Cyberscope

Audit Report

Tavarus Blackmon Art

June 2024

Network ETH

Address 0x606F55Eb7ca470B6D551a356b1EBA6b7CAe86d2e

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	Unresolved
●	MT	Mints Tokens	Unresolved
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
Medium	US	Untrusted Source	Unresolved
Minor	IDI	Immutable Declaration Improvement	Unresolved
Minor	MEE	Missing Events Emission	Unresolved
Minor	L04	Conformance to Solidity Naming Conventions	Unresolved
Minor	L16	Validate Variable Setters	Unresolved

Table of Contents

Review

Contract Name	TokenFiERC20
Compiler Version	v0.8.23+commit.f704f362
Optimization	10 runs
Explorer	https://etherscan.io/address/0x606f55eb7ca470b6d551a356b1eba6b7cae86d2e
Address	0x606f55eb7ca470b6d551a356b1eba6b7cae86d2e
Network	ETH
Symbol	TAV
Decimals	18
Total Supply	250,000,000
Badge Eligibility	Must Fix Criticals

Audit Updates

Initial Audit	07 Jun 2024

Source Files

Filename	SHA256
contracts/token-launcher/templates/TokenFiERC20.sol	034acc140d2b9f812e17916f7e2606452d9e34495b79db96f3f40e81857d7436
contracts/token-launcher/interfaces/ITokenLauncherLiquidityPoolFactory.sol	165aee4a810df42e398b39050f8d1642171fe006e70d688541773eafefbaa85e

contracts/token-launcher/interfaces/ITokenLauncherERC20.sol	8b6ff21b2248451fb748759dc8e614830e481abdef0dd84fc5697b054b78b8b4
contracts/token-launcher/interfaces/ITokenLauncherCommon.sol	9f0b7efece6778400a5500175d5220224d6083dfba39b535d7039f97eccc5dd9
contracts/token-launcher/interfaces/IBuyBackHandler.sol	0f060dedb8c93644370867ee44957c036045372c1a4efeca265b9f315c74a49a
@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router02.sol	a2900701961cb0b6152fc073856b972564f7c798797a4a044e83d2ab8f0e8d38
@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router01.sol	0439ffe0fd4a5e1f4e22d71ddbda76d63d61679947d158cba4ee0a1da60cf663
@openzeppelin/contracts/utils/Strings.sol	cb2df477077a5963ab50a52768cb74ec6f32177177a78611ddbbe2c07e2d36de
@openzeppelin/contracts/utils/Context.sol	b2cfee351bcafd0f8f27c72d76c054df9b571b62cfac4781ed12c86354e2a56c
@openzeppelin/contracts/utils/Address.sol	8b85a2463eda119c2f42c34fa3d942b61aee65df381f48ed436fe8edb3a7d602
@openzeppelin/contracts/utils/structs/EnumerableSet.sol	a64e5d0e83019d9caa51e6fe6f68ac54b583ac15792b8557cb8e4fab20711b9f
@openzeppelin/contracts/utils/math/SignedMath.sol	420a5a5d8d94611a04b39d6cf5f02492552ed4257ea82aba3c765b1ad52f77f6
@openzeppelin/contracts/utils/math/Math.sol	85a2caf3bd06579fb55236398c1321e15fd524a8fe140dff748c0f73d7a52345
@openzeppelin/contracts/utils/introspection/IERC165.sol	701e025d13ec6be09ae892eb029cd83b3064325801d73654847a5fb11c58b1e5
@openzeppelin/contracts/utils/introspection/ERC165.sol	8806a632d7b656cadb8133ff8f2acae4405b3a64d8709d93b0fa6a216a8a6154
@openzeppelin/contracts/token/ERC20/IERC20.sol	7ebde70853ccafcf1876900dad458f46eb9444d591d39bfc58e952e2582f5587

