RD AUDITORS

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Customer: Swapp Protocol
Prepared on: 7/04/2021
Platform: Ethereum
Language: Solidity

Table of Contents

THIS DOCUMENT MAY CONTAIN CONFIDENTIAL INFORMATION ABOUT ITS SYSTEMS AND 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 DECISION OF CUSTOMER.

Document

Name	Smart Contract Code Review and Security Analysis Report for Swapp Protocol
Platform	Ethereum / Solidity
File 1	LiquidityGuard.sol
MD5 hash	F928CB95502B9073051DC8EBCA0005D2
SHA256 hash	8F127744A574A5F835553C933D17B7BCA9D6CF31C6A1C58AFDD00220CA22ED70
File 2	LiquidityTransformer.sol
MD5 hash	18344D464B55BF0ABA4966901A7A19AA
SHA256 hash	8CF6BED26484EFB6C24A12F77AFDD45DAD44A3D40CE42D765DB729B6566C8C31
File 3	Swapp.sol
MD5 hash	B6F012F3789269BF2556F2C31E841843
SHA256 hash	D0FF3DA4E2C380E592CEF753FA0758D00026209EFEF2F45BF1306920C709C0DE
File 4	ProvableAPI.sol
MD5 hash	41E51159DD3A55768B36FBF5E2913E02
SHA256 hash	C5E992B1F319DE24999CE1E1F1D691EA8EF7344F78F546A15C4F4CED3B615350
File 5	Snapshot.sol
MD5 hash	532EE5B380C5FCC1D61921FC9C37FDCB
SHA256 hash	59BA7BF31D7A2D4A73E08975686BF1B9D718018756E9FD7106EBC97D609C0B3B
File 6	Migration.sol
MD5 hash	CA8D6CA8A6EDF34F149A5095A8B074C9

SHA256 hash	8A6B38936C738A0E612391EE231F39352CC8878F4A5B41C05F0895FB662B3FD6
File 7	YieldFarm.Sol
MD5 hash	513B071B6805C8422091190991B59C0A
SHA256 hash	08FA9D320BC3DBD6A60A1CDBB3476AEA2F5181275922FC989D21CB5811FD6CF9
File 8	YieldFarmLP.Sol
MD5 hash	7AC2EACF68A64477E3381891110AD71A
SHA256 hash	BB022B3D22375F0D27B2569012970DA6463BD876958E834168E96EEA7F7FB05A
File 9	Staking.Sol
MD5 hash	710DD9D6D8EEEE01C5B4470057E21D57
SHA256 hash	E6C4CAEA6B730D107A4E3A07BFE7378DFAA00C4B5D1BEB7D2CDCD74398E797E1
File 10	StakingToken.Sol
MD5 hash	FA40F38DD22E55CCA19C249B6CE2F0D4
SHA256 hash	625758925824E4E5D158B2A161B14A41F587EB5D02A1A5E948997D6CFB640CE2
File 11	Declaration.sol
MD5 hash	E05FBBCB99725505FADFF3A46CD030D6
SHA256 hash	949BDF5A53D670F3FEBF9467571DA1AA15A18BF6F58C87FAEB4ABD5BBDB9A022
File 12	Helper.sol
MD5 hash	3DAF10A492B78AD24A8FC9180CCAFF0E
SHA256 hash	80D6B863A41643C0D915A34AA70C1B8926BC14E2656F32DE1F807DE7191663D7
Date	**7/04/2021**

Introduction

RD Auditors (Consultant) was contracted by Swapp Protocol (Customer) to conduct a Smart Contracts Code Review and Security Analysis. This report presents the findings of the security assessment of Customer`s smart contracts and its code review conducted between March 20, 2021 – April 7, 2021.

This contract consists of many files.

