



Cyberscope

Audit Report

Cyberdoge

April 2024

Network MATIC

Address 0xc78d4982247264eDD89079744390dc192dA954f5

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
●	RSML	Redundant SafeMath Library	Unresolved
●	L09	Dead Code Elimination	Unresolved
●	L16	Validate Variable Setters	Unresolved

Table of Contents

Analysis	1
Diagnostics	2
Table of Contents	3
Review	4
Audit Updates	4
Source Files	4
Findings Breakdown	6
RSML - Redundant SafeMath Library	7
Description	7
Recommendation	7
L09 - Dead Code Elimination	8
Description	8
Recommendation	9
L16 - Validate Variable Setters	10
Description	10
Recommendation	10
Functions Analysis	11
Inheritance Graph	13
Flow Graph	14
Summary	15
Disclaimer	16
About Cyberscope	17

Review

Contract Name	StandardToken
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	200 runs
Explorer	https://polygonscan.com/address/0xc78d4982247264edd89079744390dc192da954f5
Address	0xc78d4982247264edd89079744390dc192da954f5
Network	MATIC
Symbol	CDT
Decimals	18
Total Supply	1,000,000,000,000
Badge Eligibility	Yes

Audit Updates

Initial Audit	02 Apr 2024
---------------	-------------

Source Files

Filename	SHA256
contracts/BaseToken.sol	36925847a9d08350d6d67643d11cc384aa0a30cf25d38ebefc8e7d48b89d47fa
contracts/standard/StandardToken.sol	fe416e378f75ff5d57983c1936ec59e14e7c43286d3f7874982baa2f926d3cc2

@openzeppelin/contracts/utils/Context.sol	5828bf38f9376b659a8edbbe2df0d06b29a09e37ecd470465dda2bbcb612c85d
@openzeppelin/contracts/utils/math/SafeMath.sol	a6357cd855b7f26b18812e89084475213054c3c76ce12ff2b82f4f9b3b5ae76b
@openzeppelin/contracts/token/ERC20/IERC20.sol	b2565dec975f684ef0edfa505e212d0d0b602e1311afab782ea06ea8d3f49bb6
@openzeppelin/contracts/access/Ownable.sol	0195650aabf5270babe540969c56f8f244342aebce89266787a3b015e41d608f

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

RSML - Redundant SafeMath Library

Criticality	Minor / Informative
Location	contracts/standard/StandardToken.sol
Status	Unresolved

Description

SafeMath is a popular Solidity library that provides a set of functions for performing common arithmetic operations in a way that is resistant to integer overflows and underflows.

Starting with Solidity versions that are greater than or equal to 0.8.0, the arithmetic operations revert to underflow and overflow. As a result, the native functionality of the Solidity operations replaces the SafeMath library. Hence, the usage of the SafeMath library adds complexity, overhead and increases gas consumption unnecessarily in cases where the explanatory error message is not used.

```
library SafeMath {...}
```

Recommendation

The team is advised to remove the SafeMath library in cases where the revert error message is not used. Since the version of the contract is greater than `0.8.0` then the pure Solidity arithmetic operations produce the same result.

If the previous functionality is required, then the contract could exploit the `unchecked { ... }` statement.

Read more about the breaking change on

<https://docs.soliditylang.org/en/v0.8.16/080-breaking-changes.html#solidity-v0-8-0-breaking-changes>.

L09 - Dead Code Elimination

Criticality	Minor / Informative
Location	contracts/standard/StandardToken.sol#L287,332
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);

    _balances[account] = _balances[account].sub(
        amount,
        "ERC20: burn amount exceeds balance"
    );
    _totalSupply = _totalSupply.sub(amount);
    emit Transfer(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.

