Cyberscope

Audit Report

Golden Retriever

September 2022

Type	BEP20
Network	BSC
Address	0x844d92ED44070B81A2eDC895cd715c4D8fF0DE94
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	GoldenRetriever
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0x844d92ED44070B81A2eDC895cd715c4D8fF0DE94
Symbol	GRV
Decimals	8
Total Supply	300,000,000,000
Domain	http://www.goldenretriever.finance

Source Files

Filename	**SHA256**
contract.sol	bf1427288caf3cc041751d405c22b0b4260ff93225791379452e5134d09a130a

Audit Updates

Initial Audit	21st September 2022
Corrected	

Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OCTD	Transfers Contract's Tokens	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	ULTW	Transfers Liquidity to Team Wallet	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	FSA	Fixed Swap Address	Unresolved
●	CO	Code Optimization	Unresolved
●	L01	Public Function could be Declared External	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L03	Redundant Statements	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L09	Dead Code Elimination	Unresolved

FSA - Fixed Swap Address

Criticality	minor / informative
Location	contract.sol#L314
Status	Unresolved

Description

The swap address is assigned once in the constructor and it can not be changed. The decentralized swaps sometimes create a new swap version or abandon the current. A contract that cannot change the swap address may not be able to catch-up the upgrade.

```
constructor () {

        IUniswapV2Router02 _uniswapV2Router =
IUniswapV2Router02(0x10ED43C718714eb63d5aA57B78B54704E256024E);

        uniswapPair = IUniswapV2Factory(_uniswapV2Router.factory())
            .createPair(address(this), _uniswapV2Router.WETH());
```

Recommendation

It could be better to allow the swap address mutation in case of future swap updates.

CO - Code Optimization

Criticality	minor / informative
Location	contract.sol#L506
Status	Unresolved

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.

Since the variable _liquidityShare is always set to zero. The code segment to add liquidity is redundant.

```
function swapAndLiquify(uint256 tAmount) private lockTheSwap {

        uint256 tokensForLP =
tAmount.mul(_liquidityShare).div(_totalDistributionShares).div(2);
        uint256 tokensForSwap = tAmount.sub(tokensForLP);

        swapTokensForEth(tokensForSwap);
        uint256 amountReceived = address(this).balance;

        uint256 totalBNBFee = _totalDistributionShares.sub(_liquidityShare.div(2));

        uint256 amountBNBLiquidity =
amountReceived.mul(_liquidityShare).div(totalBNBFee).div(2);
        uint256 amountBNBBurned = amountReceived.mul(_BurnedShare).div(totalBNBFee);
        uint256 amountBNBMarketing =
amountReceived.sub(amountBNBLiquidity).sub(amountBNBBurned);

        if(amountBNBMarketing > 0)
            transferToAddressETH(marketingWalletAddress, amountBNBMarketing);

        if(amountBNBBurned > 0)
            transferToAddressETH(BurnedWalletAddress, amountBNBBurned);

        if(amountBNBLiquidity > 0 && tokensForLP > 0)
            addLiquidity(tokensForLP, amountBNBLiquidity);
}
```

Since the variable coolBlock is always set to zero. The code segment to burn the the transaction that take place in the first blocks is redundant.

```
if (block.number < ( genesisBlock + coolBlock) && sender == uniswapPair )
 {
    _basicTransfer(recipient,deadAddress, finalAmount);
 }
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L343,339,400,117,335,394,369,347,410,122,373,355,386,364,415,359
Status	Unresolved

Description

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

```
decimals
symbol
getCirculatingSupply
renounceOwnership
name
setIsExcludedFromFee
minimumTokensBeforeSwapAmount
totalSupply
transfer
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	contract.sol#L241,257,242,259,240,261,244,278,256,260,245,279,258
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
_symbol
_buyMarketingFee
_decimals
_sellLiquidityFee
_name
_sellBurnedFee
marketingWalletAddress
coolBlock
_buyLiquidityFee
...
```

Recommendation

Add the constant attribute to state variables that never change.

L03 - Redundant Statements

Criticality	minor / informative
Location	contract.sol#L4
Status	Unresolved

Description

The contract contains statements that are not used and have no effect. As a result, those segments increase the code size of the contract unnecessarily.

```
Context
```

Recommendation

Remove the redundant statements in order to decrease the code size.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L256,269,267,259,264,260,248,245,258,268,265,261,142,257,263,279
Status	Unresolved

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.

