Cyberscope

Audit Report

Mavia

December 2023

Network	ETH
Address	0x24fcfc492c1393274b6bcd568ac9e225bec93584
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	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Unresolved

Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	OCTD	Transfers Contract's Tokens	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved

Table of Contents

Analysis **1**

Diagnostics **2**

Table of Contents **3**

Review **4**

Audit Updates 4

Source Files 4

Findings Breakdown **6**

ST - Stops Transactions 7

Description 7

Recommendation 7

BC - Blacklists Addresses 9

Description 9

Recommendation 9

OCTD - Transfers Contract's Tokens 10

Description 10

Recommendation 10

L04 - Conformance to Solidity Naming Conventions 11

Description 11

Recommendation 12

Functions Analysis **13**

Inheritance Graph **14**

Flow Graph **15**

Summary **16**

Disclaimer **17**

About Cyberscope **18**

Review

Contract Name	MaviaToken
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	200 runs
Explorer	https://etherscan.io/address/0x24fcfc492c1393274b6bcd568ac9e225bec93584
Address	0x24fcfc492c1393274b6bcd568ac9e225bec93584
Network	ETH
Symbol	MAVIA
Decimals	18
Total Supply	250,000,000

Audit Updates

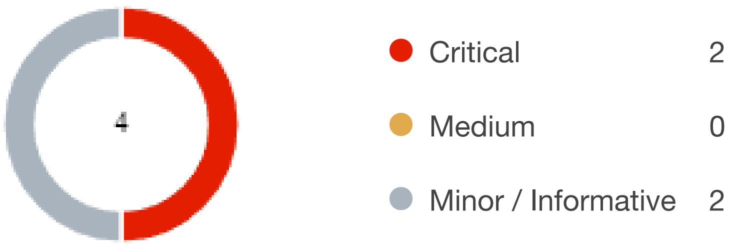
Initial Audit	27 Dec 2023

Source Files

Filename	SHA256
project:/src/active/token/MaviaToken.sol	436944682ccc452003f41c4e2447d94433e3d9f3457389a6a38e80b54dbc6999
@openzeppelin/contracts/utils/Strings.sol	3b2b0d75c7e5688950d3b6e63e46473054395dad6e390431f73febb2199913c5
@openzeppelin/contracts/utils/Context.sol	5828bf38f9376b659a8edbbe2df0d06b29a09e37ecd470465dda2bbcb612c85d

@openzeppelin/contracts/utils/Address.sol	1370d859f5c6d11025afb409d1b724279f663c4cf4bc4d2ba057290bdcf45a66
@openzeppelin/contracts/utils/introspection/IERC165.sol	072805b211a653c333b232a3199b9e65fa7b82fc7a40ee5a3bc8a2dadd1cba01
@openzeppelin/contracts/utils/introspection/ERC165.sol	381b0589da0e1a32242d7314905d2cc6edd8dce8193ddb6bfacc5b685e311422
@openzeppelin/contracts/token/ERC20/IERC20.sol	b2565dec975f684ef0edfa505e212d0d0b602e1311afab782ea06ea8d3f49bb6
@openzeppelin/contracts/token/ERC20/ERC20.sol	80e33e340442acecc4bd995b4ead9b51adc4231c8213357fca18996b945f850b
@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol	729097c056b8bf1dd93ac16831380ce4ff54703d75983f57354240cc8be2edec
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	4e2ce556a0419415ec3b01a0fa0322c20d6d53de5a05728c068e90d5684486c1
@openzeppelin/contracts/access/IAccessControl.sol	81a867af9f5344a0efffcfb2970db5354c8684d4d50139db1524321fbd60979b
@openzeppelin/contracts/access/AccessControl.sol	6815a22e5b2ef7e0e813961ad06afac5c9d6e7cdced9165f2cedbf11032044bd

Findings Breakdown



	Severity	Unresolved	Acknowledged	Resolved	Other
●	Critical	2	0	0	0
●	Medium	0	0	0	0
●	Minor / Informative	2	0	0	0

ST - Stops Transactions

Criticality	Critical
Location	project:/src/active/token/MaviaToken.sol#L100
Status	Unresolved

Description

The `_EDITOR_ROLE`, which is assigned to the contract owner, has the authority to stop the sales for all users excluding the scenarios, where both the `_pSender` and the `_pRecipient` are a `whitelist` address. The editor may take advantage of it by setting the `tfMaxAmount` to zero or by setting the `tfStartTime` to a very high value. As a result, the contract may operate as a honeypot.

```
if (!whitelist[_pSender] && !whitelist[_pRecipient]) {
  require(block.timestamp >= tfStartTime, "Invalid time");
  require(_pAmount <= tfMaxAmount, "Invalid amount");
}
```

Recommendation

The contract could embody a check for not allowing setting the `tfmaxAmount` less than a reasonable amount and the `tfStartTime` to a very high value. A suggested implementation could check that the maximum amount should be more than a fixed percentage of the total supply. 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.

