

SOLIDITY AUDIT



Smart Contract Solidity Audit

Audit Details:

Audited project: USDT PRO

Deployer Address: 0x7E3Ec175a03349b356e2d96D1CD706A19022c102

Blockchain: Binance Smart Chain

Jan 07

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 USDT PRO (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/0x7E3Ec175a03349b356e2d96D1CD706A19022c102>

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

Contracts Description Table

Contract	Type	Bases	Visibility	Mutability	Modifiers
SafeMath	Library				
└─┬─	add	Internal	🔒		
└─┬─	sub	Internal	🔒		
└─┬─	sub	Internal	🔒		
└─┬─	mul	Internal	🔒		
└─┬─	div	Internal	🔒		
└─┬─	div	Internal	🔒		
IBEP20	Interface				
└─┬─	totalSupply	External	🔒	NO	
└─┬─	decimals	External	🔒	NO	
└─┬─	symbol	External	🔒	NO	
└─┬─	name	External	🔒	NO	
└─┬─	getOwner	External	🔒	NO	
└─┬─	balanceOf	External	🔒	NO	
└─┬─	transfer	External	🔒	🔴	NO
└─┬─	allowance	External	🔒	NO	
└─┬─	approve	External	🔒	🔴	NO
└─┬─	transferFrom	External	🔒	🔴	NO
Auth	Implementation				
└─┬─	<Constructor>	Public	🔒	🔴	NO
└─┬─	authorize	Public	🔒	🔴	onlyOwner
└─┬─	unauthorize	Public	🔒	🔴	onlyOwner
└─┬─	isOwner	Public	🔒	NO	
└─┬─	isAuthorized	Public	🔒	NO	
└─┬─	transferOwnership	Public	🔒	🔴	onlyOwner
IDEXFactory	Interface				
└─┬─	createPair	External	🔒	🔴	NO
IDEXRouter	Interface				
└─┬─	factory	External	🔒	NO	
└─┬─	WETH	External	🔒	NO	
└─┬─	addLiquidity	External	🔒	🔴	NO
└─┬─	addLiquidityETH	External	🔒	🟢	NO
└─┬─	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	🔒	🔴	NO
└─┬─	swapExactETHForTokensSupportingFeeOnTransferTokens	External	🔒	🟢	NO
└─┬─	swapExactTokensForETHSupportingFeeOnTransferTokens	External	🔒	🔴	NO

```

**IDividendDistributor** | Interface | |||
| L | setDistributionCriteria | External | 🔒 | 🔴 | NO |
| L | setShare | External | 🔒 | 🔴 | NO |
| L | deposit | External | 🔒 | 🟢 | NO |
| L | process | External | 🔒 | 🔴 | NO |
|||||
**DividendDistributor** | Implementation | IDividendDistributor |||
| L | <Constructor> | Public | 🔒 | 🔴 | NO |
| L | setDistributionCriteria | External | 🔒 | 🔴 | onlyToken |
| L | setShare | External | 🔒 | 🔴 | onlyToken |
| L | deposit | External | 🔒 | 🟢 | onlyToken |
| L | process | External | 🔒 | 🔴 | onlyToken |
| L | shouldDistribute | Internal | 🔒 | | |
| L | distributeDividend | Internal | 🔒 | 🔴 | |
| L | claimDividend | External | 🔒 | 🔴 | onlyToken |
| L | getUnpaidEarnings | Public | 🔒 | | NO |
| L | getCumulativeDividends | Internal | 🔒 | | |
| L | addShareholder | Internal | 🔒 | 🔴 | |
| L | removeShareholder | Internal | 🔒 | 🔴 | |
|||||
**USDTPRO** | Implementation | IBEP20, Auth |||
| L | <Constructor> | Public | 🔒 | 🔴 | Auth |
| L | <Receive Ether> | External | 🔒 | 🟢 | NO |
| L | totalSupply | External | 🔒 | | NO |
| L | decimals | External | 🔒 | | NO |
| L | symbol | External | 🔒 | | NO |
| L | name | External | 🔒 | | NO |
| L | getOwner | External | 🔒 | | NO |
| L | balanceOf | Public | 🔒 | | NO |
| L | allowance | External | 🔒 | | NO |
| L | approve | Public | 🔒 | 🔴 | NO |
| L | approveMax | External | 🔒 | 🔴 | NO |
| L | transfer | External | 🔒 | 🔴 | NO |
| L | transferFrom | External | 🔒 | 🔴 | NO |
| L | _transferFrom | Internal | 🔒 | 🔴 | |
| L | _basicTransfer | Internal | 🔒 | 🔴 | |
| L | checkTxLimit | Internal | 🔒 | | |
| L | shouldTakeFee | Internal | 🔒 | | |
| L | getTotalFee | Public | 🔒 | | NO |
| L | getMultipliedFee | Public | 🔒 | | NO |
| L | takeFee | Internal | 🔒 | 🔴 | |
| L | shouldSwapBack | Internal | 🔒 | | |
| L | swapBack | Internal | 🔒 | 🔴 | swapping |
| L | shouldAutoBuyback | Internal | 🔒 | | |
| L | triggerManualBuyback | External | 🔒 | 🔴 | authorized |
| L | removeMaxWallet | Public | 🔒 | 🔴 | onlyOwner |
| L | clearBuybackMultiplier | External | 🔒 | 🔴 | authorized |
| L | triggerAutoBuyback | Internal | 🔒 | 🔴 | |
| L | buyTokens | Internal | 🔒 | 🔴 | swapping |
| L | setLockTransfer | Public | 🔒 | 🔴 | onlyOwner |

