Cyberscope

Audit Report

PIZZA INU

June 2022

SHA256 9b8b73d5c619c89dfd2430d99d7979ec6fc4c2112c167c4c4974b62eba809692

Audited by © cyberscope

Table of Contents

Contract Review

Contract Name	PINUToken
Compiler Version	v0.8.0+commit.c7dfd78e
Optimization	200 runs
Licence	
Testing Deploy	https://testnet.bscscan.com/token/0xa98A0b1F66773098D8dEBA5C3d294a13Ab68E19e
Symbol	PINU
Decimals	18
Total Supply	100,000,000,000
Domain	pizzainu.io

Source Files

Filename	**SHA256**
contract.sol	9b8b73d5c619c89dfd2430d99d7979ec6fc4c2112c167c4c4974b62eba809692

Audit Updates

Initial Audit	4th June 2022
Corrected	

Contract Analysis

Critical Medium Minor Pass

Severity	Code	Description
●	ST	Contract Owner is not able to stop or pause transactions
●	OCTD	Contract Owner is not able to transfer tokens from specific address
●	OTUT	Owner Transfer User's Tokens
●	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
●	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
●	MT	Contract Owner is not able to mint new tokens
●	BT	Contract Owner is not able to burn tokens from specific wallet
●	BC	Contract Owner is not able to blacklist wallets from selling

ST - Stop Transactions

Criticality	critical
Location	contract.sol#L1178

Description

The contract owner has the authority to stop transactions for all users excluding the owner. The contract owner can also convert the contract into a honeypot and prevent users from selling by increasing the selling taxes (sellRewardFeeRate, sellMarketingFeeRate).

```
else if (lpAddresses[recipient]) {//sell
            feeRate = sellRewardFeeRate + sellMarketingFeeRate;
}
//
if (feeRate > 0 && !excludeFee[sender] && !excludeFee[recipient] && recipient
!= marketingAddress) {
            uint256 fee = amount.mul(feeRate).div(100);
            amount = amount.sub(fee);
```

Recommendation

The contract could embody a check for not allowing setting the sellRewardFeeRate and sellMarketingFeeRate more than a reasonable amount.

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L1264, 1268, 1272, 1276

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the setBuyRewardFeeRate, setSellRewardFeeRate, setBuyMarketingFeeRate and setBuyMarketingFeeRate functions with a high percentage value.

```
function setBuyRewardFeeRate(uint256 _fee) external onlyOwner {
    buyRewardFeeRate = _fee;
}
```

```
function setSellRewardFeeRate(uint256 _fee) external onlyOwner {
    sellRewardFeeRate = _fee;
}
```

```
function setBuyMarketingFeeRate(uint256 _fee) external onlyOwner {
    buyMarketingFeeRate = _fee;
}
```

```
function setSellMarketingFeeRate(uint256 _fee) external onlyOwner {
    sellMarketingFeeRate = _fee;
}
```

Recommendation

The contract could embody a check for the maximum acceptable value.

The team should carefully manage the private keys of the owner’s account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

ULTW - Unlimited Liquidity to Team Wallet

Criticality	minor
Location	contract.sol#L1196, 1200

Description

The contract owner has the authority to transfer funds without limit to the team wallet. These funds have been accumulated from fees collected from the contract. The owner may take advantage of it by calling the withdrawToken and withdrawBNB functions.

```
function withdrawToken(address token) public onlyOwner {
   IERC20(token).transfer(msg.sender, IERC20(token).balanceOf(address(this)));
}
```

```
function withdrawBNB() public onlyOwner {
   payable(msg.sender).transfer(address(this).balance);
}
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may volatile the token's price.

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

BC - Blacklisted Contracts

Criticality	critical
Location	contract.sol#L1168, 1153, 1306

Description

The contract owner has the authority to stop contracts from transactions. The owner may take advantage of it by calling the `blacklistAddress` function.

```
require(!blacklist[sender] && !blacklist[recipient], "the address is blacklisted");
```

The contract embodies a check that allows the owner to modify the `blacklist` list only for the first 10 days after the deployment of the contract.

```
blacklistTime = block.timestamp + 10 days;
```

```
function multiBlacklist(address[] memory addresses, bool isBlacklist) external onlyOwner {

        require(block.timestamp <= blacklistTime, "blacklistTime: invalid");

        for (uint256 i = 0; i < addresses.length; i++) {
            blacklist[addresses[i]] = isBlacklist;
        }
    }
```

Recommendation

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	CO	Code Optimization
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L05	Unused State Variable
	L07	Missing Events Arithmetic
	L09	Dead Code Elimination
	L14	Uninitialized Variables in Local Scope

CO - Code Optimization

Criticality	minor
Location	contract.sol#L1120, 1260

Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations. The antiBotEnabled variable is used only in the code segments below.

```
bool public antiBotEnabled;
```

```
function setEnableAntiBot(bool _enable) external onlyOwner {
   antiBotEnabled = _enable;
}
```

Recommendation

Rewrite some code segments so the runtime will be more performant. Try to remove the code segments.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L310,318,725,733,750,776,795,813,841,860,1040,1055,1196,1200,1235,1241,1280,1284,1288,1292,1325,1329

Description

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

```
setMinHolder
setClaimable
setLpAddress
getRewardedByAddressAndToken
getRewardByAddressAndToken
viewTransferFee
depositReward
claim
withdrawBNB
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L1122,1139

Description

Constant state variables should be declared constant to save gas.

```
maxSupply
factoryAddress
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L1235,1260,1264,1268,1272,1276,1292,1296,1302,1321,1325,1329

Description

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

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

```
_minHolder
_isClaimable
_rewardToken
_marketingAddress
_exclude
_addresses
_isLp
_lpAddress
_fee
...
```

Recommendation

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

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L1122

Description

There are segments that contain unused state variables.

