Cyberscope

Audit Report

Pilot Doge

April 2022

Type	BEP20
Network	BSC
Address	0xB5626C51be7DaFC4647cAEB80dFb22ae17eD9DeE
Audited by	© cyberscope

Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 3
Contract Analysis 4
OCTD - Owner Contract Tokens Drain 5
Description 5
Recommendation 5
BC - Blacklisted Contracts 6
Description 6
Recommendation 6
Contract Diagnostics 7
L01 - Public Function could be Declared External 8
Description 8
Recommendation 8
L02 - State Variables could be Declared Constant 9
Description 9
Recommendation 9
L04 - Conformance to Solidity Naming Conventions 10
Description 10
Recommendation 10
L05 - Unused State Variable 11
Description 11
Recommendation 11
L14 - Uninitialized Variables in Local Scope 12
Description 12

Recommendation 12
Contract Functions 13
Contract Flow 16
Summary 17
Disclaimer 18
About Cyberscope 19

Contract Review

Contract Name	PILOTDOGE
Compiler Version	v0.8.7+commit.e28d00a7
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0xB5626C51be7DaFC4647cAEB80dFb22ae17eD9DeE
Symbol	$PDoge
Decimals	9
Total Supply	100,000,000,000
Domain	

Source Files

Filename	**SHA256**
contract.sol	d0aab842def132225e22da812473ea55e01a7762d1ca8c5bf06b17dad7e6909a

Audit Updates

Initial Audit	24rd April 2022
Corrected	27th April 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

OCTD - Owner Contract Tokens Drain

Criticality	minor
Location	contract.sol#L390,396

Description

The contract owner has the authority to claim all the balance of the contract. The owner may take advantage of it by calling the `manualswap` or `manualsend` functions.

```
    function manualswap() external {
        require(_msgSender() == _developmentAddress || _msgSender() ==
_marketingAddress || _msgSender() == owner());
        uint256 contractBalance = balanceOf(address(this));
        swapTokensForEth(contractBalance);
    }

    function manualsend() external {
        require(_msgSender() == _developmentAddress || _msgSender() ==
_marketingAddress || _msgSender() == owner());
        uint256 contractETHBalance = address(this).balance;
        sendETHToFee(contractETHBalance);
    }
```

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.

BC - Blacklisted Contracts

Criticality	critical
Location	contract.sol#L240

Description

The contract owner has the authority to stop contracts from transactions. The owner may take advantage of it by calling the `manage_blacklist` function.

```
if(blacklistMode){
          require(!isBlacklisted[from] && !isBlacklisted[to],"Blacklisted");
        }
```

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
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L05	Unused State Variable
	L14	Uninitialized Variables in Local Scope

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L110,116,187,191,195,199,207,212,216,221,240,246,316,322,329,403,414,418

Description

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

```
excludeMultipleAccountsFromFees
toggleSwap
setFee
setNewMarketingAddress
setNewDevAddress
rescueForeignTokens
enable_blacklist
manage_blacklist
transferFrom
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L93

Description

Constant state variables should be declared constant to save gas.

```
_previousOwner
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L34,123,315,321,328,240,246,316,414,135,148,149,150

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.

```
_decimals
_symbol
_name
_tTotal
_swapEnabled
_amount
_to
_tokenAddr
_status
...
```

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#L93,127

Description

There are segments that contain unused state variables.

```
_tOwned
_previousOwner
```

Recommendation

Remove unused state variables.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L241

Description

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

```
i
```

Recommendation

All the local scoped variables should be initialized.

