AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Happy

DATED : 30 May 23'

AUDIT SUMMARY

Project name – Happy

Date: 30 May, 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- BSC Test Network:
All tests were conducted on the BSC Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.

3- Slither :
The code has undergone static analysis using Slither.

Testnet version:
The tests were performed using the contract deployed on the BSC Testnet, which can be found at the following address:
https://testnet.bscscan.com/token/0x32ac41dCE80eb383E71E94dD203082177ecBD0f2

Token Information

Token Name : Happy

Token Symbol: HAPPY

Decimals: 9

Token Supply: 888.888888888

Token Address:
0xb8502F3cD879956D07D191f312a3EF2D67FD4eb6

Checksum:
84c0a8f3e6360e9a865371f9db56e473a4832d1a

Owner:
0xE65d5CdCc1BeB2A54Cfc49A561DC3AC766331353
(at time of writing the audit)

Deployer:
0xE65d5CdCc1BeB2A54Cfc49A561DC3AC766331353

TOKEN OVERVIEW

Fees:

Buy Fees: 0-10%

Sell Fees: 0-10%

Transfer Fees: 0-10%

Fees Privilege: Owner

Ownership: yes

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Privileges: Initial distribution of the tokens

- excluding from fees

- including in fees

- modyfting fees

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

- ✅ Return values of low-level calls
- ✅ Private modifier
- ✅ Multiple Sends
- ✅ Using Suicide
- ✅ Gas Limitand Loops
- ✅ Address hardcoded
- ✅ Exception Disorder
- ✅ Using inline assembly
- ✅ Divide before multiply
- ✅ Missing Zero Address Validation
- ✅ Compiler version not fixed

- ✅ **Gasless Send**
- ✅ Using block.timestamp
- ✅ Re-entrancy
- ✅ Tautology or contradiction
- ✅ Timestamp Dependence
- ✅ Revert/require functions
- ✅ Use of tx.origin
- ✅ Integer overflow/underflow
- ✅ Dangerous strict equalities
- ✅ Using SHA3
- ✅ 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
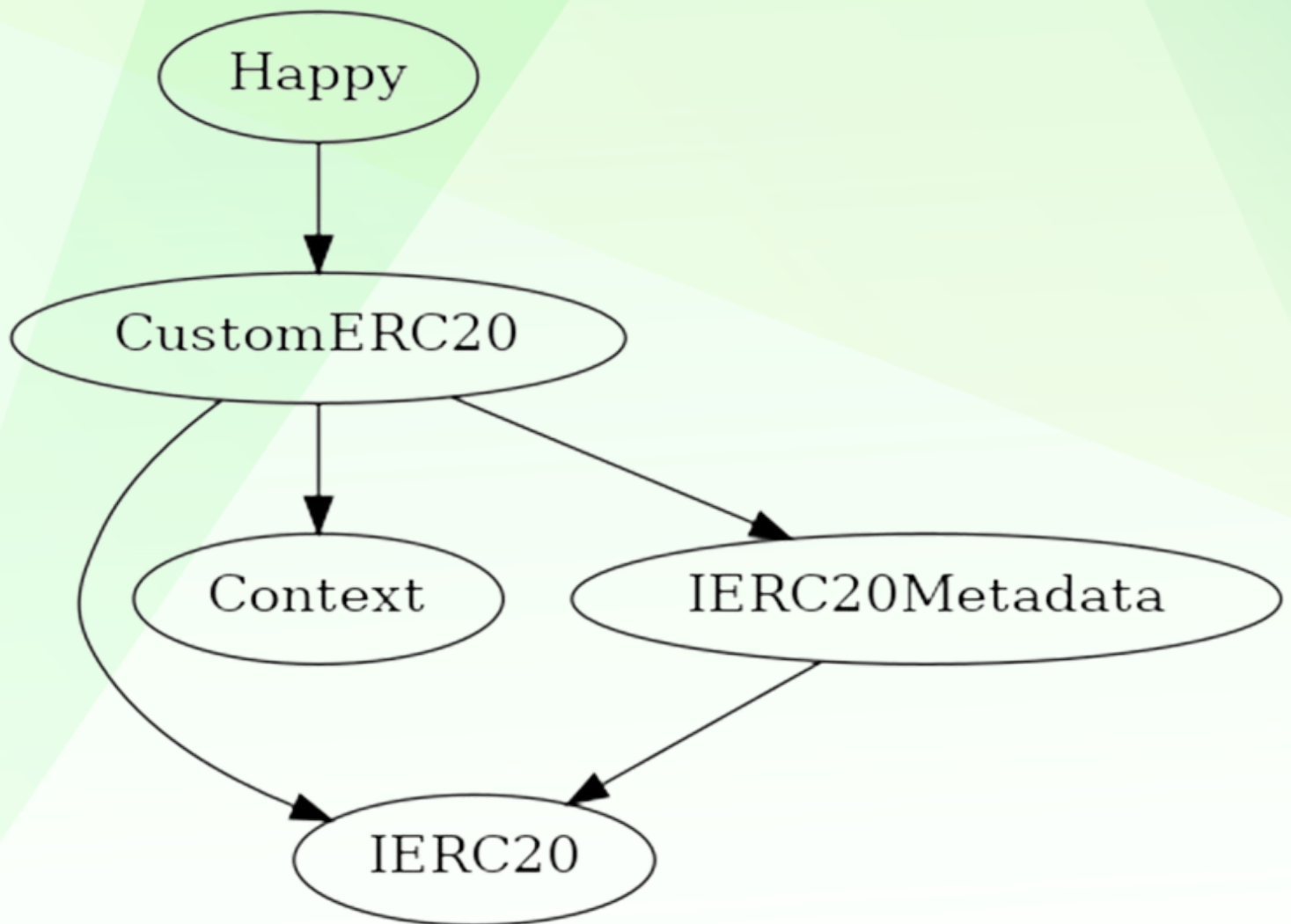
POINTS TO NOTE

