Gennix Smart Contract Security Audit Report Abstract

Customer: TheGennix
Website: https://www.gennix.io/
Platform: Binance Smart Chain
Language: Solidity
Date: September 9th, 2021

Claimed Smart Contracts Features

Claimed Feature Detail	Our Observation
File 1: AdminRole.sol ● The Adminrole owner can access functionality like : add new admin and remove admin.	**YES, This is valid.**
File 2: FarmingInterface.sol ● The FarmingInterface can access functions like: totalAllocPoint, poolsCount, rewardPerBlock, etc.	**YES, This is valid.**
File 3: FarmingLens.sol ● The FarmingLens owner can access functions like:getBalanceAll, getMetadataAll, etc. **File 4: FarmingStorage.sol** ● The FarmingStorage can store pool details, LP etc.	**YES, This is valid.** **YES, This is valid.**

File 5: FeeToken.sol

YES, This is valid.

- The FeeToken can set fees and treasury addresses.

File 6: GennixToken_old.sol

YES, This is valid.

- Name: Gennix Token
- Symbol: GNNX
- Decimals: 8

File 7: LPToken.sol

YES, This is valid.

- Name: Pancake LPs
- Symbol: Cake-LP
- Decimals: 12

File 8: MintFarming.sol

YES, This is valid.

- MintFarming can add a new LP, Update the given pool's Reward allocation point, etc.

File 9: Whitelist.sol

YES, This is valid.

- The Whitelist owner can add and remove wallet addresses from whitelist.

File 10: Stacking.sol

YES, This is valid.

- Stage Interval: 10 minutes

File 11: StackingErc20.sol

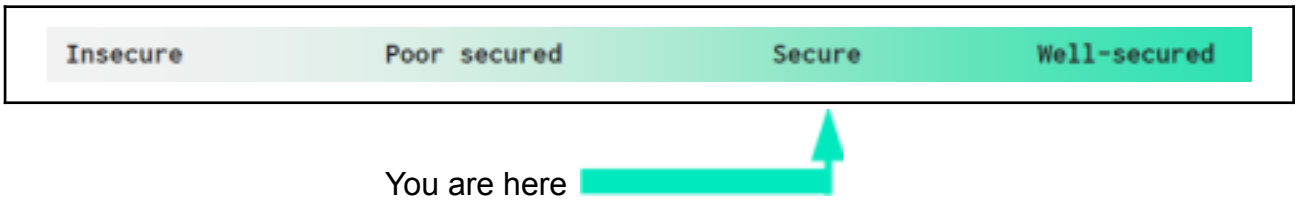
YES, This is valid.

- The StackingErc20 can access functions like: transfer, transferFrom.

File 12: StackingInterface.sol **YES, This is valid.** ● The StackingInterInterface file is empty.
File 13: AdminRole.sol **YES, This is valid.** ● The Adminrole owner can access functionality like : add new admin and remove admin, check admin.
File 14: GennixToken.sol **YES, This is valid.** ● Name: Gennix Token ● Symbol: GNNX ● Decimals: 8

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are **"Secured"**. These contracts also have owner functions (described in the centralization section below), which does not make everything 100% decentralized. Thus, the owner must execute those smart contract functions as per the business plan.



We used various tools like MythX, Slither and Remix IDE. At the same time this finding is based on critical analysis of the manual audit.

All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the Audit overview section. General overview is presented in the AS-IS section and all identified issues can be found in the Audit overview section.

We found 0 critical, 0 high, 0 medium and 1 low and some very low-level issues.

