



Cyberscope

Audit Report

I'm Meta Trader

April 2024

Network BSC

Address 0xc1db036499641dc14d2dee5a357741dcf4622e67

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
●	IDI	Immutable Declaration Improvement	Unresolved
●	RF	Redundant Function	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved

Table of Contents

Analysis	1
Diagnostics	2
Table of Contents	3
Review	4
Audit Updates	4
Source Files	5
Findings Breakdown	6
IDI - Immutable Declaration Improvement	7
Description	7
Recommendation	7
RF - Redundant Function	8
Description	8
Recommendation	8
L02 - State Variables could be Declared Constant	9
Description	9
Recommendation	9
L04 - Conformance to Solidity Naming Conventions	10
Description	10
Recommendation	10
Functions Analysis	11
Inheritance Graph	14
Flow Graph	15
Summary	16
Disclaimer	17
About Cyberscope	18

Review

Contract Name	BEP20Token
Compiler Version	v0.5.16+commit.9c3226ce
Optimization	200 runs
Explorer	https://bscscan.com/address/0xc1db036499641dc14d2dee5a357741dcf4622e67
Address	0xc1db036499641dc14d2dee5a357741dcf4622e67
Network	BSC
Symbol	IMMT
Decimals	8
Total Supply	1,000,000,000
Badge Eligibility	Yes

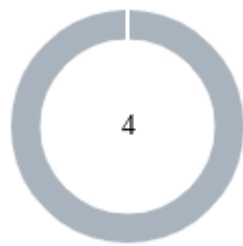
Audit Updates

Initial Audit	17 Apr 2024
Corrected Phase 2	22 Apr 2024
Corrected Phase 2	23 Apr 2024

Source Files

Filename	SHA256
immt.sol	9c0c2eb974a85d1637e1f0eb65f8c873a2e28f58be4185902fe435406b6e1d6d
SafeMath.sol	bb8e16d39f118eda96b401e503c227b0fb42f305de7528d2779f9a158234f488
IBEP20.sol	1523ec45a4e33044959922147b3d4838193cfbd8166451004b55959aec63e76e

Findings Breakdown



● Critical	0
● Medium	0
● Minor / Informative	4

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

IDI - Immutable Declaration Improvement

Criticality	Minor / Informative
Location	immt.sol#L387,388,389
Status	Unresolved

Description

The contract declares state variables that their value is initialized once in the constructor and are not modified afterwards. The `immutable` is a special declaration for this kind of state variables that saves gas when it is defined.

```
uint8 private _decimals;  
string private _symbol;  
string private _name;
```

Recommendation

By declaring a variable as immutable, the Solidity compiler is able to make certain optimizations. This can reduce the amount of storage and computation required by the contract, and make it more gas-efficient.

RF - Redundant Function

Criticality	Minor / Informative
Location	immt.sol#L311,480
Status	Unresolved

Description

The contract contains a redundant function `getOwner()` that mirrors the functionality of the `owner()` function. Both functions return the same value - the address of the contract owner. While redundancy itself may not necessarily pose a security risk, it can introduce unnecessary complexity.

Redundant code segments like `getOwner()` may confuse developers or users interacting with the contract. Additionally, redundant functions can increase gas costs and bloating of the contract size without providing any tangible benefit.

```
function owner() public view returns (address) {  
    return _owner;  
}  
...  
function getOwner() external view returns (address) {  
    return owner();  
}
```

Recommendation

Consider removing the redundant function `getOwner()` and encouraging users to utilize the existing `owner()` function directly. Simplifying the contract interface by eliminating unnecessary duplication enhances readability and improves overall contract efficiency.

