

# Security Assessment



*Audited by Pegasy Cybersecurity*



PEGASY  
CYBERSECURITY

*Prometheus*

*February 2022*

### Table of Contents

Executive Summary.....	2
Testing Summary.....	3
Vulnerability Summary.....	4
Findings.....	5
Findings Review.....	5
01 Slither - Tautology or contradiction.....	5
Manual Analysis.....	6
Disclaimer.....	7
Auditors Disclaimer.....	7
Technical Disclaimer.....	7

### Executive Summary

This report has been prepared for Prometheus to discover issues and vulnerabilities in the source code of their smart contract. A comprehensive examination has been performed using static analysis, symbolic execution and manual review techniques.

This audit process pays special attention to the following considerations:

- Testing the smart contract against common vectors of attacks.
- Assessing the code base to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Thorough line-by-line manual review of the entire code-base.

# Testing Summary







### Smart Contract Audit

This report has been prepared as a product of the Smart Contract Audit requested by Prometheus.

This audit was conducted to discover issues and vulnerabilities in the source code of Prometheus Smart Contract.

Type	Smart Contract
Platform	EVM
Network	Binance Smart Chain
Language	Solidity
Files Analyzed	Prometheus.sol
Prometheus.sol sha256-sum	c98641aacf2dd8810e6723c2480934ad01a43cf3ac6b473d7cf3013896ca227e
Request date	January 26, 2022
Delivery date	February 05, 2022
Methods	A comprehensive examination has been performed using static analysis, symbolic execution and manual review.

## Vulnerability Summary



Total issues :	3	Status:
 Critical	0	
 Major	0	
 Medium	1	Solved
 Minor	0	
 Informational	2	All Solved
 Discussion	0	

Static analysis has been performed using Slither static analyzer and Mythril for symbolic execution.

No vulnerabilities have been discovered using Mythril analyzer, one issue discovered with Slither is reported in the findings list.

# Findings

## Findings Review

ID	Title	Category	Severity	Status
01	Tautology or contradiction	Tautology or contradiction	 Medium	Solved
02	Remove unused commented code	Coding Style	 Informational	Solved
03	Remove Testing functionalities	Coding Style	 Informational	Solved

## 01 Slither - Tautology or contradiction

Slither analyzer output:

```
Prometheus._applyFees(address,address,uint256,uint256,uint256,uint256,uint16,address)
(Prometheus_flat.sol) contains a tautology or contradiction:
- require(bool,string)(remainingAfterTax >= 0,Tax > to amount) (Prometheus_flat.sol#2911)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#tautology-or-contradiction
```

### Manual Analysis

The source code of the smart contract has been manually audited for the following known attacks on solidity language:

- Reentrancy
- Oracle Manipulation
- Frontrunning
- Timestamp Dependence
- Insecure Arithmetic
- Denial of Service
- Griefing
- Force Feeding

During manual code review, no flaws have been detected.

## Disclaimer

### Auditors Disclaimer

The smart contracts given for audit have been analyzed in accordance with the best industry practices at the date of this report, in relation to: Cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report, (Source Code); the Source Code compilation, deployment and functionality (performing the intended functions).

Because the total number of test cases are unlimited, the audit makes no statements or warranties on security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bug-free status or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only - we recommend proceeding with several independent audits and a public bug bounty program to ensure security of smart contracts.

### Technical Disclaimer

Smart contracts are deployed and executed on the Binance Smart Chain platform. The platform, its programming language, and other software related to the smart contract can have their own vulnerabilities that can lead to hacks. Thus, the audit cannot guarantee explicit security of the audited smart contracts.