Cyberscope

Audit Report

MeAI

January 2025

Network	BSC
Address	0x079fF857bD4FF7C4aCf49a1A02E67EF96D43E3E6
Audited by	© cyberscope

Analysis

Critical Medium Minor / Informative Pass

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

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
●	RCT	Redundant Code Template	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L17	Usage of Solidity Assembly	Unresolved

Table of Contents

Risk Classification

The criticality of findings in Cyberscope's smart contract audits is determined by evaluating multiple variables. The two primary variables are:

1. **Likelihood of Exploitation**: This considers how easily an attack can be executed, including the economic feasibility for an attacker.
2. **Impact of Exploitation**: This assesses the potential consequences of an attack, particularly in terms of the loss of funds or disruption to the contract's functionality.

Based on these variables, findings are categorized into the following severity levels:

1. **Critical**: Indicates a vulnerability that is both highly likely to be exploited and can result in significant fund loss or severe disruption. Immediate action is required to address these issues.
2. **Medium**: Refers to vulnerabilities that are either less likely to be exploited or would have a moderate impact if exploited. These issues should be addressed in due course to ensure overall contract security.
3. **Minor**: Involves vulnerabilities that are unlikely to be exploited and would have a minor impact. These findings should still be considered for resolution to maintain best practices in security.
4. **Informative**: Points out potential improvements or informational notes that do not pose an immediate risk. Addressing these can enhance the overall quality and robustness of the contract.

Severity	Likelihood / Impact of Exploitation
Critical	Highly Likely / High Impact
Medium	Less Likely / High Impact or Highly Likely/ Lower Impact
Minor / Informative	Unlikely / Low to no Impact

Review

Contract Name	MeAIToken
Compiler Version	v0.8.17+commit.8df45f5f
Optimization	200 runs
Explorer	https://bscscan.com/address/0x079ff857bd4ff7c4acf49a1a02e67ef96d43e3e6
Address	0x079ff857bd4ff7c4acf49a1a02e67ef96d43e3e6
Network	BSC
Symbol	MEAI
Decimals	18
Total Supply	950.503.714,169

Audit Updates

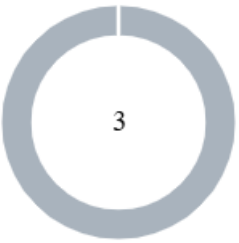
Initial Audit	20 Jan 2025

Source Files

Filename	SHA256
contracts/ReflectiveV3ERC20.sol	07c4ebed4265d8c6797bd6c501d5f0949a96a2b4a1176be7b9dda6bf274b44cf
contracts/MeAIToken.sol	4eb067a3ec12f8a4041511dc29a85a92741712b272f3307b74f843115ed3974c
contracts/lib/LibCommon.sol	ad40e79524942f0927be19739e7c96b7a52147f5cf54fdd7eb676720db70b66a

@openzeppelin/contracts/utils/Context.sol	b2cfee351bcafd0f8f27c72d76c054df9b571b62cfac4781ed12c86354e2a56c
@openzeppelin/contracts/token/ERC20/IERC20.sol	7ebde70853ccafcf1876900dad458f46eb9444d591d39bfc58e952e2582f5587
@openzeppelin/contracts/token/ERC20/ERC20.sol	d20d52b4be98738b8aa52b5bb0f88943f62128969b33d654fbca731539a7fe0a
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990
@openzeppelin/contracts/access/Ownable.sol	a8e4e1ae19d9bd3e8b0a6d46577eec098c01fbaffd3ec1252fd20d799e73393b

Findings Breakdown



Critical	0
Medium	0
Minor / Informative	3

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

RCT - Redundant Code Template

Criticality	Minor / Informative
Location	contracts/lib/LibCommon.sol contracts/ReflectiveV3ERC20.sol contracts/MeAIToken.sol
Status	Unresolved

Description

The contract is built using an `ERC20` template designed to support multiple configurations. Specifically, during deployment, the `configProps` variable holds the configuration properties for the token contract. However, the deployed contract is configured to utilize only the `_isBurnable` and `_isDocumentAllowed` properties, leaving many variables, events, and functions in the contract redundant and unused. This redundancy adds unnecessary complexity to the contract, increasing both the gas costs associated with deployment and the ongoing transaction costs for users.

```
// Example of redundant variable
uint256 private constant MAX_BPS_AMOUNT = 10_000;
// Example of redundant function
function setTaxConfig(
    address _taxAddress,
    uint256 _taxBPS
) external onlyOwner {/*…*/}
// Example of function that costs extra gas
function transfer(
    address to,
    uint256 amount
) public virtual override returns (bool) {/*…*/}
```

