



Cyberscope

Audit Report

LIKE 1000X

August 2022

Type BEP20

Network BSC

Address 0xDD8bafEb38C53Ea3B7754cc0D90eE2f2Afef14a9

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Audit Updates	3
Source Files	3
Contract Analysis	5
Contract Diagnostics	6
US - Untrusted Source	7
Description	7
Recommendation	7
CR - Code Repetition	8
Description	8
Recommendation	8
SEC - Safe External Call	9
Description	9
Recommendation	9
L01 - Public Function could be Declared External	10
Description	10
Recommendation	10
L02 - State Variables could be Declared Constant	11
Description	11
Recommendation	11
L04 - Conformance to Solidity Naming Conventions	12
Description	12
Recommendation	12
L09 - Dead Code Elimination	13
Description	13

Recommendation	13
L12 - Using Variables before Declaration	14
Description	14
Recommendation	14
L14 - Uninitialized Variables in Local Scope	15
Description	15
Recommendation	15
Contract Functions	16
Contract Flow	23
Domain	24
Summary	25
Disclaimer	26
About Cyberscope	27

Contract Review

Contract Name	LIKE1000X
Compiler Version	v0.6.12+commit.27d51765
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xDD8bafEb38C53Ea3B7754cc0D90eE2f2Afef14a9
Symbol	LIKES
Decimals	18
Total Supply	1,000,000,000
Domain	https://like1000x.one

Audit Updates

Initial Audit	15th August 2022
Corrected	

Source Files

Filename	SHA256
Context.sol	ade730fe55d7b995a6a9a81f77600d10d9ea7472be54a290a905f853495bce97
DividendPayingToken.sol	11e0755b6cce3a73e5f515418928454324cc31d230e2f33c6a55f8ee1e1062b9
DividendPayingTokenInterface.sol	7d7301f0a6321c9a83e2544342327c9eeaffd7476424e60f2f8badd302c94053

DividendPayingTokenOptionalInterface.sol	613ef8cfcd377b92e0a456548c6560ee3dd18d9a253850fcdd6b9036337feb6b
ERC20.sol	c9352c9260d5c9261d5c5449cb864887720a316ca241020d5c8a2a0e0c841fb0
IERC20.sol	40b62888fbeb089db2a8060f52214b2aba38abd295b9c6b90fbc8b52ba5158b6
IERC20Metadata.sol	5453d34cc9db3921a16eb83a551e1a9285d9285806c119d5caf399352f6bf1a6
IterableMapping.sol	4e1661030209caf939a716c3dbc413f704fc7b6d6cdb5a957f1f48e693d30d23
IUniswapV2Factory.sol	cfac3b608fe9c5c10db6e7dfbd0e52600689b41848cbb7c3f6d074ebca8b545f
IUniswapV2Pair.sol	522717b02bc1839e9024e49d6d93ebb976be01a5d0b7e459c3d50fd3dbfe6cb2
IUniswapV2Router.sol	ac1b9a6719ad80130195805ec526188b3dd3a84ddd5b8e6ac92169abfa415a04
Like1000x.sol	00a5e0dd37835e31f58767de699395afdcb0f6636da0bcde48a3114374bff267
Ownable.sol	fb7658fc325cceffba19f6cf9809119bf073d1d4cccdbbf9d439a34ff062934c
SafeMath.sol	253b3928dd6338470c3cc18945de79fa9ec77b12a36948aa36ae3e5771851fba
SafeMathInt.sol	9345ec14af97a2ed2238153d853257bcd209a8d4de6de8cb1152711bc94402bd
SafeMathUint.sol	87eae8174207cfb48ac338ad99eeeeee1e989ec141144b3b30a2ef90e08d8fb9f

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

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	US	Untrusted Source
●	CR	Code Repetition
●	SEC	Safe External Call
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L09	Dead Code Elimination
●	L12	Using Variables before Declaration
●	L14	Uninitialized Variables in Local Scope

US - Untrusted Source

Criticality	minor
Location	contract.sol#L118

Description

The contract could set an alternative external contract as a dividend tracker in order to determine the transaction's flow. The external contract is untrusted. As a result it may produce security issues and harm the transactions.

```
function updateDividendTracker(address newAddress) public onlyOwner {
    require(newAddress != address(dividendTracker), "LIKE1000X: The dividend
    tracker already has that address");

    LIKE1000XDividendTracker newDividendTracker =
    LIKE1000XDividendTracker(payable(newAddress));

    require(newDividendTracker.owner() == address(this), "LIKE1000X: The new
    dividend tracker must be owned by the LIKE1000X token contract");

    newDividendTracker.excludeFromDividends(address(newDividendTracker));
    newDividendTracker.excludeFromDividends(address(this));
    newDividendTracker.excludeFromDividends(address(uniswapV2Router));

    emit UpdateDividendTracker(newAddress, address(dividendTracker));

    dividendTracker = newDividendTracker;
}
```

Recommendation

The contract should use a trusted external source. A trusted source could be either a commonly recognized or an audited contract. The pointing addresses should not be able to change after the initialization.

