Cyberscope

Audit Report

ClickBee Token

January 2024

Network	ETH
Address	0x94da8B8D431DfB0F0F2c28ac46C70c144ce880f5
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
●	LWM	Lack of Withdrawal Mechanism	Unresolved
●	BFV	Buy Function Vulnerability	Unresolved
●	MEE	Missing Events Emission	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L19	Stable Compiler Version	Unresolved

Table of Contents

Analysis **1**
Diagnostics **2**
Table of Contents **3**
Review **4**
Audit Updates 4
Source Files 5
Findings Breakdown **6**
LWM - Lack of Withdrawal Mechanism 7
Description 7
Recommendation 7
BFV - Buy Function Vulnerability 8
Description 8
Recommendation 8
MEE - Missing Events Emission 9
Description 9
Recommendation 9
L04 - Conformance to Solidity Naming Conventions 10
Description 10
Recommendation 10
L19 - Stable Compiler Version 11
Description 11
Recommendation 11
Functions Analysis **12**
Inheritance Graph **13**
Flow Graph **14**
Summary **15**
Disclaimer **16**
About Cyberscope **17**

Review

Contract Name	ClickBeeToken
Compiler Version	v0.8.23+commit.f704f362
Optimization	200 runs
Explorer	https://etherscan.io/address/0x94da8b8d431dfb0f0f2c28ac46c70c144ce880f5
Address	0x94da8b8d431dfb0f0f2c28ac46c70c144ce880f5
Network	ETH
Symbol	BEES
Decimals	18
Total Supply	100,000,000,000
Badge Eligibility	Yes

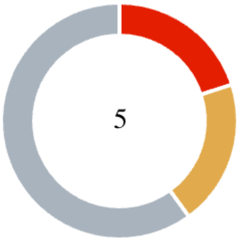
Audit Updates

Initial Audit	06 Jan 2024

Source Files

Filename	SHA256
Clickbee.sol	e05cd0617f4010317873c7fd57baf1fb0960f1ad5df887d17632de3735e7c2e7
@openzeppelin/contracts/utils/Context.sol	847fda5460fee70f56f4200f59b82ae622bb03c79c77e67af010e31b7e2cc5b6
@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	1
●	Minor / Informative	3

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

LWM - Lack of Withdrawal Mechanism

Criticality	Critical
Location	Clickbee.sol#L57,61
Status	Unresolved

Description

The contract successfully collects fees from buy and sell transactions in the form of tokens and ETH, transferring them to the contract's balance. However, it lacks a mechanism for the owner to withdraw or manage these accumulated tokens and ETH. This could result in funds being locked within the contract indefinitely, limiting the owner's ability to access or utilize the collected fees.

```
function buy() external payable {
    _transferWithFees(owner(), msg.sender, msg.value, _buyFeePercentage);
}

function sell(uint256 amount) external {
    _transferWithFees(msg.sender, address(this), amount,
_sellFeePercentage);
    payable(msg.sender).transfer(amount);
}
```

Recommendation

The team is strongly advised to implement a withdrawal mechanism that allows the owner to retrieve the accumulated tokens and ETH from the contract's balance. Implementing a withdrawal mechanism ensures that the owner has control over the collected fees and can utilize them as needed.

BFV - Buy Function Vulnerability

Criticality	Critical
Location	Clickbee.sol#L57
Status	Unresolved

Description

The contract contains a vulnerability in the `buy` function that could lead to issues if the owner renounces ownership. The `buy` function relies on the `_transferWithFees` internal function, which eventually calls `super._transfer(sender, recipient, amount - fee)`. This function checks if either the sender or the recipient is the zero address and reverts if true. If the owner renounces ownership, the owner's address becomes the zero address, leading to a failure in the `buy` function.

```
function buy() external payable {
    _transferWithFees(owner(), msg.sender, msg.value, _buyFeePercentage);
}
```

Recommendation

The team is advised to review and revise the business logic of the buy functionality. The team should take into account the possibility of ownership renouncement and adjust the logic accordingly to maintain the intended functionality.