L02 - State Variables could be Declared Constant

Criticality	Minor / Informative
Location	immt.sol#L149
Status	Unresolved

Description

State variables can be declared as constant using the constant keyword. This means that the value of the state variable cannot be changed after it has been set. Additionally, the constant variables decrease gas consumption of the corresponding transaction.

```
IBEP20 public token
```

Recommendation

Constant state variables can be useful when the contract wants to ensure that the value of a state variable cannot be changed by any function in the contract. This can be useful for storing values that are important to the contract's behavior, such as the contract's address or the maximum number of times a certain function can be called. The team is advised to add the constant keyword to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	immt.sol#L168
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.

```
uint _tokenAmount
address[] calldata _path
uint _expectedMinOut
```

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>.

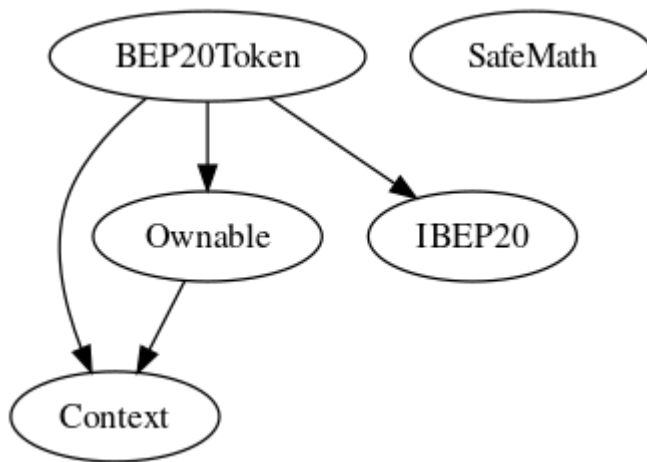
Functions Analysis

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
		Internal	✓	
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
		Internal	✓	
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
BEP20Token	Implementation	Context, IBEP20, Ownable		
		Public	✓	-
	getOwner	External		-
	decimals	External		-
	symbol	External		-
	name	External		-

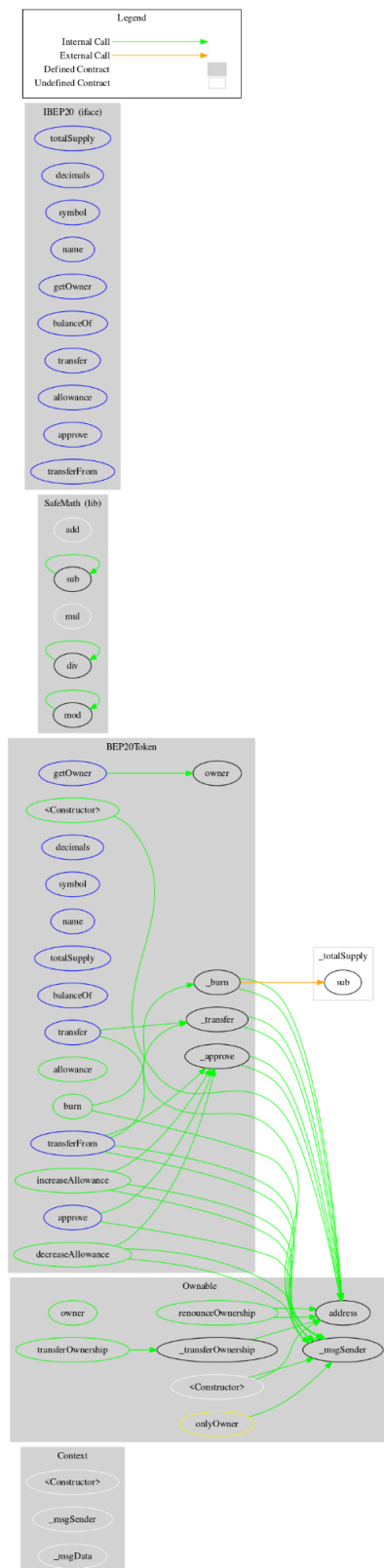
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	Public		-
	approve	External	✓	-
	_approve	Internal	✓	
	transferFrom	External	✓	-
	_transfer	Internal	✓	
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	burn	Public	✓	-
	_burn	Internal	✓	
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		

IBEP20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-

Inheritance Graph



Flow Graph



Summary

I'm Meta Trader contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements.

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>