



**SOLID**Proof  
*Bring trust into your projects*

**Blockchain Security | Smart Contract Audits | KYC**

MADE IN GERMANY

# CleverAgent Finance

# Audit

**Security Assessment**

05.August,2022

**For**



[SolidProof.io](https://solidproof.io)



[@solidproof\\_io](https://t.me/solidproof_io)

Disclaimer	2
Description	5
Project Engagement	5
Logo	5
Contract Link	5
Methodology	7
Used Code from other Frameworks/Smart Contracts (direct imports)	8
Tested Contract Files	10
Source Lines	11
Risk Level	11
Capabilities	12
Inheritance Graph	13
CallGraph	14
Scope of Work/Verify Claims	15
Modifiers and public functions	22
Source Units in Scope	23
Critical issues	24
High issues	24
Medium issues	24
Low issues	25
Informational issues	26
Audit Comments	26
SWC Attacks	27

## 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	01.August,2022	<ul style="list-style-type: none"><li>• Layout project</li><li>• Automated- /Manual-Security Testing</li><li>• Summary</li></ul>

## Network

### Website

<https://cleveragent.finance>

### Twitter

<https://twitter.com/Ceveragent>

### Discord

<https://discord.gg/pHAVYf4H>

### Telegram

<https://t.me/cleveragent>

### YouTube

<https://www.youtube.com/channel/UCL86G-Mlwi8LiB8Aj3vBE1g>

### Medium

<https://medium.com/@cleveragentfinance>

## Description

Clever Agent is a saving protocol that's based on the Binance Smart Chain network which offer investors huge opportunities to earn up to 15% interest rate per year on the deposited stablecoins (USDT, USDC, BUSD or DAI) with instant paid out.

Clever Agent also offers an additional APY in the form of the lottery for lucky depositors besides the constant APY. The more and earlier your stablecoins are deposited, the higher APY and the more chance you will win the lottery.

## Project Engagement

During the 1<sup>st</sup> of August 2022, **Clever Agent** team engaged Solidproof.io to audit the smart contracts that they created. The engagement was technical in nature and focused on identifying the security flaws in the design and implementation of the contracts. They provided Solidproof.io with access to their code repository and whitepaper.

## Logo



## Contract Links

v1.0

<https://github.com/cleveragentfinance/contract/tree/main/contracts>

Commit: 78a531b8191ff1dc8efc88b6b75442c4c7db433a

# 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)
<b>Critical</b>	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.
<b>High</b>	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.
<b>Medium</b>	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.
<b>Low</b>	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.
<b>Informational</b>	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 analyzing 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.

# Used Code from other Frameworks/Smart Contracts (direct imports)

## Imported packages:

### AgentManager

```
@openzeppelin/contracts/token/ERC20/IERC20.sol
@openzeppelin/contracts/access/Ownable.sol
@openzeppelin/contracts/utils/math/SafeMath.sol
@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router01.sol
@openzeppelin/contracts/proxy/transparent/TransparentUpgradeableProxy.sol
./interfaces/IAgent.sol
```

### AutoFarm

```
@openzeppelin/contracts/utils/Address.sol
@openzeppelin/contracts/token/ERC20/ERC20.sol
@openzeppelin/contracts/token/ERC20/IERC20.sol
@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol
@openzeppelin/contracts/access/Ownable.sol
@openzeppelin/contracts/utils/math/SafeMath.sol
@openzeppelin/contracts/security/ReentrancyGuard.sol
@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router02.sol
```

### CALottery

```
@openzeppelin/contracts/access/Ownable.sol
@openzeppelin/contracts/security/ReentrancyGuard.sol
@openzeppelin/contracts/token/ERC20/IERC20.sol
@openzeppelin/contracts/utils/Address.sol
@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol
./interfaces/IRandomNumberGenerator.sol
./interfaces/ICALotteryOld.sol
```

### Free

```
@openzeppelin/contracts/access/Ownable.sol
@openzeppelin/contracts/utils/math/SafeMath.sol
@openzeppelin/contracts/token/ERC20/ERC20.sol
```

