Cyberscope

A TAC Security Company

Audit Report

TikTrix

August 2025

Network ETH

Address 0xDA5d03f441E13aD74876e5Fa1851cBfADAb1f332

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
●	ROF	Redundant Ownable Functionality	Unresolved
●	L04	Conformance to Solidity Naming Conventions	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	TrixToken
Compiler Version	v0.8.28+commit.7893614a
Optimization	200 runs
Explorer	https://etherscan.io/address/0xda5d03f441e13ad74876e5fa1851cbfadab1f332
Address	0xda5d03f441e13ad74876e5fa1851cbfadab1f332
Network	ETH
Symbol	TRIX
Decimals	18
Total Supply	2,000,000,000
Badge Eligibility	Yes

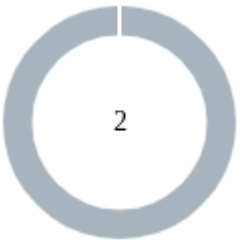
Audit Updates

Initial Audit	02 Jul 2025 https://github.com/cyberscope-io/audits/blob/main/1-trix/v1/audit.pdf
Corrected Phase 2	06 Aug 2025

Source Files

Filename	SHA256
contract/tiktrix-main-token/TrixToken.sol	8880f9f86d867876523b89584cc5bfd403b84e6a6fc7fd062732e779783c3b45

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	2	0	0	0

ROF - Redundant Ownable Functionality

Criticality	Minor / Informative
Location	contract/token/TrixToken.sol#L13
Status	Unresolved

Description

The contract inherits from the `Ownable` abstract contract to define an owner. In smart contracts, an owner typically has elevated privileges to execute administrative functions. However, in this case, while the contract defines an owner, it does not include any administrative functionalities. Therefore, the inheritance of `Ownable` is redundant.

```
constructor(
    string memory name_,
    string memory symbol_,
    uint256 initialSupply_,
    string memory contractURI_
) ERC20(name_, symbol_) Ownable(msg.sender) {
    uint256 supplyWithDecimals = initialSupply_ * 10 **
uint256(_decimals);
    _mint(msg.sender, supplyWithDecimals);
    _contractURI = contractURI_;
}
```

Recommendation

Eliminating redundancies will reduce code size and enhance readability. By removing the unnecessary inheritance, the contract becomes more efficient and aids in future maintainability.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	contract/token/TrixToken.sol#L14
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.

```
uint8 private constant _decimals = 18
```

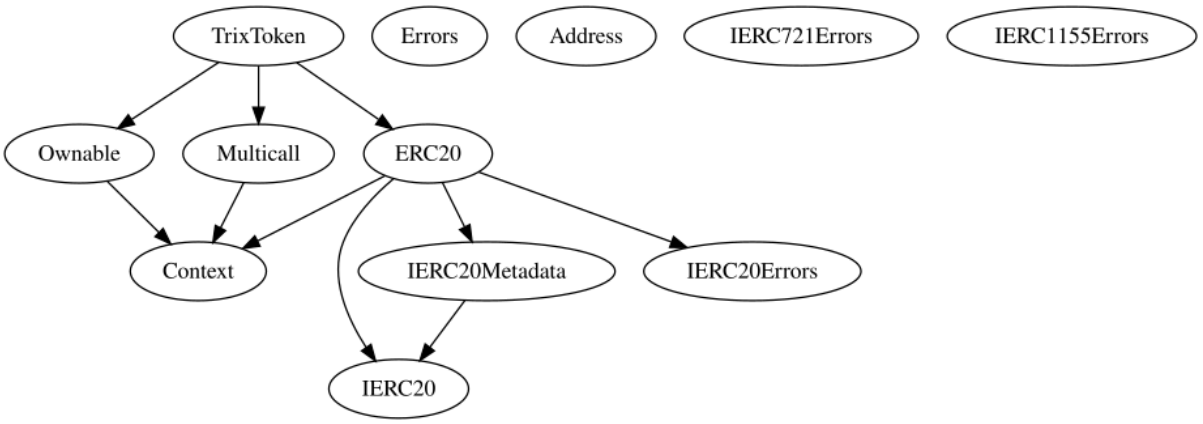
Recommendation

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.

