



Security Assessment
DogeFarm - Audit

TechRight Verified on 01 Apr, 2023



Table of contents

- [Disclaimer](#)
- [Description](#)
- [Vulnerability & Risk Level](#)
- [Auditing Strategy and Techniques](#)
- [Tested Contract Files](#)
- [Scope](#)
 - [Source Units in Scope](#)
 - [Out of Scope](#)
 - [Excluded Source Units](#)
 - [Duplicate Source Units](#)
 - [Doppelganger Contracts](#)
- [Report Overview](#)
 - [Risk Summary](#)
 - [Source Lines](#)
 - [Inline Documentation](#)
 - [Components](#)
 - [Exposed Functions](#)
 - [StateVariables](#)
 - [Capabilities](#)
 - [Dependencies](#)
 - [Totals](#)
- [Detectors Issue](#)

Disclaimer

TechRight.io Reports do not constitute an endorsement or disapproval of any specific project or team, and they should not be taken as an indication of the economic value of any product or asset created by a team. Additionally, TechRight.io does not perform testing or auditing of integration with external contracts or services like Unicrypt, Uniswap, PancakeSwap, and others.

TechRight.io Audits do not offer any assurance or pledge about the complete absence of bugs in the evaluated technology, and they do not give any hint about the owners of the technology. These audits should not be relied upon to make any investment or participation decisions in any specific project, nor should they be used as any form of investment advice.

TechRight.io Reports involve a comprehensive auditing process to support our clients in enhancing their code quality while reducing the risk associated with blockchain technology and cryptographic assets. Please note that every company and individual is responsible for conducting their own due diligence and maintaining continuous security. Please note that TechRight does not guarantee the security or functionality of the technology we confirm to evaluate.

Description

Network

zkSync

Vulnerability & Risk Level

Risk represents the probability that a certain source-threat will exploit vulnerability, and the impact of that event on the organization or system. Risk Level is computed based on CVSS version 3.0

Level	Value	Vulnerability	Risk (Required Action)
Critical	9 - 10	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Immediate action to reduce risk level.
High	7 - 8.9	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.	Implementation of corrective actions as soon as possible.
Medium	4 - 6.9	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.	Implementation of corrective actions in a certain period.
Low	2 - 3.9	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
Informational	0 - 1.9	A vulnerability that has informational character but is not affecting any of the code.	An observation that does not determine a level of risk

Auditing Strategy and Techniques Applied

During the evaluation process, the repository was thoroughly examined to identify any security-related concerns, assess code quality, and ensure adherence to specifications and best practices. Our team of expert pentesters and smart contract developers reviewed the code line-by-line and documented any issues identified.

Methodology

The auditing process follows a step-by-step routine:

1. Code review that includes:
 - i. Review of the specifications, sources and instructions provided to TechRight to ensure a thorough understanding of the size, scope, and functionality of the smart contract's.
 - ii. Manual review of code, which involves carefully reading the source code line-by-line to identify potential vulnerabilities.
 - iii. Comparison to specification, which is the process of confirming whether the code performs as described in the specifications, sources, and instructions provided.
2. Testing and automated analysis that includes the following:
 - i. Test coverage analysis, which involves assessing the degree to which test cases cover the code and how much of the code is executed while running those test cases.
 - ii. Symbolic execution, which refers to the analysis of a program to identify the inputs that trigger each component of the program to execute.
3. Best practices review, which involves evaluating smart contracts to enhance efficiency, effectiveness, clarity, maintainability, security, and control in accordance with industry and academic practices, recommendations, and research.
4. Specific, itemized, actionable recommendations that enable you to take necessary measures to secure your smart contracts.

Tested Contract Files

This audit covered the following files listed below with a SHA-1 Hash.

A file with a different Hash has been modified, intentionally or otherwise, after the security review. A different Hash could be (but not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of this review















Scope

This section lists files that are in scope for the metrics report.

