RugFreeCoins Audit

ETH Fan Eco System

Smart Contract Security Audit

February 05, 2022

Contents

Audit details 1

Disclaimer 2

Background 3

About the project 4

Target market and the concept 7

Potential to grow with score points 8

Total Points 8

Contract details 9

Contract code function details 10

Contract description table 11

Security issue checking status 24

Owner privileges 26

Audit conclusion 31

Audit details

Audited project

ETH Fan Eco System Token

Contract Address

0xb72962568345253f71a18318d67e13a282b187e6

Client contact

ETH Fan Eco System Token Team

Blockchain

Binance smart chain

Project website

https://ethfan.club/

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 ETH Fan Eco System Token to perform an audit of the smart contract.

https://bscscan.com/address/0xb72962568345253f71a18318d67e13a282b187e6

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

ETH Fan Ecosystem token is a token built on the Binance Smart Chain that is with an innovative investment use case the main purpose of which is to seek out constant revenue sources, and heading towards building even greater Community, DAO Governed ecosystem, a Multi Chain NFT marketplace, staking platform and an eventual future Game + metaverse. Each transaction, purchase incurs 10% fee, and sale incurs a 14% fee.

Features

- The **ETH Fan Eco System Token rewards** will be distributed in ETH among every holder proportional to how many tokens each individual holds in values of **4% when buying and 5% when selling**.

- The **sustainability fee of 2% when buying and 3% when selling for dev and 3% when buying and selling for marketing** is what allows ETH Fan Eco System Token to hold the aforementioned promise. Tokens will be swapped into BNB and will be sent to a marketing wallet per transaction. This way, ETH Fan Eco System Token will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.

- The additional component included under the sustainability section is a **liquidity fee of 1% when buying and 3% when selling**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

Tokenomics

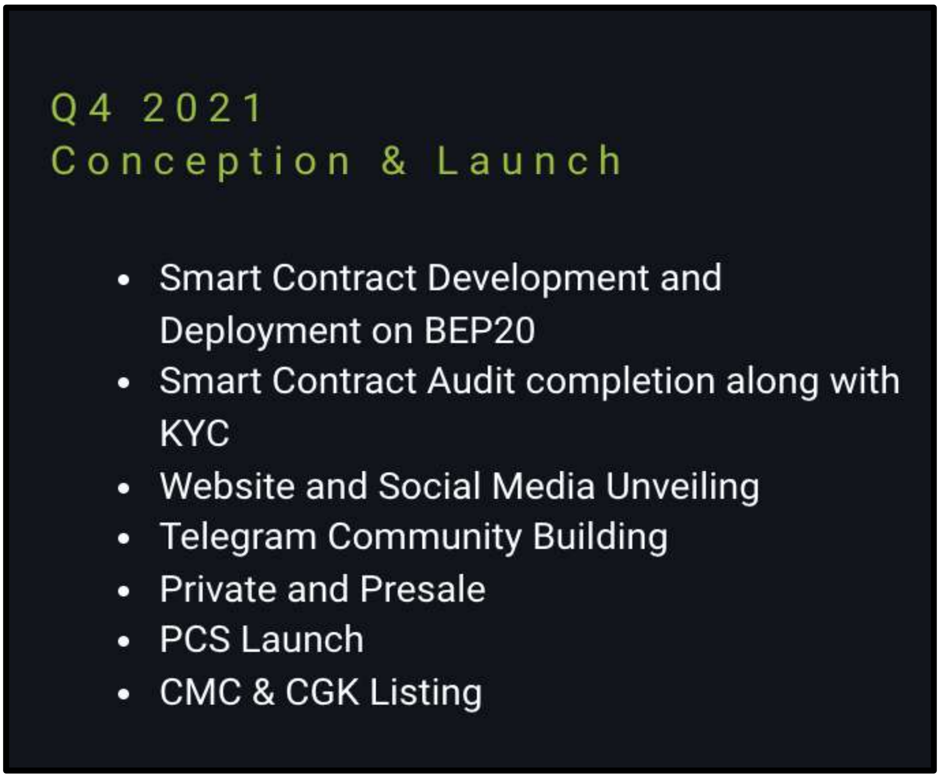
10% fee when buying

- 4% of trade goes to holders' pockets in ETH.
- 2% of trade goes to the development wallet.
- 3% of trade goes to the marketing wallet.
- 1% of trade goes to the liquidity pool.

14% fee when selling

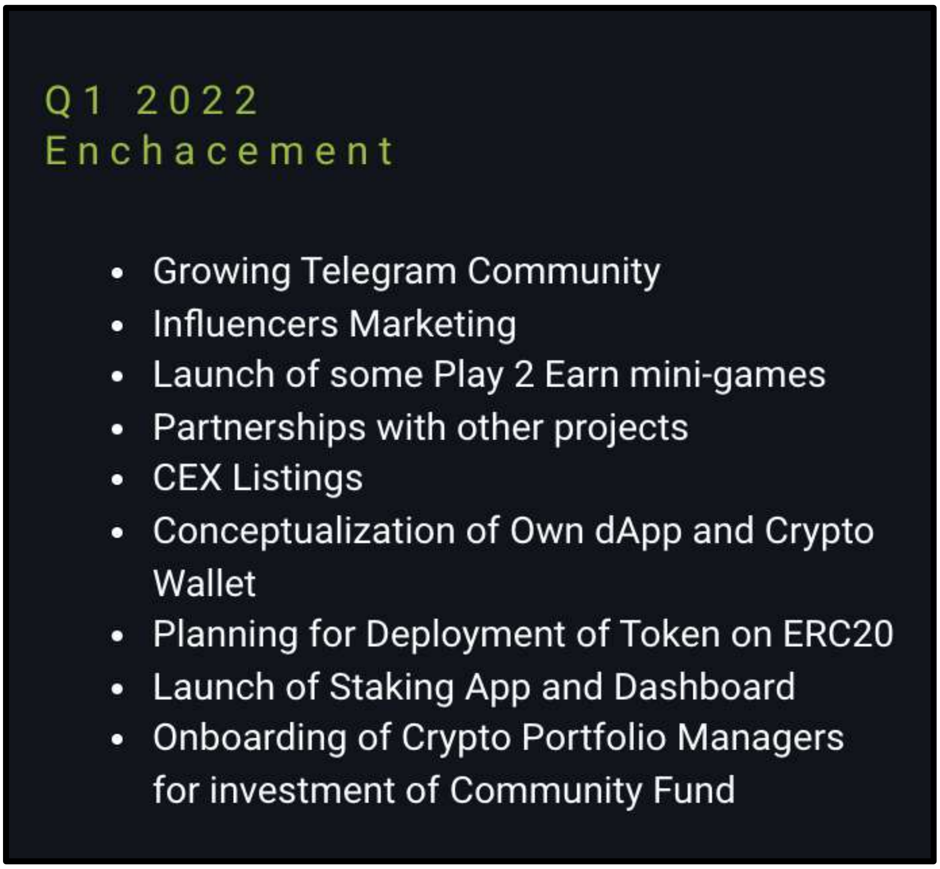
- 5% of trade goes to holders' pockets in ETH.
- 3% of trade goes to the development wallet.
- 3% of trade goes to the marketing wallet.
- 3% of trade goes to the liquidity pool.

