Cyberscope

Audit Report

Antcoin

July 2022

SHA256 d8f801d17a0755e8433cd1009ac983cac9b81bdf3649d0a074d700c430b21a61

Audited by © cyberscope

Table of Contents

Contract Review

Contract Name	Antcoin
Testing Deploy	https://testnet.bscscan.com/address/0xEC8eea03ec866514F8e7Caa2131f1c4B4564d974
Symbol	Ants
Decimals	18
Total Supply	1,000,000,000
Domain	https://antcoin.app/en/

Source Files

Filename	**SHA256**
contract.sol	d8f801d17a0755e8433cd1009ac983cac9b81bdf3649d0a074d700c430b21a61

Audit Updates

Initial Audit	8th July 2022
Corrected	15th July 2022

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

OCTD - Owner Contract Tokens Drain

Criticality	minor
Location	contract.sol#L422

Description

The contract owner has the authority to claim all the balance of the contract. The owner may take advantage of it by calling the `emergencyBNBrecovery` function.

```
function emergencyBNBrecovery(uint256 amountPercentage) external onlyOwner {
        uint256 amountBNB = address(this).balance;
        payable(msg.sender).transfer(amountBNB * amountPercentage / 100);
        emit RecoverBNB();
}
```

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.

ULTW - Unlimited Liquidity to Team Wallet

Criticality	minor
Location	contract.sol#L410

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 `LiquidityRelease` method.

```
function LiquidityRelease() public onlyOwner {
        require(block.timestamp >= _liquidityUnlockTime, "Not yet unlocked");
        IBEP20 liquidityToken = IBEP20(_pancakePairAddress);
        uint amount = liquidityToken.balanceOf(address(this));
        if(LPReleaseLimitedTo20Percent)
        {
            _liquidityUnlockTime=block.timestamp+DefaultLiquidityLockTime;
            amount=amount*2/10;
        }
        liquidityToken.transfer(msg.sender, amount);
        emit OnReleaseLP();
}
```

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.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	STC	Succeeded Transfer Check
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L13	Divide before Multiply Operation
	L15	Local Scope Variable Shadowing

STC - Succeeded Transfer Check

Criticality	minor
Location	contract.sol#L418,L326

Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
liquidityToken.transfer(msg.sender, amount);

(bool marketing,)=marketingWallet.call{value:marketbalance}("");
marketing=true;
(bool dev,)=devWallet.call{value:devbalance}("");
```

Recommendation

The contract should check if the result of the transfer methods is successful.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L190,370,283,387,178,61,273,174,383,278,378,410,397,365,390,357,362,170,401,374

Description

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

```
SwapContractToken
LockLiquidityForSeconds
ChangeMarketingWallet
getBurnedTokens
getLiquidityReleaseTimeInSeconds
SetupEnableTrading
SetAMM
limitLiquidityReleaseTo20Percent
LiquidityRelease
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L114,89

Description

Constant state variables should be declared constant to save gas.

```
_circulatingSupply
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L365,170,178,87,110,283,98,410,374,117,118,174,378,88,390,278,370,82,101,83,401,85,436,42

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.

```
AMM
SetAMM
WETH
_owner
_decimals
LockLiquidityForSeconds
_symbol
_liquidityUnlockTime
_name
...
```

Recommendation

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

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L298

Description

Performing divisions before multiplications may cause lose of prediction.

```
LiqHalf = tokenForLiquidity / 2
tokenToSwap = _balances[_pancakePairAddress] * swapTreshold / 1000
```

Recommendation

The multiplications should be prior to the divisions.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L436

Description

The are variables that are defined in the local scope containing the same name from an upper scope.

