



Cyberscope

# Audit Report

## **Big Mike**

March 2024

Network    ETH

Address    0x1929761e87667283f087ea9ab8370c174681b4e9

Audited by    © cyberscope

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>Review</b>	<b>2</b>
Audit Updates	2
Source Files	2
<b>Findings Breakdown</b>	<b>3</b>
<b>Diagnostics</b>	<b>4</b>
IDI - Immutable Declaration Improvement	5
Description	5
Recommendation	5
L04 - Conformance to Solidity Naming Conventions	6
Description	6
Recommendation	6
L09 - Dead Code Elimination	8
Description	8
Recommendation	9
L14 - Uninitialized Variables in Local Scope	10
Description	10
Recommendation	10
L15 - Local Scope Variable Shadowing	11
Description	11
Recommendation	11
<b>Functions Analysis</b>	<b>12</b>
<b>Inheritance Graph</b>	<b>16</b>
<b>Flow Graph</b>	<b>17</b>
<b>Summary</b>	<b>18</b>
<b>Disclaimer</b>	<b>19</b>
<b>About Cyberscope</b>	<b>20</b>

## Review

Contract Name	BigMike
Compiler Version	v0.8.23+commit.f704f362
Optimization	200 runs
Explorer	<a href="https://etherscan.io/address/0x1929761e87667283f087ea9ab8370c174681b4e9">https://etherscan.io/address/0x1929761e87667283f087ea9ab8370c174681b4e9</a>
Address	0x1929761e87667283f087ea9ab8370c174681b4e9
Network	ETH
Symbol	BIGMIKE
Decimals	18
Total Supply	100,000,000

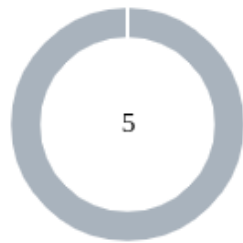
## Audit Updates

Initial Audit	09 Mar 2024
---------------	-------------

## Source Files

Filename	SHA256
BigMike.sol	c9298a0cb5e475d4dfd4376cfe35275e223e6645bf49f747204ed4f37a540747

## Findings Breakdown



Critical	0
Medium	0
Minor / Informative	5

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	0	0	0	0
Medium	0	0	0	0
Minor / Informative	5	0	0	0

# Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	IDI	Immutable Declaration Improvement	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L09	Dead Code Elimination	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved
●	L15	Local Scope Variable Shadowing	Unresolved

## IDI - Immutable Declaration Improvement

<b>Criticality</b>	Minor / Informative
<b>Location</b>	BigMike.sol#L541,548
<b>Status</b>	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.

```
swapTokensAtAmt  
lpPair
```

### 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.

## L04 - Conformance to Solidity Naming Conventions

<b>Criticality</b>	Minor / Informative
<b>Location</b>	BigMike.sol#L477,653,659,668,689
<b>Status</b>	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.

```
function WETH() external pure returns (address);  
bool _isExempt  
address _address  
uint256 _buyTax  
uint256 _sellTax
```

### 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>.



## L09 - Dead Code Elimination

Criticality	Minor / Informative
Location	BigMike.sol#L353
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.

```
function _burn(address account, uint256 amount) internal virtual {
    require(account != address(0), "ERC20: burn from the zero address");

    _beforeTokenTransfer(account, address(0), amount);

    uint256 accountBalance = _balances[account];
    ...
    _totalSupply -= amount;
}

emit Transfer(account, address(0), amount);

_afterTokenTransfer(account, address(0), amount);
}
```

## 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.

## L14 - Uninitialized Variables in Local Scope

<b>Criticality</b>	Minor / Informative
<b>Location</b>	BigMike.sol#L526
<b>Status</b>	Unresolved

### Description

Using an uninitialized local variable can lead to unpredictable behavior and potentially cause errors in the contract. It's important to always initialize local variables with appropriate values before using them.

```
address _v2Router
```

### Recommendation

By initializing local variables before using them, the contract ensures that the functions behave as expected and avoid potential issues.