Temporary Solutions:

These measurements do not decrease the severity of the finding

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-signature wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

Permanent Solution:

- Renouncing the ownership, which will eliminate the threats but it is non-reversible.

BC - Blacklists Addresses

Criticality	Critical
Location	project:/src/active/token/MaviaToken.sol#L70
Status	Unresolved

Description

The `_EDITOR_ROLE` , which is assigned to the contract owner has the authority to stop addresses from transactions. The editor may take advantage of it by calling the `fSetBlacklist` function.

```
function fSetBlacklist(address _pAddr, bool _pIsBlacklist)
external onlyRole(_EDITOR_ROLE) {
  require(_pAddr != address(0), "Invalid address");
  blacklist[_pAddr] = _pIsBlacklist;
  emit ESetBlacklist(_pAddr, _pIsBlacklist);
}
```

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.

Temporary Solutions:

These measurements do not decrease the severity of the finding

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-signature wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

Permanent Solution:

- Renouncing the ownership, which will eliminate the threats but it is non-reversible.

OCTD - Transfers Contract's Tokens

Criticality	Minor / Informative
Location	project:/src/active/token/MaviaToken.sol#L88
Status	Unresolved

Description

The `_EMERGENCY_ROLE` has the authority to claim all the balance of the contract. They may take advantage of it by calling the `fEmerERC20Tokens` function.

```
function fEmerERC20Tokens(IERC20 _pToken, address _pTo)
external onlyRole(_EMERGENCY_ROLE) {
  require(_pTo != address(0), "Invalid address");
  uint256 bal_ = _pToken.balanceOf(address(this));
  _pToken.safeTransfer(_pTo, bal_);
  emit EEmerERC20Tokens(_pToken, _pTo);
}
```

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.

Temporary Solutions:

These measurements do not decrease the severity of the finding

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-signature wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

Permanent Solution:

- Renouncing the ownership, which will eliminate the threats but it is non-reversible.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	project:/src/active/token/MaviaToken.sol#L14,66,70,76,82,88
Status	Unresolved

Description

The Solidity style guide is a set of guidelines for writing clean and consistent Solidity code. Adhering to a style guide can help improve the readability and maintainability of the Solidity code, making it easier for others to understand and work with.

The followings are a few key points from the Solidity style guide:

1. Use camelCase for function and variable names, with the first letter in lowercase (e.g., myVariable, updateCounter).
2. Use PascalCase for contract, struct, and enum names, with the first letter in uppercase (e.g., MyContract, UserStruct, ErrorEnum).
3. Use uppercase for constant variables and enums (e.g., MAX_VALUE, ERROR_CODE).
4. Use indentation to improve readability and structure.
5. Use spaces between operators and after commas.
6. Use comments to explain the purpose and behavior of the code.
7. Keep lines short (around 120 characters) to improve readability.

```
bytes32 private _DOMAIN_SEPARATOR
uint256 _pAmount
bool _pIsBlacklist
address _pAddr
bool _pIsWhitelist
uint256 _pMaxAmount
uint256 _pStartTime
address _pTo
IERC20 _pToken
```

Recommendation

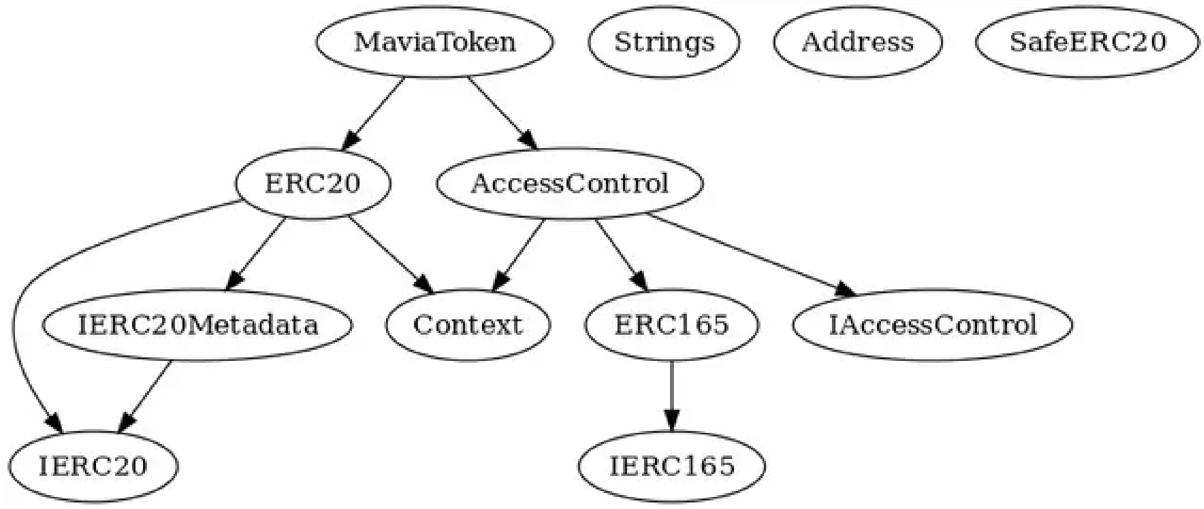
By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.

