Cyberscope

Audit Report

Diamond Token

January 2025

Network	BSC
Address	0xbfa362937BFD11eC22a023aBF83B6dF4E5E303d4
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
●	L04	Conformance to Solidity Naming Conventions	Unresolved

Table of Contents

Risk Classification

The criticality of findings in Cyberscope's smart contract audits is determined by evaluating multiple variables. The two primary variables are:

1. **Likelihood of Exploitation**: This considers how easily an attack can be executed, including the economic feasibility for an attacker.
2. **Impact of Exploitation**: This assesses the potential consequences of an attack, particularly in terms of the loss of funds or disruption to the contract's functionality.

Based on these variables, findings are categorized into the following severity levels:

1. **Critical**: Indicates a vulnerability that is both highly likely to be exploited and can result in significant fund loss or severe disruption. Immediate action is required to address these issues.
2. **Medium**: Refers to vulnerabilities that are either less likely to be exploited or would have a moderate impact if exploited. These issues should be addressed in due course to ensure overall contract security.
3. **Minor**: Involves vulnerabilities that are unlikely to be exploited and would have a minor impact. These findings should still be considered for resolution to maintain best practices in security.
4. **Informative**: Points out potential improvements or informational notes that do not pose an immediate risk. Addressing these can enhance the overall quality and robustness of the contract.

Severity	Likelihood / Impact of Exploitation
Critical	Highly Likely / High Impact
Medium	Less Likely / High Impact or Highly Likely/ Lower Impact
Minor / Informative	Unlikely / Low to no Impact

Review

Contract Name	Diamond_Token
Compiler Version	v0.8.24+commit.e11b9ed9
Optimization	200 runs
Explorer	https://bscscan.com/address/0xbfa362937bfd11ec22a023abf83b6df4e5e303d4
Address	0xbfa362937bfd11ec22a023abf83b6df4e5e303d4
Network	BSC
Symbol	DIT
Decimals	18
Total Supply	100,000,000
Badge Eligibility	Yes

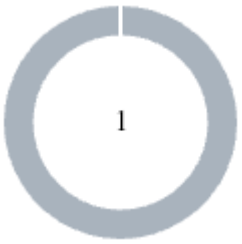
Audit Updates

Initial Audit	03 Jan 2025 https://github.com/cyberscope-io/audits/blob/main/dit/v1/audit.pdf
Corrected Phase 2	14 Jan 2025

Source Files

Filename	SHA256
DiamondToken.sol	0b2b255e27a3241c5002d9f34d17290aa8dad3b8ab3d385b669c5df2b9889ba2

Findings Breakdown



Critical	0
Medium	0
Minor / Informative	1

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

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	DiamondToken.sol#L8,56
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.

```
contract Diamond_Token is ERC20, Ownable {
    uint256 public transferFeePercentage = 300; // Fee percentage in basis points
(e.g., 100 = 1%)
    uint256 public liquidityFeePercentage = 150; // Fee percentage in basis points
(e.g., 100 = 1%)
    address public feeCollector = 0xd841972Ac48461517f561CB6785E2f1CBe37Ea07; //
Address to receive the fee
    address public constant ownerWallet =
        0x1d64FD1e4eB9Df7C75Ad4B4DAe6A23aa8C4B5fe8; // Owner's wallet
...
            super._update(sender, feeCollector, totalFee);
        }

        // Transfer remaining amount
        super._update(sender, recipient, amountAfterFee);
    }
}

...
```

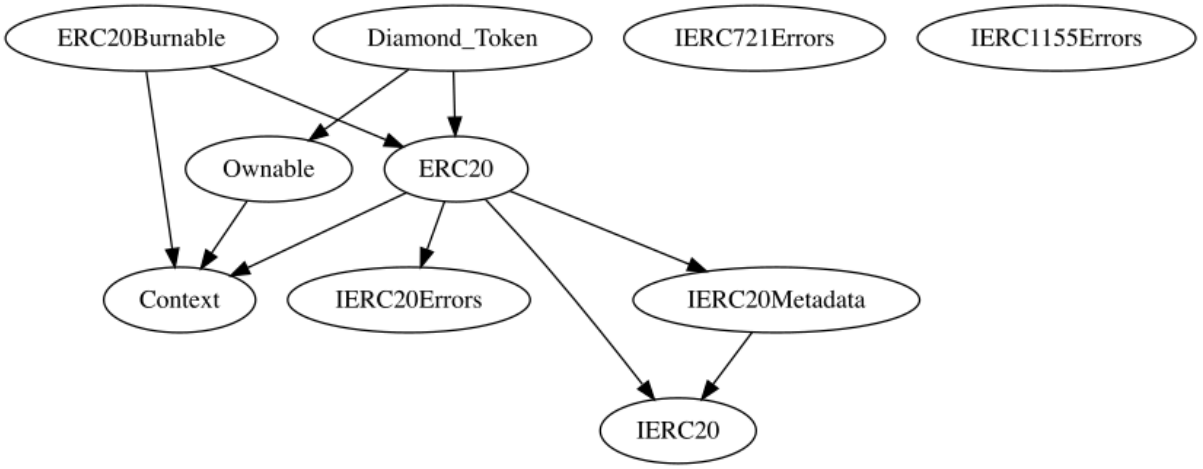
Recommendation

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/stable/style-guide.html#naming-conventions.