```

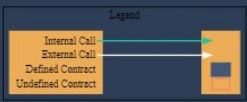
```

| L | removeMaxWallet | Public | | | | onlyOwner | |
| L | clearBuybackMultiplier | External | | | | authorized |
| L | triggerAutoBuyback | Internal | | | | |
| L | buyTokens | Internal | | | | swapping |
| L | setLockTransfer | Public | | | | onlyOwner |
| L | setAutoBuybackSettings | External | | | | authorized |
| L | setBuybackMultiplierSettings | External | | | | authorized |
| L | launched | Internal | | | | |
| L | launch | Internal | | | | |
| L | setTxLimit | External | | | | | authorized |
| L | setIsDividendExempt | External | | | | | authorized |
| L | setIsFeeExempt | External | | | | | authorized |
| L | setIsTxLimitExempt | External | | | | | authorized |
| L | setFees | External | | | | | authorized |
| L | setFeeReceivers | External | | | | | authorized |
| L | setSwapBackSettings | External | | | | | authorized |
| L | setTargetLiquidity | External | | | | | authorized |
| L | manualSend | External | | | | | authorized |
| L | updateManualSend | Public | | | | | onlyOwner |
| L | setDistributionCriteria | External | | | | | authorized |
| L | claimDividend | External | | | | | NO |
| L | getUnpaidEarnings | Public | | | | | NO |
| L | setDistributorSettings | External | | | | | authorized |
| L | getCirculatingSupply | Public | | | | | NO |
| L | getLiquidityBacking | Public | | | | | NO |
| L | isOverLiquified | Public | | | | | NO |

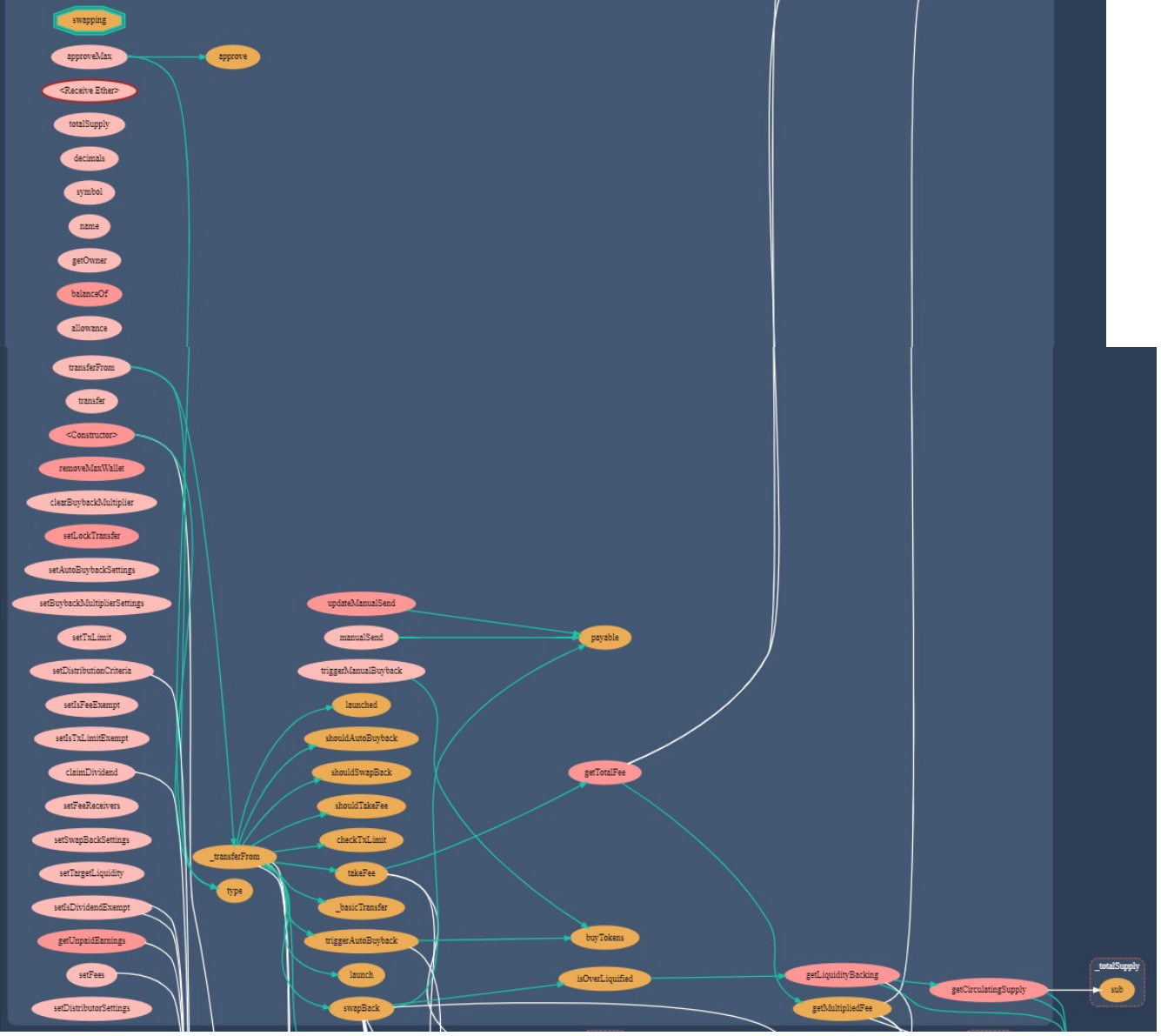
```

