Cyberscope

A TAC Security Company

Audit Report

Bitcoin Limited Edition

September 2025

Network	BSC
Address	0x9d2144328e1d618f54cd38540f5ee50671f6a208
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
●	L19	Stable Compiler Version	Unresolved
●	L16	Validate Variable Setters	Unresolved

Table of Contents

Analysis 1
Diagnostics 2
Table of Contents 3
Risk Classification 4
Review 5
Audit Updates 5
Source Files 5
Findings Breakdown 7
ROF - Redundant Ownable Functionality 8
Description 8
Recommendation 8
L16 - Validate Variable Setters 9
Description 9
Recommendation 9
L19 - Stable Compiler Version 10
Description 10
Recommendation 10
Functions Analysis 11
Summary 12
Disclaimer 13
About Cyberscope 14

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	BTCLEToken
Compiler Version	v0.8.19+commit.7dd6d404
Optimization	200 runs
Explorer	https://bscscan.com/address/0x9d2144328e1d618f54cd38540f5ee50671f6a208
Address	0x9d2144328e1d618f54cd38540f5ee50671f6a208
Network	BSC
Symbol	BTCLE
Decimals	18
Total Supply	210,000

Audit Updates

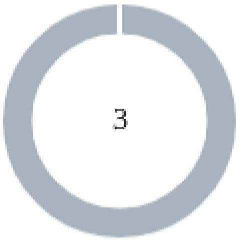
Initial Audit	25 Sep 2025

Source Files

Filename	SHA256
BTCLEToken.sol	ca9ed31bbdcefb85230c005725de3e84a08166c2ca36e2f8c82c94c34543416c
libraries/BEP20Base.sol	2c4f94b4ce861b0b1207040117a7314a5863879b831cc3c5a18b7c1abf1b549f
libraries/BEP20.sol	44b0815befb3829ed77173890ded47d08ba9eb586bf86a8add1f3a863cfeb5fb

interfaces/IBEP20.sol	f313a08143bc9ce32d8966602ba46fcf1a68668372dd3928aeab8341ff2eeb87

Findings Breakdown



●	Critical	0
●	Medium	0
●	Minor / Informative	3

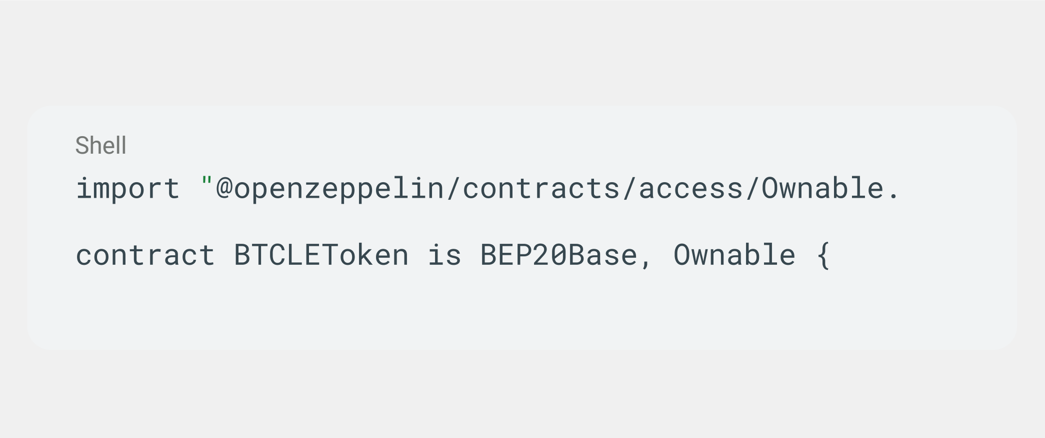
Severity		Unresolved	Acknowledged	Resolved	Other
●	Critical	0	0	0	0
●	Medium	0	0	0	0
●	Minor / Informative	3	0	0	0

ROF - Redundant Ownable Functionality

Criticality	Minor / Informative
Location	BTCLEToken.sol#L3,11
Status	Unresolved

Description

The BTCLEToken contract inherits from the Ownable contract. This contract is typically used to implement access control by designating an owner account with exclusive privileges for executing restricted functions. However, in the current implementation, none of the contract's functions utilize onlyOwner or any ownership-related logic. As a result, the inheritance of Ownable is redundant and introduces unnecessary code complexity.