MEE - Missing Events Emission

Criticality	Minor / Informative
Location	Clickbee.sol#L49,54
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.

```
_buyFeePercentage = newBuyFeePercentage;
_sellFeePercentage = newSellFeePercentage;
```

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	Clickbee.sol#L22,25
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.

```
uint256 private constant _totalSupply = 100_000_000_000 * 10**18
uint256 private constant _maxFeePercentage = 25
```

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.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	Clickbee.sol#L15
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 ^0.8.0;
```

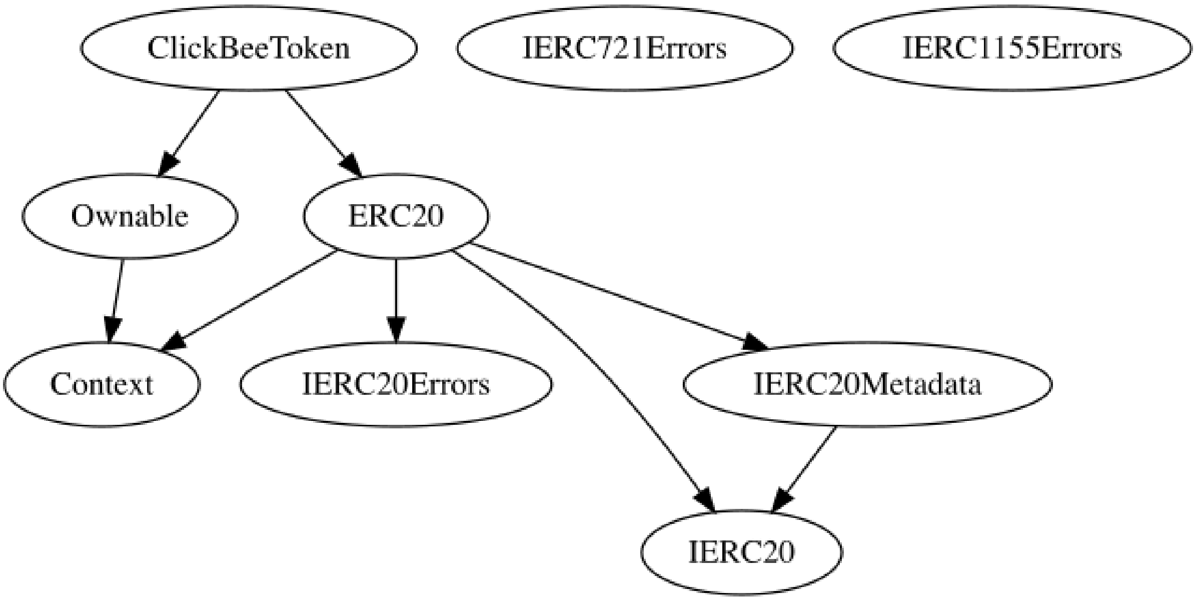
Recommendation

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.

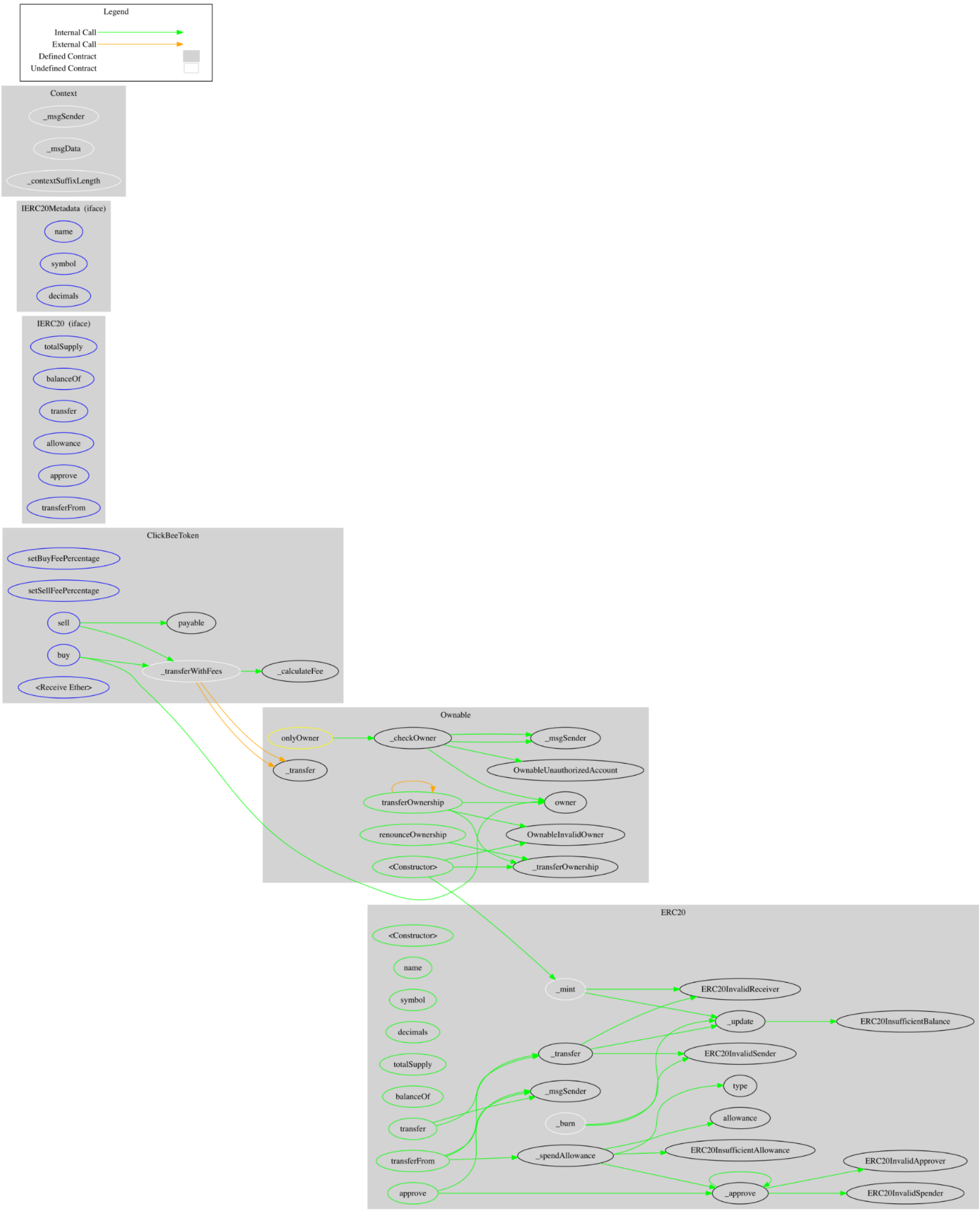
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
ClickBeeToken	Implementation	ERC20, Ownable		
		Public	✓	ERC20
	_calculateFee	Internal		
	_transferWithFees	Internal	✓	
	setBuyFeePercentage	External	✓	onlyOwner
	setSellFeePercentage	External	✓	onlyOwner
	buy	External	Payable	-
	sell	External	✓	-
	transferOwnership	Public	✓	onlyOwner
		External	Payable	-

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Context
_msgSender
_msgData
_contextSuffixLength
IERC20Metadata (iface)
name
symbol
decimals
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
ClickBeeToken
