SOLIDProof
Bring trust into your projects

Blockchain Security | Smart Contract Audits | KYC Development | Marketing

MADE IN GERMANY

NeoKingdom

Audit

Security Assessment

08. May, 2023

For

NEO KINGDOM

SolidProof_io

@solidproof_io

Disclaimer

SolidProof.io reports are not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team. SolidProof.io do not cover testing or auditing the integration with external contract or services (such as Unicrypt, Uniswap, PancakeSwap etc'...)

SolidProof.io Audits do not provide any warranty or guarantee regarding the absolute bug- free nature of the technology analyzed, nor do they provide any indication of the technology proprietors. SolidProof Audits should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.

SolidProof.io Reports represent an extensive auditing 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. SolidProof's position is that each company and individual are responsible for their own due diligence and continuous security. SolidProof in no way claims any guarantee of security or functionality of the technology we agree to analyze.

Version	Date	Description
1.0	21. April 2023 - 26. Apr 2023	• Layout project • Automated- /Manual-Security Testing • Summary

Network
EVMOS

Website
https://neokingdom.org/

Twitter
https://twitter.com/NEOKingdomDAO

Description

NEOKingdom DAO aims to revolutionize the traditional company structure by developing a new type of organization that is truly owned by its workers. By leveraging blockchain technology, we are building a company that values and rewards the efforts of those who contribute to its growth and success, fostering a sense of purpose and belonging.

Project Engagement

During the Date of 19 April 2023, **NEOKingdom Team** engaged Solidproof.io to audit smart contracts that they created. The engagement was technical in nature and focused on identifying security flaws in the design and implementation of the contracts. They provided Solidproof.io with access to their code repository and whitepaper.

Logo

Contract Link

v1.0

- https://github.com/NeokingdomDAO/contracts
- Commit: a17ef49

Deployed Contracts

NeokingdomToken (NEOK):
0x655ecB57432CC1370f65e5dc2309588b71b473A9

ShareholderRegistry (NEOS):
0x4706eD7a10064801F260BBf94743f241FCEf815e

GovernanceToken: 0x05d1b235572190315276 8F0ec1B105Be1c35BCb4

Voting: 0x5DC219C8CaeF7c9ECd0b97372e6Ef4fC5D827975

RedemptionController:
0x7045bfaB66B55074C56aBeE34308CDa0916e086C

InternalMarket: 0x7687155fB855e24d1416C288CbaC0AFC3B65353c

ResolutionManager: 0xE5714C29b7acE2C6a3A80BE511ED7e5b92594204

DAORoles: 0x6A176C92985430535E738A79749A4137BEC6C4Db

Note - This Audit report consists of security analysis of the NeoKingdom smart contracts, functional testing (or unit testing) of the contract's logic was not included in this analysis.

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 aspossible.
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 have informational character but is not effecting any of the code.	An observation that does not determine a level of risk

Auditing Strategy and Techniques Applied

Throughout the review process, care was taken to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices. To do so, reviewed line-by-line by our team of expert pentesters and smart contract developers, documenting any issues as there were discovered.

Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:
    i) Review of the specifications, sources, and instructions provided to SolidProof to make sure we understand the size, scope, and functionality of the smart contract.
    ii) Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
    iii) Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to SolidProof describe.

2. Testing and automated analysis that includes the following:
    i) Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
    ii) Symbolic execution, which is analysing a program to determine what inputs causes each part of a program to execute.

3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.

