AuditAce

FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

SPE

DATED : 28 MAY 23'

AUDIT SUMMARY

Project name – SPE

Date: 28 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: The code has undergone a line-by-line review by the **Ace** team.

2.ETH Test Network: All tests were conducted on the ETH 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/address/0xf640377DFC 4EDab07867D6Bc8a86077BBD09a0D3

Token Information

Name : SpacePepeExchange

Symbol : SPE

Decimals: 18

Network: BSC

Token Type:BEP20

Token Address:
0x5d32fd09f13a1D30D1CD0c560b610bB6dDdb9E6d

Owner:
0xBfF1A8D9eA7D9dc72285716d97043F6fC951e969
(at time of writing the audit)

Deployer:0xBfF1A8D9eA7D9dc72285716d97043F6fC951e969

Token Information

Fees:

Buy Fees: 2%

Sell Fees: 2%

Transfer Fees: 2%

Fees Privilige: immutable fees

Ownership :

0xBfF1A8D9eA7D9dc72285716d97043F6fC951e969

Minting: None

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: - including in fees

- excluding from fees

- 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
- ✓ **Gasless Send**
- ✓ Private modifier
- ✓ Using block.timestamp
- ✓ Multiple Sends
- ✓ Re-entrancy
- ✓ Using Suicide
- ✓ Tautology or contradiction
- ✓ Gas Limitand Loops
- ✓ Timestamp Dependence
- ✓ Address hardcoded
- ✓ Revert/require functions
- ✓ Exception Disorder
- ✓ Use of tx.origin
- ✓ Using inline assembly
- ✓ Integer overflow/underflow
- ✓ Divide before multiply
- ✓ Dangerous strict equalities
- ✓ Missing Zero Address Validation
- ✓ Using SHA3
- ✓ Compiler version not fixed
- ✓ 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

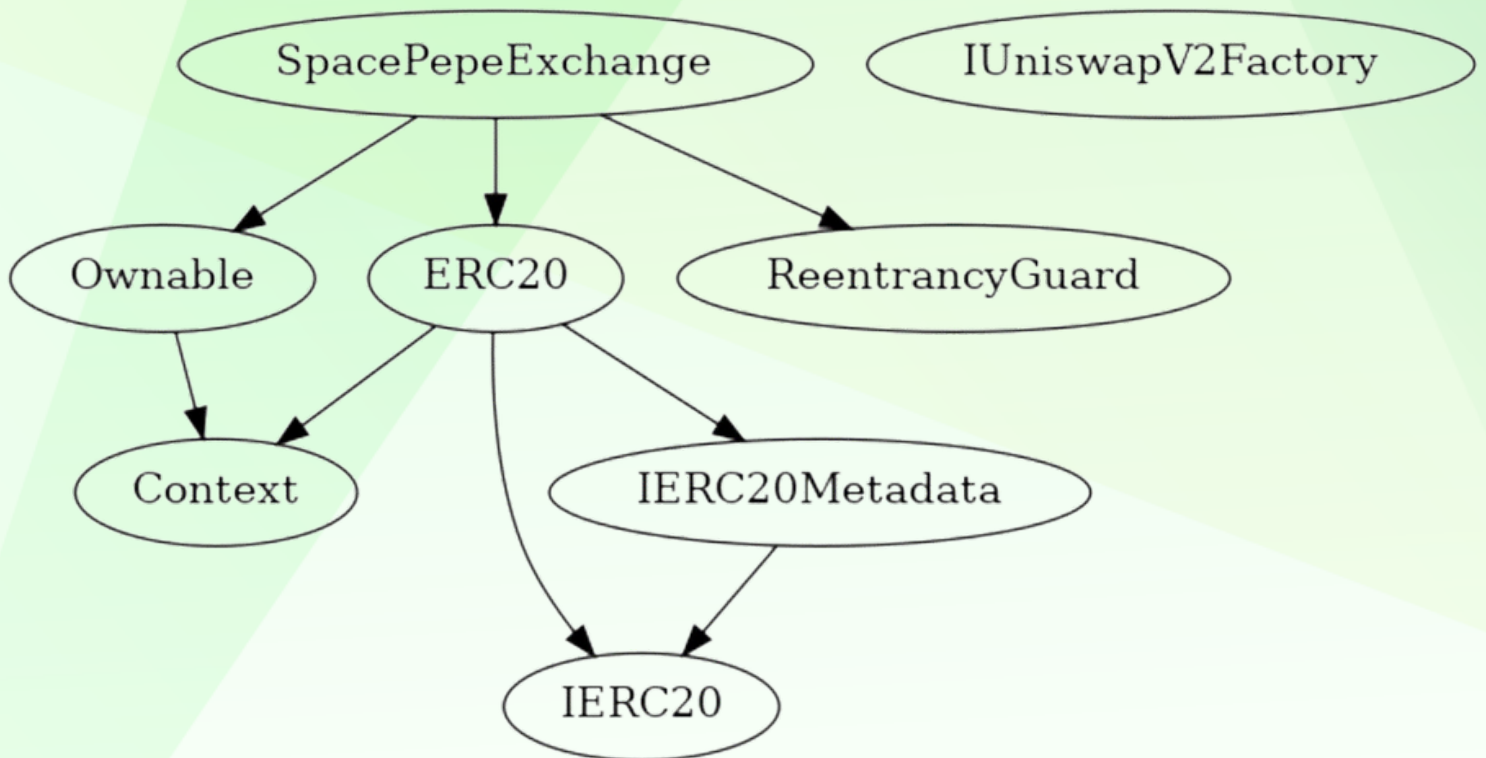| Severity	Found
◆ Critical	0
◆ High-Risk	0
◆ Medium-Risk	0
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	0

