auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

CHIHUAHUA INU

DATED : 20 July 23'

MANUAL TESTING

Centralization – Trades are disabled by default

Severity: High
function: StartTrading
Status: Not Resolved
Overview:
Owner must enable trades manually, otherwise holders wont be able to buy/sell/transfer tokens.

```
function StartTrading() external onlyOwner {
    launch = true;
    InitialBlockNo = block.number;
}
```

Suggestion
It is suggested to either enable trades prior to presale, or transfer ownership of the contract to a trusted 3rd party like a certified Pinksale safu developer.

MANUAL TESTING

Centralization – Excessive fees

Severity: High
function: EditTaxes
Status: Not Resolved
Overview:
Owner is able to set up to 70% tax for buy or sells.

```
    function EditTaxes(uint256 newBuyTax, uint256 newSellTax) external onlyOwner
{
        require(newBuyTax + newSellTax <= 70, "Tax too high");
        buyTax = newBuyTax;
        sellTax = newSellTax;
    }
```

Suggestion

Ensure that fees are within a safe and reasonable range. Usually 0-10% (For each type of tax) is suggested by Pinksale safu criteria.

AUDIT SUMMARY

Project name – CHIHUAHUA INU

Date: 20 July, 2023

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

Audit Status: **Passed with High Risk**

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	2	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/0x16f16c8b3F7CAF5b0f0792bb02F57e9DB6FB3761

Token Information

Token Name : CHIHUAHUA INU

Token Symbol: CHINU

Decimals: 18

Token Supply: 1,000,000,000

Token Address:
0x8DE62Ab305FD50BDfbdb2Ce3b9fd06aE0D2bE4AD

Checksum:
573bf4eed1106f9b77a1a8ef24d9cac562ed5992

Owner:
0x2b25B877179397fFDD8fBB2c4506C71533c06122
(at time of writing the audit)

Deployer:
0x2b25B877179397fFDD8fBB2c4506C71533c06122

TOKEN OVERVIEW

Fees:

Buy Fees: 0-70%

Sell Fees: 0-70%

Transfer Fees: 0%

Fees Privilege: owner

Ownership: owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: no

Blacklist: No

Other Privileges: Initial distribution of the tokens

modifying fees

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

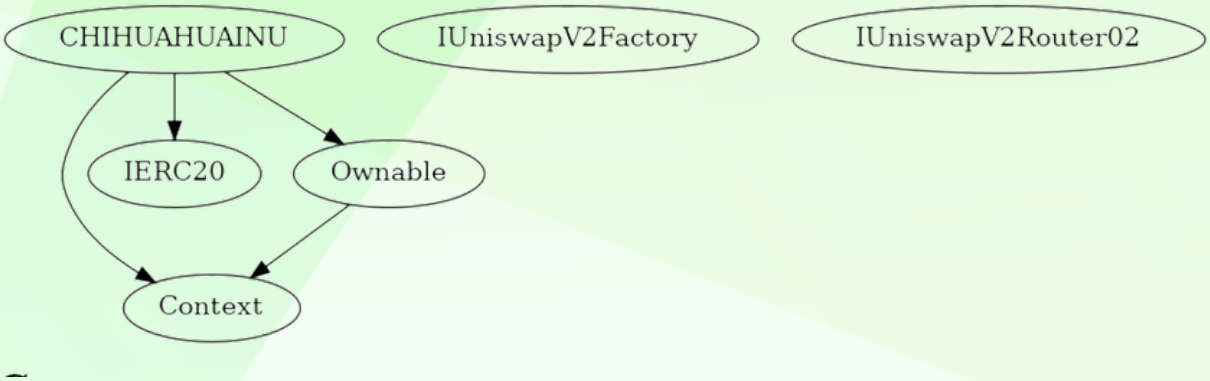
CLASSIFICATION OF RISK

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	2
Medium-Risk	0
Low-Risk	0
Gas Optimization / Suggestions	0

