auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

UNIBOT2

DATED : 30 July 23'

AUDIT SUMMARY

Project name –UNIBOT2

Date: 30 July, 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	1
Acknowledged	0	0	0	0	0
Resolved	0	1	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/0x634b275F6A83F8E1FC25581aCB08086213be4Fe4

Token Information

Token Name : Unibot 2.0

Token Symbol: UNIBOT2

Decimals: 18

Token Supply: 100,000,000

Token Address:
0xA3CF968B08DAd9f41275a99bbdda8FF002eb0ACa

Checksum:
fb4b0aca845e30bcf71fb9972e23e762248b1ac1

Owner:
0xE26F72ba141Db9b9f3066e04c1C1a25502bae9C5
(at time of writing the audit)

Deployer:
0xE26F72ba141Db9b9f3066e04c1C1a25502bae9C5

TOKEN OVERVIEW

Fees:

Buy Fees: 0%

Sell Fees: 0%

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

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

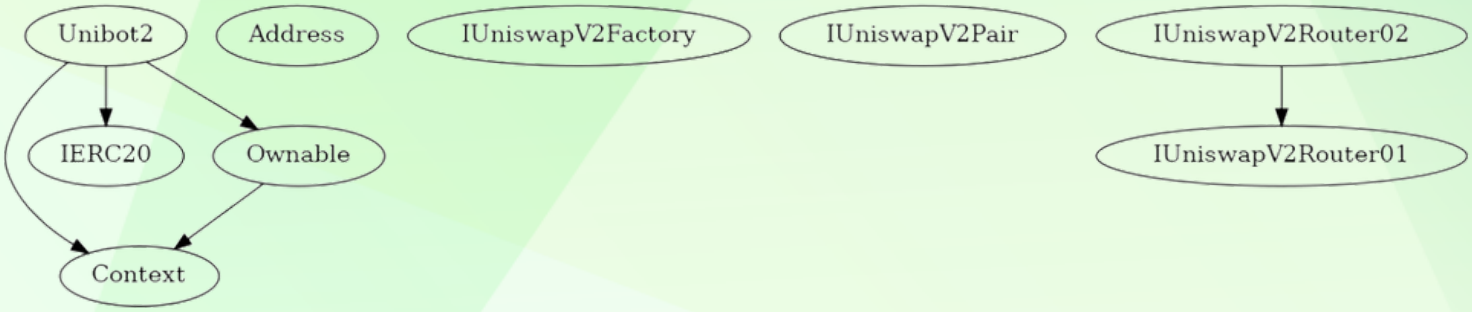
CLASSIFICATION OF RISK

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	1
Medium-Risk	0
Low-Risk	0
Gas Optimization / Suggestions	1

INHERITANCE TREE



POINTS TO NOTE

- Owner is not able to set fee on buy/sell/transfers (0% all fees)
- Owner is not able to blacklist an address
- Owner is not able to disable buy/sell/transfers
- Owner is not able to set max wallet limit and minimum wallet limits
- Owner is not able to mint new tokens

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
IERC20	Interface			
└	totalSupply	External ❗		NO ❗
└	balanceOf	External ❗		NO ❗
└	transfer	External ❗	🛑	NO ❗
└	allowance	External ❗		NO ❗
└	approve	External ❗	🛑	NO ❗
└	transferFrom	External ❗	🛑	NO ❗
Address	Library			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionStaticCall	Internal 🔒		
└	functionStaticCall	Internal 🔒		
└	functionDelegateCall	Internal 🔒	🛑	
└	functionDelegateCall	Internal 🔒	🛑	
└	verifyCallResultFromTarget	Internal 🔒		
└	verifyCallResult	Internal 🔒		
└	_revert	Private 🔐		
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO ❗
└	owner	Public ❗		NO ❗
└	_checkOwner	Internal 🔒		
└	renounceOwnership	Public ❗	🛑	onlyOwner

CONTRACT ASSESMENT

└	transferOwnership	Public ❗	🛑	onlyOwner
└	_transferOwnership	Internal 🔒	🛑	
IUniswapV2Factory	Interface			
└	feeTo	External ❗	|NO ❗	
└	feeToSetter	External ❗	|NO ❗	
└	getPair	External ❗		|NO ❗	
└	allPairs	External ❗	|NO ❗	
└	allPairsLength	External ❗	|NO ❗	
└	createPair	External ❗	🛑 |NO ❗	
└	setFeeTo	External ❗	🛑 |NO ❗	
└	setFeeToSetter	External ❗	🛑 |NO ❗	
IUniswapV2Pair	Interface			
└	name	External ❗	|NO ❗	
└	symbol	External ❗	|NO ❗	
└	decimals	External ❗		|NO ❗	
└	totalSupply	External ❗		|NO ❗	
└	balanceOf	External ❗		|NO ❗	
└	allowance	External ❗		|NO ❗	
└	approve	External ❗	🛑 |NO ❗	
└	transfer	External ❗	🛑 |NO ❗	
└	transferFrom	External ❗	🛑 |NO ❗	
└	DOMAIN_SEPARATOR	External ❗		|NO ❗	
└	PERMIT_TYPEHASH	External ❗		|NO ❗	
└	nonces	External ❗	|NO ❗	
└	permit	External ❗	🛑 |NO ❗	
└	MINIMUM_LIQUIDITY	External ❗		|NO ❗	
└	factory	External ❗	|NO ❗	
└	token0	External ❗		|NO ❗	
└	token1	External ❗		|NO ❗	
└	getReserves	External ❗	|NO ❗	
└	price0CumulativeLast	External ❗		|NO ❗	
└	price1CumulativeLast	Exte ❗		|NO ❗	
└	kLast	External ❗		NO ❗
└	burn	External ❗	🛑