4. Specific, itemized, actionable recommendations to help you take steps 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.

v1.0

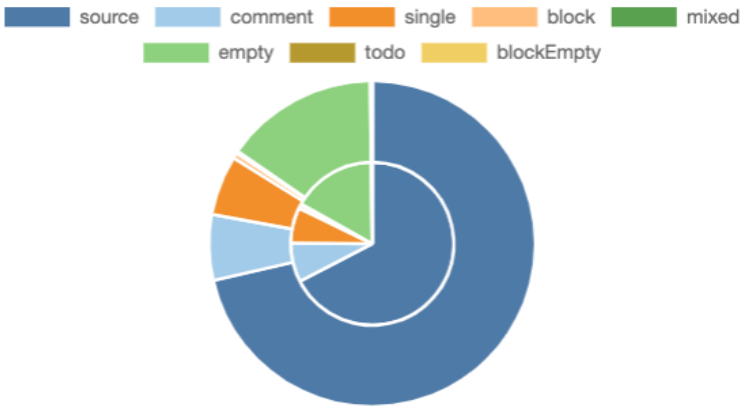
File Name	SHA-1 Hash
contracts/NeokingdomToken/NeokingdomToken.sol	f4396532ddef0c8c9a3a237930ec9e0cbf6ab249
contracts/NeokingdomToken/INeokingdomToken.sol	a346ec60856918bc28d1ba7d447943e41f731891
contracts/ResolutionManager/ResolutionManager.sol	a49d5ccd46194a03c90b5efd7ffe8481c6b04516
contracts/ResolutionManager/ResolutionManagerBase.sol	31d929df13ad7c5ab2e6a692b762568e9b586dbb
contracts/ShareholderRegistry/ShareholderRegistry.sol	b82a3e69f4acca3089aee8d96a70cc22f22323a4
contracts/ShareholderRegistry/ShareholderRegistrySnapshot.sol	431003678852b4f2d29f590ad62e2c76607e2f41
contracts/ShareholderRegistry/IShareholderRegistry.sol	b4a69028de2cf28191f852c127b10a8d87595e72
contracts/ShareholderRegistry/ShareholderRegistryBase.sol	a56b81cc8b5afb01216f56b5c33271c2a05b6523
contracts/extensions/HasRole.sol	5cbd77b97f5f389ddf8f7eb8aa1551108bf76263
contracts/extensions/DAORoles.sol	9884ea990cb0b2038932559c15f9406d66f744ea
contracts/extensions/ISnapshot.sol	b1b3d7d21b924659bb7d7482f51cb15600e9e264
contracts/extensions/Snapshottable.sol	987b3fafbcd48712b0d0836fb24546e0913c6f2b

contracts/extensions/Roles.sol	d8aab5832599c156dc852b3b32eb27ac7eb3de89
contracts/PriceOracle/IStdReference.sol	9eb4c4ffb0c34baa23e1fdcd1a674aece62ec434
contracts/PriceOracle/PriceOracle.sol	e20cc3fca0175888ae8798d587e68c772dbda15f
contracts/Voting/VotingBase.sol	8b794fa2164db6bbc05e3b5c64b405f380e5171e
contracts/Voting/IVoting.sol	6d28e8612b72a864a8c22dbdde3f5873118bcc44
contracts/Voting/Voting.sol	021f553495ff98c37aac8214b42650a9e6253ea0
contracts/Voting/VotingSnapshot.sol	6ca673fd4235554715ea63a614049fe16a495b4d
contracts/GovernanceToken/GovernanceTokenSnapshot.sol	7a33aba94a5fc09cc272deab3b8ee7f9ae3a6772
contracts/GovernanceToken/GovernanceToken.sol	273f9f1f1c57f7336bcc21f209f6d828bd216c37
contracts/GovernanceToken/GovernanceTokenBase.sol	446963825fc6bf2bf0d4da504ef3567e9fa8b273
contracts/GovernanceToken/IGovernanceToken.sol	2ccab74266c76aaa8626ffd554e30ca36a65d710
contracts/RedemptionController/RedemptionControllerBase.sol	a34c65d8479d331e75fd1ab879feffe38b073db7
contracts/RedemptionController/IRedemptionController.sol	94b4d3a38128b0b5af9ab6957ad837dfe982432f
contracts/RedemptionController/RedemptionController.sol	98b09346171508ff2d6d8d19be8b010b510a48ba
contracts/InternalMarket/InternalMarket.sol	c092e16512dcb05bc5273d287c2459e0b1a98ee6

