Cyberscope

Audit Report

Minteo Wagmi

September 2023

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

Commit b85fd1cfe2e14132a334f228e67d8b74027422fb

Audited by © cyberscope

Table of Contents

Table of Contents 1
Review 2
Audit Updates 2
Source Files 2
Findings Breakdown 2
Diagnostics 4
CCR - Contract Centralization Risk 5
Description 5
Recommendation 6
Team Update 7
OCTD - Transfers Contract's Tokens 8
Description 8
Recommendation 8
Team Update 8
Functions Analysis 9
Inheritance Graph 11
Flow Graph 12
Summary 13
Disclaimer 14
About Cyberscope 15

Review

Repository	https://github.com/minteo-wagmi/rwa-contracts/tree/main/src
Commit	b85fd1cfe2e14132a334f228e67d8b74027422fb
Testing Deploy	https://testnet.bscscan.com/address/0x0eae9053a2a864262255f1d3d0f79571cd45ccf5

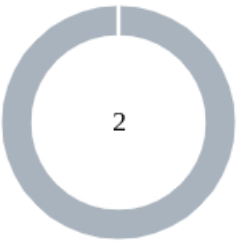
Audit Updates

Initial Audit	13 Sep 2023 https://github.com/cyberscope-io/audits/blob/main/minteo-wagmi/v1/audit.pdf
Corrected Phase 2	27 Sep 2023

Source Files

Filename	SHA256
contracts/Token.sol	a777a912767a7e431925664e7d00a8b2a87d80c1ff931d55a1dfe94c0e09e7df
contracts/Freezable.sol	3ac0b7215cd636346d47cc023e3c6d00ba425bb7b30dbf414b0ffe7885fef5e8

Findings Breakdown



Critical	0
Medium	0
Minor / Informative	2

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	0	0	0	0
Medium	0	0	0	0
Minor / Informative	0	1	0	1

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
	CCR	Contract Centralization Risk	Multisign
	OCTD	Transfers Contract's Tokens	Acknowledged

CCR - Contract Centralization Risk

Criticality	Minor / Informative
Location	contracts/Token.sol#L66,74,78,90,96 Freezable.sol#L31
Status	Multisign

Description

The contract is heavily dependent on configurations determined by specific role accounts, which control the minting, burning, and freezing of tokens. These configurations and the associated permissions are concentrated by specific address, and as a result, produce a centralization risk.

Specifically, the contract contains critical functionality functions that can be invoked by specific accounts with the corresponding role. Namely, the following role accounts can invoke the following functions:

- 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.
- The `MINTER_ROLE` account has the authority to mint tokens by invoking the `mint` function. As a result, the contract tokens can be highly inflated.
- The `FREEZER_ROLE` account has the authority to burn tokens from a specific address by invoking the `burnFrozen` function. As a result, the targeted address will lose the corresponding tokens.
- The `FREEZER_ROLE` account also has the authority to stop addresses from transacting. The `FREEZER_ROLE` account can blacklist addresses by calling the `freeze` function, thereby preventing them from making transactions.

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

function mint(address to, uint256 amount) external onlyRole(MINTER_ROLE)
{
    _mint(to, amount);
  }

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

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

To address this finding and mitigate centralization risks, it is recommended to evaluate the feasibility of migrating critical configurations and functionality into the contract's codebase itself. This approach would reduce external dependencies and enhance the contract's self-sufficiency. It is essential to carefully weigh the trade-offs between external configuration flexibility and the risks associated with centralization.

Team Update

The team has acknowledged that this is not a security issue and states:

"We want to clarify that these functions, which include mint, burn, pause, and freeze, are not critical vulnerabilities but essential components for regulatory compliance and security in real-world asset tokenization. We employ a multisig wallet and rigorous security measures to safeguard these functions, ensuring accountability. Our commitment to transparency of our reserves further enhances trust. We welcome collaboration to address concerns and enhance our project's security."

OCTD - Transfers Contract's Tokens

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

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.