Technical Quick Stats

Main Category	Subcategory	Result
Contract Programming	Solidity version not specified	Passed
	Solidity version too old	Moderated
	Integer overflow/underflow	Passed
	Function input parameters lack of check	Passed
	Function input parameters check bypass	Passed
	Function access control lacks management	Moderated
	Critical operation lacks event log	Passed
	Random number generation/use vulnerability Fallback function misuse	Passed Passed
	Race condition Logical vulnerability	Passed Passed
	Features claimed Other programming issues	Passed Passed
Code Specification	Function visibility not explicitly declared	Passed
	Var. storage location not explicitly declared	Passed
	Use keywords/functions to be deprecated	Passed
	Other code specification issues	Passed
Gas Optimization	“Out of Gas” Issue	Passed
	High consumption ‘for/while’ loop	Moderated
	High consumption ‘storage’ storage	Passed
	Assert() misuse	Passed
Business Risk	The maximum limit for mintage not set	Passed
	“Short Address” Attack	Passed
	“Double Spend” Attack	Passed

Overall Audit Result: **PASSED**

Code Quality

These audit scope have 24 smart contracts. These smart contracts also contain Libraries, Smart contracts inherits and Interfaces. These are compact and well written contracts.

The libraries in the TheGennix contracts are part of its logical algorithm. A library is a different type of smart contract that contains reusable code. Once deployed on the blockchain (only once), it is assigned a specific address and its properties / methods can be reused many times by other contracts in the TheGennix contracts.

The team has not provided scenario and unit test scripts, which would have helped to determine the integrity of the code in an automated way.

Some code parts are not well commented on smart contracts.

Documentation

We were given The Gennix smart contracts code in the form of a hash code. The details of that code are mentioned above in the table.

As mentioned above, some code parts are **not well** commented. So, it is difficult to quickly understand the programming flow as well as complex code logic. Comments are very helpful in understanding the overall architecture of the protocol.

Another source of information was its official website which provided rich information about the project architecture and tokenomics.

Use of Dependencies

As per our observation, the libraries are used in this smart contract infrastructure that are based on well-known industry standard open-source projects. And their core code blocks are written well.

Apart from libraries, its functions are used in external smart contract calls.

AS-IS overview

AdminRole.sol

Sl.	Functions	Type	Observation	Conclusion
1	constructor	internal	Passed	No Issue
2	onlyAdmin	mdifier	Passed	No Issue
3	isAdmin	read	Passed	No Issue
4	addAdmin	write	access only Admin	No Issue
5	renounceAdmin	write	Passed	No Issue
6	_addAdmin	internal	Passed	No Issue
7	_removeAdmin	internal	Passed	No Issue

FarmingInterface.sol

Sl.	Functions	Type	Observation	Conclusion
1	getPendingReward	read	Passed	No Issue
2	totalAllocPoint	external	Passed	No Issue
3	rewardPerBlock	external	Passed	No Issue
4	tokens	external	Passed	No Issue
5	poolsCount	external	Passed	No Issue

FarmingLens.sol

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	getBalanceOf	external	Passed	No Issue
3	getBalanceAll	external	Passed	No Issue
4	getMetadataOf	external	Passed	No Issue
5	getMetadataAll	external	Passed	No Issue
6	getLPMetadata	external	Passed	No Issue
7	getLPMetadataAll	external	Passed	No Issue

8	getTokens	external	Passed	No Issue

FarmingStorage.sol

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	transfer	write	Passed	No Issue
3	transferFrom	internal	Passed	No Issue
4	setFee	write	access only Admin	No Issue
5	setPendingTreasury	write	access only Admin	No Issue
6	acceptTreasury	write	Passed	No Issue

LPToken.sol

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue

MintFarming.sol

Sl.	Functions	Type Observation	Observation	Conclusion
1	init	external	Passed	No Issue
2	poolsCount	external Passed	Passed	No Issue
3	_requirePoolNotExists	internal Passed	Passed	No Issue
4	_registerPool	internal Passed	Passed	No Issue
5	addToken	write	Critical operation lacks event log	Refer audit findings section below
6	setAllocPoint	write Critical operation lacks event log	Critical operation lacks event log	Refer audit findings section below
7	setRewardAddress	write Critical operation lacks event log	Critical operation lacks event log	Refer audit findings section below

