RugFreeCoins Audit

Multiverse Capital Token

Smart Contract Security Audit

December 1, 2021

Contents

Audit details

Audited project

Multiverse Capital Token

Contract Address

0x80d04E44955AA9c3F24041B2A824A20A88E735a8

Client contact

Multiverse Capital Team

Blockchain

Binance smart chain

Project website

https://www.mvc.finance/

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 Multiverse Capital Token to perform an audit of the smart contract.

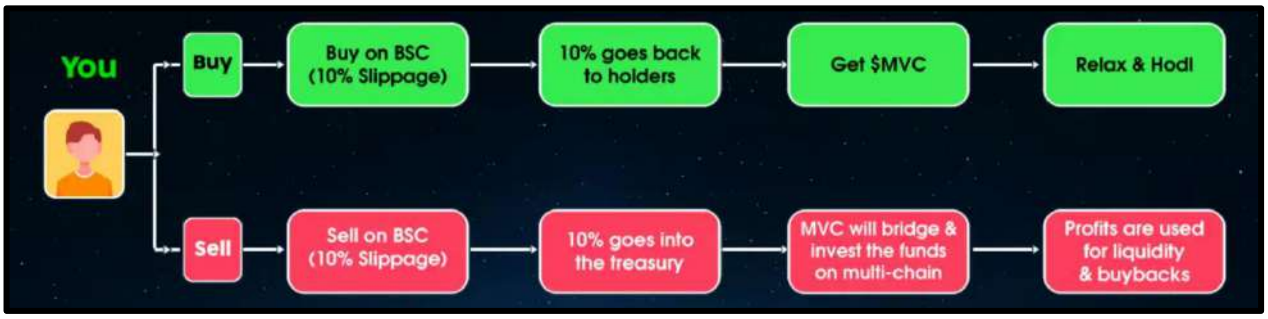
https://bscscan.com/address/0x80d04E44955AA9c3F24041B2A824A20A88E735a8

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

MultiVerse Capital ($MVC) is is a token built on the Binance Smart Chain with a unique Farming Deflationary use case. Each transaction, purchase, and sale incur a 10% fee. Holders can earn the juiciest yields across the metaverse simply by holding $MVC tokens, backed by frequent, positive-sum, and compounded buybacks.


You
Buy
Buy on BSC (10% Slippage)
10% goes back to holders
Get $MVC
Relax & Hodl
Sell
Sell on BSC (10% Slippage)
10% goes into the treasury
MVC will bridge & invest the funds on multi-chain
Profits are used for liquidity & buybacks

Features

- ❖ **The Multiverse Capital Token rewards** will be distributed among every holder proportional to how many tokens each individual holds in values of **10% when buying.**

- ❖ 5% of the sale, in $BNB, is deposited to the farming/buyback wallet. This fund will be bridged to multi-chains (such as Avax, Polygon, Fantom, etc.) to farm on the most profitable farm.

- ❖ 5% of the sale, in $BNB, is sent to the marketing wallet to pay for big marketing (~3%), dev, mods, servers, and other costs (~2%).

- ❖ In case the marketing wallet grows too big even after big marketing, 50% of it will also be used as a farming/buyback fund.

Tokenomics

10% fee when buying

- ❖ 10% of trade goes to holders pockets in token.

10% fee when selling

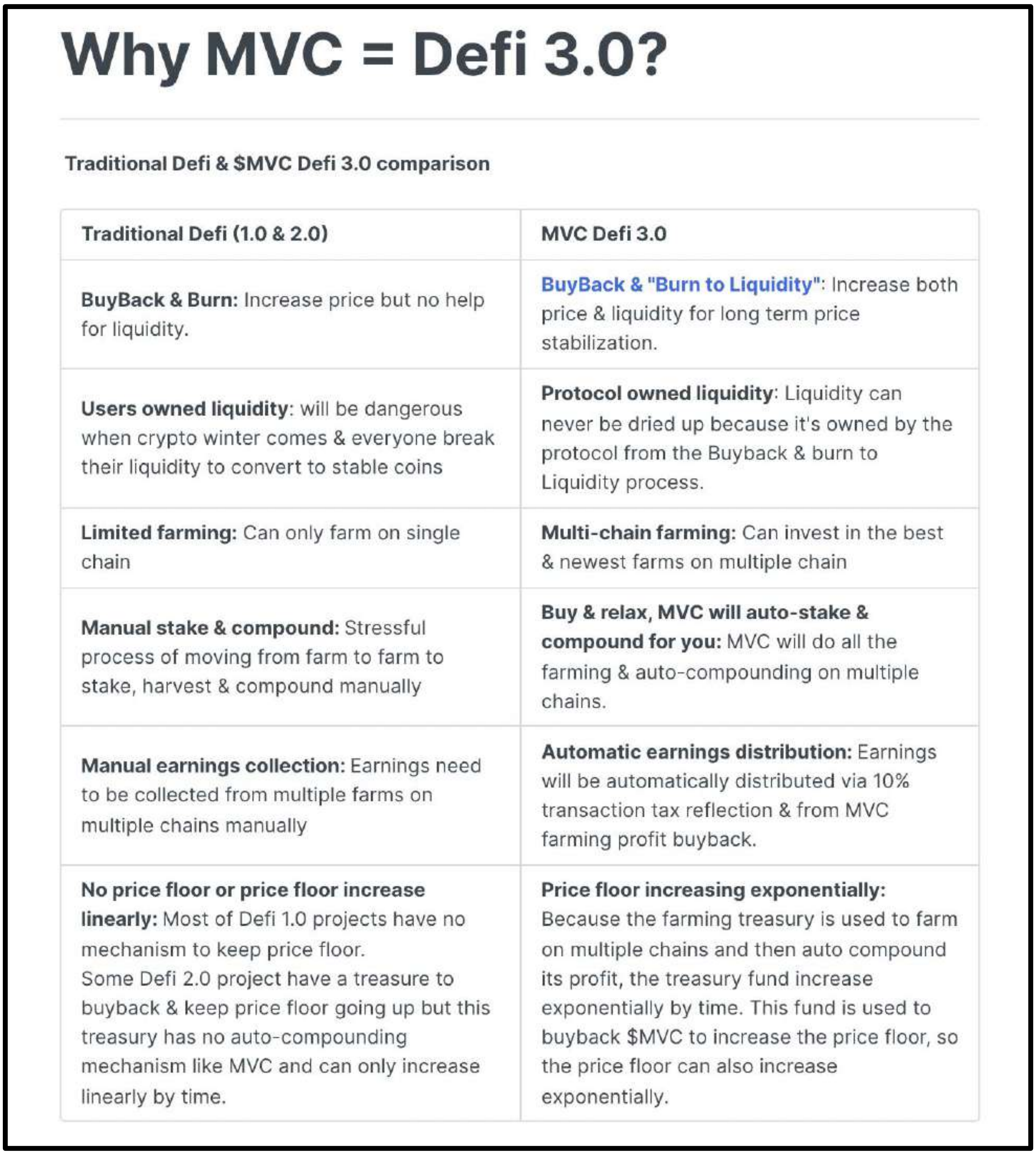
- ❖ 5% of trade goes to farming/buyback wallet in BNB.
- ❖ 3% of trade goes to the marketing wallet in BNB.
- ❖ 2% of trade goes to the development wallet in BNB.

