Cyberscope

Audit Report

DogepadFinance

November 2022

Type	BEP20
Network	BSC
Address	0x182fd4f68695f1951018b5f8c1b2f778919ff0ce
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	DogepadFinance
Compiler Version	v0.8.6+commit.11564f7e
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0x182fd4f68695f1951018b5f8c1b2f778919ff0ce
Symbol	DPF
Decimals	9
Total Supply	10,000,000
Domain	dogepad.pro

Audit Updates

Initial Audit	26th November 2022
Corrected	

Source Files

Filename	SHA256
Context.sol	f24761813202348c0b68d767e1bbc488191d654b526f5e817af9af54525abc99
DividendPayingToken.sol	2992658c19de8bb98f25782099a5338e77629c9d64466dc1bbcc27b1eddade8b
DividendPayingTokenInterface.sol	d7a558ad379fbd0b0b1822aebd104e2fa99ff7c3eac13b4ee45fdd34eb29a23f
Dogepad Finance.sol	bfcc458677a6e605dbe23837c91bf8e376cb94aa2a97648bb70b35154e511c0c
ERC20.sol	f2afd1560d4b5cfa574c4ae7189da11f7d31d052e92a9d7755b8b4fbb47fadab
IDex.sol	f42c60eab527a2a8aab45407c03773c249237056e0f712c7949fd81099814e60
IERC20.sol	7c69e0bf19c4248ee1982923d6a421abcad9c5741ab499f344c64b23fd3e1001
IterableMapping.sol	cc86bca02e7cca2407c00a505d463e89ae274d5cfe3664f152ec30f9ca5b1bc6
Ownable.sol	a88be4357fa62460235dfd732182e02f1ecbf645abe05fc67f1ebcd1c23d0672
SafeMath.sol	f39d9ee58c3ad0f46c8a1886875a5de9e833d8a97f6957075e18cf7ca6a3de27

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
	PTR	Potential Transfer Revert	Unresolved
	RSML	Redundant SafeMath Library	Unresolved
	L02	State Variables could be Declared Constant	Unresolved
	L04	Conformance to Solidity Naming Conventions	Unresolved
	L05	Unused State Variable	Unresolved
	L07	Missing Events Arithmetic	Unresolved
	L11	Unnecessary Boolean equality	Unresolved
	L12	Using Variables before Declaration	Unresolved
	L13	Divide before Multiply Operation	Unresolved
	L14	Uninitialized Variables in Local Scope	Unresolved

PTR - Potential Transfer Revert

Criticality	minor / informative
Location	contract.sol#L461
Status	Unresolved

Description

The contract sends funds to a `marketingWallet` as part of the transfer flow. This address can either be a wallet address or a contract. If the address is a contract then it may revert from incoming payment. As a result, the error will propagate to the token's contract and revert the transfer.

```
function setMarketingWallet(address newWallet) external onlyOwner {
    require(newWallet != address(0),"Fee Address cannot be zero address");
    marketingWallet = newWallet;
}


payable(marketingWallet).sendValue(marketingWalletAmt);
```

Recommendation

The contract should tolerate the potential revert from the underlying contracts when the interaction is part of the main transfer flow. This could be archived by not allowing set contract addresses or by sending the funds in a non-revertable way.

RSML - Redundant SafeMath Library

Criticality	minor / informative
Location	DividendPayingToken.sol
Status	Unresolved

Description

The Solidity versions that are greater than or equal to 0.8.0 do not need the use of SafeMath Library. The usage of the SafeMath library produces unnecessary additional gas.

```
using SafeMath for uint256;
using SafeMathUint for uint256;
using SafeMathInt for int256;
```

Recommendation

The team is advised to remove the SafeMath library as it is safe to do math operations without it.

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	Dogepad Finance.sol#L37,61
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
currentRewardToken
launchtax
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	Dogepad Finance.sol#L212,207,219,32,582
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.

```
_liquidity
_enabled
_rewards
deadWallet
_marketing
_account
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.8.17/style-guide.html#naming-conventions.

L05 - Unused State Variable

Criticality	minor / informative
Location	Dogepad Finance.sol#L37
Status	Unresolved

Description

There are segments that contain unused state variables.