### InsuraceAgent

```
@openzeppelin/contracts/utils/Address.sol
@openzeppelin/contracts/token/ERC20/ERC20.sol
@openzeppelin/contracts/token/ERC20/IERC20.sol
@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol
@openzeppelin/contracts/access/Ownable.sol
@openzeppelin/contracts/utils/math/SafeMath.sol
@openzeppelin/contracts/security/ReentrancyGuard.sol
@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router02.sol
```



## Liquidity

- 📄 @openzeppelin/contracts/utils/Address.sol
- 📄 @openzeppelin/contracts/token/ERC20/ERC20.sol
- 📄 @openzeppelin/contracts/token/ERC20/IERC20.sol
- 📄 @openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol
- 📄 @openzeppelin/contracts/access/Ownable.sol
- 📄 @openzeppelin/contracts/utils/math/SafeMath.sol
- 📄 @openzeppelin/contracts/security/ReentrancyGuard.sol
- 📄 @uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router02.sol

## RandomNumberGenerator

- 📄 @openzeppelin/contracts/access/Ownable.sol
- 📄 @openzeppelin/contracts/token/ERC20/IERC20.sol
- 📄 @openzeppelin/contracts/utils/Address.sol
- 📄 @openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol
- 📄 @chainlink/contracts/src/v0.8/VRFConsumerBase.sol
- 📄 ./interfaces/IRandomNumberGenerator.sol
- 📄 ./interfaces/ICALottery.sol

## Tested Contract Files

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

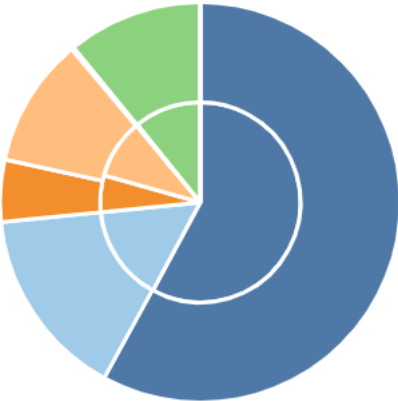
A file with a different Hash has been modified, intentionally or otherwise, after the security review. A different Hash could be (but not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of this review.

v1.0

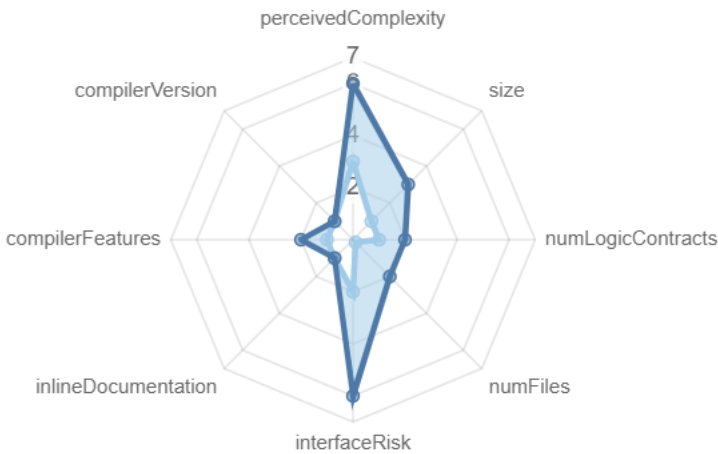
File Name	SHA-1 Hash
contracts/interfaces/IBaseToken.sol	8ebd8b109805dedcae6dff58209f72776bba1d5d
contracts/interfaces/ICombinableTokenBasis.sol	9e40efecac4b64255366bdd486551e4aa2995dfd
contracts/interfaces/IWyvernProxyRegistry.sol	93c348e3ad04401d57478c53068c13dc362950cf
contracts/interfaces/ICombinationToken.sol	fe527363905221e62c375071cfd442b2ce32fcc6
contracts/interfaces/IBasis.sol	4f777c7d3d95b2aec7989974817eb96b9ab80fc7
contracts/interfaces/IMembershipToken.sol	f616710134186788ea4957ebfb9b7893664f7292
contracts/interfaces/IWithdrawable.sol	7d652291b8c74088f39f4152a8415acaeb40be01
contracts/S1CombinationToken.sol	6aeed66c53e46c79f83e7048feb28c5aed713b9a
contracts/BaseToken.sol	26d911a9f5de5e9f5a3803c60d1bc65afb92f2eb
contracts/MembershipToken.sol	811726278cb376c083882b0790370d3bb475897a
contracts/library/Basis.sol	39bda8e1e605df111e5d657fd999bc89a552d4da
contracts/library/CombinableTokenBasis.sol	67b32b0de0622cc133e14375b1efc940b59bd07e
contracts/library/ERC721Buyable.sol	7ca4bc8d45283c78c6c7c3c0e316f6de9890552c
contracts/library/Withdrawable.sol	bf07c707d366035200497d49aeffe9026dfb4dc2
contracts/opensea/ERC721Tradable.sol	2ca444463869f36530c3c67a295d97625731ea6d

