

A low-angle, upward-looking photograph of several tall skyscrapers, creating a sense of height and architectural scale. The image is overlaid with a semi-transparent blue filter.

SMART CONTRACT SECURITY AUDIT

Final report

Plan: Simple

Miner PEPE

March 2024

TABLE OF CONTENTS

1. Introduction	3
2. Analyzed Contracts	3
3. Audit Process	3
3.1 Auto-analysis	3
3.2 Expert audit	3
4. Known issues checked	4
5. Issue Classification	7
6. Issues	7
6.1 High risk issues	7
6.2 Medium risk issues	7
6.3 Low risk issues	7
7. Conclusion	8
8. Disclaimer	9
9. Static analysis	10

INTRODUCTION

The report has been prepared for Miner PEPE.

MPEPE stands out as the initial Pepe coin for mining on Binance Smart Chain. It comes with its own decentralized app and Mobile App, offering miners a 12% daily return in BNB and MPEPE tokens. Additionally, Mpepe's native token boosts the Total Value Locked (TVL) of miners by 2% with each transaction (whether buying or selling), directly benefiting the miner's contract. Only those who hold \$MPEPE/Cake/BNB tokens can kickstart mining activities in the available miner pools and receive daily rewards for lifetime.

Name	Miner PEPE
Audit date	2024-03-05 - 2024-03-05
Language	Solidity
Platform	Binance Smart Chain

ANALYZED CONTRACTS

Name	Address
Redis	0x6d306C2C9CD931160763D99376a68C14D33DC954

AUDIT PROCESS

Our audit structure consists of two stages:

Auto-analysis

- Our automated tools allow us to scan smart contract code and find potential issues
- We hand pick and verify all the issues found by the tools

Expert audit

- Manual analysis of potential issues and vulnerabilities
- Contract code is reviewed thoroughly

KNOWN ISSUES CHECKED

Title	Result
Unencrypted Private Data On-Chain	✓ passed
Code With No Effects	✓ passed
Message call with hardcoded gas amount	✓ passed
Typographical Error	✓ passed
DoS With Block Gas Limit	✓ passed
Presence of unused variables	✓ passed
Incorrect Inheritance Order	✓ passed
Requirement Violation	✓ passed
Weak Sources of Randomness from Chain Attributes	✓ passed
Shadowing State Variables	✓ passed

Incorrect Constructor Name	✓ passed
----------------------------	----------

Block values as a proxy for time	✓ passed
----------------------------------	----------

Authorization through tx.origin	✓ passed
---------------------------------	----------

DoS with Failed Call	✓ passed
----------------------	----------

Delegatecall to Untrusted Callee	✓ passed
----------------------------------	----------

Use of Deprecated Solidity Functions	✓ passed
--------------------------------------	----------

Assert Violation	✓ passed
------------------	----------

State Variable Default Visibility	✓ passed
-----------------------------------	----------

Reentrancy	✓ passed
------------	----------

Unprotected SELFDESTRUCT Instruction	✓ passed
--------------------------------------	----------

Unprotected Ether Withdrawal	✓ passed
------------------------------	----------

Unchecked Call Return Value	✓ passed
-----------------------------	----------

Floating Pragma	✓ passed
-----------------	----------

Outdated Compiler Version	✓ passed
---------------------------	----------

Integer Overflow and Underflow	✓ passed
--------------------------------	----------

Function Default Visibility

✓ passed

ISSUE CLASSIFICATION

High risk	Issues leading to assets theft, locking or any other loss of assets or leading to contract malfunctioning.
Medium risk	Issues that can trigger a contract failure of malfunctioning.
Low risk	Issues that do not affect contract functionality. For example, unoptimised gas usage, outdated or unused code, code style violations, etc.

ISSUES

High risk issues

No issues were found

Medium risk issues

No issues were found

Low risk issues

No issues were found

CONCLUSION

Miner PEPE Redis contract was audited. No risk issues were found.