Roadmap



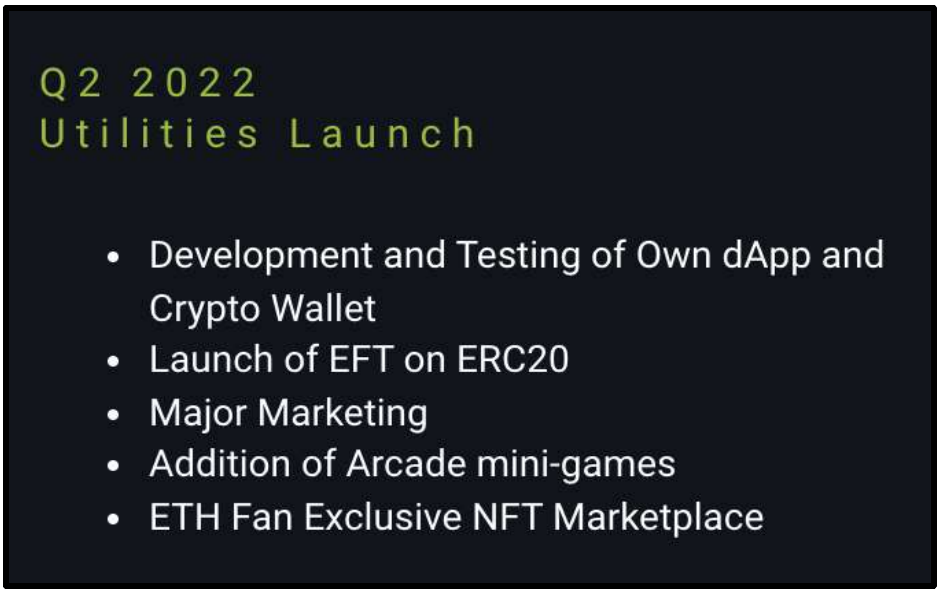
Q4 2021
Conception & Launch

- Smart Contract Development and Deployment on BEP20
- Smart Contract Audit completion along with KYC
- Website and Social Media Unveiling
- Telegram Community Building
- Private and Presale
- PCS Launch
- CMC & CGK Listing



Q1 2022
Enchacement

- Growing Telegram Community
- Influencers Marketing
- Launch of some Play 2 Earn mini-games
- Partnerships with other projects
- CEX Listings
- Conceptualization of Own dApp and Crypto Wallet
- Planning for Deployment of Token on ERC20
- Launch of Staking App and Dashboard
- Onboarding of Crypto Portfolio Managers for investment of Community Fund



Q2 2022
Utilities Launch

- Development and Testing of Own dApp and Crypto Wallet
- Launch of EFT on ERC20
- Major Marketing
- Addition of Arcade mini-games
- ETH Fan Exclusive NFT Marketplace

Target market and the concept

Target market

- Anyone who's interested in the Crypto space with long-term investment plans.
- Anyone who's ready to earn a passive income by holding tokens.
- Anyone who's interested in trading tokens.
- Anyone who's interested in collecting NFTs or trading NFTs.
- Anyone who's interested in taking part with staking.
- Anyone who's interested in taking part of decision making.
- Anyone who's interested in taking part with the future plans of the ETH Fan Token Ecosystem.
- Anyone who's interested in making financial transactions with any other party using ETH Fan Eco System Token as the currency.

Core concept

The ETH Fan Eco System Token reward system

4% of each transaction when buying and 5% when selling gets converted to ETH, and is split amongst all holders. Holders will be eligible to receive tokens in each transaction and rewards are proportional to how many tokens each individual holds.

Sustainable mechanism

The **sustainability fee of 2% when buying and 3% when selling for dev and 3% when buying and selling for marketing** is what allows ETH Fan Eco System Token to promote the token and use funds to further the development of the platform. Tokens will be swapped into BNB and will be sent to a marketing wallet and dev wallet. This way, ETH Fan Eco System Token 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 liquidity fee of 1% when buying and 3% when selling, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

Potential to grow with score points

1.	Project efficiency	10/10
2.	Project uniqueness	9/10
3	Information quality	10/10
4	Service quality	10/10
5	System quality	10/10
6	Impact on the community	10/10
7	Impact on the business	10/10
8	Preparing for the future	10/10
Total Points		**9.875/10**

Contract details

Token contract details for 05th February 2022

Contract name	ETH Fan Token Ecosystem
Contract address	0xb72962568345253f71a18318d67e13a282b187e6
Token supply	1,000,000,000,000,000
Token ticker	EFT
Decimals	18
Token holders	4
Transaction count	15
Development wallet address	0xa65c2c35cc10265bfc1ac83e2a143d21d3787da1
Marketing address	0x54f563a778065f97e1bf8fd0f521502bf9c7d867
Dividend tracker	0x9ee944c13a7f4aa9d69237e9dfc01e58cff37cb5
Contract deployer address	0x374818fA8920228104dcb83D8231Ed81e755a100
Contract’s current owner address	0x374818fa8920228104dcb83d8231ed81e755a100

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	low issue
		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
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 🔒		
ERC20	**Implementation**	**Context, IERC20, IERC20Metadata**		
└		Public ❗	🛑	NO❗
└	name	External ❗		NO❗
└	symbol	External ❗		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 🔒	🛑	
SafeMathUint	**Library**			
└	toInt256Safe	Internal 🔒		
SafeMathInt	**Library**			
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	sub	Internal 🔒		
└	add	Internal 🔒		
└	abs	Internal 🔒		

└	toUint256Safe	Internal 🔒		
DividendPayingToken Interface	**Interface**			
└	dividendOf	External ❗		NO❗
└	withdrawDividend	External ❗	🛑	NO❗
DividendPayingToken OptionalInterface	**Interface**			
└	withdrawableDividendOf	External ❗		NO❗
└	withdrawnDividendOf	External ❗		NO❗
└	accumulativeDividendOf	External ❗		NO❗
Ownable	**Implementation**	**Context**		
└		Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
DividendPayingToken	**Implementation**	**ERC20, Ownable, DividendPayingToken Interface, DividendPayingToken OptionalInterface**		
└		Public ❗	🛑	ERC20

└	distributeETHDividends	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 🔒	🛑	
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❗
EFT	**Implementation**	**ERC20, Ownable**		
└		Public ❗	🛑	ERC20
└		External ❗	💵	NO❗
└	setTradingEnabled	External ❗	🛑	onlyOwner
└	addToBlackList	External ❗	🛑	onlyOwner
└	removeFromBlackList	External ❗	🛑	onlyOwner
└	isBlacklisted	Public ❗		NO❗
└	addToWhitelist	External ❗	🛑	onlyOwner

└	setAutomatedMarketMakerPair	Public ❗️	🛑	onlyOwner
└	updateDividendTracker	Public ❗️	🛑	onlyOwner
└	updateUniswapV2Router	Public ❗️	🛑	onlyOwner
└	excludeOrIncludeFromFees	Public ❗️	🛑	onlyOwner
└	excludeOrIncludeMultipleAccountsFromFees	Public ❗️	🛑	onlyOwner
└	excludeOrIncludeFromMaxTx	Public ❗️	🛑	onlyOwner
└	excludeOrIncludeFromMaxWallet	Public ❗️	🛑	onlyOwner
└	setMaxWalletToken	External ❗️	🛑	onlyOwner
└	setMaxtx	Public ❗️	🛑	onlyOwner
└	setMarketingWallet	External ❗️	🛑	onlyOwner
└	setDevelopmentWallet	External ❗️	🛑	onlyOwner
└	setBuyFees	External ❗️	🛑	onlyOwner
└	setSellFees	External ❗️	🛑	onlyOwner
└	_setAutomatedMarketMakerPair	Private 🔐	🛑	
└	updateGasForProcessing	Public ❗️	🛑	onlyOwner
└	updateClaimWait	External ❗️	🛑	onlyOwner
└	getClaimWait	External ❗️		NO❗️
└	getTotalDividendsDistributed	External ❗️		NO❗️
└	isExcludedFromFees	Public ❗️		NO❗️

