



Smart Contract Audit

FOR

Price Tools

DATED : 25 FEB 23'



AUDIT SUMMARY

Project name – Price Tools

Date: 25 February, 2023

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

Audit Status: Passed (Contract is developed by pinksale's Safu Dev)

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	0	0	0	2
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 done on BSC Test network, each test has its transaction has attached to it.

3- Slither : Static Analysis

Testnet Link: all tests were done using this contract, tests are done on BSC Testnet

<https://testnet.bscscan.com/token/0xAc23921dF369474B409e0f5DAB39b71cbEc2A267>



Token Information

Token Name : Price Tools

Token Symbol: PTOOLS

Decimals: 18

Token Supply: 1,000,000,000

Token Address:

0x34609CC0bE4f69982f5781a893F3FCc44f414Ced

Checksum:

adeaf35f858118ef35b1dc566ee56096f8c938d8

Network : ARBISCAN Network



TOKEN OVERVIEW

Fees:

Buy Fees: 0%

Sell Fees: 0%

Transfer Fees: 0%

Fees Privilige: None

Ownership : Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Privileges: None



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 is exercised 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

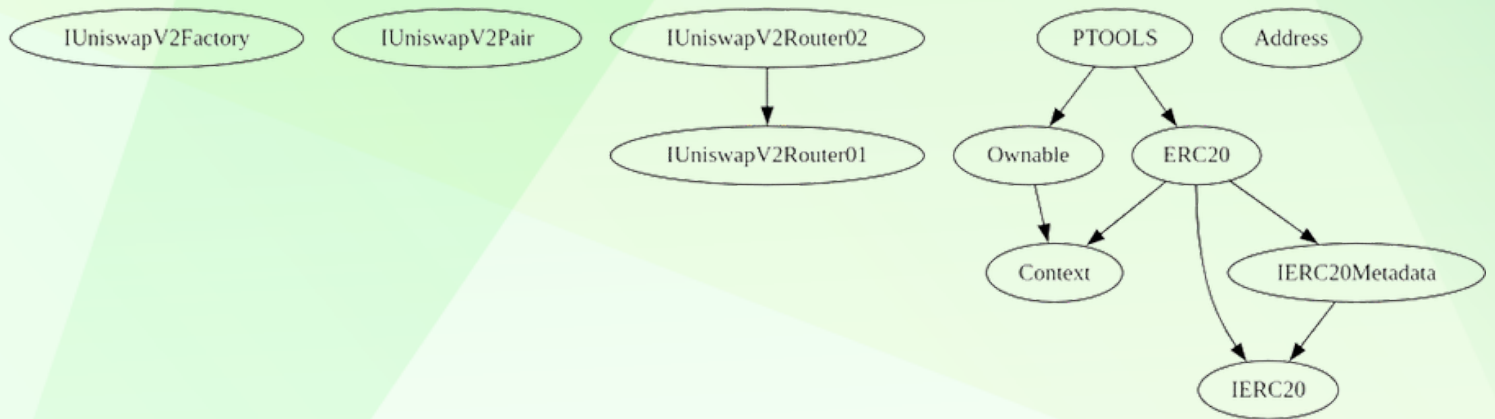
◆ Low-Risk

0

◆ Gas Optimization / Suggestions

2

INHERITANCE TREE



POINTS TO NOTE

- Owner is not able to set buy/sell/transfer taxes (0% static)
 - Owner is not able to blacklist an arbitrary wallet
 - Owner is not able to set max buy/sell/transfer amounts
 - Owner is not able to disable trades
 - Owner is not able to mint new tokens
 - Owner must enable trading in order for everyone to be able to trade (otherwise no one would be able to buy/sell/transfer)
-



CONTRACT ASSESMENT

Contract	Type	Bases			
└──┐	**Function Name**	**Visibility**	**Mutability**	**Modifiers**	
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	!
└ sync	External	!	⊗	NO	!
└ initialize	External	!	⊗	NO	!






































































CONTRACT ASSESMENT

```
||||| |
| **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! |
|||||
| **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! |
|||||
```

CONTRACT ASSESMENT

```

| **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  | | |
| | | |
| **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   | | |
|  | _mint | Internal   | | |
|  | _burn | Internal   | | |

```

CONTRACT ASSESMENT

```

| ^ | _approve | Internal | 🔒 | 🛑 | |
| ^ | _beforeTokenTransfer | Internal | 🔒 | 🛑 | |
| ^ | _afterTokenTransfer | Internal | 🔒 | 🛑 | |
| | | |
| **PTOOLS** | Implementation | ERC20, Ownable | | |
| ^ | <Constructor> | Public | ! | 🛑 | ERC20 |
| ^ | <Receive Ether> | External | ! | 💰 | NO! |
| ^ | claimStuckTokens | External | ! | 🛑 | onlyOwner |
| ^ | excludeFromFees | External | ! | 🛑 | onlyOwner |
| ^ | isExcludedFromFees | Public | ! | | NO! |
| ^ | enableTrading | External | ! | 🛑 | onlyOwner |
| ^ | _transfer | Internal | 🔒 | 🛑 | |