# Metrics

## Source Lines v1.0



## Risk Level v1.0



## Capabilities

### Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	9	1	13	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.*

Version	Public	Payable
1.0	238	10

Version	External	Internal	Private	Pure	View
1.0	157	144	0	36	93

### State Variables

Version	Total	Public
1.0	80	74

## Capabilities

Version	Solidity Versions observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	<div>^0.8.0</div> <div>^0.8.4</div>		Yes		

Version	Transfers ETH	Low-Level Calls	DelegateCall	Uses Hash Functions	EC Recover	New/Create/Create2
1.0	Yes			Yes		

## Inheritance Graph v1.0



# Call Graph

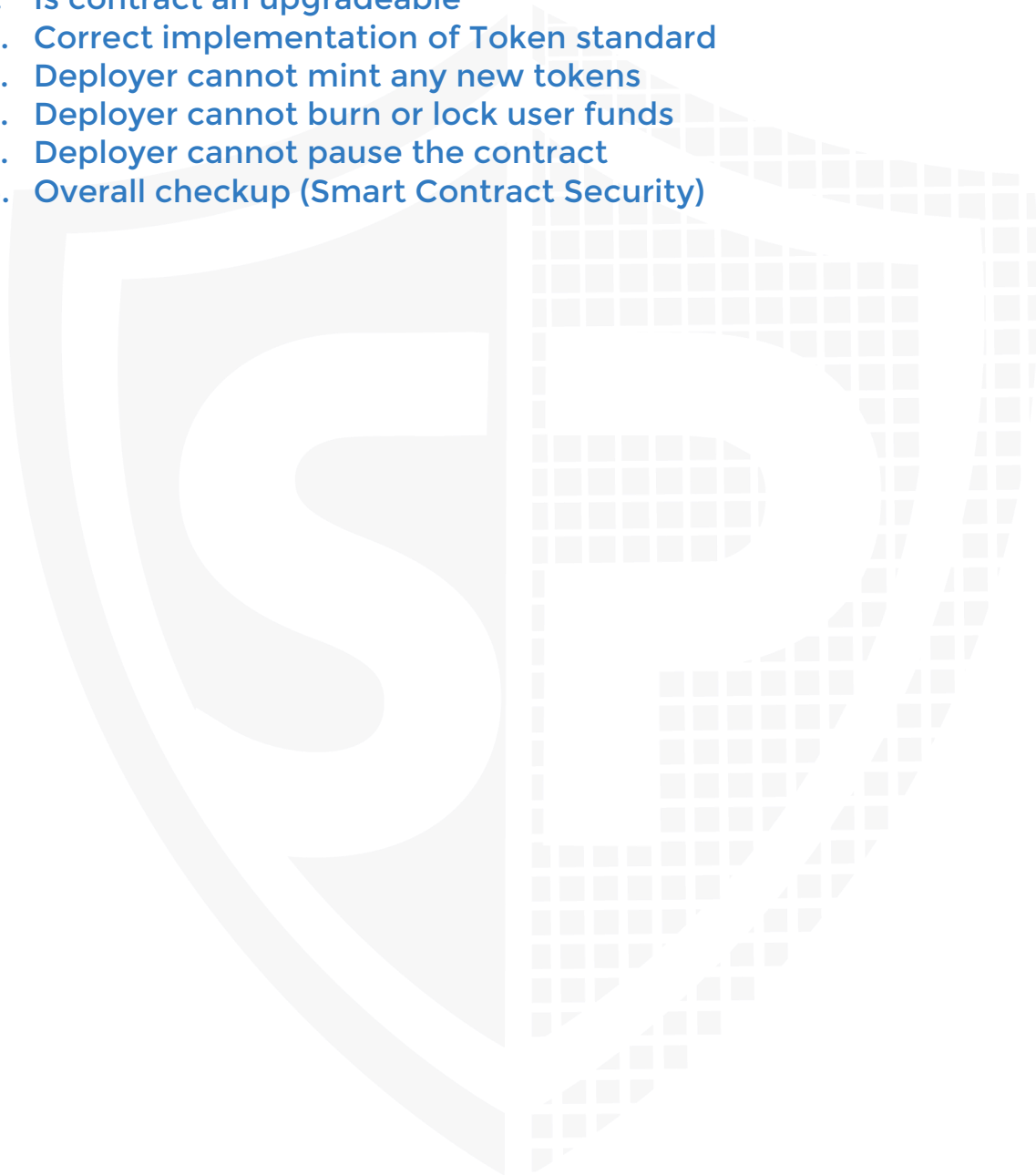


## Scope of Work/Verify Claims

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. Deployer cannot mint any new tokens
4. Deployer cannot burn or lock user funds
5. Deployer cannot pause the contract
6. Overall checkup (Smart Contract Security)



## Is contract an upgradeable

Name	
Is contract an upgradeable?	Yes

### Comment:

The owner can update the contract code after deployment by uploading a new one





## 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			

## Deployer cannot mint any new tokens

Name	Exist	Tested	Status
Deployer cannot mint			
Max / Total Supply and last Token ID	N/A		



## Deployer cannot burn or lock user funds

Name	Exist	Tested	Status
Deployer cannot lock			
Deployer cannot burn			



Deployer cannot pause the contract

Name	Exist	Tested	Status
Deployer cannot pause			