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 in Multiverse Capital Tokens by holding tokens.
- Anyone who's ready to earn a passive income in tokens by letting the token take care of farming & auto-compounding on multiple chains.
- Anyone who's interested in doing daily tasks in the game and gets rewards in $EpicHero tokens.
- Anyone who's interested in trading tokens.
- Anyone who's interested in taking part with the future plans of the Multiverse Capital token.
- Anyone who's interested in making financial transactions with any other party using Multiverse Capital as the currency.

Core concept



Why MVC = Defi 3.0?

Traditional Defi & $MVC Defi 3.0 comparison

Traditional Defi (1.0 & 2.0)	MVC Defi 3.0
BuyBack & Burn: Increase price but no help for liquidity.	**BuyBack & "Burn to Liquidity"**: Increase both price & liquidity for long term price stabilization.
Users owned liquidity: will be dangerous when crypto winter comes & everyone break their liquidity to convert to stable coins	**Protocol owned liquidity**: Liquidity can never be dried up because it's owned by the protocol from the Buyback & burn to Liquidity process.
Limited farming: Can only farm on single chain	**Multi-chain farming:** Can invest in the best & newest farms on multiple chain
Manual stake & compound: Stressful process of moving from farm to farm to stake, harvest & compound manually	**Buy & relax, MVC will auto-stake & compound for you:** MVC will do all the farming & auto-compounding on multiple chains.
Manual earnings collection: Earnings need to be collected from multiple farms on multiple chains manually	**Automatic earnings distribution:** Earnings will be automatically distributed via 10% transaction tax reflection & from MVC farming profit buyback.
No price floor or price floor increase linearly: Most of Defi 1.0 projects have no mechanism to keep price floor. Some Defi 2.0 project have a treasure to buyback & keep price floor going up but this treasury has no auto-compounding mechanism like MVC and can only increase linearly by time.	**Price floor increasing exponentially:** Because the farming treasury is used to farm on multiple chains and then auto compound its profit, the treasury fund increase exponentially by time. This fund is used to buyback $MVC to increase the price floor, so the price floor can also increase exponentially.

Potential to grow with score points

1.	Project efficiency	9/10
2.	Project uniqueness	10/10
3	Information quality	10/10
4	Service quality	10/10
5	System quality	9/10
6	Impact on the community	9/10
7	Impact on the business	10/10
8	Preparing for the future	9/10
Total Points		**9.5/10**

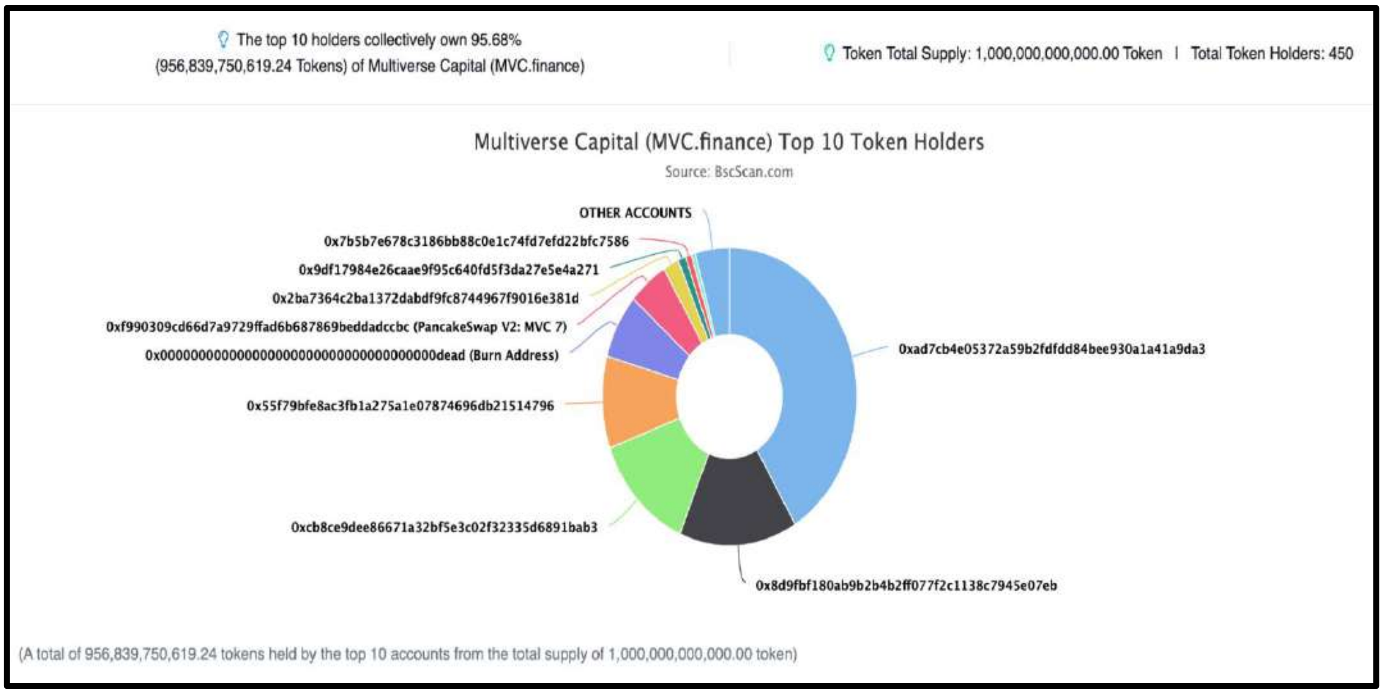
Contract details

Token contract details for 01st December 2021

Contract name	Multiverse Capital (MVC.finance)
Contract address	0x80d04E44955AA9c3F24041B2A824A20A88E735a8
Token supply	1,000,000,000,000
Token ticker	MVC
Decimals	18
Token holders	439
Transaction count	614
Top 100% holders dominance	99.5%
Dev wallet	0xd214d098bbfd072837a68c61ac98ed8439cf31e6
Farm & Buyback wallet	0xcae017595027a8e33ff7f905efacbb53d557b598
Marketing wallet	0xf13a2113446eaca394513769d268a181c6ed55c1
Contract deployer address	0xAC8AEe86CbF0f7169F82f5e1A2941d317816a380
Contract's current owner address	0xac8aee86cbf0f7169f82f5e1a2941d317816a380