Team Update

The team has acknowledged that this is not a security issue and states:

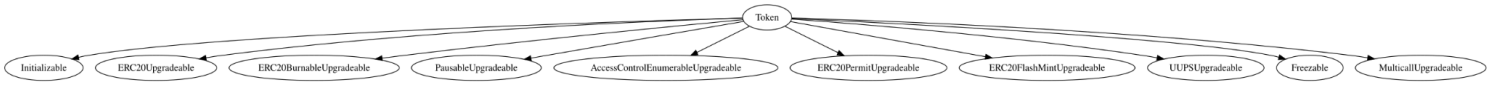
"*Regarding the role-based access control for the rescueFunds function, we want to clarify that this contract is not meant to hold any token balances by design, and the existence of it is purely to return the funds a user might transfer by mistake to the contract.*"

Functions Analysis

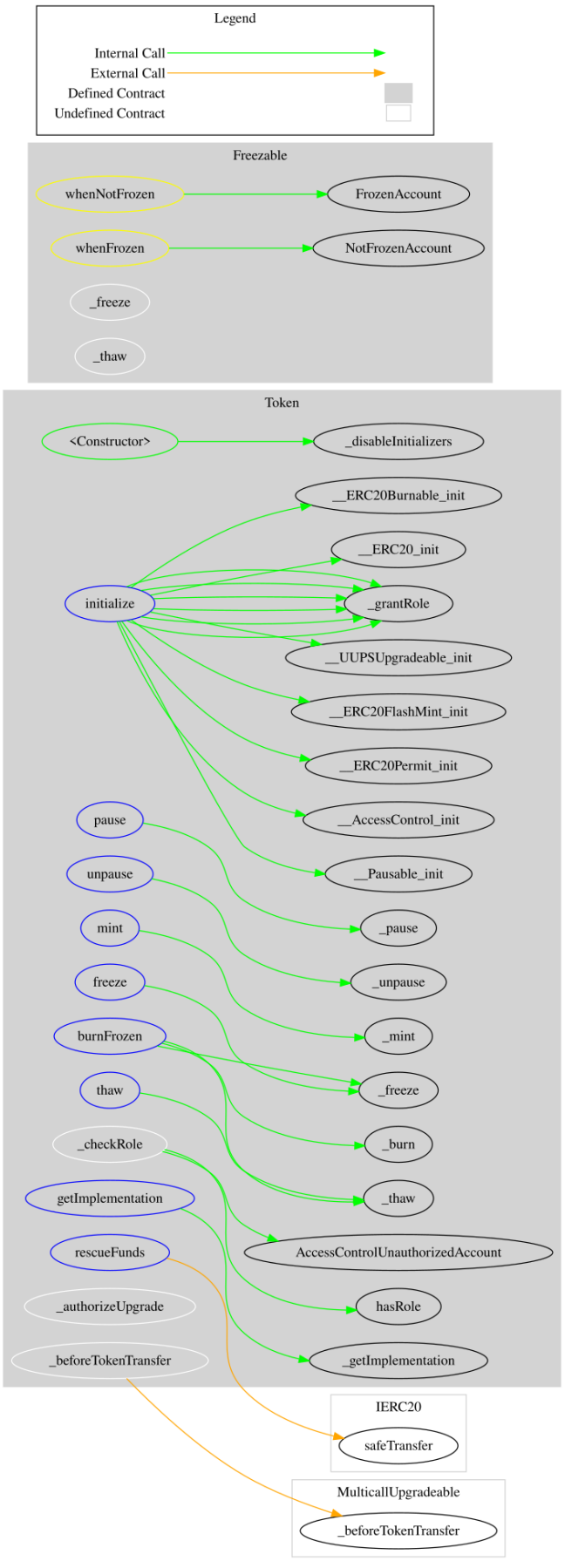
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Token	Implementation	Initializable, ERC20Upgradeable, ERC20BurnableUpgradeable, PausableUpgradeable, AccessControlEnumerableUpgradeable, ERC20PermitUpgradeable, ERC20FlashMintUpgradeable, UUPSUpgradeable, Freezable, MulticallUpgradeable		
		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		-

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
__AccessControl_init
pause
__Pausable_init
unpause
_pause
mint
_unpause
freeze
_mint
burnFrozen
_freeze
thaw
_burn
_checkRole
_thaw
getImplementation
AccessControlUnauthorizedAccount
rescueFunds
hasRole
_authorizeUpgrade
_getImplementation
_beforeTokenTransfer
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 invoked by specific role accounts. These designated accounts have the authority to execute specific functions within the contract. 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