## 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

## AgentManager

- ◆ initialize
- ◆ updateUser
- ◆ updatePool
- ◆ massUpdatePools
- ◆ deposit
- ◆ withdraw
- ◆ emergencyWithdraw
- ◆ cancelPendingWithdraw
- ◆ harvest
- ◆ harvestAll
- ◆ add
- Ⓜ onlyOwner
- ◆ addTarget
- Ⓜ onlyOwner
- ◆ updateAgents
- Ⓜ onlyOwner
- ◆ updateAgentToNewContract
- Ⓜ onlyOwner
- ◆ setAutoTarget
- Ⓜ onlyOwner
- ◆ toggleAutoTarget
- Ⓜ onlyOwner
- ◆ setFeeAddress
- ◆ updateConfig
- Ⓜ onlyOwner
- ◆ distributeProfit
- Ⓜ onlyOwner

## AutoFarm

- ◆ <Constructor> 💰
- ◆ init
- ◆ deposit
- Ⓜ onlyOwner
- ◆ withdraw
- Ⓜ onlyOwner

## RandomNumberGenerator

- ◆ <Constructor>
- ◆ getRandomNumber
- ◆ setFee
- Ⓜ onlyOwner
- ◆ setKeyHash
- Ⓜ onlyOwner
- ◆ setLotteryAddress
- Ⓜ onlyOwner
- ◆ withdrawTokens

## InsuraceAgent

- ◆ <Constructor> 💰
- ◆ init
- ◆ deposit
- Ⓜ onlyOwner
- ◆ withdraw
- Ⓜ onlyOwner
- ◆ unlockWithdraw
- Ⓜ onlyOwner
- ◆ harvest
- Ⓜ onlyOwner
- ◆ unlockHarvest
- Ⓜ onlyOwner

## CALottery































- ◆ buyTickets
- Ⓜ notContract
- Ⓜ nonReentrant
- ◆ claimTickets
- Ⓜ notContract
- Ⓜ nonReentrant
- ◆ closeLottery
- Ⓜ onlyOperator
- Ⓜ nonReentrant
- ◆ drawFinalNumberAndMakeLotteryClaimable
- Ⓜ onlyOperator
- Ⓜ nonReentrant
- ◆ changeRandomGenerator
- Ⓜ onlyOwner
- ◆ injectFunds
- Ⓜ onlyOwnerOrInjector
- ◆ startLottery
- Ⓜ onlyOperator

## Comments:

- The owner add targets, update agents to a new contract and distribute profits.
- Owner can manage the lottery like, start lottery, end lottery and inject funds into it.
- Owner can withdraw and harvest from farm