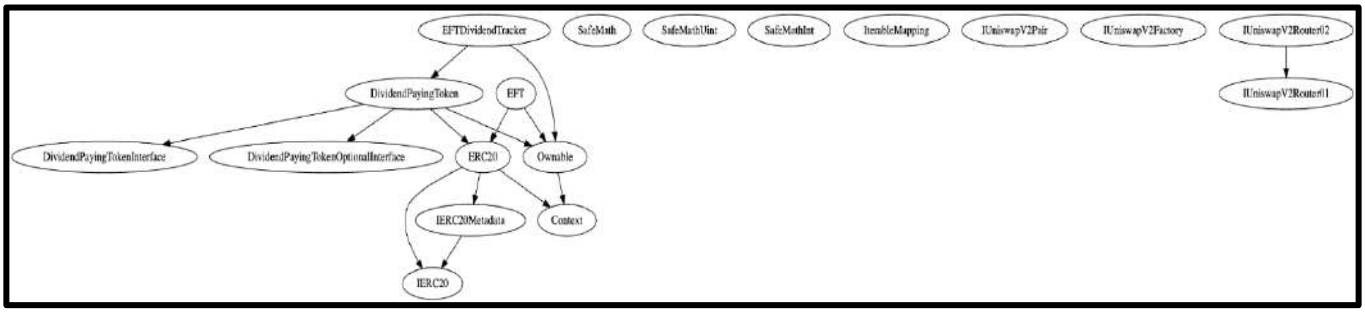
L	isExcludedFromMaxTx	Public ❗️		NO❗️
L	isExcludedFromMaxWallet	Public ❗️		NO❗️
L	withdrawableDividendOf	Public ❗️		NO❗️
L	dividendTokenBalanceOf	Public ❗️		NO❗️
L	excludeFromDividends	External ❗️	🛑	onlyOwner
L	getAccountDividendsInfo	External ❗️		NO❗️
L	getAccountDividendsInfoAtIndex	External ❗️		NO❗️
L	processDividendTracker	External ❗️	🛑	NO❗️
L	claim	External ❗️	🛑	NO❗️
L	getLastProcessedIndex	External ❗️		NO❗️
L	getNumberOfDividendTokenHolders	External ❗️		NO❗️
L	changeSwapTokensAtAmount	External ❗️	🛑	onlyOwner
L	extraFeeOnSell	Internal 🔒		
L	_transfer	Internal 🔒	🛑	
L	swapAndLiquify	Private 🔐	🛑	
L	swapTokensForEth	Private 🔐	🛑	
L	manualSwapTokensForEth	Public ❗️	🛑	onlyOwner
L	swapTokensForETH	Private 🔐	🛑	
L	addLiquidity	Private 🔐	🛑	

L	swapAndSendDividends	Private 🔐	🛑	
L	withdrawStuckBNB	External ❗	🛑	onlyOwner
L	removeStuckToken	External ❗	🛑	onlyOwner
EFTDividendTracker	**Implementation**	**Ownable, DividendPayingToken**		
L		Public ❗	🛑	DividendPayingToken
L	_transfer	Internal 🔒		
L	withdrawDividend	Public ❗		NO❗
L	excludeFromDividends	External ❗	🛑	onlyOwner
L	updateClaimWait	External ❗	🛑	onlyOwner
L	getLastProcessedIndex	External ❗		NO❗
L	getNumberOfTokenHolders	External ❗		NO❗
L	getAccount	Public ❗		NO❗
L	getAccountAtIndex	Public ❗		NO❗
L	canAutoClaim	Private 🔐		
L	setBalance	External ❗	🛑	onlyOwner
L	process	Public ❗	🛑	NO❗
L	processAccount	Public ❗	🛑	onlyOwner

Legend

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

Inheritance Hierarchy



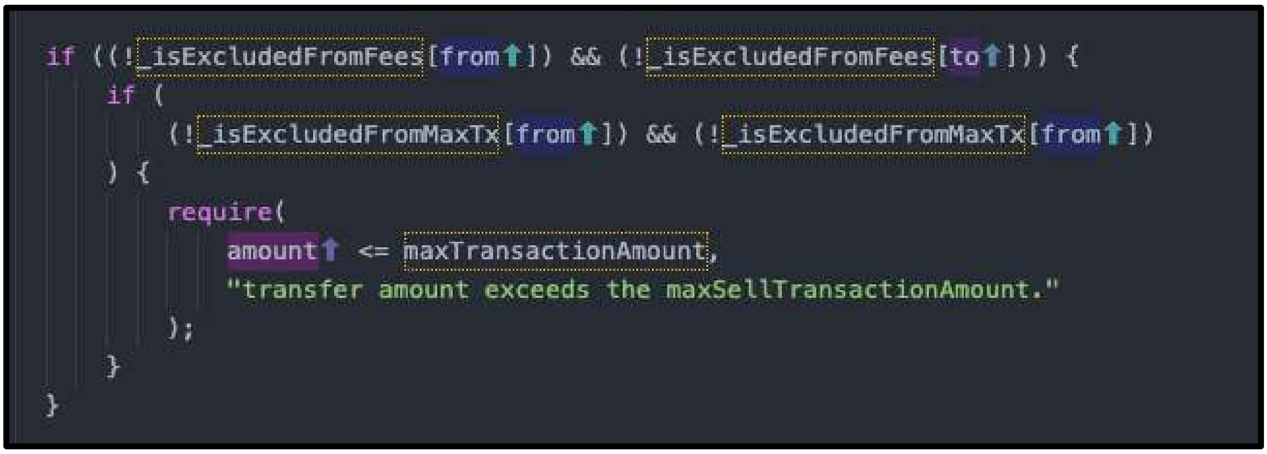
Security issue checking status

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

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

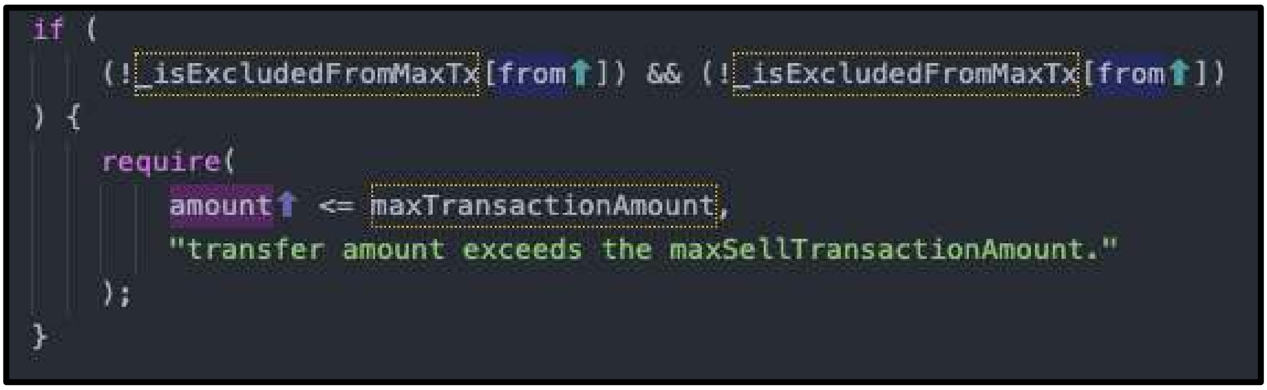
- **Low severity issues**

  - Max transaction limit check only users are not excluding from fees, if they exclude from fees, they will automatically exclude from max transaction limit.



```
if ((!_isExcludedFromFees[from]) && (!_isExcludedFromFees[to])) {
    if (
        (!_isExcludedFromMaxTx[from]) && (!_isExcludedFromMaxTx[from])
    ) {
        require(
            amount <= maxTransactionAmount,
            "transfer amount exceeds the maxSellTransactionAmount."
        );
    }
}
```

