Cyberscope

Audit Report

DirtiCoin

October 2022

Address	0x10e449fb87cde6fd6b82a3cdd4bae283c2f34729
Network	ETH
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	DIDToken
Compiler Version	v0.8.6+commit.11564f7e
Optimization	200 runs
Token	https://etherscan.io/address/0x10e449fb87cde6fd6b82a3cdd4bae283c2f34729
Proxy Contract	https://etherscan.io/token/0x6a11ac79a3968a4cec0b0aba8cca3bb71ff4e27c
Symbol	DID
Decimals	18
Total Supply	15000000

Audit Updates

Initial Audit	2nd September 2022 https://github.com/cyberscope-io/audits/blob/main/dirticoin/v1/audit.pdf
Corrected phase 1	2nd October 2022 https://github.com/cyberscope-io/audits/blob/main/dirticoin/v2/audit.pdf
Corrected phase 2	7th October 2022

Source Files

Filename	SHA256
@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol	da66c17044345dc892d85bd7ddc9745d25df0b3dacfba8f84eb87c60d6e40fe3
@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol	cd823c76cbf5f5b6ef1bda565d58be66c843c37707cd93eb8fb5425deebd6756
@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol	36a6477c6263d9441dab59861e0ca97a201caf2843598af2a8e04e897a738c2f
@openzeppelin/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol	b97515a88e75c313eacf0a27c9439ef371d86d4c2730d3b13076640942f813df
@openzeppelin/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol	68bcca423fc72ec9625e219c9e36306c726a347e43f3711467c579bd3f6500c8
@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol	4e09a7479aa3e7c313f8fc141c4c8fc04e0abfeb8754615ef7d78ec94c298b07

@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol	b7410d275fc7d26e36b0851541d6ff290593ba72d64b5c906978124b123915c1
@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol	35fb271561f3dc72e91b3a42c6e40c2bb2e788cd8ca58014ac43f6198b8d32ca
@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol	5fb301961e45cb482fe4e05646d2f529aa449fe0e90c6671475d6a32356fa2d4
contracts/DIDToken.sol	b1ca297ba3804ef02abd9f600524a3f4d713257032e786155447065c44fdedf1
contracts/libraries/ERC20TaxTokenU.sol	7f22599e490bdac07432cccec55771f50c269f1ac232486a88c86f3ef9cb78c2

Introduction

The contract that implements the token functionality is https://etherscan.io/address/0x10e449fb87cde6fd6b82a3cdd4bae283c2f34729

The proxy address that points to this contract during the audit phase is https://etherscan.io/token/0x6a11ac79a3968a4cec0b0aba8cca3bb71ff4e27c

This audit is dedicated for the address 0x10e449fb87cde6fd6b82a3cdd4bae283c2f34729.

The proxy contract has applied a multi-sig mechanism. The GnosisSafeProxy with the address https://etherscan.io/address/0x9345193fB1509525bCFe2c90B71EC204dbB9A8E8 is handling the multi-sig functionality.

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	multi-sig
🔴	BT	Burns Tokens	multi-sig
🔵	BC	Blacklists Addresses	Passed

MT - Mints Tokens

Criticality	critical
Location	contract.sol#L37
Status	multi-sig

Description

The contract owner has the authority to mint tokens. The owner may take advantage of it by calling the mint function. As a result, the contract tokens will be highly inflated.

```
function mint(address to, uint256 amount) external onlyOwner {
        require(to != address(0x0), "zero address");
        _mint(to, amount);
    }
```

Recommendation

The owner should carefully manage the credentials of the owner's account. We advised considering an extra-strong security mechanism that the actions may be quarantined by many users instead of one. The owner could also renounce the contract ownership for a period of time or pass the access to the zero address.

Team update

The ownership of the contract has been moved to a multi-signature wallet. It requires multiple wallets to sign and approve the owner functions. This is an extra security mechanism.