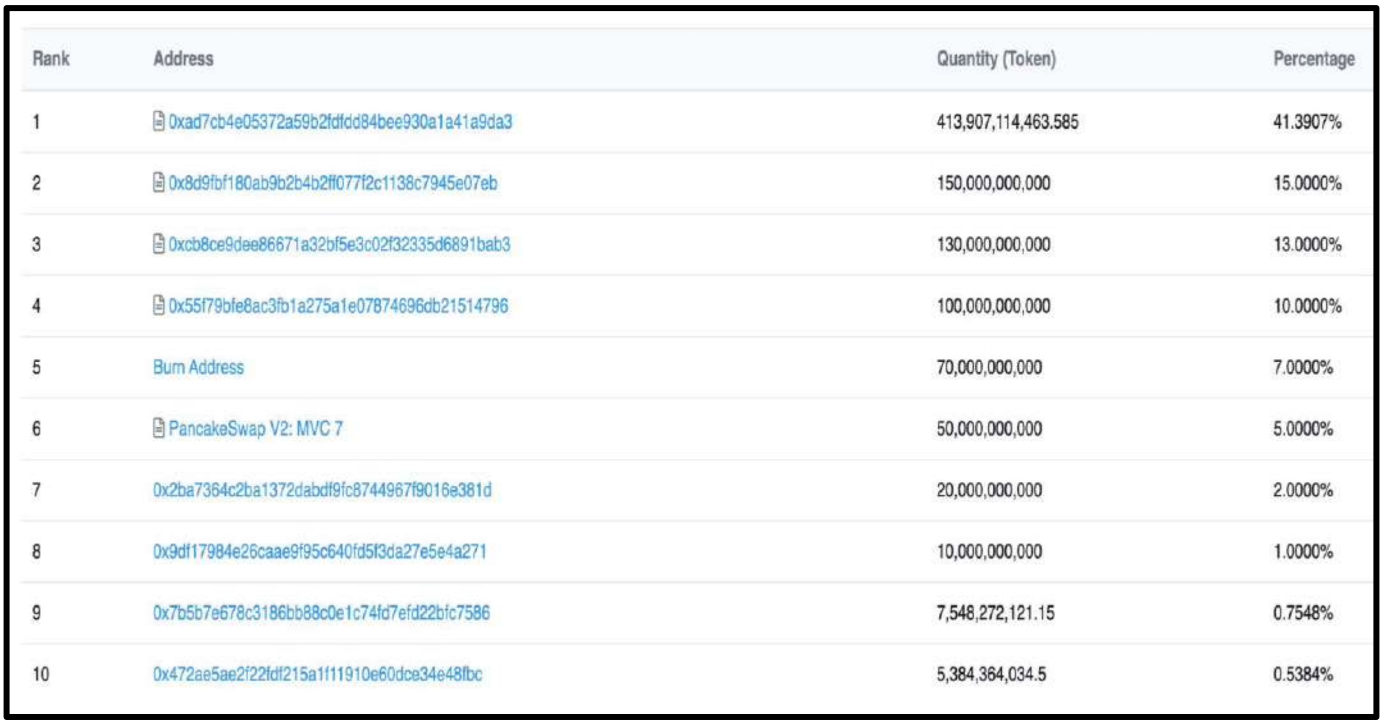
Top token holders

Top 10 Token Holders


The top 10 holders collectively own 95.68% (956,839,750,619.24 Tokens) of Multiverse Capital (MVC.finance)

Token Total Supply: 1,000,000,000,000.00 Token | Total Token Holders: 450

(A total of 956,839,750,619.24 tokens held by the top 10 accounts from the total supply of 1,000,000,000,000.00 token)



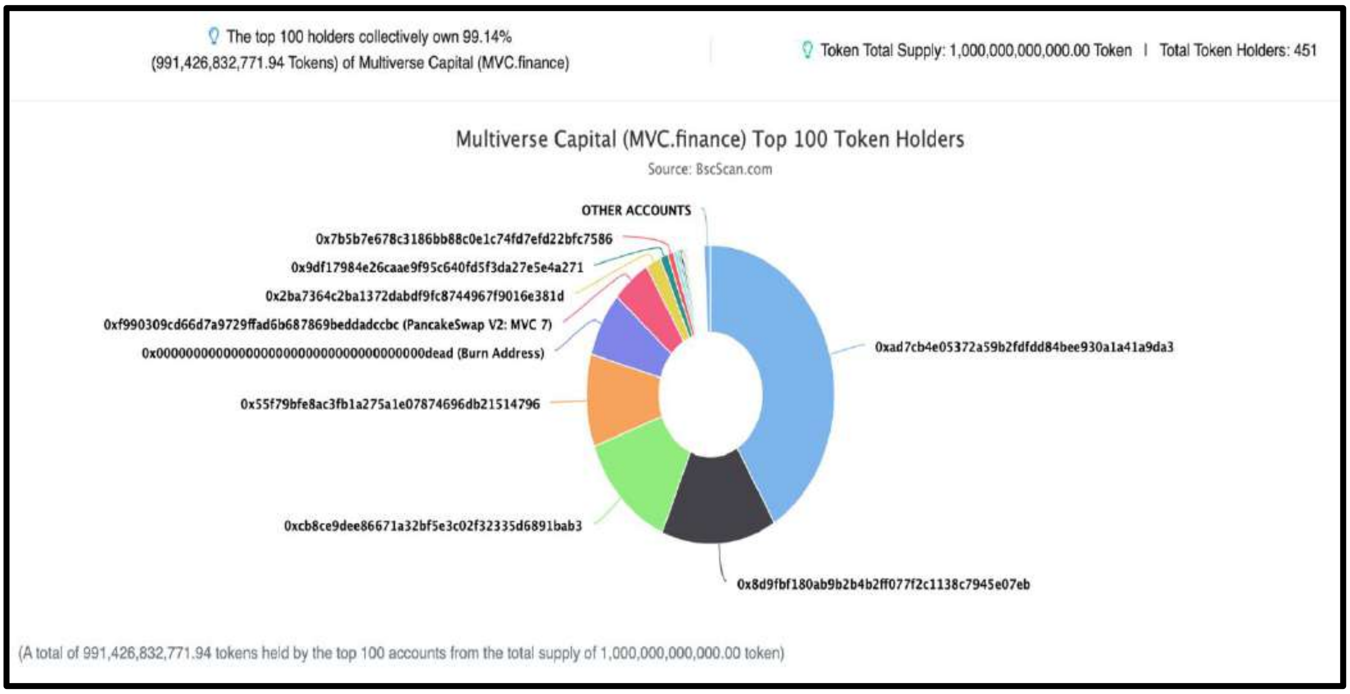
Rank	Address	Quantity (Token)	Percentage
1	0xad7cb4e05372a59b2fdfdd84bee930a1a41a9da3	413,907,114,463.585	41.3907%
2	0x8d9fbf180ab9b2b4b2ff077f2c1138c7945e07eb	150,000,000,000	15.0000%
3	0xcb8ce9dee86671a32bf5e3c02f32335d6891bab3	130,000,000,000	13.0000%
4	0x55f79bfe8ac3fb1a275a1e07874696db21514796	100,000,000,000	10.0000%
5	Burn Address	70,000,000,000	7.0000%
6	PancakeSwap V2: MVC 7	50,000,000,000	5.0000%
7	0x2ba7364c2ba1372dabdf9fc8744967f9016e381d	20,000,000,000	2.0000%
8	0x9df17984e26caae9f95c640fd5f3da27e5e4a271	10,000,000,000	1.0000%
9	0x7b5b7e678c3186bb88c0e1c74fd7efd22bfc7586	7,548,272,121.15	0.7548%
10	0x472ae5ae2f22fdf215a1f11910e60dce34e48fbc	5,384,364,034.5	0.5384%