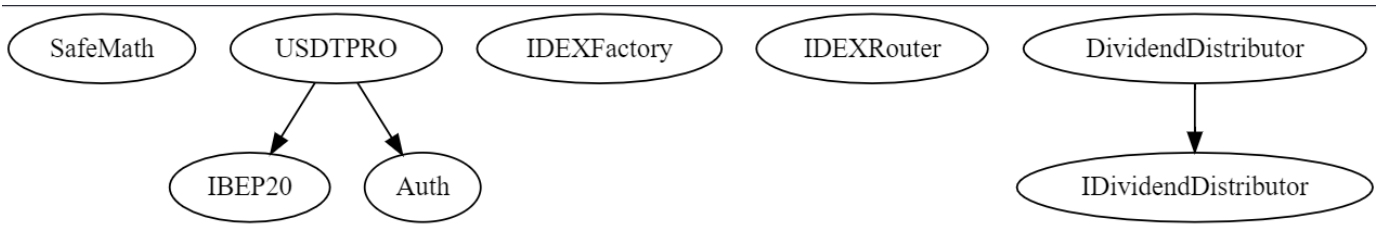
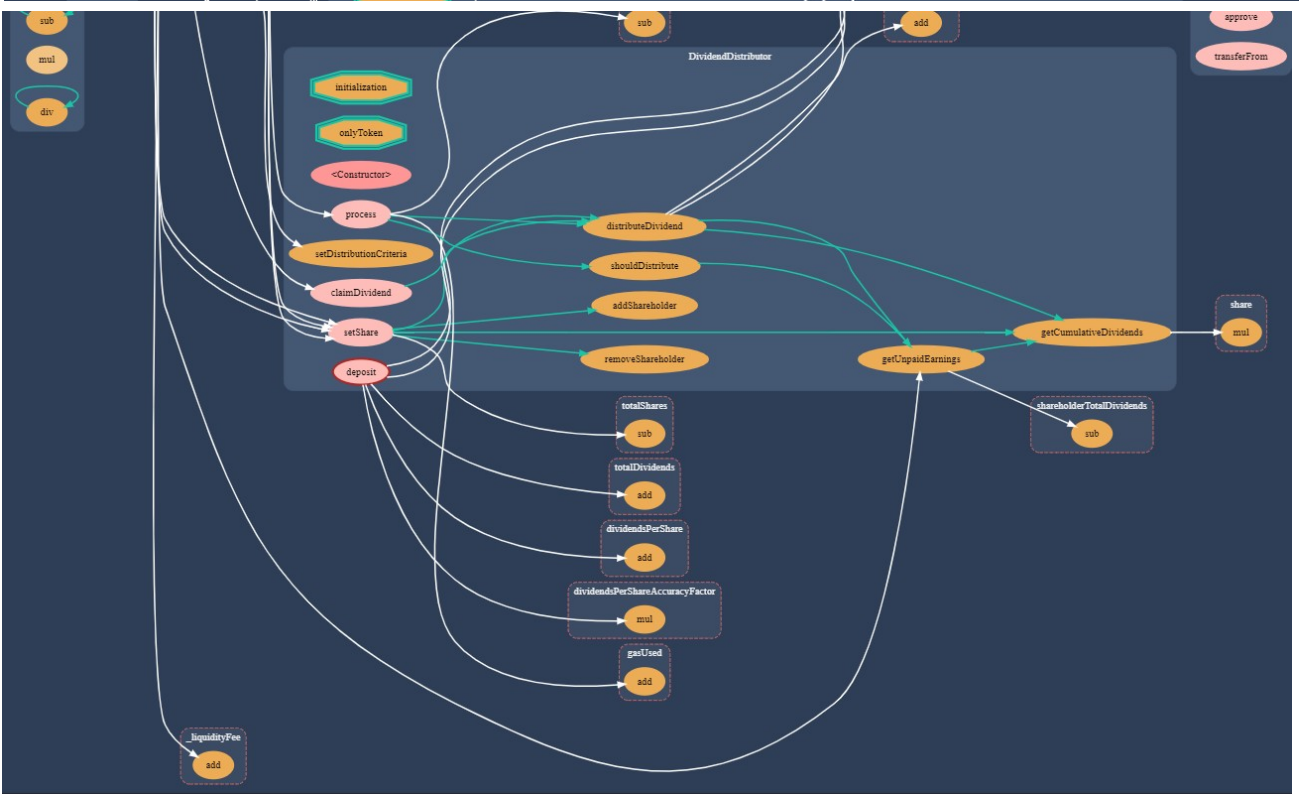