```
_owner
```

Recommendation

The local variables should have different names from the upper scoped variables.

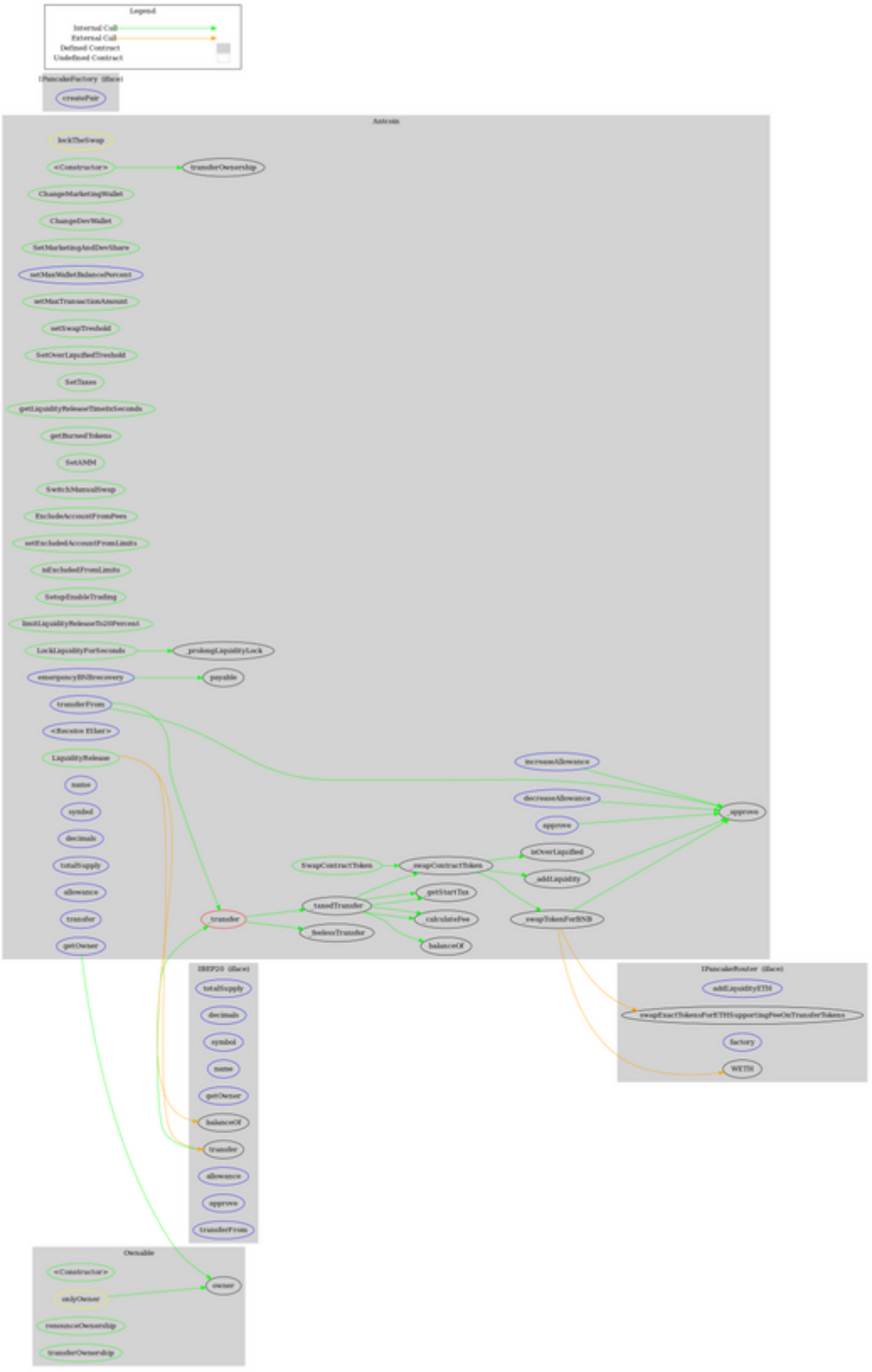
Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IBEP20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IPancakeFactory	Interface			
	createPair	External	✓	-
IPancakeRouter	Interface			
	addLiquidityETH	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	factory	External		-
	WETH	External		-
Ownable	Implementation			
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner

Antcoin	Implementation	IBEP20, Ownable		
	<Constructor>	Public	✓	-
	ChangeMarketingWallet	Public	✓	onlyOwner
	ChangeDevWallet	Public	✓	onlyOwner
	SetMarketingAndDevShare	Public	✓	onlyOwner
	setMaxWalletBalancePercent	External	✓	onlyOwner
	setMaxTransactionAmount	Public	✓	onlyOwner
	_transfer	Private	✓	
	_taxedTransfer	Private	✓	
	_getStartTax	Private		
	_calculateFee	Private		
	_feelessTransfer	Private	✓	
	setSwapTreshold	Public	✓	onlyOwner
	SetOverLiquifiedTreshold	Public	✓	onlyOwner
	SetTaxes	Public	✓	onlyOwner
	isOverLiquified	Public		-
	_swapContractToken	Private	✓	lockTheSwap
	_swapTokenForBNB	Private	✓	
	_addLiquidity	Private	✓	
	getLiquidityReleaseTimeInSeconds	Public		-
	getBurnedTokens	Public		-
	SetAMM	Public	✓	onlyOwner
	SwitchManualSwap	Public	✓	onlyOwner
	SwapContractToken	Public	✓	onlyOwner
	ExcludeAccountFromFees	Public	✓	onlyOwner
	setExcludedAccountFromLimits	Public	✓	onlyOwner
	isExcludedFromLimits	Public		-
	SetupEnableTrading	Public	✓	onlyOwner
	limitLiquidityReleaseTo20Percent	Public	✓	onlyOwner
	LockLiquidityForSeconds	Public	✓	onlyOwner
	_prolongLiquidityLock	Private	✓	
	LiquidityRelease	Public	✓	onlyOwner
	emergencyBNBrecovery	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-

	getOwner	External		-
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	Public		-
	allowance	External		-
	transfer	External	✓	-
	approve	External	✓	-
	_approve	Private	✓	
	transferFrom	External	✓	-
	increaseAllowance	External	✓	-
	decreaseAllowance	External	✓	-

Contract Flow



Domain Info

Domain Name	antcoin.app
Registry Domain ID	494D0CA2C-APP
Creation Date	2022-06-25T20:09:16Z
Updated Date	2022-06-30T20:09:16Z
Registry Expiry Date	2023-06-25T20:09:16Z
Registrar WHOIS Server	whois.nic.google
Registrar URL	None
Registrar	NameSilo, LLC
Registrar IANA ID	1479

The domain has been created in 12 months before the creation of the audit.

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 transferring tokens to the team’s wallet and transferring funds to the team’s wallet. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. There is also a max limit of 20%.

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