Token distribution

Token distribution is as follows:

❖ **Initial Burn**	:7% - this amount will be sent to the dead wallet.
❖ **Deflationary mechanism**	: The dead wallet is treated as a normal wallet that can receive reflections. So it will accumulate MVC from every transaction and will be ever-increasing to 10% and more. It means at least 0.7% of every transaction will be sent to the dead wallet. This served as an effective burning mechanism and make MVC deflationary.
❖ **Public sale**	:47%
❖ **Liquidity**	:5%
❖ **Team**	:15%. Distribution: 2 months full locked, then 0.15% weekly (~2 years fully vested)
❖ **Marketing fund**	:10% Marketing, Audit, Exchanges, etc.
❖ **Reserve fund**	:13%, for future burns or future strategic investors
❖ **Airdrop**	:1%
❖ **ITO fee**	:2%

Top 100 Token Holders



The top 100 holders collectively own 99.14%
(991,426,832,771.94 Tokens) of Multiverse Capital (MVC.finance)

Token Total Supply: 1,000,000,000,000.00 Token | Total Token Holders: 451

Multiverse Capital (MVC.finance) Top 100 Token Holders

Source: BscScan.com

OTHER ACCOUNTS
0x7b5b7e678c3186bb88c0e1c74fd7efd22bfc7586
0x9df17984e26caae9f95c640fd5f3da27e5e4a271
0x2ba7364c2ba1372dabdf9fc8744967f9016e381d
0xf990309cd66d7a9729ffad6b687869beddadccbc (PancakeSwap V2: MVC 7)
0x000000000000000000000000000000000000dead (Burn Address)
0x55f79bfe8ac3fb1a275a1e07874696db21514796
0xcb8ce9dee86671a32bf5e3c02f32335d6891bab3
0xad7cb4e05372a59b2fdfdd84bee930a1a41a9da3
0x8d9fbf180ab9b2b4b2ff077f2c1138c7945e07eb

(A total of 991,426,832,771.94 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000.00 token)

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	low issue
		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
Address	**Library**			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	_functionCallWithValue	Private 🔐	🛑	
Context	**Implementation**			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Ownable	**Implementation**	**Context**		

L		Internal 🔒	🛑	
L	owner	Public ❗		NO❗
L	renounceOwnership	Public ❗	🛑	onlyOwner
L	transferOwnership	Public ❗	🛑	onlyOwner
IERC20	**Interface**			
L	totalSupply	External ❗		NO❗
L	balanceOf	External ❗		NO❗
L	transfer	External ❗	🛑	NO❗
L	allowance	External ❗		NO❗
L	approve	External ❗	🛑	NO❗
L	transferFrom	External ❗	🛑	NO❗
SafeMath	**Library**			
L	add	Internal 🔒		
L	sub	Internal 🔒		
L	sub	Internal 🔒		
L	mul	Internal 🔒		
L	div	Internal 🔒		

└	div	Internal 🔒		
└	mod	Internal 🔒		
└	mod	Internal 🔒		
IUniswapV2Factory	**Interface**			
└	createPair	External ❗	🛑	NO❗
IUniswapV2Pair	**Interface**			
└	sync	External ❗	🛑	NO❗
IUniswapV2Router01	**Interface**			
└	factory	External ❗		NO❗
└	WETH	External ❗		NO❗
└	addLiquidity	External ❗	🛑	NO❗
└	addLiquidityETH	External ❗	💵	NO❗
IUniswapV2Router02	**Interface**	**IUniswapV2Router01**		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗

└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
MultiVerseCapital	**Implementation**	**Context, IERC20, Ownable**		
└		Public ❗	🛑	NO❗
└		External ❗	💵	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❗

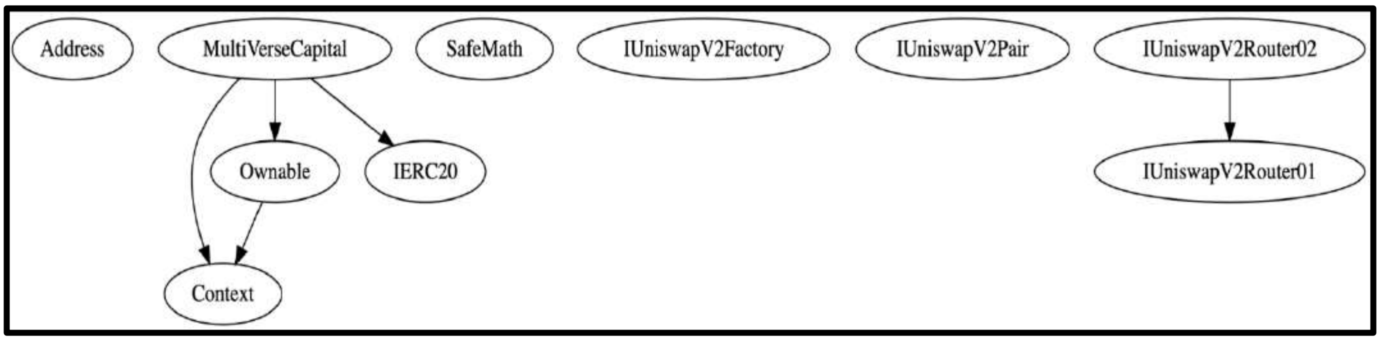
L	isExcludedFromReward	Public ❗️		NO❗️
L	reflectionFromToken	Public ❗️		NO❗️
L	tokenFromReflection	Public ❗️		NO❗️
L	excludeFromReward	Public ❗️	🛑	onlyOwner
L	includeInReward	External ❗️	🛑	onlyOwner
L	_approve	Private 🔐	🛑	
L	_transfer	Private 🔐	🛑	
L	calculateFee	Internal 🔒	🛑	
L	collectFee	Private 🔐	🛑	
L	swap	Private 🔐	🛑	lockTheSwap
L	_getReflectionRate	Private 🔐		
L	setPairRouterRewardToken	External ❗️	🛑	onlyOwner
L	setExcludeFromFee	External ❗️	🛑	onlyOwner
L	setSwapEnabled	External ❗️	🛑	onlyOwner
L	setFeeActive	External ❗️	🛑	onlyOwner
L	setTransferFee	External ❗️	🛑	onlyOwner
L	setMarketingFee	External ❗️	🛑	onlyOwner
L	setFarmAndBuybackFee	External ❗️	🛑	onlyOwner