Recommendation

It is recommended to remove redundant code segments or utilize more streamlined templates, such as OpenZeppelin's `ERC20Burnable` extension, to ensure the contract remains efficient and easy to maintain.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	contracts/ReflectiveV3ERC20.sol#L48 contracts/MeAIToken.sol#L164,266,280,281,300
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.

```
function _tTotal() public view virtual returns (uint256) {
    return totalSupply();
  }
address _taxAddress
uint256 _feeBPS
uint256 _taxBPS
uint256 _deflationBPS
```

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/stable/style-guide.html#naming-conventions.

L17 - Usage of Solidity Assembly

Criticality	Minor / Informative
Location	contracts/lib/LibCommon.sol#L27,43,79,108
Status	Unresolved

Description

Using assembly can be useful for optimizing code, but it can also be error-prone. It's important to carefully test and debug assembly code to ensure that it is correct and does not contain any errors.

Some common types of errors that can occur when using assembly in Solidity include Syntax, Type, Out-of-bounds, Stack, and Revert.

```
assembly {
    ...
    if iszero(call(gas(), to, amount, 0, 0, 0, 0)) {
      ...
      mstore(0x00, 0xb12d13eb)
      revert(0x1c, 0x04)
    }
  }
  ...
  assembly {
    if iszero(shl(96, addr)) {
      ...
      mstore(0x00, 0xd92e233d)
      revert(0x1c, 0x04)
    }
  }
  ...
  assembly {
    let returndata_size := mload(data)
    revert(add(32, data), returndata_size)
  }
  ...
  assembly {
    let returndata_size := mload(data)
    revert(add(32, data), returndata_size)
  }
```

Recommendation

It is recommended to use assembly sparingly and only when necessary, as it can be difficult to read and understand compared to Solidity code.

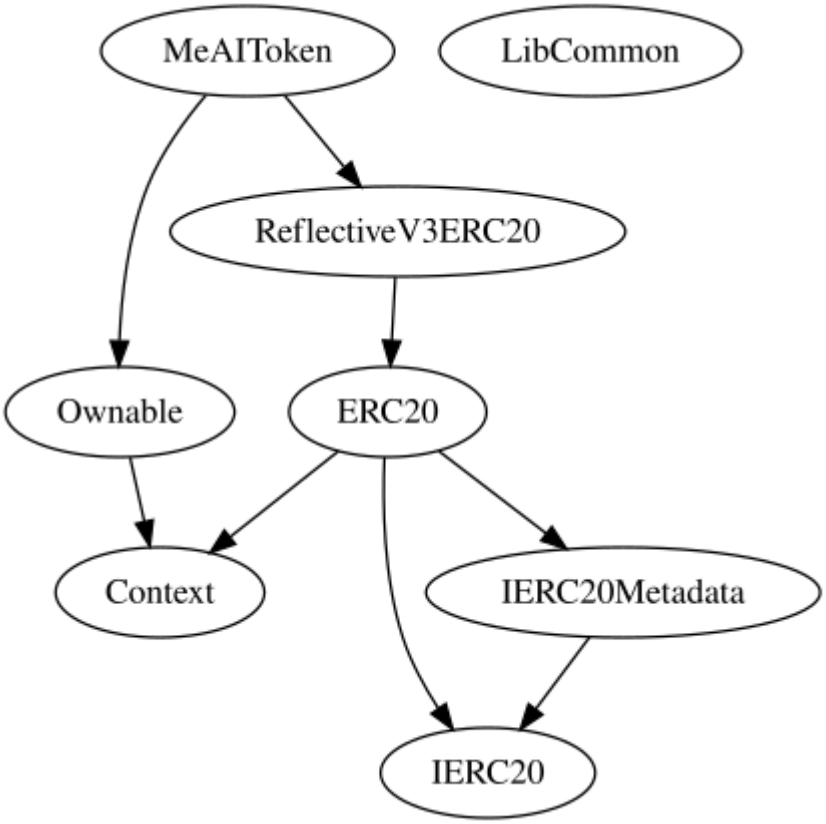
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
ReflectiveV3ERC20	Implementation	ERC20		
	_tTotal	Public		-
		Public	✓	ERC20
	balanceOf	Public		-
	transferFrom	Public	✓	-
	transfer	Public	✓	-
	_transfer	Internal	✓	
	_burn	Internal	✓	
	_setReflectionFee	Internal	✓	
	tokenFromReflection	Public		-
	_excludeFromRewards	Internal	✓	
	_includeInRewards	Internal	✓	
	_transferStandard	Private	✓	
	_transferFromExcluded	Private	✓	
	_transferToExcluded	Private	✓	
	_transferBothExcluded	Private	✓	
	_reflectFee	Private	✓	
	calculateFee	Private		
	_transferNonReflectedTax	Internal	✓	