INHERITANCE TREE



POINTS TO NOTE

- Owner is able to change the current fee structure (0-70% for buy and sell and 0% for transfers)
- Owner is not able to blacklist an arbitrary address.
- Owner is not able to set max wallet/transfer/buy/sell
- 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 🔒		
IERC20	Interface			
└	totalSupply	External ❗		NO ❗
└	balanceOf	External ❗		NO ❗
└	transfer	External ❗	🛑	NO ❗
└	allowance	External ❗		NO ❗
└	approve	External ❗	🛑	NO ❗
└	transferFrom	External ❗	🛑	NO ❗
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO ❗
└	owner	Public ❗		NO ❗
└	transferOwnership	Public ❗	🛑	onlyOwner
└	_transferOwnership	Internal 🔒	🛑	
└	renounceOwnership	Public ❗	🛑	onlyOwner
IUniswapV2Factory	Interface			
└	createPair	External ❗	🛑	NO ❗
IUniswapV2Router02	Interface			
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	factory	External ❗		NO ❗
└	WETH	External ❗		NO ❗

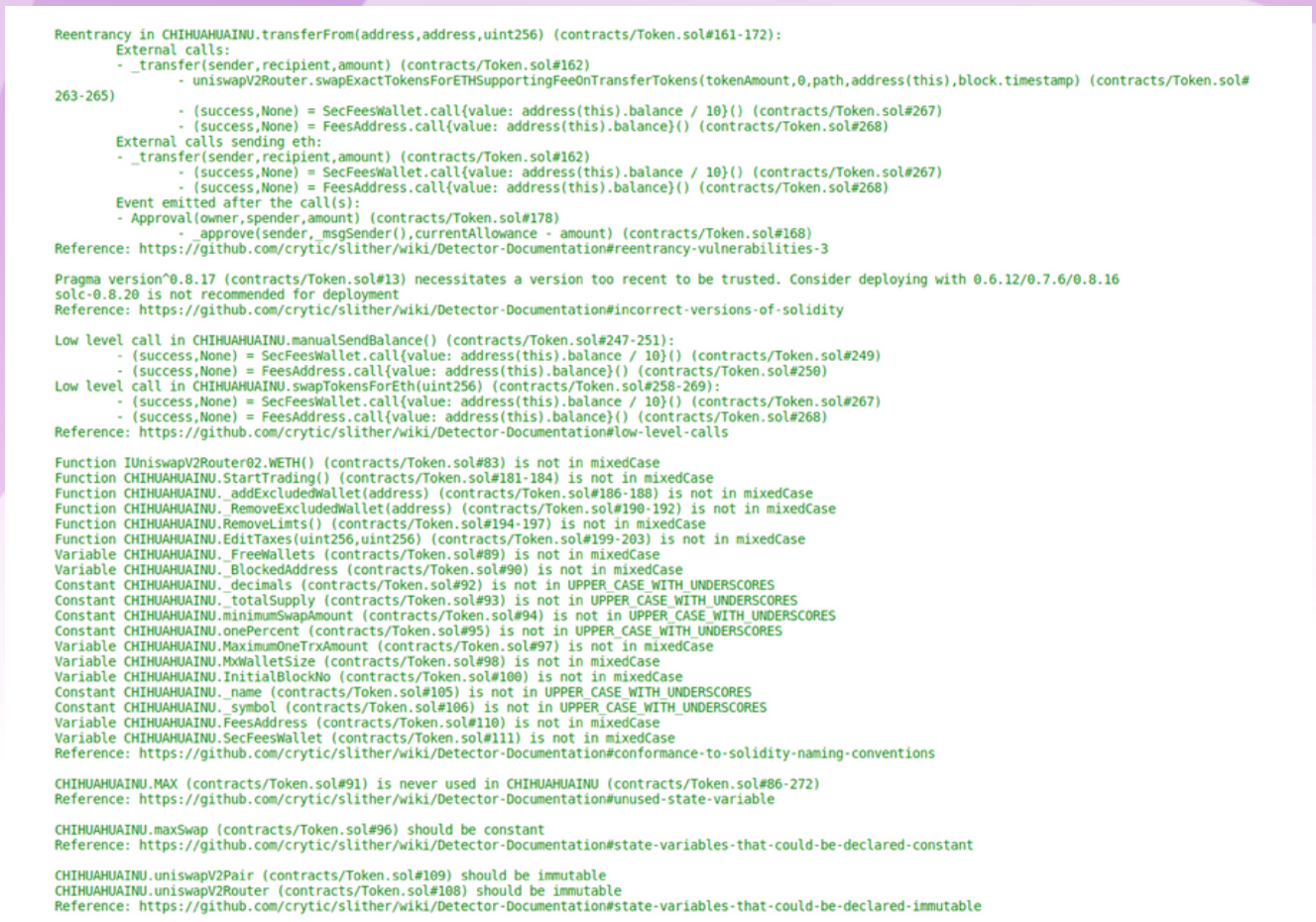
CONTRACT ASSESMENT

CHIHUAHUAINU	Implementation	Context, IERC20, Ownable		
└	<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 ❗
└	_approve	Private 🔐	🛑		
└	StartTrading	External ❗	🛑	onlyOwner
└	_addExcludedWallet	External ❗	🛑	onlyOwner
└	_RemoveExcludedWallet	External ❗	🛑	onlyOwner
└	RemoveLimts	External ❗	🛑	onlyOwner
└	EditTaxes	External ❗	🛑	onlyOwner
└	_tokenTransfer	Private 🔐	🛑		
└	_transfer	Private 🔐	🛑		
└	manualSendBalance	External ❗	🛑	onlyOwner
└	manualSwapTokens	External ❗	🛑	onlyOwner
└	swapTokensForEth	Private 🔐	🛑		
└	<Receive Ether>	External ❗	💵	NO ❗