- **Project:** DogeFarm
- **Included Files:**
 - ``
- **Excluded Paths:**
 - ``
- **File Limit:** undefined
 - **Exclude File list Limit:** undefined
- **Workspace Repository:** unknown (undefined @ undefined)

Source Units in Scope

Source Units Analyzed: 6
Source Units in Scope: 6 (100%)

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
	DogeFarmToken.sol	1	<div></div>	148	144	110	5	111	
	LockReward.sol	1	<div></div>	6	6	3	1	1	
	MasterChef.sol	1	<div></div>	307	301	226	45	167	
	RealYieldClaim.sol	1	<div></div>	91	79	57	1	67	
	StakingRewards.sol	1	<div></div>	213	209	157	18	101	
	PrivateSale.sol	1	<div></div>	268	265	195	25	170	
	Totals	6	<div></div>	1033	1004	748	95	617	

Legend:

- **Lines:** total lines of the source unit
- **nLines:** normalized lines of the source unit (e.g. normalizes functions spanning multiple lines)
- **nSLOC:** normalized source lines of code (only source-code lines; no comments, no blank lines)
- **Comment Lines:** lines containing single or block comments
- **Complexity Score:** a custom complexity score derived from code statements that are known to introduce code complexity (branches, loops, calls, external interfaces, ...)

Out of Scope

Excluded Source Units

Source Units Excluded: 0

File

None

Duplicate Source Units

Duplicate Source Units Excluded: 0

File

None

Doppelganger Contracts

Doppelganger Contracts: 1

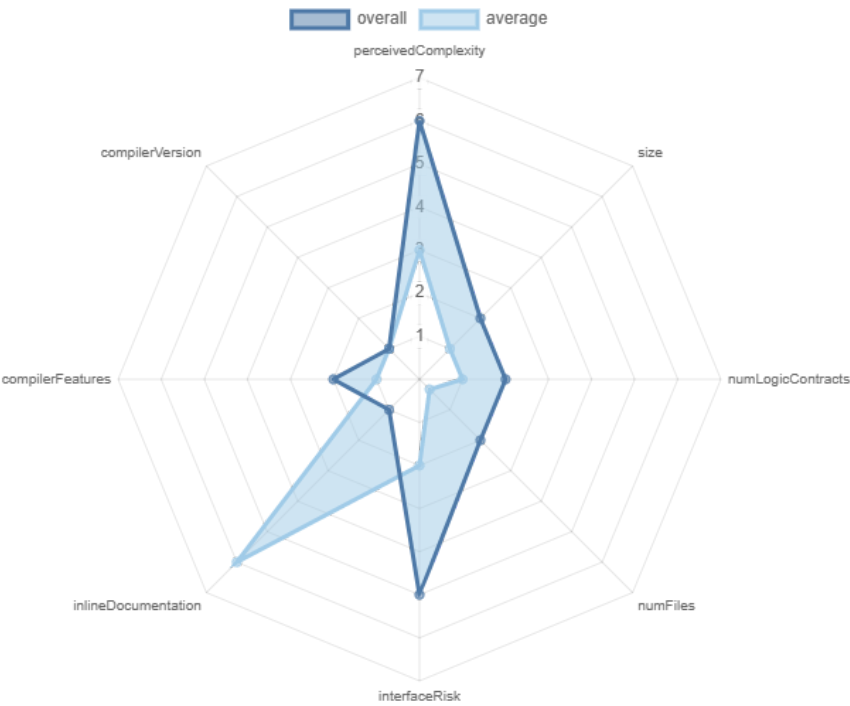
File	Contract	Doppelganger
LockReward.sol	LockReward	(fuzzy) 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45

