RugFreeCoins Audit

Cosmosium BUZZ Token

Smart Contract Security Audit

January 29, 2022

Contents

Audit details

Audited project

Cosmosium BUZZ Token

Contract Address

0xa73C15620bfA79646E9A11d0D638d66588456462

Client contact

Cosmosium BUZZ Team

Blockchain

Binance smart chain

Project website

https://cosmosium.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 Cosmosium BUZZ Token to perform an audit of the smart contract.

https://bscscan.com/token/0xa73C15620bfA79646E9A11d0D638d66588456462

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

Cosmosium BUZZ 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, it will burn most of total supply at long-term. For utilization system Cosmosium Finance offers various products for users, and in order to use these products users have to burn some $BUZZ tokens and it will also cause decrement on total supply. Each transaction, purchase and sale incurs a 5% fee.

Features

- Cosmosium BUZZ has a burn strategy that benefits and rewards those who invest long-term. This feature slowly reduces supply making each Cosmosium BUZZ more and more valuable. The burn fee is 2.5% when buying and selling.

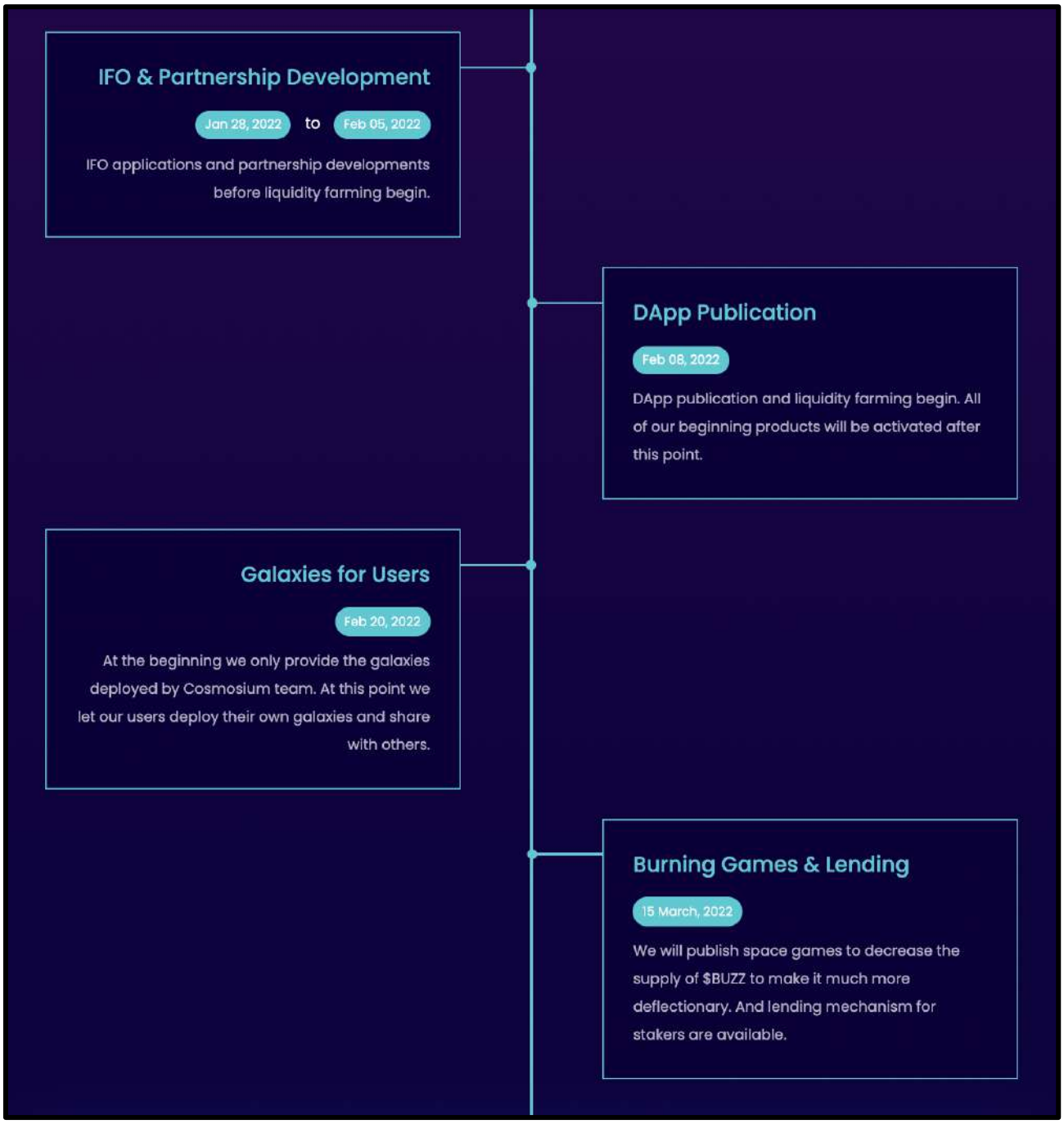
- The additional component included under the sustainability section is a **liquidity fee of 2.5% when buying and selling**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