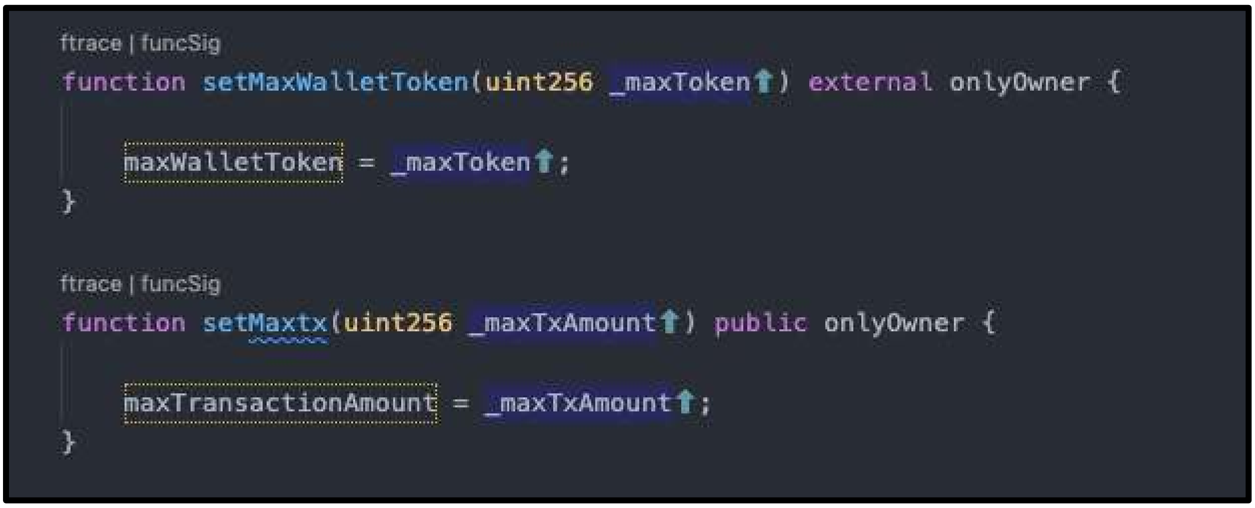
  - Checking the same condition twice.



```
if (
    (!_isExcludedFromMaxTx[from]) && (!_isExcludedFromMaxTx[from])
) {
    require(
        amount <= maxTransactionAmount,
        "transfer amount exceeds the maxSellTransactionAmount."
    );
}
```

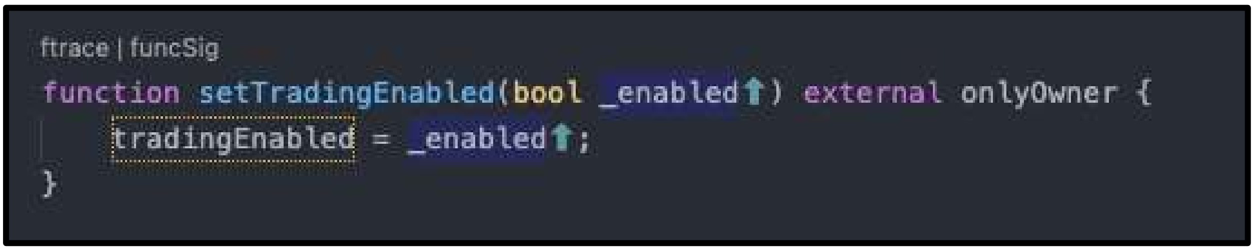
❖ Informational

- The owner can change max wallet and max transaction amount without any limitation (can see to 0)



```
ftrace | funcSig
function setMaxWalletToken(uint256 _maxToken) external onlyOwner {

    maxWalletToken = _maxToken;
}

ftrace | funcSig
function setMaxtx(uint256 _maxTxAmount) public onlyOwner {

    maxTransactionAmount = _maxTxAmount;
}
```

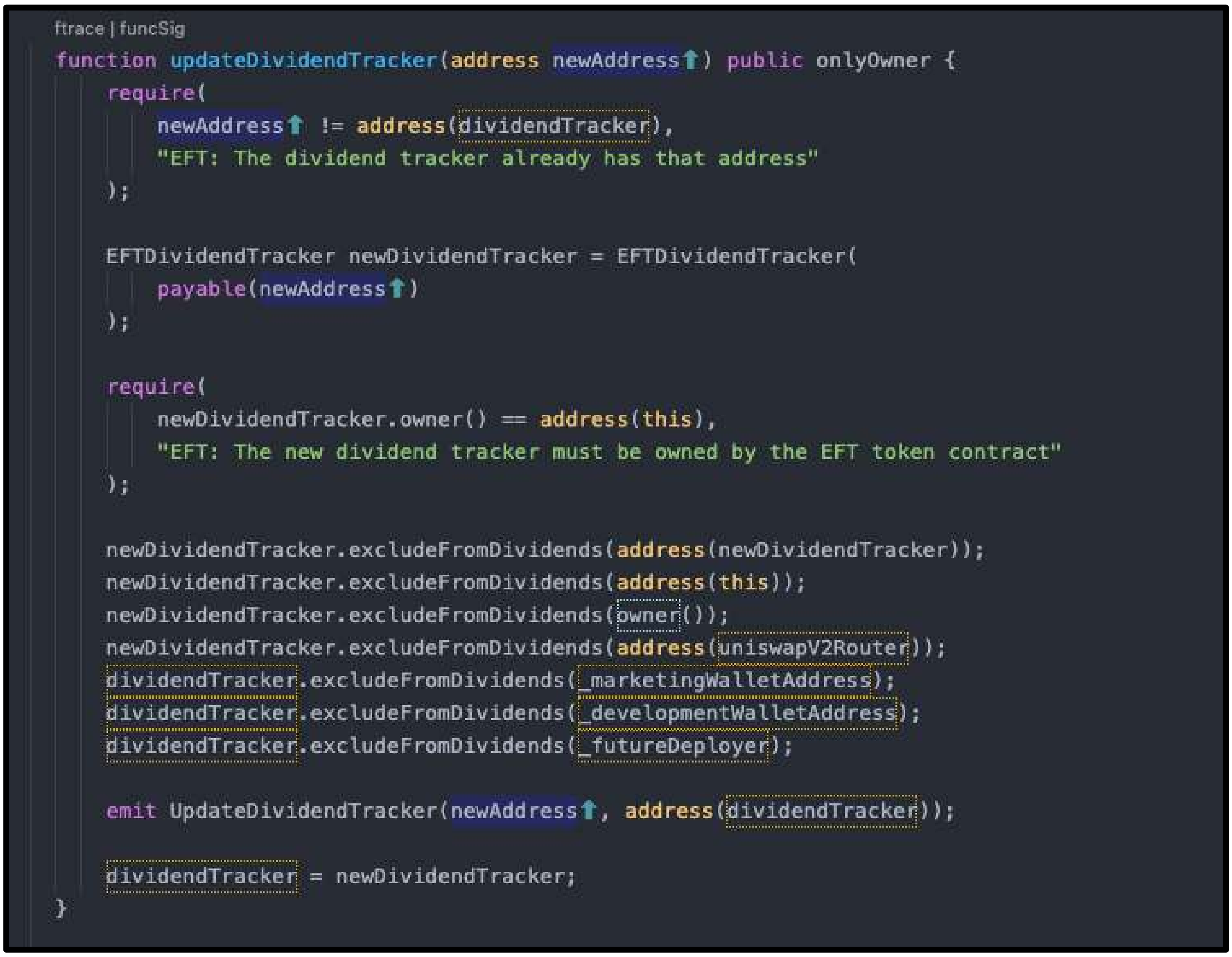
- The owner can disable trading.



```
ftrace | funcSig
function setTradingEnabled(bool _enabled) external onlyOwner {
    tradingEnabled = _enabled;
}
```

