Cyberscope

A TAC Security Company

Audit Report

Libertas Token

December 2025

Network BSC_TESTNET

Address 0x211cd4c8079a599170fcc9431aa22854959e03f8

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 Ownership Functionality	Unresolved
●	DSV	Deprecated Solidity Version	Unresolved
●	L09	Dead Code Elimination	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	Libertas
Compiler Version	v0.5.16+commit.9c3226ce
Optimization	200 runs
Explorer	https://testnet.bscscan.com/token/0x211cD4c8079a599170fcc9431aA22854959E03f8#code
Address	0xe0cd64409e3128c6cf48d428f8f53f966476fcdf
Network	BSC
Symbol	LBC
Decimals	8
Total Supply	1,000,000,000

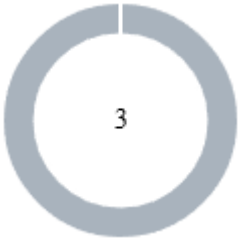
Audit Updates

Initial Audit	28 Oct 2025 https://github.com/cyberscope-io/audits/blob/main/3-lbc/v1/audit.pdf
Corrected Phase 2	29 Dec 2025

Source Files

Filename	SHA256
lbc.sol	674d992387ff758fe2bae2dac1d850236d8c7865a86507594fccf6666d653fe1

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

ROF - Redundant Ownership Functionality

Criticality	Minor / Informative
Location	lbc.sol#L288
Status	Unresolved

Description

The contract inherits an Ownable module. However, ownership is not used to control or configure any token-related functionality. The owner is only able to transfer or renounce ownership and has no privileges affecting token supply, transfers, allowances, or parameters. As a result, ownership does not provide any functional control over the contract's behavior and may be considered unnecessary.

```
Shell

contract Ownable is Context {
  address private _owner;
```

Recommendation

The team is advised to remove the ownable functionality from the contract to enhance code optimization and readability.

DSV - Deprecated Solidity Version

Criticality	Minor / Informative
Location	lbc.sol#L9
Status	Unresolved

Description

The contracts are written using Solidity version 0.5.16, which is deprecated and no longer maintained by the Solidity development team. Relying on outdated compiler versions can expose the codebase to known vulnerabilities, unresolved bugs, and missed performance improvements. This may compromise the security, efficiency, and long-term maintainability of the contracts.

```
Shell

pragma solidity 0.5.16;
```

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.

L09 - Dead Code Elimination

Criticality	Minor / Informative
Location	lbc.sol#L538,557,592
Status	Unresolved

Description

In Solidity, dead code is code that is written in the contract, but is never executed or reached during normal contract execution. Dead code can occur for a variety of reasons, such as:

- Conditional statements that are always false.
- Functions that are never called.
- Unreachable code (e.g., code that follows a return statement).

Dead code can make a contract more difficult to understand and maintain, and can also increase the size of the contract and the cost of deploying and interacting with it.

```
Shell
function _mint(address account, uint256 amount)
internal {
...
function _burn(address account, uint256 amount)
internal {

function _burnFrom(address account, uint256
amount) internal {

...
```

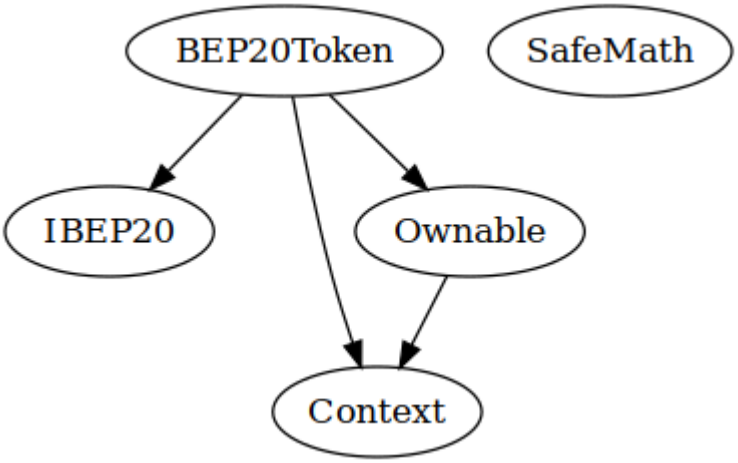
Recommendation

To avoid creating dead code, it's important to carefully consider the logic and flow of the contract and to remove any code that is not needed or that is never executed. This can help improve the clarity and efficiency of the contract.

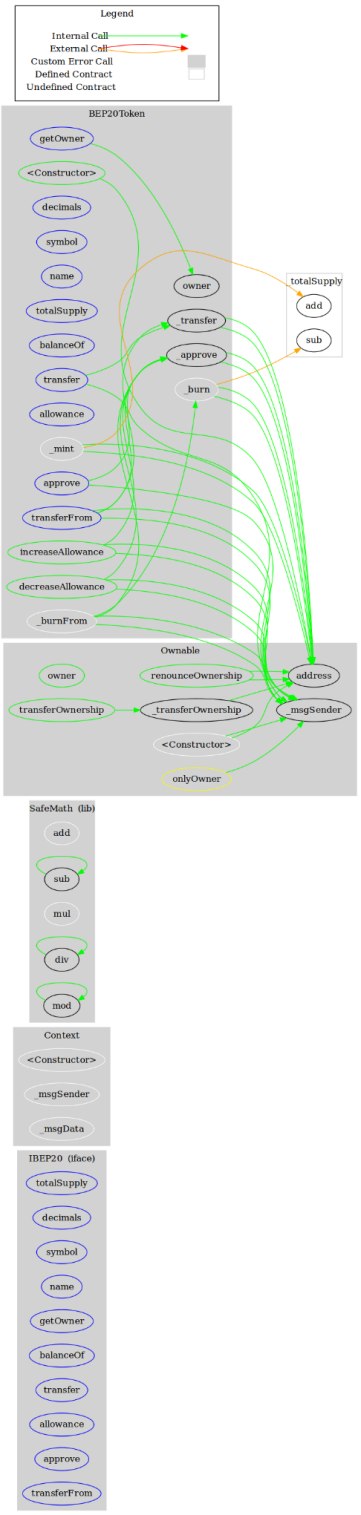
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
BEP20Token	Implementation	Context, IBEP20, Ownable		
		Public	✓	-
	getOwner	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_burnFrom	Internal	✓	

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Custom Error Call
Defined Contract
Undefined Contract
BEP20Token
getOwner
<Constructor>
decimals
symbol
name
totalSupply
balanceOf
transfer
allowance
_mint
approve
transferFrom
increaseAllowance
decreaseAllowance
_burnFrom
owner
transfer
approve
_burn
totalSupply
add
sub
Ownable
owner
transferOwnership
renounceOwnership
_transferOwnership
<Constructor>
onlyOwner
address
_msgSender
SafeMath (lib)
add
sub
mul
div
mod
Context
<Constructor>
_msgSender
_msgData
IBEP20 (iface)
totalSupply
decimals
symbol
name
getOwner
balanceOf
transfer
allowance
approve
transferFrom

Summary

Libertas contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. Libertas 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