Tokenomics

5% fee when buying and selling

- 2.5% of trade goes to the burn wallet
- 2.5% of trade goes to the liquidity pool.

Roadmap


IFO & Partnership Development
Jan 28, 2022
to
Feb 05, 2022
IFO applications and partnership developments before liquidity farming begin.
DApp Publication
Feb 08, 2022
DApp publication and liquidity farming begin. All of our beginning products will be activated after this point.
Galaxies for Users
Feb 20, 2022
At the beginning we only provide the galaxies deployed by Cosmosium team. At this point we let our users deploy their own galaxies and share with others.
Burning Games & Lending
15 March, 2022
We will publish space games to decrease the supply of $BUZZ to make it much more deflectionary. And lending mechanism for stakers are available.

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 taking part with BUZZ token's galaxies program
- Anyone who's interested in takinmg part with farming and staking.
- Anyone who's interested in taking part with BUZZ token's satellites system.
- Anyone who's interested in taking part with the future plans of the Cosmosium BUZZ token.
- Anyone who's interested in making financial transactions with any other party using Cosmosium BUZZ as the currency.

Core concept

Cosmosium BUZZ has the burn strategy that benefits and rewards those who invest long-term. This feature slowly reduces supply making Cosmosium BUZZ price more and more valuable.

The liquidity fee of 2.5% when buying and selling, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

The use cases

Cosmic Farms

Cosmic Farms is an auto-compounded earning mechanism by Cosmosium Finance. With Cosmic Farms they offer users to stake their $BUZZ tokens to earn $BUZZ from liquidity mining. User can stake various tokens such as $BUZZ LP tokens and our partners tokens to earn $BUZZ.

Also Cosmic Farms has a different structure than other yield farm mechanisms. They will distribute their partners tokens with same auto-compounded farm at Cosmic Farms. Holders can earn different tokens with Cosmic Farms as well.

Galaxies Program

Galaxies Programs is an unique technique on DeFi world. It's similar to Index system. Users can deploy their own indexes with an amount of token ratios they provide. Every user who wants to join this index should have to mint the index token with providing same ratio on tokens, or they can buy index token directly from contract. There is also zap mechanism which helps user to join index with just one click.

Clusters

Cosmosium Finance offers locked stakings for users. You can select a locking period for your staking. It’s not able to withdraw & harvest until this period ends. Higher locking period benefits with higher APYs.

Short-term: 7 days locking.
Mid-term: 30 days locking.
Long-term: 90 days locking.

Satellites

Satellites are the unique token offering system for our users. It is similar with Call/Put options trading. Users can call BUZZ tokens with discounted price & vesting period. Also they can put BUZZ tokens for higher price with vesting period.