## L15 - Local Scope Variable Shadowing

<b>Criticality</b>	Minor / Informative
<b>Location</b>	BigMike.sol#L523
<b>Status</b>	Unresolved

### Description

Local scope variable shadowing occurs when a local variable with the same name as a variable in an outer scope is declared within a function or code block. When this happens, the local variable "shadows" the outer variable, meaning that it takes precedence over the outer variable within the scope in which it is declared.

```
uint256 _totalSupply = 100_000_000 * (10 ** decimals())
```

### Recommendation

It's important to be aware of shadowing when working with local variables, as it can lead to confusion and unintended consequences if not used correctly. It's generally a good idea to choose unique names for local variables to avoid shadowing outer variables and causing confusion.

## Functions Analysis

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
<b>Context</b>	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
<b>IERC20</b>	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
<b>IERC20Metadata</b>	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
<b>ERC20</b>	Implementation	Context, IERC20, IERC20Meta data		

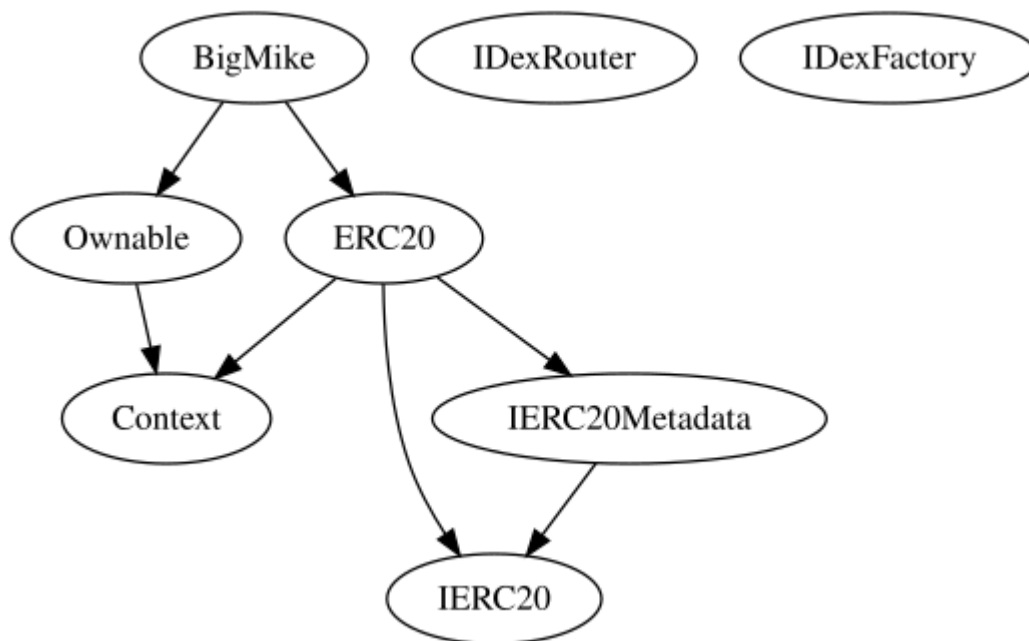
		Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
<b>Ownable</b>	Implementation	Context		
		Public	✓	-
	owner	Public		-

	renounceOwnership	External	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
<b>IDexRouter</b>	Interface			
	factory	External		-
	WETH	External		-
	swapExactTokensForETHSupportingFee OnTransferTokens	External	✓	-
<b>IDexFactory</b>	Interface			
	createPair	External	✓	-
<b>BigMike</b>	Implementation	ERC20, Ownable		
		Public	✓	ERC20
	_transfer	Internal	✓	
	checkTransactionLimits	Internal		
	handleTaxes	Internal	✓	
	swapTokensForETH	Private	✓	
	swapBack	Private	✓	
	setExemptFromTax	External	✓	onlyOwner
	setExemptFromLimits	External	✓	onlyOwner
	updateTaxes	External	✓	onlyOwner
	openTrading	External	✓	onlyOwner
	removeLimits	External	✓	onlyOwner

