


# SMART CONTRACT SECURITY AUDIT


Final report

Plan: Simple

**Pawthereum**

January 2022

 [rugdog.net](https://rugdog.net)

 [the@rugdog.net](mailto:the@rugdog.net)



## CONTENTS

<b>1. Introduction</b>	<b>3</b>
<b>2. Contracts checked</b>	<b>3</b>
<b>3. Audit Process</b>	<b>3</b>
<b>4. Attacks checked</b>	<b>4</b>
<b>5. Classification of issues</b>	<b>5</b>
<b>6. Issues</b>	<b>6</b>
6.1 High severity issues	6
6.2 Medium severity issues	6
6.3 Low severity issues	6
<b>7. Conclusion</b>	<b>8</b>
<b>8. Disclaimer</b>	<b>9</b>

## **INTRODUCTION**

This report has been compiled by the RugDog auditing team for the Pawthereum project.

The reviewed project is a PAWTH token contract.

Name	Pawthereum
Audit date	2022-01-17 - 2022-01-26
Language	Solidity
Network	Binance Smart Chain

## **CONTRACTS CHECKED**

Name	Address
Pawthereum	0x409e215738E31d8aB252016369c2dd9c2008Fee0

## **AUDIT PROCESS**

The code was audited by the team according to the following order:

### Automated analysis

- ♦ Scanning the project's smart contracts with several publicly available automated Solidity analysis tools
- ♦ Manual confirmation of all the issues found by the tools

### Manual audit

- ♦ Thorough manual analysis of smart contracts for security vulnerabilities

Smart contracts' logic check

## **ATTACKS CHECKED**

Title	Check 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	✗ not 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Outdated Compiler Version	✓ passed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

## CLASSIFICATION OF ISSUES

<b>High severity</b>	Issues leading to assets theft, locking or any other loss of assets or leading to contract malfunctioning.
<b>Medium severity</b>	Issues that can trigger a contract failure of malfunctioning.
<b>Low severity</b>	Issues that do now affect contract functionality. For example,



## Low severity

Issues that do not affect the functionality of the code, for example, style violations, gas usage, outdated or unused code, code style violations, etc.

## ❖ ISSUES

### High severity issues

No issues were found

### Medium severity issues

No issues were found

### Low severity issues

#### 1. Not enough gas (Pawthereum)

The loop is used by the `includeAccount()` function to detect and remove addresses from `_excluded`.

The function gets aborted if the list of excluded addresses is long with an `OUT_OF_GAS` exception.

The same goes for the `_getReflectionRate()` while evaluating the total supply.

**Recommendation:** Check for the length of the excluded array

#### 2. Excessive owner privileges (Pawthereum)

If the owner is not renounced, they have excessive privileges:

Initializing liquidity;

Changing fees, the address of IPTokenHolder, marketing, staking wallet and charity address, the maximum transaction amount the maximum number of tokens in a swap;

Withdrawing BNB and ERC20;

Updating Uniswap pair and router;

Including and excluding addresses from taxes, from automatedMarketMakerPairs;

Enabling and disabling swapAndLiquifyMarketing & swapAndLiquifyCharity, taxes, burn, liquidity, marketing, and charity fees;

Recalculating \_liquidityTokensToSwap, \_marketingTokensToSwap, \_charityTokensToSwap values.

## ◆ CONCLUSION

Pawthereum Pawthereum contract was audited. 2 low severity issues were found.

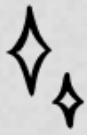


## ❖ **DISCLAIMER**

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 RugDog 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 RugDog 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.



# WOOF!

 [rugdog.net](https://rugdog.net)

 [the@rugdog.net](mailto:the@rugdog.net)