Potential to grow with score points

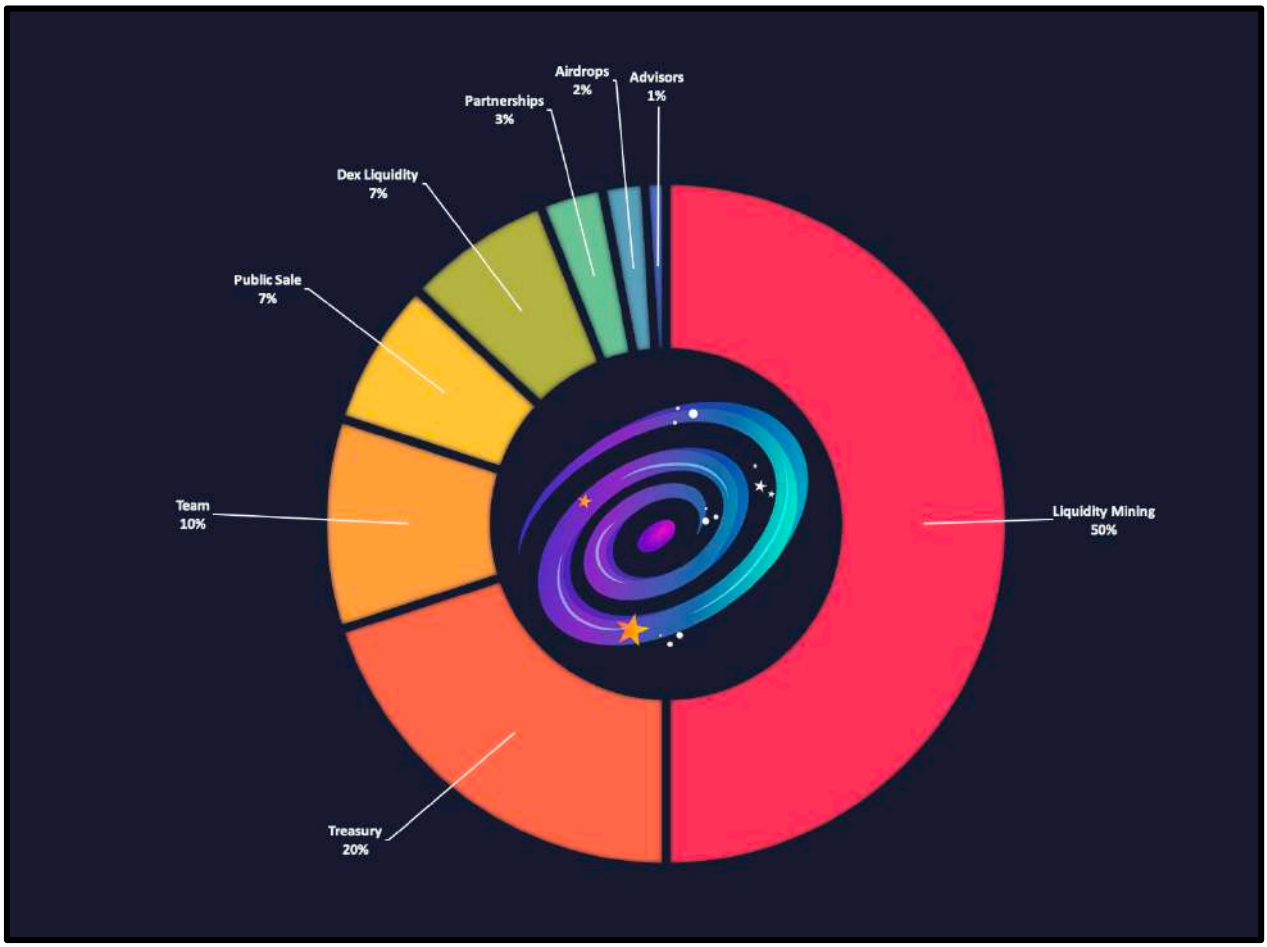
1.	Project efficiency	9/10
2.	Project uniqueness	8/10
3	Information quality	9/10
4	Service quality	9/10
5	System quality	9/10
6	Impact on the community	9/10
7	Impact on the business	9/10
8	Preparing for the future	9/10
Total Points		**8.875/10**