Owner privileges

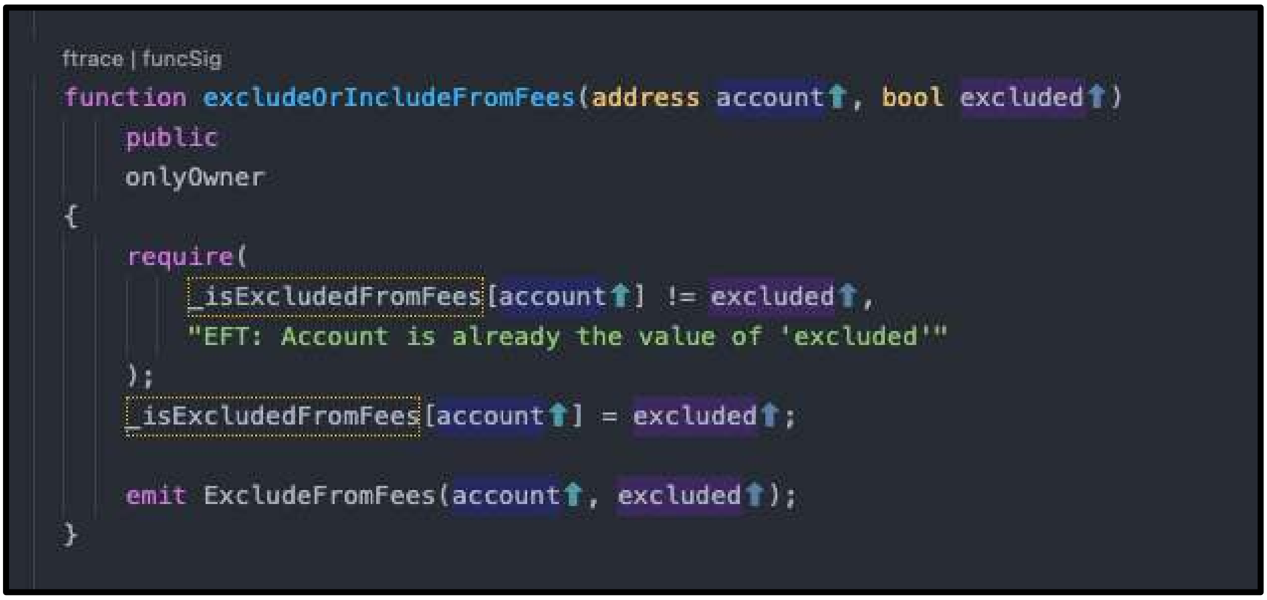
- The owner can update the dividend tracker.



ftrace | funcSig

```
function updateDividendTracker(address newAddress) public onlyOwner {
    require(
        newAddress != address(dividendTracker),
        "EFT: The dividend tracker already has that address"
    );

    EFTDividendTracker newDividendTracker = EFTDividendTracker(
        payable(newAddress)
    );

    require(
        newDividendTracker.owner() == address(this),
        "EFT: The new dividend tracker must be owned by the EFT token contract"
    );

    newDividendTracker.excludeFromDividends(address(newDividendTracker));
    newDividendTracker.excludeFromDividends(address(this));
    newDividendTracker.excludeFromDividends(owner());
    newDividendTracker.excludeFromDividends(address(uniswapV2Router));
    dividendTracker.excludeFromDividends(_marketingWalletAddress);
    dividendTracker.excludeFromDividends(_developmentWalletAddress);
    dividendTracker.excludeFromDividends(_futureDeployer);

    emit UpdateDividendTracker(newAddress, address(dividendTracker));

    dividendTracker = newDividendTracker;
}
```

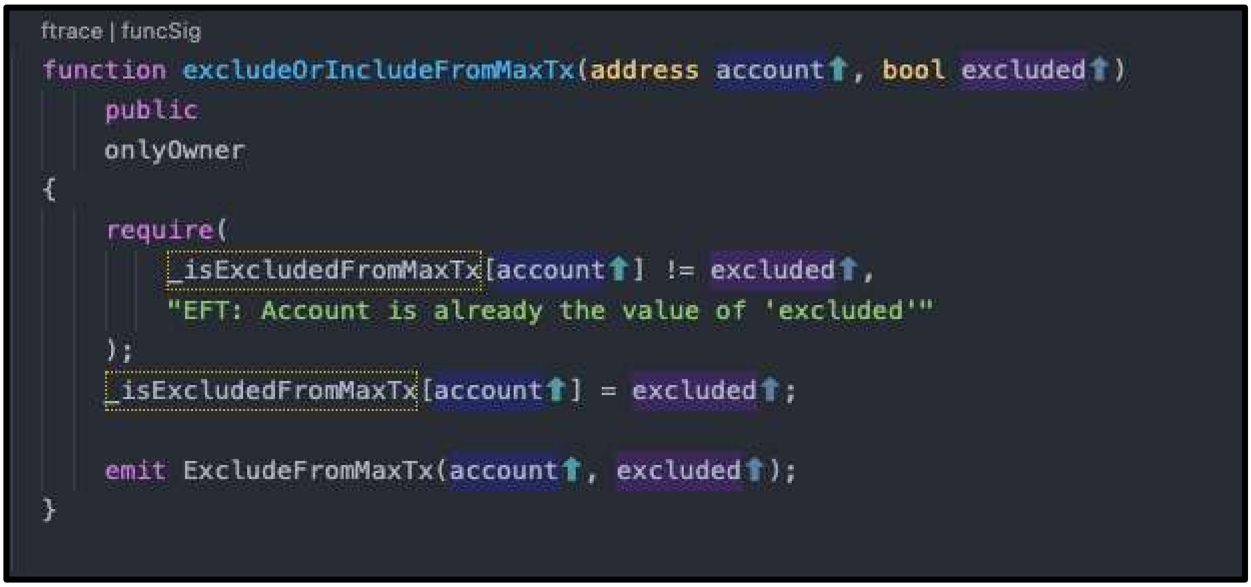
- The owner can include/exclude wallets from fees.



ftrace | funcSig

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

    emit ExcludeFromFees(account, excluded);
}
```

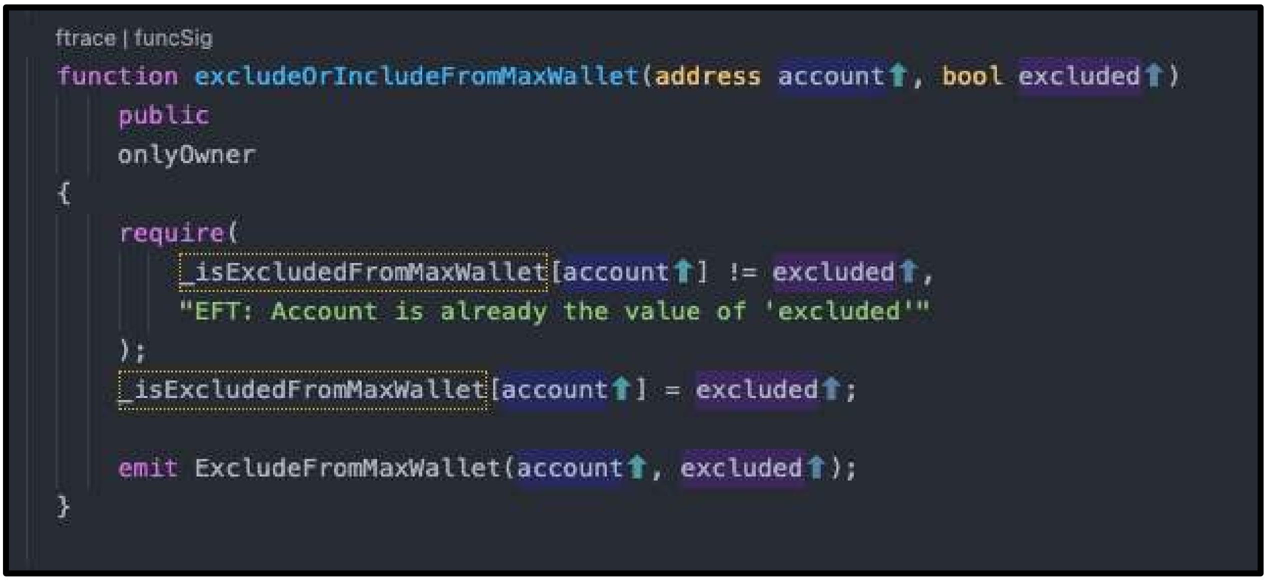
❖ The owner can include/exclude wallets from max transaction limit.



ftrace | funcSig