8	_updateStakingPool	internal	Infinite loop	Refer audit findings section below
9	getMultiplier	read	Passed	No Issue
10	getPendingReward	read	Passed	No Issue
11	massUpdatePools	write	Infinite loop	Refer audit findings section below
12	updatePool	write	Critical operation lacks event log	Refer audit findings section below
13	deposit	write	Passed	No Issue
14	withdraw	write	Passed	No Issue
15	emergencyWithdraw	write	Passed	No Issue
16	_reward	write	Passed	No Issue

Whitelist.sol

Sl.	Functions	Type	Observation	Conclusion
1	addWhitelist	write	access only Admin	No Issue
2	removeWhitelist	write	access only Admin	No Issue
3	isWhitelisted	internal	Passed	No Issue

Stacking.sol

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	deposit	external	Passed	No Issue
3	claim	external	Passed	No Issue
4	withdraw	external	Critical operation lacks event log	Refer audit findings section below
5	addReward	external	Critical operation lacks event log	Refer audit findings section below
6	rewardOf	read	Passed	No Issue
7	_innerClaimWithChecks	internal	Passed	No Issue
8	_innerAddReward	internal	Passed	No Issue
9	_innerDepositNoSend	internal	Passed	No Issue
10	_innerDeposit	internal	Passed	No Issue
11	_innerWithdrawNoSend	internal	Passed	No Issue

12	_innerWithdraw	internal	Passed	No Issue
13	_calculateReward	internal	Passed	No Issue
14	_innerClaim	internal	Passed	No Issue
15	getStateTs	write	Passed	No Issue
16	getCurrentStateTs	read	Passed	No Issue
17	_transferUnderlying	internal	Passed	No Issue
18	_transferUnderlyingFrom	internal	Passed	No Issue
19	_allowanceUnderlying	internal	Passed	No Issue

StackingErc20.sol

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	transfer	write	Passed	No Issue
3	transferFrom	write	Passed	No Issue

StackingStorage.sol

Sl.	Functions	Type	Observation	Conclusion
1	totalSupply	read	Passed	No Issue
2	balanceOf	read	Passed	No Issue
3	transfer	write	Passed	No Issue
4	allowance	read	Passed	No Issue
5	approve	write	Passed	No Issue
6	transferFrom	write	Passed	No Issue
7	increaseAllowance	write	Passed	No Issue
8	decreaseAllowance	write	Passed	No Issue
9	_transfer	internal	Passed	No Issue
10	_mint	internal	Passed	No Issue
11	_burn	internal	Passed	No Issue
12	_approve	internal	Passed	No Issue
13	_burnFrom	internal	Passed	No Issue

ERC20Burnable.sol

Sl.	Functions	Type	Observation	Conclusion
1	burn	write	Passed	No Issue
2	burnFrom	write	Passed	No Issue

ERC20Detailed.sol

Sl.	Functions	Type	Observation	Conclusion
1	name	read	Passed	No Issue
2	symbol	read	Passed	No Issue
3	decimals	read	Passed	No Issue

ERC20Mintable.sol

Sl.	Functions	Type	Observation	Conclusion
1	mint	write	access only Minter	No Issue

AdminRole.sol

Sl.	Functions	Type	Observation	Conclusion
1	onlyAdmin	modifier	Passed	No Issue
2	isAdmin	read	Passed	No Issue
3	addAdmin	write access only Admin	Passed	No Issue
4	renounceAdmin	write	Passed	No Issue
5	_addAdmin	internal	Passed	No Issue
6	_removeAdmin	internal	Passed	No Issue

GennixToken.sol

Sl.	Functions	Type	Observation	Conclusion
1	init	write	Passed	No Issue

MinterRole.sol

Sl.	Functions	Type	Observation	Conclusion
1	onlyMinter	modifier	Passed	No Issue
2	isMinter	read	Passed	No Issue
3	addMinter	write access only Admin	Passed	No Issue
4	renounceMinter	write	Passed	No Issue
5	_addMinter	internal	Passed	No Issue
6	_removeMinter	internal	Passed	No Issue

