auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Shiba X

DATED : 9 Sep 23'

High Risk

Logical – Setting swap threshold to zero

Severity: High

function: setSwapTokensAtAmount

Status: Open

Overview:

Owner is able to set swapTokensAtAmount to zero. If swapTokensAtAmount is set to zero, contract will try to perform internal swap with 0 tokens which will revert the transaction

```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner {
    swapTokensAtAmount = newAmount;
}
```

Suggestion

Ensure that swapTokensAtAmount is always greater than zero

```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner {
    require(swapTokensAtAmount > 0);
    swapTokensAtAmount = newAmount;
}
```

AUDIT SUMMARY

Project name – Shiba X

Date: 9 Sep, 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	1	1	0	1
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/0xF2CAFd4375A5345978BE3F97e9f82C1DCe6be536

Token Information

Token Name : Shiba X

Token Symbol: Shiba X

Decimals: 18

Token Supply: 1,000,000,000

Token Address:
0xbAdce7D73Ac5e95266AA76334428a75620D2c741

Checksum:
37688849caf886145a1a84c68eec8ffb8821aa7d

Owner:
0x715179CEcAc15a5cf2e752017ecb9003B3642eC8
(at time of writing the audit)

Deployer:
0x0913cFCA3896Cc67131C8e90FC47Ff5268BDBb48

TOKEN OVERVIEW

Fees:

Buy Fees: 3%

Sell Fees: 4%

Transfer Fees: 0%

Fees Privilege: Static fees

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

- [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

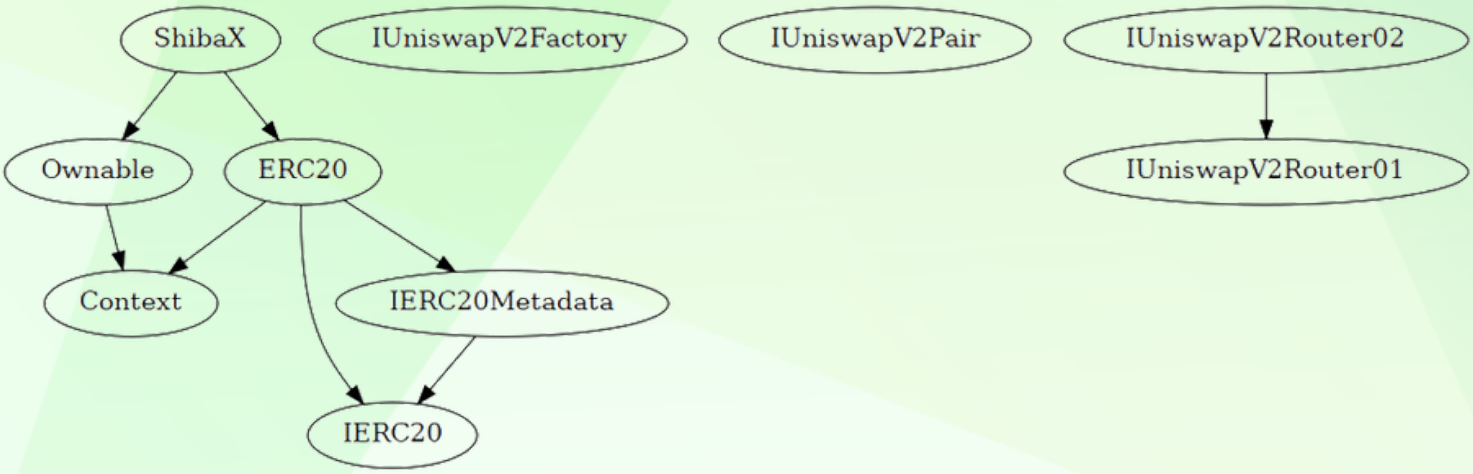
CLASSIFICATION OF RISK

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

INHERITANCE TREE



POINTS TO NOTE

- **Owner is not able to change fee structure (3% buy / 4% sell / 0% transfer)**
- Owner is not able to blacklist an arbitrary wallet
- Owner is not able to disable trades
- Owner is not able to mint new tokens
- Owner is not able to set maximum wallet and maximum buy/sell/transfer limits
- **Owner must enable trades manually**