# Source Units in Scope

## v1.0

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
	contracts/interfaces/IBaseToken.sol	_____	1	28	6	4	_____	21	
	contracts/interfaces/ICombinableTokenBasis.sol	_____	1	17	8	5	1	13	_____
	contracts/interfaces/IWyvernProxyRegistry.sol	1	1	10	9	5	2	5	_____
	contracts/interfaces/ICombinationToken.sol	_____	1	26	8	5	1	13	_____
	contracts/interfaces/IBasis.sol	_____	1	15	7	4	1	9	_____
	contracts/interfaces/IMembershipToken.sol	_____	1	15	7	4	1	11	_____
	contracts/interfaces/IWithdrawable.sol	_____	1	8	5	3	1	7	_____
	contracts/S1CombinationToken.sol	1	_____	346	315	180	84	106	  
	contracts/BaseToken.sol	1	_____	542	511	291	156	136	  
	contracts/MembershipToken.sol	1	_____	147	134	60	52	41	_____
	contracts/library/Basis.sol	1	_____	72	58	40	6	27	_____
	contracts/library/CombinableTokenBasis.sol	1	_____	75	71	58	_____	33	_____
	contracts/library/ERC721Buyable.sol	1	_____	247	209	186	4	80	  
	contracts/library/Withdrawable.sol	1	_____	35	35	26	1	24	
	contracts/opensea/ERC721Tradable.sol	1	_____	28	23	13	5	13	_____
	Totals	9	7	1611	1406	884	315	539	  

## Legend

Attribute	Description
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, ...)

# Audit Results

# AUDIT PASSED

## Critical issues

No critical issues

## High issues

Issue	File	Type	Line	Description
#1	AgentManager.sol	The contract can be initialized more than once	90	The function has public visibility, it can be called by anyone and the initialize variable has never been set to 'true'
#2	AutoFarm.sol	The contract can be initialized more than once	100	The function has public visibility, it can be called by anyone and the initialize variable has never been set to 'true'
#3	Liquidity.sol	The contract can be initialized more than once	87	The function has public visibility, it can be called by anyone and the initialize variable has never been set to 'true'

## Medium issues

Issue	File	Type	Line	Description
#1	AgentManager.sol	Function not working properly	269	If user amount is higher than 0 it will refund the user but if the amount is set to 0 before transferring, everyone will get 0 funds



#2	CALottery.sol	Wrong calculation for buying	139	The amount to buy a ticket will be divided by 1e18 but while transferring the amount, it is not converted again.
----	---------------	------------------------------	-----	--

## Low issues

Issue	File	Type	Line	Description
#1	All	A floating pragma is set	7	The current pragma Solidity directive is „^0.8.0/6“.
#2	AgentManager.sol	Access Control	209	The function has public visibility which means any user can update other users.
#3	AgentManager.sol	Access Control	253	The function has public visibility which means users can pass some random _pid for a particular amount. We suggest to put a check whether the user exists in the particular pool
#4	AgentManager.sol	Missing Length check	365,398	The function should check that the lengths of the array passed in the parameters is the same
#5	AgentManager.sol	Missing zero check	90,209,350,365,444,464,469	Check that the address is not zero
#6	AgentManager.sol	Missing Events	90,350,365,398,453,460,464,469	Emit an event for critical parameter changes.
#7	AgentManager.sol	Unnecessary Check	403	Unnecessary check because it was checked before and uint256 can't be below 0. It will always be true
#8	Liquidity.sol	Missing Events	133	Emit an event for critical parameter changes.
#9	AutoFarm.sol	Missing Events	100,114	Emit an event for critical parameter changes.
#10	InsurAceAgent.sol	Missing Events	137,114	Emit an event for critical parameter changes.
#11	Free.sol	Missing zero check	19	Check that the address is not zero
#12	RandomNumberGenerator.sol	Missing zero check	75	Check that the address is not zero

#13	Timelock.sol	Missing zero check	207	Check that the address is not zero
-----	--------------	--------------------	-----	------------------------------------

## Informational issues

Issue	File	Type	Line	Description
#1	AgentManager.sol	Wrong Error Message	456	Misleading error message, there should be "bigger than 0"
#2	AutoFarm.sol	Redundant Code	181,185,189, 233,237,247, 251	These functions are redundant and have no functionality in code. They, should either be removed or used.
#3	AutoFarm.sol	Redundant Code	3	This line is redundant and have no functionality in code. It should be removed.
#4	AgentManager /AutoFarm/InsuranceAgent/CALottery/Liquidity.sol	Unused Return Value	-	Always check and take care of the return value from a function call

## Audit Comments

We recommend you to use the special form of comments (NatSpec Format, Follow link for more information <https://docs.soliditylang.org/en/v0.5.10/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.