```
function excludeOrIncludeFromMaxTx(address account, bool excluded)
    public
    onlyOwner
{
    require(
        _isExcludedFromMaxTx[account] != excluded,
        "EFT: Account is already the value of 'excluded'"
    );
    _isExcludedFromMaxTx[account] = excluded;

    emit ExcludeFromMaxTx(account, excluded);
}
```

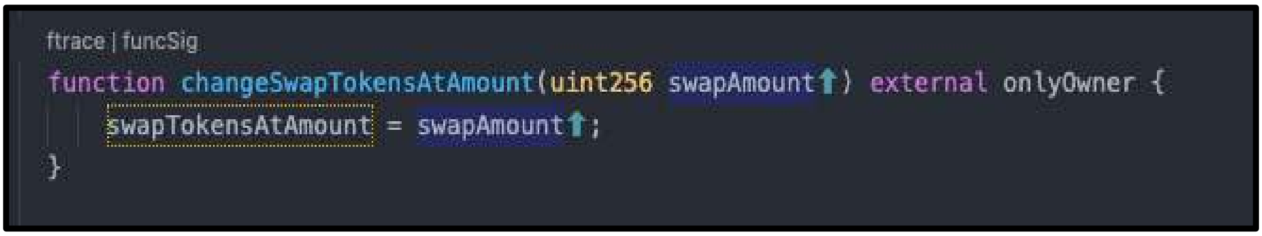
❖ The owner can include/exclude wallet from max wallet.



ftrace | funcSig

```
function excludeOrIncludeFromMaxWallet(address account, bool excluded)
    public
    onlyOwner
{
    require(
        _isExcludedFromMaxWallet[account] != excluded,
        "EFT: Account is already the value of 'excluded'"
    );
    _isExcludedFromMaxWallet[account] = excluded;

    emit ExcludeFromMaxWallet(account, excluded);
}
```

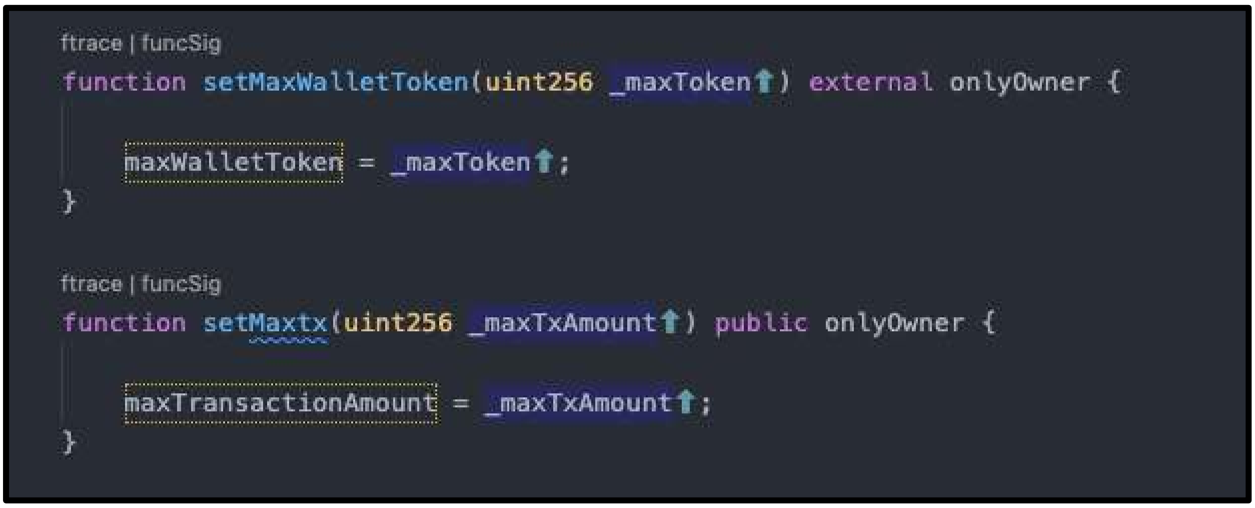
❖ The owner can change the swap point.



ftrace | funcSig

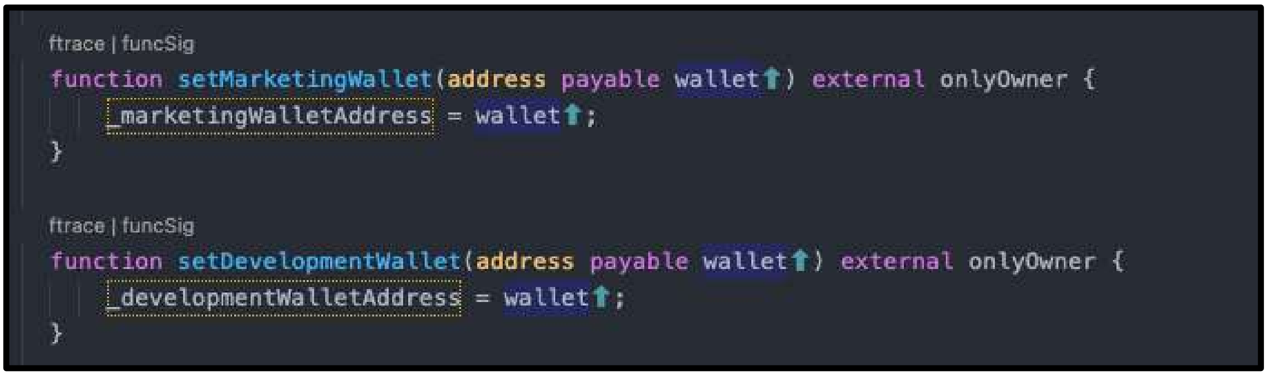
```
function changeSwapTokensAtAmount(uint256 swapAmount) external onlyOwner {
    swapTokensAtAmount = swapAmount;
}
```

- The owner can change max wallet and max transaction token amount.



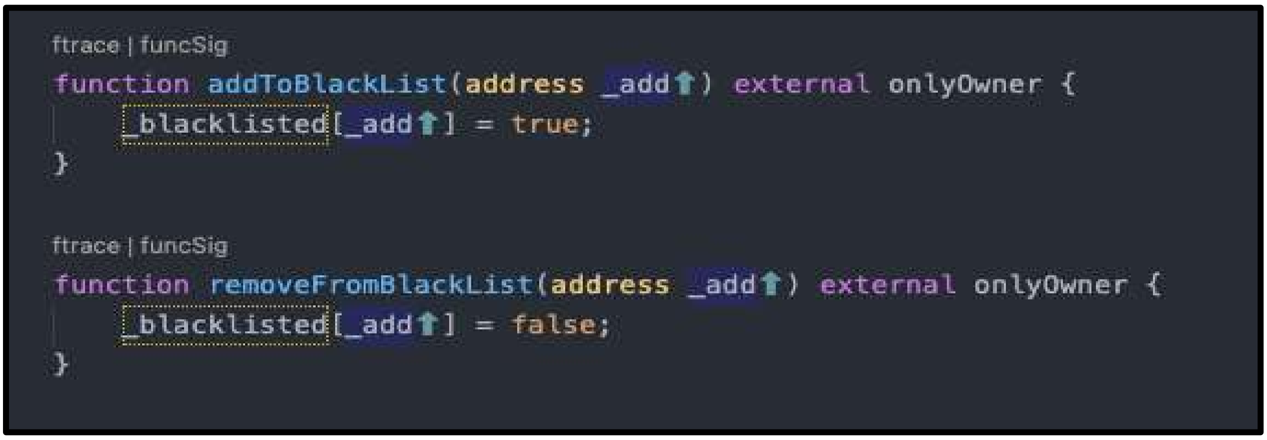
```
ftrace | funcSig
function setMaxWalletToken(uint256 _maxToken) external onlyOwner {

