Cyberscope

Audit Report

CloudBTC

February 2024

SHA256 2d014f6b946b02a6c30a4d7c6e4b966921f8ce9681b43b47d6b49359be94e031

Audited by © cyberscope

Analysis

Critical Medium Minor / Informative Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Acknowledged
●	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
	MEE	Missing Events Emission	Unresolved
	RSW	Redundant Storage Writes	Unresolved
	L13	Divide before Multiply Operation	Unresolved
	L19	Stable Compiler Version	Unresolved

Table of Contents

Review

Contract Name	CloudBTC
Testing Deploy	https://goerli.etherscan.io/address/0x1fd39ee8f122f0cb7c01bf1a20c6de088b7141f1
Symbol	TSD
Decimals	18
Total Supply	1,000,000,000,000,000,000,000,000
Badge Eligibility	Must Fix Criticals

Audit Updates

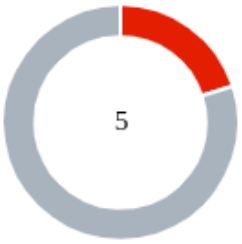
Initial Audit	16 Feb 2024

Source Files

Filename	SHA256
contracts/CloudBTC.sol	2d014f6b946b02a6c30a4d7c6e4b966921f8ce9681b43b47d6b49359be94e031
@openzeppelin/contracts/utils/Context.sol	9c1cc43aa4a2bde5c7dea0d4830cd42c54813ff883e55c8d8f12e6189bf7f10a
@openzeppelin/contracts/token/ERC20/IERC20.sol	6f2faae462e286e24e091d7718575179644dc60e79936ef0c92e2d1ab3ca3cee
@openzeppelin/contracts/token/ERC20/ERC20.sol	2d874da1c1478ed22a2d30dcf1a6ec0d09a13f897ca680d55fb49fbcc0e0c5b1
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	1d079c20a192a135308e99fa5515c27acfbb071e6cdb0913b13634e630865939

@openzeppelin/contracts/interfaces/draft-IERC6093.sol	4aea87243e6de38804bf8737bf86f750443d3b5e63dd0fd0b7ad92f77cdbd3e3
@openzeppelin/contracts/access/Ownable.sol	38578bd71c0a909840e67202db527cc6b4e6b437e0f39f0c909da32c1e30cb81

Findings Breakdown



Critical	1
Medium	0
Minor / Informative	4

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

ST - Stops Transactions

Criticality	Critical
Location	contracts/CloudBTC.sol#L58
Status	Unresolved

Description

The contract owner has the authority to stop the transfers for all users excluding the `whitelist` addresses. The owner may take advantage of it by setting the `isPresaleActive` to `true`. As a result, the contract may prevent any transfer including buy and sell transactions.

```
if (isPresaleActive) {
      return whitelist[from] || whitelist[to];
```

Recommendation

The contract could embody a check for not allowing setting the `_maxTxAmount` less than a reasonable amount. A suggested implementation could check that the minimum amount should be more than a fixed percentage of the total supply. The team should carefully manage the private keys of the owner's account. 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, which will eliminate the threats but it is non-reversible.

Team Update

The team has acknowledged that this is not a security issue and states:

During the presale period, token liquidity will not be available, yet participants will still receive their tokens. To prevent unauthorized liquidity provisioning, token transactions are temporarily disabled. Upon completion of the presale, all transactions will be enabled, allowing trading to commence as usual. This measure is part of our commitment to ensure a secure and orderly market introduction for our token.

MEE - Missing Events Emission

Criticality	Minor / Informative
Location	contracts/CloudBTC.sol#L35,39,43
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 setIsPresaleActive(bool active) external onlyOwner
{
        isPresaleActive = active;
    }

    function addToWhitelist(address addr) external onlyOwner {
        whitelist[addr] = true;
    }

    function removeFromWhitelist(address addr) external
onlyOwner {
        whitelist[addr] = false;
    }
```

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.

RSW - Redundant Storage Writes

Criticality	Minor / Informative
Location	contracts/CloudBTC.sol#L35,39,43
Status	Unresolved

Description

The contract modifies the state of the following variables without checking if their current value is the same as the one given as an argument. As a result, the contract performs redundant storage writes, when the provided parameter matches the current state of the variables, leading to unnecessary gas consumption and inefficiencies in contract execution.

```
    function setIsPresaleActive(bool active) external onlyOwner
{
        isPresaleActive = active;
    }

    function addToWhitelist(address addr) external onlyOwner {
        whitelist[addr] = true;
    }

    function removeFromWhitelist(address addr) external
onlyOwner {
        whitelist[addr] = false;
    }
```

Recommendation

The team is advised to implement additional checks within to prevent redundant storage writes when the provided argument matches the current state of the variables. By incorporating statements to compare the new values with the existing values before proceeding with any state modification, the contract can avoid unnecessary storage operations, thereby optimizing gas usage.

L13 - Divide before Multiply Operation

Criticality	Minor / Informative
Location	contracts/CloudBTC.sol#L30
Status	Unresolved

Description

It is important to be aware of the order of operations when performing arithmetic calculations. This is especially important when working with large numbers, as the order of operations can affect the final result of the calculation. Performing divisions before multiplications may cause loss of prediction.