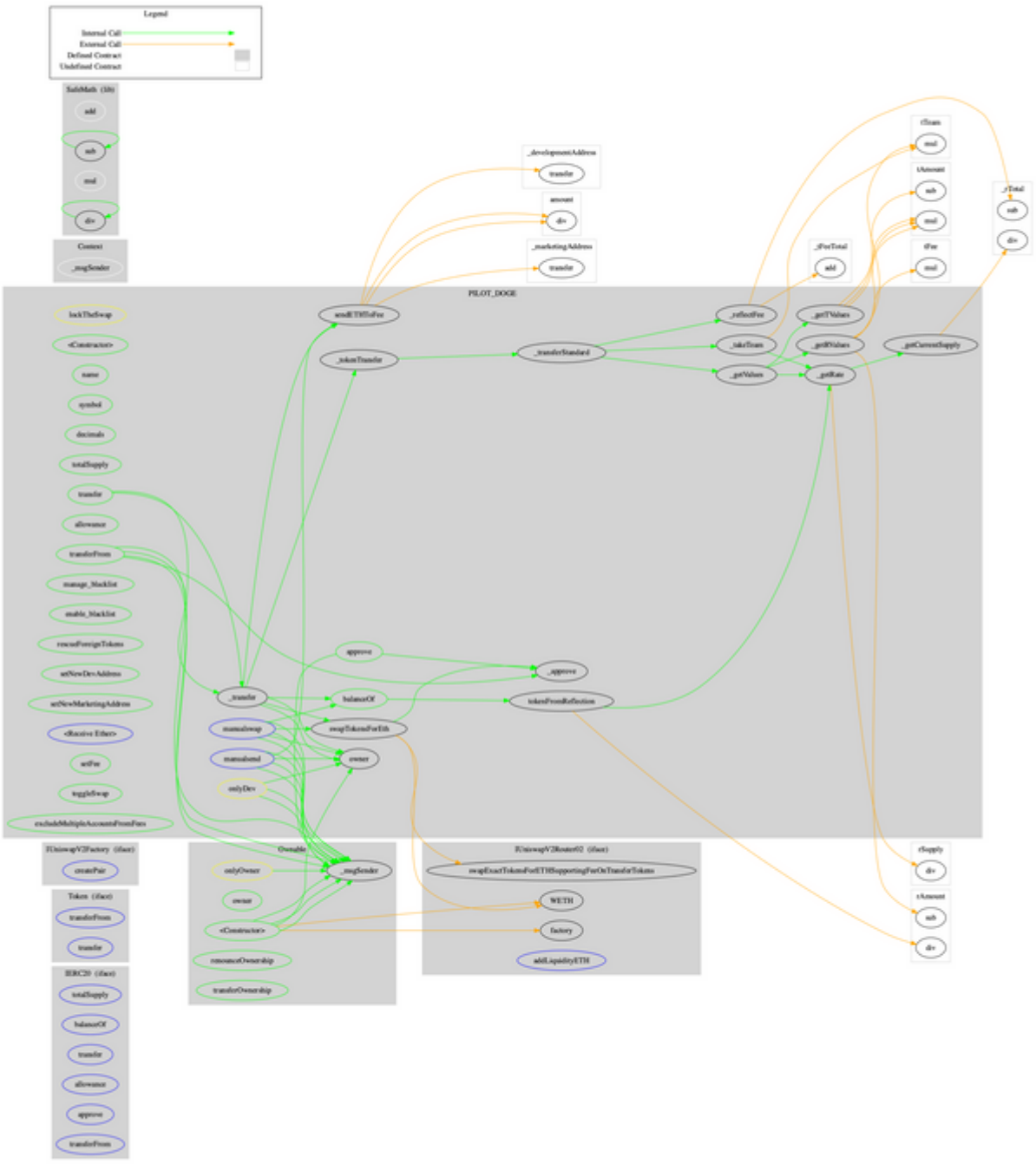
Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Token	Interface			
	transferFrom	External	✓	-
	transfer	External	✓	-
IUniswapV2Factory	Interface			
	createPair	External	✓	-
IUniswapV2Router02	Interface			
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
Context	Implementation			
	_msgSender	Internal		
SafeMath	Library			
	add	Internal		
	sub	Internal		

	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
PILOT_DOGE	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	tokenFromReflection	Private		
	_approve	Private	✓	
	manage_blacklist	Public	✓	onlyDev
	enable_blacklist	Public	✓	onlyDev
	_transfer	Private	✓	
	swapTokensForEth	Private	✓	lockTheSwap
	sendETHToFee	Private	✓	
	_tokenTransfer	Private	✓	
	rescueForeignTokens	Public	✓	onlyDev
	setNewDevAddress	Public	✓	onlyDev
	setNewMarketingAddress	Public	✓	onlyDev
	_transferStandard	Private	✓	

	_takeTeam	Private	✓	
	_reflectFee	Private	✓	
	<Receive Ether>	External	Payable	-
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	manualswap	External	✓	-
	manualsend	External	✓	-
	setFee	Public	✓	onlyDev
	toggleSwap	Public	✓	onlyDev
	excludeMultipleAccountsFromFees	Public	✓	onlyOwner

Contract Flow



Summary

Pilot Doge Token is an interesting project that has a friendly and growing community. There are some functions that can be abused by the owner, like massively blacklisting wallets from transactions, and transferring the contract's accumulated fees to the team wallet. The maximum fee percentage that can be set is 20%. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

