Cyberscope

Audit Report

FIFA Inu

July 2022

Type	BEP20
Network	BSC
Address	0x4182a8fa344a59b7bf813e29954e4ca07701571c
Audited by	© cyberscope

Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 3
Contract Analysis 4
Contract Diagnostics 5
US - Untrusted Source 6
Description 6
Recommendation 6
STC - Succeeded Transfer Check 7
Description 7
Recommendation 7
L01 - Public Function could be Declared External 8
Description 8
Recommendation 8
L04 - Conformance to Solidity Naming Conventions 9
Description 9
Recommendation 9
L07 - Missing Events Arithmetic 10
Description 10
Recommendation 10
L12 - Using Variables before Declaration 11
Description 11
Recommendation 11
L14 - Uninitialized Variables in Local Scope 12
Description 12

Recommendation 12
L15 - Local Scope Variable Shadowing 13
Description 13
Recommendation 13
Contract Functions 14
Contract Flow 21
Domain Info 22
Summary 23
Disclaimer 24
About Cyberscope 25

Contract Review

Contract Name	Fifalnu
Compiler Version	v0.8.15+commit.e14f2714
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x4182a8fa344a59b7bf813e29954e4ca07701571c
Symbol	FINU
Decimals	18
Total Supply	100,000,000,000
Domain	https://fifainu.finance

Source Files

Filename	**SHA256**
contract.sol	512f879a1b8b25daeb3b1e1776222eff1a0375a006046420d3e164034f73fc01

Audit Updates

Initial Audit	18th July 2022
Corrected	27th 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

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	US	Untrusted Source
	STC	Succeeded Transfer Check
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L07	Missing Events Arithmetic
	L12	Using Variables before Declaration
	L14	Uninitialized Variables in Local Scope
	L15	Local Scope Variable Shadowing

US - Untrusted Source

Criticality	minor
Location	contract.sol#L1325

Description

The contract uses an external contract in order to determine the transaction's flow. The external contract is untrusted. As a result the accumulated funds may be transferred to an untrusted source.

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

    DividendTracker newDividendTracker = DividendTracker(payable(newAddress));

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

    newDividendTracker.excludeFromDividends(address(newDividendTracker));
    newDividendTracker.excludeFromDividends(address(this));
    newDividendTracker.excludeFromDividends(owner());
    newDividendTracker.excludeFromDividends(address(uniswapV2Router));
    newDividendTracker.excludeFromDividends(address(0xdead));

    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.

STC - Succeeded Transfer Check

Criticality	minor
Location	contract.sol#L1700

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.

```
IERC20(_token).transfer (_account, tokenBalance);
```

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#L1730

Description

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

```
withdrawDividend
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L1764,1679,57,74,1000,1198,56,1369,1089,1103,1096,1688,859,1113

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.

```
_marketingFee
_owner
_account
WETH
_token
_rewardsFee
_liquidityFee
DOMAIN_SEPARATOR
_isExcludedMaxTxn
...
```

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#L1369

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.

```
rewardsFee = _rewardsFee
```

Recommendation

Emit an event for critical parameter changes.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract.sol#L1598

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.

```
claims
iterations
lastProcessedIndex
```

Recommendation

The variables should be declared before any usage of them.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L1598

Description

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

```
iterations
claims
lastProcessedIndex
```

Recommendation

All the local scoped variables should be initialized.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L1022,1113,1096,1248,1089,1103

Description

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