Project Scope

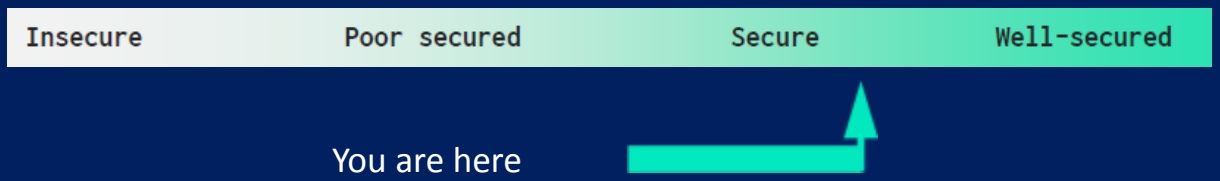
The scope of the project is a smart contract.

We have scanned these smart contracts for commonly known and more specific vulnerabilities, below are those considered (the full list includes but not limited to):

- Reentrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with (Unexpected) Throw
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Byte array vulnerabilities
- Style guide violation
- Transfer forwards all gas
- ERC20 API violation
- Malicious libraries
- Compiler version not fixed
- Unchecked external call - Unchecked math
- Unsafe type inference
- Implicit visibility level

Executive Summary

According to the assessment, the customer`s solidity smart contract is close to secured.



Automated checks are with smartDec, Mythril, Slither and remix IDE. All issues were performed by our team, which included the analysis of code functionality, manual audit found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the audit overview section. The general overview is presented in the AS-IS section and all issues found are located in the audit overview section.

We found 0 critical, high, 0 medium, 0 low and some very low level issues.

Code Quality

Swapp Protocol consists of multiple smart contract files. The team has also conducted unit tests using scripts provided through the same github link which fortify functionality and security of the contract, which also helped us to determine the integrity of the code in an automated way.

Overall, the code is not commented. Commenting provides rich documentation for functions, return variables and more and also helps auditors to quick cover the flow behind code logic. Use of Ethereum Natural Language Specification Format (NatSpec) for commenting is recommended.

Documentation

We were given Swapp contract and its supporting files in the form of a github link:

https://github.com/Swapp-Token/swapp-contracts

The hash of that file is mentioned in the table. It's recommended to write comments in the smart contract code, so anyone can quickly understand the programming flow as well as complex code logic. Comments are very helpful in understanding the overall architecture of the protocol. It also provides a clear overview of the system components, including helpful details, like the lifetime of the background script.

Use of Dependencies

As per our observation, the libraries are used in this smart contract infrastructure. Those were based on well known industry standard open source projects. And even core code blocks are written well and systematically.

AS-IS Overview

Swapp Contract Overview

It's collection of smart contracts which provides swapp functionality to users.

File And Function Level Report

File: Declaration.sol

Contract: Declaration
Import: Global.sol
Observation: All Passed including security check
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	CreatePair	write	Passed	All Passed	No Issue	Passed

File: LiquidityGuard.sol

Contract: LiquidityGuard
Observation: All Passed including security check
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	getInflation	read	Passed	All Passed	No Issue	Passed
2	assignInflation	write	Passed	All Passed	No Issue	Passed