Report

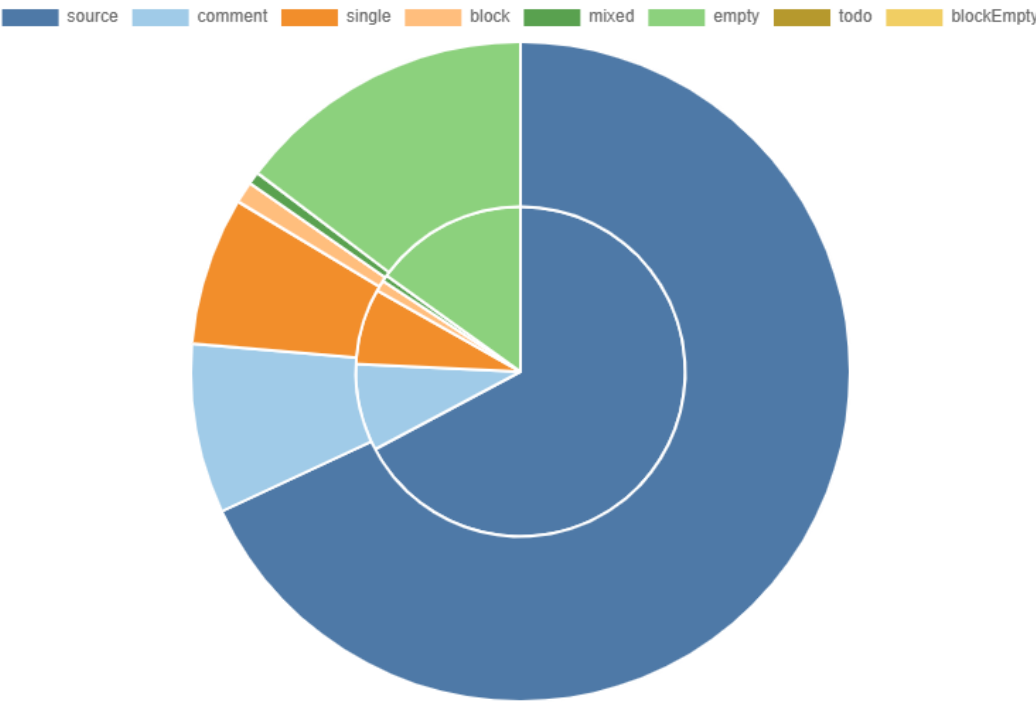
Overview

The analysis finished with 0 errors and 0 duplicate files.

Risk



Source Lines (sloc vs. nsloc)



Inline Documentation

- Comment-to-Source Ratio:** On average there are 8.18 code lines per comment (lower=better).
- ToDo's:** 0

Components

Contracts	Libraries	Interfaces	Abstract
6	0	0	0


Exposed Functions

This section lists functions that are explicitly declared public or payable. Please note that getter methods for public stateVars are not included.












 Public	 Payable
75	8

External	Internal	Private	Pure	View
52	66	2	3	8

StateVariables

Total	 Public
66	65

Capabilities

Solidity Versions observed	 Experimental Features	 Can Receive Funds	 Uses Assembly	 Has Destroyable Contracts	
<div><div>^0.8.0</div><div>^0.8</div></div>		<div>yes</div>	<div></div>	<div></div>	
 Transfers	 Low-Level Calls	 DelegateCall	 Uses Hash Functions	 ECRrecover	 New/Create/Create2
<div>yes</div>	<div></div>	<div></div>	<div>yes</div>	<div></div>	<div></div>
 TryCatch	Σ Unchecked				
<div></div>	<div></div>				

