Cyberscope

A TAC Security Company

Audit Report

Luminex DeFi Finance

February 2026

Network	BSC
Address	0x313e3ecc0dcbfd09d799572e576fcf81227c2b2f
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
	MC	Missing Check	Acknowledged
	NWES	Nonconformity with ERC-20 Standard	Acknowledged
	L19	Stable Compiler Version	Acknowledged

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	LuminexDefiFinance
Compiler Version	v0.8.28+commit.7893614a
Optimization	200 runs
Explorer	https://bscscan.com/address/0x313e3ecc0dcbfd09d799572e576fcf81227c2b2f
Address	0x313e3ecc0dcbfd09d799572e576fcf81227c2b2f
Network	BSC
Symbol	LFUSD
Decimals	18
Total Supply	49.500.000.000

Audit Updates

Initial Audit	02 Feb 2026

Source Files

Filename	SHA256
LFUSDcoin.sol	d0935f2fa6a4f61c924c5eb57d87dcd10d15a0221d9a10635adcf2fc898fc8eb

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	0	3	0	0

MC - Missing Check

Criticality	Minor / Informative
Location	LFUSDcoin.sol#L90,109,115
Status	Acknowledged

Description

The contract is processing variables that have not been properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues.

Specifically the `approve` function does not ensure that the updated allowance does not belong to address(0). The same applies for `increaseAllowance` and `decreaseAllowance` .

```
Shell
function approve(address spender, uint256 amount)
external returns (bool) {
    allowance[msg.sender][spender] = amount;
    emit Approval(msg.sender, spender, amount);
    return true;
}

function increaseAllowance(address spender,
uint256 addedValue) external returns (bool)

function decreaseAllowance(address spender,
uint256 subtractedValue) external returns (bool)
```

Recommendation

The team is advised to properly check the variables according to the required specifications.

NWES - Nonconformity with ERC-20 Standard

Criticality	Minor / Informative
Location	LFUSDcoin.sol#L42
Status	Acknowledged

Description

The contract does not fully conform to the ERC20 Standard. Specifically, according to the standard, the `totalSupply` should have an explicitly implemented totalSupply() function rather than declaring a `totalSupply` public variable. This discrepancy between the contract's implementation and the ERC20 standard may lead to inconsistencies and incompatibilities with other contracts.

```
Shell
uint256 public totalSupply;
```

Recommendation

The incorrect implementation of the ERC20 standard could potentially lead to problems when interacting with the contract, as other contracts or applications that expect the ERC20 interface may not behave as expected. The team is advised to review and revise the implementation of the `totalSupply` mechanism to ensure full compliance with the ERC20 standard.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	LFUSDcoin.sol#L7
Status	Acknowledged

Description

The `^` symbol indicates that any version of Solidity that is compatible with the specified version (i.e., any version that is a higher minor or patch version) can be used to compile the contract. The version lock is a mechanism that allows the author to specify a minimum version of the Solidity compiler that must be used to compile the contract code. This is useful because it ensures that the contract will be compiled using a version of the compiler that is known to be compatible with the code.

```shell
pragma solidity ^0.8.20;
```