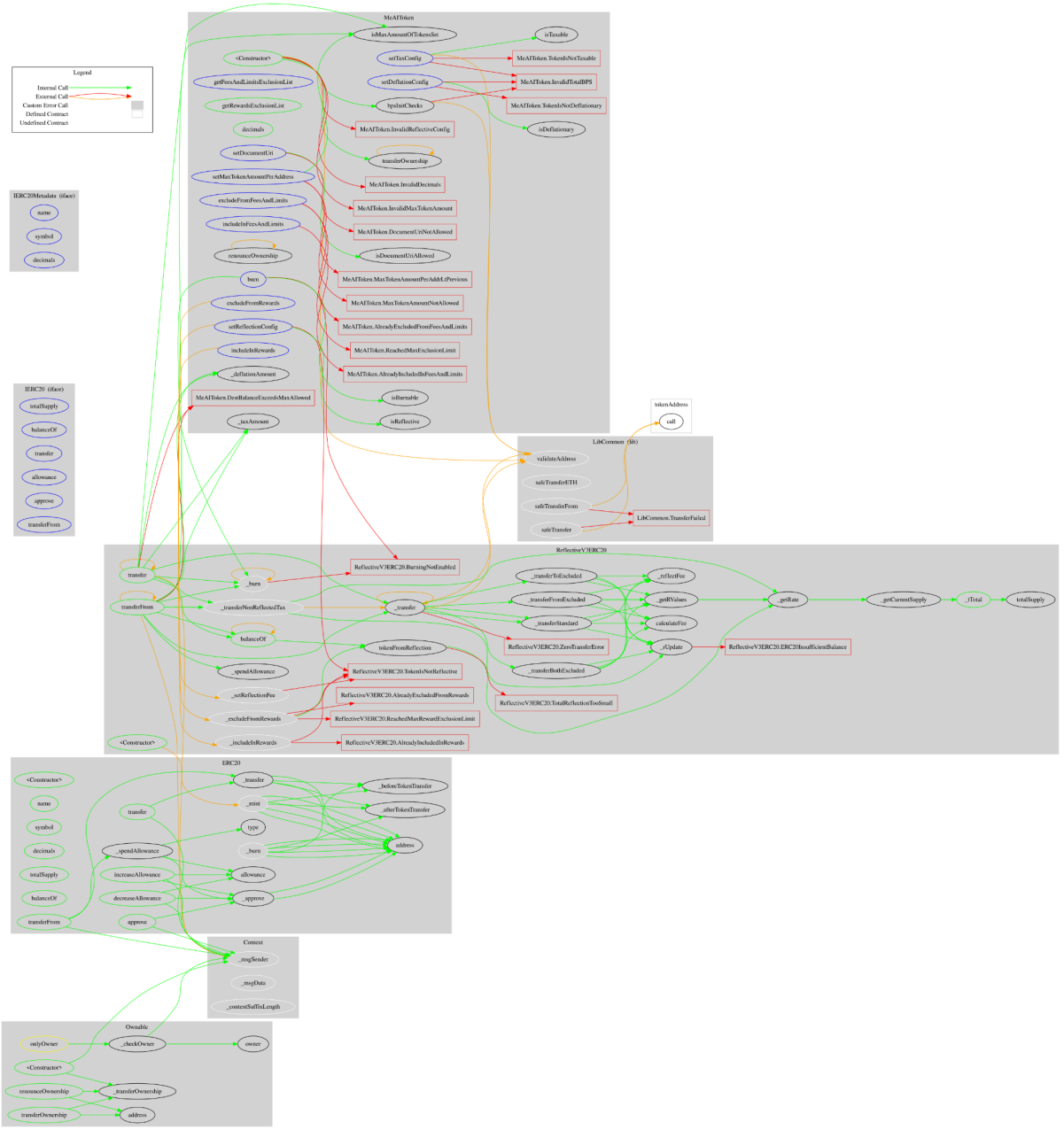
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_rUpdate	Private	✓	
MeAIToken	Implementation	ReflectiveV3 ERC20, Ownable		
		Public	✓	ReflectiveV3ERC20
	bpsInitChecks	Private		
	getFeesAndLimitsExclusionList	External		-
	isBurnable	Public		-
	getRewardsExclusionList	Public		-
	isMaxAmountOfTokensSet	Public		-
	isDocumentUriAllowed	Public		-
	decimals	Public		-
	isTaxable	Public		-
	isDeflationary	Public		-
	isReflective	Public		-
	setDocumentUri	External	✓	onlyOwner
	setMaxTokenAmountPerAddress	External	✓	onlyOwner
	setReflectionConfig	External	✓	onlyOwner
	setTaxConfig	External	✓	onlyOwner
	setDeflationConfig	External	✓	onlyOwner
	transfer	Public	✓	-
	transferFrom	Public	✓	-

