



Cyberscope

Audit Report

ShieldDAO Token

July 2022

Github <https://github.com/shielddoodapp/ShieldDao-Contracts>
File [contracts/TimeERC20Token.sol](#)
Commit [6f12869de64b26f3a68fc4e2bd7717790e63cdb2](#)
Audited by © cyberscope

Table of Contents

| | |
|---|-----------|
| Table of Contents | 1 |
| Contract Review | 2 |
| Source Files | 2 |
| Audit Updates | 2 |
| Contract Analysis | 3 |
| MT - Mint Tokens | 4 |
| Description | 4 |
| Recommendation | 4 |
| Contract Diagnostics | 5 |
| L01 - Public Function could be Declared External | 6 |
| Description | 6 |
| Recommendation | 6 |
| L02 - State Variables could be Declared Constant | 7 |
| Description | 7 |
| Recommendation | 7 |
| L04 - Conformance to Solidity Naming Conventions | 8 |
| Description | 8 |
| Recommendation | 8 |
| L09 - Dead Code Elimination | 9 |
| Description | 9 |
| Recommendation | 9 |
| Contract Functions | 10 |
| Contract Flow | 13 |
| Summary | 14 |
| Disclaimer | 15 |
| About Cyberscope | 16 |

Contract Review

| | |
|-------------------------|---|
| Contract Name | TimeERC20Token |
| Compiler Version | v0.8.12+commit.f00d7308 |
| Github | https://github.com/shielddaodapp/ShieldDao-Contracts |
| File | contracts/TimeERC20Token.sol |
| Commit | 6f12869de64b26f3a68fc4e2bd7717790e63cdb2 |
| Test Deploy | https://testnet.bscscan.com/token/0x78d893ccC7c80F3Fbc8Ac0B82Ab7D4C6B6415373 |
| Symbol | SDD |
| Decimals | 9 |

Source Files

| Filename | SHA256 |
|--------------|--|
| contract.sol | a30a0d8ed9d489e82c8a838068603c617c4977396d1538257358c9cecf170c2a |

Audit Updates

| | |
|----------------------|----------------|
| Initial Audit | 18th July 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 |

MT - Mint Tokens

| | |
|--------------------|-------------------|
| Criticality | critical |
| Location | contract.sol#L411 |

Description

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

```
function mint(address account_, uint256 amount_) external onlyVault() {  
    _mint(account_, amount_);  
}
```

Recommendation

The owner and the 'vault' role should carefully manage the credentials of the owner's account. We state that the owner privileges are necessary and required for proper protocol operations. Thus, we emphasise the contract owner to be extra careful with the credentials. 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

| Severity | Code | Description |
|----------|------|--|
| ● | L01 | Public Function could be Declared External |
| ● | L02 | State Variables could be Declared Constant |
| ● | L04 | Conformance to Solidity Naming Conventions |
| ● | L09 | Dead Code Elimination |

L01 - Public Function could be Declared External

| | |
|--------------------|---|
| Criticality | minor |
| Location | contract.sol#L158,162,166,170,174,187,192,199,204,312,335,359,368,373,392 |

Description

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

```
vault
transferOwnership
renounceOwnership
owner
nonces
permit
decreaseAllowance
increaseAllowance
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#L146,144

Description

Constant state variables should be declared constant to save gas.

```
otherFee  
baseAmount
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

| | |
|--------------------|---|
| Criticality | minor |
| Location | contract.sol#L127,130,133,136,139,142,293,350,382 |

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.

```
_vault  
_owner  
DOMAIN_SEPARATOR  
_decimals  
_symbol  
_name  
_totalSupply  
_allowances  
_balances
```

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#L265

Description

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

`decrement`

Recommendation

Remove unused functions.

Contract Functions

| Contract | Type | Bases | | |
|-----------------------|-------------------|------------|------------|-----------|
| | Function Name | Visibility | Mutability | Modifiers |
| IERC20 | Interface | | | |
| | totalSupply | External | | - |
| | balanceOf | External | | - |
| | transfer | External | ✓ | - |
| | allowance | External | | - |
| | approve | External | ✓ | - |
| | transferFrom | External | ✓ | - |
| LowGasSafeMath | Library | | | |
| | add | Internal | | |
| | sub | Internal | | |
| | mul | Internal | | |
| | div | Internal | | |
| | add | Internal | | |
| | sub | Internal | | |
| ERC20 | Implementation | IERC20 | | |
| | <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 | ✓ | |
| | | | | |
| Counters | Library | | | |
| | current | Internal | | |
| | increment | Internal | ✓ | |
| | decrement | Internal | ✓ | |
| | | | | |
| IERC2612Permit | Interface | | | |
| | permit | External | ✓ | - |
| | nonces | External | | - |
| | | | | |
| ERC20Permit | Implementation | ERC20, IERC2612Permit | | |
| | <Constructor> | Public | ✓ | - |
| | permit | Public | ✓ | - |
| | nonces | Public | | - |
| | | | | |
| IOwnable | Interface | | | |
| | owner | External | | - |
| | renounceOwnership | External | ✓ | - |
| | transferOwnership | External | ✓ | - |
| | | | | |
| Ownable | Implementation | IOwnable | | |
| | <Constructor> | Public | ✓ | - |
| | owner | Public | | - |
| | renounceOwnership | Public | ✓ | onlyOwner |
| | transferOwnership | Public | ✓ | onlyOwner |
| | | | | |
| VaultOwned | Implementation | Ownable | | |

| | | | | |
|-----------------------|----------------|----------------------------|---|-----------|
| | setVault | External | ✓ | onlyOwner |
| | vault | Public | | - |
| | | | | |
| TimeERC20Token | Implementation | ERC20Permit, VaultOwned | | |
| | <Constructor> | Public | ✓ | ERC20 |
| | mint | External | ✓ | onlyVault |
| | burn | External | ✓ | - |
| | burnFrom | External | ✓ | - |
| | _burnFrom | Internal | ✓ | |

Contract Flow



Summary

The Smart Contract analysis reported one critical severity issue. The contract owner has the authority to mint tokens. If the contract owner abuses the mint functionality, then the contract will be highly inflated. We state that the owner privileges are necessary and required for proper protocol operations. Thus, we emphasise the contract owner to be extra careful with the credentials. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. The fees are fixed to 3%.

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>