	setFeeToSetter	External ❗️	🛑	NO❗️
IUniswapV2Router01	**Interface**			
└	factory	External ❗️		NO❗️
└	WETH	External ❗️		NO❗️
└	addLiquidity	External ❗️	🛑	NO❗️
└	addLiquidityETH	External ❗️	💵	NO❗️
└	removeLiquidity	External ❗️	🛑	NO❗️
└	removeLiquidityETH	External ❗️	🛑	NO❗️
└	removeLiquidityWithPermit	External ❗️	🛑	NO❗️
└	removeLiquidityETHWithPermit	External ❗️	🛑	NO❗️
└	swapExactTokensForTokens	External ❗️	🛑	NO❗️
└	swapTokensForExactTokens	External ❗️	🛑	NO❗️
└	swapExactETHForTokens	External ❗️	💵	NO❗️
└	swapTokensForExactETH	External ❗️	🛑	NO❗️
└	swapExactTokensForETH	External ❗️	🛑	NO❗️
└	swapETHForExactTokens	External ❗️	💵	NO❗️
└	quote	External ❗️		NO❗️
└	getAmountOut	External ❗️		NO❗️
└	getAmountIn	External ❗️		NO❗️
└	getAmountsOut	External ❗️		NO❗️
└	getAmountsIn	External ❗️		NO❗️

IUniswapV2Router02	**Interface**	**IUniswapV2Router01**		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗

Legend

Symbol	Meaning
🛑	**Function can modify state**
💵	**Function is payable**

Inheritance Hierarchy



Security issue checking status

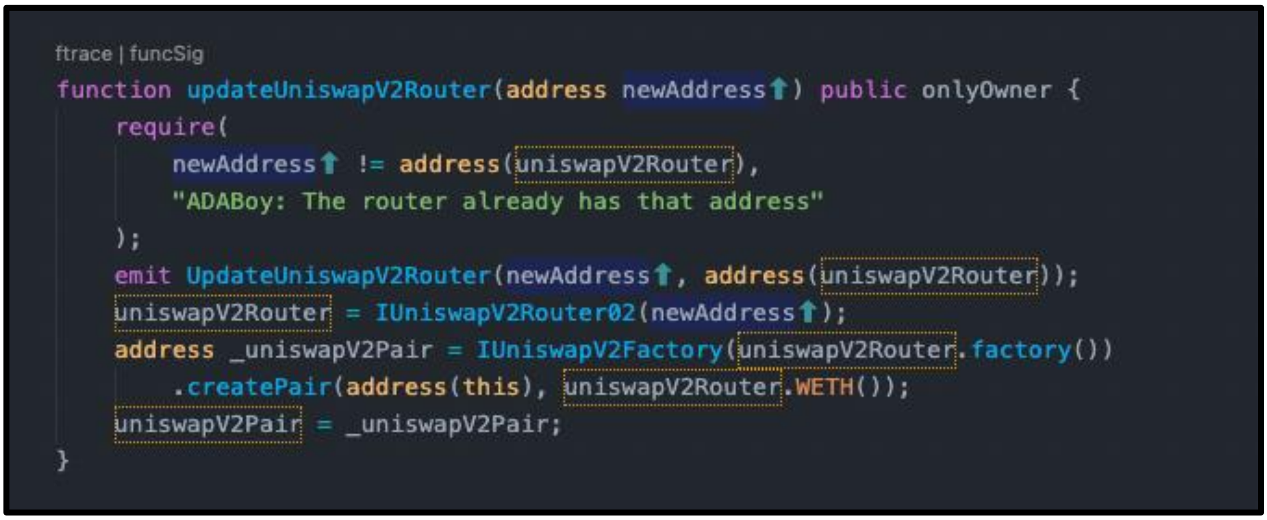
- **High severity issues**
  No high severity issues found.

- **Medium severity issues**
  No medium severity issues found.

- **Low severity issues**
  No low severity issues found.

