Cyberscope

Audit Report

Paws and Claws

March 2022

Type	ERC20
Network	ETH
Address	0xB1d55f362b9c68e7Ed431F3ad2Ff178102dE0201
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	CoinToken
Compiler Version	
Optimization	200 runs
Licence	
Explorer	https://etherscan.io/token/0xb1d55f362b9c68e7ed431f3ad2ff178102de0201
Symbol	PAWS
Decimals	18
Total Supply	1,000,000,000
Source	contract.sol
Domain	pawsandclaws.io

Audit Updates

Initial Audit	9th March 2022
Corrected	10th March 2022

Contract Analysis

Critical Medium Minor Pass

Severity	Code	Description	Resolved
●	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	✓

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L1088

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the `setSellTax` or the `setBuyTax` function with a high percentage value. If the setSellTax total amount is more than 100, then the contract will operate as a honeypot.

```
function setSellTax(uint256 dev, uint256 marketing, uint256 liquidity, uint256
charity) public onlyOwner {

    sellTaxes["dev"] = dev;
    sellTaxes["marketing"] = marketing;
    sellTaxes["liquidity"] = liquidity;
    sellTaxes["charity"] = charity;
}
```

Recommendation

The contract could embody a check for the maximum acceptable value.

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.

Update (10th March)

The team has renounced the ownership of the contract. Hence contract cannot disturb the user's transactions.

BC - Blacklisted Contracts

Criticality	medium
Location	contract.sol#L1002

Description

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

```
require(!isBlacklisted(msg.sender), "CoinToken: sender blacklisted");
require(!isBlacklisted(recipient), "CoinToken: recipient blacklisted");
require(!isBlacklisted(tx.origin), "CoinToken: sender 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.

Update (10th March)

The team has renounced the ownership of the contract. Hence the contract cannot disturb the user's transactions.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	FSA	Fixed Swap Address
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L05	Unused State Variable
	L04	Conformance to Solidity Naming Conventions
	L14	Uninitialized Variables in Local Scope
	L13	Divide before Multiply Operation

FSA - Fixed Swap Address

Criticality	minor
Location	contract.sol#L869

Description

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

```
uniswapV2Router02 = IUniswapV2Router02(_addr[1]);
uniswapV2Factory = IUniswapV2Factory(uniswapV2Router02.factory());
uniswapV2Pair = IUniswapV2Pair(uniswapV2Factory.createPair(address(this),
uniswapV2Router02.WETH()));
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L177,185,202,209,216,228,236,247,265,293 and 12 more

Description

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

```
disableTax
enableTax
removeExclude
disableBlacklist
enableBlacklist
burn
unpause
pause
triggerTax
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L838,843,848,831,835,840,845,837,842,847 and 4 more

Description

Constant state variables should be declared constant to save gas.

```
swapThreshold
marketingTaxWallet
marketingTaxSell
marketingTaxBuy
liquidityTaxWallet
liquidityTaxSell
liquidityTaxBuy
devTaxWallet
devTaxSell
...
```

Recommendation

Add the constant attribute to state variables that never change.

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L835,836,837,838,840,841,842,843,845,846 and 2 more

Description

There are segments that contain unused state variables.

```
charityTaxWallet
liquidityTaxWallet
marketingTaxWallet
devTaxWallet
charityTaxSell
liquidityTaxSell
marketingTaxSell
devTaxSell
charityTaxBuy
...
```

Recommendation

Remove unused state variables.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L638,639,656,692

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.

```
WETH
MINIMUM_LIQUIDITY
PERMIT_TYPEHASH
DOMAIN_SEPARATOR
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L121

Description

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

```
_msgData
```

Recommendation

Remove unused functions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L896

Description

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

```
tax
```

Recommendation

All the local scoped variables should be initialized.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L890 and 10 more