File: LiquidityTransformer.sol

Contract: LiquidityTransformer
Import: ProvableAPI_0.6.sol
Inherit: usingProvable
Observation: All Passed including security check
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	defineToken	write	Passed	All Passed	No Issue	Passed
2	revokeAccess	write	Passed	All Passed	No Issue	Passed
3	reserveSwapp	write	Passed	All Passed	No Issue	Passed
4	_addBalance	write	Passed	All Passed	No Issue	Passed
5	_trackInvesters	write	Passed	All Passed	No Issue	Passed
6	_trackReferrals	write	Passed	All Passed	No Issue	Passed
7	generateSupply	write	Passed	All Passed	No Issue	Passed
8	_generateStaticSupply	write	Passed	All Passed	No Issue	Passed
9	_generateRandomSupply	write	Passed	All Passed	No Issue	Passed
10	_callback	write	Passed	All Passed	No Issue	Passed
11	_timeOut	write	Passed	All Passed	No Issue	Passed
12	PrepareReferralBonus	write	Passed	All Passed	No Issue	Passed
13	_superReferralBonus	write	Passed	All Passed	No Issue	Passed
14	_influencerReferralBonus	write	Passed	All Passed	No Issue	Passed
15	_familyReferralBonus	write	Passed	All Passed	No Issue	Passed
16	forwardLiquidity	write	Passed	All Passed	No Issue	Passed
17	getMyTokens	write	Passed	All Passed	No Issue	Passed
18	PayoutInvestorAddress	write	Passed	All Passed	No Issue	Passed
19	PayoutReferralAddress	write	Passed	All Passed	No Issue	Passed
20	PayoutInvestmentDayBatch	write	Passed	All Passed	No Issue	Passed
21	PayoutReferralBatch	write	Passed	All Passed	No Issue	Passed
22	userInvestmentAmountAllDays	read	Passed	All Passed	No Issue	Passed
23	userTotalInvestmentAmount	read	Passed	All Passed	No Issue	Passed
24	InvestorsOnDay	read	Passed	All Passed	No Issue	Passed
25	InvestorsOnAllDays	read	Passed	All Passed	No Issue	Passed
26	InvestmentOnAllDays	read	Passed	All Passed	No Issue	Passed
27	SupplyOnAllDays	read	Passed	All Passed	No Issue	Passed
28	CheckInvestmentDays	read	Passed	All Passed	No Issue	Passed
29	PreparePath	read	Passed	All Passed	No Issue	Passed
30	_teamContribution	read	Passed	All Passed	No Issue	Passed
31	fundedDays	read	Passed	All Passed	No Issue	Passed

32	_CalculateDailyRatio	read	Passed	All Passed	No Issue	Passed
33	_CurrentSwappDay	read	Passed	All Passed	No Issue	Passed
34	requestRefund	write	Passed	All Passed	No Issue	Passed
35	requestTeamFunds	write	Passed	All Passed	No Issue	Passed
36	notContract	read	Passed	All Passed	No Issue	Passed

File:Swapp.sol

Contract: Context

Observation: All Passed including security check

Test Report: Passed

Score: Passed

Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	_msgSender	read	Passed	All Passed	No Issue	Passed
2	_msgData	read	Passed	All Passed	No Issue	Passed

Contract: ERC20

Observation: All Passed including security check

Test Report: Passed

Score: Passed

Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	name	read	Passed	All Passed	No Issue	Passed
2	symbol	read	Passed	All Passed	No Issue	Passed
	decimals	read	Passed	All Passed	No Issue	Passed
	totalSupply	read	Passed	All Passed	No Issue	Passed
	balanceOf	read	Passed	All Passed	No Issue	Passed
	transfer	write	Passed	All Passed	No Issue	Passed
	allowance	read	Passed	All Passed	No Issue	Passed
	approve	write	Passed	All Passed	No Issue	Passed
	tansferFrom	write	Passed	All Passed	No Issue	Passed
	_transfer	write	Passed	All Passed	No Issue	Passed

	_mint	write	Passed	All Passed	No Issue	Passed
	burn	write	Passed	All Passed	No Issue	Passed
	_burn	write	Passed	All Passed	No Issue	Passed
	approve	write	Passed	All Passed	No Issue	Passed

Contract: Global

Import: ERC20, Events

Observation: All Passed including security check

Test Report: Passed

Score: Passed

Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	_increaseGlobals	write	Passed	All Passed	No Issue	Passed
2	_decreaseGlobals	write	Passed	All Passed	No Issue	Passed
3	_logGlobals	write	Passed	All Passed	No Issue	Passed