@openzeppelin/contracts/token/ERC20/ERC20.sol	d20d52b4be98738b8aa52b5bb0f88943f62128969b33d654fbca731539a7fe0a
@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol	82dc918d8df553e2461b96595580e565424b407d73dab8a7ce4bde479810fb2a
@openzeppelin/contracts/token/ERC20/extensions/IERC20Permit.sol	bbf4d26d660ce8f9c9887af3a46bf4eab5f4c15eb8413ad89b65ecdd05dee7a9
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990
@openzeppelin/contracts/access/IAccessControl.sol	d03c1257f2094da6c86efa7aa09c1c07ebd33dd31046480c5097bc2542140e45
@openzeppelin/contracts/access/AccessControl.sol	afd98330d27bddff0db7cb8fcf42bd4766dda5f60b40871a3bec6220f9c9edf7

Findings Breakdown

Critical	3
Medium	1
Minor / Informative	4

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	3	0	0	0
Medium	1	0	0	0
Minor / Informative	4	0	0	0

ST - Stops Transactions

Criticality	Critical
Location	contracts/token-launcher/templates/TokenFiERC20.sol#L470
Status	Unresolved

Description

As described in detail in the `US` finding, the contract owner and the addresses that have the `DEFAULT_ADMIN_ROLE` have the authority to stop transactions.

Recommendation

It is recommended to implement the recommendation of the `US` finding, in order to mitigate the risk of the contract operating as a honeypot.

ELFM - Exceeds Fees Limit

Criticality	Critical
Location	contracts/token-launcher/templates/TokenFiERC20.sol#L129
Status	Unresolved

Description

The addresses that are granted the `DEFAULT_ADMIN_ROLE` have the authority to increase over the allowed limit of 25%. They may take advantage of it by calling the `updateFees` function with a high percentage value.

```
function updateFees(ITokenLauncherERC20.Fees memory _fees)
external onlyRole(DEFAULT_ADMIN_ROLE) {
    if (isReflectionToken) {
        require(_fees.reflection.percentage > 0, "TokenFiERC20:
reflection percentage must be non-zero");
    } else {
        require(_fees.reflection.percentage == 0,
"TokenFiERC20: reflection percentage must be zero");
    }
    uint256 maxFee = _fees.transferFee.percentage +
_fees.burn.percentage + _fees.reflection.percentage +
_fees.buyback.percentage;
    require(maxFee <= MULTIPLIER_BASIS, "TokenFiERC20: fees sum
must be less than 100%");
    fees = _fees;
}
```

Recommendation

The contract could embody a check for the maximum acceptable value. The team should carefully manage the private keys of the owner's account and of the addresses that have `DEFAULT_ADMIN_ROLE` . 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 and revoke the `DEFAULT_ADMIN_ROLE` , which will eliminate the threats but it is non-reversible.

MT - Mints Tokens

Criticality	Critical
Location	contracts/token-launcher/templates/TokenFiERC20.sol#L461
Status	Unresolved

Description

The contract owner and the addresses that are granted the `DEFAULT_ADMIN_ROLE` have the authority to mint tokens, up to the `maxSupply` , which is `500,000,000` . The current total supply is `250,000,000` , so an additional of `250,000,000` tokens can be minted. They 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(DEFAULT_ADMIN_ROLE) {
        require(totalSupply() + amount <= maxSupply,
"TokenFiERC20: max supply exceeded");
        if (isReflectionToken) {
            _mintReflection(to, amount);
        } else {
            super._mint(to, amount);
        }
    }
```

Recommendation

The team should carefully manage the private keys of the owner's account and of the addresses that have the `DEFAULT_ADMIN_ROLE`. 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 and revoke the `DEFAULT_ADMIN_ROLE`, which will eliminate the threats but it is non-reversible.

US - Untrusted Source

Criticality	Medium
Location	contracts/token-launcher/templates/TokenFiERC20.sol#L386,470
Status	Unresolved

Description

The contract uses an external contract in order to determine the transaction's flow. The external contract is untrusted. As a result, it may produce security issues and harm the transactions.

