PROMETHEUS

PROMETHEUS TOKEN INDEPENDENT SECURITY AUDIT

14.1.2022

1. Introduction

This is the audit security report of the token Prometheus.

Our audit focuses on the security and on the quality of the Prometheus smart contract code to ensure that the contracts can be trusted by all the users.

1.1 Testing methodology

The smart contract was analyzed rigorously using following points

- ☐ Automatic detections of flaws with software (using Mythril)
- ☐ Static analysis code line by line to detect most common vulnerabilities.
- ☐ Testnet deployment and dynamic analysis to find logical vulnerabilities.
- ☐ Custom unit testing to see if each function works as expected
- ☐ Search for possible optimisation and bad practices.

1.2 Project informations

Name	Prometheus
Network	Binance Smart Chain
Language	Solidity
Files analyzed	Token.sol

1.3 Legend

Security flaws are classified using a standard scale going from 0 to 10. (10 is critical and 0 is minor.)

Level	Description	Severity
Critical	Flaw which allows the attacker to steal money or to break the contract without much difficulty.	7-10
High	Possible use of the contract in a dangerous intended way.	5-7
Medium	Possible use of the contract in an intended way if some conditions are met, these issues should be fixed.	3-5
Low	Bad practices or low severity issues.	1-3
Information	Gas optimization or warnings, these issues are usually not dangerous.	0-1

2. Results

2.1 Summary

After the rigorous process of analyzing the code.
No dangerous vulnerabilities were found. The developers successfully paid attention to the security.

As a result, the contract can be considered as safe to use for all users in terms of security. However, some minor issues still exist in the code but with a neglectable impact for the users.

EDIT : All issues were corrected by the developer

2.2 Detailed flaws, suggestions and recommendations

1- Misleading Comments		
Name	**Severity**	**Location**
Misleading Comments	3.0 Medium	Token.sol:188
Description		
`uint private constant DIVISOR = 10000;  // Allows precision to 0.0001%` If the divisor is 10 000, the precision is 0.01% and not 0.0001%, relying in this comment can lead to incorrect estimation of token number in the next functions especially for high numbers or by another user.		
Recommandation		

Change 0.0001 to 0.01 in the code, and check if the amounts are right in others parts of the code

Corrected

2- useless feature

Name	Severity	Location
Useless feature	4.0 Medium	Token.sol:626

Description

```
354: _setMaxTransactionSize(10)

626:  function _setMaxTransactionSize(uint16 _newMaxTransactionRatio)
private
```

After calculation MaxTransaction amount is equal to 1.000.000.
The total supply is 1.000.000.000, but 991.000.000 token were burnt, as a result the real supply is 9.000.000

MaxTransaction amount is thus to high (11% of the total supply)

As a result, the anti whale system will not work and is useless

Recommandation

Decrease the the maxtransactionsize line 354

Corrected

3- useless feature

Name	Severity	Location
Useless feature	4.0 Medium	Token.sol:626

Description

```
353: _setMaxWalletSize(100)

637:  function _setMaxTransactionSize(uint16 _newMaxTransactionRatio)
private
```

After calculation `MaxWalletSize` amount is equal to 10.000.000.
The total supply is 1.000.000.000, but 991.000.000 token were burnt, as a result the real supply is 9.000.000

`MaxWalletSize` amount is thus to high (111% of the real total supply)

As a result, the max wallet system will not work and is useless

Recommandation

Decrease the the maxWalletSize line 354

Corrected

4- Bad practice

Name	Severity	Location
Bad practice	0.0 Information	Token.sol:1061

Description

```
else if (_GlobalFees.CashoutFeesEnabled && transferType ==
T_CASHOUT_REWARDS)
```

Don’t need to use an “else if” if the last block executed for sure

Recommandation

Replace the “else if” by “else”

Corrected

3. Complete Flaws Checklist

Identifiant	Name	Passed ?
SWC-107	Reentrancy	YES
SWC-101	Arithmetical flaws (underflow, overflow, round....)	YES
SWC-113 (128)	DoS (Denial of services)	YES
SWC-100	Insufficient permissions	YES
SWC-102	Outdated compiler or libraries	YES
SWC-104 (112)	Unsecure external call or delegatecall	YES
SWC-115	Relying on tx.origin (authentification)	YES
	Costly code in gas	YES
	Logical Flaws	YES
SWC-132	Unexpected Ether received	YES
SWC-120	Weak random	YES
	Insufficient validation of external inputs	YES
SWC-114	Race conditions	YES
	To much dependance on timestamp	YES
	Unsecure extcodesize	YES
	Unsecure oracle calls	YES
SWC-136	Relying on private data	YES
	Bad practices	YES
	Bad error handling	YES
	“Economics” flaws (see part 4)	YES
	Misleading comments	YES
SWC-105	Security of fallback, receive and withdrawal function	YES
SWC-109	Uninitialized storage pointer	YES
	Dangerous or incorrect implementation of the proxy	YES
	Anti-whale system	YES

MaxWallet size	YES

More informations on SWC classification : https://swcregistry.io/

4. 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 the auditor's 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 contract’s auditor 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.

Blockchain technology and cryptographic assets present a high level of ongoing risk. Auditor’s position is that each company and individual are responsible for their own due diligence and continuous Security.

Auditor's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

Blockchain technology and cryptographic assets present a high level of ongoing risk. Auditor’s position is that each company and individual are responsible for their own due diligence and continuous Security.

Auditor's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.