L	setFarmAndBuybackWallet	External ❗	🛑	onlyOwner
L	setMarketingWallet	External ❗	🛑	onlyOwner
L	setDevWallet	External ❗	🛑	onlyOwner
L	setMaxTxAmount	External ❗	🛑	onlyOwner
L	setMinTokensBeforeSwap	External ❗	🛑	onlyOwner
L	getStuckBNB	External ❗	🛑	onlyOwner
L	getStuckToken	External ❗	🛑	onlyOwner

Legend

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

Inheritance Hierarchy



Security issue checking status

❖ **High severity issues**

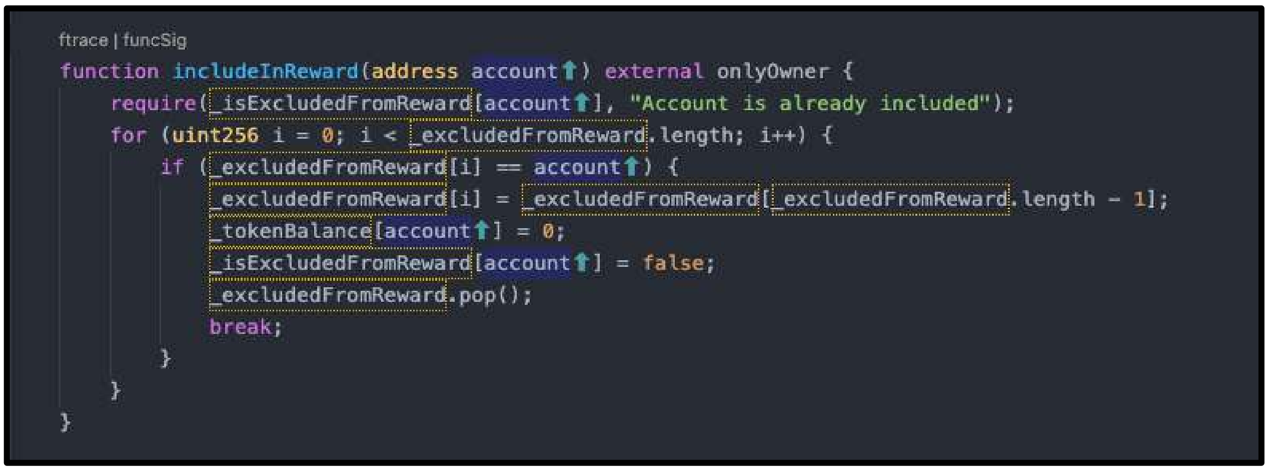
- **No high severity issues found.**

❖ **Medium severity issues**

- **No medium severity issues found.**

❖ **Low severity issues**

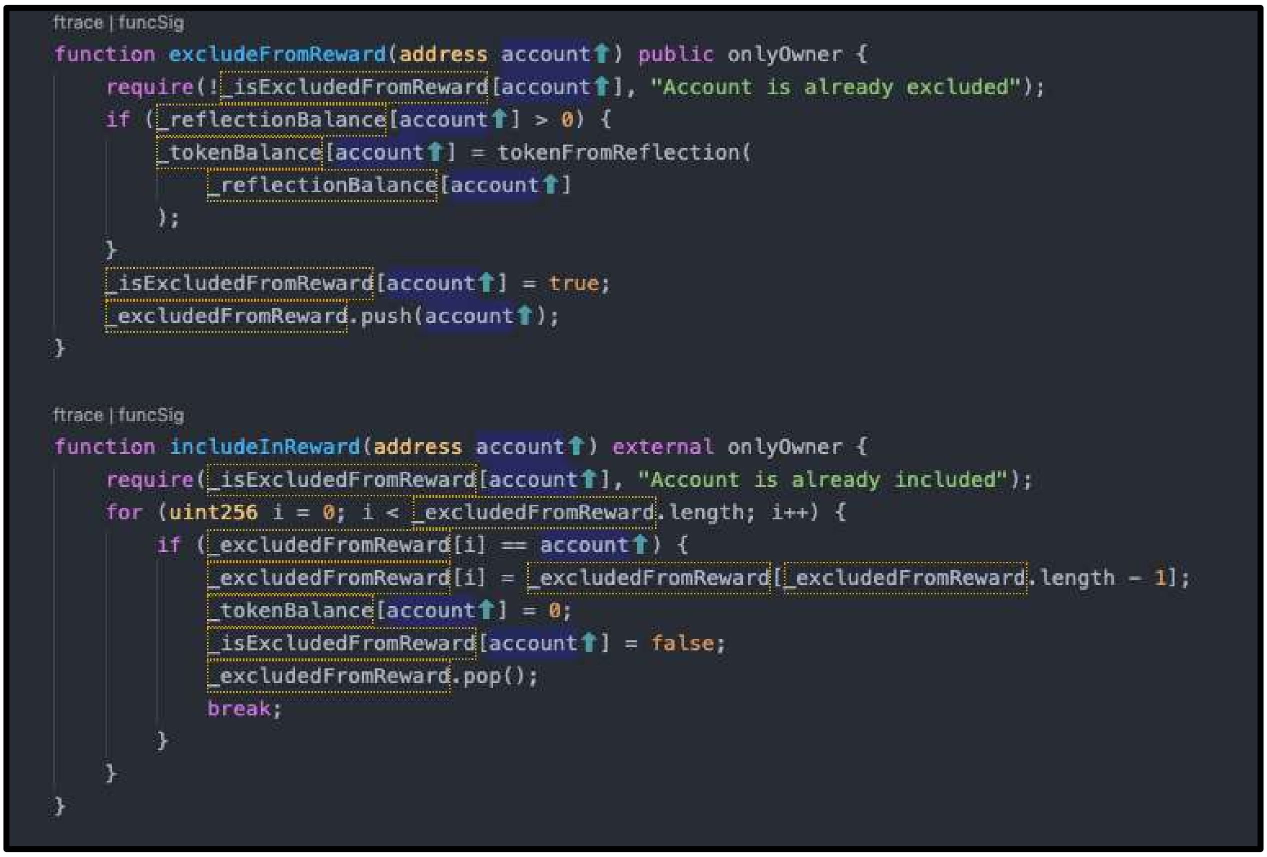
- **In the includeInReward function, if they use a long wallet list there can be an OUT_OF_GAS issue, better to use a small array list at once.**



```
ftrace | funcSig
function includeInReward(address account) external onlyOwner {
    require(_isExcludedFromReward[account], "Account is already included");
    for (uint256 i = 0; i < _excludedFromReward.length; i++) {
        if (_excludedFromReward[i] == account) {
            _excludedFromReward[i] = _excludedFromReward[_excludedFromReward.length - 1];
            _tokenBalance[account] = 0;
            _isExcludedFromReward[account] = false;
            _excludedFromReward.pop();
            break;
        }
    }
}
```