```
if (!_exchangePools.contains(sender) && buybackDetails.router
!= address(0)) {
        IBuyBackHandler(buybackHandler).buyback(treasury,
buybackDetails);
    }

...

function updateTokenLauncher(address _newTokenLauncher)
external onlyRole(DEFAULT_ADMIN_ROLE) {
    _revokeRole(FEE_MANAGER_ROLE, tokenLauncher);
    tokenLauncher = _newTokenLauncher;

    _grantRole(FEE_MANAGER_ROLE, _newTokenLauncher);
    emit TokenLauncherUpdated(_newTokenLauncher);
}
```

Recommendation

The contract should use a trusted external source. A trusted source could be either a commonly recognized or an audited contract. The pointing addresses should not be able to change after the initialization.

IDI - Immutable Declaration Improvement

Criticality	Minor / Informative
Location	contracts/token-launcher/templates/TokenFiERC20.sol#L66,67,69,74,82
Status	Unresolved

Description

The contract declares state variables that their value is initialized once in the constructor and are not modified afterwards. The `immutable` is a special declaration for this kind of state variables that saves gas when it is defined.

```
treasury
maxSupply
_decimals
buybackHandler
isReflectionToken
```

Recommendation

By declaring a variable as immutable, the Solidity compiler is able to make certain optimizations. This can reduce the amount of storage and computation required by the contract, and make it more gas-efficient.

MEE - Missing Events Emission

Criticality	Minor / Informative
Location	contracts/token-launcher/templates/TokenFiERC20.sol#L129
Status	Unresolved

Description

The contract performs actions and state mutations from external methods that do not result in the emission of events. Emitting events for significant actions is important as it allows external parties, such as wallets or dApps, to track and monitor the activity on the contract. Without these events, it may be difficult for external parties to accurately determine the current state of the contract.

```
function updateFees(ITokenLauncherERC20.Fees memory _fees)
external onlyRole(DEFAULT_ADMIN_ROLE) {
    if (isReflectionToken) {
        require(_fees.reflection.percentage > 0, "TokenFiERC20:
reflection percentage must be non-zero");
    } else {
        require(_fees.reflection.percentage == 0,
"TokenFiERC20: reflection percentage must be zero");
    }
    uint256 maxFee = _fees.transferFee.percentage +
_fees.burn.percentage + _fees.reflection.percentage +
_fees.buyback.percentage;
    require(maxFee <= MULTIPLIER_BASIS, "TokenFiERC20: fees sum
must be less than 100%");
    fees = _fees;
}
```

Recommendation

It is recommended to include events in the code that are triggered each time a significant action is taking place within the contract. These events should include relevant details such as the user's address and the nature of the action taken. By doing so, the contract will be more transparent and easily auditable by external parties. It will also help prevent potential issues or disputes that may arise in the future.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	contracts/token-launcher/templates/TokenFiERC20.sol#L101,129,470
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.

```
ITokenLauncherLiquidityPoolFactory.BuyBackDetails memory
_buybackDetails
ITokenLauncherERC20.Fees memory _fees
address _newTokenLauncher
```

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.

L16 - Validate Variable Setters

Criticality	Minor / Informative
Location	contracts/token-launcher/templates/TokenFiERC20.sol#L472
Status	Unresolved

Description

The contract performs operations on variables that have been configured on user-supplied input. These variables are missing of proper check for the case where a value is zero. This can lead to problems when the contract is executed, as certain actions may not be properly handled when the value is zero.