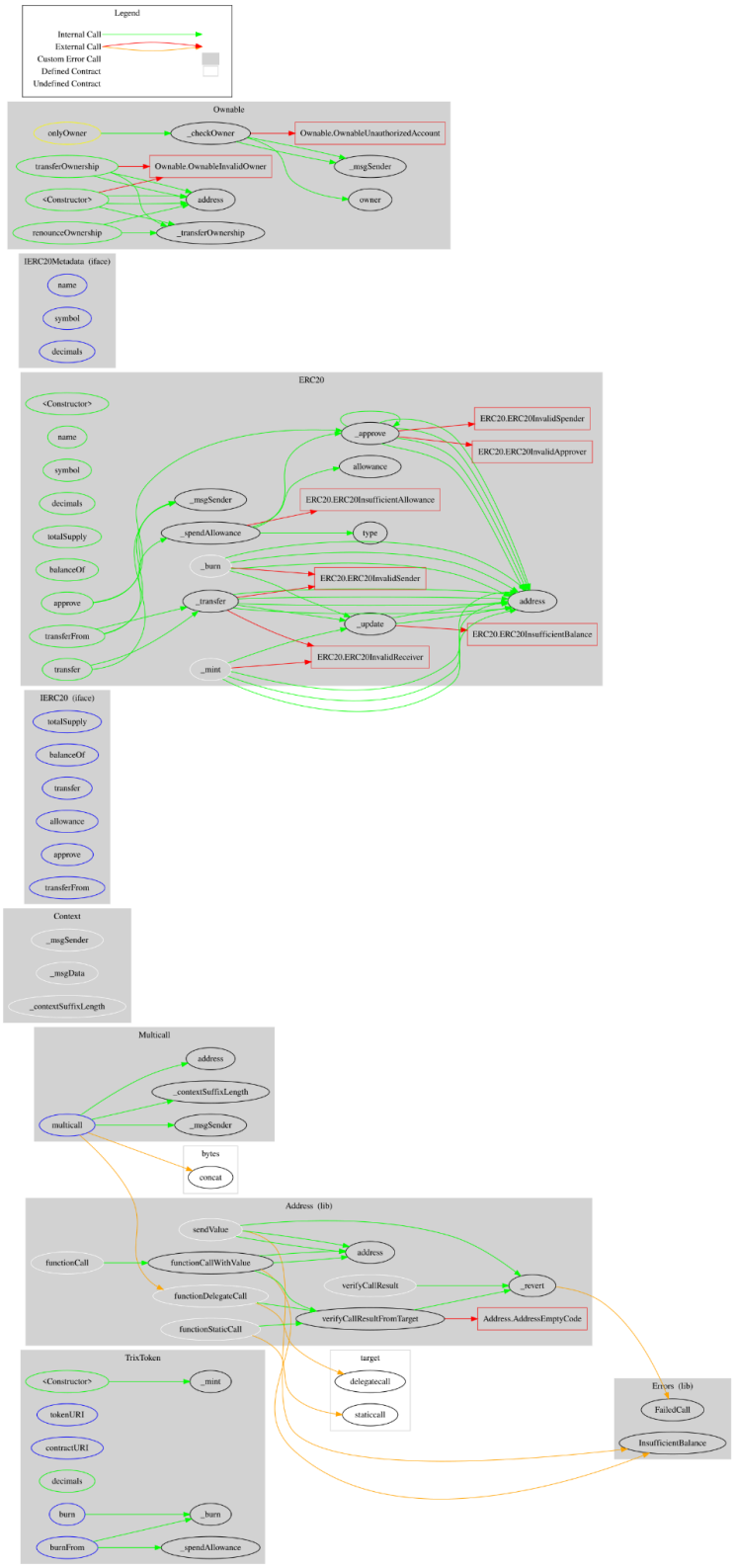
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
TrixToken	Implementation	ERC20, Ownable, Multicall		
		Public	✓	ERC20 Ownable
	tokenURI	External		-
	contractURI	External		-
	decimals	Public		-
	burn	External	✓	-
	burnFrom	External	✓	-

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Custom Error Call
Defined Contract
Undefined Contract
Ownable
onlyOwner
_checkOwner
Ownable.OwnableUnauthorizedAccount
transferOwnership
Ownable.OwnableInvalidOwner
_msgSender
<Constructor>
address
owner
renounceOwnership
_transferOwnership
IERC20Metadata (iface)
name
symbol
decimals
ERC20
<Constructor>
name
symbol
decimals
totalSupply
balanceOf
approve
transferFrom
transfer
_approve
ERC20.ERC20InvalidSpender
ERC20.ERC20InvalidApprover
allowance
_msgSender
ERC20.ERC20InsufficientAllowance
_spendAllowance
type
_burn
ERC20.ERC20InvalidSender
_transfer
address
_update
ERC20.ERC20InsufficientBalance
ERC20.ERC20InvalidReceiver
_mint
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
Context
_msgSender
_msgData
_contextSuffixLength
Multicall
address
_contextSuffixLength
multicall
_msgSender
bytes
concat
Address (lib)
sendValue
functionCall
functionCallWithValue
address
verifyCallResult
_revert
functionDelegateCall
verifyCallResultFromTarget
Address.AddressEmptyCode
functionStaticCall
target
delegatecall
staticcall
Errors (lib)
FailedCall
InsufficientBalance
TrixToken
<Constructor>
_mint
tokenURI
contractURI
decimals
burn
_burn
burnFrom
_spendAllowance

Summary

TikTrix contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. TikTrix is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues.

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 TAC 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.

A **TAC Security** Company

The Cyberscope team

cyberscope.io