INHERITANCE TREE

POINTS TO NOTE

- Owner is not able to change buy/sell/transfer fees (2% all fees)
- Owner is not able to blacklist an arbitrary address.
- Owner is not able to disable trades
- Owner is not able to set max buy/sell/transfer/hold amount to 0
- Owner is not able to mint new tokens

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	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 ❗

CONTRACT ASSESMENT

└	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 ❗
IERC20Permit	Interface			
└	permit	External ❗	🔴	NO ❗
└	nonces	External ❗		NO ❗
└	DOMAIN_SEPARATOR	External ❗		NO ❗
Address	Library			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🔴	
└	functionCall	Internal 🔒	🔴	
└	functionCall	Internal 🔒	🔴	

CONTRACT ASSESMENT

└	functionCallWithValue	Internal 🔐	🔴	
└	functionCallWithValue	Internal 🔐	🔴	
└	functionStaticCall	Internal 🔒		
└	functionStaticCall	Internal 🔒		
└	functionDelegateCall	Internal 🔐	🔴	
└	functionDelegateCall	Internal 🔐	🔴	
└	verifyCallResultFromTarget	Internal 🔒		
└	verifyCallResult	Internal 🔒		
└	_revert	Private 🔐		
IERC20	Interface			
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	transfer	External ❗	🔴	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🔴	NO❗
└	transferFrom	External ❗	🔴	NO❗
SafeERC20	Library			
└	safeTransfer	Internal 🔐	🔴	
└	safeTransferFrom	Internal 🔐	🔴	
└	safeApprove	Internal 🔐	🔴	
└	safeIncreaseAllowance	Internal 🔐	🔴	
└	safeDecreaseAllowance	Internal 🔐	🔴	
└	safePermit	Internal 🔐	🔴	
└	_callOptionalReturn	Private 🔐	🔴	
IERC20Metadata	Interface	IERC20		
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
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❗

CONTRACT ASSESMENT

└	transferFrom	Public ❗	🔴	NO❗
└	increaseAllowance	Public ❗	🔴	NO❗
└	decreaseAllowance	Public ❗	🔴	NO❗
└	_transfer	Internal 🔒	🔴	
└	_mint	Internal 🔒	🔴	
└	_burn	Internal 🔒	🔴	
└	_approve	Internal 🔒	🔴	
└	_spendAllowance	Internal 🔒	🔴	
└	_beforeTokenTransfer	Internal 🔒	🔴	
└	_afterTokenTransfer	Internal 🔒	🔴	
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🔴	NO❗
└	owner	Public ❗		NO❗
└	_checkOwner	Internal 🔒		
└	renounceOwnership	Public ❗	🔴	onlyOwner
└	transferOwnership	Public ❗	🔴	onlyOwner
└	_transferOwnership	Internal 🔒	🔴	
ReentrancyGuard	Implementation			
└	<Constructor>	Public ❗	🔴	NO❗
└	_nonReentrantBefore	Private 🔒	🔴	
└	_nonReentrantAfter	Private 🔒	🔴	
└	_reentrancyGuardEntered	Internal 🔒		
SpacePepeExchange	Implementation	ERC20, Ownable, ReentrancyGuard		
└	<Constructor>	Public ❗	🔴	ERC20
└	<Receive Ether>	External ❗	💵	NO❗
└	<Fallback>	External ❗	💵	NO❗
└	getRouterAddress	Public ❗		NO❗
└	claimStuckTokens	External ❗	🔴	onlyOwner
└	excludeFromFees	External ❗	🔴	onlyOwner
└	isExcludedFromFees	Public ❗		NO❗
└	setAutomatedMarketMakerPair	Public ❗	🔴	onlyOwner
└	isAutomatedMarketMakerPair	Public ❗		NO❗
└	toggleSwapBack	External ❗	🔴	onlyOwner
└	setSwapTokensAtAmount	External ❗	🔴	onlyOwner
└	_transfer	Internal 🔒	🔴	
└	_swapBack	Internal 🔒	🔴	
└	sendBNB	Internal 🔒	🔴	nonReentrant
└	manualSwapBack	External ❗	🔴	NO❗