```
tokenLauncher = _newTokenLauncher
```

Recommendation

By adding the proper check, the contract will not allow the variables to be configured with zero value. This will ensure that the contract can handle all possible input values and avoid unexpected behavior or errors. Hence, it can help to prevent the contract from being exploited or operating unexpectedly.

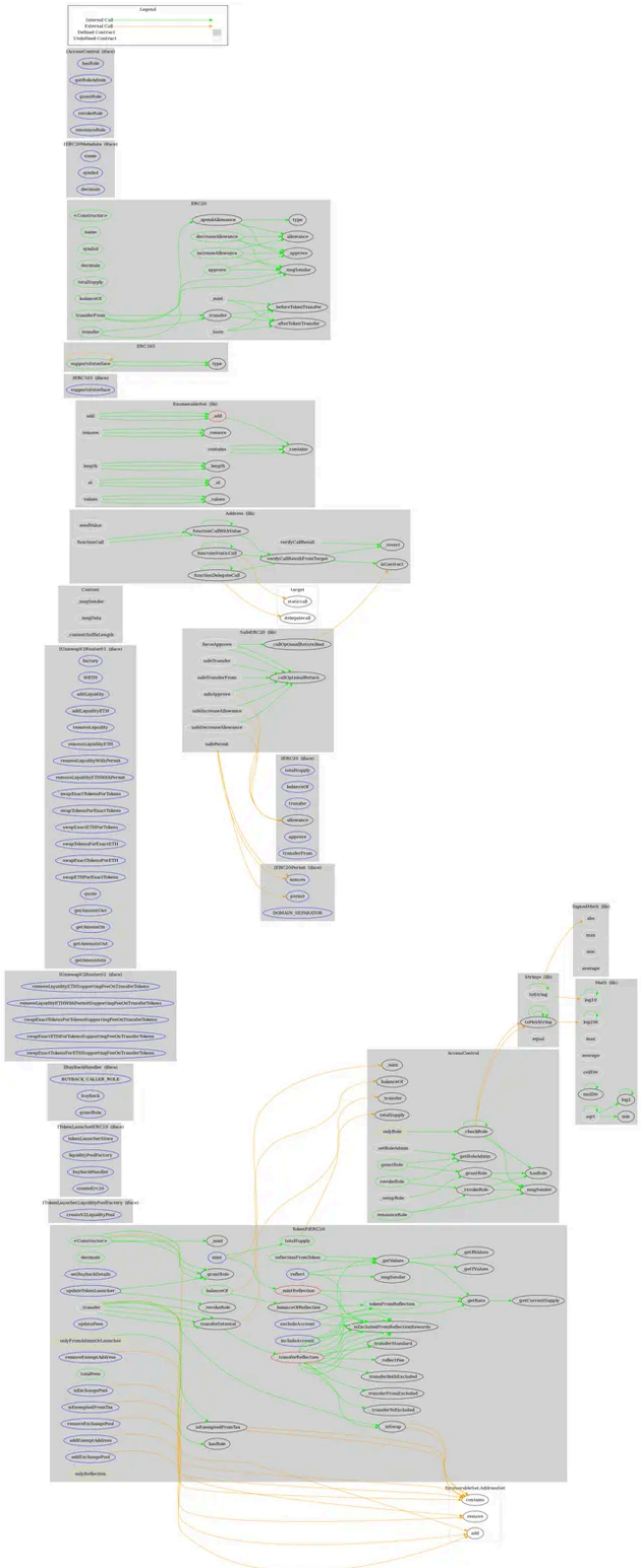
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
TokenFiERC20	Implementation	ERC20, AccessControl		
		Public	✓	ERC20
	decimals	Public		-
	setBuybackDetails	External	✓	onlyRole
	addExchangePool	External	✓	onlyRole
	addExemptAddress	External	✓	onlyRole
	updateFees	External	✓	onlyRole
	isExemptedFromTax	External		-
	isExchangePool	External		-
	balanceOf	Public		-
	totalSupply	Public		-
	isExcludedFromReflectionRewards	Public		-
	reflect	External	✓	onlyReflection
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-
	excludeAccount	External	✓	onlyReflection onlyFromAdminOrLauncher
	includeAccount	External	✓	onlyReflection onlyFromAdminOrLauncher

	totalFees	Public		-
	_balanceOfReflection	Private		
	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	
	_transferBothExcluded	Private	✓	
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	removeExchangePool	External	✓	onlyRole
	removeExemptAddress	External	✓	onlyRole
	_transfer	Internal	✓	
	_transferReflection	Private	✓	
	_transferInternal	Private	✓	
	_isSwap	Internal		
	_isExemptedFromTax	Internal		
	_mintReflection	Private	✓	
	mint	External	✓	onlyRole
	updateTokenLauncher	External	✓	onlyRole

Inheritance Graph

Flow Graph



Summary

Tavarus Blackmon Art 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, manipulate the fees and mint tokens. if the contract owner abuses the mint functionality, then the contract will be highly inflated. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership and revoking the roles 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