    maxWalletToken = _maxToken;
}

ftrace | funcSig
function setMaxtx(uint256 _maxTxAmount) public onlyOwner {

    maxTransactionAmount = _maxTxAmount;
}
```

- The owner can change marketing and dev wallet.



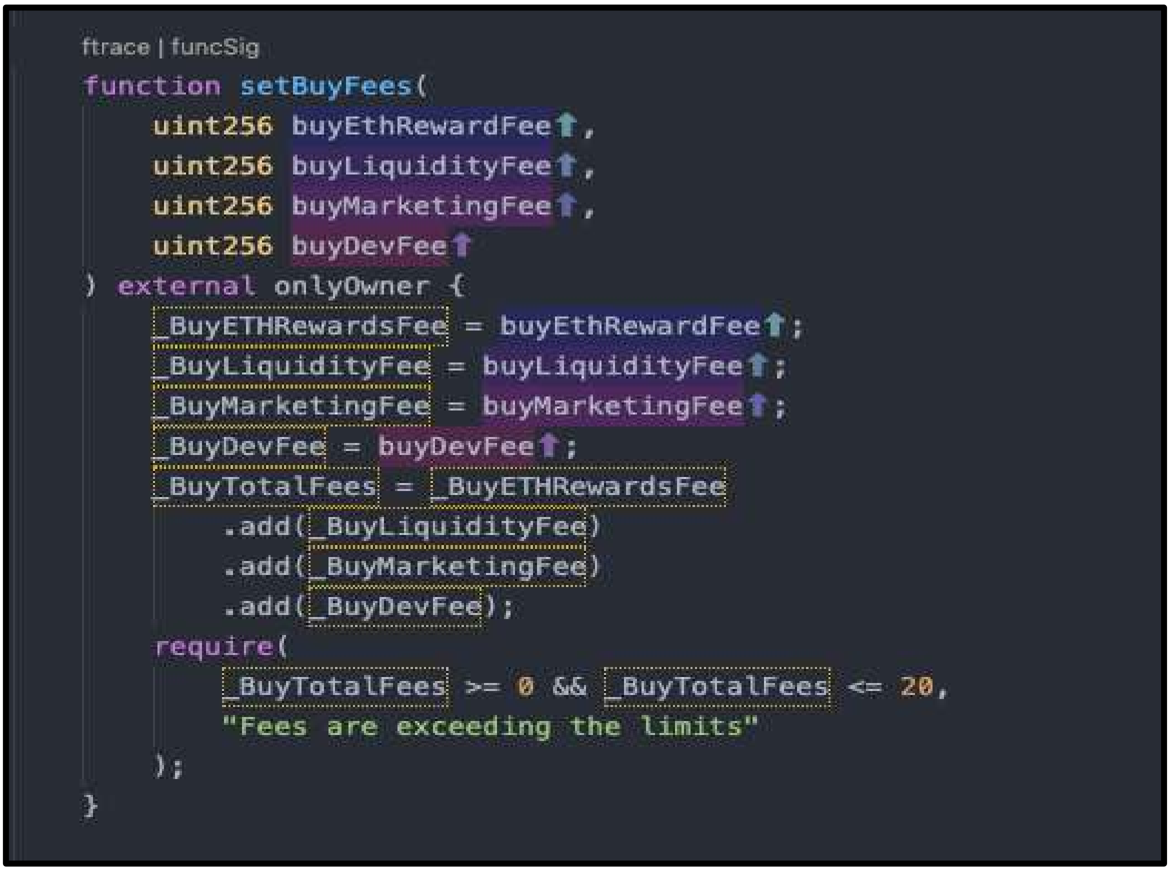
```
ftrace | funcSig
function setMarketingWallet(address payable wallet) external onlyOwner {
    _marketingWalletAddress = wallet;
}

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

- The owner can add/remove wallets from blacklist.



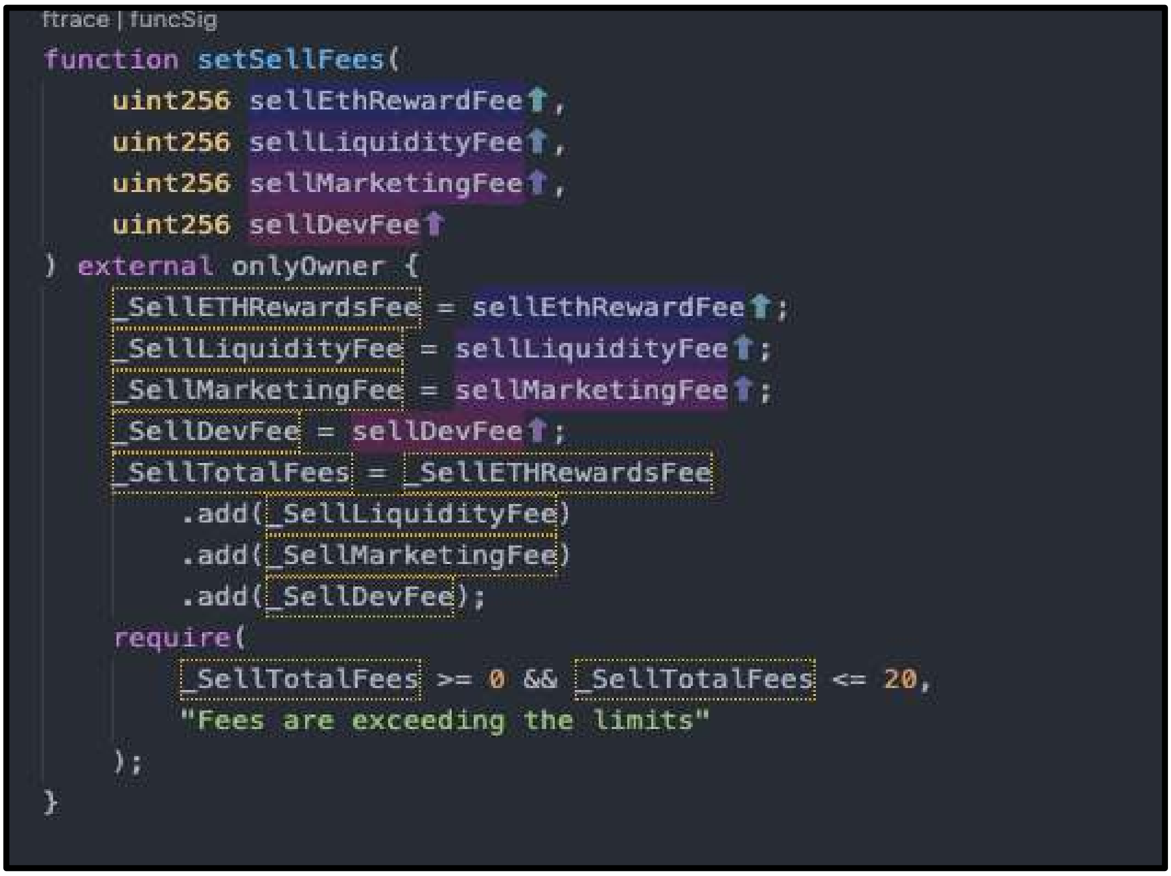
```
ftrace | funcSig
function addToBlackList(address _add) external onlyOwner {
    _blacklisted[_add] = true;
}

ftrace | funcSig
function removeFromBlackList(address _add) external onlyOwner {
    _blacklisted[_add] = false;
}
```