06. August, 2022:

- There is still an owner (Owner still has not renounced ownership)
- We recommend to use randomizations by external sources like VRF because solidity has no randomization feature of its own.
- We recommend to unit test these tests with more than 95% of test coverage before deployment
- Read the whole report and modifiers section for more information.

## SWC Attacks

ID	Title	Relationships	Status
<a href="#">SWC-1136</a>	Unencrypted Private Data On-Chain	<a href="#">CWE-767: Access to Critical Private Variable via Public Method</a>	PASSED
<a href="#">SWC-1135</a>	Code With No Effects	<a href="#">CWE-1164: Irrelevant Code</a>	PASSED
<a href="#">SWC-1134</a>	Message call with hardcoded gas amount	<a href="#">CWE-655: Improper Initialization</a>	PASSED
<a href="#">SWC-1133</a>	Hash Collisions With Multiple Variable Length Arguments	<a href="#">CWE-294: Authentication Bypass by Capture-replay</a>	PASSED
<a href="#">SWC-1132</a>	Unexpected Ether balance	<a href="#">CWE-667: Improper Locking</a>	PASSED
<a href="#">SWC-1131</a>	Presence of unused variables	<a href="#">CWE-1164: Irrelevant Code</a>	PASSED

131			
SWC:130	Right-To-Left-Override control character (U+202E)	<a href="#">CWE-451: User Interface (UI) Misrepresentation of Critical Information</a>	PASSED
SWC:129	Typographical Error	<a href="#">CWE-480: Use of Incorrect Operator</a>	PASSED
SWC:128	DoS With Block Gas Limit	<a href="#">CWE-400: Uncontrolled Resource Consumption</a>	PASSED
SWC:127	Arbitrary Jump with Function Type Variable	<a href="#">CWE-695: Use of Low-Level Functionality</a>	PASSED
SWC:125	Incorrect Inheritance Order	<a href="#">CWE-696: Incorrect Behavior Order</a>	PASSED
SWC:	Write to Arbitrary	<a href="#">CWE-123: Write-what-where Condition</a>	PASSED

<u>1</u> <u>2</u> <u>4</u>	Storage Location		
<u>S</u> <u>W</u> <u>C</u> : <u>1</u> <u>2</u> <u>3</u>	Requirement Violation	<a href="#">CWE-573: Improper Following of Specification by Caller</a>	PASSED
<u>S</u> <u>W</u> <u>C</u> : <u>1</u> <u>2</u> <u>2</u>	Lack of Proper Signature Verification	<a href="#">CWE-345: Insufficient Verification of Data Authenticity</a>	PASSED
<u>S</u> <u>W</u> <u>C</u> : <u>1</u> <u>2</u> <u>1</u>	Missing Protection against Signature Replay Attacks	<a href="#">CWE-347: Improper Verification of Cryptographic Signature</a>	PASSED
<u>S</u> <u>W</u> <u>C</u> : <u>1</u> <u>2</u> <u>0</u>	Weak Sources of Randomness from Chain Attributes	<a href="#">CWE-330: Use of Insufficiently Random Values</a>	PASSED
<u>S</u> <u>W</u> <u>C</u> : <u>1</u> <u>1</u> <u>1</u> <u>9</u>	Shadowing State Variables	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	NOT PASSED

