Cyberscope

Audit Report

Minteo Wagmi

September 2023

Repository	https://github.com/minteo-wagmi/rwa-contracts/tree/main/src
Commit	0ccb0d151cba3bb6546179630e479ee943f1dd1e
Audited by	© cyberscope

Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Unresolved
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	MT	Mints Tokens	Unresolved
●	BT	Burns Tokens	Unresolved
●	BC	Blacklists Addresses	Unresolved

Diagnostics

🔴 Critical 🟠 Medium ● Minor / Informative

Severity	Code	Description	Status
●	OCTD	Transfers Contract's Tokens	Unresolved
●	RES	Redundant Event Statement	Unresolved
●	RSW	Redundant Storage Writes	Unresolved

Table of Contents

Review

Repository	https://github.com/minteo-wagmi/rwa-contracts/tree/main/src
Commit	0ccb0d151cba3bb6546179630e479ee943f1dd1e

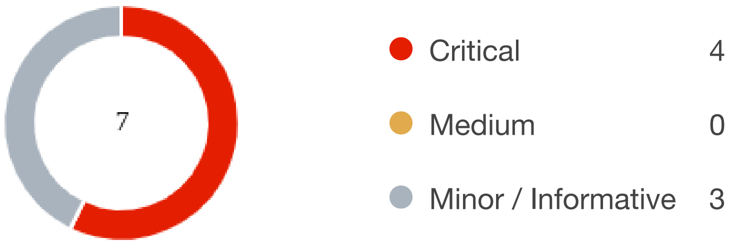
Audit Updates

Initial Audit	13 Sep 2023

Source Files

Filename	SHA256
Token.sol	f9595eb4395751561bcc209bdbfc4a169eeba2e49da0bb62d09cd096d0e59cc3
Freezable.sol	08f5cb37d9ca57e6148cd9fd287dd098b734d8b47c14a07472831163d39b2b18

Findings Breakdown



Severity		Unresolved	Acknowledged	Resolved	Other
●	Critical	4	0	0	0
●	Medium	0	0	0	0
●	Minor / Informative	3	0	0	0

ST - Stops Transactions

Criticality	Critical
Location	Token.sol#L66
Status	Unresolved

Description

The `PAUSER_ROLE` account has the authority to stop the transactions for all users. The `PAUSER_ROLE` account may take advantage of it by calling the `pause` function.

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

Recommendation

The team should carefully manage the private keys of the `PAUSER_ROLE` account's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract `PAUSER_ROLE` functions. Some suggestions are:

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-sign wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.
- Renouncing the ownership will eliminate the threats but it is non-reversible.

MT - Mints Tokens

Criticality	Critical
Location	Token.sol#L74
Status	Unresolved

Description

The `MINTER_ROLE` account has the authority to mint tokens. The `MINTER_ROLE` account 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
onlyRole(MINTER_ROLE) {
    _mint(to, amount);
  }
```

Recommendation

The team should carefully manage the private keys of the `MINTER_ROLE` account's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract `MINTER_ROLE` functions. Some suggestions are:

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-sign wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.
- Renouncing the ownership will eliminate the threats but it is non-reversible.

BT - Burns Tokens

Criticality	Critical
Location	Token.sol#L90
Status	Unresolved

Description

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

```
 function burnFrozen(address account, uint256 amount) external
onlyRole(FREEZER_ROLE) whenFrozen(account) {
     _thaw(account);
     _burn(account, amount);
     _freeze(account);
  }
```

Recommendation

The team should carefully manage the private keys of the `FREEZER_ROLE` account's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract `FREEZER_ROLE` functions. Some suggestions are:

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-sign wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.
- Renouncing the ownership will eliminate the threats but it is non-reversible.

BC - Blacklists Addresses

Criticality	Critical
Location	Token.sol#L78,96Freezable.sol#L31
Status	Unresolved

Description

The `FREEZER_ROLE` account has the authority to stop addresses from transactions. The `FREEZER_ROLE` account may take advantage of it by calling the `freeze` function.