CONTRACT ASSESMENT

└	swap	External ❗	🛑	NO ❗
└	skim	External ❗	🛑	NO ❗
└	sync	External ❗	🛑	NO ❗
└	initialize	External ❗	🛑	NO ❗

IUniswapV2Router01	Interface				
└	factory	External ❗		NO ❗
└	WETH	External ❗		NO ❗
└	addLiquidity	External ❗	🛑	NO ❗
└	addLiquidityETH	External ❗	💵	NO ❗
└	removeLiquidity	External ❗	🛑	NO ❗
└	removeLiquidityETH	External ❗	🛑	NO ❗
└	removeLiquidityWithPermit	External ❗	🛑	NO ❗
└	removeLiquidityETHWithPermit	External ❗	🛑	NO ❗
└	swapExactTokensForTokens	External ❗	🛑	NO ❗
└	swapTokensForExactTokens	External ❗	🛑	NO ❗
└	swapExactETHForTokens	External ❗	💵	NO ❗
└	swapTokensForExactETH	External ❗	🛑	NO ❗
└	swapExactTokensForETH	External ❗	🛑	NO ❗
└	swapETHForExactTokens	External ❗	💵	NO ❗
└	quote	External ❗		NO ❗
└	getAmountOut	External ❗		NO ❗
└	getAmountIn	External ❗		NO ❗
└	getAmountsOut	External ❗		NO ❗
└	getAmountsIn	External ❗		NO ❗

IUniswapV2Router02	Interface	IUniswapV2Router01			
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗

Unibot2	Implementation	Context, IERC20, Ownable			
└	<Constructor>	Public ❗	🛑	NO ❗
└	totalSupply	Public ❗		NO ❗

CONTRACT ASSESMENT

└	balanceOf	Public ❗	|NO ❗
└	transfer	Public ❗	🛑 |NO ❗
└	allowance	Public ❗	|NO ❗
└	approve	Public ❗	🛑 |NO ❗
└	transferFrom	Public ❗	🛑 |NO ❗
└	increaseAllowance	Public ❗	🛑 |NO ❗
└	decreaseAllowance	Public ❗	🛑 |NO ❗
└	_approve	Private 🔐	🛑	
└	_transfer	Private 🔐	🛑	
└	swapAndLiquify	Public ❗	🛑	lockTheSwap
└	swapTokensForEth	Private 🔐	🛑	
└	_tokenTransfer	Private 🔐	🛑	
└	isExcludedFromFee	External ❗	|NO ❗
└	excludeFromFee	External ❗	🛑	onlyOwner
└	includeInFee	External ❗	🛑	onlyOwner
└	setTokensToSwap	External ❗	🛑	onlyOwner
└	setSwapAndLiquifyEnabled	External ❗	🛑	onlyOwner
└	setMarketingWallet	External ❗	🛑	onlyOwner
└	transferToAddressETH	Private 🔐	🛑	
└	<Receive Ether>	External ❗	💲 |NO ❗
└	swapETHForTokens	Private 🔐	🛑	
└	recoverETHfromContract	External ❗	🛑	onlyOwner
└	recoverTokensFromContract	External ❗	🛑	onlyOwner
└	enableTrading	External ❗	🛑	onlyOwner