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

Cyberscope

Audit Report

Pilot Doge

April 2022

Type	BEP20
Network	BSC
Address	0x113E87026F6d70ae74C0f9189bD8b45701dB8A9d
Audited by	© cyberscope

Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 3
Contract Analysis 4
OCTD - Owner Contract Tokens Drain 5
Description 5
Recommendation 5
BC - Blacklisted Contracts 6
Description 6
Recommendation 6
Contract Diagnostics 7
L01 - Public Function could be Declared External 8
Description 8
Recommendation 8
L02 - State Variables could be Declared Constant 9
Description 9
Recommendation 9
L04 - Conformance to Solidity Naming Conventions 10
Description 10
Recommendation 10
L05 - Unused State Variable 11
Description 11
Recommendation 11
L14 - Uninitialized Variables in Local Scope 12
Description 12

Recommendation 12
Contract Functions 13
Contract Flow 16
Summary 17
Disclaimer 18
About Cyberscope 19

Contract Review

Contract Name	PILOT_DOGE
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0x113e87026f6d70ae74c0f9189bd8b45701db8a9d
Symbol	$PDoge
Decimals	9
Total Supply	100,000,000,000
Domain	

Source Files

Filename	**SHA256**
contract.sol	eb628a48451e5dd85d6dcdde78bcdd279590732944806eb3ea678583c54922b6

Audit Updates

Initial Audit	23rd April 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

OCTD - Owner Contract Tokens Drain

Criticality	minor
Location	contract.sol#L390,396

Description

The contract owner has the authority to claim all the balance of the contract. The owner may take advantage of it by calling the `manualswap` or `manualsend` functions.

```
    function manualswap() external {
        require(_msgSender() == _developmentAddress || _msgSender() ==
_marketingAddress || _msgSender() == owner());
        uint256 contractBalance = balanceOf(address(this));
        swapTokensForEth(contractBalance);
    }

    function manualsend() external {
        require(_msgSender() == _developmentAddress || _msgSender() ==
_marketingAddress || _msgSender() == owner());
        uint256 contractETHBalance = address(this).balance;
        sendETHToFee(contractETHBalance);
    }
```

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.

BC - Blacklisted Contracts

Criticality	critical
Location	contract.sol#L240

Description

The contract owner has the authority to stop contracts from transactions. The owner may take advantage of it by calling the `manage_blacklist` function.

```
if(blacklistMode){
          require(!isBlacklisted[from] && !isBlacklisted[to],"Blacklisted");
        }
```

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
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L05	Unused State Variable
	L14	Uninitialized Variables in Local Scope

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L110,116,187,191,195,199,207,212,216,221,240,246,316,322,329,403,414,418

Description

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

```
excludeMultipleAccountsFromFees
toggleSwap
setFee
setNewMarketingAddress
setNewDevAddress
rescueForeignTokens
enable_blacklist
manage_blacklist
transferFrom
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L93

Description

Constant state variables should be declared constant to save gas.

```
_previousOwner
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L34,123,315,321,328,240,246,316,414,135,148,149,150

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.

```
_decimals
_symbol
_name
_tTotal
_swapEnabled
_amount
_to
_tokenAddr
_status
...
```

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#L93,127

Description

There are segments that contain unused state variables.

```
_tOwned
_previousOwner
```

Recommendation

Remove unused state variables.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L241

Description

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

```
i
```

Recommendation

All the local scoped variables should be initialized.

