AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Shin Coin

DATED : 26 Sep 23'

High Risk

Centralization – Enabling Trades

Severity: High

function: EnableTrading

Status: Open

Overview:

The **enableTrading** function permits only the contract owner to activate trading capabilities. Until this function is executed, no investors can buy, sell, or transfer their tokens. This places a high degree of control and centralization in the hands of the contract owner.

```
function EnableTrading() external onlyOwner {
    require(!tradingEnabled, "Cannot re-enable trading");
    tradingEnabled = true;
    swapEnabled = true;
    genesis_block = block.number;
}
```

Suggestion

To reduce centralization and potential manipulation, consider one of the following approaches:

1. Automatically enable trading after a specified condition, such as the completion of a presale, is met.
2. If manual activation is still desired, consider transferring the ownership of the contract to a trustworthy, third-party entity like a certified "PinkSale Safu" developer. This can provide investors with more confidence in the eventual activation of trading capabilities, mitigating concerns of potential bad faith actions by the original owner

High Risk

Logical – Setting swap threshold to zero may revert swaps on internal swap

Severity: High
function: updateSwapTokensAtAmount
Status: Open
Overview:
Setting swapTokensAtAmount to zero may revert sell (SHIN => BNB) or wallet to wallet transfers. This is because contract will try to perform internal swap with 0 tokens as input.

```
function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {
    require(
        amount <= 42e14,
        "Cannot set swap threshold amount higher than 1% of tokens"
    );
    swapTokensAtAmount = amount * 10 ** _decimals;
}
```

Suggestion

Ensure that swapTokensAtAmount is always greater than zero

```
function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {
    require(
        amount <= 42e14,
        "Cannot set swap threshold amount higher than 1% of tokens"
    );
    require(amount >= totalSupply() / 100000 * 10 ** _decimals(), "new swap threshold must be higher than 0.00001% of total supply");
    swapTokensAtAmount = amount * 10 ** _decimals;
}
```

Alternatively, you can use a try-catch block when calling pancake swap router

```
try
    router.swapExactTokensForETHSupportingFeeOnTransferTokens(
        tokenAmount,
        0, // accept any amount of ETH
        path,
        address(this),
        block.timestamp
    )
{} catch {}
```

AUDIT SUMMARY

Project name – Shin Coin

Date: 26 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	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/0xe57B1bC5d1af4901A380873723d9104775DD38B6

Token Information

Token Name : Shin Coin

Token Symbol: SHIN

Decimals: 9

Token Supply: 420,000,000,000,000,000

Token Address:
0xaf3CECF00b5898c22444a0014884E7fEA9Fe884E

Checksum:
51599f7ac46156632bb42ac8e11439bbf74cc982

Owner:
0xFEE12658B27901532eA8eb7e0EfF6dA53FE81a64
(at time of writing the audit)

Deployer:
0xFEE12658B27901532eA8eb7e0EfF6dA53FE81a64

TOKEN OVERVIEW

Fees:

Buy Fees: 5%

Sell Fees: 5%

Transfer Fees: 5%

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

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

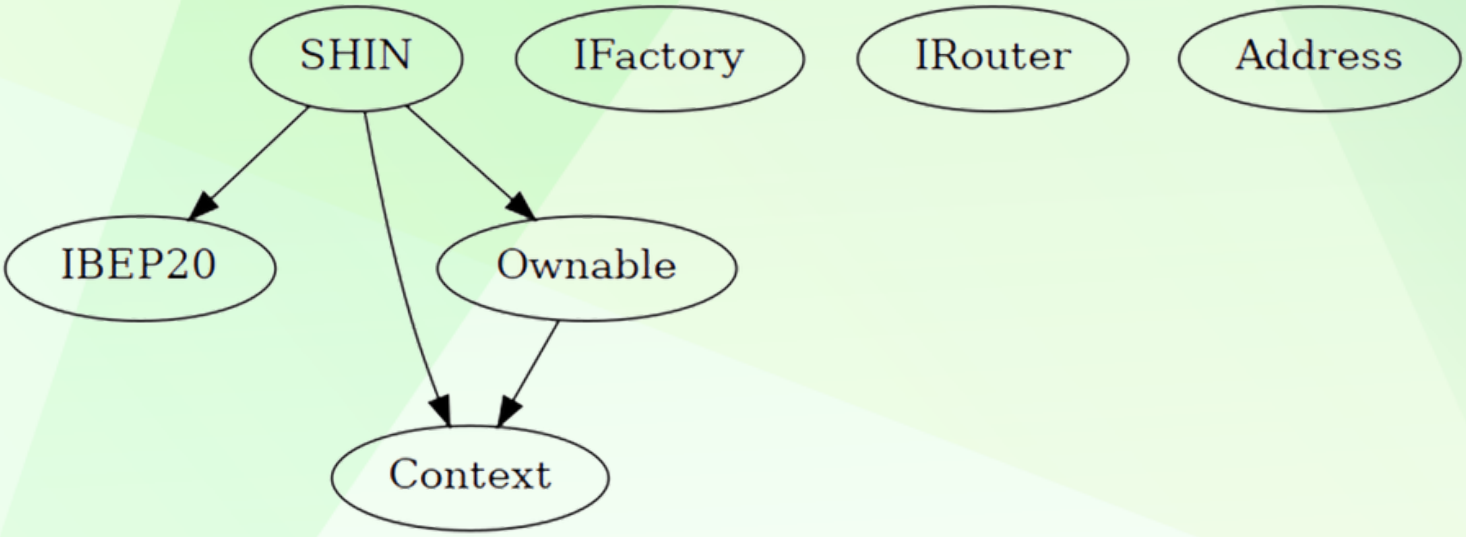
CLASSIFICATION OF RISK

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 not able to adjust buy/sell/transfer fees (5% each)**
- 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 must enable trades manually**