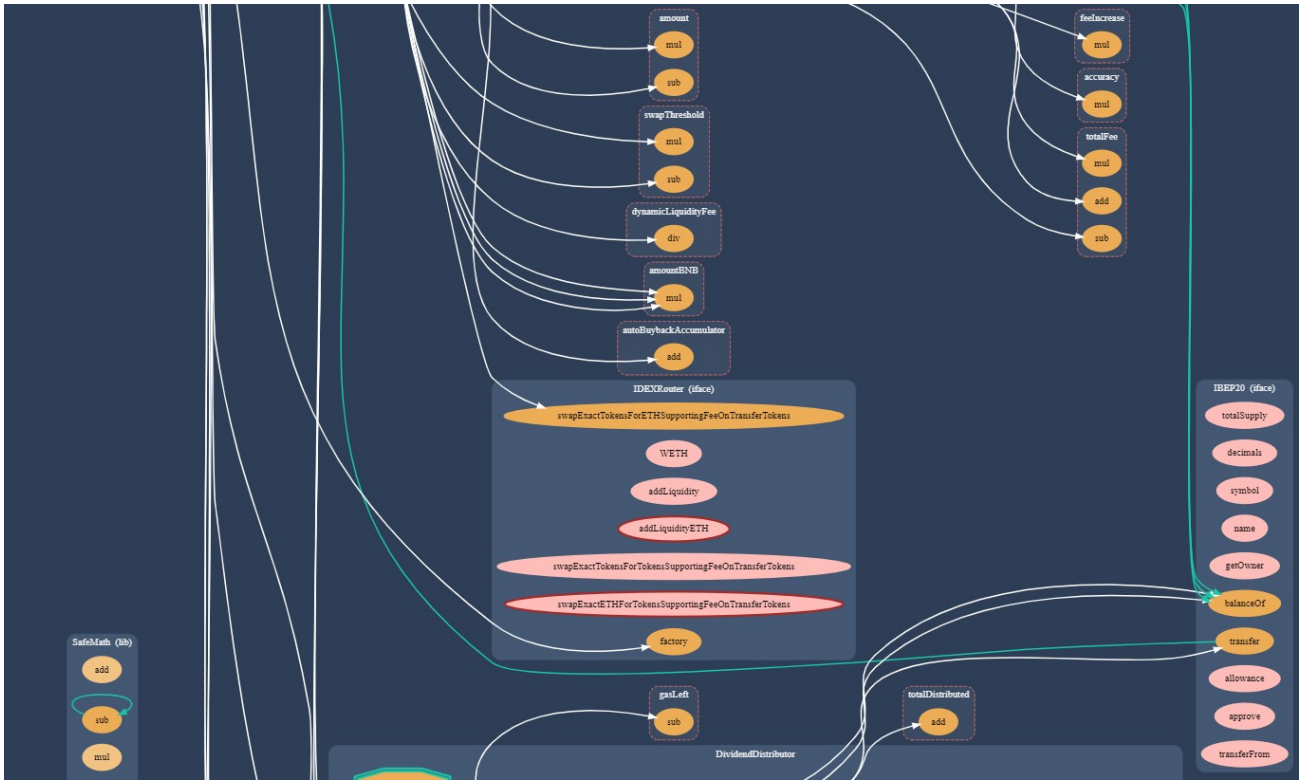
Legend