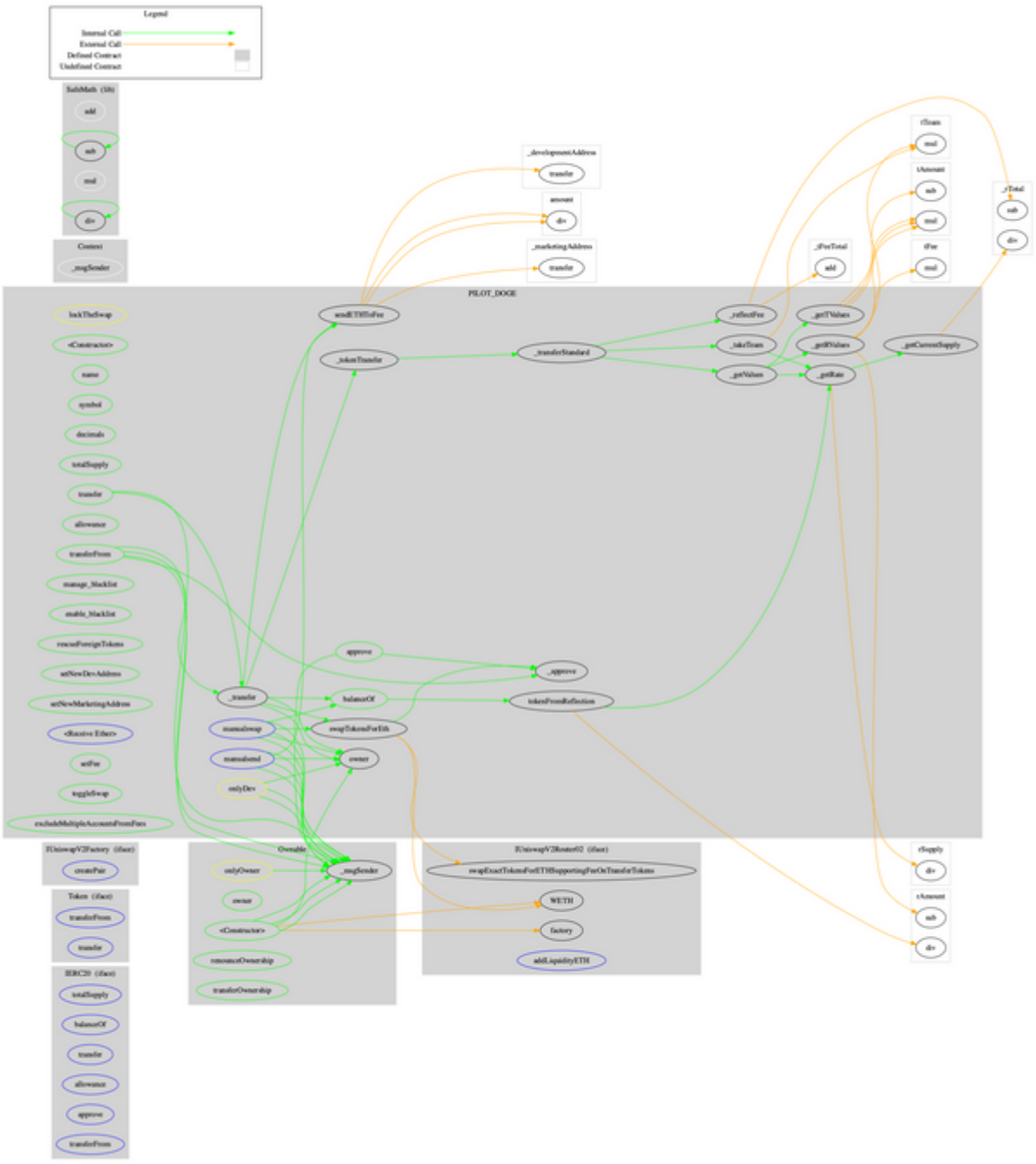
Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Token	Interface			
	transferFrom	External	✓	-
	transfer	External	✓	-
IUniswapV2Factory	Interface			
	createPair	External	✓	-
IUniswapV2Router02	Interface			
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
Context	Implementation			
	_msgSender	Internal		
SafeMath	Library			
	add	Internal		
	sub	Internal		

	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
PILOT_DOGE	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	tokenFromReflection	Private		
	_approve	Private	✓	
	manage_blacklist	Public	✓	onlyDev
	enable_blacklist	Public	✓	onlyDev
	_transfer	Private	✓	
	swapTokensForEth	Private	✓	lockTheSwap
	sendETHToFee	Private	✓	
	_tokenTransfer	Private	✓	
	rescueForeignTokens	Public	✓	onlyDev
	setNewDevAddress	Public	✓	onlyDev
	setNewMarketingAddress	Public	✓	onlyDev
	_transferStandard	Private	✓	

	_takeTeam	Private	✓	
	_reflectFee	Private	✓	
	<Receive Ether>	External	Payable	-
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	manualswap	External	✓	-
	manualsend	External	✓	-
	setFee	Public	✓	onlyDev
	toggleSwap	Public	✓	onlyDev
	excludeMultipleAccountsFromFees	Public	✓	onlyOwner

Contract Flow



Summary

Pilot Doge Token is an interesting project that has a friendly and growing community. There are some functions that can be abused by the owner, like massively blacklisting wallets from transactions, and transferring the contract's accumulated fees to the team wallet. The maximum fee percentage that can be set is 20%. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