CONTRACT ASSESMENT

Contract	Type	Bases		
	└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
IBEP20	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 🔒		
Ownable	Implementation	Context		
	└	<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 ❗
	└	addLiquidityETH	External ❗	💵 |NO ❗
	└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑 |NO ❗
Address	Library			
	└	sendValue	Internal 🔒	🛑|
SHIN	Implementation	Context, IBEP20, Ownable		
	└	<Constructor>	Public ❗	🛑|NO ❗
	└	name	Public ❗		|NO ❗
	└	symbol	Public ❗		|NO ❗
	└	decimals	Public ❗		|NO ❗
	└	totalSupply	Public ❗		|NO ❗

CONTRACT ASSESMENT

└	balanceOf	Public ❗	|NO❗
└	allowance	Public ❗	|NO❗
└	approve	Public ❗	🔴 |NO❗
└	transferFrom	Public ❗	🔴|NO❗
└	increaseAllowance	Public ❗	🔴|NO❗
└	decreaseAllowance	Public ❗	🔴|NO❗
└	transfer	Public ❗	🔴 |NO❗
└	isExcludedFromReward	Public ❗	|NO❗
└	reflectionFromToken	Public ❗	|NO❗
└	EnableTrading	External ❗	🔴| onlyOwner
└	updatedeadline	External ❗	🔴	onlyOwner
└	tokenFromReflection	Public ❗	|NO❗
└	excludeFromReward	Public ❗	🔴| onlyOwner
└	includeInReward	External ❗	🔴| onlyOwner
└	excludeFromFee	Public ❗	🔴	onlyOwner
└	includeInFee	Public ❗	🔴	onlyOwner
└	isExcludedFromFee	Public ❗	|NO❗
└	_reflectRfi	Private 🔐	🔴|
└	_takeLiquidity	Private 🔐	🔴|
└	_takeMarketing	Private 🔐	🔴|
└	_takeOps	Private 🔐	🔴|
└	_takeDev	Private 🔐	🔴|
└	_getValues	Private 🔐		
└	_getTValues	Private 🔐		
└	_getRValues1	Private 🔐		
└	_getRValues2	Private 🔐		
└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	_approve	Private 🔐	🔴|
└	_transfer	Private 🔐	🔴|
└	_tokenTransfer	Private 🔐	🔴|
└	swapAndLiquify	Private 🔐	🔴| lockTheSwap
└	addLiquidity	Private 🔐	🔴|
└	swapTokensForBNB	Private 🔐	🔴	
└	bulkExcludeFee	External ❗	🔴| onlyOwner
└	updateMarketingWallet	External ❗	🔴| onlyOwner
└	updateDevWallet	External ❗	🔴| onlyOwner
└	updateOpsWallet	External ❗	🔴| onlyOwner
└	updateSwapTokensAtAmount	External ❗	🔴	onlyOwner
└	updateSwapEnabled	External ❗	🔴| onlyOwner
└	rescueBNB	External ❗	🔴	onlyOwner

CONTRACT ASSESMENT

| └ | rescueAnyBEP20Tokens | Public ❗ | 🛑| onlyOwner |
| └ | <Receive Ether> | External ❗ | 💵 |NO❗ |

