Cyberscope

Audit Report

Bitsun

May 2022

Type	BEP20
Network	BSC
Address	0xf892932f6a995660dbe42e25af21536aa734884d
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	
Compiler Version	
Optimization	runs
Licence	
Explorer	https://bscscan.com/token/
Symbol	
Decimals	0
Total Supply	-
Domain	

Source Files

Filename	**SHA256**
contract.sol	b93d0c2da53b3a7841a1a7bb23a8facc54bb73437bb4b100b63d5e8c4d9fedab

Audit Updates

Initial Audit	18th May 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	minor
Location	contract.sol#L1899

Description

The contract does not allow the regular users to trade more than once per day. That means that if a user buy tokens, he will be able to sell them after one dat.

```
if (!(_isBuyAddress[from])) {
    if (transferDays > 0 && lastTransferTime[from] != 0) {
        uint256 time = getUserTransferLockedDays(from);
        uint256 TotalDays = transferDays * 1 days;
        require(
            time > TotalDays,
            "transfer failed, Wait for cooldown time"
        );
    }
}
```

Recommendation

The contract could decrease the trade blocking timespan since one day is a quite long period.

BC - Blacklisted Contracts

Criticality	medium
Location	contract.sol#L1895

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(
    !_isBlacklisted[from] && !_isBlacklisted[to],
    "Blacklisted address"
);
```

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
Medium	BLC	Business Logic Concern
Medium	MC	Missing Check
Minor	L01	Public Function could be Declared External
Minor	L02	State Variables could be Declared Constant
Minor	L04	Conformance to Solidity Naming Conventions
Minor	L07	Missing Events Arithmetic
Minor	L09	Dead Code Elimination
Minor	L11	Unnecessary Boolean equality
Minor	L13	Divide before Multiply Operation
Minor	L14	Uninitialized Variables in Local Scope

BLC - Business Logic Concern

Criticality	medium
Location	contract.sol#L1

Description

The business logic seems peculiar. The implementation may not follow the expected behavior. There are sections that all the calculations are redundant. For instance, if a number x is greater than a number y, then the number x will never be less than or equal to the number y.

```
uint256 tWAmnt = balanceOf(from).mul(_minWTxPercent).div(
    _minWTxPercentDivider
);
if (amount > tWAmnt) {
    require(
        amount <= tWAmnt,
        "Transfer amount exceeds the wallet limit."
    );
}
```

```
uint256 tWAmnt = balanceOf(from).mul(10**9).div(
    totalSupply()
);
uint256 allowedAmnt = tWAmnt.mul(yesterdaysBuy).div(10**9);
if (amount > allowedAmnt) {
    require(
        amount <= allowedAmnt,
        "Transfer amount exceeds the user allowed limit as per yesterday's buy."
    );
}
```

```
uint256 todaysSaleAmount = GetTodaysSale();
// uint256 remAmnt=yesterdaysBuy.sub(todaysSaleAmount);
//  if(remAmnt.sub(amount) < 0)
if (todaysSaleAmount.add(amount) > yesterdaysBuy) {
    require(
        todaysSaleAmount.add(amount) <= yesterdaysBuy,
```

```
        "Transfer amount exceeds the total allowed limit as per yesterday's
buy."
    );
}
```

Recommendation

The team is advised to carefully check if the implementation follows the expected business logic.

MC - Missing Check

Criticality	medium
Location	contract.sol#L1933

Description

The contract is processing variables that have not properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues.

The variable `_minWTxPercentDivider` is used as a divisor. The setter function `setMinWTxPercentDivider` can set the zero value. As a result the method will revert.

```
uint256 tWAmnt = balanceOf(from).mul(_minWTxPercent).div(
    _minWTxPercentDivider
);
```

Recommendation

The contract should properly check the variables according to the required specifications

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L900,908,1692,957,970,987,1010,1041,1068,1294,1302,1394,1516,1520,1540,1548,1557,1704,1753,1757,1761,1768,1772,1782,1842,1866,1987,1997,2003,2031,2035,2045,2055

Description

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

```
GetTodaysBuy
GetDateCurrent
GetSaleOfDate
GetBuyOfDate
GetUserRemainingEligibleSaleForDay
GetRemainingEligibleSaleForDay
GetTimeForOpenSale
isBuyAddress
canBypassCheck
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L1641,1637

Description

Constant state variables should be declared constant to save gas.

```
transferDays
_noCheckMinsForDay
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L92,370,372,403,1328,1337,1987,1997,2003,2031,2035,2039,2045,2055,2060,2065,2073,2078,2105,2106,2107,2112,1634,1635,1636,1638,1639,1641,1643,1649,1650,1651,1653,1654,1655

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.