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

Cyberscope

Audit Report

Pilot Doge

April 2022

Type	BEP20
Network	BSC
Address	0xb5626c51be7dafc4647caeb80dfb22ae17ed9dee
Audited by	© cyberscope

Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 3
Contract Analysis 4
OCTD - Owner Contract Tokens Drain 5
Description 5
Recommendation 5
BC - Blacklisted Contracts 6
Description 6
Recommendation 6
Contract Diagnostics 7
L01 - Public Function could be Declared External 8
Description 8
Recommendation 8
L02 - State Variables could be Declared Constant 9
Description 9
Recommendation 9
L04 - Conformance to Solidity Naming Conventions 10
Description 10
Recommendation 10
L05 - Unused State Variable 11
Description 11
Recommendation 11
L14 - Uninitialized Variables in Local Scope 12
Description 12

Recommendation 12
Contract Functions 13
Contract Flow 16
Summary 17
Disclaimer 18
About Cyberscope 19

Contract Review

Contract Name	PILOTDOGE
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0xb5626c51be7dafc4647caeb80dfb22ae17ed9dee
Symbol	$PDoge
Decimals	9
Total Supply	100,000,000,000
Domain	

Source Files

Filename	**SHA256**
contract.sol	eb628a48451e5dd85d6dcdde78bcdd279590732944806eb3ea678583c54922b6

Audit Updates

Initial Audit	24rd April 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

OCTD - Owner Contract Tokens Drain

Criticality	minor
Location	contract.sol#L390,396

Description

The contract owner has the authority to claim all the balance of the contract. The owner may take advantage of it by calling the `manualswap` or `manualsend` functions.

```
    function manualswap() external {
        require(_msgSender() == _developmentAddress || _msgSender() ==
_marketingAddress || _msgSender() == owner());
        uint256 contractBalance = balanceOf(address(this));
        swapTokensForEth(contractBalance);
    }

    function manualsend() external {
        require(_msgSender() == _developmentAddress || _msgSender() ==
_marketingAddress || _msgSender() == owner());
        uint256 contractETHBalance = address(this).balance;
        sendETHToFee(contractETHBalance);
    }
```

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.

BC - Blacklisted Contracts

Criticality	critical
Location	contract.sol#L240

Description

The contract owner has the authority to stop contracts from transactions. The owner may take advantage of it by calling the `manage_blacklist` function.

```
if(blacklistMode){
          require(!isBlacklisted[from] && !isBlacklisted[to],"Blacklisted");
        }
```

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
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L05	Unused State Variable
	L14	Uninitialized Variables in Local Scope

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L110,116,187,191,195,199,207,212,216,221,240,246,316,322,329,403,414,418

Description

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

```
excludeMultipleAccountsFromFees
toggleSwap
setFee
setNewMarketingAddress
setNewDevAddress
rescueForeignTokens
enable_blacklist
manage_blacklist
transferFrom
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L93

Description

Constant state variables should be declared constant to save gas.

```
_previousOwner
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L34,123,315,321,328,240,246,316,414,135,148,149,150

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.

```
_decimals
_symbol
_name
_tTotal
_swapEnabled
_amount
_to
_tokenAddr
_status
...
```

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#L93,127

Description

There are segments that contain unused state variables.

```
_tOwned
_previousOwner
```

Recommendation

Remove unused state variables.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L241

Description

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

```
i
```

Recommendation

All the local scoped variables should be initialized.