Ownable.sol

Sl.	Functions	Type	Observation	Conclusion

1	constructor	write	Passed	No Issue
2	owner	read	Passed	No Issue
3	onlyOwner	modifier	Passed	No Issue
4	isOwner	read	Passed	No Issue
5	renounceOwnership	write	access only Owner	No Issue
6	transferOwnership	write	access only Owner	No Issue
7	_transferOwnership	internal	Passed	No Issue

Token.sol

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	owner	write	Passed	No Issue
3	onlyOwner	internal	Passed	No Issue
4	isOwner	write	access only Admin	No Issue
5	setPendingTreasury	write	access only Admin	No Issue
6	acceptTreasury	write	Passed	No Issue

TokenStorage.sol

Sl.	Functions	Type	Observation	Conclusion
1	addWhitelist	write	access only Admin	No Issue
2	removeWhitelist	write	access only Admin	No Issue
3	isWhitelisted	internal	Passed	No Issue

Audit Findings

Critical

No Critical severity vulnerabilities were found.

High

No High severity vulnerabilities were found.

Medium

No Medium severity vulnerabilities were found.

Low

(1) Infinite loop - (MintFarming.sol, FarmingLens.sol)

```
function _updateStakingPool() internal {
    uint256 length = tokens.length;
    uint256 points = 0;
    for (uint256 pid = 1; pid < length; ++pid) {
        points = points.add(poolInfo[address(tokens[pid])].allocPoint);
    }
    if (points != 0) {
        totalAllocPoint = totalAllocPoint.sub(poolInfo[address(tokens[0])].allocPoint).add(points);
        poolInfo[address(tokens[0])].allocPoint = uint128(points);
    }
}
```

```
// Update reward variables for all pools. Be careful of g
function massUpdatePools() public {
    uint256 length = tokens.length;
    for (uint256 pid = 0; pid < length; ++pid) {
        updatePool(tokens[pid]);
    }
}
```

There are several places in the smart contracts, where array.length is used directly in the loop. It is recommended to put some kind of limits.
Resolution: Adjust logic to replace loops with mapping or other code structures. **Status:** Open

Very Low / Discussion / Best practices:

(1) Use the latest solidity version - (All contracts)

```
pragma solidity ^0.5.16;
```

Using the latest solidity will prevent any compiler-level bugs.
Resolution: Please use 0.8.7 which is the latest version.

Status: Open

(2) All functions which are not called internally, must be declared as external. It is more efficient as sometimes it saves some gas.
https://ethereum.stackexchange.com/questions/19380/external-vs-public-best-practices
Status: Open

(3) Critical operation lacks event log

There are several places in the smart contracts, were not added a critical function call event log.
Resolution: Below functions should log events.

MintFarming.sol - addToken, setAllocPoint, setRewardAddress,
updatePool Stacking.sol - withdraw, addReward
Status: Open

Centralization

These smart contracts have some functions which can be executed by Admin (Owner) only. If the admin wallet private key would be compromised, then it would create trouble. Following are Admin functions:

- addToken: The MintFarming owner can add a new token.
- setAllocPoint: The MintFarming owner can set allocated points.
- setRewardAddress: The MintFarming owner can set reward wallet addresses. ● addWhitelist: The Whitelist Owner can add wallet addresses in white list. ● removeWhitelist: The Whitelist Owner can remove wallet addresses from whitelist. ● mint: The ERC20Mintable owner can create `amount` tokens and assign them to `account`, increasing the total supply.
- addAdmin: The AdminRole admin can add a new admin owner address.
- addMinter: The MinterRole admin can add minter.
- renounceOwnership: The Ownable owner can renounce ownership.
- transferOwnership: The Ownable owner can transfer ownership.
- setFee: The Token owner can set a fee.
- setPendingTreasury: The Token owner can set the pending Treasury.

Conclusion

We were given a contract code. And we have used all possible tests based on given objects as files. We observed some issues, but they are not critical. So, **it's good to go to production**.

Since possible test cases can be unlimited for such smart contracts protocol, we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan everything.

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools. Smart Contract's high level description of functionality was presented in As-is overview section of the report.