CR - Code Repetition

Criticality

minor

Location

contract.sol#L352-402

Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily.

The methods 'swapTokensForEth' and 'swapTokensForBNB' are identical. The methods 'swapEthForCustomToken' and 'swapTokensForEth' are not used by the contract.

```
function swapTokensForEth(uint256 tokenAmount) private {  
    ...  
}  
function swapTokensForBNB(uint256 tokenAmount) private {  
    ...  
}  
function swapEthForCustomToken(uint256 tokenAmount) private {  
    ...  
}
```

Recommendation

The contract could remove the unnecessary method declarations.

SEC - Safe External Call

Criticality	medium
Location	contract.sol#L408

Description

The `sendDividends()` method is used in the swap and liquify feature in order to transfer funds to the dividend tracker. Since the dividend tracker is an external contract, it may propagate an exception to the LIKE1000X contract and revert the transactions. Since the swap feature may be called in sale transactions, it may prevent the users from selling.

```
(bool success,)=address(dividendTracker).call{value: tokens, gas: 3000}("");  
if (success) {  
    dividendTracker.distributeBNBDividends(tokens);  
}
```

Recommendation

The contract should guard the external calls with try-catch statements, so it cannot be affected by potential exceptions.

L01 - Public Function could be Declared External

Criticality	minor
Location	Like1000x.sol#L118,134,149,157,175,194,198,202,516,560

Description

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

```
process
getAccountAtIndex
dividendTokenBalanceOf
withdrawableDividendOf
isExcludedFromFees
updateGasForProcessing
setAutomatedMarketMakerPair
excludeMultipleAccountsFromFees
updateUniswapV2Router
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	Like1000x.sol#L29,35,24,30,28,26

Description

Constant state variables should be declared constant to save gas.

```
swapTokensAtAmount  
marketingFee  
lpFee  
deadWallet  
_lpWallet  
BNBRewardsFee
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality

minor

Location

Like1000x.sol#L29,34,471

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.

```
_account  
_marketingWallet  
BNBRewardsFee
```

Recommendation

Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>.

L09 - Dead Code Elimination

Criticality

minor

Location

Like1000x.sol#L390,352

Description

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

```
swapTokensForEth  
swapEthForCustomToken
```

Recommendation

Remove unused functions.

L12 - Using Variables before Declaration

Criticality

minor

Location

Like1000x.sol#L326

Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
iterations  
claims  
lastProcessedIndex
```

Recommendation

The variables should be declared before any usage of them.

L14 - Uninitialized Variables in Local Scope

Criticality

minor

Location

Like1000x.sol#L326

Description

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