```
_owner
totalSupply
_symbol
_name
```

Recommendation

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

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
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	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IterableMapping	Library			
	get	External		-
	getIndexOfKey	External		-
	getKeyAtIndex	External		-
	size	External		-
	set	Public	✓	-
	remove	Public	✓	-
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	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
	increaseAllowance	External	✓	-
	decreaseAllowance	External	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
DividendPayingTokenOptionalInterface	Interface			
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	External		-
DividendPayingTokenInterface	Interface			
	dividendOf	External		-
	distributeDividends	External	Payable	-

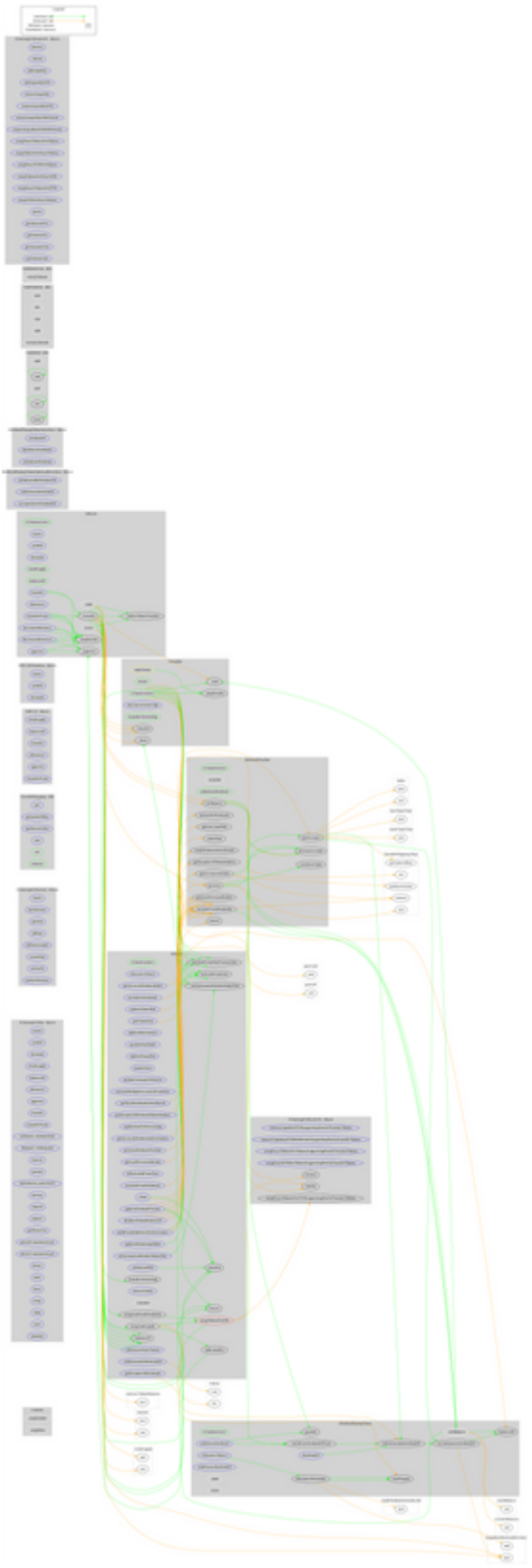
	withdrawDividend	External	✓	-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	External	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	toUint256Safe	Internal		
SafeMathUint	Library			
	toInt256Safe	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	✓	-
DividendPayingToken	Implementation	ERC20, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface, Ownable		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	distributeDividends	Public	Payable	-

	withdrawDividend	External	✓	-
	_withdrawDividendOfUser	Internal	✓	
	dividendOf	External		-
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	Public		-
	_mint	Internal	✓	
	_burn	Internal	✓	
	_setBalance	Internal	✓	
Fifalnu	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	addPresaleAddressForExclusions	External	✓	onlyOwner
	updateDividendTracker	External	✓	onlyOwner
	excludeFromDividends	External	✓	onlyOwner
	includeInDividends	External	✓	onlyOwner
	updateMaxAmount	External	✓	onlyOwner
	updateMaxWallet	External	✓	onlyOwner
	updateSwapSize	External	✓	onlyOwner
	updateFees	External	✓	onlyOwner
	updateUniswapV2Router	External	✓	onlyOwner
	excludeFromMaxTransaction	Public	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	excludeMultipleAccountsFromFees	External	✓	onlyOwner
	setAutomatedMarketMakerPair	External	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	updateMarketingWallet	External	✓	onlyOwner
	updateGasForProcessing	External	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getClaimWait	External		-
	getTotalDividendsDistributed	External		-
	isExcludedFromFees	External		-
	withdrawableDividendOf	External		-

	dividendTokenBalanceOf	External		-
	getAccountDividendsInfo	External		-
	getAccountDividendsInfoAtIndex	External		-
	processDividendTracker	External	✓	-
	claim	External	✓	-
	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	getNumberOfDividends	External		-
	removeLimits	External	✓	onlyOwner
	_transfer	Internal	✓	
	swapTokensForEth	Private	✓	
	swapAndLiquify	Private	✓	
	addLiquidity	Private	✓	
	swapAndSendDividends	Private	✓	
	withdrawBNB	External	✓	onlyOwner
	withdrawOtherTokens	External	✓	onlyOwner
DividendTracker	Implementation	DividendPayingToken		
	<Constructor>	Public	✓	DividendPayingToken
	_transfer	Internal		
	withdrawDividend	Public		-
	excludeFromDividends	External	✓	onlyOwner
	includeInDividends	External	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getLastProcessedIndex	External		-
	getNumberOfTokenHolders	External		-
	getAccount	Public		-
	getAccountAtIndex	External		-