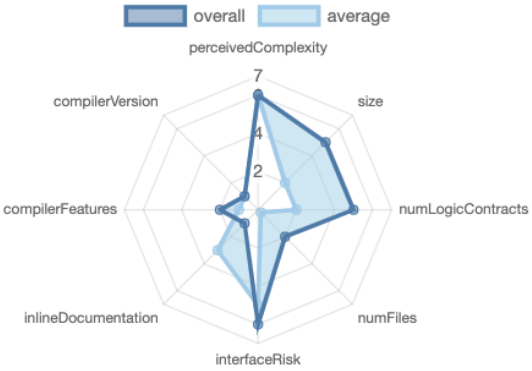
contracts/InternalMarket/InternalMarketBase.sol	c09f1df2e3abd2a940dff3eb8bddeda1d00c3ead

Metrics

Source Lines
v1.0



Risk Level
v1.0



Capabilities

Components

📝Contracts	📚Libraries	🔍Interfaces	🎨Abstract
11	1	7	9

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
141	0

External	Internal	Private	Pure	View
73	261	0	1	60

StateVariables

Total	🌐Public
59	27

Capabilities

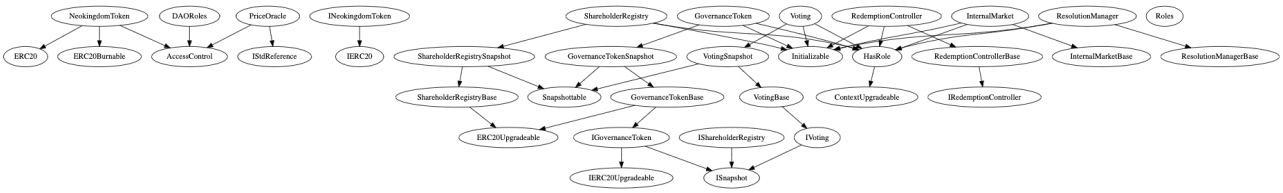
Solidity Versions observed	🧪 Experimental Features	💰 Can Receive Funds	🖥️ Uses Assembly	💣 Has Destroyable Contracts

`^0.8.9`
`^0.8.16`
`^0.8.0`
		—	—	—

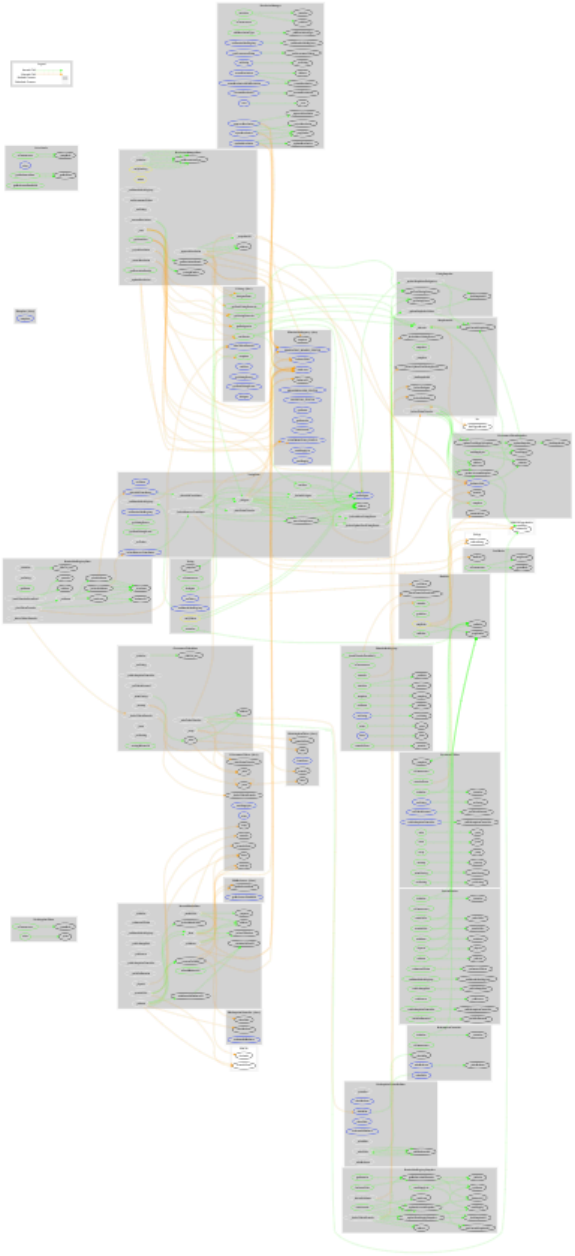
📩 Transfers ETH	⚡ Low-Level Calls	👥 DelegateCall	🧮 Uses Hash Functions	🔏 ECRecover	🌀 New/Create/Create2
`yes`	—	—	`yes`	—	—