Find more information on the Solidity documentation https://docs.soliditylang.org/en/v0.8.17/style-guide.html#naming-convention.

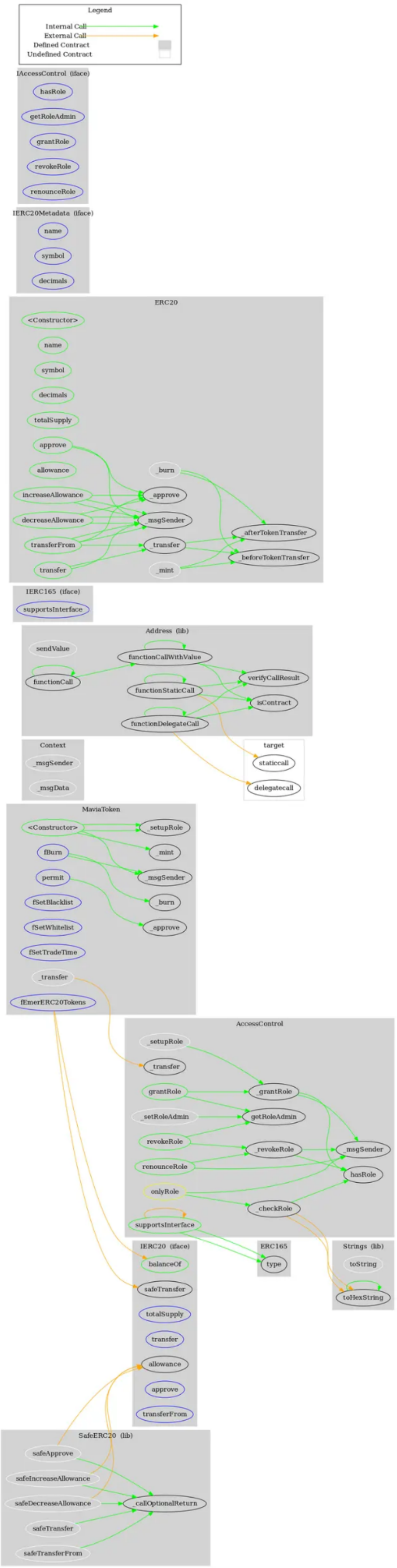
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
MaviaToken	Implementation	ERC20, AccessControl		
		Public	✓	ERC20
	permit	External	✓	-
	fBurn	External	✓	-
	fSetBlacklist	External	✓	onlyRole
	fSetWhitelist	External	✓	onlyRole
	fSetTradeTime	External	✓	onlyRole
	fEmerERC20Tokens	External	✓	onlyRole
	_transfer	Internal	✓	

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
IAccessControl (iface)
hasRole
getRoleAdmin
grantRole
revokeRole
renounceRole
IERC20Metadata (iface)
name
symbol
decimals
ERC20
<Constructor>
name
symbol
decimals
totalSupply
approve
allowance
increaseAllowance
decreaseAllowance
transferFrom
transfer
_burn
_approve
_msgSender
_transfer
_mint
_afterTokenTransfer
_beforeTokenTransfer
IERC165 (iface)
supportsInterface
Address (lib)
sendValue
functionCall
functionCallWithValue
functionStaticCall
functionDelegateCall
verifyCallResult
isContract
Context
_msgSender
_msgData
target
staticcall
delegatecall
MaviaToken
<Constructor>
fBurn
permit
fSetBlacklist
fSetWhitelist
fSetTradeTime
_transfer
fEmerERC20Tokens
_setupRole
_mint
_msgSender
_burn
_approve
AccessControl
_setupRole
_transfer
grantRole
_setRoleAdmin
revokeRole
renounceRole
onlyRole
supportsInterface
_grantRole
getRoleAdmin
_revokeRole
_checkRole
_msgSender
hasRole
IERC20 (iface)
balanceOf
safeTransfer
totalSupply
transfer
allowance
approve
transferFrom
ERC165
type
Strings (lib)
toString
toHexString
SafeERC20 (lib)
safeApprove
safeIncreaseAllowance
safeDecreaseAllowance
safeTransfer
safeTransferFrom
_callOptionalReturn

Summary

Heroes of Mavia 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 the owner like stop transactions and massively blacklist addresses. 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