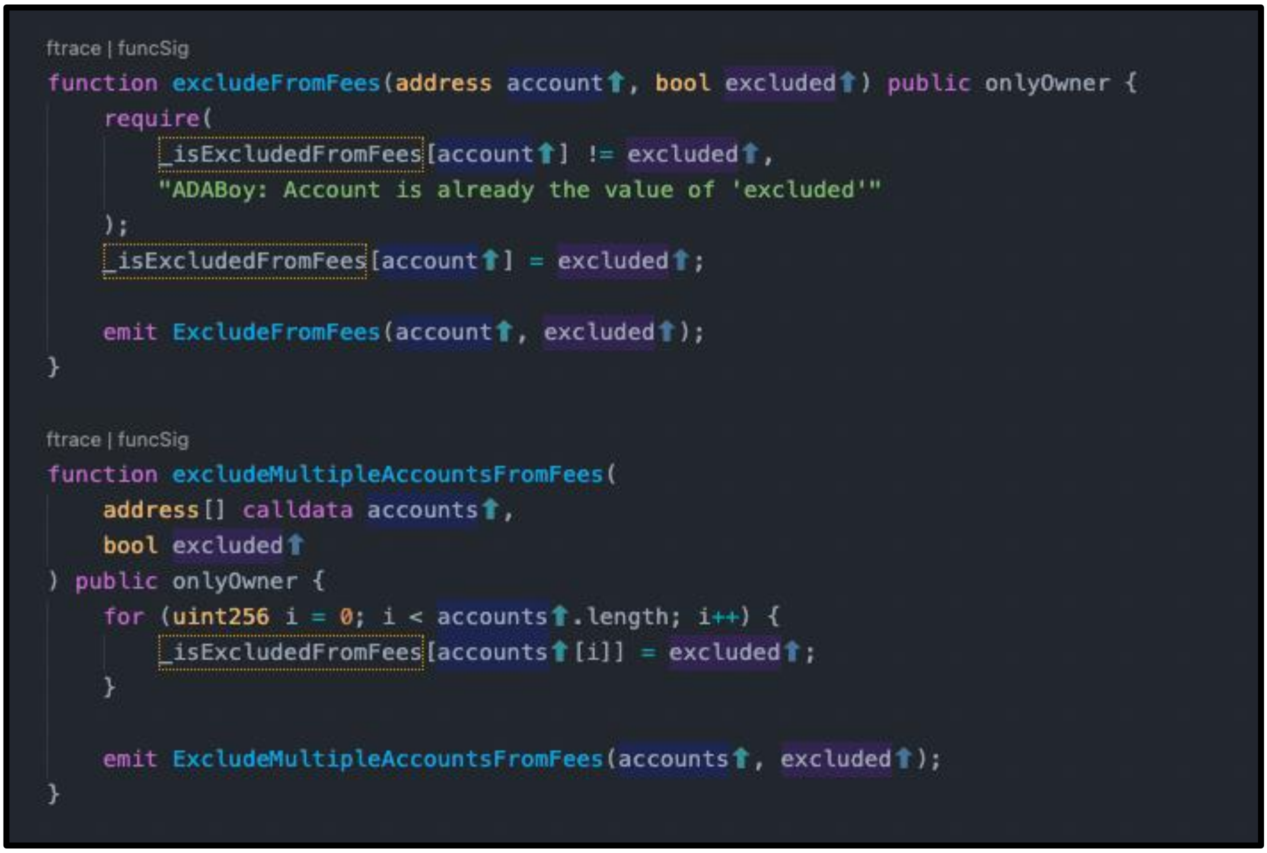
Owner privileges

- ❖ The owner can change the router address.



