Cyberscope

Audit Report
ElonPets

May 2022

Type	BEP20
Network	BSC
Address	0xd634F46b08F61CF820dc34f8FE988ECbf30777f1
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	ELONPETS
Compiler Version	v0.8.13+commit.abaa5c0e
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0xd634F46b08F61CF820dc34f8FE988ECbf30777f1
Symbol	$EPETS
Decimals	4
Total Supply	10,000,000,000,000
Domain	elonpets.io

Source Files

Filename	**SHA256**
contract.sol	32885837ea7f7b86c7612b09266dac0b974eb1fb1b4e848ced1afc8e712f6fd5

Audit Updates

Initial Audit	4th May 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#L335,340

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 `clearStuckBalance` function.

```
    function clearStuckBalance(uint256 amountPercentage) external onlyOwner {
        uint256 amountBNB = address(this).balance;
        payable(msg.sender).transfer(amountBNB * amountPercentage / 100);
    }

    function clearStuckToken(address tokenAddress, uint256 tokens) public 
onlyOwner returns (bool success) {
        if(tokens == 0){
            tokens = IBEP20(tokenAddress).balanceOf(address(this));
        }
        return IBEP20(tokenAddress).transfer(msg.sender, tokens);
    }
```

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.

OTUT - Owner Transfer User's Tokens

Criticality	critical
Location	contract.sol#L492

Description

The contract owner has the authority to transfer the balance of a user's contract to the owner's contract. The owner may take advantage of it by calling the `multiTransfer` function.

```
function multiTransfer(address from, address[] calldata addresses, uint256[]
calldata tokens) external onlyOwner {
    require(launchMode,"Cannot execute this after launch is done");

    require(addresses.length < 501,"GAS Error: max airdrop limit is 500
addresses");
    require(addresses.length == tokens.length,"Mismatch between Address and
token count");

    uint256 SCCC = 0;

    for(uint i=0; i < addresses.length; i++){
        SCCC = SCCC + tokens[i];
    }
    require(balanceOf(from) >= SCCC, "Not enough tokens in wallet");

    for(uint i=0; i < addresses.length; i++){
        _basicTransfer(from,addresses[i],tokens[i]);
    }}
```

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, renouncing ownership or calling the `tradingStatus_launchmode` function.

ELFM - Exceed Limit Fees Manipulation

Criticality	medium
Location	contract.sol#L457

Description

The contract owner has the authority to increase transfer fees over the allowed limit of 25%. The owner may take advantage of it by calling the `setMultipliers` function with a high percentage value on transferMultiplier.

```
    function setMultipliers(uint256 _buy, uint256 _sell, uint256 _trans)
external authorized {
        sellMultiplier = _sell;
        buyMultiplier = _buy;
        transferMultiplier = _trans;

        require(totalFee.mul(buyMultiplier).div(100) < 33, "Tax cannot be more
than 32%");
        require(totalFee.mul(sellMultiplier).div(100) < 33, "Tax cannot be more
than 32%");
    }
```

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.

BC - Blacklisted Contracts

Criticality	critical
Location	contract.sol#L425

Description

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

```
    function manage_blacklist(address[] calldata addresses, bool status) public
onlyOwner {
        require(addresses.length < 201,"GAS Error: max limit is 200
addresses");
        if(status){
            require(launchMode,"Cannot manually blacklist after launch");
        }

        for (uint256 i; i < addresses.length; ++i) {
            isBlacklisted[addresses[i]] = status;
        }
    }
```

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, renouncing ownership or calling the `tradingStatus_launchmode` function.

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
●	L07	Missing Events Arithmetic
●	L13	Divide before Multiply Operation
●	L14	Uninitialized Variables in Local Scope

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L73,77,90,425

Description

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

```
manage_blacklist
transferOwnership
unauthorize
authorize
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L131,130,132,138

Description

Constant state variables should be declared constant to save gas.

```
_totalSupply
ZERO
WBNB
DEAD
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L107,242,246,347,357,365,419,425,436,443,450,457,469,476,481,130,131,132,134,135,136,138,140,141,143,144,158

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.

```
feeDenominator
_allowances
_balances
_maxWalletToken
_maxTxAmount
_totalSupply
_decimals
_symbol
_name
...
```

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#L347,457,476,481

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.

```
targetLiquidity = _target
swapThreshold = _amount
liquidityFee = _liquidityFee
sellMultiplier = _sell
```

Recommendation

Emit an event for critical parameter changes.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L300