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS



```
Reentrancy in CHIHUAHUAINU.transferFrom(address,address,uint256) (contracts/Token.sol#161-172):
        External calls:
        - _transfer(sender,recipient,amount) (contracts/Token.sol#162)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (contracts/Token.sol#
263-265)
                - (success,None) = SecFeesWallet.call{value: address(this).balance / 10}() (contracts/Token.sol#267)
                - (success,None) = FeesAddress.call{value: address(this).balance}() (contracts/Token.sol#268)
        External calls sending eth:
        - _transfer(sender,recipient,amount) (contracts/Token.sol#162)
                - (success,None) = SecFeesWallet.call{value: address(this).balance / 10}() (contracts/Token.sol#267)
                - (success,None) = FeesAddress.call{value: address(this).balance}() (contracts/Token.sol#268)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (contracts/Token.sol#178)
                - _approve(sender,_msgSender(),currentAllowance - amount) (contracts/Token.sol#168)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Pragma version^0.8.17 (contracts/Token.sol#13) 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

Low level call in CHIHUAHUAINU.manualSendBalance() (contracts/Token.sol#247-251):
        - (success,None) = SecFeesWallet.call{value: address(this).balance / 10}() (contracts/Token.sol#249)
        - (success,None) = FeesAddress.call{value: address(this).balance}() (contracts/Token.sol#250)
Low level call in CHIHUAHUAINU.swapTokensForEth(uint256) (contracts/Token.sol#258-269):
        - (success,None) = SecFeesWallet.call{value: address(this).balance / 10}() (contracts/Token.sol#267)
        - (success,None) = FeesAddress.call{value: address(this).balance}() (contracts/Token.sol#268)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IUniswapV2Router02.WETH() (contracts/Token.sol#83) is not in mixedCase
Function CHIHUAHUAINU.StartTrading() (contracts/Token.sol#181-184) is not in mixedCase
Function CHIHUAHUAINU._addExcludedWallet(address) (contracts/Token.sol#186-188) is not in mixedCase
Function CHIHUAHUAINU._RemoveExcludedWallet(address) (contracts/Token.sol#190-192) is not in mixedCase
Function CHIHUAHUAINU.RemoveLimts() (contracts/Token.sol#194-197) is not in mixedCase
Function CHIHUAHUAINU.EditTaxes(uint256,uint256) (contracts/Token.sol#199-203) is not in mixedCase
Variable CHIHUAHUAINU._FreeWallets (contracts/Token.sol#89) is not in mixedCase
Variable CHIHUAHUAINU._BlockedAddress (contracts/Token.sol#90) is not in mixedCase
Constant CHIHUAHUAINU._decimals (contracts/Token.sol#92) is not in UPPER_CASE_WITH_UNDERSCORES
Constant CHIHUAHUAINU._totalSupply (contracts/Token.sol#93) is not in UPPER_CASE_WITH_UNDERSCORES
Constant CHIHUAHUAINU.minimumSwapAmount (contracts/Token.sol#94) is not in UPPER_CASE_WITH_UNDERSCORES
Constant CHIHUAHUAINU.onePercent (contracts/Token.sol#95) is not in UPPER_CASE_WITH_UNDERSCORES
Variable CHIHUAHUAINU.MaximumOneTrxAmount (contracts/Token.sol#97) is not in mixedCase
Variable CHIHUAHUAINU.MxWalletSize (contracts/Token.sol#98) is not in mixedCase
Variable CHIHUAHUAINU.InitialBlockNo (contracts/Token.sol#100) is not in mixedCase
Constant CHIHUAHUAINU._name (contracts/Token.sol#105) is not in UPPER_CASE_WITH_UNDERSCORES