Contract details

Token contract details for 29th January 2022

Contract name	Cosmosium BUZZ
Contract address	0xa73C15620bfA79646E9A11d0D638d66588456462
Token supply	500,000,000
Token ticker	BUZZ
Decimals	18
Token holders	1
Transaction count	3
Operator	0xad8f748b2e87bddcbf98f11a63f9f3cbcdd34b38
Contract deployer address	0xD7051141d24C3c06c7741215d324A07a0fFf9499
Contract's current owner address	0xad8f748b2e87bddcbf98f11a63f9f3cbcdd34b38

Token distribution



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	low issue
		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
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❗
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❗
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❗

L	swapExactTokensForETH	External ❗	🛑	NO❗
L	swapETHForExactTokens	External ❗	💵	NO❗
L	quote	External ❗		NO❗
L	getAmountOut	External ❗		NO❗
L	getAmountIn	External ❗		NO❗
L	getAmountsOut	External ❗		NO❗
L	getAmountsIn	External ❗		NO❗
IUniswapV2Router02	**Interface**	**IUniswapV2Router01**		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
Address	**Library**			

└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionStaticCall	Internal 🔒		
└	functionStaticCall	Internal 🔒		
└	functionDelegateCall	Internal 🔒	🛑	
└	functionDelegateCall	Internal 🔒	🛑	
└	_verifyCallResult	Private 🔐		
SafeMath	**Library**			
└	tryAdd	Internal 🔒		
└	trySub	Internal 🔒		
└	tryMul	Internal 🔒		
└	tryDiv	Internal 🔒		
└	tryMod	Internal 🔒		
└	add	Internal 🔒		

└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	sub	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
IBEP20	**Interface**			
└	totalSupply	External ❗		NO❗
└	decimals	External ❗		NO❗
└	symbol	External ❗		NO❗
└	name	External ❗		NO❗
└	getOwner	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	transfer	External ❗	🛑	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗

Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Ownable	**Implementation**	**Context**		
└		Internal 🔒	🛑	
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
BEP20	**Implementation**	**Context, IBEP20, Ownable**		
└		Public ❗	🛑	NO❗
└	getOwner	External ❗		NO❗
└	name	Public ❗		NO❗
└	decimals	Public ❗		NO❗
└	symbol	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❗
└	mint	Public ❗	🛑	onlyOwner
└	_transfer	Internal 🔒	🛑	
└	_mint	Internal 🔒	🛑	
└	_burn	Internal 🔒	🛑	
└	_approve	Internal 🔒	🛑	
└	_burnFrom	Internal 🔒	🛑	
Buzz	**Implementation**	**BEP20**		
└		Public ❗	🛑	BEP20
└	mint	Public ❗	🛑	onlyOwner
└	_transfer	Internal 🔒	🛑	antiWhale
└	swapAndLiquify	Private 🔐	🛑	lockTheSwap transferTaxFree
└	swapTokensForEth	Private 🔐	🛑	
└	addLiquidity	Private 🔐	🛑	