BT - Burns Tokens

Criticality	critical
Location	contract.sol#L44
Status	multi-sig

Description

The contract owner has the authority to burn tokens from a specific address. The owner may take advantage of it by calling the `burn` function. As a result, the targeted contract address will lose the corresponding tokens.

```
function burn(address from, uint256 amount) external onlyOwner {
        require(from != address(0x0), "zero address");
        _burn(from, amount);
    }
```

Recommendation

The owner should carefully manage the credentials of the owner's account. We advised considering an extra-strong security mechanism that the actions may be quarantined by many users instead of one. The owner could also renounce the contract ownership for a period of time or pass the access to the zero address.

Team update

The ownership of the contract has been moved to a multi-signature wallet. It requires multiple wallets to sign and approve the owner functions. This is an extra security mechanism.

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	L01	Public Function could be Declared External	Unresolved
●	L05	Unused State Variable	Unresolved

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contracts/DIDToken.sol#L20
Status	Unresolved

Description

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

```
initialize
```

Recommendation

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

L05 - Unused State Variable

Criticality	minor / informative
Location	contracts/DIDToken.sol#L7
Status	Unresolved

Description

There are segments that contain unused state variables.

```
DIDToken
```

Recommendation

Remove unused state variables.

Contract Functions

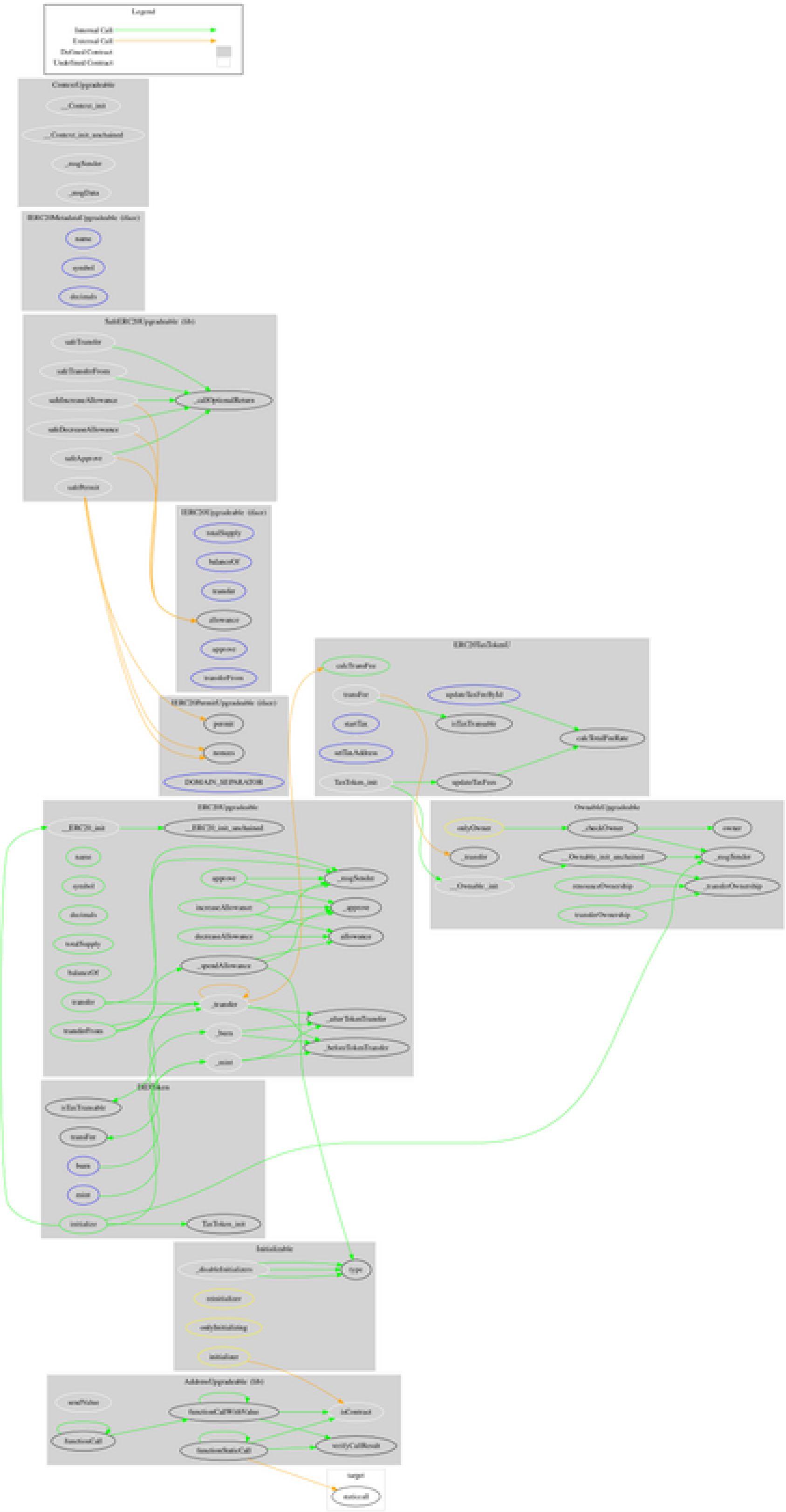
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
OwnableUpgradeable	Implementation	Initializable, ContextUpgradeable		
	__Ownable_init	Internal	✓	onlyInitializing
	__Ownable_init_unchained	Internal	✓	onlyInitializing
	owner	Public		-
	_checkOwner	Internal		
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
Initializable	Implementation			
	_disableInitializers	Internal	✓	
ERC20Upgradeable	Implementation	Initializable, ContextUpgradeable, IERC20Upgradeable, IERC20MetadataUpgradeable		
	__ERC20_init	Internal	✓	onlyInitializing
	__ERC20_init_unchained	Internal	✓	onlyInitializing
	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	✓	
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
IERC20PermitUpgradeable	Interface			
	permit	External	✓	-
	nonces	External		-
	DOMAIN_SEPARATOR	External		-
IERC20MetadataUpgradeable	Interface	IERC20Upgradeable		
	name	External		-
	symbol	External		-
	decimals	External		-
IERC20Upgradeable	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
SafeERC20Upgradeable	Library			