L16 - Validate Variable Setters

Criticality	Minor / Informative
Location	contracts/standard/StandardToken.sol#L37
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.

```
payable(serviceFeeReceiver_).transfer(serviceFee_)
```

Recommendation

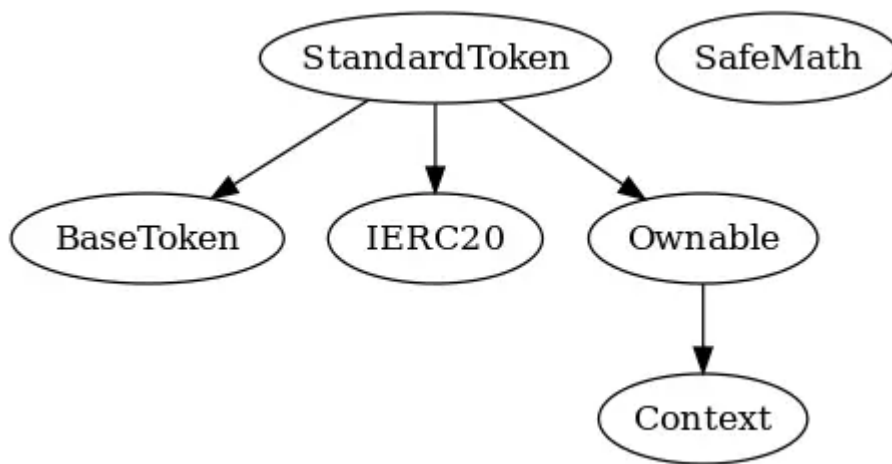
By adding the proper check, the contract will not allow the variables to be configured with zero value. This will ensure that the contract can handle all possible input values and avoid unexpected behavior or errors. Hence, it can help to prevent the contract from being exploited or operating unexpectedly.

Functions Analysis

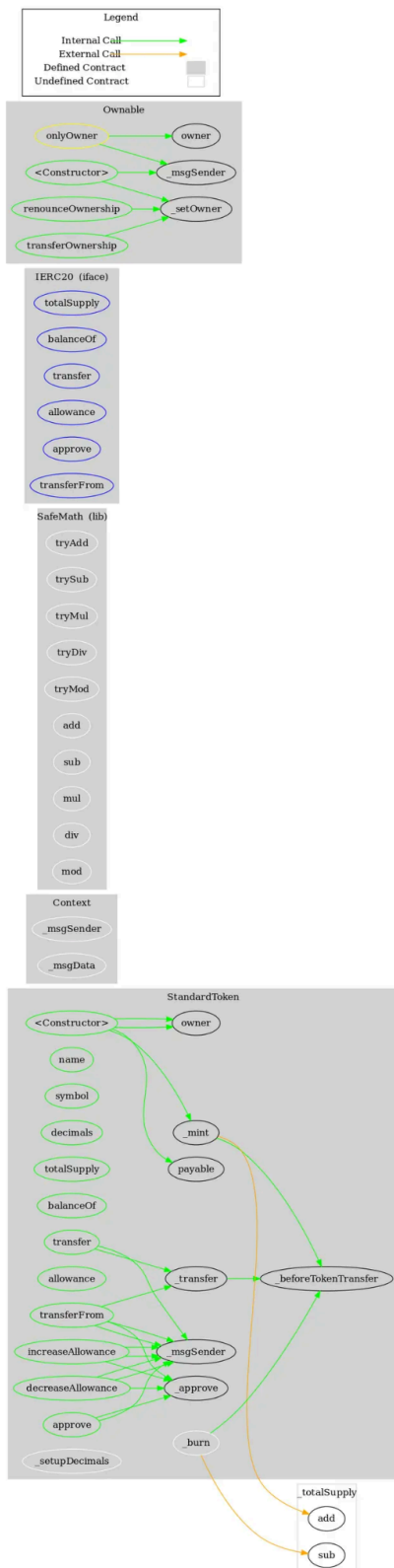
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
StandardToken	Implementation	IERC20, Ownable, BaseToken		
		Public	Payable	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_setupDecimals	Internal	✓	

	_beforeTokenTransfer	Internal	✓	
--	----------------------	----------	---	--

Inheritance Graph



Flow Graph



Summary

Cyberdoge contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. Cyberdoge 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 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>