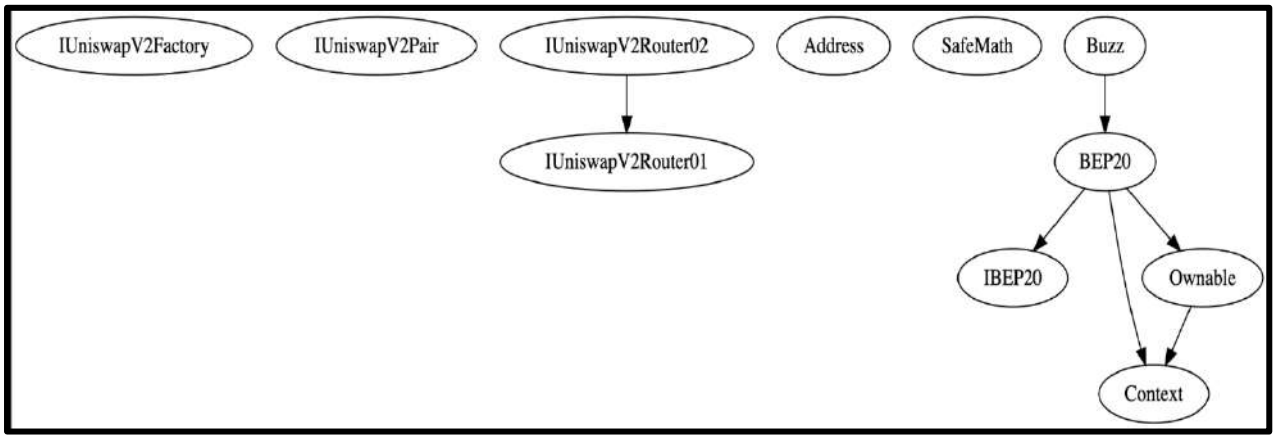
L	maxTransferAmount	Public ❗		NO❗
L	isExcludedFromAntiWhale	Public ❗		NO❗
L		External ❗	💵	NO❗
L	updateTransferTaxRate	Public ❗	🛑	onlyOperator
L	updateBurnRate	Public ❗	🛑	onlyOperator
L	updateMaxTransferAmountRate	Public ❗	🛑	onlyOperator
L	setSpecificFeeForRecipient	Public ❗	🛑	onlyOperator
L	excludeFromTransferFee	Public ❗	🛑	onlyOperator
L	excludedToTransferFee	Public ❗	🛑	onlyOperator
L	updateMinAmountToLiquify	Public ❗	🛑	onlyOperator
L	setExcludedFromAntiWhale	Public ❗	🛑	onlyOperator
L	updateSwapAndLiquifyEnabled	Public ❗	🛑	onlyOperator
L	updateCosmosiumRouter	Public ❗	🛑	onlyOperator
L	operator	Public ❗		NO❗
L	transferOperator	Public ❗	🛑	onlyOperator
L	delegates	External ❗		NO❗
L	delegate	External ❗	🛑	NO❗
L	delegateBySig	External ❗	🛑	NO❗
L	getCurrentVotes	External ❗		NO❗

└	getPriorVotes	External ❗		NO❗
└	_delegate	Internal 🔒	🛑	
└	_moveDelegates	Internal 🔒	🛑	
└	_writeCheckpoint	Internal 🔒	🛑	
└	safe32	Internal 🔒		
└	getChainId	Internal 🔒		

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**

- **The owner cannot mint more tokens as it states, but this is a redundant issue and can remove it since it has no use**.

```
function mint(address _to⬆, uint256 _amount⬆) public onlyOwner {
    require(totalSupply().add(_amount⬆) <= MAX_TOKENS, "max token exceeds")
    /*
        Instead of using _burn method, we are currently sending the burnt to
        So there wont be any additionaly minting can be done after burns.
    */
    _mint(_to⬆, _amount⬆);
    _moveDelegates(address(0), _delegates[_to⬆], _amount⬆);
}
```

Owner privileges

❖ Owner can change fees maximum upto 10%.

```
ftrace | funcSig
function updateTransferTaxRate(uint16 _transferTaxRate⬆)
    public
    onlyOperator
{
    require(
        _transferTaxRate⬆ <= MAXIMUM_TRANSFER_TAX_RATE,
        "BUZZ::updateTransferTaxRate: Transfer tax rate must not exceed the maximum rate."
    );
    emit TransferTaxRateUpdated(
        msg.sender,
        transferTaxRate,
        _transferTaxRate⬆
    );
    transferTaxRate = _transferTaxRate⬆;
}
```