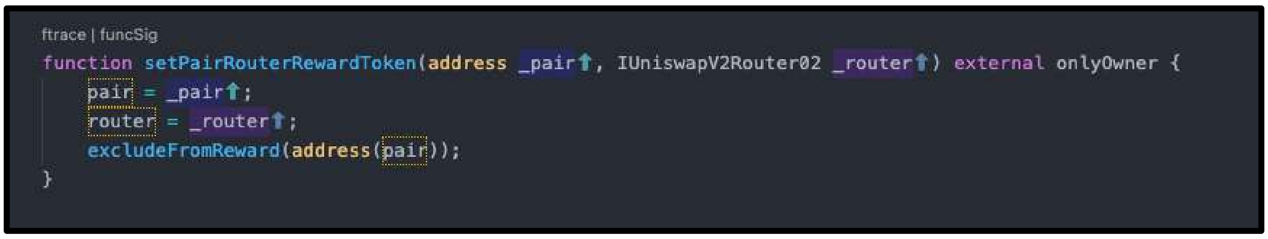
Owner privileges

❖ The owner can exclude and include wallet from reward.



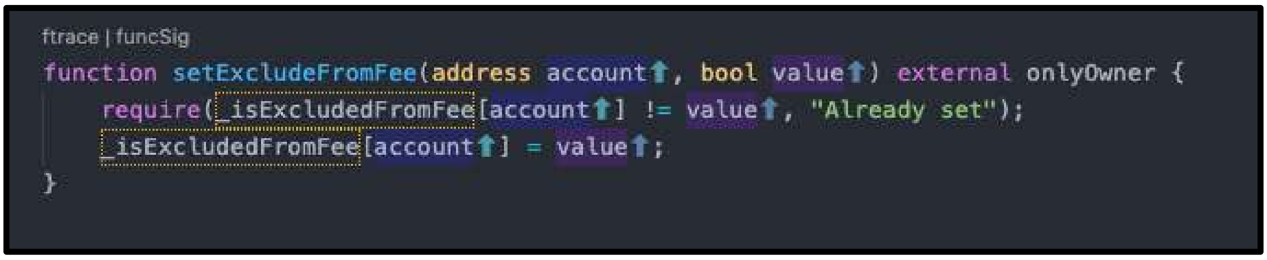
```
ftrace | funcSig
function excludeFromReward(address account) public onlyOwner {
    require(!_isExcludedFromReward[account], "Account is already excluded");
    if (_reflectionBalance[account] > 0) {
        _tokenBalance[account] = tokenFromReflection(
            _reflectionBalance[account]
        );
    }
    _isExcludedFromReward[account] = true;
    _excludedFromReward.push(account);
}

ftrace | funcSig
function includeInReward(address account) external onlyOwner {
    require(_isExcludedFromReward[account], "Account is already included");
    for (uint256 i = 0; i < _excludedFromReward.length; i++) {
        if (_excludedFromReward[i] == account) {
            _excludedFromReward[i] = _excludedFromReward[_excludedFromReward.length - 1];
            _tokenBalance[account] = 0;
            _isExcludedFromReward[account] = false;
            _excludedFromReward.pop();
            break;
        }
    }
}
```

❖ The owner can change the router address.



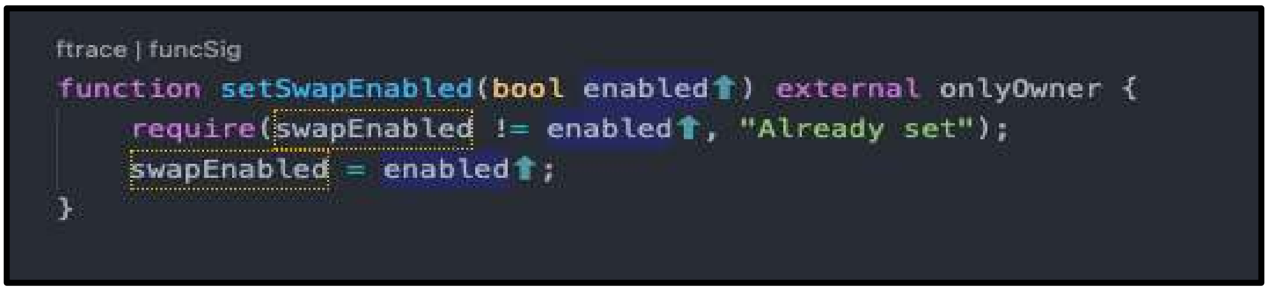
```
ftrace | funcSig
function setPairRouterRewardToken(address _pair, IUniswapV2Router02 _router) external onlyOwner {
    pair = _pair;
    router = _router;
    excludeFromReward(address(pair));
}
```

❖ The owner can exclude wallets from the fee.



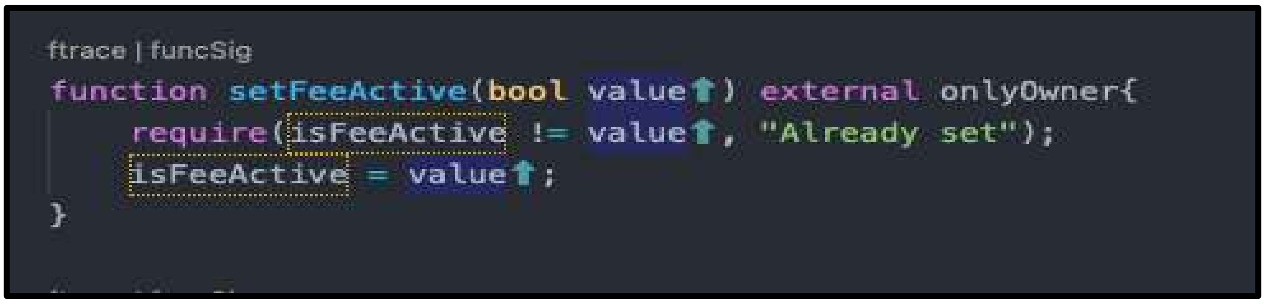
```
ftrace | funcSig
function setExcludeFromFee(address account, bool value) external onlyOwner {
    require(_isExcludedFromFee[account] != value, "Already set");
    _isExcludedFromFee[account] = value;
}
```

❖ The owner can enable/disable swapping.