Shell

```
import "@openzeppelin/contracts/access/Ownable.

contract BTCLEToken is BEP20Base, Ownable {
```

Recommendation

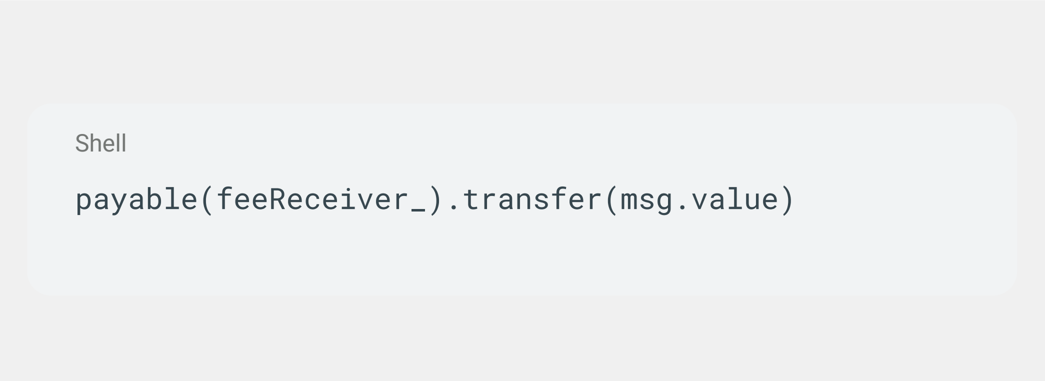
It is recommended to remove the unused Ownable inheritance to eliminate redundancy, improve code clarity, and reduce the overall contract size. This will enhance readability, maintainability, and gas efficiency.

L16 - Validate Variable Setters

Criticality	Minor / Informative
Location	BTCLEToken.sol#L17
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.



Shell

```
payable(feeReceiver_).transfer(msg.value)
```

Recommendation

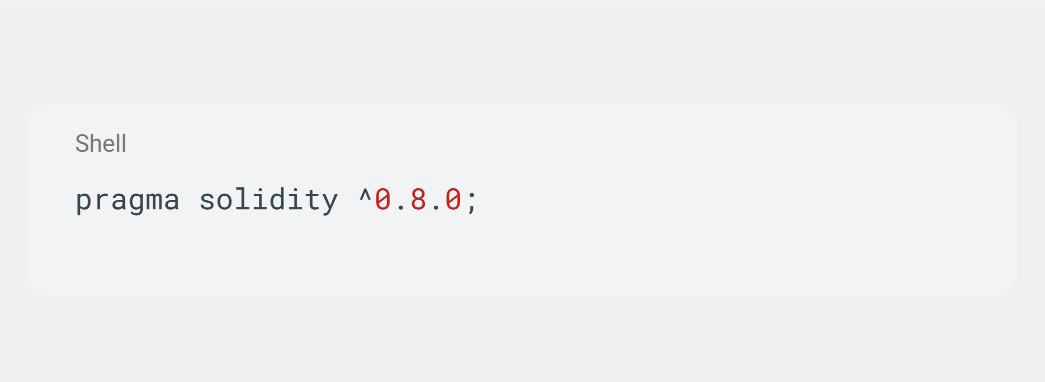
By adding the proper check, the contract will not allow the variables to be configured with zero value. This will ensure that the contract can handle all possible input values and avoid unexpected behavior or errors. Hence, it can help to prevent the contract from being exploited or operating unexpectedly.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	BTCLEToken.sol#L3
Status	Unresolved

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 ^0.8.0;
```

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
BTCLEToken	Implementation	BEP20Base, Ownable		
		Public	Payable	BEP20Base
BEP20Base	Implementation	BEP20		
		Public	✓	BEP20
	decimals	Public		-
BEP20	Implementation	ERC20, IBEP20		
		Public	✓	ERC20
	_beforeTokenTransfer	Internal	✓	
	_transfer	Internal	✓	
	_mint	Internal	✓	
IBEP20	Interface	IERC20		

Summary

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