❖ Owner can change burn rate maximum upto 1% from the fee amount.

```
ftrace | funcSig
function updateBurnRate(uint16 _burnRate⬆) public onlyOperator {
    require(
        _burnRate⬆ <= 100,
        "BUZZ::updateBurnRate: Burn rate must not exceed the maximum rate."
    );
    emit BurnRateUpdated(msg.sender, burnRate, _burnRate⬆);
    burnRate = _burnRate⬆;
}
```

❖ Owner can set specific tax for users maximum upto 10%.

```
ftrace | funcSig
function setSpecificFeeForRecipient(address _recipient⬆, uint16 _fee⬆)
    public
    onlyOperator
{
    require(_fee⬆ <= MAXIMUM_TRANSFER_TAX_RATE, "Tax rate too high.");
    _specificToTransferFee[_recipient⬆] = _fee⬆;
    emit SpecificTransferFeeSet(_recipient⬆, _fee⬆);
}
```

❖ Owner can exclude wallets from the fees.

```
ftrace | funcSig
function excludeFromTransferFee(address _from⬆, bool _value⬆)
    public
    onlyOperator
{
    _excludedFromTransferFee[_from⬆] = _value⬆;

    emit ExcludedFromTransferFee(_from⬆, _value⬆);
}
```

❖ Owner can change minimum token amount to add liquidity.

```
ftrace | funcSig
function updateMinAmountToLiquify(uint256 _minAmount⬆) public onlyOperator {
    emit MinAmountToLiquifyUpdated(
        msg.sender,
        minAmountToLiquify,
        _minAmount⬆
    );
    minAmountToLiquify = _minAmount⬆;
}
```

❖ Owner can exclude wallets from max transaction amount.

```
ftrace | funcSig
function setExcludedFromAntiWhale(address _account⬆, bool _excluded⬆)
    public
    onlyOperator
{
    _excludedFromAntiWhale[_account⬆] = _excluded⬆;
}
```

❖ Owner can enable/disable swapping.

```
ftrace | funcSig
function updateSwapAndLiquifyEnabled(bool _enabled⬆) public onlyOperator {
    emit SwapAndLiquifyEnabledUpdated(msg.sender, _enabled⬆);
    swapAndLiquifyEnabled = _enabled⬆;
}

...
```

❖ Owner can change the router address.

```
ftrace | funcSig
function updateCosmosiumRouter(address _router⬆) public onlyOperator {
    CosmosiumRouter = IUniswapV2Router02(_router⬆);
    CosmosiumPair = IUniswapV2Factory(CosmosiumRouter.factory()).createPair(
            address(this),
            CosmosiumRouter.WETH()
        );
    require(
        CosmosiumPair != address(0),
        "BUZZ::updateCosmosiumRouter: Invalid pair address."
    );
    emit CosmosiumRouterUpdated(
        msg.sender,
        address(CosmosiumRouter),
        CosmosiumPair
    );
}
```

❖ Owner can transfer the ownership.

```
ftrace | funcSig
function transferOperator(address newOperator⬆) public onlyOperator {
    require(
        newOperator⬆ != address(0),
        "BUZZ::transferOperator: new operator is the zero address"
    );
    emit OperatorTransferred(_operator, newOperator⬆);
    _operator = newOperator⬆;
}
```

❖ Owner can change max transaction amount up to 0.1% from the total supply.

```
ftrace | funcSig
function updateMaxTransferAmountRate(uint16 _maxTransferAmountRate⬆)
    public
    onlyOperator
{
    require(
        _maxTransferAmountRate⬆ <= 10000,
        "BUZZ::updateMaxTransferAmountRate: Max transfer amount rate must not exceed the maximum rate."
    );
    require(
        _maxTransferAmountRate⬆ >= 10,
        "BUZZ:: max transfer amount too low."
    );
    emit MaxTransferAmountRateUpdated(
        msg.sender,
        maxTransferAmountRate,
        _maxTransferAmountRate⬆
    );
    maxTransferAmountRate = _maxTransferAmountRate⬆;
}
```

Audit conclusion

While conducting the audit of the Cosmosium BUZZ smart contract, it was observed that it only contains a low severity issue.