```
_excludedFromPercent
_isExcludedFromPercent
_isBlacklisted
_bypassAddresses
_bypassAddressBalance
_canBypassAddress
_dt
_noCheckMinsForDay
_voteStartTime
...
```

Recommendation

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

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1846,1850,1858,1862

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.

```
_minTxAmount = value.mul(10 ** 9)
_minWTxPercentDivider = value
_percentForBypass = value
_minWTxPercent = value
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L2085,1156,42,58

Description

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

```
toHexString
_burn
swapTokensForBnb
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L1704,1782,1884,2003

Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
bypassCheckFlag == false && _canBypassAddress[_userAddress]
bypassCheckFlag == true
noCheckFlag == true
require(bool,string)((_canBypassAddress[from] || _canBypassAddress[to]) ==
false,Address Locked For Transaction as set to bypass buy check.)
noCheckFlag == false
bypassCheckFlag == false && (_canBypassAddress[from] || _canBypassAddress[to])
canBypass == true
require(bool,string)(bypassCheckFlag != true,Bypass check already set)
excluded == true
...
```

Recommendation

Remove the equality to the boolean constant.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L1884,2003

Description

Performing divisions before multiplications may cause lose of prediction.

```
tWAmnt = balanceOf(_userAddress).mul(10 ** 9).div(totalSupply())
tWAmnt_scope_0 = balanceOf(from).mul(10 ** 9).div(totalSupply())
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L1402

Description

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

```
dt
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
IUniswapV2Router02	Interface	IUniswapV2Router01		

	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Pair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-

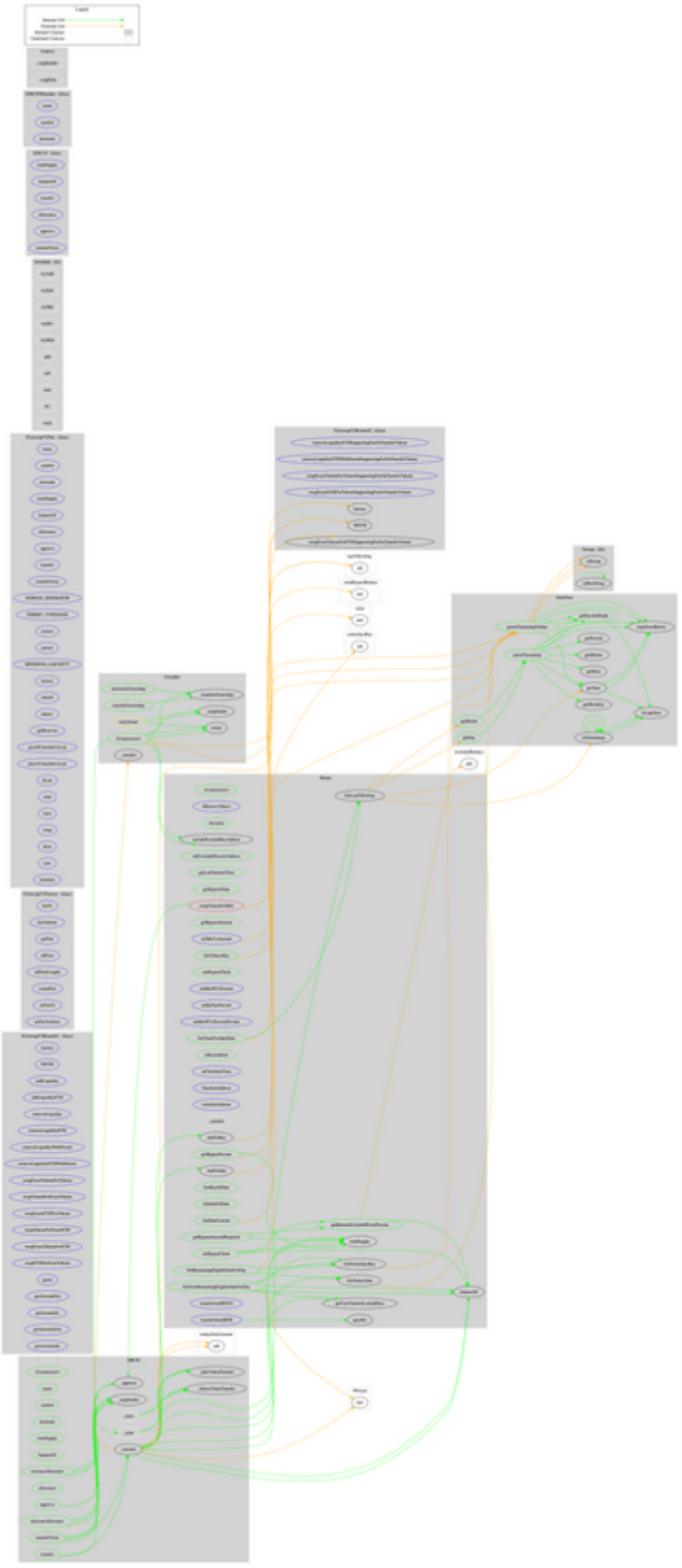
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
SafeMath	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		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-

