

# KISHIELD

Security Audit

**Pounder Protocol Token**

April 27, 2022



# Table of Contents



- 1 Audit Summary**
- 2 Project Overview**
  - 2.1 Token Summary
  - 2.2 Main Contract Assessed
- 3 Smart Contract Vulnerability Checks**
- 4 Contract Ownership**
  - 4.1 Privileged Functions
- 5 Important Notes To The Users**
- 6 Findings Summary**
  - 6.1 Classification of Issues
  - 6.1 Findings Table
    - 01 Division before Multiplication
    - 02 Assignment with no effects
    - 03 Boolean equality
    - 04 Too many digits
- 7 Statistics**
  - 7.1 Liquidity
  - 7.2 Token Holders
  - 7.3 Liquidity Holders
- 8 Liquidity Ownership**
- 9 Disclaimer**



# Audit Summary

This report has been prepared for PounderProtocol Token on the Binance Chain network. KISHIELD provides both client-centered and user-centered examination of the smart contracts and their current status when applicable. This report represents the security assessment made to find issues and vulnerabilities on the source code along with the current liquidity and token holder statistics of the protocol.

A comprehensive examination has been performed, utilizing Cross Referencing, Static Analysis, In-House Security Tools, and line-by-line Manual Review.

The auditing process pays special attention to the following considerations:

- Ensuring contract logic meets the specifications and intentions of the client without exposing the user's funds to risk.
- Testing the smart contracts against both common and uncommon attack vectors.
- Inspecting liquidity and holders statistics to inform the current status to both users and client when applicable.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Verifying contract functions that allow trusted and/or untrusted actors to mint, lock, pause, and transfer assets.
- Thorough line-by-line manual review of the entire codebase by industry experts.

# Project Overview

## Token Summary

Parameter	Result
Address	0xbC6246f22f5D6A883E5acCB69016655e1744393C
Name	PounderProtocol
Token Tracker	PounderProtocol (POUND)
Decimals	18
Supply	5,000,000
Platform	Binance Chain
compiler	v0.8.7+commit.e28d00a7
Optimization	Yes with 1 runs
LicenseType	None
Language	Solidity
Codebase	<a href="https://bscscan.com/address/0xbC6246f22f5D6A883E5acCB69016655e1744393C">https://bscscan.com/ address/0xbC6246f22f5D6A883E5acCB69016655e1744393C</a>
Url	<a href="https://www.pounderprotocol.com/">https://www.pounderprotocol.com/</a>

## Main Contract Assessed

Name	Contract	Live
PounderProtocol	0xbC6246f22f5D6A883E5acCB69016655e1744393C	Yes



# Smart Contract Vulnerability Checks

Vulnerability	Automatic Scan	Manual Scan	Result
Unencrypted Private Data On-Chain	Complete	Complete	✔ Low / No Risk
Code With No Effects	Complete	Complete	✔ Low / No Risk
Message call with hardcoded gas amount	Complete	Complete	✔ Low / No Risk
Hash Collisions With Multiple Variable Length Arguments	Complete	Complete	✔ Low / No Risk
Unexpected Ether balance	Complete	Complete	✔ Low / No Risk
Presence of unused variables	Complete	Complete	✔ Low / No Risk
Right-To-Left-Override control character (U+202E)	Complete	Complete	✔ Low / No Risk
Typographical Error	Complete	Complete	✔ Low / No Risk
DoS With Block Gas Limit	Complete	Complete	✔ Low / No Risk
Arbitrary Jump with Function Type Variable	Complete	Complete	✔ Low / No Risk
Insufficient Gas Griefing	Complete	Complete	✔ Low / No Risk
Incorrect Inheritance Order	Complete	Complete	✔ Low / No Risk
Write to Arbitrary Storage Location	Complete	Complete	✔ Low / No Risk
Requirement Violation	Complete	Complete	✔ Low / No Risk
Missing Protection against Signature Replay Attacks	Complete	Complete	✔ Low / No Risk
Weak Sources of Randomness from Chain Attributes	Complete	Complete	✔ Low / No Risk



Vulnerability	Automatic Scan	Manual Scan	Result
Authorization through tx.origin	Complete	Complete	✔ Low / No Risk
Delegatecall to Untrusted Callee	Complete	Complete	✔ Low / No Risk
Use of Deprecated Solidity Functions	Complete	Complete	✔ Low / No Risk
Assert Violation	Complete	Complete	✔ Low / No Risk
Reentrancy	Complete	Complete	✔ Low / No Risk
Unprotected SELFDESTRUCT Instruction	Complete	Complete	✔ Low / No Risk
Unprotected Ether Withdrawal	Complete	Complete	✔ Low / No Risk
Unchecked Call Return Value	Complete	Complete	✔ Low / No Risk
Outdated Compiler Version	Complete	Complete	✔ Low / No Risk
Integer Overflow and Underflow	Complete	Complete	✔ Low / No Risk
Function Default Visibility	Complete	Complete	✔ Low / No Risk

## Contract Ownership

The contract ownership of PounderProtocol is not currently renounced. The ownership of the contract grants special powers to the protocol creators, making them the sole addresses that can call sensible ownable functions that may alter the state of the protocol.

The current owner is the address 0x1ccCE182F4B5d1e57FF09eFEC2cEcb3f84a81E50 which can be viewed from:

[HERE](#)

The owner wallet has the power to call the functions displayed on the privileged functions chart below, if the owner wallet is compromised this privileges could be exploited.

We recommend the team to renounce ownership at the right timing if possible, or gradually migrate to a timelock with governing functionalities in respect of transparency and safety considerations.