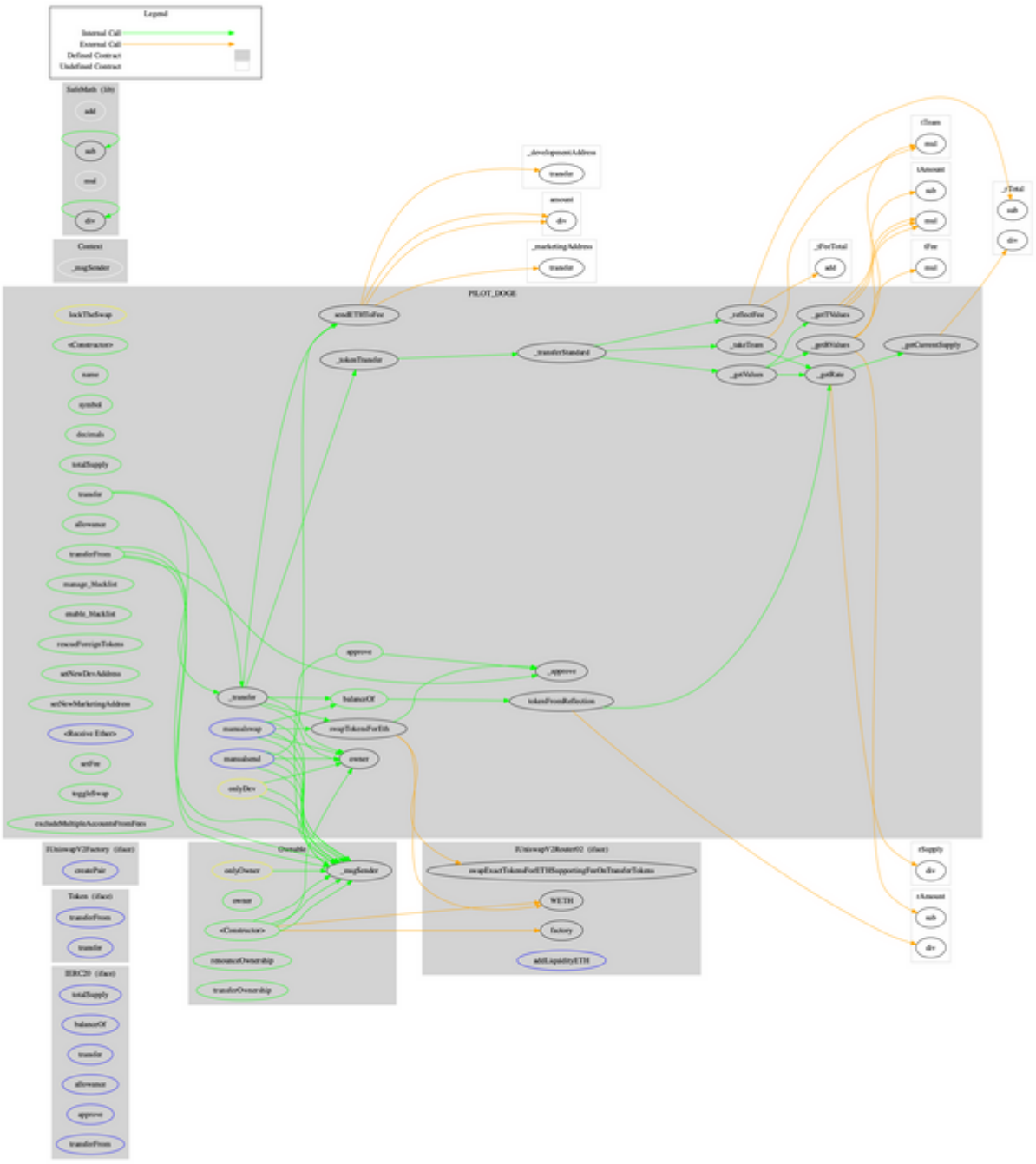
Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Token	Interface			
	transferFrom	External	✓	-
	transfer	External	✓	-
IUniswapV2Factory	Interface			
	createPair	External	✓	-
IUniswapV2Router02	Interface			
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
Context	Implementation			
	_msgSender	Internal		
SafeMath	Library			
	add	Internal		
	sub	Internal		

	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
PILOT_DOGE	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	tokenFromReflection	Private		
	_approve	Private	✓	
	manage_blacklist	Public	✓	onlyDev
	enable_blacklist	Public	✓	onlyDev
	_transfer	Private	✓	
	swapTokensForEth	Private	✓	lockTheSwap
	sendETHToFee	Private	✓	
	_tokenTransfer	Private	✓	
	rescueForeignTokens	Public	✓	onlyDev
	setNewDevAddress	Public	✓	onlyDev
	setNewMarketingAddress	Public	✓	onlyDev
	_transferStandard	Private	✓	

	_takeTeam	Private	✓	
	_reflectFee	Private	✓	
	<Receive Ether>	External	Payable	-
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	manualswap	External	✓	-
	manualsend	External	✓	-
	setFee	Public	✓	onlyDev
	toggleSwap	Public	✓	onlyDev
	excludeMultipleAccountsFromFees	Public	✓	onlyOwner

Contract Flow



Summary

Pilot Doge Token is an interesting project that has a friendly and growing community. There are some functions that can be abused by the owner, like massively blacklisting wallets from transactions, and transferring the contract's accumulated fees to the team wallet. The maximum fee percentage that can be set is 20%. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

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