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 ❗
IERC20Metadata	Interface	IERC20		
	└	name	External ❗		|NO ❗
	└	symbol	External ❗		|NO ❗
	└	decimals	External ❗		|NO ❗
Context	Implementation			
	└	_msgSender	Internal 🔒		
	└	_msgData	Internal 🔒		
Ownable	Implementation	Context		
	└	<Constructor>	Public ❗	🛑|NO ❗
	└	owner	Public ❗		|NO ❗
	└	renounceOwnership	Public ❗	🛑| onlyOwner
	└	transferOwnership	Public ❗	🛑| onlyOwner
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 🔒	🛑|

CONTRACT ASSESMENT

└	_mint	Internal 🔒	🔴	
└	_burn	Internal 🔒	🔴	
└	_approve	Internal 🔒	🔴	
└	_beforeTokenTransfer	Internal 🔒	🔴	
└	_afterTokenTransfer	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	External ❗		NO❗
└	kLast	External ❗		NO❗
└	mint	External ❗	🔴	NO❗
└	burn	External ❗	🔴	NO❗
└	swap	External ❗	🔴	NO❗
└	skim	External ❗	🔴	NO❗

CONTRACT ASSESMENT

└	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❗
ShibaX	Implementation	ERC20, Ownable		
└	<Constructor>	Public ❗	🔴	ERC20
└	<Receive Ether>	External ❗	💵	NO❗
└	claimStuckTokens	External ❗	🔴	onlyOwner
└	isContract	Internal 🔒		
└	sendBNB	Internal 🔒	🔴	
└	_setAutomatedMarketMakerPair	Private 🔐	🔴	
└	excludeFromFees	External ❗	🔴	onlyOwner
└	isExcludedFromFees	Public ❗		NO❗
└	changeMarketingWallet	External ❗	🔴	onlyOwner
└	changeRewardPool	External ❗	🔴	onlyOwner

CONTRACT ASSESMENT

└	setAntibotStatus	External ❗	🛑	onlyOwner
└	_transfer	Internal 🔒	🛑	
└	setSwapEnabled	External ❗	🛑	onlyOwner
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	setSwapWithLimit	External ❗	🛑	onlyOwner
└	swapAndLiquify	Private 🔐	🛑	
└	swapAndSendBNB	Private 🔐	🛑	

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS

```
INFO:Detectors:
Context._msgData() (contracts/Token.sol#53-56) is never used and should be removed
ERC20._burn(address,uint256) (contracts/Token.sol#243-258) is never used and should be removed
ShibaX.isContract(address) (contracts/Token.sol#682-684) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
ShibaX.totalSellFee (contracts/Token.sol#621-622) is set pre-construction with a non-constant function or state variable:
        - liquiditySellFee + marketingSellFee + rewardPoolSellFee
ShibaX.totalBuyFee (contracts/Token.sol#627-628) is set pre-construction with a non-constant function or state variable:
        - liquidityBuyFee + marketingBuyFee + rewardPoolBuyFee
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state
INFO:Detectors:
Pragma version^0.8.17 (contracts/Token.sol#7) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in ShibaX.sendBNB(address,uint256) (contracts/Token.sol#686-697):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#692)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function IUniswapV2Pair.DOMAIN_SEPARATOR() (contracts/Token.sol#349) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (contracts/Token.sol#351) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (contracts/Token.sol#382) is not in mixedCase
Function IUniswapV2Router01.WETH() (contracts/Token.sol#424) is not in mixedCase
Parameter ShibaX.changeMarketingWallet(address)._marketingWallet (contracts/Token.sol#723) is not in mixedCase
Parameter ShibaX.changeRewardPool(address)._address (contracts/Token.sol#732) is not in mixedCase
Parameter ShibaX.setAntibotStatus(bool)._antibotEnabled (contracts/Token.sol#740) is not in mixedCase
Parameter ShibaX.setSwapEnabled(bool)._swapEnabled (contracts/Token.sol#826) is not in mixedCase
Parameter ShibaX.setSwapWithLimit(bool)._swapWithLimit (contracts/Token.sol#838) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Redundant expression "this (contracts/Token.sol#54)" inContext (contracts/Token.sol#48-57)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
INFO:Detectors:
Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#429) is too similar to IUniswapV2Router01.addLiqu
idity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#430)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Detectors:
ShibaX.liquidityBuyFee (contracts/Token.sol#624) should be constant
ShibaX.liquiditySellFee (contracts/Token.sol#618) should be constant
ShibaX.marketingBuyFee (contracts/Token.sol#625) should be constant
ShibaX.marketingSellFee (contracts/Token.sol#619) should be constant
ShibaX.rewardPoolBuyFee (contracts/Token.sol#626) should be constant
ShibaX.rewardPoolSellFee (contracts/Token.sol#620) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
ShibaX.totalBuyFee (contracts/Token.sol#627-628) should be immutable
ShibaX.totalSellFee (contracts/Token.sol#621-622) should be immutable
ShibaX.uniswapV2Pair (contracts/Token.sol#634) should be immutable
ShibaX.uniswapV2Router (contracts/Token.sol#633) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:./contracts/Token.sol analyzed (10 contracts with 88 detectors), 39 result(s) found
```