Recommendation

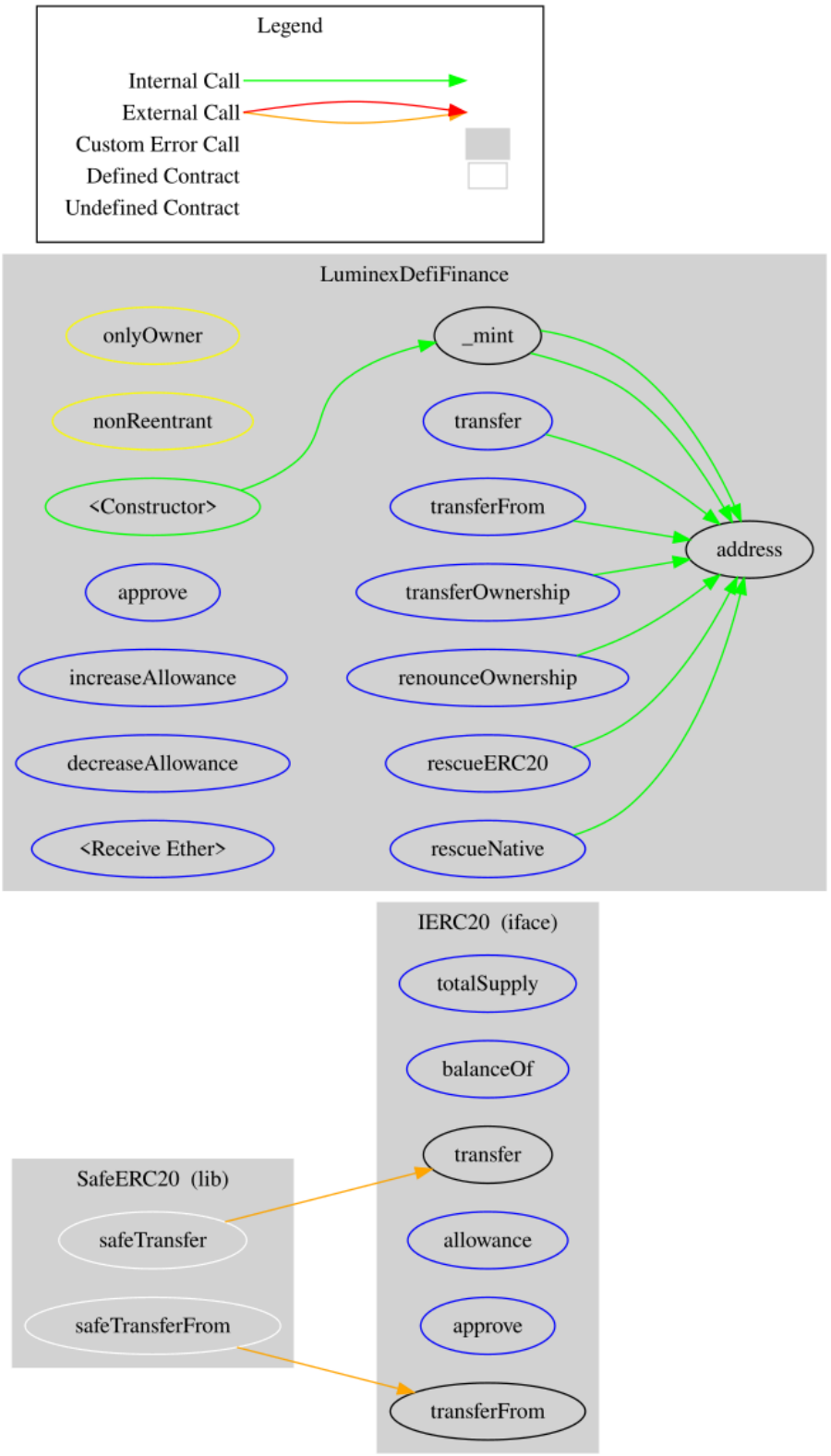
The team is advised to lock the pragma to ensure the stability of the codebase. The locked pragma version ensures that the contract will not be deployed with an unexpected version. An unexpected version may produce vulnerabilities and undiscovered bugs. The compiler should be configured to the lowest version that provides all the required functionality for the codebase. As a result, the project will be compiled in a well-tested LTS (Long Term Support) environment.

Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
LuminexDefiFinance	Implementation			
		Public	✓	-
	transfer	External	✓	-
	approve	External	✓	-
	transferFrom	External	✓	-
	increaseAllowance	External	✓	-
	decreaseAllowance	External	✓	-
	_mint	Internal	✓	
	transferOwnership	External	✓	onlyOwner
	renounceOwnership	External	✓	onlyOwner
	rescueERC20	External	✓	onlyOwner nonReentrant
	rescueNative	External	✓	onlyOwner nonReentrant
		External	Payable	-

Inheritance Graph

Flow Graph


Legend
Internal Call
External Call
Custom Error Call
Defined Contract
Undefined Contract
LuminexDefiFinance
onlyOwner
nonReentrant
<Constructor>
approve
increaseAllowance
decreaseAllowance
<Receive Ether>
_mint
transfer
transferFrom
transferOwnership
renounceOwnership
rescueERC20
rescueNative
address
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
SafeERC20 (lib)
safeTransfer
safeTransferFrom

Summary

Luminex DeFi Finance contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. Luminex DeFi Finance 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 can access some admin functions that can not be used in a malicious way to disturb the users' transactions.

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