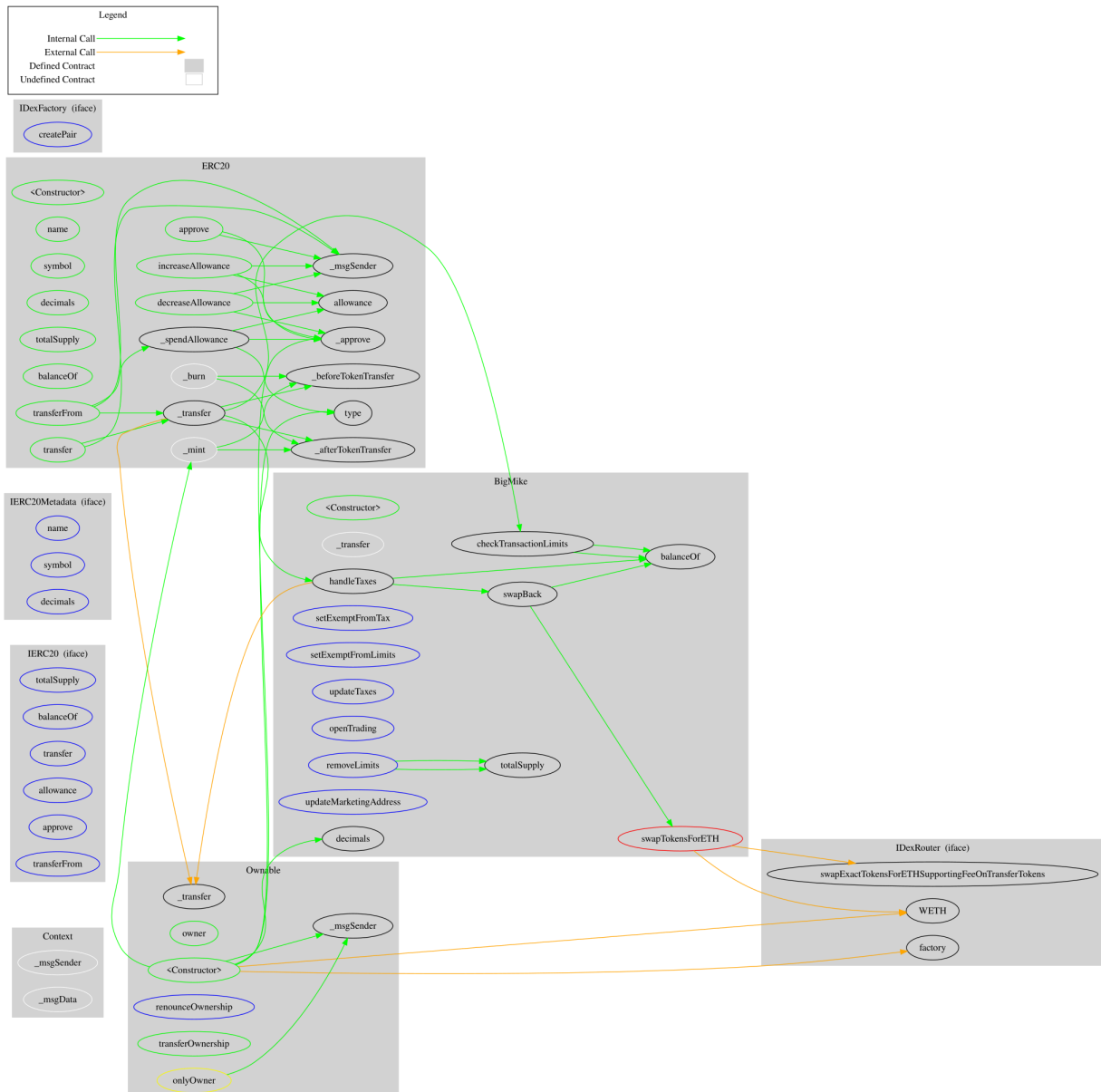
	updateMarketingAddress	External	✓	onlyOwner
--	------------------------	----------	---	-----------



## Inheritance Graph



# Flow Graph



## Summary

Big Mike contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. Big Mike 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. The fees are locked at 2% for buy transactions and 3% for sell transactions.

The contract's ownership has been renounced. The information regarding the transaction can be accessed through the following link:

<https://etherscan.io/tx/0xa7fa85040547103c493bced08d77f0b1242fc63578e8b10d8e45b1a8be63be33>

## 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>