Cyberscope

Audit Report

Unidoge Finance

November 2022

Type	ERC20
Network	DOGE
Address	0x565F1D2Fc198C8661b63dCf39bEFB39E5Bb736D6
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	UnidogeFinanceToken
Compiler Version	v0.8.16+commit.07a7930e
Optimization	200 runs
Explorer	https://explorer.dogechain.dog/address/0x565F1D2Fc198C8661b63dCf39bEFB39E5Bb736D6
Current Supply	0
Symbol	UNIDO
Decimals	18
Domain	unidoge.finance

Source Files

Filename	**SHA256**
contract.sol	1c954b1894ba79d67129097ee619fd60b14457478b7c6585e6d6ebd3eccfeee1

Audit Updates

Initial Audit	6th November 2022
Corrected	

Contract Analysis

🔴 Critical 🟡 Medium ⚪ Minor / Informative 🔵 Pass

Severity	Code	Description	Status
⚪	ST	Stops Transactions	Unresolved
🔵	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	Unresolved
🔵	BT	Burns Tokens	Passed
🔵	BC	Blacklists Addresses	Passed

ST - Stops Transactions

Criticality	minor / informative
Location	contract.sol#L2193
Status	Unresolved

Description

The PAUSER role has the authority to stop the transactions for all users including the owner. The caller may take advantage of it by calling the pause method.

```
function pause() public onlyRole(PAUSER_ROLE) {
    _pause();
}

function unpause() public onlyRole(PAUSER_ROLE) {
    _unpause();
}
```

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.

MT - Mints Tokens

Criticality	critical
Location	contract.sol#L2104
Status	Unresolved

Description

The MINTER role has the authority to mint tokens. The caller may take advantage of it by calling the `mint` function. Currently, the token has 0 total supply and the MINTER can mint up to 250,000,000 tokens. If the mint is abused by the MINTER role then contract tokens will be highly inflated.

```
function mint(address to, uint256 amount) public onlyRole(MINTER_ROLE) {
    require(
        amount + totalSupply() <= MAX_TOTAL_SUPPLY,
        "Cant mint more than max supply"
    );
    _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.

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	L02	State Variables could be Declared Constant	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L09	Dead Code Elimination	Unresolved
●	L12	Using Variables before Declaration	Unresolved
●	L15	Local Scope Variable Shadowing	Unresolved

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	contract.sol#L2095
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
MAX_TOTAL_SUPPLY
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L966,2095,1110,962,94,961,1155,964,960,965
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.

```
_TYPE_HASH
MAX_TOTAL_SUPPLY
_PERMIT_TYPEHASH
_CACHED_THIS
DOMAIN_SEPARATOR
_CACHED_CHAIN_ID
_HASHED_NAME
_CACHED_DOMAIN_SEPARATOR
_HASHED_VERSION
...
```

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 / informative
Location	contract.sol#L1963,1558,1822,1876,2074,799,1946,2007,668,1936,2046,754,897,2031,643,1077,2015,1890,911,1069,1909,812,2055,1798,1857,1919,827,1847
Status	Unresolved

Description

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

```
verifyCallResult
_setRoleAdmin
sendValue
functionCallWithValue
_callOptionalReturn
recover
functionDelegateCall
safeTransfer
toHexString
...
```

Recommendation

Remove unused functions.

L12 - Using Variables before Declaration

Criticality	minor / informative
Location	contract.sol#L759
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.

```
r
```

Recommendation

The variables should be declared before any usage of them.

L15 - Local Scope Variable Shadowing

Criticality	minor / informative
Location	contract.sol#L1118
Status	Unresolved

Description

The are variables that are defined in the local scope containing the same name from an upper scope.