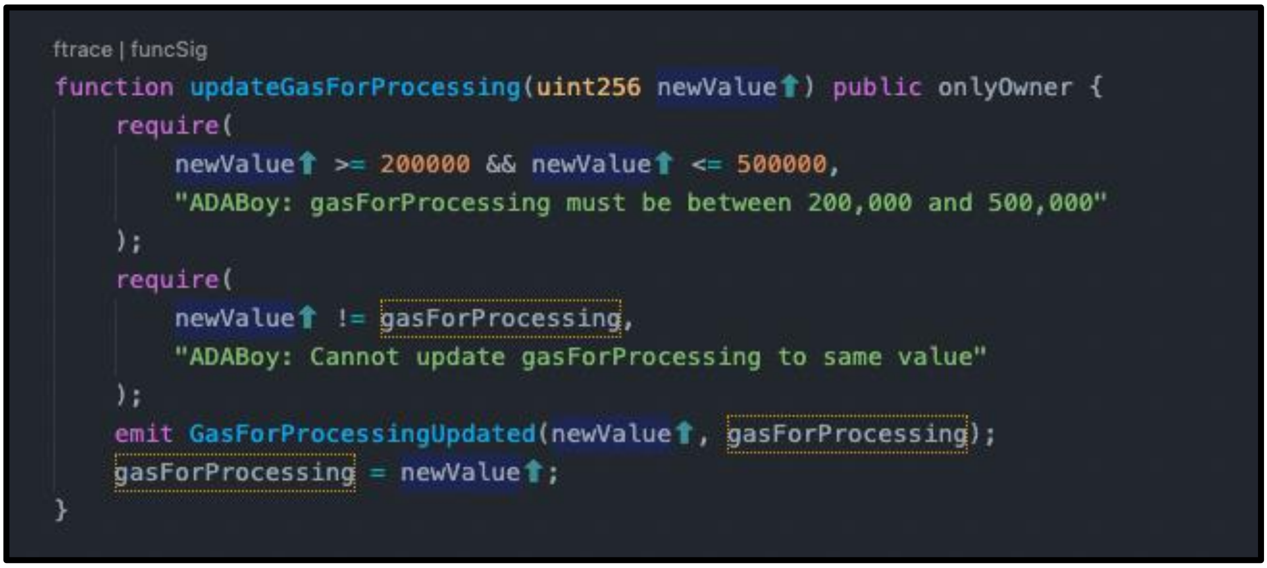
```
ftrace | funcSig
function updateUniswapV2Router(address newAddress) public onlyOwner {
    require(
        newAddress != address(uniswapV2Router),
        "ADABoy: The router already has that address"
    );
    emit UpdateUniswapV2Router(newAddress, address(uniswapV2Router));
    uniswapV2Router = IUniswapV2Router02(newAddress);
    address _uniswapV2Pair = IUniswapV2Factory(uniswapV2Router.factory())
        .createPair(address(this), uniswapV2Router.WETH());
    uniswapV2Pair = _uniswapV2Pair;
}
```

- ❖ The owner can include/exclude accounts from fees.



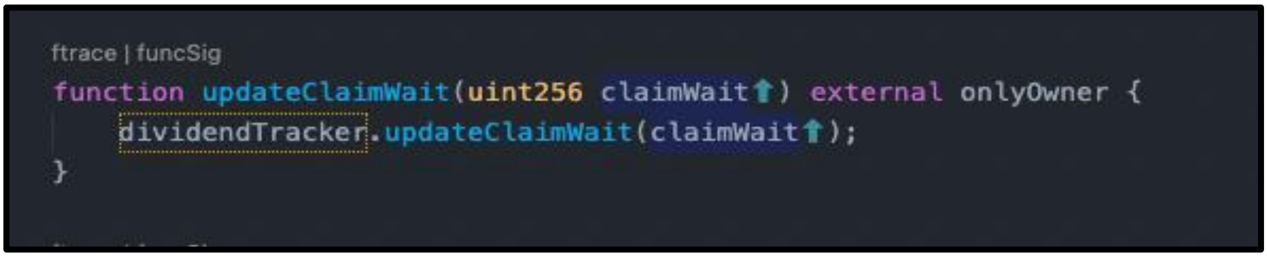
```
ftrace | funcSig
function excludeFromFees(address account, bool excluded) public onlyOwner {
    require(
        _isExcludedFromFees[account] != excluded,
        "ADABoy: Account is already the value of 'excluded'"
    );
    _isExcludedFromFees[account] = excluded;

    emit ExcludeFromFees(account, excluded);
}

ftrace | funcSig
function excludeMultipleAccountsFromFees(
    address[] calldata accounts,
    bool excluded
) public onlyOwner {
    for (uint256 i = 0; i < accounts.length; i++) {
        _isExcludedFromFees[accounts[i]] = excluded;
    }

    emit ExcludeMultipleAccountsFromFees(accounts, excluded);
}
```

- The owner can change the gas fee up to minimum 200,000 and max 500,000.



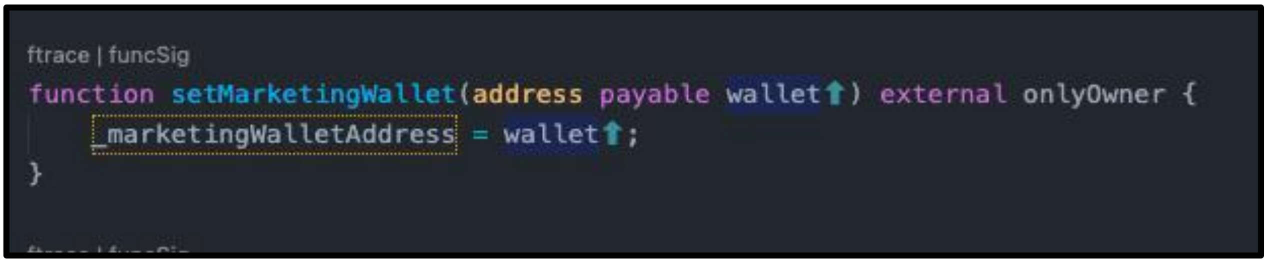
```
ftrace | funcSig
function updateGasForProcessing(uint256 newValue) public onlyOwner {
    require(
        newValue >= 200000 && newValue <= 500000,
        "ADABoy: gasForProcessing must be between 200,000 and 500,000"
    );
    require(
        newValue != gasForProcessing,
        "ADABoy: Cannot update gasForProcessing to same value"
    );
    emit GasForProcessingUpdated(newValue, gasForProcessing);
    gasForProcessing = newValue;
}
```