♻️ TryCatch	Σ Unchecked
—	`yes`

Inheritance Graph
v1.0



CallGraph
v1.0



Scope of Work/Verify Claims

The above token Team provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract (usual the same name as team appended with .sol).

We will verify the following claims:

1. Is contract an upgradeable
2. Correct implementation of Token standard
3. Overall checkup (Smart Contract Security)

Is contract an upgradeable

Name	
Is contract an upgradeable?	Yes

Comments:

v1.0

- Owner/DAO can deploy a new version contracts which can change any limit and give owner new privileges
    - Be aware of this and do your own research for the contract which is the contract pointing to

Correct implementation of Token standard

ERC20				
Function	**Description**	**Exist**	**Tested**	**Verified**
TotalSupply	Provides information about the total token supply	✓	✓	✓
BalanceOf	Provides account balance of the owner's account	✓	✓	✓
Transfer	Executes transfers of a specified number of tokens to a specified address	✓	✓	✓
TransferFrom	Executes transfers of a specified number of tokens from a specified address	✓	✓	✓
Approve	Allow a spender to withdraw a set number of tokens from a specified account	✓	✓	✓
Allowance	Returns a set number of tokens from a spender to the owner	✓	✓	✓

Overall checkup (Smart Contract Security)

Tested	Verified
✓	✓

Legend

Attribute	Symbol
Verified / Checked	✓
Partly Verified	⚑
Unverified / Not checked	✗
Not available	–

Modifiers and public functions
v1.0

GovernanceToken

- snapshot
  - onlyRole
- setVoting
  - onlyRole
- setTokenExternal
  - onlyRole
- setRedemptionController
  - onlyRole
- mint
  - onlyRole
- burn
  - onlyRole
- wrap
  - onlyRole
- unwrap
  - onlyRole
- mintVesting
  - onlyRole
- setVesting
  - onlyRole
- transfer
  - onlyRole
- transferFrom
  - onlyRole

InternalMarket

- makeOffer
- matchOffer
- withdraw
- deposit
- redeem
- setInternalToken
  - onlyRole
- setShareholderRegistry
  - onlyRole
- setExchangePair
  - onlyRole
- setReserve
  - onlyRole
- setRedemptionController
  - onlyRole
- setOfferDuration
  - onlyRole

ShareholderRegistry

- snapshot
  - onlyRole
- setStatus
  - onlyRole
- setVoting
  - onlyRole
- mint
  - onlyRole
- burn
  - onlyRole
- batchTransferFromDAO
  - onlyRole
- transferFrom
  - onlyRole
- transfer
  - onlyRole

Voting.sol

```
setToken
onlyRole
setShareholderRegistry
onlyRole
snapshot
onlyRole
afterTokenTransfer
onlyToken
beforeRemoveContributor
onlyRole
afterAddContributor
onlyRole
delegate
delegateFrom
onlyRole
```