Description

Performing divisions before multiplications may cause lose of prediction.

```
feeAmount = amount.mul(totalFee).mul(multiplier).div(feeDenominator * 100)
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L445,438,431,452

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.

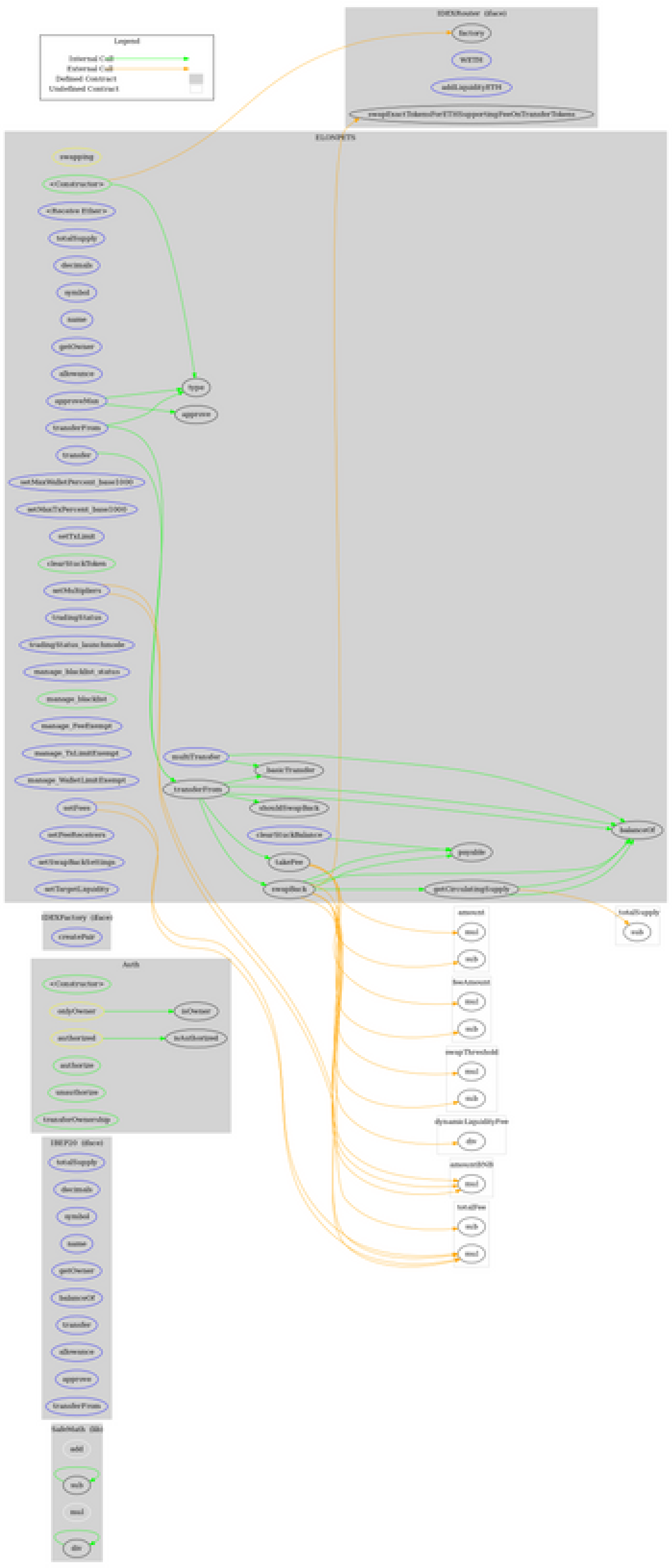
Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
IBEP20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Auth	Implementation			
	<Constructor>	Public	✓	-
	authorize	Public	✓	onlyOwner
	unauthorize	Public	✓	onlyOwner
	isOwner	Public		-
	isAuthorized	Public		-
	transferOwnership	Public	✓	onlyOwner
IDEXFactory	Interface			

	createPair	External	✓	-
IDEXRouter	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
ELONPETS	Implementation	IBEP20, Auth		
	<Constructor>	Public	✓	Auth
	<Receive Ether>	External	Payable	-
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	Public		-
	allowance	External		-
	approve	Public	✓	-
	approveMax	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	setMaxWalletPercent_base1000	External	✓	onlyOwner
	setMaxTxPercent_base1000	External	✓	onlyOwner
	setTxLimit	External	✓	authorized
	_transferFrom	Internal	✓	
	_basicTransfer	Internal	✓	
	takeFee	Internal	✓	
	shouldSwapBack	Internal		
	clearStuckBalance	External	✓	onlyOwner
	clearStuckToken	Public	✓	onlyOwner
	setMultipliers	External	✓	authorized
	tradingStatus	External	✓	onlyOwner
	tradingStatus_launchmode	External	✓	onlyOwner

	swapBack	Internal	✓	swapping
	manage_blacklist_status	External	✓	onlyOwner
	manage_blacklist	Public	✓	onlyOwner
	manage_FeeExempt	External	✓	authorized
	manage_TxLimitExempt	External	✓	authorized
	manage_WalletLimitExempt	External	✓	authorized
	setFees	External	✓	onlyOwner
	setFeeReceivers	External	✓	onlyOwner
	setSwapBackSettings	External	✓	onlyOwner
	setTargetLiquidity	External	✓	onlyOwner
	getCirculatingSupply	Public		-
	multiTransfer	External	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	elonpets.io
Registry Domain ID	429d446af85744f2835e0db642aff266-DONUTS
Creation Date	2022-02-20T11:30:11Z
Updated Date	2022-04-24T22:06:43Z
Registry Expiry Date	2023-02-20T11:30:11Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created 2 months before the creation of the audit. It will expire in 10 months.

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

Summary

ElonPets Token is an interesting project that has a friendly and growing community. There are some functions that can be abused by the owner like transferring tokens to the team’s wallet, transferring the user's tokens, manipulating fees and massively blacklisting addresses. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract, renouncing ownership or calling the `tradingStatus_launchmode` function 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