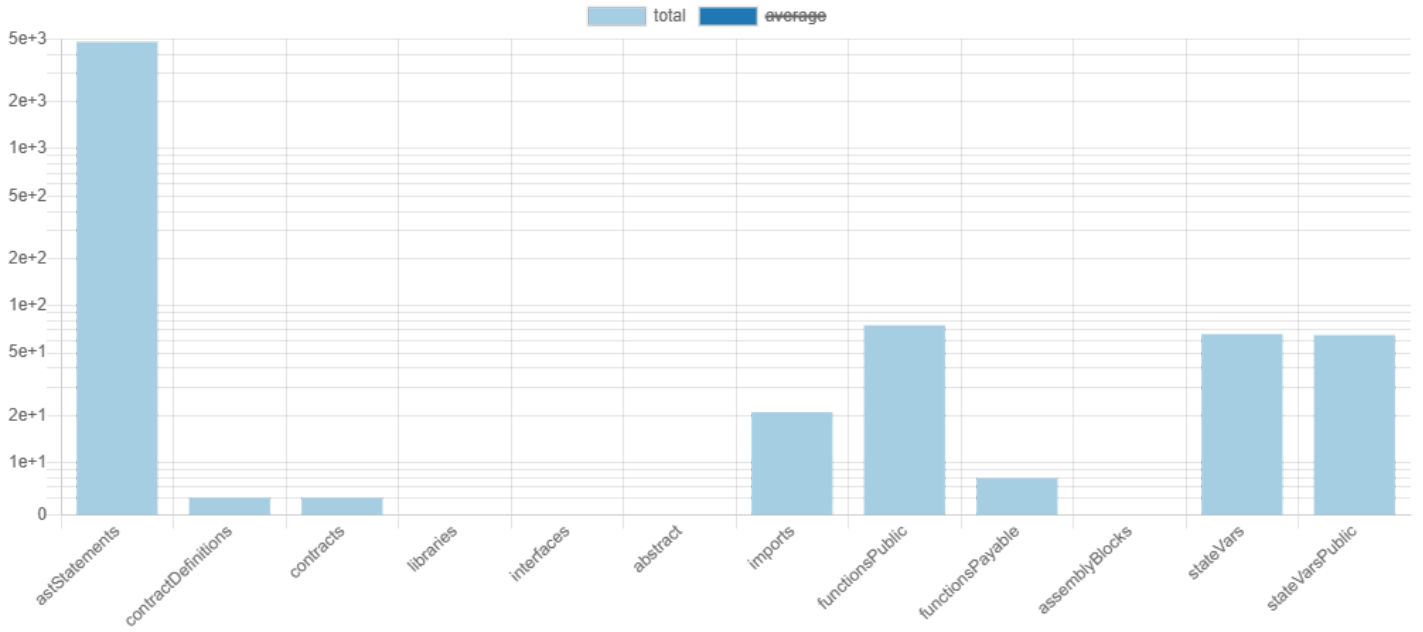
Dependencies / External Imports

Dependency / Import Path	Count
@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol	2
@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol	3
@openzeppelin/contracts/access/Ownable.sol	2
@openzeppelin/contracts/security/ReentrancyGuard.sol	1
@openzeppelin/contracts/token/ERC20/ERC20.sol	2
@openzeppelin/contracts/token/ERC20/IERC20.sol	2
@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol	1
@openzeppelin/contracts/utils/cryptography/ECDSA.sol	1
@openzeppelin/contracts/utils/math/SafeMath.sol	3

