ApeAudits

Presents

PROANUS

Smart Contract Security Audit

Date 21/June/2021

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and ApeAudits and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (ApeAudits) owe no duty of care towards you or any other person, nor does ApeAudits make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and ApeAudits hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, ApeAudits hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against ApeAudits, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Purpose:

This document details ApeAudits' findings and recommended solutions. This audit was performed on June 20, 2021.

Token Name	PROANUS
Token Symbol	ANUS
Contract Address	0x177c2cfe27db0f0358ca93c3a6d907db83e7f958
Etherscan Link	https://etherscan.io/token/0x177c2cfe27db0f0358ca93c3a6d907db83e7f958
Liquidity Lock Transaction	https://team.finance/view-coin/0x177c2cfe27db0f0358ca93c3a6d907db83e7f958
Ownership Renounce Transaction	https://etherscan.io/tx/0xf5a6b24acdf22155d84fccd6c3b420f956383d4044b77bd5cad7bf947b47dae7

Findings

NO.	Audit Items	Audit Subclass	Audit Subclass Result
1	Overflow Audit.	N/A	Passed
2	Race Conditions Audit	N/A	Passed
3	Authority Control Audit	Permission Vulnerability Audit Excessive Auditing Authority	Passed Passed
4	Safe Design Audit	Zeppelin Module Safe Compiler Version Hard-Coded Version Fallback Function Safeuse Show Coding Security Function Return Value Security Call Function Security.	Passed Passed Passed Passed Passed Passed Passed
5	Denial of Service Audit	N/A	Passed
6	Gas Optimization Audit	N/A	Passed
7	Design Logic Audit	N/A	Passed
8	Malicious Event Log Audit	N/A	Passed
9	“False Deposit” Vulnerability Audit	N/A	Passed
10	Uninitialized Storage Pointers Audit	N/A	Passed
11	Arithmetic Accuracy Deviation Audit	N/A	Passed

Audit Result: Passed

Audit Date: June 21st of 2021

Proanus.sol

High Severity Issues:

- None Found.

Medium Severity Issues:

- None Found.

Low Severity Issues:

- None Found

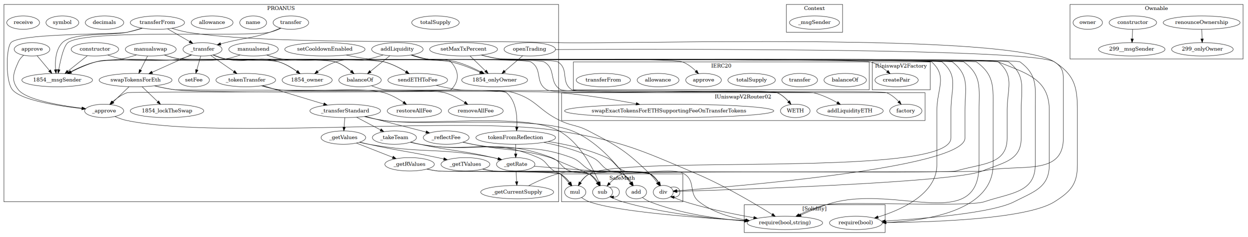
Conclusion:

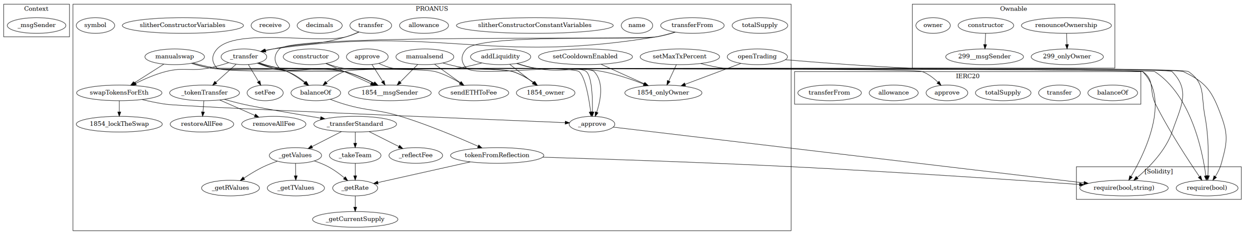
Smart Contract contain low severity issues. No medium and high severity issues were detected. LP locking details not provided by team.