Contract: Helper

Observation: All Passed including security check

Test Report: Passed

Score: Passed

Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	notContract	read	Passed	All Passed	No Issue	Passed
2	getCreateID	read	Passed	All Passed	No Issue	Passed
3	StakePagination	read	Passed	All Passed	No Issue	Passed
4	referralsPagination	read	Passed	All Passed	No Issue	Passed
5	isMatureStake	read	Passed	All Passed	No Issue	Passed
6	_stakeNotStarted	readf	Passed	All Passed	No Issue	Passed
7	_daysLeft	read	Passed	All Passed	No Issue	Passed
8	_PreparePath	read	Passed	All Passed	No Issue	Passed
9	SafeTransfer	write	Passed	All Passed	No Issue	Passed
10	SafeTransferFrom	write	Passed	All Passed	No Issue	Passed

Contract: Snapshot
inherit: Helper
Observation: All Passed including security check
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	LiquidityGuardTrigger	write	Passed	All Passed	No Issue	Passed
2	_dailySnapshotPoint	write	Passed	All Passed	No Issue	Passed
3	adjustLiquidityRates	write	Passed	All Passed	No Issue	Passed

Contract: ReferralToken
inherit: Helper
Observation: All Passed including security check
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	_addReferrerShareToEnd	write	Passed	All Passed	No Issue	Passed
2	_removeeReferrerSharesToEnd	write	Passed	All Passed	No Issue	Passed
3	belowThresholdLevel	read	Passed	All Passed	No Issue	Passed
4	_addCriticalMass	write	Passed	All Passed	No Issue	Passed
5	_removeCriticalMass	write	Passed	All Passed	No Issue	Passed
6	_determineActivationDay	read	Passed	All Passed	No Issue	Passed
7	_activationDay	read	Passed	All Passed	No Issue	Passed
8	_updateDaiEquivalent	write	Passed	All Passed	No Issue	Passed
9	referrerInterest	write	Passed	All Passed	No Issue	Passed
10	referrerInterestBulk	write	Passed	All Passed	No Issue	Passed
11	_referrerIntrest	write	Passed	All Passed	No Issue	Passed
12	checkReferralsByID	read	Passed	All Passed	No Issue	Passed
13	_getReferralInterest	read	Passed	All Passed	No Issue	Passed
14	_determineStartDay	read	Passed	All Passed	No Issue	Passed
15	_determineFinalDay	read	Passed	All Passed	No Issue	Passed

Contract: StakingToken

inherit: ReferralToken

Observation: All Passed including security check

Test Report: Passed

Score: Passed

Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	CreateStakeBulk	write	Passed	All Passed	No Issue	Passed
2	CreateStake	write	Passed	All Passed	No Issue	Passed
3	_CreateStake	write	Passed	All Passed	No Issue	Passed
4	endStake	write	Passed	All Passed	No Issue	Passed
5	_endStake	write	Passed	All Passed	No Issue	Passed
6	ScrapeInterest	write	Passed	All Passed	No Issue	Passed
7	_addScheduledShares	write	Passed	All Passed	No Issue	Passed
8	_removeScheduledShares	write	Passed	All Passed	No Issue	Passed
9	_SharePriceUpdate	write	Passed	All Passed	No Issue	Passed
10	_getNewShareprice	read	Passed	All Passed	No Issue	Passed
11	CheckMatureStake	read	Passed	All Passed	No Issue	Passed
12	CheckStakeByID	read	Passed	All Passed	No Issue	Passed
13	_stakesShares	read	Passed	All Passed	No Issue	Passed
14	_SharesAmount	read	Passed	All Passed	No Issue	Passed
15	_getBonus	read	Passed	All Passed	No Issue	Passed
16	_regularBonus	read	Passed	All Passed	No Issue	Passed
17	_baseAmount	read	Passed	All Passed	No Issue	Passed
18	_referrerShares	read	Passed	All Passed	No Issue	Passed
19	_StorePenalty	write	Passed	All Passed	No Issue	Passed
20	_CalculateRewardAmount	read	Passed	All Passed	No Issue	Passed
21	_LoopRewardAmount	read	Passed	All Passed	No Issue	Passed