Totals

Summary

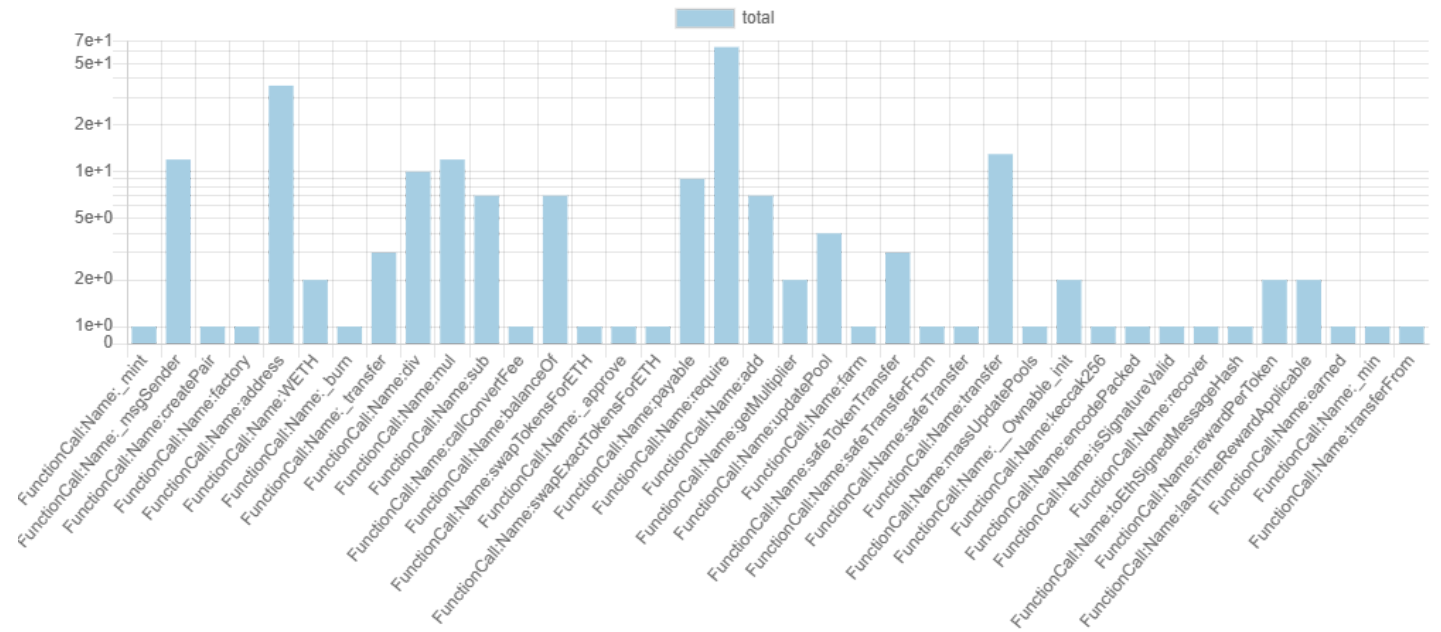
Summary



AST Node Statistics

Function Calls

Function Calls



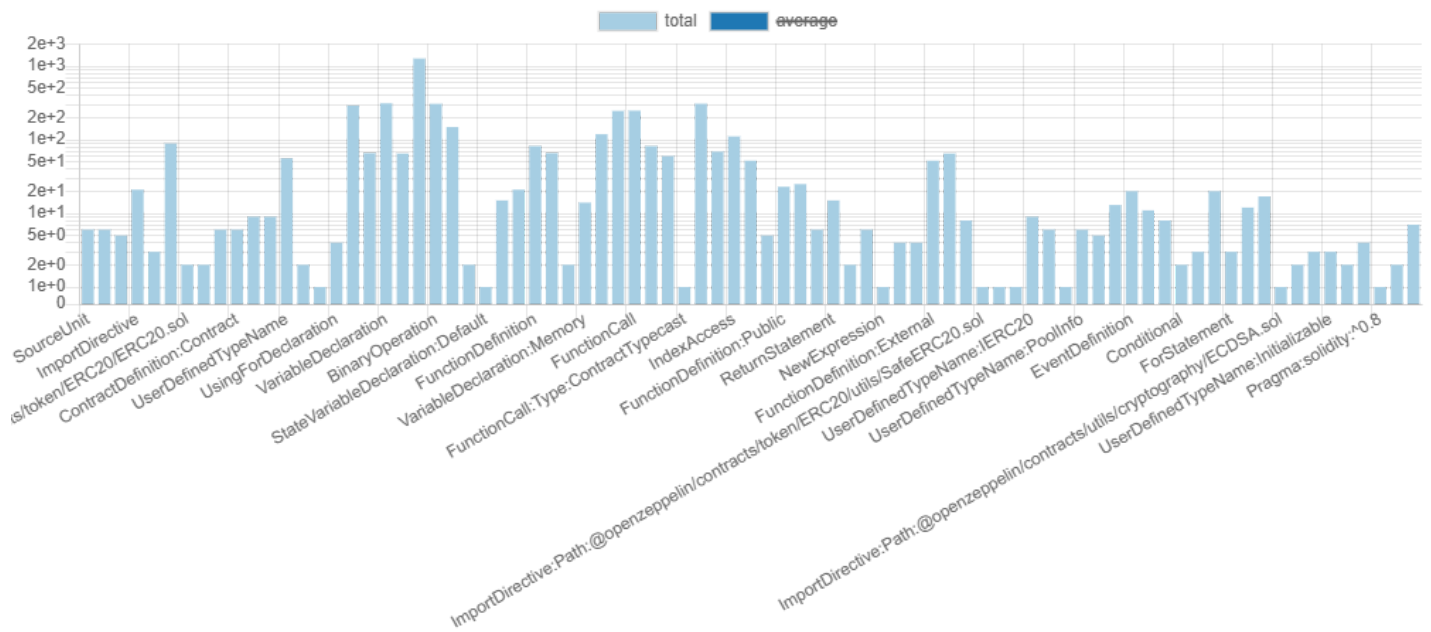
Assembly Calls

Assembly Calls



AST Total

AST Elements



Inheritance Graph

Contract Summary
























Sūrya's Description Report Files Description Table

File Name	SHA-1 Hash
DogeFarmToken.sol	2f7629caeb7125e3b6c7599471921f4b2af34622
LockReward.sol	5e2db11e8fbae96a0cc764a9281b894b041c0bba
MasterChef.sol	47a1ff664b4ba443dd979299141a20c7914b44d4
RealYieldClaim.sol	d435801a5fd3a121088aee64bdc461cba74412aa
StakingRewards.sol	b31d627b9ba9244f16588bdd361a3458007dbad3
PrivateSale.sol	03079fb1744c062ed8e8420b40524a829a4f3daf