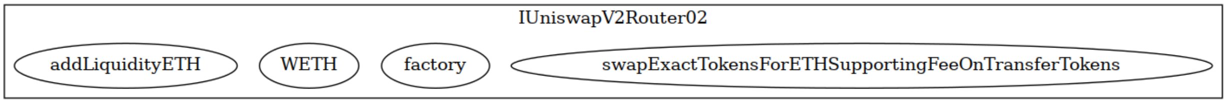
Ape Auditor's Remider:

Please check the disclaimer and above and note, the audit makes no statement or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mention in the report and does not include any other potential contacts by the Owner.

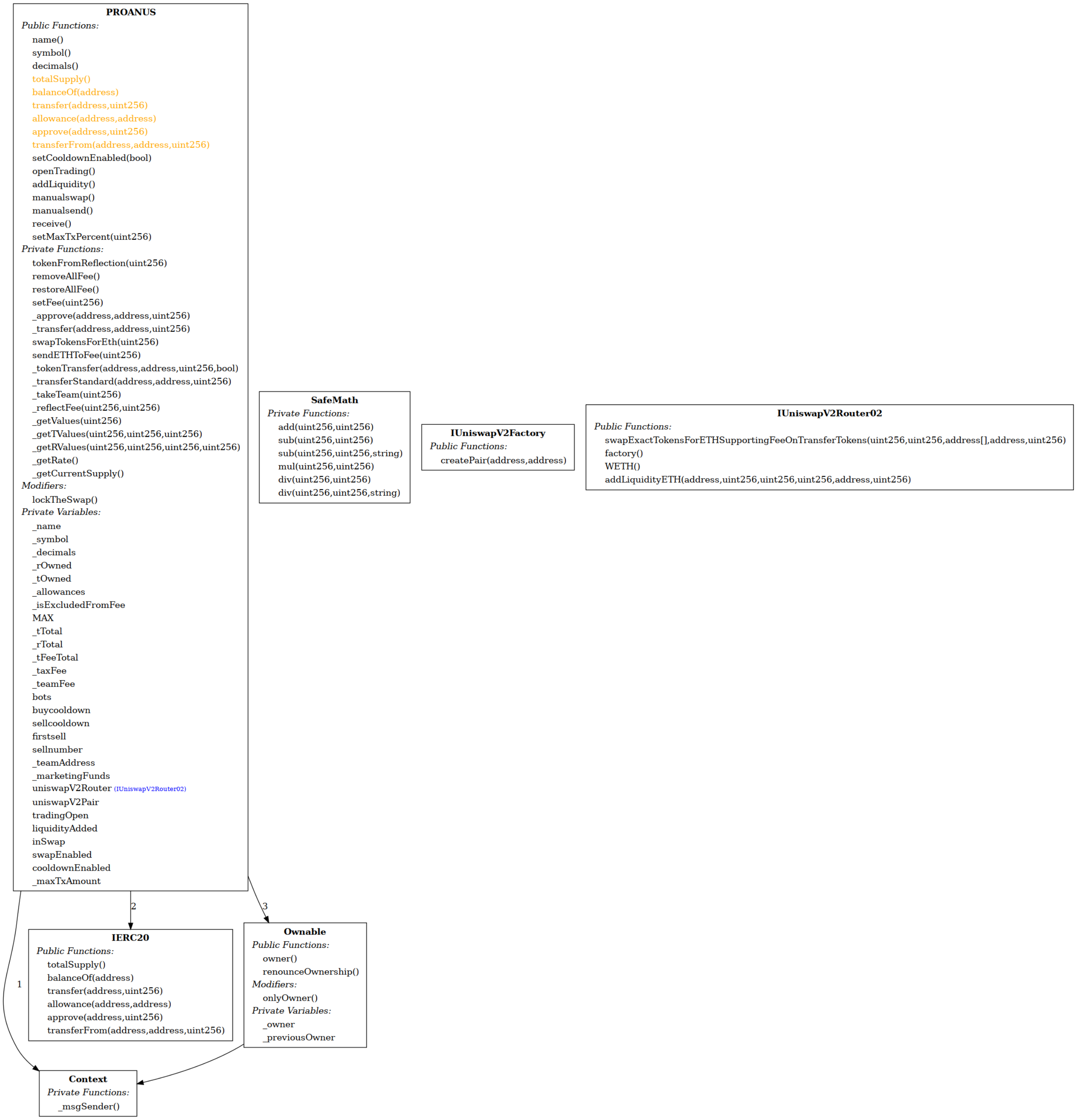
Appendix A - Call Graph







Appendix B - Inheritance Graph



Appendix C - Function Information

+ Contract IERC20

  - From IERC20

    - allowance(address,address) (external)

    - approve(address,uint256) (external)

    - balanceOf(address) (external)

    - totalSupply() (external)

    - transfer(address,uint256) (external)

    - transferFrom(address,address,uint256) (external)

+ Contract SafeMath (Most derived contract)

  - From SafeMath

    - add(uint256,uint256) (internal)

    - div(uint256,uint256) (internal)

    - div(uint256,uint256,string) (internal)

    - mul(uint256,uint256) (internal)

    - sub(uint256,uint256) (internal)

    - sub(uint256,uint256,string) (internal)

+ Contract Ownable

  - From Context

    - _msgSender() (internal)

```
 - From Ownable
   - constructor() (public)
   - owner() (public)
   - renounceOwnership() (public)

+ Contract IUniswapV2Factory (Most derived contract)
 - From IUniswapV2Factory
   - createPair(address,address) (external)

+ Contract IUniswapV2Router02 (Most derived contract)
 - From IUniswapV2Router02
   - WETH() (external)
   - addLiquidityETH(address,uint256,uint256,uint256,address,uint256) (external)
   - factory() (external)
   -