- The owner can change the claim wait time.



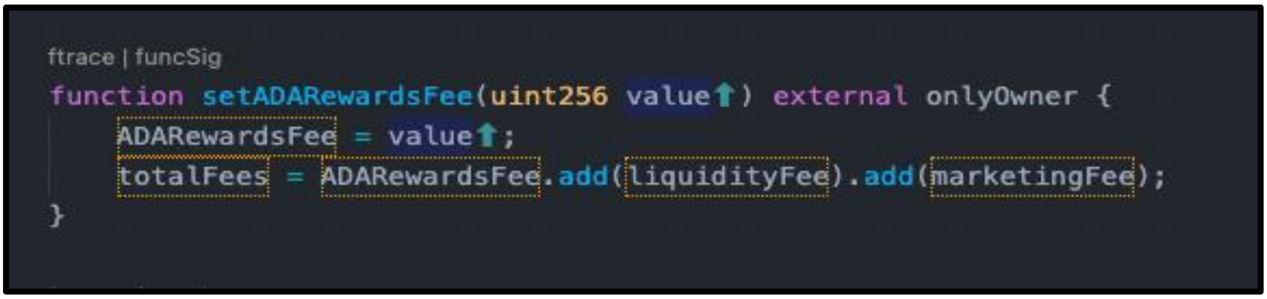
```
ftrace | funcSig
function updateClaimWait(uint256 claimWait) external onlyOwner {
    dividendTracker.updateClaimWait(claimWait);
}
```

- The owner can change the marketing wallet address.



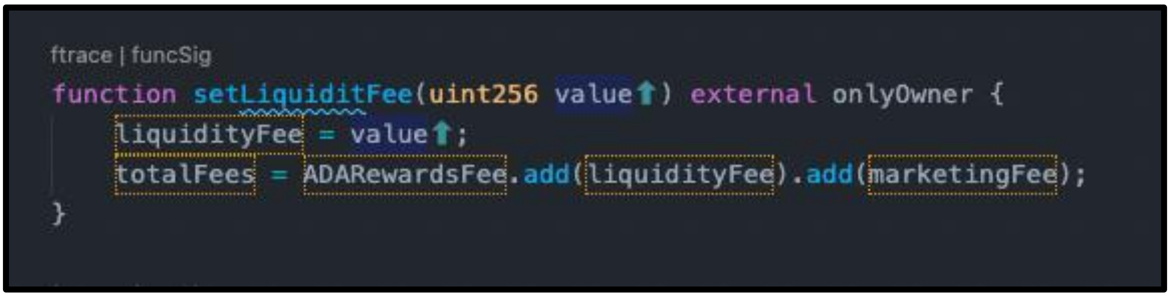
```
ftrace | funcSig
function setMarketingWallet(address payable wallet) external onlyOwner {
    _marketingWalletAddress = wallet;
}
```

- The owner can change the ADA reward fee.



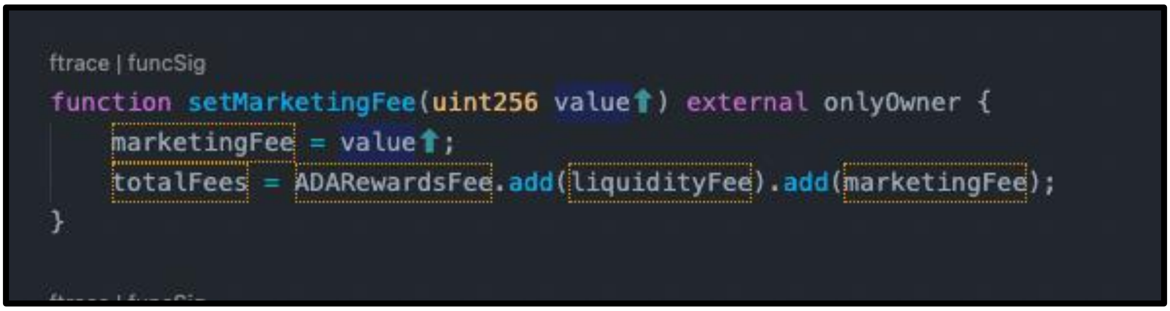
```
ftrace | funcSig
function setADARewardsFee(uint256 value) external onlyOwner {
    ADARewardsFee = value;
    totalFees = ADARewardsFee.add(liquidityFee).add(marketingFee);
}
```