```
_buyLiquidityFee
_totalDistributionShares
_totalTaxIfBuying
_sellLiquidityFee
_marketingShare
_sellMarketingFee
_balances
BurnedWalletAddress
_buyBurnedFee
...
```

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 / informative
Location	contract.sol#L88,81
Status	Unresolved

Description

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

```
sendValue
isContract
```

Recommendation

Remove unused functions.

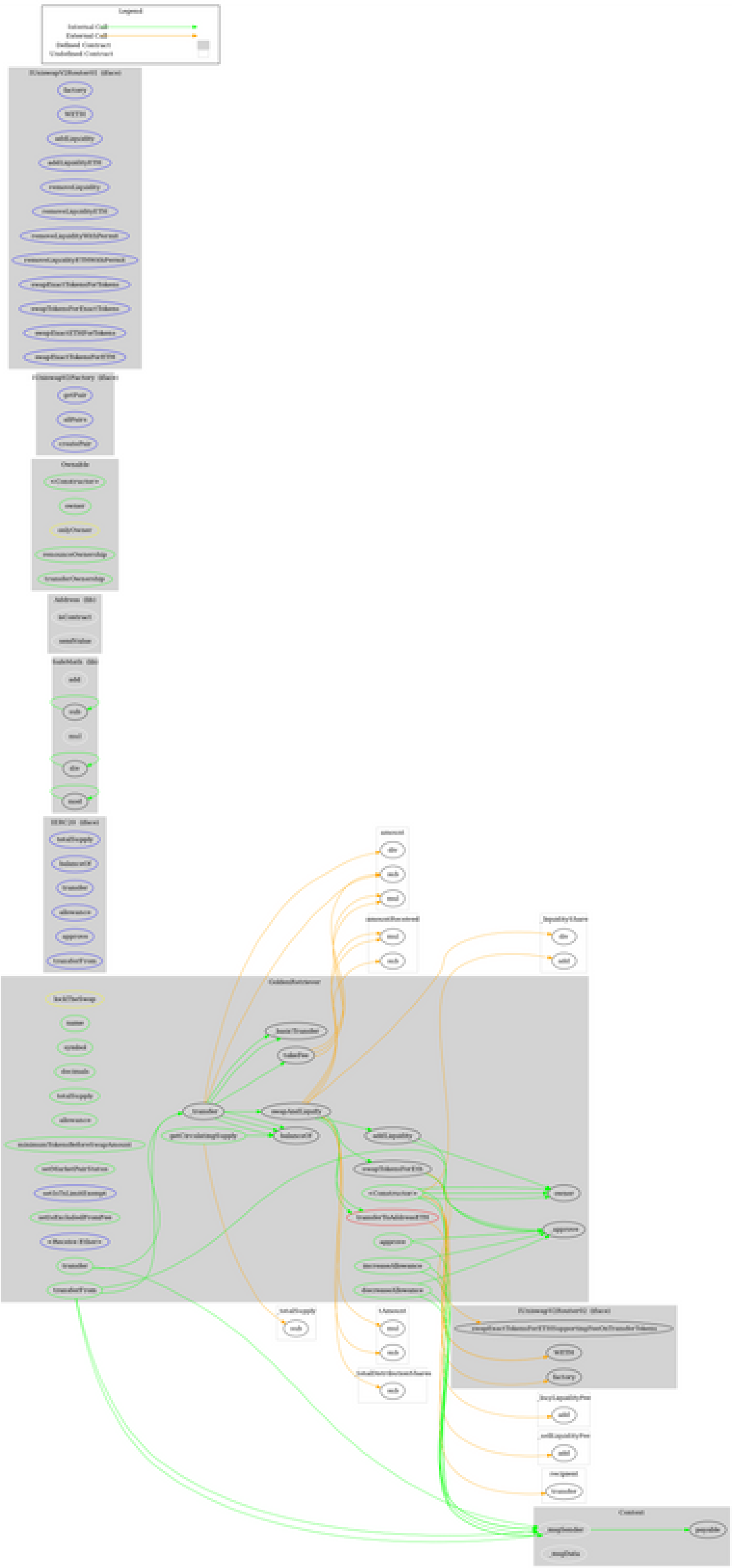
Contract Functions

Contract	**Type**	**Bases**		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
Ownable	Implementation			
	<Constructor>	Public	✓	-
	owner	Public		-

	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
IUniswapV2Factory	Interface			
	getPair	External		-
	allPairs	External		-
	createPair	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	-
	swapExactTokensForETH	External	✓	-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
GoldenRetriever	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-

	balanceOf	Public		-
	allowance	Public		-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	minimumTokensBeforeSwapAmount	Public		-
	approve	Public	✓	-
	_approve	Private	✓	
	setMarketPairStatus	Public	✓	onlyOwner
	setIsTxLimitExempt	External	✓	onlyOwner
	setIsExcludedFromFee	Public	✓	onlyOwner
	getCirculatingSupply	Public		-
	transferToAddressETH	Private	✓	
	<Receive Ether>	External	Payable	-
	transfer	Public	✓	-
	transferFrom	Public	✓	-
	_transfer	Private	✓	
	_basicTransfer	Internal	✓	
	swapAndLiquify	Private	✓	lockTheSwap
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	takeFee	Internal	✓	

Contract Flow



Domain Info

Domain Name	goldenretriever.finance
Registry Domain ID	6d632a042c4a4531afcbfd130ff17011-DONUTS
Creation Date	2022-09-08T12:11:49Z
Updated Date	2022-09-13T12:12:36Z
Registry Expiry Date	2023-09-08T12:11:49Z
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 was created 13 days before the creation of the audit. It will expire in 12 months.

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

Summary

Golden Retriever Token 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 fee of 2%. The accumulated tokens are liquidated to the 0x840E696ddD64B16B734F774B8893Fc2549D329B9 and 0xa92517E124B77E452c48dD362849F7CC85859937 addresses.

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