swapExactTokensForETHSupportingFeeOnTransferTokens(uint256,uint256,address[],address,ui
nt256) (external)

+ Contract PROANUS (Most derived contract)
 - From Ownable
   - constructor() (public)
   - owner() (public)
```

  - renounceOwnership() (public)

- From Context

  - _msgSender() (internal)

- From PROANUS

  - _approve(address,address,uint256) (private)

  - _getCurrentSupply() (private)

  - _getRValues(uint256,uint256,uint256,uint256) (private)

  - _getRate() (private)

  - _getTValues(uint256,uint256,uint256) (private)

  - _getValues(uint256) (private)

  - _reflectFee(uint256,uint256) (private)

  - _takeTeam(uint256) (private)

  - _tokenTransfer(address,address,uint256,bool) (private)

  - _transfer(address,address,uint256) (private)

  - _transferStandard(address,address,uint256) (private)

  - addLiquidity() (external)

  - allowance(address,address) (public)

  - approve(address,uint256) (public)

  - balanceOf(address) (public)

  - constructor(address,address) (public)

  - decimals() (public)

  - manualsend() (external)

- manualswap() (external)
- name() (public)
- openTrading() (public)
- receive() (external)
- removeAllFee() (private)
- restoreAllFee() (private)
- sendETHToFee(uint256) (private)
- setCooldownEnabled(bool) (external)
- setFee(uint256) (private)
- setMaxTxPercent(uint256) (external)
- swapTokensForEth(uint256) (private)
- symbol() (public)
- tokenFromReflection(uint256) (private)
- totalSupply() (public)
- transfer(address,uint256) (public)
- transferFrom(address,address,uint256) (public)