- ❖ The owner can change buy fees maximum up to 20%.



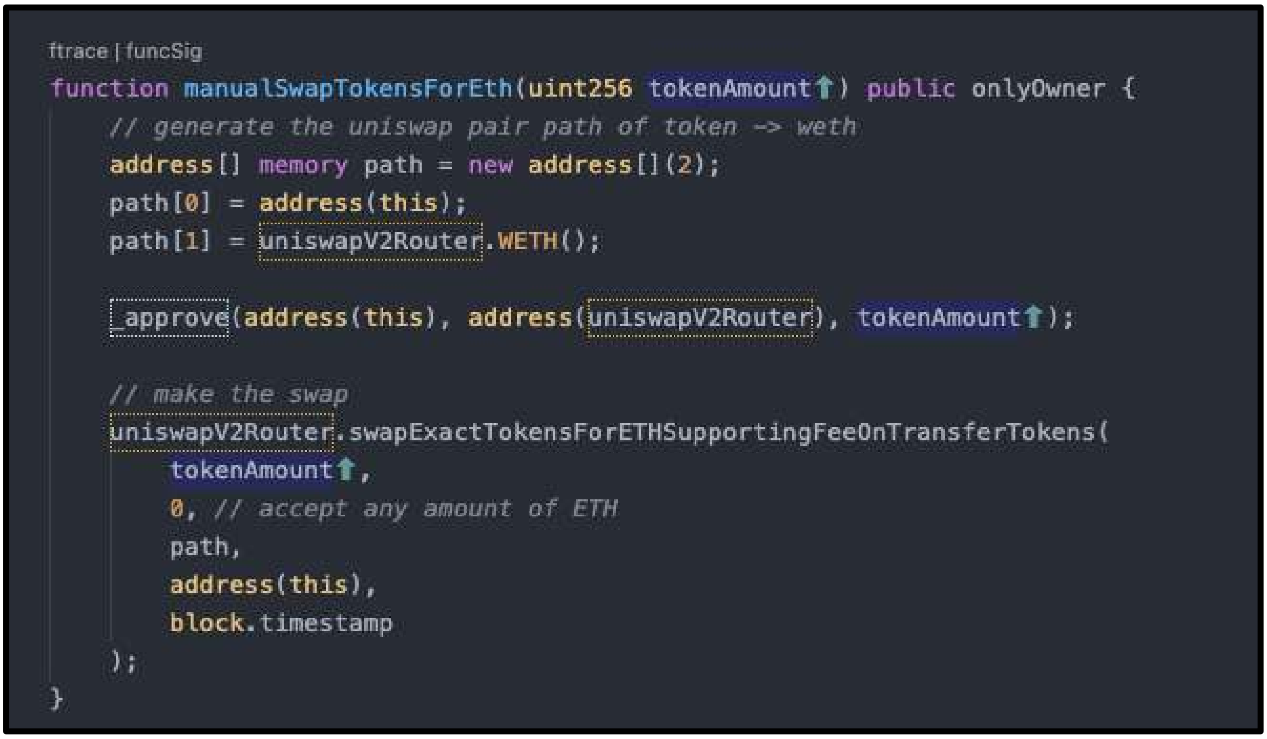
```
function setBuyFees(
    uint256 buyEthRewardFee,
    uint256 buyLiquidityFee,
    uint256 buyMarketingFee,
    uint256 buyDevFee
) external onlyOwner {
    _BuyETHRewardsFee = buyEthRewardFee;
    _BuyLiquidityFee = buyLiquidityFee;
    _BuyMarketingFee = buyMarketingFee;
    _BuyDevFee = buyDevFee;
    _BuyTotalFees = _BuyETHRewardsFee
        .add(_BuyLiquidityFee)
        .add(_BuyMarketingFee)
        .add(_BuyDevFee);
    require(
        _BuyTotalFees >= 0 && _BuyTotalFees <= 20,
        "Fees are exceeding the limits"
    );
}
```

- ❖ The owner can change sell fees maximum up to 20%.



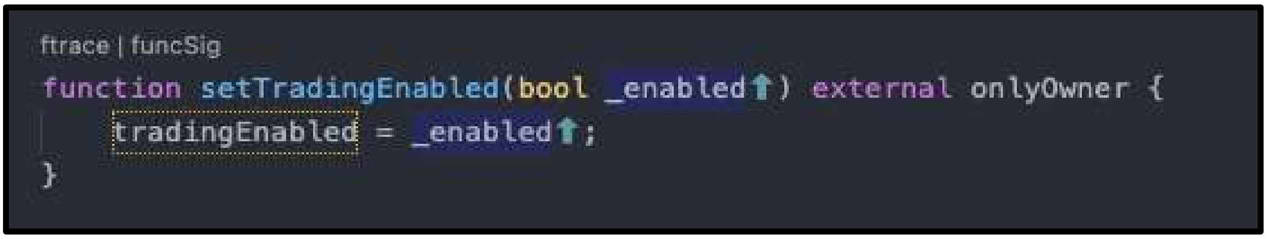
```
function setSellFees(
    uint256 sellEthRewardFee,
    uint256 sellLiquidityFee,
    uint256 sellMarketingFee,
    uint256 sellDevFee
) external onlyOwner {
    _SellETHRewardsFee = sellEthRewardFee;
    _SellLiquidityFee = sellLiquidityFee;
    _SellMarketingFee = sellMarketingFee;
    _SellDevFee = sellDevFee;
    _SellTotalFees = _SellETHRewardsFee
        .add(_SellLiquidityFee)
        .add(_SellMarketingFee)
        .add(_SellDevFee);
    require(
        _SellTotalFees >= 0 && _SellTotalFees <= 20,
        "Fees are exceeding the limits"
    );
}
```

- The owner can manually swap tokens.



```
ftrace | funcSig
function manualSwapTokensForEth(uint256 tokenAmount) public onlyOwner {
    // generate the uniswap pair path of token -> weth
    address[] memory path = new address[](2);
    path[0] = address(this);
    path[1] = uniswapV2Router.WETH();

    _approve(address(this), address(uniswapV2Router), tokenAmount);

    // make the swap
    uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
        tokenAmount,
        0, // accept any amount of ETH
        path,
        address(this),
        block.timestamp
    );
}
```

- The owner can enable/disable trading.



```
ftrace | funcSig
function setTradingEnabled(bool _enabled) external onlyOwner {
    tradingEnabled = _enabled;
}
```

Audit conclusion

While conducting the audit of the ETH Fan Eco System Token smart contract, it was observed that there is nothing alarming with the code, and it only contains a low severity issues and informational concerns.