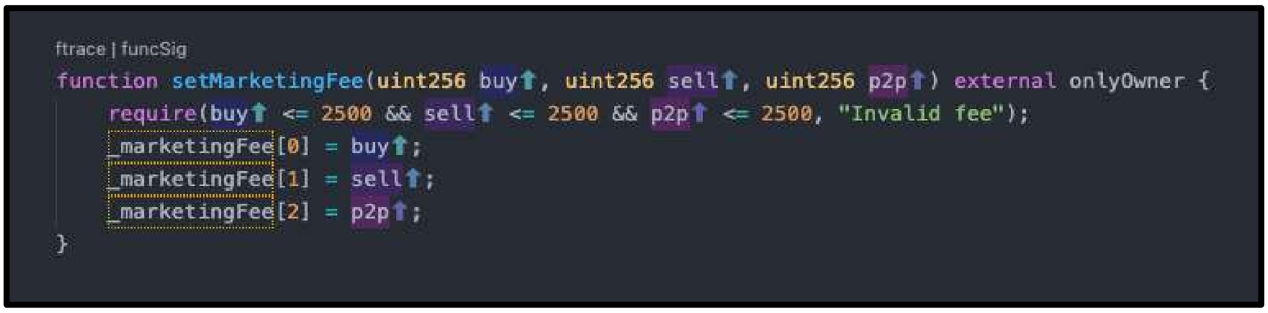
```
ftrace | funcSig
function setSwapEnabled(bool enabled) external onlyOwner {
    require(swapEnabled != enabled, "Already set");
    swapEnabled = enabled;
}
```

❖ The owner can enable/disable fees.



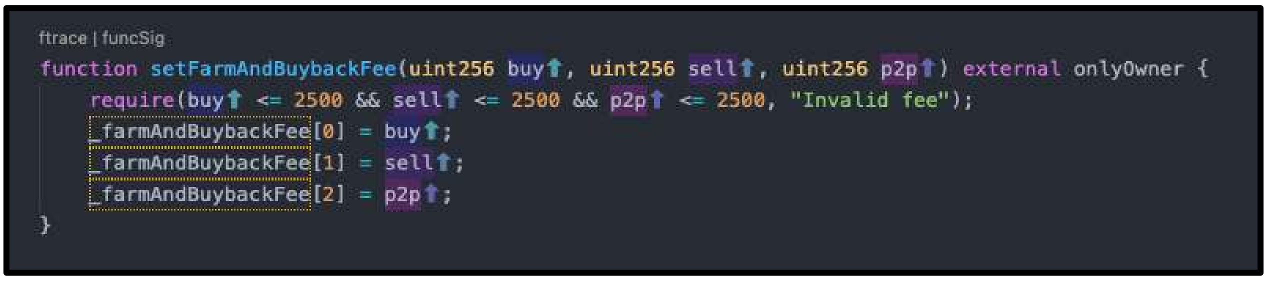
```
ftrace | funcSig
function setFeeActive(bool value) external onlyOwner{
    require(isFeeActive != value, "Already set");
    isFeeActive = value;
}
```

❖ The owner can change marketing fees.



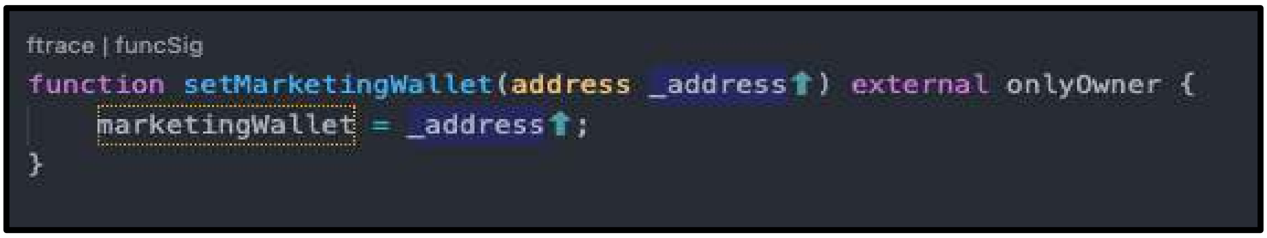
```
ftrace | funcSig
function setMarketingFee(uint256 buy, uint256 sell, uint256 p2p) external onlyOwner {
    require(buy <= 2500 && sell <= 2500 && p2p <= 2500, "Invalid fee");
    _marketingFee[0] = buy;
    _marketingFee[1] = sell;
    _marketingFee[2] = p2p;
}
```

❖ The owner can change buyback fees.



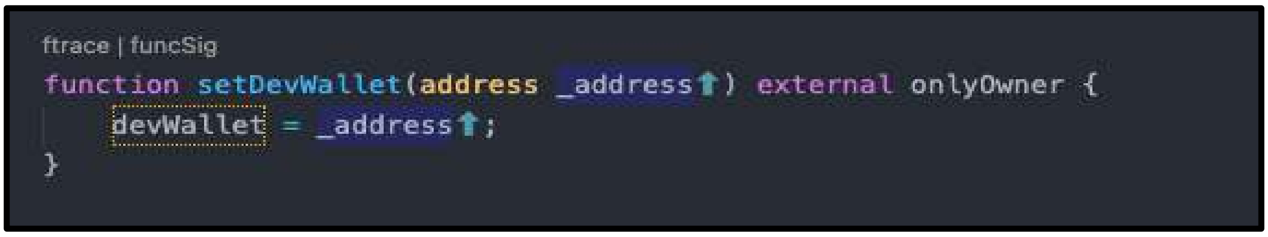
```
ftrace | funcSig
function setFarmAndBuybackFee(uint256 buy, uint256 sell, uint256 p2p) external onlyOwner {
    require(buy <= 2500 && sell <= 2500 && p2p <= 2500, "Invalid fee");
    _farmAndBuybackFee[0] = buy;
    _farmAndBuybackFee[1] = sell;
    _farmAndBuybackFee[2] = p2p;
}
```

❖ The owner can change the marketing wallet.