<a href="#">S</a> <a href="#">W</a> <a href="#">C</a> : <a href="#">1</a> <a href="#">1</a> <a href="#">8</a>	Incorrect Constructor Name	<a href="#">CWE-665: Improper Initialization</a>	PASSED
<a href="#">S</a> <a href="#">W</a> <a href="#">C</a> : <a href="#">1</a> <a href="#">1</a> <a href="#">7</a>	Signature Malleability	<a href="#">CWE-347: Improper Verification of Cryptographic Signature</a>	PASSED
<a href="#">S</a> <a href="#">W</a> <a href="#">C</a> : <a href="#">1</a> <a href="#">1</a> <a href="#">6</a>	Timestamp Dependence	<a href="#">CWE-829: Inclusion of Functionality from Untrusted Control Sphere</a>	PASSED
<a href="#">S</a> <a href="#">W</a> <a href="#">C</a> : <a href="#">1</a> <a href="#">1</a> <a href="#">5</a>	Authorization through tx.origin	<a href="#">CWE-477: Use of Obsolete Function</a>	PASSED
<a href="#">S</a> <a href="#">W</a> <a href="#">C</a> : <a href="#">1</a> <a href="#">1</a> <a href="#">4</a>	Transaction Order Dependence	<a href="#">CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')</a>	PASSED
<a href="#">S</a> <a href="#">W</a> <a href="#">C</a> : <a href="#">1</a> <a href="#">1</a> <a href="#">3</a>	DoS with Failed Call	<a href="#">CWE-703: Improper Check or Handling of Exceptional Conditions</a>	PASSED

<a href="#">S</a> <a href="#">W</a> <a href="#">C</a> : 1 1 2	Delegatecall to Untrusted Callee	<a href="#">CWE-829: Inclusion of Functionality from Untrusted Control Sphere</a>	PASSED
<a href="#">S</a> <a href="#">W</a> <a href="#">C</a> : 1 1 1	Use of Deprecated Solidity Functions	<a href="#">CWE-477: Use of Obsolete Function</a>	PASSED
<a href="#">S</a> <a href="#">W</a> <a href="#">C</a> : 1 1 0	Assert Violation	<a href="#">CWE-670: Always-Incorrect Control Flow Implementation</a>	PASSED
<a href="#">S</a> <a href="#">W</a> <a href="#">C</a> : 1 0 9	Uninitialized Storage Pointer	<a href="#">CWE-824: Access of Uninitialized Pointer</a>	PASSED
<a href="#">S</a> <a href="#">W</a> <a href="#">C</a> : 1 0 8	State Variable Default Visibility	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	PASSED
<a href="#">S</a> <a href="#">W</a> <a href="#">C</a> : 1 0 7	Reentrancy	<a href="#">CWE-841: Improper Enforcement of Behavioral Workflow</a>	PASSED

<a href="#">S</a> <a href="#">W</a> <a href="#">C</a> : : 1 0 6	Unprotected SELFDESTR UCT Instruction	<a href="#">CWE-284: Improper Access Control</a>	PASSED
<a href="#">S</a> <a href="#">W</a> <a href="#">C</a> : : 1 0 5	Unprotected Ether Withdrawal	<a href="#">CWE-284: Improper Access Control</a>	PASSED
<a href="#">S</a> <a href="#">W</a> <a href="#">C</a> : : 1 0 4	Unchecked Call Return Value	<a href="#">CWE-252: Unchecked Return Value</a>	PASSED
<a href="#">S</a> <a href="#">W</a> <a href="#">C</a> : : 1 0 3	Floating Pragma	<a href="#">CWE-664: Improper Control of a Resource Through its Lifetime</a>	NOT PASSED
<a href="#">S</a> <a href="#">W</a> <a href="#">C</a> : : 1 0 2	Outdated Compiler Version	<a href="#">CWE-937: Using Components with Known Vulnerabilities</a>	PASSED
<a href="#">S</a> <a href="#">W</a> <a href="#">C</a> : : 1 0 1	Integer Overflow and Underflow	<a href="#">CWE-682: Incorrect Calculation</a>	PASSED



<div> <div>S</div> <div>W</div> <div>C</div> <div>.</div> <div>1</div> <div>1</div> <div>0</div> <div>0</div> <div>0</div> <div>1</div> </div>	Function Default Visibility	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	PASSED
--	-----------------------------------	---	--------





[SolidProof.io](https://SolidProof.io)



[@solidproof\\_io](https://t.me/solidproof_io)

Solid  
Proofed

**Blockchain Security | Smart Contract Audits | KYC**

  
MADE IN GERMANY