CONTRACT ASSESMENT

Legend

Symbol	Meaning
🔴	Function can modify state
💵	Function is payable

STATIC ANALYSIS

```
Pragma version^0.8.17 (contracts/Token.sol#7) 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 Address.sendValue(address,uint256) (contracts/Token.sol#372-383):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#378)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (contracts/Token.sol#420-440):
        - (success,returndata) = target.call{value: value}(data) (contracts/Token.sol#430-432)
Low level call in Address.functionStaticCall(address,bytes,string) (contracts/Token.sol#454-467):
        - (success,returndata) = target.staticcall(data) (contracts/Token.sol#459)
Low level call in Address.functionDelegateCall(address,bytes,string) (contracts/Token.sol#481-494):
        - (success,returndata) = target.delegatecall(data) (contracts/Token.sol#486)
Low level call in SpacePepeExchange.sendBNB(address,uint256) (contracts/Token.sol#1177-1186):
        - (success) = address(_to).call{value: amount}() (contracts/Token.sol#1183)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IUniswapV2Pair.DOMAIN_SEPARATOR() (contracts/Token.sol#83) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (contracts/Token.sol#85) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (contracts/Token.sol#116) is not in mixedCase
Function IUniswapV2Router01.WETH() (contracts/Token.sol#158) is not in mixedCase
Function IERC20Permit.DOMAIN_SEPARATOR() (contracts/Token.sol#364) is not in mixedCase
Parameter SpacePepeExchange.sendBNB(address,uint256)._to (contracts/Token.sol#1177) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#163) is too similar to IUniswapV2Router01.
addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

SpacePepeExchange.constructor() (contracts/Token.sol#986-1016) uses literals with too many digits:
        - _mint(owner(),100_000_00000 * (10 ** 18)) (contracts/Token.sol#987)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

SpacePepeExchange.marketingWallet (contracts/Token.sol#965) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

SpacePepeExchange.buyTax (contracts/Token.sol#961) should be immutable
SpacePepeExchange.marketingWalletShares (contracts/Token.sol#967) should be immutable
SpacePepeExchange.sellTax (contracts/Token.sol#962) should be immutable
SpacePepeExchange.setSwapTokensLimit (contracts/Token.sol#970) should be immutable
SpacePepeExchange.taxDenominator (contracts/Token.sol#960) should be immutable
SpacePepeExchange.totalTax (contracts/Token.sol#963) should be immutable
SpacePepeExchange.uniswapV2Pair (contracts/Token.sol#975) should be immutable
SpacePepeExchange.uniswapV2Router (contracts/Token.sol#974) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

Static Analysis

an static analysis of the code were performed using slither. No issues were found

FUNCTIONAL TESTING

Router (PCS V2):
0xD99D1c33F9fC3444f8101754aBC46c52416550D1

1- Adding liquidity by owner(passed):
https://testnet.bscscan.com/tx/0x74ad4798bd759af40a573fca2e
2b49a7490c22a9021087d656bf8f22e3d9337e

2- Buying when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x79b93df4dc94bc938024a118d9
e9853d8a2c79141e0da1d89b9c7b12d85ad376

3- Selling when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x01727cccdae8f68fe0254ee8485
f5e8b4420ea90c9cc29d5031954d71c17ef5b

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0xb04d6fea6190e5651a2825c4816
69a4f78cd47c376401306132047c19a6f9cf5

5- Buying when not excluded from fees (2% tax) (passed):
https://testnet.bscscan.com/tx/0x4d6306b8b728624e6a79b5477
e12eaae6f79cba4947fb366f189d78fe26752b6

6- Selling when not excluded from fees (2% tax) (passed):
https://testnet.bscscan.com/tx/0xe8803fe945695aced4c961fbd0
584b2c046f7388b780aca2419d8372c0387889

7- Transferring when not excluded from fees (2% tax) (passed):
https://testnet.bscscan.com/tx/0x534247331a9d81f98e84b0a8f4
bd8c9cf8313e235812e63aa3305466b6abf78b

FUNCTIONAL TESTING

8- Internal swap(passed):
- Accumulated fees converted to BNB and sent to marketing wallet
https://testnet.bscscan.com/address/0x1d7b4000eb9723ec5c626
517fb537496347e2488#internaltx

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