```
currentRewardToken
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor / informative
Location	Dogepad Finance.sol#L229,202
Status	Unresolved

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.

```
antiBotBlocks = numberOfBlocks
swapTokensAtAmount = amount * 10 ** 9
```

Recommendation

Emit an event for critical parameter changes.

L11 - Unnecessary Boolean equality

Criticality	minor / informative
Location	Dogepad Finance.sol#L544
Status	Unresolved

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.

```
value == true
```

Recommendation

Remove the equality to the boolean constant.

L12 - Using Variables before Declaration

Criticality	minor / informative
Location	Dogepad Finance.sol#L423,424,425
Status	Unresolved

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.

L13 - Divide before Multiply Operation

Criticality	minor / informative
Location	Dogepad Finance.sol#L439
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
unitBalance = deltaBalance / (denominator - sellTaxes.liquidity)
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	Dogepad Finance.sol#L397,425,424,423
Status	Unresolved

Description

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

```
swapAmt
lastProcessedIndex
claims
iterations
```

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, DividendPayingTokenInterface, Ownable		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	distributeDividends	Public	Payable	-
	_withdrawDividendOfUser	Internal	✓	
	setRewardToken	External	✓	onlyOwner
	swapBnbForCustomToken	Internal	✓	
	dividendOf	Public		-
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	_transfer	Internal	✓	
	_tokengeneration	Internal	✓	
	_burn	Internal	✓	
	_setBalance	Internal	✓	
DividendPayingTokenInterface	Interface			
	dividendOf	External		-
	distributeDividends	External	Payable	-
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-

	accumulativeDividendOf	External		-
Address	Library			
	sendValue	Internal	✓	
DogepadFinance	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	updateDividendTracker	Public	✓	onlyOwner
	processDividendTracker	External	✓	-
	claim	External	✓	-
	rescueBEP20Tokens	External	✓	onlyOwner
	forceSend	External	✓	-
	excludeFromFees	Public	✓	onlyOwner
	excludeMultipleAccountsFromFees	Public	✓	onlyOwner
	excludeFromDividends	External	✓	onlyOwner
	setMarketingWallet	External	✓	onlyOwner
	setSwapTokensAtAmount	External	✓	onlyOwner
	setBuyTaxes	External	✓	onlyOwner
	setSellTaxes	External	✓	onlyOwner
	setSwapEnabled	External	✓	onlyOwner
	enableTradingEnabled	External	✓	onlyOwner
	setAntiBotBlocks	External	✓	onlyOwner
	setMinBalanceForDividends	External	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	setGasForProcessing	External	✓	onlyOwner
	setClaimWait	External	✓	onlyOwner
	getClaimWait	External		-
	getTotalDividendsDistributed	External		-
	isExcludedFromFees	Public		-
	withdrawableDividendOf	Public		-
	getCurrentRewardToken	External		-
	dividendTokenBalanceOf	Public		-
	getAccountDividendsInfo	External		-
	getAccountDividendsInfoAtIndex	External		-