Contract: Swapp Token

inherit: StakingToken

Observation: All Passed including security check

Test Report: Passed

Score: Passed

Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	SetMinters	write	Passed	All Passed	No Issue	Passed
2	burnMinterDefiner	write	Passed	All Passed	No Issue	Passed
3	mintSupply	write	Passed	All Passed	No Issue	Passed
4	giveStatus	write	Passed	All Passed	No Issue	Passed
5	CreateStakeWithETH	write	Passed	All Passed	No Issue	Passed
6	createStakeWithToken	write	Passed	All Passed	No Issue	Passed

File: Migrations.sol

Contract: Migration

Observation: All Passed including security check

Test Report: Passed

Score: Passed

Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	SetMinters	write	Passed	All Passed	No Issue	Passed

File: ProvableAPI_0.6

Contract: UsingProvable
Observation: All Passed including security check
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	Provable_SetNetwork	write	Passed	All Passed	No Issue	Passed
2	Provable_SetNetworkName	write	Passed	All Passed	No Issue	Passed
3	Provable_SetNetworkName	write	Passed	All Passed	No Issue	Passed
4	Provable_getNetworkName	read	Passed	All Passed	No Issue	Passed
5	Provable_setNetwork	write	Passed	All Passed	No Issue	Passed
6	Provable_randomOS_Proof Verify	write	Passed	All Passed	No Issue	Passed
7	_Callback	write	Passed	All Passed	No Issue	Passed
8	Provable_getPrice (multiple version)	write	Passed	All Passed	No Issue	Passed

File: Snapshot

Contract: Snapshot
inherit: Helper
Import; Helper
Observation: All Passed including security check
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	liquidityGuardTrigger	write	Passed	All Passed	No Issue	Passed
2	enableLiquidityGuard	write	Passed	All Passed	No Issue	Passed
3	disableLiquidityGuard	write	Passed	All Passed	No Issue	Passed
4	manualDailySnapshot	write	Passed	All Passed	No Issue	Passed
5	_dailySnapShotPoint	write	Passed	All Passed	No Issue	Passed
6	adjustLiquidityRates	write	Passed	All Passed	No Issue	Passed
7	_inflationAmount	read	Passed	All Passed	No Issue	Passed
8	_referralInflation	read	Passed	All Passed	No Issue	Passed
9	_liquidityInflation	read	Passed	All Passed	No Issue	Passed

File:Staking.sol

Contract: Staking
inherit: ReentrancyGuard
Import: IERC20.sol, safeMath.sol, ReentrancyGuard
Observation: All Passed including security check
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	CheckStableCoin	read	Passed	All Passed	No Issue	Passed
2	deposit	write	Passed	All Passed	No Issue	Passed
3	withdraw	write	Passed	All Passed	No Issue	Passed
4	manualEpochInit	write	Passed	All Passed	No Issue	Passed
5	emergencyWithdraw	write	Passed	All Passed	No Issue	Passed
6	getEPOchUserBalance	read	Passed	All Passed	No Issue	Passed
7	balanceOf	read	Passed	All Passed	No Issue	Passed
8	getCurrentEPoch	read	Passed	All Passed	No Issue	Passed
9	getEPochPoolSize	read	Passed	All Passed	No Issue	Passed
10	currentEPochMultiplier	read	Passed	All Passed	No Issue	Passed
11	ComputeNewMultiplier	read	Passed	All Passed	No Issue	Passed
12	epochIsInitialized	read	Passed	All Passed	No Issue	Passed
13	getCheckPointBalance	read	Passed	All Passed	No Issue	Passed
14	getCheckPointEffectiveBalance	read	Passed	All Passed	No Issue	Passed