	burn	External	✓	onlyOwner
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	excludeFromFeesAndLimits	External	✓	onlyOwner
	excludeFromRewards	External	✓	onlyOwner
	includeInFeesAndLimits	External	✓	onlyOwner
	includeInRewards	External	✓	onlyOwner
	_taxAmount	Internal		
	_deflationAmount	Internal		
LibCommon	Library			
	safeTransferETH	Internal	✓	
	validateAddress	Internal		
	safeTransferFrom	Internal	✓	
	safeTransfer	Internal	✓	

Inheritance Graph



Flow Graph


MeAIToken
isMaxAmountOfTokensSet
isTaxable
<Constructor>
setTaxConfig
MeAIToken.TokenIsNotTaxable
getFeesAndLimitsExclusionList
setDeflationConfig
MeAIToken.InvalidTotalBPS
getRewardsExclusionList
bpsInitChecks
MeAIToken.TokenIsNotDeflationary
decimals
isDeflationary
MeAIToken.InvalidReflectiveConfig
setDocumentUri
transferOwnership
setMaxTokenAmountPerAddress
MeAIToken.InvalidDecimals
excludeFromFeesAndLimits
MeAIToken.InvalidMaxTokenAmount
includeInFeesAndLimits
MeAIToken.DocumentUriNotAllowed
renounceOwnership
isDocumentUriAllowed
burn
MeAIToken.MaxTokenAmountPerAddrLtPrevious
excludeFromRewards
MeAIToken.MaxTokenAmountNotAllowed
setReflectionConfig
MeAIToken.AlreadyExcludedFromFeesAndLimits
includeInRewards
MeAIToken.ReachedMaxExclusionLimit
_deflationAmount
MeAIToken.AlreadyIncludedInFeesAndLimits
MeAIToken.DestBalanceExceedsMaxAllowed
isBurnable
_taxAmount
isReflective
Legend
Internal Call
External Call
Custom Error Call
Defined Contract
Undefined Contract
IERC20Metadata (iface)
name
symbol
decimals
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
tokenAddress
call
LibCommon (lib)
validateAddress
safeTransferETH
safeTransferFrom
LibCommon.TransferFailed
safeTransfer
ReflectiveV3ERC20
transfer
_burn
ReflectiveV3ERC20.BurningNotEnabled
_transferToExcluded
_reflectFee
transferFrom
_transferNonReflectedTax
_transfer
_transferFromExcluded
_getRValues
_getRate
_getCurrentSupply
_rTotal
totalSupply
balanceOf
_transferStandard
calculateFee
tokenFromReflection
ReflectiveV3ERC20.ZeroTransferError
_rUpdate
ReflectiveV3ERC20.ERC20InsufficientBalance
_spendAllowance
ReflectiveV3ERC20.TokenIsNotReflective
_transferBothExcluded
_setReflectionFee
ReflectiveV3ERC20.AlreadyExcludedFromRewards
ReflectiveV3ERC20.TotalReflectionTooSmall
_excludeFromRewards
ReflectiveV3ERC20.ReachedMaxRewardExclusionLimit
<Constructor>
_includeInRewards
ReflectiveV3ERC20.AlreadyIncludedInRewards
ERC20
<Constructor>
transfer
_beforeTokenTransfer
name
_mint
_afterTokenTransfer
transfer
symbol
type
decimals
_burn
address
_spendAllowance
totalSupply
increaseAllowance
allowance
balanceOf
decreaseAllowance
approve
transferFrom
approve
Context
_msgSender
_msgData
_contextSuffixLength
Ownable
onlyOwner
_checkOwner
owner
<Constructor>
renounceOwnership
_transferOwnership
transferOwnership
address

Summary

MeAI contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. MeAI is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues. The contract owner has renounced ownership. While the contract implements tax fees the configuration is such that they stay permanently zero.

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

cyberscope.io