DISCLAIMER

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without RapidLabs prior written consent.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts RapidLabs to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

STATIC ANALYSIS

INFO:Detectors:

Reentrancy in Redis._transfer(address,address,uint256) (contracts/contract.sol#338-357):

External calls:

- swapTokensForEth(contractTokenBalance) (contracts/contract.sol#353)
 - uniswapV2Router.swapExactTokensForETCSupportingFeeOnTransferTokens(tokenAmount,0,path,treasuryAddress,block.timestamp) (contracts/contract.sol#368-379)
 - uniswapV2Router.swapExactTokensForROSESupportingFeeOnTransferTokens(tokenAmount,0,path,treasuryAddress,block.timestamp) (contracts/contract.sol#382-393)
 - uniswapV2Router.swapExactTokensForAVAXSupportingFeeOnTransferTokens(tokenAmount,0,path,treasuryAddress,block.timestamp) (contracts/contract.sol#396-407)
 - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,treasuryAddress,block.timestamp) (contracts/contract.sol#409-420)

State variables written after the call(s):

- _transferStandard(from,to,amount) (contracts/contract.sol#356)
 - rOwned[sender] = rOwned[sender] - rAmount (contracts/contract.sol#467)
 - rOwned[recipient] = rOwned[recipient] + rTransferAmount (contracts/contract.sol#468)
 - rOwned[address(this)] = rOwned[address(this)] + rTreasury (contracts/contract.sol#473)

Redis.rOwned (contracts/contract.sol#145) can be used in cross function reentrancies:

- Redis._transferStandard(address,address,uint256) (contracts/contract.sol#447-481)
- Redis.balanceOf(address) (contracts/contract.sol#273-275)
- Redis.constructor(string,string,uint8,uint256,address,address,uint16,uint16,address) (contracts/contract.sol#186-237)
- _transferStandard(from,to,amount) (contracts/contract.sol#356)
 - rTotal = rTotal - rReflection (contracts/contract.sol#478)

Redis.rTotal (contracts/contract.sol#151) can be used in cross function reentrancies:

- Redis._getCurrentSupply() (contracts/contract.sol#549-554)

```

- Redis._transferStandard(address,address,uint256) (contracts/
contract.sol#447-481)
- Redis.constructor(string,string,uint8,uint256,address,address,uint16,uint
16,address) (contracts/contract.sol#186-237)
- Redis.tokenFromReflection(uint256) (contracts/contract.sol#316-325)
Reference: https://github.com/crytic/slither/wiki/Detector-
Documentation#reentrancy-vulnerabilities-1
INFO:Detectors:
Reentrancy in Redis._transfer(address,address,uint256) (contracts/
contract.sol#338-357):
  External calls:
    - swapTokensForEth(contractTokenBalance) (contracts/contract.sol#353)
      - uniswapV2Router.swapExactTokensForETCSupportingFeeOnTransferToken
s(tokenAmount,0,path,treasuryAddress,block.timestamp) (contracts/
contract.sol#368-379)
      - uniswapV2Router.swapExactTokensForROSESupportingFeeOnTransferToke
ns(tokenAmount,0,path,treasuryAddress,block.timestamp) (contracts/
contract.sol#382-393)
      - uniswapV2Router.swapExactTokensForAVAXSupportingFeeOnTransferToke
ns(tokenAmount,0,path,treasuryAddress,block.timestamp) (contracts/
contract.sol#396-407)
      - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferToken
s(tokenAmount,0,path,treasuryAddress,block.timestamp) (contracts/
contract.sol#409-420)
    Event emitted after the call(s):
    - Reflected(sender,tReflection) (contracts/contract.sol#479)
      - _transferStandard(from,to,amount) (contracts/contract.sol#356)
    - Transfer(sender,recipient,tTransferAmount) (contracts/contract.sol#470)
      - _transferStandard(from,to,amount) (contracts/contract.sol#356)
    - Transfer(sender,address(this),tTreasury) (contracts/contract.sol#474)
      - _transferStandard(from,to,amount) (contracts/contract.sol#356)
Reentrancy in Redis.swapTokensForEth(uint256) (contracts/contract.sol#359-422):
  External calls:
    - uniswapV2Router.swapExactTokensForETCSupportingFeeOnTransferTokens(tokenA
mount,0,path,treasuryAddress,block.timestamp) (contracts/contract.sol#368-379)
    Event emitted after the call(s):
    - SwapTokensForEthFailed(tokenAmount) (contracts/contract.sol#378)
Reentrancy in Redis.swapTokensForEth(uint256) (contracts/contract.sol#359-422):
  External calls:
    - uniswapV2Router.swapExactTokensForROSESupportingFeeOnTransferTokens(token
Amount,0,path,treasuryAddress,block.timestamp) (contracts/contract.sol#382-393)
    Event emitted after the call(s):