File: StakingToken

Contract: StakingToken
import: ReferralToken.sol
Observation: All Passed including security check
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	CreateStakeBulk	write	Passed	All Passed	No Issue	Passed
2	CreateStake	write	Passed	All Passed	No Issue	Passed
3	_CreateStake	write	Passed	All Passed	No Issue	Passed
4	endStake	write	Passed	All Passed	No Issue	Passed
5	_endStake	write	Passed	All Passed	No Issue	Passed
6	ScrapeInterest	write	Passed	All Passed	No Issue	Passed
7	_addScheduledShares	write	Passed	All Passed	No Issue	Passed
8	_removeScheduledShares	write	Passed	All Passed	No Issue	Passed
9	_SharePriceUpdate	write	Passed	All Passed	No Issue	Passed
10	_getNewShareprice	read	Passed	All Passed	No Issue	Passed
11	CheckMatureStake	read	Passed	All Passed	No Issue	Passed
12	CheckStakeByID	read	Passed	All Passed	No Issue	Passed
13	_stakesShares	read	Passed	All Passed	No Issue	Passed
14	_SharesAmount	read	Passed	All Passed	No Issue	Passed
15	_getBonus	read	Passed	All Passed	No Issue	Passed
16	_regularBonus	read	Passed	All Passed	No Issue	Passed
17	_baseAmount	read	Passed	All Passed	No Issue	Passed
18	_referrerShares	read	Passed	All Passed	No Issue	Passed
19	_StorePenalty	write	Passed	All Passed	No Issue	Passed
20	_CalculateRewardAmount	read	Passed	All Passed	No Issue	Passed
21	_LoopRewardAmount	read	Passed	All Passed	No Issue	Passed

Contract: Helper
Inherit: Declaration
Import: Declaration.sol
Observation: All Passed including security check
Test Report: passed
Score: passed
Conclusion: passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	CurrentSwappDay	read	Passed	All Passed	No Issue	Passed
2	_CurrentSwppDay	read	Passed	All Passed	No Issue	Passed
3	nextSwappDay	read	Passed	All Passed	No Issue	Passed
4	PreviousSwappDay	read	Passed	All Passed	No Issue	Passed
5	SwappDayFromStamp	read	Passed	All Passed	No Issue	Passed
6	getNow	read	Passed	All Passed	No Issue	Passed
7	notContract	read	Passed	All Passed	No Issue	Passed
8	toByte16	read	Passed	All Passed	No Issue	Passed
9	generateID	read	Passed	All Passed	No Issue	Passed
10	generateStakeID	read	Passed	All Passed	No Issue	Passed
11	generateReferralID	read	Passed	All Passed	No Issue	Passed
12	StakesPagination	read	Passed	All Passed	No Issue	Passed
13	referralsPagination	read	Passed	All Passed	No Issue	Passed
14	latestStakeID	read	Passed	All Passed	No Issue	Passed
15	latestReferralID	read	Passed	All Passed	No Issue	Passed
16	_increaseStakeCount	write	Passed	All Passed	No Issue	Passed
17	_increaseReferralCount	write	Passed	All Passed	No Issue	Passed
18	_isMatureStake	read	Passed	All Passed	No Issue	Passed
19	_notCriticalMassReferrer	read	Passed	All Passed	No Issue	Passed
20	_StakeNotStarted	read	Passed	All Passed	No Issue	Passed
21	_StakeEnded	read	Passed	All Passed	No Issue	Passed
22	_daysLeft	read	Passed	All Passed	No Issue	Passed
23	_daysDiff	read	Passed	All Passed	No Issue	Passed
24	_CalculationDay	read	Passed	All Passed	No Issue	Passed
25	_StartingDay	read	Passed	All Passed	No Issue	Passed
26	_notFuture	read	Passed	All Passed	No Issue	Passed
27	_notPast	read	Passed	All Passed	No Issue	Passed
28	_nonZeroAddress	read	Passed	All Passed	No Issue	Passed
29	_getLockDays	read	Passed	All Passed	No Issue	Passed
30	_PreparePath	read	Passed	All Passed	No Issue	Passed
31	SafeTransfer	write	Passed	All Passed	No Issue	Passed