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/0xd63ed2d56eddbd3cdbeb1b16fec5c54b9bd3afc65fc49c645587341896ad7b98

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

https://testnet.bscscan.com/tx/0xf1a6855130786b6ecba86590c51b6012715b46072c4d308681548a260fbdb751

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

https://testnet.bscscan.com/tx/0x84bbeb336adcde9188f0f70757ba3a50d37678f3b2214b82b22a36cc1124eb8a

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

https://testnet.bscscan.com/tx/0x1da20ca8ab43e12aed84fc536fe61411d27b9afeac98b0b3e957b144c9bab62d

5- Buying when not excluded from fees (tax 3%) (passed):

https://testnet.bscscan.com/tx/0x7c7c741f6db7dc77c646104ef98e4b618f576291766ed797cf193db14235e16a

6- Selling when not excluded from fees (tax 4%) (passed):

https://testnet.bscscan.com/tx/0x6083241742f0a25b75a720ec4e9e69ef23f55f8569e95339ddcb076ea491ca40

FUNCTIONAL TESTING

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

https://testnet.bscscan.com/tx/0x48ee0808c282c77ffc6310f169876f77313f55564a2b2b89969b1479bce48bc9

8- Internal swap (BNB set to Marketing wallet) (passed):

https://testnet.bscscan.com/tx/0xe32d72ae84f6d1c716dcdf13adb5905e168d247dc198a67405aa8037c076e47c

High Risk

Logical – Setting swap threshold to zero

Severity: High

function: setSwapTokensAtAmount

Status: Open

Overview:

Owner is able to set swapTokensAtAmount to zero. If swapTokensAtAmount is set to zero, contract will try to perform internal swap with 0 tokens which will revert the transaction

```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner {
    swapTokensAtAmount = newAmount;
}
```

Suggestion

Ensure that swapTokensAtAmount is always greater than zero

```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner {
    require(swapTokensAtAmount > 0);
    swapTokensAtAmount = newAmount;
}
```

Medium Risk

Centralization – EOA receiving LP tokens

Severity: Medium
function: addLiquidityETH
Status: Open
Overview:
owner is receiving LP tokens which are generated from auto-liquidity. This LP tokens can be used to remove a portion of liquidity pool

```
uniswapV2Router.addLiquidityETH{value: newBalance}(
    address(this),
    otherHalf,
    0, // slippage is unavoidable
    0, // slippage is unavoidable
    owner(),
    block.timestamp
);
```

Suggestion
Lock or burn new LP tokens

Informational Risk

Logical – Setting fee receiver to BNB rejector

Severity: Informational
function: sendBNB
Status: Open
Overview:
If marketingWallet or rewardPool are set to a contract that rejects receiving BNB internal swap will be failed (failing the whole transaction)

```
function sendBNB(address payable recipient, uint256 amount) internal {
    require(
        address(this).balance >= amount,
        "Address: insufficient balance"
    );
    (bool success, ) = recipient.call{value: amount}("");
}
```

BNB rejector example:

```
contract rejector {
        receive() external payable{
                revert();
        }
}
```

Suggestion
Ignore "status" variable, this means transaction will not be reverted even if calls to one of this addresses faild

```
function sendBNB(address payable recipient, uint256 amount) internal {
    require(
        address(this).balance >= amount,
        "Address: insufficient balance"
    );
    (bool success, ) = recipient.call{value: amount}("");
```

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