	canAutoClaim	Private		
	setBalance	External	✓	onlyOwner
	process	External	✓	-
	processAccount	Public	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	fifainu.finance
Registry Domain ID	6592a3fdd18744b193e92d9be99e126c-DONUTS
Creation Date	2022-07-14T14:02:55Z
Updated Date	2022-07-16T10:57:56Z
Registry Expiry Date	2023-07-14T14:02:55Z
Registrar WHOIS Server	whois.namesilo.com
Registrar URL	http://www.namesilo.com
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

FIFA Inu 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

Cyberscope

Audit Report

FIFA Inu

July 2022

Type	BEP20
Network	BSC
Address	0xA8775015Cb527b9bF6d392eB1B584D9b42feE642
Audited by	© cyberscope

Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 3
Contract Analysis 4
Contract Diagnostics 5
US - Untrusted Source 6
Description 6
Recommendation 6
STC - Succeeded Transfer Check 7
Description 7
Recommendation 7
L01 - Public Function could be Declared External 8
Description 8
Recommendation 8
L04 - Conformance to Solidity Naming Conventions 9
Description 9
Recommendation 9
L05 - Unused State Variable 10
Description 10
Recommendation 10
L07 - Missing Events Arithmetic 11
Description 11
Recommendation 11
L09 - Dead Code Elimination 12
Description 12

Recommendation 12
L12 - Using Variables before Declaration 13
Description 13
Recommendation 13
L14 - Uninitialized Variables in Local Scope 14
Description 14
Recommendation 14
L15 - Local Scope Variable Shadowing 15
Description 15
Recommendation 15
Contract Functions 16
Contract Flow 24
Domain Info 25
Summary 26
Disclaimer 27
About Cyberscope 28

Contract Review

Contract Name	Fifalnu
Compiler Version	v0.8.15+commit.e14f2714
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xA8775015Cb527b9bF6d392eB1B584D9b42feE642
Symbol	FINU
Decimals	18
Total Supply	99,999,999,999
Domain	fifainu.finance

Source Files

Filename	**SHA256**
contract.sol	532ed982e01d5a08aadce50579fda0672ab11bf50a8da838b82f6e27b1a8b1b1

Audit Updates

Initial Audit	18th July 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

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	US	Untrusted Source
	STC	Succeeded Transfer Check
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L05	Unused State Variable
	L07	Missing Events Arithmetic
	L09	Dead Code Elimination
	L12	Using Variables before Declaration
	L14	Uninitialized Variables in Local Scope
	L15	Local Scope Variable Shadowing

US - Untrusted Source

Criticality	minor
Location	contract.sol#L1313

Description

The contract uses an external contract in order to determine the transaction's flow. The external contract is untrusted. As a result the accumulated funds may be transferred to an untrusted source.

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

    DividendTracker newDividendTracker = DividendTracker(payable(newAddress));

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

    newDividendTracker.excludeFromDividends(address(newDividendTracker));
    newDividendTracker.excludeFromDividends(address(this));
    newDividendTracker.excludeFromDividends(owner());
    newDividendTracker.excludeFromDividends(address(uniswapV2Router));
    newDividendTracker.excludeFromDividends(address(0xdead));

    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.

STC - Succeeded Transfer Check

Criticality	minor
Location	contract.sol#L1683

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.