Ownership/Authorized Privileges

- *GovernanceToken.sol*
  - OPERATOR_ROLE
    - Set voting contract, external token contract, Redemption controller address
    - Mint external token contract tokens
    - Burn external token contract tokens from any arbitrary wallet
    - Set vesting for an account
  - MARKET_ROLE
    - Transfer tokens

- *InternalMarket.sol*
  - RESOLUTION_ROLE
    - Set internal token contract
    - Set shareholder registry address's and exchange pair address
    - Set Reserve address, redemption controller address
    - Set offer duration to any arbitrary address

- *NeokingdomToken.sol*
  - MINTER_ROLE address can mint unlimited tokens

- *RedemptionController.sol*
  - TOKEN_MANAGER_ROLE
    - Set after mint, offer, and redeem values for wallets/addresses

- *ResolutionManager.sol*
  - OPERATOR_ROLE
    - Set Voting address
    - Set governance token address
    - Set shareholderRegistry address
  - RESOLUTION_ROLE
    - Add a resolution type

- *ShareholderRegistry.sol*
  - RESOLUTION_ROLE
    - Transfer tokens from DAO
    - Transfer tokens in general
    - Mint tokens
    - Burn tokens from any address
    - Set voting address

- *Voting.sol*
  - RESOLUTION_ROLE
    - Delegate another address for voting on behalf of a delegator
  - SHAREHOLDER_REGISTRY_ROLE
    - Add/Remove contributors
  - OPERATOR_ROLE
    - Set token address
    - Set shareholder registry address
  - Only Token address set by operator can call the "afterTokenTransfer" function

- There are several authorities which are authorized to call some functions, that means, if the owner is renounced, another address is still authorized to call functions
  - Be aware of this

Note for Development Team - The Token internal address, ShareholderRegistry, address and Reserve address lacks a zero address validation check as mentioned in the issues table on Page23. We strongly advise to fix this otherwise if these addresses are set to zero then the Transfers will be reverted.

Please check if an OnlyOwner or similar restrictive modifier has been forgotten.

Source Units in Scope
v1.0

File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
contracts/NeokingdomToken/NeokingdomToken.sol	1	—	18	18	13	1	16	🧱
contracts/NeokingdomToken/INeokingdomToken.sol	—	1	13	8	4	1	9	—
contracts/ResolutionManager/ResolutionManager.sol	1	—	149	113	96	2	63	—
contracts/ResolutionManager/ResolutionManagerBase.sol	1	—	514	468	367	22	111	—
contracts/ShareholderRegistry/ShareholderRegistry.sol	1	—	91	61	42	6	45	🕹️
contracts/ShareholderRegistry/ShareholderRegistrySnapshot.sol	1	—	149	126	94	13	55	—
contracts/ShareholderRegistry/IShareholderRegistry.sol	—	1	44	8	4	1	27	—
contracts/ShareholderRegistry/ShareholderRegistryBase.sol	1	—	143	115	90	4	46	🧱🔀
contracts/extensions/HasRole.sol	1	—	41	41	33	1	19	—
contracts/extensions/DAORoles.sol	1	—	11	11	7	1	5	—
contracts/extensions/ISnapshot.sol	—	1	7	6	3	1	3	—
contracts/extensions/Snapshottable.sol	1	—	53	46	35	1	13	—
contracts/extensions/Roles.sol	1	—	15	15	12	1	22	🧱
contracts/PriceOracle/IStdReference.sol	—	1	23	13	8	7	5	—
contracts/PriceOracle/PriceOracle.sol	1	—	72	60	49	5	39	🧱
contracts/Voting/VotingBase.sol	1	—	216	189	133	21	74	—
contracts/Voting/IVoting.sol	—	1	48	8	4	1	29	—
contracts/Voting/Voting.sol	1	—	102	81	45	20	46	—
contracts/Voting/VotingSnapshot.sol	1	—	129	106	79	7	50	—
contracts/GovernanceToken/GovernanceTokenSnapshot.sol	1	—	113	93	64	12	41	—
contracts/GovernanceToken/GovernanceToken.sol	1	—	122	68	50	2	61	🕹️
contracts/GovernanceToken/GovernanceTokenBase.sol	1	—	114	99	72	7	35	🕹️
contracts/GovernanceToken/IGovernanceToken.sol	—	1	23	9	5	1	17	—
contracts/RedemptionController/RedemptionControllerBase.sol	1	—	237	222	162	29	58	—
contracts/RedemptionController/IRedemptionController.sol	—	1	18	9	5	1	9	—
contracts/RedemptionController/RedemptionController.sol	1	—	58	49	26	14	25	—
contracts/InternalMarket/InternalMarket.sol	1	—	82	66	49	2	57	—
contracts/InternalMarket/InternalMarketBase.sol	1	—	290	259	191	19	79	🕹️
Totals	**21**	**7**	**2895**	**2367**	**1742**	**203**	**1059**	🕹️🧱🔀