Contracts Description Table

Contract	Type	Bases		
<div>L</div>	Function Name	Visibility	Mutability	Modifiers
DogeFarmToken	Implementation	Ownable, ERC20		
L		Public		ERC20
L	burn	Public		NO
L	_transfer	Internal		
L	callConvertFee	Internal		lockTheSwap
L	swapTokensForETH	Private		
L	setExcludeFromFee	External		NO
L	changeTreasuryWallet	External		NO
L	changeSharingPoolWallet	External		NO
L	changeNumTokensSellToAddToETH	External		onlyOwner
L		External		NO
LockReward	Implementation			
MasterChef	Implementation	Ownable, ReentrancyGuard		
L		Public		NO
L	poolLength	External		NO
L	add	External		onlyOwner nonDuplicated
L	set	External		onlyOwner
L	getMultiplier	Public		NO
L	pendingToken	External		NO
L	massUpdatePools	Public		NO
L	updatePool	Public		NO
L	deposit	Public		notPause nonReentrant
L	withdraw	Public		notPause nonReentrant
L	unlockLp	Public		notPause nonReentrant
L	unlockReward	Public		notPause nonReentrant
L	emergencyWithdraw	Public		nonReentrant

Contract	Type	Bases		
L	safeTokenTransfer	Internal 🔒	🛑	
L	updateEmissionRate	External !	🛑	onlyOwner
L	setPause	External !	🛑	onlyOwner
L	updateStartBlock	Public !	🛑	onlyOwner
L	updateLockLpTime	External !	🛑	onlyOwner
L	updateLockRewardTime	External !	🛑	onlyOwner
L	updateLockRewardPercent	External !	🛑	onlyOwner
RealYieldClaim	Implementation	Initializable, OwnableUpgradeable		
L	initialize	Public !	🛑	initializer
L	claimReward	Public !	🛑	NO !
L	setPause	Public !	🛑	onlyOwner
L	isSignatureValid	Public !		NO !
L	setServerSigner	Public !	🛑	onlyOwner
L	getClaimByld	Public !		NO !
L	withdrawRemainCore	External !	🛑	onlyOwner
L	deposit	External !	🏧	NO !
L		External !	🏧	NO !
L		External !	🏧	NO !
StakingRewards	Implementation	Initializable		
L	initialize	Public !	🛑	initializer
L	lastTimeRewardApplicable	Public !		NO !
L	rewardPerToken	Public !		NO !
L	stake	External !	🛑	updateReward
L	unstake	External !	🛑	updateReward
L	withdraw	External !	🛑	NO !
L	earned	Public !		NO !
L	getReward	External !	🛑	updateReward
L	unlockReward	External !	🛑	NO !
L	compoundReward	External !	🛑	updateReward
L	setRewardsDuration	External !	🛑	onlyOwner
L	setTimeLockToken	External !	🛑	onlyOwner
L	notifyRewardAmount	External !	🛑	onlyOwner updateReward
L	_min	Private 🔒		
L	setCanWithdraw	External !	🛑	onlyOwner
L	setLockRewardTime	External !	🛑	onlyOwner
L	setLockRewardPercent	External !	🛑	onlyOwner

Contract	Type	Bases		
PrivateSale	Implementation	Initializable, OwnableUpgradeable		
L	initialize	Public !		initializer
L	changeStatus	External !		onlyOwner
L	setStatus	External !		onlyOwner
L	setWhitelistTime	External !		onlyOwner
L	setPublicTime	External !		onlyOwner
L	setPublicEndTime	External !		onlyOwner
L	setWhitelistEndTime	External !		onlyOwner
L	setClaimTime	External !		onlyOwner
L	setRate	External !		onlyOwner
L	setMinMaxCommit	External !		onlyOwner
L	setReceiveAddress	External !		onlyOwner
L	setWhiteListSameAmount	External !		onlyOwner
L	setWhiteListAmount	External !		onlyOwner
L	withdrawRemainToken	External !		onlyOwner
L	withdrawRemainCore	External !		onlyOwner
L	joinWhitelist	External !		NO !
L	joinSubscription	External !		NO !
L	claimWhitelistToken	External !		NO !
L	claimSubscription	External !		NO !
L	refundWhitelist	External !		NO !
L	refundPublic	External !		NO !
L	getTimes	Public !		NO !
L	getMinMaxCommit	Public !		NO !
L		External !		NO !
L		External !		NO !

Legend

Symbol	Meaning
	Function can modify state
	Function is payable

Detectors Issue

Description	Check	Impact	Confidence
Pragma version^0.8.0 (contracts/LockReward.sol#2) allows old versions	solc-version	Informational	High

Thinking about smart contract security? We can provide training, ongoing advice, and smart contract auditing. [Contact us](#).



TECHRIGHT

The Best Smart Contract Safeguard



TechRight



@techright