32	safeTransferFrom	write	Passed	All Passed	No Issue	Passed

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to tokens loss etc.
High	High-level vulnerabilities are difficult to exploit; however, they also have significant impact on smart contract execution, e.g. public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to tokens lose
Low	Low-level vulnerabilities are mostly related to outdated, unused etc. code snippets, that can't have significant impact on execution
Lowest / Code Style / Best Practice	Lowest-level vulnerabilities, code style violations and info statements can't affect smart contract execution and can be ignored.

Audit Findings

Critical

No high severity vulnerabilities were found.

High

Medium

No Medium severity vulnerabilities were found.

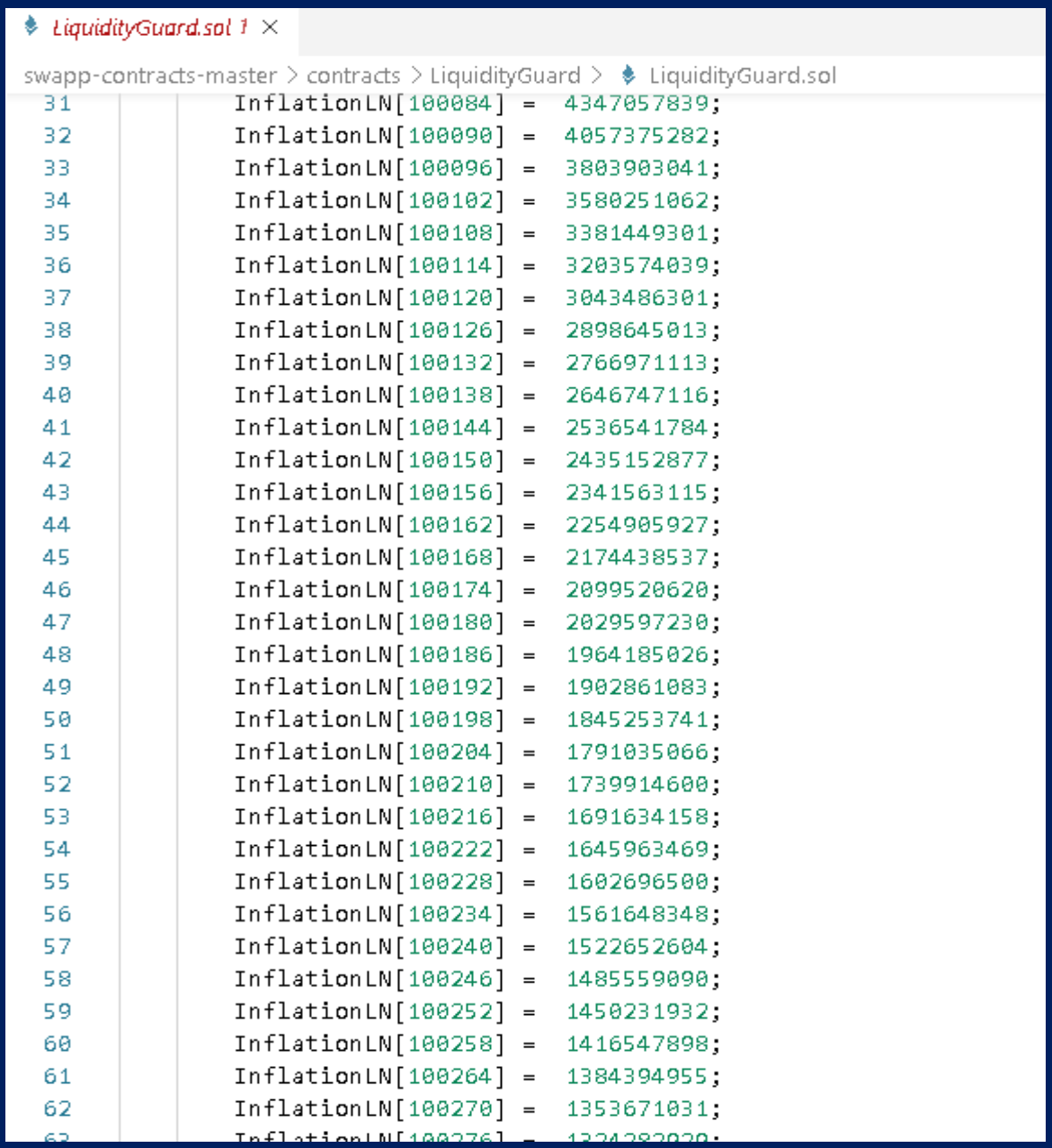
Low

No Low severity vulnerabilities were found.

Very Low

File: LiquidityGuard.sol

In constructor it is recommended to avoid too many updates of variables which may fail if any blockchain core proposal is applied on the system.



It can be split into parts as an ontime update via function.

File: ProvableAPI_0.6

Use of safeMemoryCleaner() is recommended to be used even if the compiler version is upgraded to avoid any invalid data (overlap) while reading memory using assembly code in line 1027 and 1039.

Discussion

Remember to verify/check all hard coded addresses across all contracts just before deployment.

Conclusion

We were given a contract file. And we have used all possible tests based on the given object. The contract is written systematically but comments were missing. We found no critical, high and medium level issue So it is good to go for production.

Security state of reviewed contract is “secured”.

Our Methodology

We like to work with a transparent process and make our reviews a collaborative effort. The goals of our security audits are to improve the quality of systems we review and aim for sufficient remediation to help protect users. The following is the methodology we use in our security audit process.

Manual Code Review:

In manually reviewing all of the code, we look for any potential issues with code logic, error handling, protocol and header parsing, cryptographic errors, and random number generators. We also watch for areas where more defensive programming could reduce the risk of future mistakes and speed up future audits. Although our primary focus is on the in-scope code, we examine dependency code and behavior when it is relevant to a particular line of investigation.