Audit report contains all found security vulnerabilities and other issues in the reviewed code.
Security state of the reviewed contract, based on standard audit procedure scope, is **"Secured".**

Gennix Smart Contract Security Audit Report Abstract

Customer: TheGennix
Website: https://www.gennix.io/
Platform: Binance Smart Chain
Language: Solidity
Date: September 9th, 2021

Claimed Smart Contracts Features

Claimed Feature Detail	Our Observation
File 1: GgovernorBravoDelegate.sol ● Name: Compound Governor Bravo ● Minimum proposal threshold : 50,000 Comp ● Maximum proposal threshold : 100,000 Comp ● Minimum voting period : 24 hours ● Max voting period : 2 weeks ● Min voting delay : 1 second ● Max voting delay : 1 week ● QuorumVotes : 4% of Comp ● Max number of actions included in a proposal : 10	**YES, This is valid.**
File 2: GovernorAlpha.sol ● Name : Compound Governor Alpha	**YES, This is valid.**
File 3: Comp.sol ● Name : Compound ● Symbol : COMP ● Decimals : 18 ● Tokens in circulation : 10 million Comp	**YES, This is valid.**

File 4: BaseJumpRateModelV2.sol

YES, This is valid.

- Blocks Per Year : 2102400

File 5: Comptroller.sol

YES, This is valid.

- Initial COMP index for a market : 1 Quintillion
- closeFactorMinMantissa : 0.05
- closeFactorMaxMantissa : 0.9
- collateralFactorMaxMantissa : 0.9

File 6: ComptrollerG1.sol

YES, This is valid.

- closeFactorMinMantissa : 0.05
- closeFactorMaxMantissa : 0.9
- collateralFactorMaxMantissa : 0.9
- liquidationIncentiveMinMantissa : 1
- liquidationIncentiveMaxMantissa : 1.5

File 7: ComptrollerG2.sol

YES, This is valid.

- closeFactorMinMantissa : 0.05
- closeFactorMaxMantissa : 0.9
- collateralFactorMaxMantissa : 0.9
- liquidationIncentiveMinMantissa : 1
- liquidationIncentiveMaxMantissa : 1.5

File 8: ComptrollerG3.sol

YES, This is valid.

- compClaimThreshold : 0.001
- Initial COMP index for a market : 1 Quintillion
- closeFactorMinMantissa : 0.05
- closeFactorMaxMantissa : 0.9
- collateralFactorMaxMantissa : 0.9
- liquidationIncentiveMinMantissa : 1
- liquidationIncentiveMaxMantissa : 1.5

File 9: ComptrollerG4.sol

YES, This is valid.

- compClaimThreshold : 0.001
- Initial COMP index for a market : 1 Quintillion
- closeFactorMinMantissa : 0.05
- closeFactorMaxMantissa : 0.9
- collateralFactorMaxMantissa : 0.9
- liquidationIncentiveMinMantissa : 1
- liquidationIncentiveMaxMantissa : 1.5

File 10: ComptrollerG5.sol

YES, This is valid.

- compClaimThreshold : 0.001
- Initial COMP index for a market : 1 Quintillion
- closeFactorMinMantissa : 0.05
- closeFactorMaxMantissa : 0.9
- collateralFactorMaxMantissa : 0.9
- liquidationIncentiveMinMantissa : 1
- liquidationIncentiveMaxMantissa : 1.5

File 11: ComptrollerG6.sol

YES, This is valid.