```
factoryAddress
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1264,1268,1272,1276,1329

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
minHolder = _minHolder
sellMarketingFeeRate = _fee
buyMarketingFeeRate = _fee
sellRewardFeeRate = _fee
buyRewardFeeRate = _fee
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L410,420,439,453,499,509,472,482,357,385,526,1097,1254

Description

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

```
_mint
verifyCallResult
sendValue
isContract
functionStaticCall
functionDelegateCall
functionCallWithValue
functionCall
...
```

Recommendation

Remove unused functions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L1334,1243

Description

The are variables that are defined in the local scope and are not initialized.

```
i
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

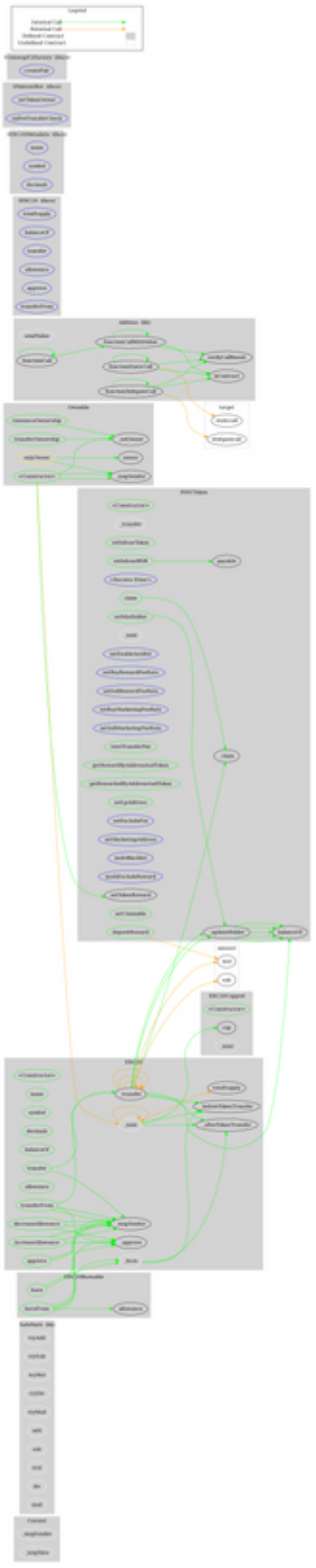
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
SafeMath	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
	div	Internal		
	mod	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	

	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResult	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-

	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
ERC20Burnable	Implementation	Context, ERC20		
	burn	Public	✓	-
	burnFrom	Public	✓	-
ERC20Capped	Implementation	ERC20		
	<Constructor>	Public	✓	-
	cap	Public		-
	_mint	Internal	✓	
IPinkAntiBot	Interface			
	setTokenOwner	External	✓	-
	onPreTransferCheck	External	✓	-
IUniswapV2Factory	Interface			
	createPair	External	✓	-
PINUToken	Implementation	ERC20Capped, ERC20Burnable, Ownable		
	<Constructor>	Public	✓	Ownable ERC20Capped ERC20

	_transfer	Internal	✓	
	withdrawToken	Public	✓	onlyOwner
	withdrawBNB	Public	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	_updateHolder	Internal	✓	
	claim	Public	✓	-
	depositReward	Public	✓	onlyOwner
	_claim	Internal	✓	
	_mint	Internal	✓	
	setEnableAntiBot	External	✓	onlyOwner
	setBuyRewardFeeRate	External	✓	onlyOwner
	setSellRewardFeeRate	External	✓	onlyOwner
	setBuyMarketingFeeRate	External	✓	onlyOwner
	setSellMarketingFeeRate	External	✓	onlyOwner
	viewTransferFee	Public		-
	getRewardByAddressAndToken	Public		-
	getRewardedByAddressAndToken	Public		-
	setLpAddress	Public	✓	onlyOwner
	setExcludeFee	External	✓	onlyOwner
	setMarketingAddress	External	✓	onlyOwner
	multiBlacklist	External	✓	onlyOwner
	multiExcludeReward	External	✓	onlyOwner
	setTokenReward	Public	✓	onlyOwner
	setClaimable	Public	✓	onlyOwner
	setMinHolder	Public	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	pizzainu.io
Registry Domain ID	019cf49e792045feb849ee23ee7e9700-DONUTS
Creation Date	2022-05-25T18:05:10Z
Updated Date	2022-05-30T23:27:24Z
Registry Expiry Date	2024-05-25T18:05:10Z
Registrar WHOIS Server	whois.godaddy.com/
Registrar URL	http://www.godaddy.com/domains/search.aspx?ci=8990
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain has been created 10 days before the creation of the audit. It will expire in almost 2 years.

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

Summary

There are some functions that can be abused by the owner like stopping transactions, manipulating fees, transferring funds to the team’s wallet and massively blacklisting addresses. The contract can be converted into a honeypot and prevent users from selling if the owner abuses the admin functions. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io