```
IERC20(_token).transfer (_account, tokenBalance);
```

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#L113,117,124,130,287,295,312,338,346,357,375,397,416,767,1058,1086,1100,1313,1341,1346,1351,1356,1364,1382,1390,1407,1414,1432,1436,1440,1798,1844

Description

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

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

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L52,53,70,856,1086,1093,1100,1110,997,1356,1666,1675,1195,1753

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
_isExcludedMaxTxn
_token
_liquidityFee
_rewardsFee
_marketingFee
magnitude
_owner
WETH
...
```

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#L787

Description

There are segments that contain unused state variables.

```
MAX_INT256
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1341,1346,1351,1356

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.

```
rewardsFee = _rewardsFee
swapTokensAtAmount = val * (10 ** 18)
maxWallet = newNum * (10 ** 18)
maxTxn = newNum * (10 ** 18)
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L1120,833

Description

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

```
abs
_transfer
```

Recommendation

Remove unused functions.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract.sol#L1585

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.

```
lastProcessedIndex
iterations
claims
```

Recommendation

The variables should be declared before any usage of them.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L1585

Description

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

```
lastProcessedIndex
claims
iterations
```

Recommendation

All the local scoped variables should be initialized.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L1019,1086,1093,1100,1110,1242

Description

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

```
totalSupply
_owner
_symbol
_name
```

Recommendation

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

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
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	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IterableMapping	Library			
	get	Public		-
	getIndexOfKey	Public		-
	getKeyAtIndex	Public		-
	size	Public		-
	set	Public	✓	-
	remove	Public	✓	-
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	✓	
DividendPayingTokenOptionalInterface	Interface			
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	External		-
DividendPayingTokenInterface	Interface			
	dividendOf	External		-
	distributeDividends	External	Payable	-

	withdrawDividend	External	✓	-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
	toUint256Safe	Internal		
SafeMathUint	Library			
	toInt256Safe	Internal		
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-

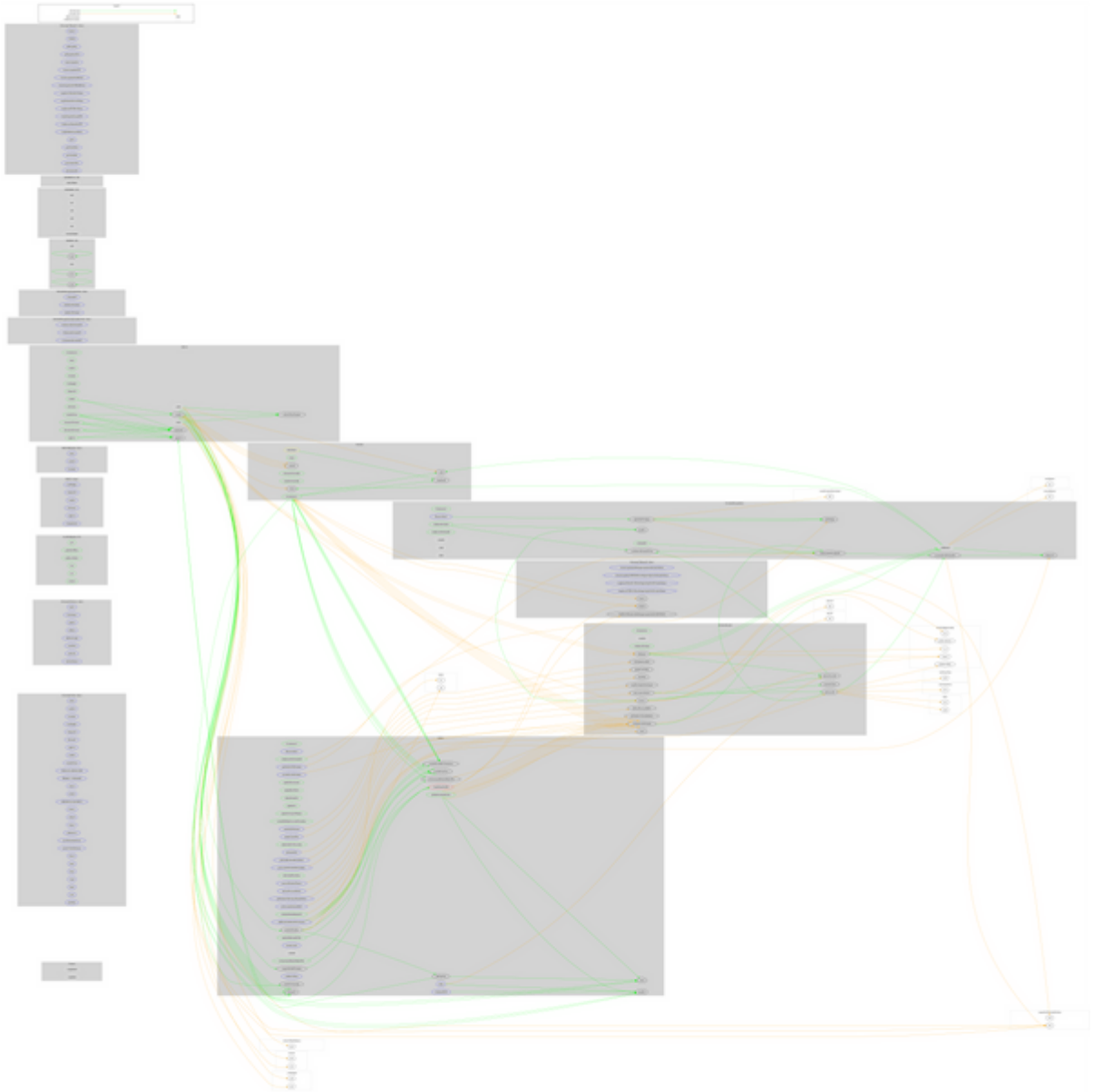
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	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	✓	-
DividendPayingToken	Implementation	ERC20, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface, Ownable		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-