```
function freeze(address account) external
onlyRole(FREEZER_ROLE) {
    _freeze(account);
  }

  function _beforeTokenTransfer(address from, address to,
uint256 amount)
    internal
    override
    whenNotPaused
    whenNotFrozen(from)
    whenNotFrozen(to)
  {
    super._beforeTokenTransfer(from, to, amount);
  }

  function _freeze(address _account) internal {
    isFrozen[_account] = true;
    emit Frozen(_account);
  }
```

Recommendation

The team should carefully manage the private keys of the `FREEZER_ROLE` account's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract `FREEZER_ROLE` functions. Some suggestions are:

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-sign wallet so that many addresses will confirm the action.

- Introduce a governance model where users will vote about the actions.
- Renouncing the ownership will eliminate the threats but it is non-reversible.

OCTD - Transfers Contract's Tokens

Criticality	Minor / Informative
Location	Token.sol#L86
Status	Unresolved

Description

The `RESCUER_ROLE` account has the authority to claim all the balance of the contract. The `RESCUER_ROLE` account may take advantage of it by calling the `rescueFunds` function.

```
 function rescueFunds(IERC20 tokenContract, address to, uint256
amount) external onlyRole(RESCUER_ROLE) {
     tokenContract.safeTransfer(to, amount);
   }
```

Recommendation

The team should carefully manage the private keys of the `RESCUER_ROLE` account's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract `RESCUER_ROLE` functions. Some suggestions are:

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-sign wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.
- Renouncing the ownership will eliminate the threats but it is non-reversible.

RES - Redundant Event Statement

Criticality	Minor / Informative
Location	Freezable.sol#L19
Status	Unresolved

Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations.

The `BurnedFrozen` event statement is not used in the contract's implemantation.

```
event BurnedFrozen(address indexed account, uint256 amount);
```

Recommendation

The team is advised to take these segments into consideration and rewrite them so the runtime will be more performant. That way it will improve the efficiency and performance of the source code and reduce the cost of executing it. It is recommend removing the unused event statement from the contract..

RSW - Redundant Storage Writes

Criticality	Minor / Informative
Location	Freezable.sol#L31
Status	Unresolved

Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations.

The contract updates the `isFrozen` status of an account within the `_freeze` and `_thaw` functions even if its current state is the same as the one passed as an argument. As a result, the contract performs redundant storage writes.

```
function _freeze(address _account) internal {
    isFrozen[_account] = true;
    emit Frozen(_account);
}