	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
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	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-

	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
DateTime	Implementation			
	isLeapYear	Public		-
	leapYearsBefore	Public		-
	getDaysInMonth	Public		-
	parseTimestampToDate	Public		-
	parseTimestamp	Public		-
	getYear	Public		-
	getMonth	Public		-
	getDay	Public		-
	getHour	Public		-
	getMinute	Public		-
	getSecond	Public		-
	getWeekday	Public		-
	toTimestamp	Public		-
	toTimestamp	Public		-
	toTimestamp	Public		-
	toTimestamp	Public		-
Bitsun	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	decimals	Public		-
	includeExcludeBuyAddress	Public	✓	onlyOwner
	setExcludedPercentAddress	Public	✓	onlyOwner
	getBalanceExcludedFromPercent	Public		-
	getUserTransferLockedDays	Public		-
	getLastTransferTime	Public		-
	getBypassValue	Public		-
	getBypassPercent	Public		-
	getBypassAmount	Public		-

	getBypassAmountRequired	Public		-
	setBypassCheck	Public	✓	-
	canBypassCheck	Public		-
	setMinWTxPercent	External	✓	onlyOwner
	setByPassPercent	External	✓	onlyOwner
	setMinWTxPercentDivider	External	✓	onlyOwner
	setMinTxAmount	External	✓	onlyOwner
	isBuyAddress	Public		-
	setVoteStartTime	External	✓	onlyOwner
	blacklistAddress	External	✓	onlyOwner
	whitelistAddress	External	✓	onlyOwner
	_transfer	Internal	✓	
	GetTimeForOpenSale	Public		-
	GetRemainingEligibleSaleForDay	Public		-
	GetUserRemainingEligibleSaleForDay	Public		-
	GetBuyOfDate	Public		-
	GetSaleOfDate	Public		-
	GetYesterdaysBuy	Private		
	GetDateCurrent	Public		onlyOwner
	GetTodaysBuy	Public		-
	GetTodaysSale	Public		-
	GetLastTsForDay	Private		
	AddToBuy	Private	✓	
	AddToSale	Private	✓	
	swapTokensForBnb	Private	✓	
	transferAnyBEP20	External	✓	onlyOwner
	transferStuckBNB	External	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	bitsun.app
Registry Domain ID	48B11CD86-APP
Creation Date	2022-04-17T18:52:56Z
Updated Date	2022-04-26T11:05:06Z
Registry Expiry Date	2024-04-17T18:52:56Z
Registrar WHOIS Server	whois.godaddy.com
Registrar URL	https://www.godaddy.com/
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain has been created about 1 month 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 and blacklisting addresses. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate some of the contract threats. The contract contains many sections with redundant calculations that may lead to an odd state.

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 provides all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io