Legend

Attribute	Description
Lines	total lines of the source unit
nLines	normalised lines of the source unit (e.g. normalises functions spanning multiple lines)
nSLOC	normalised 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, ...)

Audit Results

Critical issues

No critical issues

High issues

No high issues

Medium issues

No medium issues

Low issues

Issue	File	Type	Line	Description
#1	GovernanceToken.sol	Missing Zero Address Validation (missing-zero-check)	35-47	Check that the address is not zero
#2	InternalMarket.sol	Missing Zero Address Validation (missing-zero-check)	46-71	Check that the address is not zero
#3	ResolutionManager.sol	Missing Zero Address Validation (missing-zero-check)	41-53	Check that the address is not zero
#4	ResolutionManager.sol	Missing Events Arithmetic	41-53	Emit an event for critical parameter changes. The contract has no events
#5	GovernanceToken.sol	Missing Events Arithmetic	35-47	Emit an event for critical parameter changes. The contract has no events
#6	Voting.sol	Missing Zero Address Validation (missing-zero-check)	30, 36	Check that the address is not zero

Informational issues

Issue	File	Type	Line	Description

#1	ReolutionManagerBase.sol	Uninitialised local variable	217	Ensure that all the local variables are initialised
#2	All	NatSpec documentation missing	—	If you started to comment your code, also comment all other functions, variables etc.

Audit Comments

We recommend you to use the special form of comments (NatSpec Format, Follow link for more information https://docs.soliditylang.org/en/latest/natspec-format.html) for your contracts to provide rich documentation for functions, return variables and more. This helps investors to make clear what that variables, functions etc. do.

Alleviation

Comments from the NeoKingdom team over the ownership of the contracts -

"All that is currently operated by multisig (and related ACL), will be offloaded to the DAO one day (meaning: only a resolution can execute it)
- isDAO will be the role to enable inwards dependencies -> functions that can only be called after a resolution is approved
this role can be delegated to the operators or to a multisig temporarily
- the rest of the roles will be the so called "interface dependencies" roles and they will be used to enable the cross communication between inter-dependent smart contracts"

08. May 2023:

- We recommend Neokingdom team to conduct unit and fuzz tests thoroughly to rule out possibilities of an unwanted logical and calculation errors.
- There is still an owner (Owner still has not renounced ownership)
- Owner can deploy a new version of the contract which can change any limit and give owner new privileges
- Read whole report and modifiers section for more information