- The owner can change the liquidity fee.



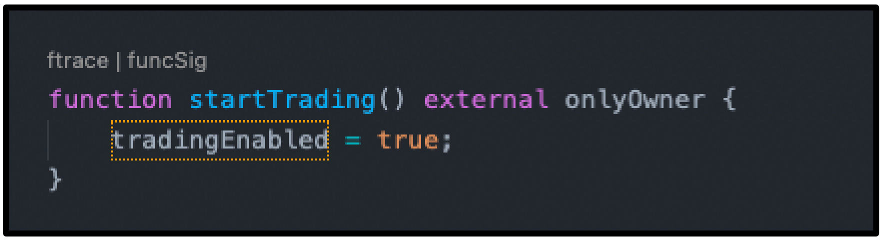
```
ftrace | funcSig
function setLiquiditFee(uint256 value) external onlyOwner {
    liquidityFee = value;
    totalFees = ADARewardsFee.add(liquidityFee).add(marketingFee);
}
```

- The owner can change the marketing fee.



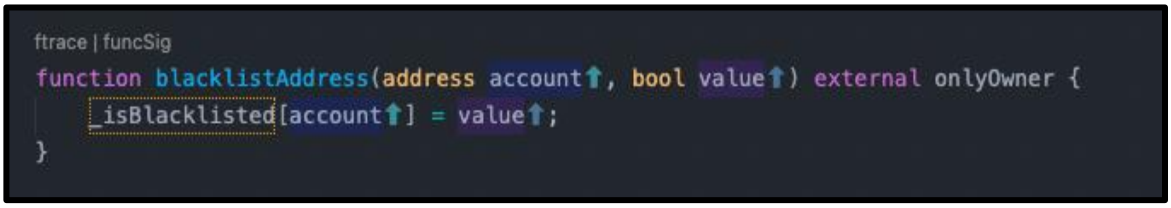
```
ftrace | funcSig
function setMarketingFee(uint256 value) external onlyOwner {
    marketingFee = value;
    totalFees = ADARewardsFee.add(liquidityFee).add(marketingFee);
}
```

- The owner can enable trading.



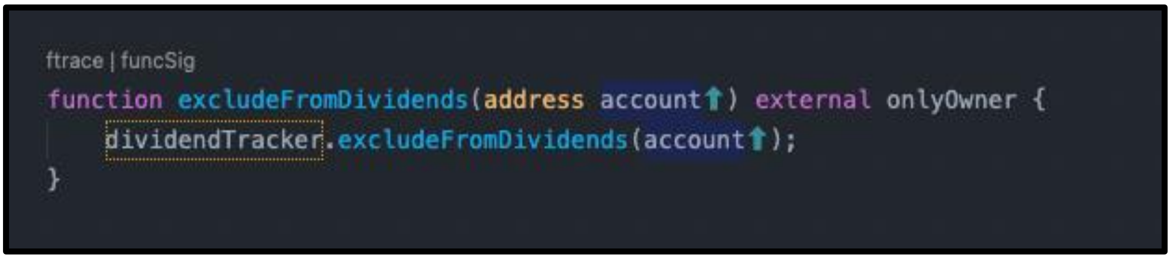
```
ftrace | funcSig
function startTrading() external onlyOwner {
    tradingEnabled = true;
}
```

- The owner can blacklist and whitelist addresses.



```
ftrace | funcSig
function blacklistAddress(address account, bool value) external onlyOwner {
    _isBlacklisted[account] = value;
}
```

- The owner can exclude accounts from dividends.



```
ftrace | funcSig
function excludeFromDividends(address account) external onlyOwner {
    dividendTracker.excludeFromDividends(account);
}
```

Audit conclusion

While conducting the audit of the ADA Boy smart contract, it was observed that there is nothing alarming with the code.