setBuyFeePercentage
setSellFeePercentage
sell
payable
buy
_transferWithFees
_calculateFee
<Receive Ether>
Ownable
onlyOwner
_checkOwner
_msgSender
_transfer
OwnableUnauthorizedAccount
transferOwnership
owner
renounceOwnership
OwnableInvalidOwner
<Constructor>
_transferOwnership
ERC20
<Constructor>
name
symbol
_mint
ERC20InvalidReceiver
decimals
_update
ERC20InsufficientBalance
totalSupply
_transfer
ERC20InvalidSender
balanceOf
_msgSender
type
transfer
_burn
allowance
transferFrom
_spendAllowance
ERC20InsufficientAllowance
ERC20InvalidApprover
approve
_approve
ERC20InvalidSpender

Summary

ClickBee Token contract implements a token mechanism. This audit investigates security issues, business logic concerns, and potential improvements. ClickBee Token is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler errors 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. There is also a limit of max 25% fees. Lastly, the contract offers the option to buy and sell tokens by using the buy and sell functions, without the need for a DEX. The buy function enables users to acquire tokens by sending Ether, with a fee deducted from the transaction and transferred to the contract owner. Conversely, the sell function permits users to sell tokens back to the contract, deducting a fee before transferring the corresponding Ether amount to the seller.

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