```

```

- SwapTokensForEthFailed(tokenAmount) (contracts/contract.sol#392)
Reentrancy in Redis.swapTokensForEth(uint256) (contracts/contract.sol#359-422):
  External calls:
  - uniswapV2Router.swapExactTokensForAVAXSupportingFeeOnTransferTokens(token
Amount,0,path,treasuryAddress,block.timestamp) (contracts/contract.sol#396-407)
  Event emitted after the call(s):
  - SwapTokensForEthFailed(tokenAmount) (contracts/contract.sol#406)
Reentrancy in Redis.swapTokensForEth(uint256) (contracts/contract.sol#359-422):
  External calls:
  - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenA
mount,0,path,treasuryAddress,block.timestamp) (contracts/contract.sol#409-420)
  Event emitted after the call(s):
  - SwapTokensForEthFailed(tokenAmount) (contracts/contract.sol#419)
Reentrancy in Redis.withdrawTokens(address,address,uint256) (contracts/
contract.sol#428-436):
  External calls:
  - require(bool,string)(IERC20(token).transfer(to,amount),transfer
rejected) (contracts/contract.sol#433)
  Event emitted after the call(s):
  - WithdrawnTokens(token,to,amount) (contracts/contract.sol#435)
Reference: https://github.com/crytic/slither/wiki/Detector-
Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
Redis.swapTokensForEth(uint256) (contracts/contract.sol#359-422) has a high
cyclomatic complexity (12).
Reference: https://github.com/crytic/slither/wiki/Detector-
Documentation#cyclomatic-complexity
INFO:Detectors:
Pragma version0.8.16 (contracts/contract.sol#3) allows old versions
solc-0.8.16 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Detectors:
Function IUniswapV2Router02.WETH() (contracts/contract.sol#84) is not in mixedCase
Function IUniswapV2Router02.WETC() (contracts/contract.sol#86) is not in mixedCase
Function IUniswapV2Router02.WHT() (contracts/contract.sol#88) is not in mixedCase
Function IUniswapV2Router02.WROSE() (contracts/contract.sol#90) is not in
mixedCase
Function IUniswapV2Router02.WAVAX() (contracts/contract.sol#92) is not in
mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-
Documentation#conformance-to-solidity-naming-conventions

```

INFO:Detectors:

Variable Redis._transferStandard(address,address,uint256).rReflection (contracts/contract.sol#460) is too similar to Redis._transferStandard(address,address,uint256).tReflection (contracts/contract.sol#463)

Variable Redis._transferStandard(address,address,uint256).rReflection (contracts/contract.sol#460) is too similar to Redis._getRValues(uint256,uint256,uint256).tReflection (contracts/contract.sol#531)

Variable Redis._getRValues(uint256,uint256,uint256).rReflection (contracts/contract.sol#537) is too similar to Redis._transferStandard(address,address,uint256).tReflection (contracts/contract.sol#463)

Variable Redis._transferStandard(address,address,uint256).rReflection (contracts/contract.sol#460) is too similar to Redis._getValues(bool,uint256).tReflection (contracts/contract.sol#495)

