Cyberscope

Audit Report

Pomerdoge

June 2023

Network ETH

Address 0xA9a81112C916bD05CfBd3E4C08fda3256132bf41

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
	RCS	Redundant Code Statement	Unresolved
	IDI	Immutable Declaration Improvement	Unresolved
	L03	Redundant Statements	Unresolved

Table of Contents

Review

Contract Name	Pomerdoge
Compiler Version	v0.8.19+commit.7dd6d404
Optimization	200 runs
Explorer	https://etherscan.io/address/0xa9a81112c916bd05cfbd3e4c08fda3256132bf41
Address	0xa9a81112c916bd05cfbd3e4c08fda3256132bf41
Network	ETH
Symbol	POMD
Decimals	18
Total Supply	1,777,777,777

Audit Updates

Initial Audit	23 Jun 2023

Source Files

Filename	SHA256
Pomerdoge.sol	6bcd849ac4f2b961283c37c531dfc66bed37673bd479aa4f3277f9c6f293855a

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

RCS - Redundant Code Statement

Criticality	Minor / Informative
Location	Pomerdoge.sol#L102
Status	Unresolved

Description

The contract includes the `_totalSupply` keyword in its constructor function.

```
constructor () {
    _name = "Pomerdoge";
    _symbol = "POMD";
    _totalSupply;
    _mint(owner(), 1_777_777_777 ether );
}
```

However, `_totalSupply` statement inside the constructor is redundant, as it does not contribute to the initialization of the contract's state.

Furthermore, the `_mint` function is used in the constructor to create a certain number of tokens and assign them to the contract's owner. This operation effectively sets the total supply of tokens, making the `_totalSupply` statement redundant at this point.

Recommendation

Consider removing the `_totalSupply` statement from the constructor, or assign it a value that represents the initial total supply of tokens. This would improve the clarity and efficiency of the contract's code.

IDI - Immutable Declaration Improvement

Criticality	Minor / Informative
Location	Pomerdoge.sol#L100,101
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.

```
_name
_symbol
```

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.

L03 - Redundant Statements

Criticality	Minor / Informative
Location	Pomerdoge.sol#L89
Status	Unresolved

Description

Redundant statements are statements that are unnecessary or have no effect on the contract's behavior. These can include declarations of variables or functions that are not used, or assignments to variables that are never used.

As a result, it can make the contract's code harder to read and maintain, and can also increase the contract's size and gas consumption, potentially making it more expensive to deploy and execute.

```
contract Pomerdoge is Context, IERC20, IERC20Metadata, Ownable{
    mapping (address => uint256) private _balances;

    mapping (address => mapping (address => uint256)) private
_allowances;

    uint256 private _totalSupply;
...

        _allowances[owner][spender] = amount;
        emit Approval(owner, spender, amount);
    }

    function _beforeTokenTransfer(address from, address to, uint256
amount) internal virtual { }
}
```

Recommendation

To avoid redundant statements, it's important to carefully review the contract's code and remove any statements that are unnecessary or not used. This can help to improve the clarity and efficiency of the contract's code.

By removing unnecessary or redundant statements from the contract's code, the clarity and efficiency of the contract will be improved. Additionally, the size and gas consumption will be reduced.

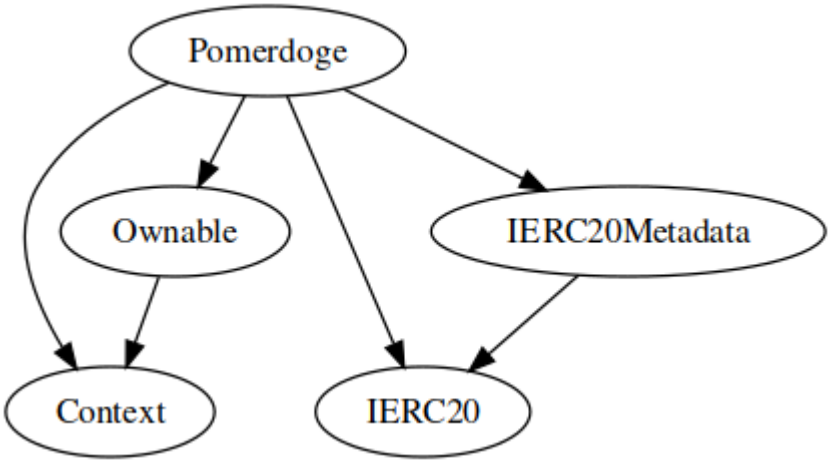
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
Ownable	Implementation	Context		