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS

```
INFO:Detectors:
SHIN._rTotal (contracts/Token.sol#167) is set pre-construction with a non-constant function or state variable:
        - (MAX - (MAX % _tTotal))
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state
INFO:Detectors:
Pragma version^0.8.17 (contracts/Token.sol#9) 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 Address.sendValue(address,uint256) (contracts/Token.sol#128-139):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#134)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function IRouter.WETH() (contracts/Token.sol#104) is not in mixedCase
Struct SHIN.valuesFromGetValues (contracts/Token.sol#204-218) is not in CapWords
Function SHIN.EnableTrading() (contracts/Token.sol#372-377) is not in mixedCase
Parameter SHIN.updatedeadline(uint256)._deadline (contracts/Token.sol#379) is not in mixedCase
Parameter SHIN.updateSwapEnabled(bool)._enabled (contracts/Token.sol#795) is not in mixedCase
Parameter SHIN.rescueAnyBEP20Tokens(address,address,uint256)._tokenAddr (contracts/Token.sol#807) is not in mixedCase
Parameter SHIN.rescueAnyBEP20Tokens(address,address,uint256)._to (contracts/Token.sol#808) is not in mixedCase
Parameter SHIN.rescueAnyBEP20Tokens(address,address,uint256)._amount (contracts/Token.sol#809) is not in mixedCase
Constant SHIN._decimals (contracts/Token.sol#163) is not in UPPER_CASE_WITH_UNDERSCORES
Variable SHIN.genesis_block (contracts/Token.sol#171) is not in mixedCase
Constant SHIN._name (contracts/Token.sol#179) is not in UPPER_CASE_WITH_UNDERSCORES
Constant SHIN._symbol (contracts/Token.sol#180) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Redundant expression "this (contracts/Token.sol#49)" inContext (contracts/Token.sol#43-52)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
INFO:Detectors:
SHIN.slitherConstructorVariables() (contracts/Token.sol#142-819) uses literals with too many digits:
        - _tTotal = 420000000000000000 * 10 ** _decimals (contracts/Token.sol#166)
SHIN.slitherConstructorVariables() (contracts/Token.sol#142-819) uses literals with too many digits:
        - swapTokensAtAmount = 210000000000000 * 10 ** 9 (contracts/Token.sol#169)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
SHIN._lastSell (contracts/Token.sol#158) is never used in SHIN (contracts/Token.sol#142-819)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable
INFO:Detectors:
Loop condition `i < _excluded.length` (contracts/Token.sol#584) should use cached array length instead of referencing `length` member of the s
torage array.
 Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#cache-array-length
INFO:Detectors:
SHIN._tTotal (contracts/Token.sol#166) should be constant
SHIN.deadWallet (contracts/Token.sol#174) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
SHIN.pair (contracts/Token.sol#161) should be immutable
SHIN.router (contracts/Token.sol#160) 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/0xa7679cf4dc221269b4dc4144b54b288a9f8575bb1776031c647c1d3e7de4fc8a

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

https://testnet.bscscan.com/tx/0xd25dc9550abe8fa8a4cc67de49e5d2db31ddf8d9cb4264bdd545f3cfb34b3013

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

https://testnet.bscscan.com/tx/0x0f6343857ce25ac610d830a3746bab6d7a57846261d1aaf53b091159bd973edf

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

https://testnet.bscscan.com/tx/0x3ded8e3990ac48aa95215fa795e120c2aa16fada2bbd41a1e01e1a4e7979d8df

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

https://testnet.bscscan.com/tx/0xc79da70aadaa5edbf8ec13ae7be263dd652a856f9354fac875253728f5dc1e99

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

https://testnet.bscscan.com/tx/0x786f5dbd0d6b6cb3f99eb0a3aef3c2cc8d2e11ce6b5c07b54f386444b34ec76d

FUNCTIONAL TESTING

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

https://testnet.bscscan.com/tx/0x9950b42d6c85eb280a132b65117f52ad7ac257e234dffab9d7b588b660ac08d2

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

https://testnet.bscscan.com/tx/0x786f5dbd0d6b6cb3f99eb0a3aef3c2cc8d2e11ce6b5c07b54f386444b34ec76d

High Risk

Centralization – Enabling Trades

Severity: High

function: **EnableTrading**

Status: Open

Overview:

The **enableTrading** function permits only the contract owner to activate trading capabilities. Until this function is executed, no investors can buy, sell, or transfer their tokens. This places a high degree of control and centralization in the hands of the contract owner.

```
function EnableTrading() external onlyOwner {
    require(!tradingEnabled, "Cannot re-enable trading");
    tradingEnabled = true;
    swapEnabled = true;
    genesis_block = block.number;
}
```

Suggestion

To reduce centralization and potential manipulation, consider one of the following approaches:

1. Automatically enable trading after a specified condition, such as the completion of a presale, is met.
2. If manual activation is still desired, consider transferring the ownership of the contract to a trustworthy, third-party entity like a certified "PinkSale Safu" developer. This can provide investors with more confidence in the eventual activation of trading capabilities, mitigating concerns of potential bad faith actions by the original owner

High Risk

Logical – Setting swap threshold to zero may revert swaps on internal swap

Severity: High
function: updateSwapTokensAtAmount
Status: Open
Overview:
Setting swapTokensAtAmount to zero may revert sell (SHIN => BNB) or wallet to wallet transfers. This is because contract will try to perform internal swap with 0 tokens as input.

```
function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {
    require(
        amount <= 42e14,
        "Cannot set swap threshold amount higher than 1% of tokens"
    );
    swapTokensAtAmount = amount * 10 ** _decimals;
}
```

Suggestion

Ensure that swapTokensAtAmount is always greater than zero

```
function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {
    require(
        amount <= 42e14,
        "Cannot set swap threshold amount higher than 1% of tokens"
    );
    require(amount >= totalSupply() / 100000 * 10 ** _decimals(), "new swap threshold must be higher than 0.00001% of total supply");
    swapTokensAtAmount = amount * 10 ** _decimals;
}
```

Alternatively, you can use a try-catch block when calling pancake swap router

```
try
    router.swapExactTokensForETHSupportingFeeOnTransferTokens(
        tokenAmount,
        0, // accept any amount of ETH
        path,
        address(this),
        block.timestamp
    )
{} catch {}
```

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