```
uint256 walletSupply = totalSupply * percentages[i] / 100 * (10
** decimals())
```

Recommendation

To avoid this issue, it is recommended to carefully consider the order of operations when performing arithmetic calculations in Solidity. It's generally a good idea to use parentheses to specify the order of operations. The basic rule is that the multiplications should be prior to the divisions.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	contracts/CloudBTC.sol#L2
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.

```
pragma solidity ^0.8.20;
```

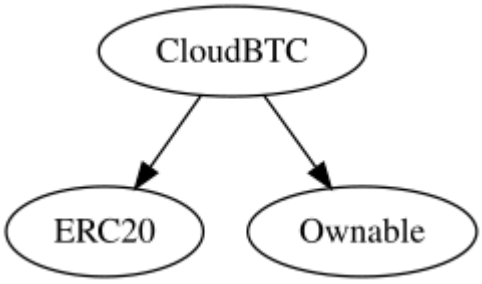
Recommendation

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.

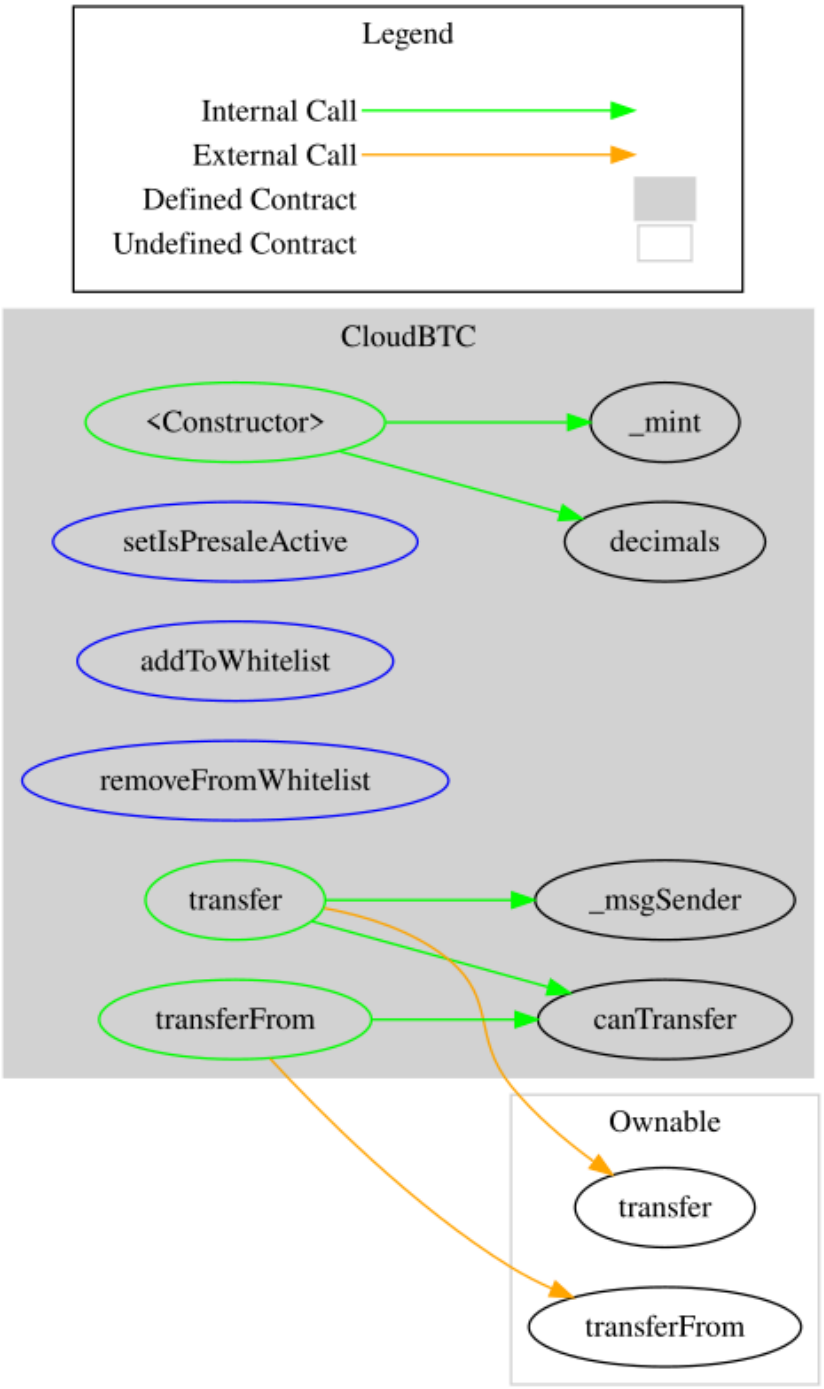
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
CloudBTC	Implementation	ERC20, Ownable		
		Public	✓	ERC20 Ownable
	setIsPresaleActive	External	✓	onlyOwner
	addToWhitelist	External	✓	onlyOwner
	removeFromWhitelist	External	✓	onlyOwner
	transfer	Public	✓	-
	transferFrom	Public	✓	-
	canTransfer	Internal		

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
CloudBTC
<Constructor>
_mint
setIsPresaleActive
decimals
addToWhitelist
removeFromWhitelist
transfer
_msgSender
transferFrom
canTransfer
Ownable
transfer
transferFrom

Summary

CloudBTC 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, however, the team has acknowledged this finding. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership 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