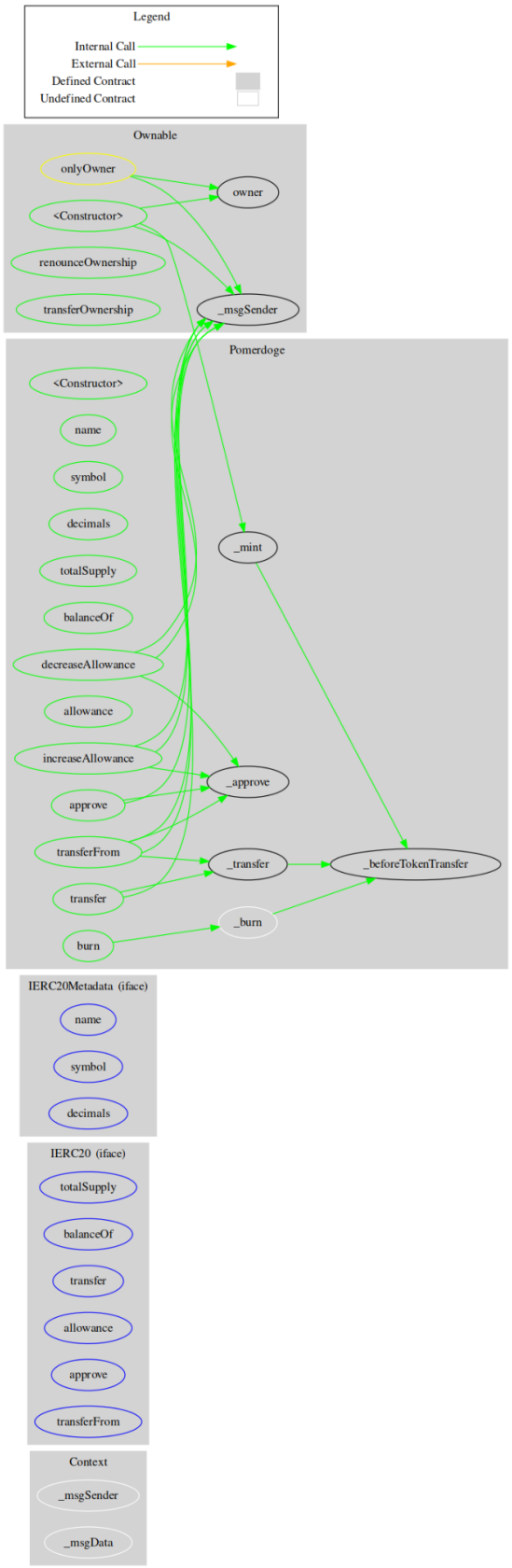
		Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
Pomerdoge	Implementation	Context, IERC20, IERC20Metadata, Ownable		
		Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	burn	Public	✓	onlyOwner

	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Ownable
onlyOwner
owner
<Constructor>
renounceOwnership
transferOwnership
_msgSender
Pomerdoge
<Constructor>
name
symbol
decimals
totalSupply
balanceOf
decreaseAllowance
allowance
increaseAllowance
approve
transferFrom
transfer
burn
_mint
_approve
_transfer
_beforeTokenTransfer
_burn
IERC20Metadata (iface)
name
symbol
decimals
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
Context
_msgSender
_msgData

Summary

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