```
name
```

Recommendation

The local variables should have different names from the upper scoped variables.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20Permit	Interface			
	permit	External	✓	-
	nonces	External		-
	DOMAIN_SEPARATOR	External		-
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		-

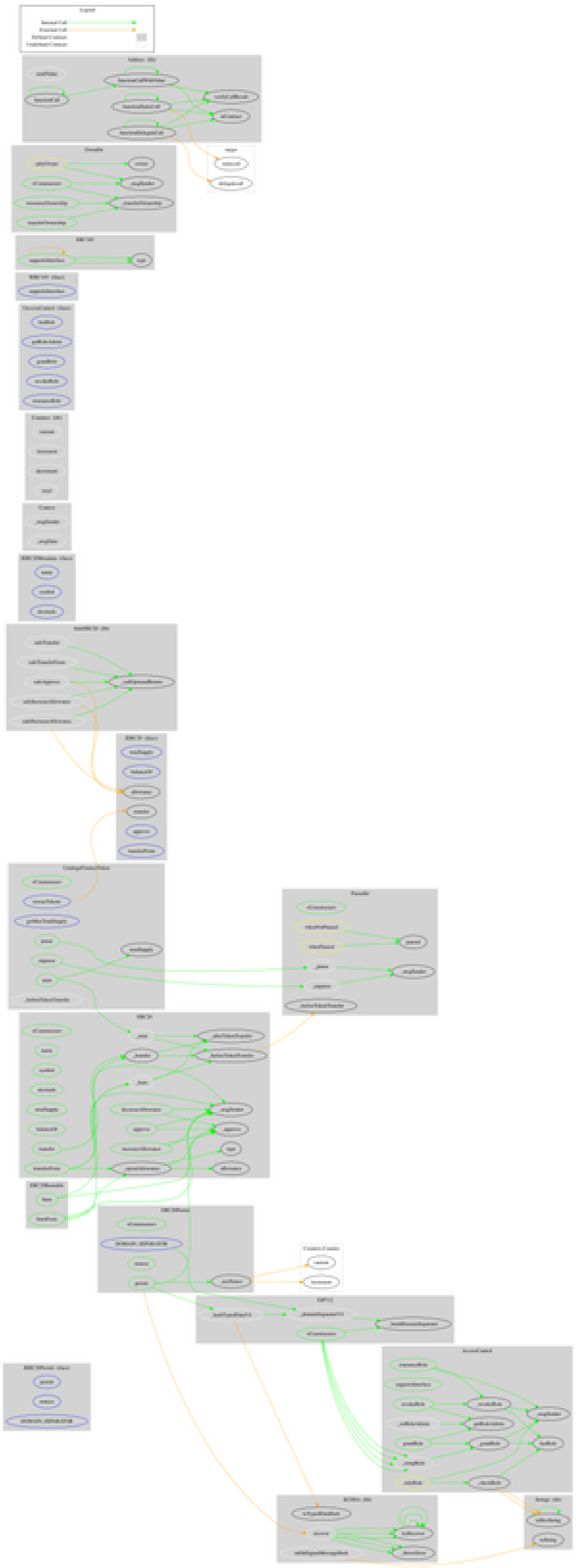
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
ECDSA	Library			
	_throwError	Private		
	tryRecover	Internal		
	recover	Internal		
	tryRecover	Internal		
	recover	Internal		
	tryRecover	Internal		
	recover	Internal		
	toEthSignedMessageHash	Internal		
	toEthSignedMessageHash	Internal		
	toTypedDataHash	Internal		
EIP712	Implementation			

	<Constructor>	Public	✓	-
	_domainSeparatorV4	Internal		
	_buildDomainSeparator	Private		
	_hashTypedDataV4	Internal		
Counters	Library			
	current	Internal		
	increment	Internal	✓	
	decrement	Internal	✓	
	reset	Internal	✓	
ERC20Permit	Implementation	ERC20, IERC20Permit, EIP712		
	<Constructor>	Public	✓	EIP712
	permit	Public	✓	-
	nonces	Public		-
	DOMAIN_SEPARATOR	External		-
	_useNonce	Internal	✓	
ERC20Burnable	Implementation	Context, ERC20		
	burn	Public	✓	-
	burnFrom	Public	✓	-
IAccessControl	Interface			
	hasRole	External		-
	getRoleAdmin	External		-
	grantRole	External	✓	-
	revokeRole	External	✓	-
	renounceRole	External	✓	-
IERC165	Interface			
	supportsInterface	External		-
ERC165	Implementation	IERC165		

	supportsInterface	Public		-
AccessControl	Implementation	Context, IAccessControl, ERC165		
	supportsInterface	Public		-
	hasRole	Public		-
	_checkRole	Internal		
	getRoleAdmin	Public		-
	grantRole	Public	✓	onlyRole
	revokeRole	Public	✓	onlyRole
	renounceRole	Public	✓	-
	_setupRole	Internal	✓	
	_setRoleAdmin	Internal	✓	
	_grantRole	Internal	✓	
	_revokeRole	Internal	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
Pausable	Implementation	Context		
	<Constructor>	Public	✓	-
	paused	Public		-
	_pause	Internal	✓	whenNotPaused
	_unpause	Internal	✓	whenPaused
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	

	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResult	Internal		
SafeERC20	Library			
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeApprove	Internal	✓	
	safeIncreaseAllowance	Internal	✓	
	safeDecreaseAllowance	Internal	✓	
	_callOptionalReturn	Private	✓	
UnidogeFinanceToken	Implementation	ERC20Burnable, ERC20Permit, AccessControl, Pausable		
	<Constructor>	Public	✓	ERC20 ERC20Permit
	mint	Public	✓	onlyRole
	getMaxTotalSupply	External		-
	rescueTokens	External	✓	onlyRole
	pause	Public	✓	onlyRole
	unpause	Public	✓	onlyRole
	_beforeTokenTransfer	Internal	✓	whenNotPaused

Contract Flow



Domain Info

Domain Name	unidoge.finance
Registry Domain ID	507ab52fa9714a0eb6e7d004fb627d0f-DONUTS
Creation Date	2022-10-07T08:11:28Z
Updated Date	2022-10-12T08:12:09Z
Registry Expiry Date	2023-10-07T08:11:28Z
Registrar WHOIS Server	whois.dynadot.com
Registrar URL	http://dynadot.com
Registrar	Dynadot, LLC
Registrar IANA ID	472

The domain was created 30 days before the creation of the audit. It will expire in 11 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 stopping transactions and minting tokens. if the contract owner abuses the mint functionality, then the contract will be highly inflated. 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 25% fees.

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