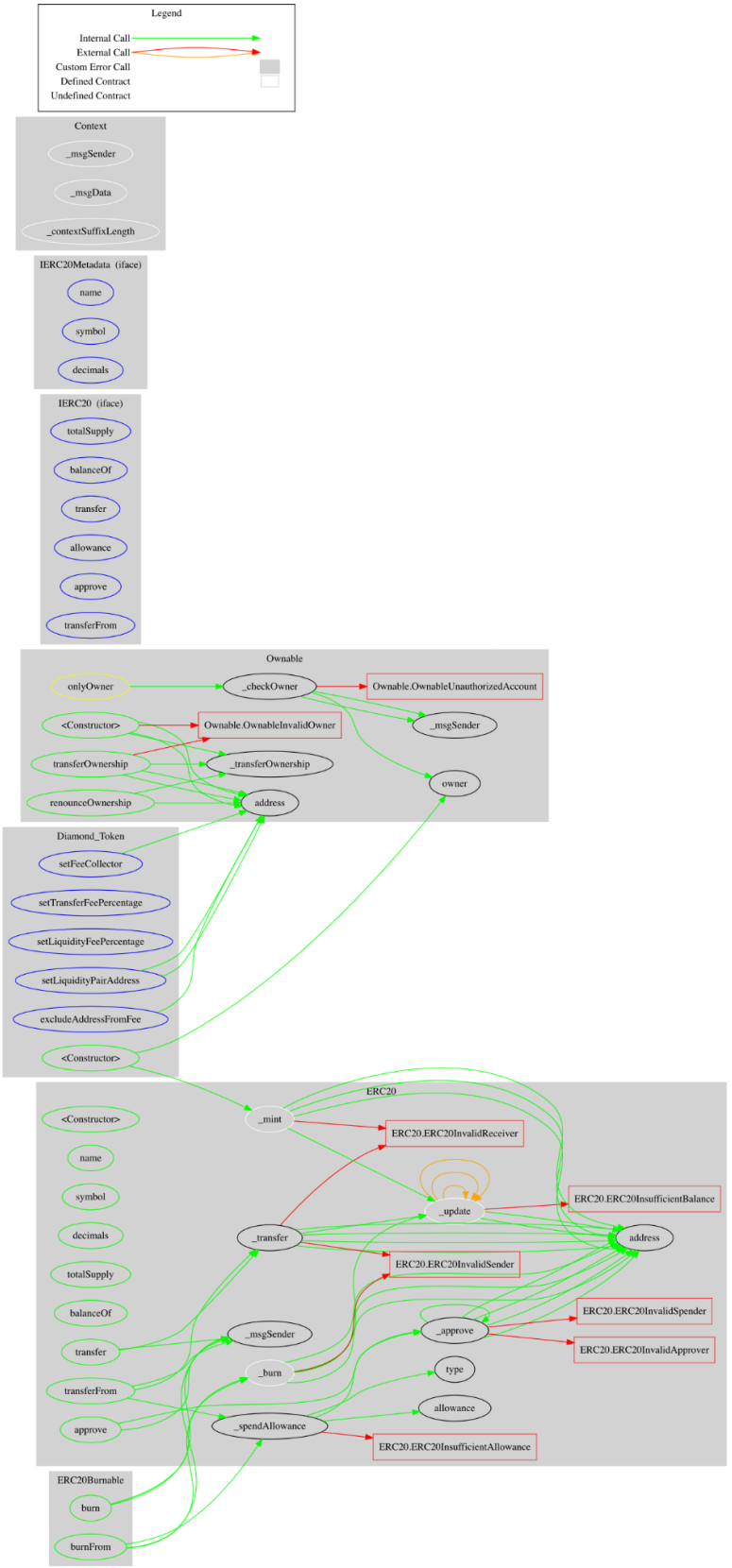
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Diamond_Token	Implementation	ERC20, Ownable		
		Public	✓	ERC20 Ownable
	setTransferFeePercentage	External	✓	onlyOwner
	setLiquidityFeePercentage	External	✓	onlyOwner
	setFeeCollector	External	✓	onlyOwner
	setLiquidityPairAddress	External	✓	onlyOwner
	excludeAddressFromFee	External	✓	onlyOwner
	_update	Internal	✓	

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Custom Error Call
Defined Contract
Undefined Contract
Context
_msgSender
_msgData
_contextSuffixLength
IERC20Metadata (iface)
name
symbol
decimals
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
Ownable
onlyOwner
_checkOwner
Ownable.OwnableUnauthorizedAccount
<Constructor>
Ownable.OwnableInvalidOwner
_msgSender
transferOwnership
_transferOwnership
owner
renounceOwnership
address
Diamond_Token
setFeeCollector
setTransferFeePercentage
setLiquidityFeePercentage
setLiquidityPairAddress
excludeAddressFromFee
<Constructor>
ERC20
<Constructor>
_mint
ERC20.ERC20InvalidReceiver
name
symbol
_update
ERC20.ERC20InsufficientBalance
decimals
_transfer
address
totalSupply
ERC20.ERC20InvalidSender
balanceOf
ERC20.ERC20InvalidSpender
transfer
_msgSender
_approve
ERC20.ERC20InvalidApprover
_burn
type
transferFrom
allowance
approve
_spendAllowance
ERC20.ERC20InsufficientAllowance
ERC20Burnable
burn
burnFrom

Summary

Diamond Token contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. Diamond Token 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. There is also a limit of max 10% buy and sell fees and 5% transfer fees.

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

cyberscope.io