	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeApprove	Internal	✓	
	safeIncreaseAllowance	Internal	✓	
	safeDecreaseAllowance	Internal	✓	
	safePermit	Internal	✓	
	_callOptionalReturn	Private	✓	
AddressUpgradeable	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	verifyCallResult	Internal		
ContextUpgradeable	Implementation	Initializable		
	__Context_init	Internal	✓	onlyInitializing
	__Context_init_unchained	Internal	✓	onlyInitializing
	_msgSender	Internal		
	_msgData	Internal		
DIDToken	Implementation	ERC20TaxTokenU		
	initialize	Public	✓	initializer
	mint	External	✓	onlyOwner
	burn	External	✓	onlyOwner
	_transfer	Internal	✓	

ERC20TaxTokenU	Implementation	ERC20Upgradeable, OwnableUpgradeable		
	TaxToken_init	Internal	✓	initializer
	updateTaxFees	Public	✓	onlyOwner
	updateTaxFeeById	External	✓	onlyOwner
	startTax	External	✓	onlyOwner
	setTaxAddress	External	✓	onlyOwner
	calcTransFee	Public		-
	isTaxTransable	Public		-
	transFee	Internal	✓	
	calcTotalFeeRate	Private	✓	

Contract Flow



Summary

There are some functions that can be abused by the owner like minting tokens and burning tokens. if the contract owner abuses the mint functionality, then the contract will be highly inflated. if the contract owner abuses the burn functionality, then the users could lose their tokens. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. There is also a limit of max 2% fees.

Team update

The ownership of the contract has been moved to a multi-signature wallet. It requires multiple wallets to sign and approve the owner functions. This is an extra security mechanism.

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