

SOLIDITY AUDIT



Smart Contract Solidity Audit

Audit Details:

Audited project: Paragominas Token (PFC)

Deployer Address: 0x8f6127911fe102aa361a2f52c895b9d4441ecc1d

Blockchain: Binance Smart Chain

Jan 26

Audit

This document may contain confidential information about IT systems and the intellectual property of the Customer as well as information about potential vulnerabilities and methods of their exploitation. The report containing confidential information can be used internally by the Customer, or it can be disclosed publicly after all vulnerabilities are fixed - upon a decision of the Customer.

Introduction

Solidity Audit (Consultant) was contracted by Paragominas Token (PFC) (Customer) to conduct a Smart Contract Code Review and Security Analysis. This report presents the findings of the security assessment of Customer's smart contract. Scope The scope of the project is main net smart contracts that can be found on BSCscan:

<https://bscscan.com/token/0x8f6127911fe102aa361a2f52c895b9d4441ecc1d>

We have scanned this smart contract for commonly known and more specific vulnerabilities. List of the commonly known vulnerabilities that are considered:

Category	Check Item
Code review	<ul style="list-style-type: none">▪ Reentrancy▪ Ownership Takeover▪ Timestamp Dependence▪ Gas Limit and Loops▪ DoS with (Unexpected) Throw▪ DoS with Block Gas Limit▪ Transaction-Ordering Dependence▪ Style guide violation▪ Costly Loop▪ ERC20 API violation▪ Unchecked external call▪ Unchecked math▪ Unsafe type inference▪ Implicit visibility level▪ Deployment Consistency▪ Repository Consistency▪ Data Consistency
Functional review	<ul style="list-style-type: none">▪ Business Logics Review▪ Functionality Checks▪ Access Control & Authorization▪ Escrow manipulation▪ Token Supply manipulation▪ Assets integrity▪ User Balances manipulation▪ Kill-Switch Mechanism▪ Operation Trails & Event Generation

Our team performed an analysis of code functionality, manual audit, and automated checks with Mythril and Slither. All issues found during automated analysis were manually reviewed, and important vulnerabilities are presented in the Audit overview section. A general overview is presented in AS-IS section, and all found issues can be found in the Audit overview section.

Security engineers found **2** medium, **1** informational issue during the audit.

Notice: the audit scope is limited and not include all files in the repository. Though, reviewed contracts are secure, we may not guarantee secureness of contracts that are not in the scope.