Constant CHIHUAHUAINU._symbol (contracts/Token.sol#106) is not in UPPER_CASE_WITH_UNDERSCORES
Variable CHIHUAHUAINU.FeesAddress (contracts/Token.sol#110) is not in mixedCase
Variable CHIHUAHUAINU.SecFeesWallet (contracts/Token.sol#111) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

CHIHUAHUAINU.MAX (contracts/Token.sol#91) is never used in CHIHUAHUAINU (contracts/Token.sol#86-272)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

CHIHUAHUAINU.maxSwap (contracts/Token.sol#96) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

CHIHUAHUAINU.uniswapV2Pair (contracts/Token.sol#109) should be immutable
CHIHUAHUAINU.uniswapV2Router (contracts/Token.sol#108) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

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

No major issues were found in the output

FUNCTIONAL TESTING

1- Adding liquidity (passed):

https://testnet.bscscan.com/tx/0x524b7269065135adb93ce4c897402b76f9380000e5f9c96e830fa3999d207779

2- Buying when excluded from fees (0% tax) (passed):

https://testnet.bscscan.com/tx/0x7c83b1d19642068be744ee81e37b54df43fd72fa7effc8464837117d6e1b72cf

3- Selling when excluded from fees (0% tax) (passed):

https://testnet.bscscan.com/tx/0xb69df1fc27a5b065ed82df2ba13b902c06cad5806d26eba6aa7ba6424492c3ce

4- Transferring when excluded from fees (0% tax) (passed):

https://testnet.bscscan.com/tx/0x6033e27e68044478061d6386b10cc1db331c8c8ad5c3b189883fc6df4622366c

5- Buying when not excluded from fees (0-70% tax) (passed):

https://testnet.bscscan.com/tx/0x151757482027bde6e90d69d33b45e6d72902d49d433b71c8c7736f265125e5cd

6- Selling when not excluded from fees (0-70% tax) (passed):

https://testnet.bscscan.com/tx/0x7bec2fe342e46a611a64bfb1ff25ad90a94560d0f029d0edd89588cf72947405

FUNCTIONAL TESTING

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

https://testnet.bscscan.com/tx/0x63672fd136eb115bdd3a61974ebb1a2514e628e43d23ff6d3c3a9faaacd250c9

8- Internal swap (passed):

All of below features can be seen in the given tx

- Fund wallet and SEC wallet received BNB

https://testnet.bscscan.com/address/0x806df78021b45f0901568d4292a3f159faaf23f2#internaltx

https://testnet.bscscan.com/address/0x2b25b877179397ffdd8fbb2c4506c71533c06122#internaltx

MANUAL TESTING

Centralization – Trades are disabled by default

Severity: High
function: StartTrading
Status: Not Resolved
Overview:
Owner must enable trades manually, otherwise holders wont be able to buy/sell/transfer tokens.

```
function StartTrading() external onlyOwner {
    launch = true;
    InitialBlockNo = block.number;
}
```

Suggestion
It is suggested to either enable trades prior to presale, or transfer ownership of the contract to a trusted 3rd party like a certified Pinksale safu developer.

MANUAL TESTING

Centralization – Excessive fees

Severity: High
function: EditTaxes
Status: Not Resolved
Overview:
Owner is able to set up to 70% tax for buy or sells.

```
    function EditTaxes(uint256 newBuyTax, uint256 newSellTax) external onlyOwner
{
        require(newBuyTax + newSellTax <= 70, "Tax too high");
        buyTax = newBuyTax;
        sellTax = newSellTax;
    }
```

Suggestion

Ensure that fees are within a safe and reasonable range. Usually 0-10% (For each type of tax) is suggested by Pinksale safu criteria.

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