	distributeDividends	Public	Payable	-
	withdrawDividend	Public	✓	-
	_withdrawDividendOfUser	Internal	✓	
	dividendOf	Public		-
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_setBalance	Internal	✓	
Fifalnu	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	addPresaleAddressForExclusions	External	✓	onlyOwner
	updateDividendTracker	Public	✓	onlyOwner
	excludeFromDividends	External	✓	onlyOwner
	includeInDividends	External	✓	onlyOwner
	updateMaxAmount	Public	✓	onlyOwner
	updateMaxWallet	Public	✓	onlyOwner
	updateSwapSize	Public	✓	onlyOwner
	updateFees	Public	✓	onlyOwner
	updateUniswapV2Router	Public	✓	onlyOwner
	excludeFromMaxTransaction	Public	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	excludeMultipleAccountsFromFees	Public	✓	onlyOwner
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	updateMarketingWallet	Public	✓	onlyOwner
	updateGasForProcessing	Public	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getClaimWait	External		-
	getTotalDividendsDistributed	External		-

	isExcludedFromFees	Public		-
	withdrawableDividendOf	Public		-
	dividendTokenBalanceOf	Public		-
	getAccountDividendsInfo	External		-
	getAccountDividendsInfoAtIndex	External		-
	processDividendTracker	External	✓	-
	claim	External	✓	-
	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	getNumberOfDividends	External		-
	removeLimits	External	✓	onlyOwner
	_transfer	Internal	✓	
	swapTokensForEth	Private	✓	
	swapAndLiquify	Private	✓	
	addLiquidity	Private	✓	
	swapAndSendDividends	Private	✓	
	withdrawBNB	External	✓	onlyOwner
	withdrawToken	External	✓	onlyOwner
DividendTracker	Implementation	DividendPayingToken		
	<Constructor>	Public	✓	DividendPayingToken
	_transfer	Internal		
	withdrawDividend	Public		-
	excludeFromDividends	External	✓	onlyOwner
	includeInDividends	External	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getLastProcessedIndex	External		-
	getNumberOfTokenHolders	External		-
	getAccount	Public		-
	getAccountAtIndex	Public		-
	canAutoClaim	Private		
	setBalance	External	✓	onlyOwner
	process	Public	✓	-

	processAccount	Public	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	fifainu.finance
Registry Domain ID	6592a3fdd18744b193e92d9be99e126c-DONUTS
Creation Date	2022-07-14T14:02:55Z
Updated Date	2022-07-16T10:57:56Z
Registry Expiry Date	2023-07-14T14:02:55Z
Registrar WHOIS Server	whois.namesilo.com
Registrar URL	http://www.namesilo.com
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

FIFA Inu 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

Cyberscope

Audit Report

FIFA Inu

July 2022

Type	BEP20
Network	BSC
Address	0x93DB06aEC78a5d605b26329CC5EFC2cA729Ba81c
Audited by	© cyberscope

Table of Contents

Table of Contents 1
Contract Review 2
Contract Analysis 3
Contract Diagnostics 4
Contract Functions 5
Contract Flow 6
Domain Info 7
Summary 8
Disclaimer 9
About Cyberscope 10

Contract Review

Contract Name	Fifalnu
Compiler Version	v0.8.15+commit.e14f2714
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x93DB06aEC78a5d605b26329CC5EFC2cA729Ba81c
Symbol	FINU
Decimals	18
Total Supply	100,000,000,000
Domain	fifainu.finance

Source Files

Filename	**SHA256**
contract.sol	a8dbc71c894509cbe6055dbc142d6723a3cc5941fc5f2a85b25aac496912ecfc

Audit Updates

Initial Audit	18th July 2022
Corrected	19th 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

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	US	Untrusted Source
	STC	Succeeded Transfer Check
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L07	Missing Events Arithmetic
	L12	Using Variables before Declaration
	L14	Uninitialized Variables in Local Scope
	L15	Local Scope Variable Shadowing

US - Untrusted Source

Criticality	minor
Location	contract.sol#L1316

Description

The contract uses an external contract in order to determine the transaction's flow. The external contract is untrusted. As a result the accumulated funds may be transferred to an untrusted source.