Vulnerability Analysis:

Our audit techniques included manual code analysis, user interface interaction, and whitebox penetration testing. We look at the project's web site to get a high level understanding of what functionality the software under review provides. We then meet with the developers to gain an appreciation of their vision of the software. We install and use the relevant software, exploring the user interactions and roles. While we do this, we brainstorm threat models and attack surfaces. We read design documentation, review other audit results, search for similar projects, examine source code dependencies, skim open issue tickets, and generally investigate details other than the implementation.

Documenting Results:

We follow a conservative, transparent process for analyzing potential security vulnerabilities and seeing them through successful remediation. Whenever a potential issue is discovered, we immediately create an Issue entry for it in this document, even though we have not yet verified the feasibility and impact of the issue. This process is conservative because we document our suspicions early even if they are later shown to not represent exploitable vulnerabilities. We generally follow a process of first documenting the suspicion with unresolved questions, then confirming the issue through code analysis, live experimentation, or automated tests. Code analysis is the most tentative, and we strive to provide test code, log captures, or screenshots demonstrating our confirmation. After this we analyze the feasibility of an attack in a live system.

Suggested Solutions:

We search for immediate mitigations that live deployments can take, and finally we suggest the requirements for remediation engineering for future releases. The mitigation and remediation recommendations should be scrutinized by the developers and deployment engineers, and successful mitigation and remediation is an ongoing collaborative process after we deliver our report, and before the details are made public.

Disclaimers

RD Auditors 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).

Because the total number of test cases are unlimited, so 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 the blockchain platform. The platform, its programming language, and other software related to the smart contract can have their own vulnerabilities that can lead to hacks. Thus, the audit can’t guarantee explicit security of the audited smart contracts.

RD
AUDITORS