```
iterations  
lastProcessedIndex  
claims
```

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		
DividendPayingToken	Implementation	ERC20, Ownable, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface		
	<Constructor>	Public	✓	ERC20
	distributeBNBDividends	Public	✓	onlyOwner
	withdrawDividend	Public	✓	-
	_withdrawDividendOfUser	Internal	✓	
	dividendOf	Public		-
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_setBalance	Internal	✓	
	<Receive Ether>	External	Payable	-
DividendPayingTokenInterface	Interface			
	dividendOf	External		-
	withdrawDividend	External	✓	-

DividendPayingTokenOptionallInterface	Interface			
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	External		-
ERC20	Implementation	Context, IERC20, IERC20Meta data		
	<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	✓	
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-

IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
IterableMapping	Library			
	get	Public		-
	getIndexOfKey	Public		-
	getKeyAtIndex	Public		-
	size	Public		-
	set	Public	✓	-
	remove	Public	✓	-
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	✓	-

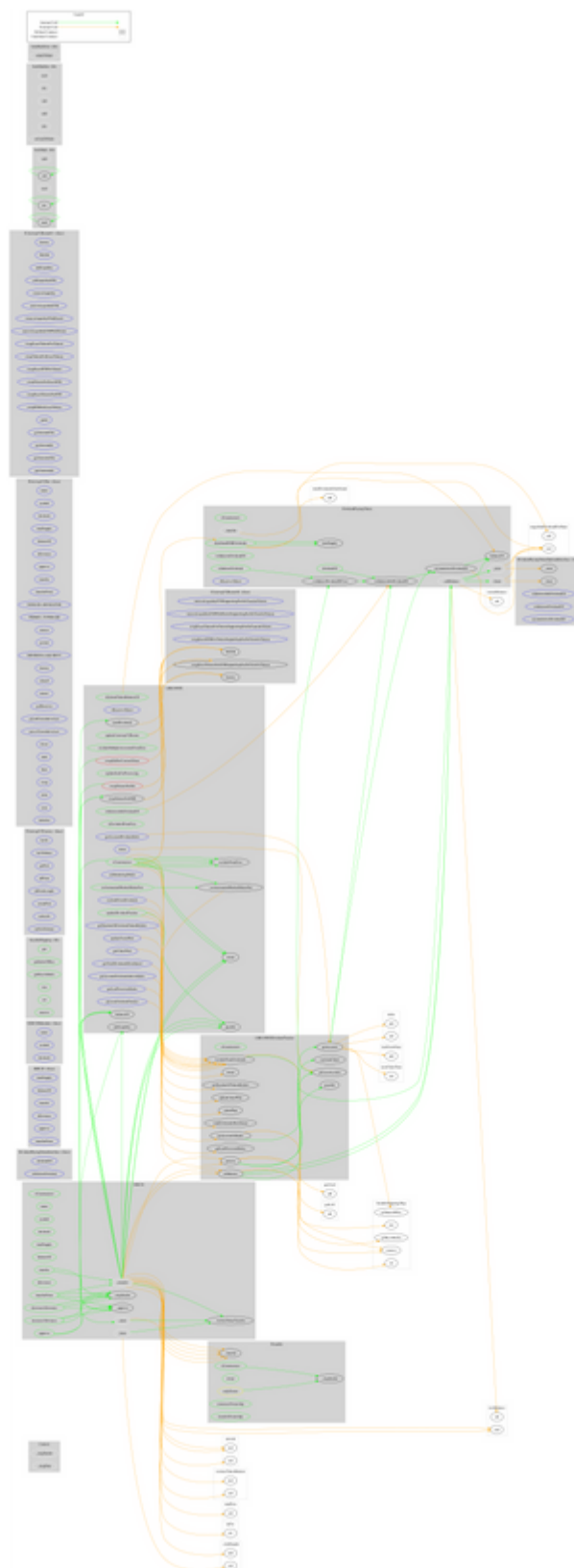
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
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	✓	-
LIKE1000X	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	updateDividendTracker	Public	✓	onlyOwner
	updateUniswapV2Router	Public	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	excludeMultipleAccountsFromFees	Public	✓	onlyOwner
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	updateGasForProcessing	Public	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getClaimWait	External		-
	getTotalDividendsDistributed	External		-
	isExcludedFromFees	Public		-
	withdrawableDividendOf	Public		-
	dividendTokenBalanceOf	Public		-
	excludeFromDividends	External	✓	onlyOwner
	setMarketingWallet	External	✓	onlyOwner

	getAccountDividendsInfo	External		-
	getAccountDividendsInfoAtIndex	External		-
	processDividendTracker	External	✓	-
	claim	External	✓	-
	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	_transfer	Internal	✓	
	addLiquidity	Private	✓	
	swapTokensForEth	Private	✓	
	swapTokensForBNB	Private	✓	
	swapEthForCustomToken	Private	✓	
	sendDividends	Private	✓	
LIKE1000XDividendTracker	Implementation	Ownable, DividendPayingToken		
	<Constructor>	Public	✓	DividendPayingToken
	_transfer	Internal	✓	
	withdrawDividend	Public	✓	-
	excludeFromDividends	External	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getLastProcessedIndex	External		-
	getNumberOfTokenHolders	External		-
	getAccount	Public		-
	getAccountAtIndex	Public		-
	canAutoClaim	Private		
	setBalance	External	✓	onlyOwner
	process	Public	✓	-
	processAccount	Public	✓	onlyOwner
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner

SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
	toUint256Safe	Internal		
SafeMathUint	Library			
	toInt256Safe	Internal		

Contract Flow



Domain

Domain Name	like1000x.one
Registry Domain ID	DA27D8055F4424C14A348C05F52D0CDCE-GDREG
Creation Date	2022-07-21T14:27:51Z
Updated Date	2022-07-26T14:27:52Z
Registry Expiry Date	2023-07-21T14:27:51Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	http://www.namecheap.com
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created in 11 months before the creation of the audit.

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

Summary

LIKE1000X is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. There is also a limit of max 25% fees.

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>