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

    DividendTracker newDividendTracker = DividendTracker(payable(newAddress));

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

    newDividendTracker.excludeFromDividends(address(newDividendTracker));
    newDividendTracker.excludeFromDividends(address(this));
    newDividendTracker.excludeFromDividends(owner());
    newDividendTracker.excludeFromDividends(address(uniswapV2Router));
    newDividendTracker.excludeFromDividends(address(0xdead));

    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.

STC - Succeeded Transfer Check

Criticality	minor
Location	contract.sol#L1692

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.

```
IERC20(_token).transfer (_account, tokenBalance);
```

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#L1726

Description

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

```
withdrawDividend
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L52,53,70,855,1085,1092,1099,1109,996,1365,1675,1684,1194,1760

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
_isExcludedMaxTxn
_token
_liquidityFee
_rewardsFee
_marketingFee
magnitude
_owner
WETH
...
```

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#L1365

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.

```
rewardsFee = _rewardsFee
```

Recommendation

Emit an event for critical parameter changes.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract.sol#L1594

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.

```
lastProcessedIndex
iterations
claims
```

Recommendation

The variables should be declared before any usage of them.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L1594

Description

The 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.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L1018,1085,1092,1099,1109,1244

Description

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

```
totalSupply
_owner
_symbol
_name
```

Recommendation

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

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
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	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IterableMapping	Library			
	get	External		-
	getIndexOfKey	External		-
	getKeyAtIndex	External		-
	size	External		-
	set	Public	✓	-
	remove	Public	✓	-
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	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
	increaseAllowance	External	✓	-
	decreaseAllowance	External	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
DividendPayingTokenOptionalInterface	Interface			
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	External		-
DividendPayingTokenInterface	Interface			
	dividendOf	External		-
	distributeDividends	External	Payable	-
	withdrawDividend	External	✓	-

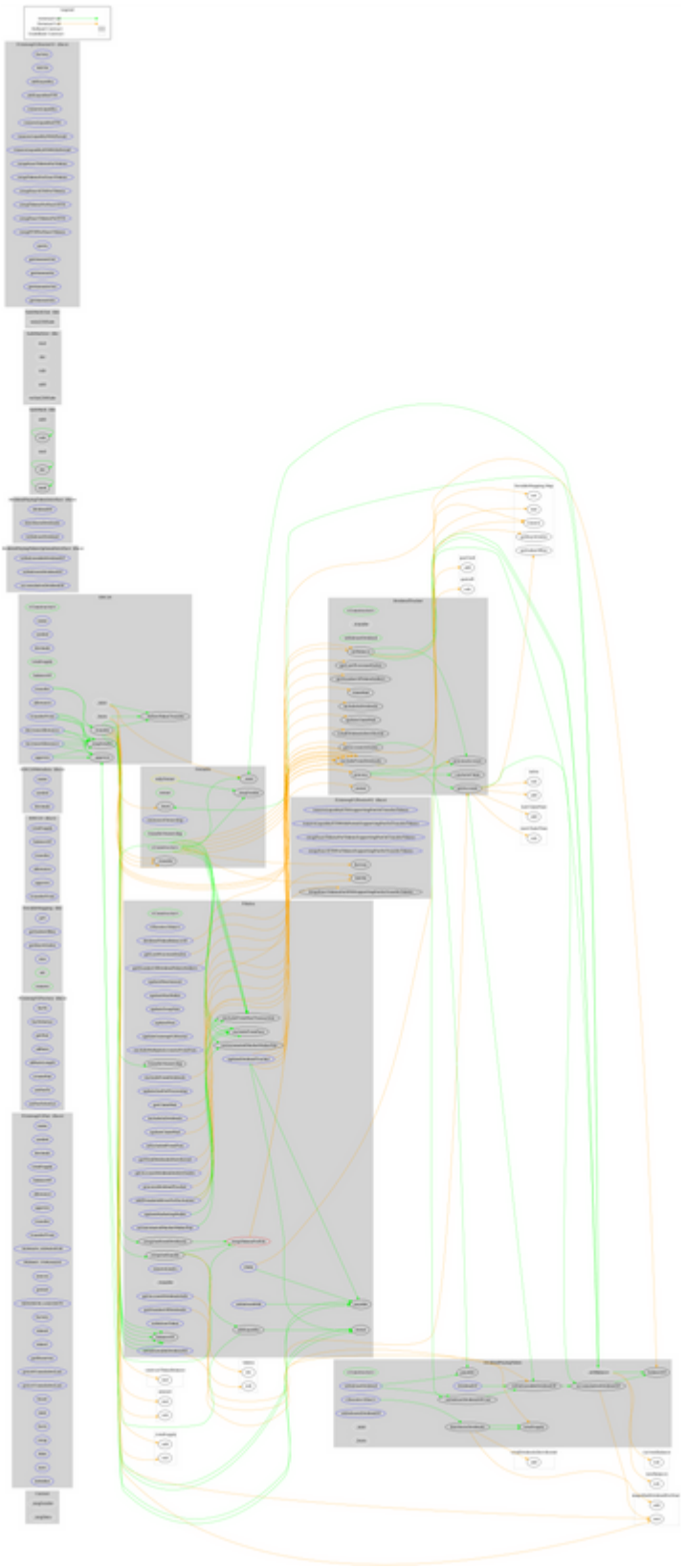
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	External	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	toUint256Safe	Internal		
SafeMathUint	Library			
	toInt256Safe	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	✓	-
DividendPayingToken	Implementation	ERC20, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface, Ownable		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	distributeDividends	Public	Payable	-
	withdrawDividend	External	✓	-
	_withdrawDividendOfUser	Internal	✓	