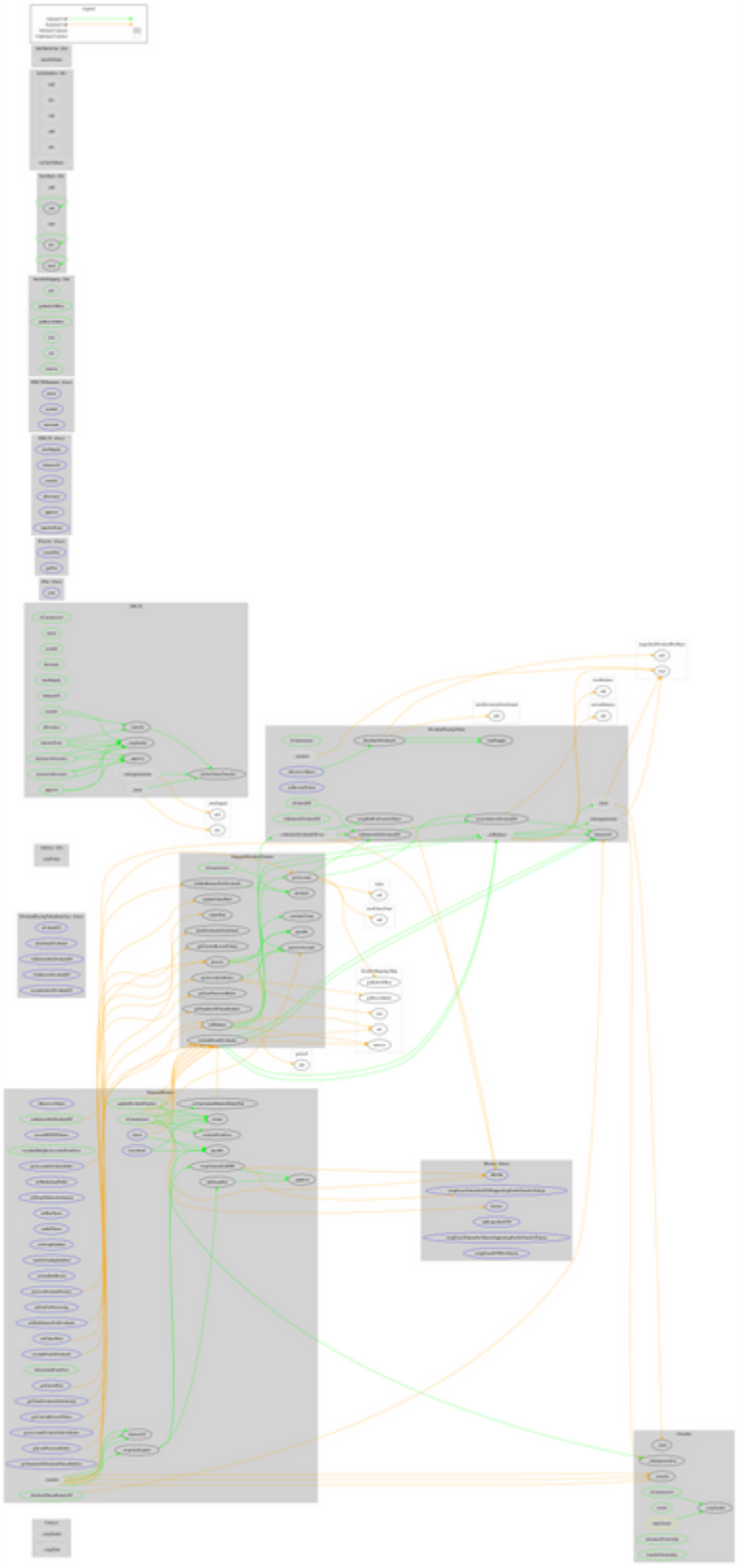
	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	_transfer	Internal	✓	
	swapAndLiquify	Private	✓	
	swapTokensForBNB	Private	✓	
	addLiquidity	Private	✓	
DogepadDividendTracker	Implementation	Ownable, DividendPayingToken		
	<Constructor>	Public	✓	DividendPayingToken
	_transfer	Internal		
	setMinBalanceForDividends	External	✓	onlyOwner
	excludeFromDividends	External	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getLastProcessedIndex	External		-
	getNumberOfTokenHolders	External		-
	getCurrentRewardToken	External		-
	getAccount	Public		-
	getAccountAtIndex	Public		-
	canAutoClaim	Private		
	setBalance	Public	✓	onlyOwner
	process	Public	✓	-
	processAccount	Public	✓	onlyOwner
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	✓	
	_tokengeneration	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
IPair	Interface			
	sync	External	✓	-
IFactory	Interface			
	createPair	External	✓	-
	getPair	External		-
IRouter	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
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	✓	-
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 Info

Domain Name	dogepad.pro
Registry Domain ID	38095d3c42ca463a921e0c7195dcc925-DONUTS
Creation Date	2022-11-21T22:03:47Z
Updated Date	2022-11-23T09:05:23Z
Registry Expiry Date	2023-11-21T22:03:47Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain was created 4 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

DogepadFinance 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 10% fees. This audit investigates security issues, business logic concerns and potential improvements.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope’s prior written consent. This report is not nor should be considered an “endorsement” or “disapproval” of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope’s position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io