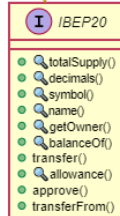
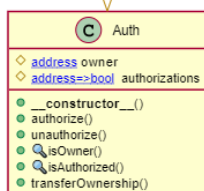
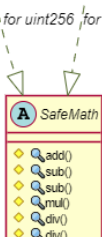
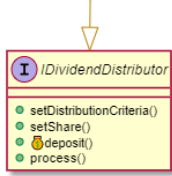
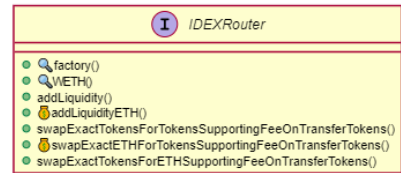
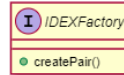
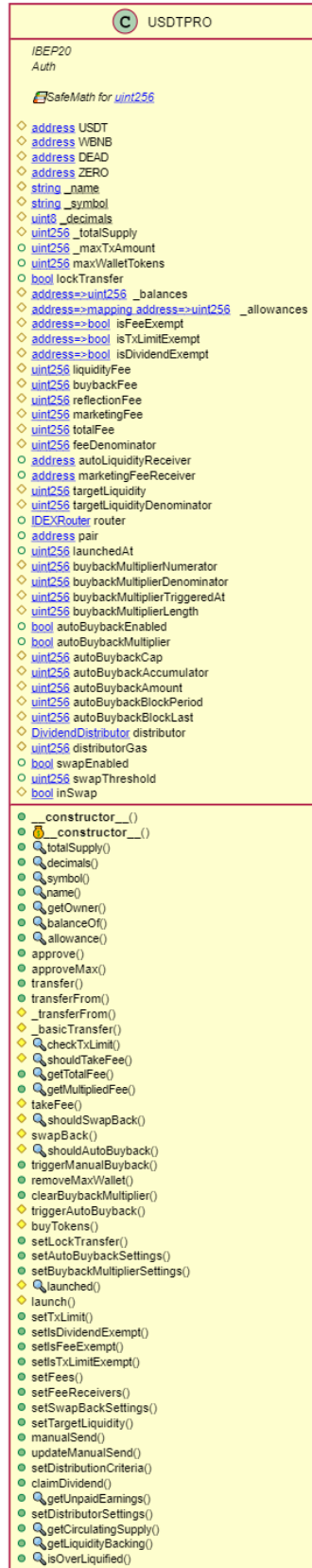
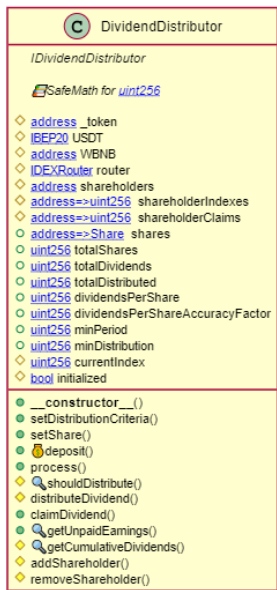
Symbol	Meaning
●	Function can modify state
🔒	Function is payable



USDTPRO







for uint256 for uint256

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.