	dividendOf	External		-
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	Public		-
	_mint	Internal	✓	
	_burn	Internal	✓	
	_setBalance	Internal	✓	
Fifalnu	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	addPresaleAddressForExclusions	External	✓	onlyOwner
	updateDividendTracker	External	✓	onlyOwner
	excludeFromDividends	External	✓	onlyOwner
	includeInDividends	External	✓	onlyOwner
	updateMaxAmount	External	✓	onlyOwner
	updateMaxWallet	External	✓	onlyOwner
	updateSwapSize	External	✓	onlyOwner
	updateFees	External	✓	onlyOwner
	updateUniswapV2Router	External	✓	onlyOwner
	excludeFromMaxTransaction	Public	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	excludeMultipleAccountsFromFees	External	✓	onlyOwner
	setAutomatedMarketMakerPair	External	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	updateMarketingWallet	External	✓	onlyOwner
	updateGasForProcessing	External	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getClaimWait	External		-
	getTotalDividendsDistributed	External		-
	isExcludedFromFees	External		-
	withdrawableDividendOf	External		-
	dividendTokenBalanceOf	External		-
	getAccountDividendsInfo	External		-
	getAccountDividendsInfoAtIndex	External		-

	processDividendTracker	External	✓	-
	claim	External	✓	-
	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	getNumberOfDividends	External		-
	removeLimits	External	✓	onlyOwner
	_transfer	Internal	✓	
	swapTokensForEth	Private	✓	
	swapAndLiquify	Private	✓	
	addLiquidity	Private	✓	
	swapAndSendDividends	Private	✓	
	withdrawBNB	External	✓	onlyOwner
	withdrawToken	External	✓	onlyOwner
DividendTracker	Implementation	DividendPayingToken		
	<Constructor>	Public	✓	DividendPayingToken
	_transfer	Internal		
	withdrawDividend	Public		-
	excludeFromDividends	External	✓	onlyOwner
	includeInDividends	External	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getLastProcessedIndex	External		-
	getNumberOfTokenHolders	External		-
	getAccount	Public		-
	getAccountAtIndex	External		-
	canAutoClaim	Private		
	setBalance	External	✓	onlyOwner
	process	External	✓	-
	processAccount	Public	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	fifainu.finance
Registry Domain ID	6592a3fdd18744b193e92d9be99e126c-DONUTS
Creation Date	2022-07-14T14:02:55Z
Updated Date	2022-07-16T10:57:56Z
Registry Expiry Date	2023-07-14T14:02:55Z
Registrar WHOIS Server	whois.namesilo.com
Registrar URL	http://www.namesilo.com
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

FIFA Inu 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