- Owner is not able to change buy/sell/transfer fees higher than 10%
- 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

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
└	totalSupply	External ❗	|NO❗	
└	balanceOf	External ❗	|NO❗	
└	transfer	External ❗	🔴 |NO❗	
└	allowance	External ❗	|NO❗	
└	approve	External ❗	🔴 |NO❗	
└	transferFrom	External ❗	🔴 |NO❗	
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
IERC20Metadata	Interface	IERC20		
└	name	External ❗	|NO❗	
└	symbol	External ❗	|NO❗	
└	decimals	External ❗	|NO❗	
CustomERC20	Implementation	Context, IERC20, IERC20Metadata		
└	changeMarketingAddress	Public ❗	🔴 |NO❗	
└	changeMarketingFee	Public ❗	🔴 |NO❗	
└	excludeFromFee	Public ❗	🔴 |NO❗	
└	rescueETH	Public ❗	🔴 |NO❗	
└	rescueAnyerc20Tokens	Public ❗	🔴 |NO❗	
└	\<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 🔒	🔴	
└	_mint	Internal 🔒	🔴	
└	_burn	Internal 🔒	🔴	
└	_approve	Internal 🔒	🔴	

CONTRACT ASSESMENT

└	_beforeTokenTransfer	Internal 🔒	🔴	
└	_afterTokenTransfer	Internal 🔒	🔴	
Happy	Implementation	CustomERC20		
└	<Constructor>	Public ❗	🔴	CustomERC20

Legend

Symbol	Meaning
🔴	Function can modify state
💲	Function is payable

STATIC ANALYSIS

```
Happy.constructor(string,string,uint256).name (contracts/Token.sol#583) shadows:
        - CustomERC20.name() (contracts/Token.sol#238-240) (function)
        - IERC20Metadata.name() (contracts/Token.sol#129) (function)
Happy.constructor(string,string,uint256).symbol (contracts/Token.sol#584) shadows:
        - CustomERC20.symbol() (contracts/Token.sol#246-248) (function)
        - IERC20Metadata.symbol() (contracts/Token.sol#134) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

CustomERC20._transfer(address,address,uint256) (contracts/Token.sol#424-464) compares to a boolean constant:
        -marketingFee == 0 || isExcludedFromFee[sender] == true || isExcludedFromFee[recipient] == true (contracts/Token.sol#443-445)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Context._msgData() (contracts/Token.sol#113-115) is never used and should be removed
CustomERC20._burn(address,uint256) (contracts/Token.sol#498-513) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.17 (contracts/Token.sol#8) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.20 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Parameter CustomERC20.changeMarketingAddress(address)._newAddr (contracts/Token.sol#184) is not in mixedCase
Parameter CustomERC20.rescueAnyerc20Tokens(address,address,uint256)._tokenAddr (contracts/Token.sol#208) is not in mixedCase
Parameter CustomERC20.rescueAnyerc20Tokens(address,address,uint256)._to (contracts/Token.sol#209) is not in mixedCase
Parameter CustomERC20.rescueAnyerc20Tokens(address,address,uint256)._amount (contracts/Token.sol#210) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

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

1- Adding liquidity (passed):

https://testnet.bscscan.com/tx/0x49cfb47bd2ebc94341d7122805dd31cd29a0f818155c51d8d9f5b0b88b4d4a61

2- Buying (0% tax) (passed):

https://testnet.bscscan.com/tx/0xb94c2fccde903e871038cbb4a922faa57a5a8e95693e926301437ff6e05a60b0

3- Selling (0% tax) (passed):

https://testnet.bscscan.com/tx/0xbe715f898afbf770c7b9e9b8cbad54d47795ff9c79053ad4481c7de95e8e7e9f

4- Transferring 0% tax) (passed):

https://testnet.bscscan.com/tx/0x6d2ebf2652891fe1bf3110f2d6798c4356dd6a6a814783506c12b4bd79051c6d

5- Buying when not excluded from fees (0-10% tax) (passed):

https://testnet.bscscan.com/tx/0x856066f130c2930bc736115bef0089e6b86e416c5290a759c96ebc46d006901d

6- Selling when not excluded from fees (0-10% tax) (passed):

https://testnet.bscscan.com/tx/0x1553f52f42a4c544d7b8ff8500641dbdc8bcfc9dc276c115acf2c330f904bf14

FUNCTIONAL TESTING

7- Transferring when not excluded from fees (0-10% tax) (passed):

https://testnet.bscscan.com/tx/0x2099ac860f6e18c235e9b41e68d48f3d8dc5b2ee8421fb90b954723afc9a7eb9

MANUAL 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