## Important Notes To The Users:

- The owner cannot change the max tx amount.
- There is no blacklist for addresses nor contracts.
- Only FeeExempt addresses can transfer until the owner calls `setInitialDistributionFinished(true)`
- Once the owner renounces ownership of the contract, none of the following are applicable.
- The owner can pause the transfers for not excluded addresses by calling `setInitialDistributionFinished(false)`
- The owner can set the buy fees to a max of 13% and the sell fees to 18%.
- The owner can withdraw stuck BNB in the contract
- The owner can enable/disable autorebase, change the rebase frequency, change the reward yield, set the time for the next rebase manually.
- The owner can add/remove wallets for fee exempt.
- The owner can change the TargetLiquidity settings and the SwapBackSettings.
- No high-risk Exploits/Vulnerabilities Were Found in token Source Code.

## Audit Passed



# Findings Summary

## Classification of Issues

Severity	Description
● High	Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, or impairment of the contract and its functions. Issues under this classification are recommended to be fixed with utmost urgency
● Medium	Bugs or issues with that may be subject to exploit, though their impact is somewhat limited. Issues under this classification are recommended to be fixed as soon as possible.
● Low	Effects are minimal in isolation and do not pose a significant danger to the project or its users. Issues under this classification are recommended to be fixed nonetheless.
● Info	Consistency, syntax or style best practices. Generally pose a negligible level of risk, if any.

## Findings

Severity	Found
● High	0
● Medium	0
● Low	1
● Info	3
Total	4



# Findings

## Division before Multiplication

ID	Severity	Contract	Function
01	● Low	PounderProtocol	function swapBack()

### Description

Precision Loss. 'contractTokenBalance = \_gonBalances[address(this)].div(\_gonsPerFragment) => amountToLiquify = contractTokenBalance.mul(dynamicLiquidityFee.mul(2)).div(realTotalFee)'. Division before multiplication can result in truncation and less accurate results

### Recommendation

Multiplication should be performed before division to not lose precision.

## Assignment with no effects

ID	Severity	Contract	Function
02	● Informational	PounderProtocol	Variables initialDistributionFinished and autoRebase = false

### Description

Bools variables in solidity are set to false by default.

### Recommendation

We recommend deleting the initialization of the boolean variable to false

## Boolean equality

ID	Severity	Contract	Function
03	● Informational	PounderProtocol	Function swapping()

### Description

Statement 'require(bool,string)(inSwap == false,ReentrancyGuard: reentrant call)'  
Boolean constants can be used directly and do not need to be compare to true or false.

### Recommendation

Remove the equality to the boolean constant. In case of comparing to false make use of '! bool'

## Too many digits

ID	Severity	Contract	Function
04	● Informational	PounderProtocol	Variable rewardYieldDenominator

### Description

Literals with many digits are difficult to read and review.

### Recommendation

Make use of scientific notation, use underscores, and/or use ether suffix.

## Privileged Functions (onlyOwner)

Function Name	Parameters	Visibility
renounceOwnership	none	external
transferOwnership	address newOwner	external
setInitialDistributionFinished	bool _value	external
setFeeExempt	address _addr, bool _value	external
setTargetLiquidity	uint256 target, uint256 accuracy	external
setSwapBackSettings	bool _enabled, uint256 _num, uint256 _denom	external
setFeeReceivers	address _liquidityReceiver, address _treasuryReceiver, address _insuranceFundsReceiver	external
setFees	uint256 _liquidityFee, uint256 _insuranceFunds, uint256 _treasuryFee, uint256 _sellFeeTreasuryAdded, uint256 _feeDenominator	external
clearStuckBalance	address _receiver	external
setAutoRebase	bool _autoRebase	external
setRebaseFrequency	uint256 _rebaseFrequency	external
setRewardYield	uint256 _rewardYield, uint256 _rewardYieldDenominator	external
setIsLiquidityInBnb	bool _value	external
setNextRebase	uint256 _nextRebase	external



# Statistics

## Liquidity Info

Parameter	Result
Pair Address	0x2b3a2E45672851F79A86ac8c4211f709E13eAfe4
POUND Reserves	0.00 POUND
BNB Reserves	0.00 BNB
Liquidity Value	\$0 USD

## Token (POUND) Holders Info

Parameter	Result
POUND Percentage Burnt	0.00%
POUND Amount Burnt	0 POUND
Top 10 Percentage Own	100.00%
Top 10 Amount Owned	5,000,000 POUND
Top 10 Aprox Value	\$NaN USD



## LP (POUND/BNB) Holders Info

Parameter	Result
POUND/BNB % Burnt	0.00%
POUND/BNB Amount Burnt	0 POUND
Top 10 Percentage Owned	0.00%
Top 10 Amount Owned	0 POUND
Locked Tokens Percentage	0.00%
Locked Tokens Amount	0 POUND

\* All the data displayed above was taken on-chain at block 17299869

\* The tokens on industry-standard burn wallets are not included on the top 10 wallets calculations

## Liquidity Ownership

The token does not have liquidity at the moment of the audit, block 17299869

# KISHIELD



## Disclaimer

KISHIELD has conducted an independent audit to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the codes that were provided for the scope of this audit. This audit report does not constitute agreement, acceptance or advocacy for the Project that was audited, and users relying on this audit report should not consider this as having any merit for financial advice in any shape, form or nature. The contracts audited do not account for any economic developments that may be pursued by the Project in question, and that the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are completely free of exploits, bugs, vulnerabilities or deprecation of technologies.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence regardless of the findings presented in this report. Information is provided 'as is', and KISHIELD is under no covenant to the completeness, accuracy or solidity of the contracts audited. In no event will KISHIELD or its partners, employees, agents or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions and/or actions with regards to the information provided in this audit report.

The assessment services provided by KISHIELD is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.