ANALYSIS, GRAPHS AND UML DIAGRAM



SafeMath

safeAdd

safeSub

safeMul

safeDiv

ERC20Interface

totalSupply

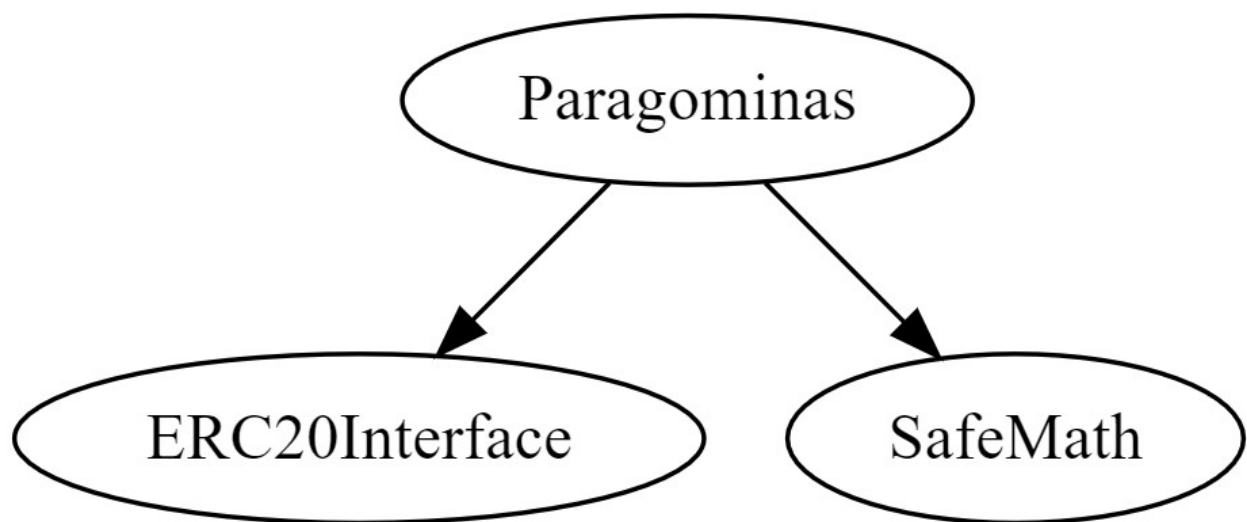
balanceOf

allowance

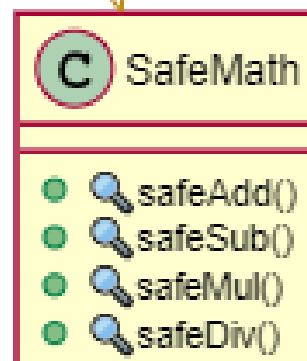
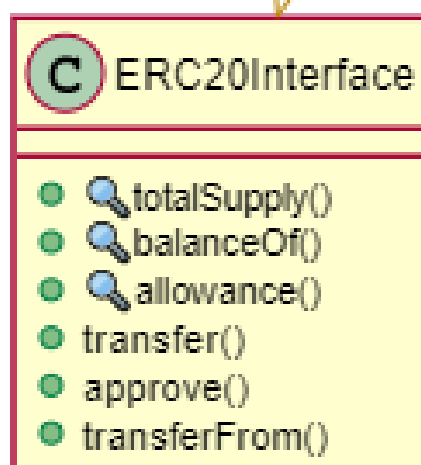
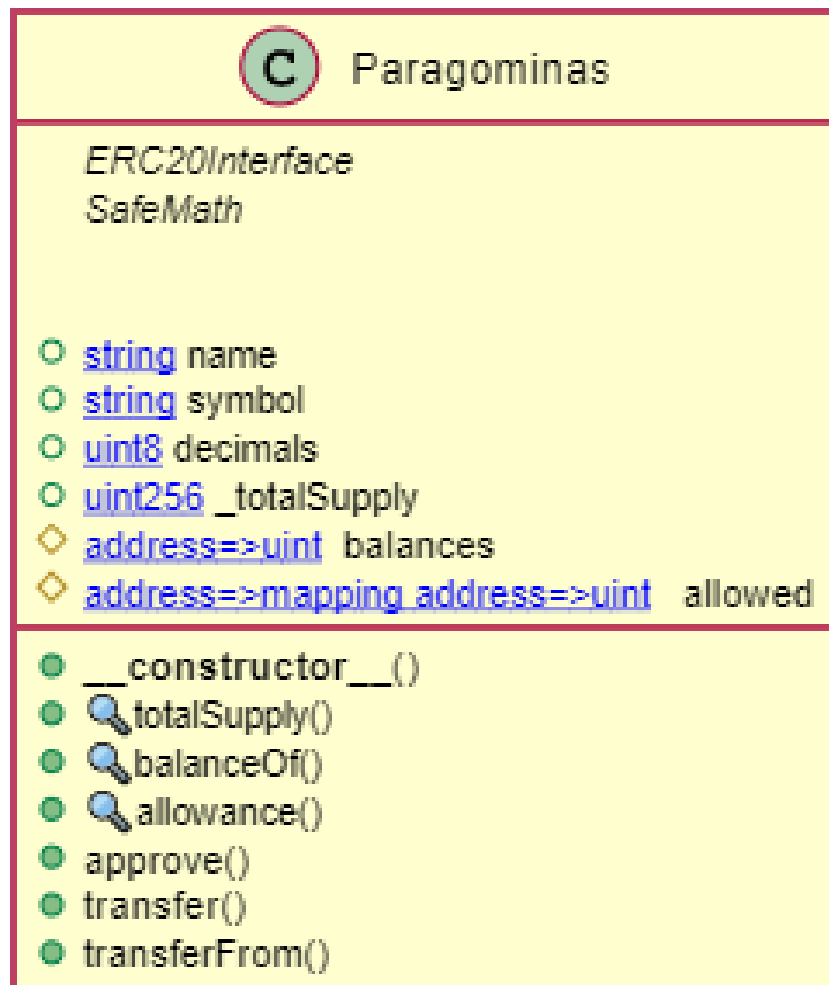
transfer

approve

transferFrom



```
1 | Sighash | Function Signature
2 | =====
3 | 18160ddd => totalSupply()
4 | 70a08231 => balanceOf(address)
5 | dd62ed3e => allowance(address,address)
6 | a9059cbb => transfer(address,uint256)
7 | 095ea7b3 => approve(address,uint256)
8 | 23b872dd => transferFrom(address,address,uint256)
9 | e6cb9013 => safeAdd(uint256,uint256)
10 | a293d1e8 => safeSub(uint256,uint256)
11 | d05c78da => safeMul(uint256,uint256)
12 | b5931f7c => safeDiv(uint256,uint256)
13 |
```



Solidity Audit 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). 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, bugfree 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 blockchain platform. The platform, its programming language, and other software related to the smart contract can have its vulnerabilities that can lead to hacks. Thus, the audit can't guarantee the explicit security of the audited smart contracts.