Legend

Symbol	Meaning
🛑	Function can modify state
💲	Function is payable

STATIC ANALYSIS

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

XSHIB.excludeFromFee(address) (contracts/Token.sol#703-707) compares to a boolean constant:
        -require(bool,string)(exemptFee[account] != true,Account is already excluded) (contracts/Token.sol#704)
XSHIB.includeFromFee(address) (contracts/Token.sol#709-713) compares to a boolean constant:
        -require(bool,string)(exemptFee[account] != false,Account is already included) (contracts/Token.sol#710)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Context._msgData() (contracts/Token.sol#23-26) 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#16) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.21 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (contracts/Token.sol#334-342):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#337)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Variable ERC20._balances (contracts/Token.sol#72) is not in mixedCase
Variable ERC20._allowances (contracts/Token.sol#74) is not in mixedCase
Function IRouter.WETH() (contracts/Token.sol#392) is not in mixedCase
Event XSHIBincludeFromFeeUpdated(address) (contracts/Token.sol#446) is not in CapWords
Function XSHIB.Liquify(uint256,XSHIB.Taxes) (contracts/Token.sol#585-623) is not in mixedCase
Function XSHIB.SetBuyTaxes(uint256,uint256) (contracts/Token.sol#662-668) is not in mixedCase
Parameter XSHIB.SetBuyTaxes(uint256,uint256)._marketing (contracts/Token.sol#663) is not in mixedCase
Parameter XSHIB.SetBuyTaxes(uint256,uint256)._liquidity (contracts/Token.sol#664) is not in mixedCase
Function XSHIB.SetSellTaxes(uint256,uint256) (contracts/Token.sol#670-676) is not in mixedCase
Parameter XSHIB.SetSellTaxes(uint256,uint256)._marketing (contracts/Token.sol#671) is not in mixedCase
Parameter XSHIB.SetSellTaxes(uint256,uint256)._liquidity (contracts/Token.sol#672) is not in mixedCase
Function XSHIB.UpdateMarketingWallet(address) (contracts/Token.sol#678-683) is not in mixedCase
Parameter XSHIB.UpdateMarketingWallet(address)._newWallet (contracts/Token.sol#678) is not in mixedCase
Parameter XSHIB.updateLiquidityTreshhold(uint256).new_amount (contracts/Token.sol#685) is not in mixedCase
Constant XSHIB.Contract_Version (contracts/Token.sol#428) is not in UPPER_CASE_WITH_UNDERSCORES
Constant XSHIB.Contract_Dev (contracts/Token.sol#429) is not in UPPER_CASE_WITH_UNDERSCORES
Constant XSHIB.Contract_Edition (contracts/Token.sol#430) is not in UPPER_CASE_WITH_UNDERSCORES
Constant XSHIB.deadWallet (contracts/Token.sol#433-434) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#24)" inContext (contracts/Token.sol#18-27)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

XSHIB.constructor() (contracts/Token.sol#458-476) uses literals with too many digits:
        - _tokengeneration(msg.sender,100000000 * 10 ** decimals()) (contracts/Token.sol#459)
XSHIB.updateLiquidityTreshhold(uint256) (contracts/Token.sol#685-695) uses literals with too many digits:
        - require(bool,string)(new_amount <= 1000000,Swap threshold amount should be lower or equal to 1% of tokens) (contracts/Token.sol#686-689)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

XSHIB.pair (contracts/Token.sol#419) should be immutable
XSHIB.router (contracts/Token.sol#418) 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/0x08e4d8b47f21b374c24b36f482405788e69e2085a4f06eb95ef8620515194ec2

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

https://testnet.bscscan.com/tx/0x9b5219a3ce827ed6eb8cccd5926233d26c04d591bfc409124820409c818dc634

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

https://testnet.bscscan.com/tx/0x9b5219a3ce827ed6eb8cccd5926233d26c04d591bfc409124820409c818dc634

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

https://testnet.bscscan.com/tx/0xc5b4bca0f8d4b3c17b7b9d388d822d967045a9885edc03e1bd3dcb739310f108

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

https://testnet.bscscan.com/tx/0xf74a79461b231b8d5d922696e08946fd0fcacb502d92594532ba85b5a589ae35

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

https://testnet.bscscan.com/tx/0x158205313e6819b8b76adfa97a9e7f82c220c7299f131255014eaab9b7a444da

FUNCTIONAL TESTING

7- Transferring (0% tax) (passed):

https://testnet.bscscan.com/tx/0xf189cc93b3f319e1dd097fe6c932046b33d4b0df5b9ef153fd8e56c41f3a6ced

MANUAL TESTING

Centralization – Enabling Trades

Severity: High

function: enableTrading

Status: Resolved (Trades are enabled)

Overview:

Owner of the contract must enable trades manually for investors, otherwise no one would be able to buy/sell/transfer their tokens (even owner of whitelisted wallets)

```
    function enableTrading() external onlyOwner {
        require(!tradingEnabled, "Trading already enabled.");
        tradingEnabled = true;
        swapAndLiquifyEnabled = true;
        emit AuditLog("We have Enable Trading and Automatic Swaps:",
msg.sender);
    }
```

Suggestion

It's suggested to either enable trades prior to presale, or transfer ownership of the contract to a certified pinsksale safu developer to guarantee enabling of trades.

MANUAL TESTING

Logical – Rejecting ETH and marketing wallet

Severity: Informational
function: enableTrading
Status: Open (not applicable)
Overview:
if Marketing wallet is set to a contract that rejects ether:

```
Contract example {
        receive() external payable {
                revert("ether is not accepted!");
        }
}
```

swapAndLiquify will be reverted which then disables sell/transfers

Suggestion

Ignore whether a call to marketing wallet was successful or not:

```
function transferToAddressETH(address payable recipient, uint256 amount) private {
    (bool succ,) = recipient.call{value: amount}("");
    require(succ, "Transfer failed."); //Delete this line
}
```

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