function _thaw(address _account) internal {
    delete isFrozen[_account];
    emit Thawed(_account);
}
```

Recommendation

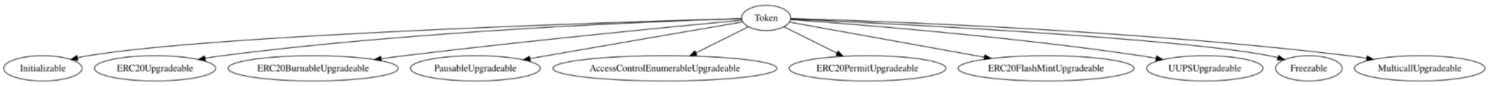
The team is advised to take these segments into consideration and rewrite them so the runtime will be more performant. That way it will improve the efficiency and performance of the source code and reduce the cost of executing it.

Functions Analysis

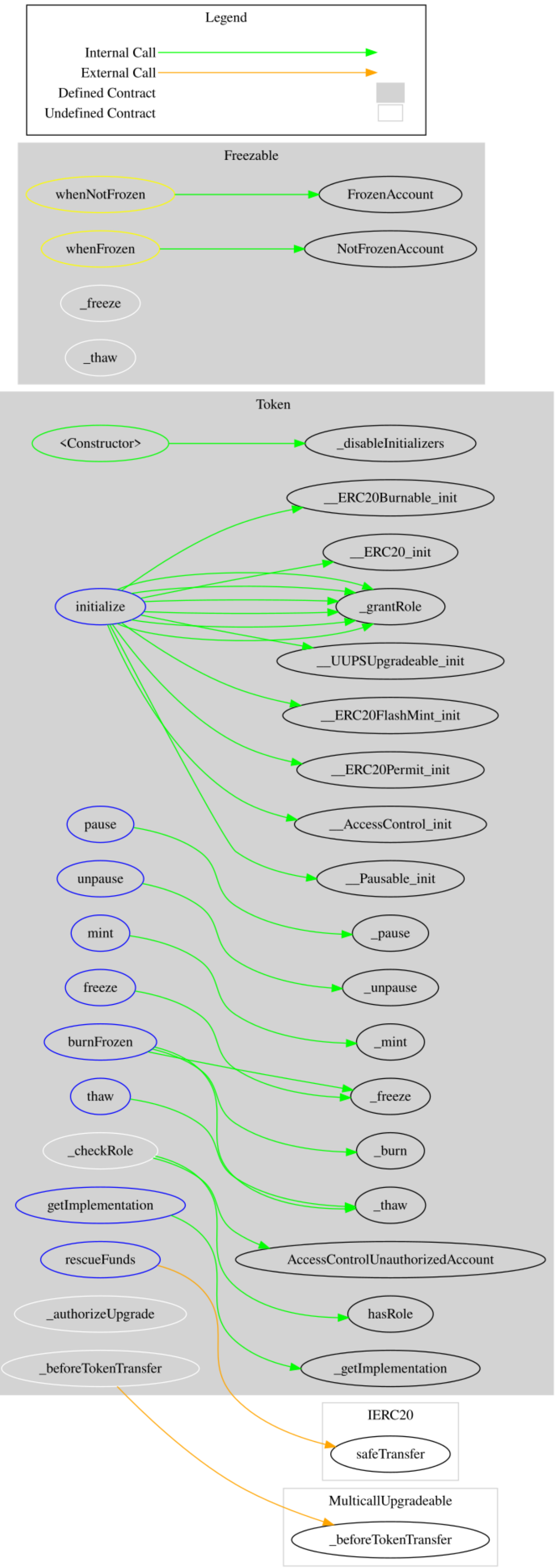
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Token	Implementation	Initializable, ERC20Upgradeable, ERC20BurnableUpgradeable, PausableUpgradeable, AccessControlEnumerableUpgradeable, ERC20PermitUpgradeable, ERC20FlashMintUpgradeable, UUPSUpgradeable, Freezable, MulticallUpgradeable		
		Public	✓	-
	initialize	External	✓	initializer
	pause	External	✓	onlyRole
	unpause	External	✓	onlyRole
	mint	External	✓	onlyRole
	freeze	External	✓	onlyRole
	thaw	External	✓	onlyRole
	rescueFunds	External	✓	onlyRole
	burnFrozen	External	✓	onlyRole whenFrozen

	_beforeTokenTransfer	Internal	✓	whenNotPaused whenNotFrozen whenNotFrozen
	_checkRole	Internal		
	_authorizeUpgrade	Internal	✓	onlyRole
	getImplementation	External		-
Freezable	Implementation			
	_freeze	Internal	✓	
	_thaw	Internal	✓	

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Freezable
whenNotFrozen
FrozenAccount
whenFrozen
NotFrozenAccount
_freeze
_thaw
Token
<Constructor>
_disableInitializers
__ERC20Burnable_init
__ERC20_init
initialize
_grantRole
__UUPSUpgradeable_init
__ERC20FlashMint_init
__ERC20Permit_init
pause
__AccessControl_init
unpause
__Pausable_init
mint
_pause
freeze
_unpause
burnFrozen
_mint
thaw
_freeze
_checkRole
_burn
getImplementation
_thaw
rescueFunds
AccessControlUnauthorizedAccount
_authorizeUpgrade
hasRole
_beforeTokenTransfer
_getImplementation
IERC20
safeTransfer
MulticallUpgradeable
_beforeTokenTransfer

Summary

Minteo Wagmi contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. There are some functions that can be abused by specific addresses like stop transactions, mint tokens, burn tokens from any address and massively blacklist addresses. If these addresses abuse the mint functionality, then the contract will be highly inflated. If abuse the burn functionality, then the users could lost 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.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope's prior written consent. This report is not nor should be considered an "endorsement" or "disapproval" of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io