Variable Redis._getRValues(uint256,uint256,uint256).rReflection (contracts/contract.sol#537) is too similar to Redis._getRValues(uint256,uint256,uint256).tReflection (contracts/contract.sol#531)

Variable Redis._getRValues(uint256,uint256,uint256).rReflection (contracts/contract.sol#537) is too similar to Redis._getValues(bool,uint256).tReflection (contracts/contract.sol#495)

Variable Redis._getRValues(uint256,uint256,uint256).rTransferAmount (contracts/contract.sol#539) is too similar to Redis._getValues(bool,uint256).tTransferAmount (contracts/contract.sol#494)

Variable Redis._transferStandard(address,address,uint256).rTransferAmount (contracts/contract.sol#459) is too similar to Redis._transferStandard(address,address,uint256).tTransferAmount (contracts/contract.sol#462)

Variable Redis._transferStandard(address,address,uint256).rTransferAmount (contracts/contract.sol#459) is too similar to Redis._getValues(bool,uint256).tTransferAmount (contracts/contract.sol#494)

Variable Redis._getValues(bool,uint256).rReflection (contracts/contract.sol#502) is too similar to Redis._getRValues(uint256,uint256,uint256).tReflection (contracts/contract.sol#531)

Variable Redis._transferStandard(address,address,uint256).rReflection (contracts/contract.sol#460) is too similar to Redis._getTValues(bool,uint256).tReflection (contracts/contract.sol#523)

Variable Redis._getValues(bool,uint256).rReflection (contracts/contract.sol#502) is too similar to Redis._transferStandard(address,address,uint256).tReflection (contracts/contract.sol#463)

Variable Redis._getValues(bool,uint256).rReflection (contracts/contract.sol#502) is too similar to Redis._getTValues(bool,uint256).tReflection (contracts/contract.sol#523)

Variable Redis._getValues(bool,uint256).rReflection (contracts/contract.sol#502) is too similar to Redis._getValues(bool,uint256).tReflection (contracts/contract.sol#495)

Variable Redis._getRValues(uint256,uint256,uint256).rReflection (contracts/contract.sol#537) is too similar to Redis._getTValues(bool,uint256).tReflection (contracts/contract.sol#523)

Variable Redis._getRValues(uint256,uint256,uint256).rTransferAmount (contracts/contract.sol#539) is too similar to

Redis._getTValues(bool,uint256).tTransferAmount (contracts/contract.sol#525)

Variable Redis._getValues(bool,uint256).rTransferAmount (contracts/contract.sol#501) is too similar to Redis._getValues(bool,uint256).tTransferAmount (contracts/contract.sol#494)

Variable Redis._transferStandard(address,address,uint256).rTransferAmount (contracts/contract.sol#459) is too similar to

Redis._getTValues(bool,uint256).tTransferAmount (contracts/contract.sol#525)

Variable Redis._getValues(bool,uint256).rTransferAmount (contracts/contract.sol#501) is too similar to

Redis._getTValues(bool,uint256).tTransferAmount (contracts/contract.sol#525)

Variable Redis._getRValues(uint256,uint256,uint256).rTransferAmount (contracts/contract.sol#539) is too similar to

Redis._transferStandard(address,address,uint256).tTransferAmount (contracts/contract.sol#462)

Variable Redis._getValues(bool,uint256).rTransferAmount (contracts/contract.sol#501) is too similar to

Redis._transferStandard(address,address,uint256).tTransferAmount (contracts/contract.sol#462)

Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar>

INFO:Detectors:

Redis.uniswapV2Router (contracts/contract.sol#162) should be immutable

Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable>

INFO:Slither:. analyzed (7 contracts with 85 detectors), 37 result(s) found



RAPID LABS

YOUR PROJECT SECURED

RL@RAPIDLABS.FINANCE

RAPIDLABS.FINANCE