Description

Performing divisions before multiplications may cause lose of prediction.

```
charityETH = (ethGained * ((charityTokens * 10 ** 18) / taxSum)) / 10 ** 18
devETH = (ethGained * ((devTokens * 10 ** 18) / taxSum)) / 10 ** 18
marketingETH = (ethGained * ((marketingTokens * 10 ** 18) / taxSum)) / 10 ** 18
liquidityETH = (ethGained * ((liquidityTokens / 2 * 10 ** 18) / taxSum)) / 10 **
18
baseUnit = amount / denominator
...
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-

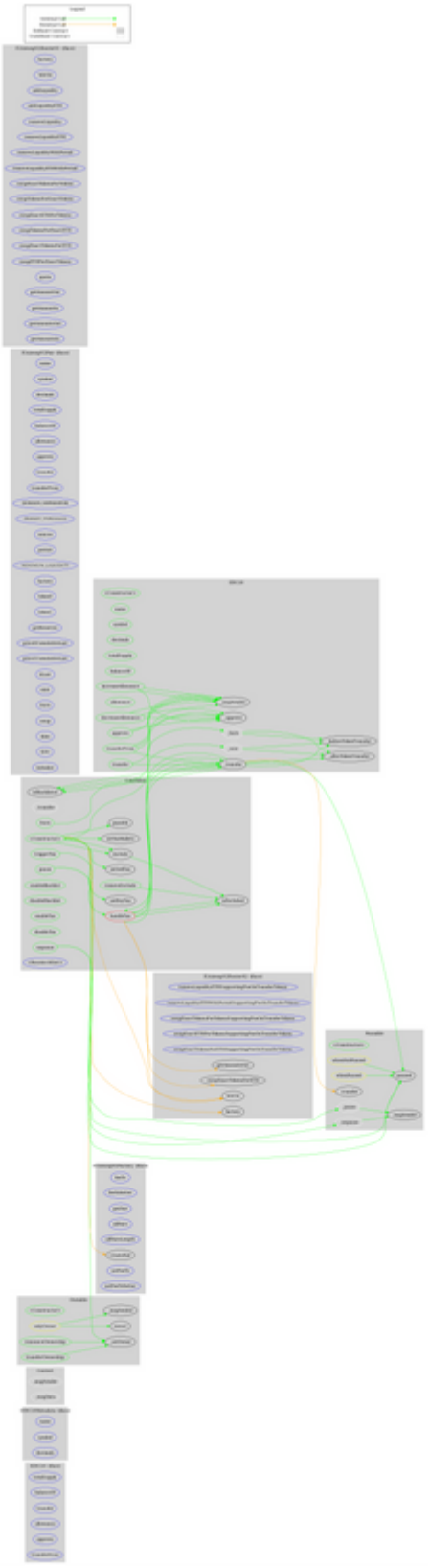
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Internal	✓	
Pausable	Implementation	Context		
	<Constructor>	Public	✓	-
	paused	Public		-
	_pause	Internal	✓	whenNotPaused
	_unpause	Internal	✓	whenPaused
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	✓	-
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	✓	-
CoinToken	Implementation	ERC20, Ownable, Pausable		
	<Constructor>	Public	Payable	ERC20
	handleTax	Private	✓	
	_transfer	Internal	✓	
	triggerTax	Public	✓	onlyOwner
	pause	Public	✓	onlyOwner
	unpause	Public	✓	onlyOwner

	burn	Public	✓	onlyOwner
	enableBlacklist	Public	✓	onlyOwner
	disableBlacklist	Public	✓	onlyOwner
	exclude	Public	✓	onlyOwner
	removeExclude	Public	✓	onlyOwner
	setBuyTax	Public	✓	onlyOwner
	setSellTax	Public	✓	onlyOwner
	setTaxWallets	Public	✓	onlyOwner
	enableTax	Public	✓	onlyOwner
	disableTax	Public	✓	onlyOwner
	isBlacklisted	Public		-
	isExcluded	Public		-
	<Receive Ether>	External	Payable	-

Contract Flow



Domain Info

Domain Name	pawsandclaws.io
Registry Domain ID	3a9d70ee145e4858a9491e599b2a78f5-DONUTS
Creation Date	2022-03-09T08:02:01Z
Updated Date	2022-03-09T08:02:02Z
Registry Expiry Date	2023-03-09T08:02:01Z
Registrar WHOIS Server	whois.godaddy.com/
Registrar URL	http://www.godaddy.com/domains/search.aspx?ci=8990
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain has been created about 7 hours 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

There are some functions that can be abused by the owner, like manipulating fees and blacklisting contracts. The contract can be converted into a honeypot and prevent users from selling if the owner abuses the sell tax function. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

Update (10th March)

The team has renounced the ownership of the contract. Hence the contract cannot disturb the user’s transactions.

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