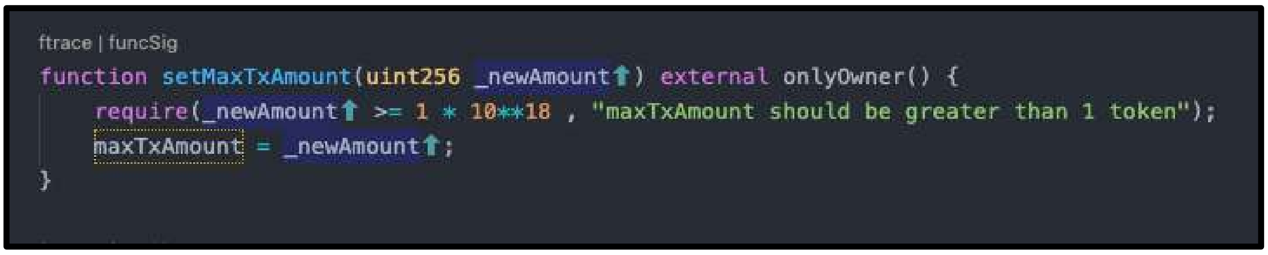
```
ftrace | funcSig
function setMarketingWallet(address _address) external onlyOwner {
    marketingWallet = _address;
}
```

❖ The owner can change the dev wallet address.



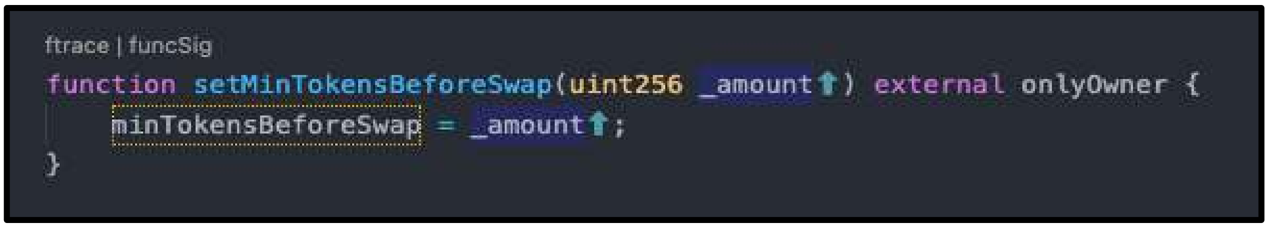
```
ftrace | funcSig
function setDevWallet(address _address) external onlyOwner {
    devWallet = _address;
}
```

❖ The owner can change the max transaction amount.



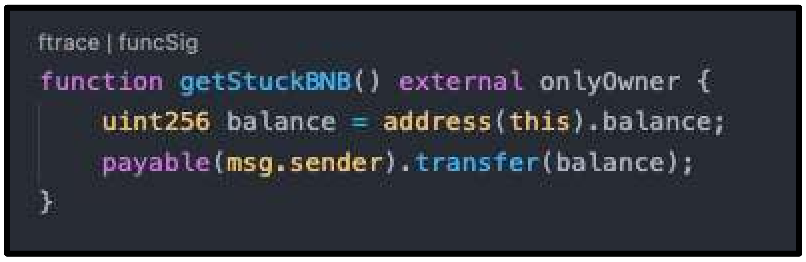
```
ftrace | funcSig
function setMaxTxAmount(uint256 _newAmount) external onlyOwner() {
    require(_newAmount >= 1 * 10**18 , "maxTxAmount should be greater than 1 token");
    maxTxAmount = _newAmount;
}
```

❖ The owner can change the swap point.



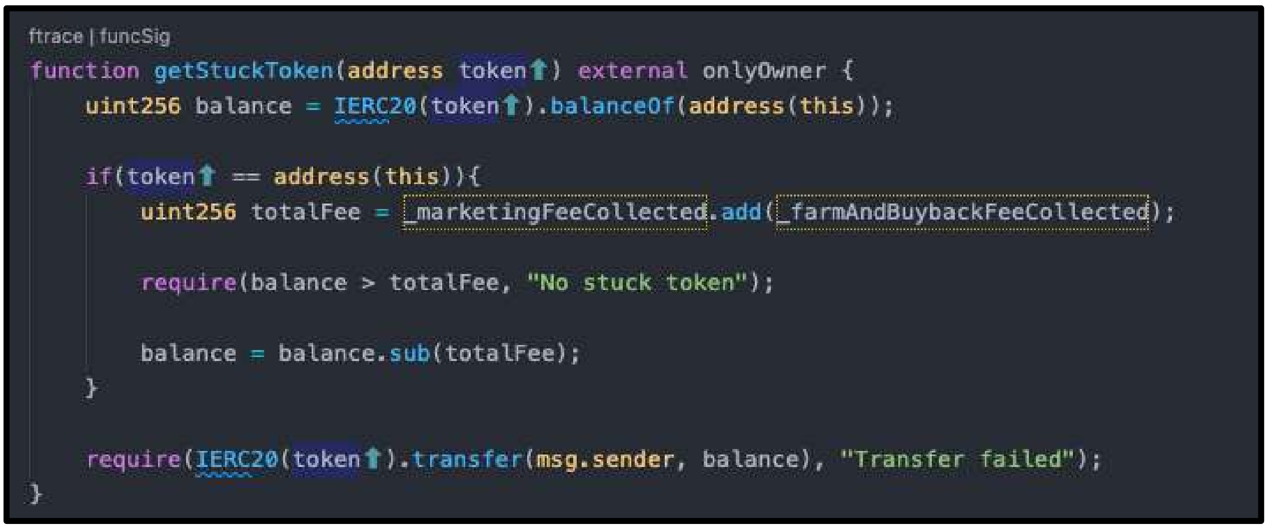
```
ftrace | funcSig
function setMinTokensBeforeSwap(uint256 _amount) external onlyOwner {
    minTokensBeforeSwap = _amount;
}
```

❖ The owner can get the contract BNB balance to the owner's account.



```
ftrace | funcSig
function getStuckBNB() external onlyOwner {
    uint256 balance = address(this).balance;
    payable(msg.sender).transfer(balance);
}
```

❖ The owner can get tokens in the contract to the owner's wallet.



```
ftrace | funcSig
function getStuckToken(address token) external onlyOwner {
    uint256 balance = IERC20(token).balanceOf(address(this));

    if(token == address(this)){
        uint256 totalFee = _marketingFeeCollected.add(_farmAndBuybackFeeCollected);

        require(balance > totalFee, "No stuck token");

        balance = balance.sub(totalFee);
    }

    require(IERC20(token).transfer(msg.sender, balance), "Transfer failed");
}
```

Audit conclusion

While conducting the audit of the Multiverse Capital smart contract, it was observed that there is nothing alarming with the code and it only contains a low severity issue.