SWC Attacks

ID	Title	Relationships	Status
SWC-136	Unencrypted Private Data On-Chain	CWE-767: Access to Critical Private Variable via Public Method	PASSED
SWC-135	Code With No Effects	CWE-1164: Irrelevant Code	PASSED
SWC-134	Message call with hardcoded gas amount	CWE-655: Improper Initialization	PASSED
SWC-133	Hash Collisions With Multiple Variable Length Arguments	CWE-294: Authentication Bypass by Capture-replay	PASSED
SWC-132	Unexpected Ether balance	CWE-667: Improper Locking	PASSED
SWC-131	Presence of unused variables	CWE-1164: Irrelevant Code	PASSED
SWC-130	Right-To-Left-Override control character (U+202E)	CWE-451: User Interface (UI) Misrepresentation of Critical Information	PASSED
SWC-129	Typographical Error	CWE-480: Use of Incorrect Operator	PASSED
SWC-128	DoS With Block Gas Limit	CWE-400: Uncontrolled Resource Consumption	PASSED

SWC-127	Arbitrary Jump with Function Type Variable	CWE-695: Use of Low-Level Functionality	PASSED
SWC-125	Incorrect Inheritance Order	CWE-696: Incorrect Behavior Order	PASSED
SWC-124	Write to Arbitrary Storage Location	CWE-123: Write-what-where Condition	PASSED
SWC-123	Requirement Violation	CWE-573: Improper Following of Specification by Caller	PASSED
SWC-122	Lack of Proper Signature Verification	CWE-345: Insufficient Verification of Data Authenticity	PASSED
SWC-121	Missing Protection against Signature Replay Attacks	CWE-347: Improper Verification of Cryptographic Signature	PASSED
SWC-120	Weak Sources of Randomness from Chain Attributes	CWE-330: Use of Insufficiently Random Values	PASSED
SWC-119	Shadowing State Variables	CWE-710: Improper Adherence to Coding Standards	PASSED
SWC-118	Incorrect Constructor Name	CWE-665: Improper Initialization	PASSED
SWC-117	Signature Malleability	CWE-347: Improper Verification of Cryptographic Signature	PASSED

SWC-116	Timestamp Dependence	CWE-829: Inclusion of Functionality from Untrusted Control Sphere	**PASSED**
SWC-115	Authorization through tx.origin	CWE-477: Use of Obsolete Function	**PASSED**
SWC-114	Transaction Order Dependence	CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')	**PASSED**
SWC-113	DoS with Failed Call	CWE-703: Improper Check or Handling of Exceptional Conditions	**PASSED**
SWC-112	Delegatecall to Untrusted Callee	CWE-829: Inclusion of Functionality from Untrusted Control Sphere	**PASSED**
SWC-111	Use of Deprecated Solidity Functions	CWE-477: Use of Obsolete Function	**PASSED**
SWC-110	Assert Violation	CWE-670: Always-Incorrect Control Flow Implementation	**PASSED**
SWC-109	Uninitialized Storage Pointer	CWE-824: Access of Uninitialized Pointer	**PASSED**
SWC-108	State Variable Default Visibility	CWE-710: Improper Adherence to Coding Standards	**PASSED**
SWC-107	Reentrancy	CWE-841: Improper Enforcement of Behavioral Workflow	**PASSED**
SWC-106	Unprotected SELFDESTRUCT Instruction	CWE-284: Improper Access Control	**PASSED**

SWC-105	Unprotected Ether Withdrawal	CWE-284: Improper Access Control	PASSED
SWC-104	Unchecked Call Return Value	CWE-252: Unchecked Return Value	PASSED
SWC-103	Floating Pragma	CWE-664: Improper Control of a Resource Through its Lifetime	NOT PASSED
SWC-102	Outdated Compiler Version	CWE-937: Using Components with Known Vulnerabilities	PASSED
SWC-101	Integer Overflow and Underflow	CWE-682: Incorrect Calculation	PASSED
SWC-100	Function Default Visibility	CWE-710: Improper Adherence to Coding Standards	PASSED

SolidProof_io

@solidproof_io

SolidProofed

Blockchain Security | Smart Contract Audits | KYC Development | Marketing

MADE IN GERMANY