```

| Symbol | Meaning |

| :-----: | ----- |

| 🛑 | Function can modify state |

| 💰 | Function is payable |



STATIC ANALYSIS

```
Address._revert(bytes,string) (contracts/Token.sol#549-564) uses assembly
- INLINE ASM (contracts/Token.sol#557-560)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Address._revert(bytes,string) (contracts/Token.sol#549-564) is never used and should be removed
Address.functionCall(address,bytes) (contracts/Token.sol#408-419) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/Token.sol#421-427) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/Token.sol#429-441) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/Token.sol#443-463) is never used and should be removed
Address.functionDelegateCall(address,bytes) (contracts/Token.sol#492-502) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (contracts/Token.sol#504-517) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/Token.sol#465-475) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/Token.sol#477-490) is never used and should be removed
Address.isContract(address) (contracts/Token.sol#391-393) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (contracts/Token.sol#537-547) is never used and should be removed
Address.verifyCallResultFromTarget(address,bool,bytes,string) (contracts/Token.sol#519-535) is never used and should be removed
Context.msgData() (contracts/Token.sol#572-575) is never used and should be removed
ERC20._burn(address,uint256) (contracts/Token.sol#762-777) 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#23) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.18 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#395-406):
- (success) = recipient.call{value: amount}() (contracts/Token.sol#401)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (contracts/Token.sol#443-463):
- (success, returndata) = target.call{value: value}(data) (contracts/Token.sol#453-455)
Low level call in Address.functionStaticCall(address,bytes,string) (contracts/Token.sol#477-490):
- (success, returndata) = target.staticcall(data) (contracts/Token.sol#482)
Low level call in Address.functionDelegateCall(address,bytes,string) (contracts/Token.sol#504-517):
- (success, returndata) = target.delegatecall(data) (contracts/Token.sol#509)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IUniswapV2Pair.DOMAIN_SEPARATOR() (contracts/Token.sol#85) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (contracts/Token.sol#87) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (contracts/Token.sol#118) is not in mixedCase
Function IUniswapV2Router01.WETH() (contracts/Token.sol#158) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#573)" inContext (contracts/Token.sol#567-576)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

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

PTTOOLS.uniswapV2Pair (contracts/Token.sol#808) should be immutable
PTTOOLS.uniswapV2Router (contracts/Token.sol#807) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
(contracts/Token.sol#808) analyzed (31 contexts with 04 detectors) - 31 results 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

Router (PCS V2):

0xD99D1c33F9fC3444f8101754aBC46c52416550D1

1- Adding Liquidity (**Passed**):

liquidity added on Pancakeswap V2:

<https://testnet.bscscan.com/tx/0x39629581bec94c57e89bd1e1bbf718d95eedefdae30d62a0da506bca52453fa3>

no issue were found on adding liquidity.

2- Buying (0% Tax) (**Passed**):

<https://testnet.bscscan.com/tx/0xa3ff4ddc9dd2a0d3161f13dfe107a077ac23e4b1aae207b91265d0338731ab88>

3- Selling (0% Tax) (**Passed**):

<https://testnet.bscscan.com/tx/0x23eeac4c9d3b6f224e91a8fa78d81340cfcd1e8dc612d091b6badb68538006764>

4-Transferring (0% tax)(**Passed**):


<https://testnet.bscscan.com/tx/0xbeca595119bf35e1e6d97be448da82208d10dd1462ba0bfc80eff9b41c9766f285>

MANUAL TESTING

SUGGESTIONS AND OPTIMIZATIONS

1- The contract contains a function named **"excludeFromFees,"** despite the fact that there are no fees associated with the contract. This may cause confusion for investors of the contract. Same thing applies for **"isExcludedFromFees"** function.

Suggestion: One possible solution to address the confusion caused by the **"excludeFromFees"** function is to change its name to **"setWhitelistedWallet"** and to add a new function called **"isWhitelisted."** This would make the purpose of the functions clearer and more intuitive for users, as it would clearly indicate that these functions are related to **whitelisting** wallets rather than **fees**.



MANUAL TESTING

SUGGESTIONS AND OPTIMIZATIONS

2) The '**claimStuckTokens**' function in the contract prevents withdrawal of native tokens, even though there are no fees associated with the contract. Since withdrawing these tokens would not harm investors, it is recommended that the function be updated to allow for withdrawal of native tokens from the contract.

```
function claimStuckTokens(address token) external onlyOwner {  
    require(  
        token != address(this),  
        "Owner cannot claim contract's balance of its own tokens"  
    );  
    if (token == address(0x0)) {  
        payable(msg.sender).sendValue(address(this).balance);  
        return;  
    }  
    IERC20 ERC20token = IERC20(token);  
    uint256 balance = ERC20token.balanceOf(address(this));  
    ERC20token.transfer(msg.sender, balance);  
}
```

Social Media Overview

**Here are the Social Media Accounts of
Price Tools**



<https://t.me/pricetools>



<https://twitter.com/pricetools>



<https://pricetools.io/>



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 specialize 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>