- compClaimThreshold : 0.001
- Initial COMP index for a market : 1 Quintillion
- closeFactorMinMantissa : 0.05
- closeFactorMaxMantissa : 0.9
- collateralFactorMaxMantissa : 0.9

File 12: ExponentialNoError.sol

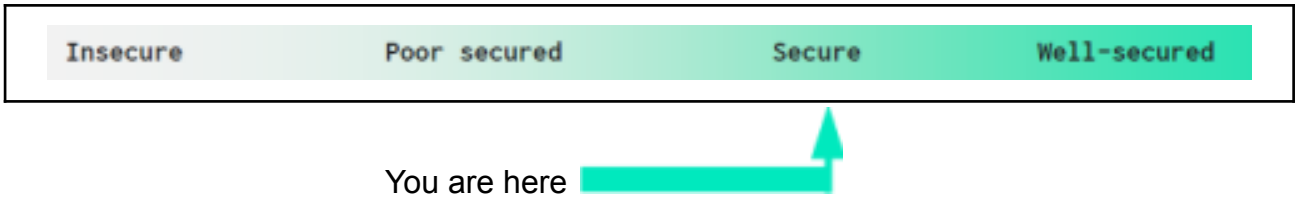
YES, This is valid.

- expScale : 1
- doubleScale : 1 Quintillion
- halfExpScale : 0.5
- mantissaOne : 1

File 13: JumpRateModel.sol ● Blocks Per Year : 2102400	**YES, This is valid.**
File 14: Timelock.sol ● Grace Period: 14 days ● Minimum Delay: 2 days ● Maximum Delay: 30 days	**YES, This is valid.**
File 15 : WhitePaperInterestRateModel.sol ● Blocks Per Year : 2102400	**YES, This is valid.**

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are **"Secured"**. These contracts also have owner functions (described in the centralization section below), which does not make everything 100% decentralized. Thus, the owner must execute those smart contract functions as per the business plan.



We used various tools like MythX, Slither and Remix IDE. At the same time this finding is based on critical analysis of the manual audit.
All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the Audit overview section. General overview is presented in the AS-IS section and all identified issues can be found in the Audit overview section.

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

Technical Quick Stats

Main Category	Subcategory	Result
Contract Programming	Solidity version not specified	Passed
	Solidity version too old	Moderated
	Integer overflow/underflow	Passed
	Function input parameters lack of check	Passed
	Function input parameters check bypass	Passed
	Function access control lacks management	Passed
	Critical operation lacks event log	Passed
	Random number generation/use vulnerability Fallback function misuse	Passed Passed
	Race condition Logical vulnerability	Passed Passed
	Features claimed Other programming issues	Passed Passed
Code Specification	Function visibility not explicitly declared	Passed
	Var. storage location not explicitly declared	Passed
	Use keywords/functions to be deprecated	Passed
	Other code specification issues	Passed
Gas Optimization	“Out of Gas” Issue	Passed
	High consumption ‘for/while’ loop	Passed
	High consumption ‘storage’ storage	Passed
	Assert() misuse	Passed

Business Risk	The maximum limit for mintage not set	Passed
	“Short Address” Attack	Passed
	“Double Spend” Attack	Passed

Overall Audit Result: **PASSED**

Code Quality

These audit scope have 44 smart contracts. These smart contracts also contain Libraries, Smart contracts inherits and Interfaces. These are compact and well written contracts.

The libraries in the TheGennix contracts are part of its logical algorithm. A library is a different type of smart contract that contains reusable code. Once deployed on the blockchain (only once), it is assigned a specific address and its properties / methods can be reused many times by other contracts in the TheGennix contracts.

The team has not provided scenario and unit test scripts, which would have helped to determine the integrity of the code in an automated way.

Some code parts are not well commented on smart contracts.

Documentation

We were given The Gennix smart contracts code in the form of a hash code. The details of that code are mentioned above in the table.

As mentioned above, some code parts are **not well** commented. So, it is difficult to quickly understand the programming flow as well as complex code logic. Comments are very helpful in understanding the overall architecture of the protocol.

Another source of information was its official website which provided rich information about the project architecture and tokenomics.

Use of Dependencies

As per our observation, the libraries are used in this smart contract infrastructure that are based on well-known industry standard open-source projects. And their core code blocks are written well.

Apart from libraries, its functions are used in external smart contract calls.

AS-IS overview

GovernorBravoInterfaces.sol

Sl.	Functions	Type	Observation	Conclusion
1	constructor	read	Passed	No Issue
2	_setImplementation	write	Passed	No Issue
3	delegateTo	internal	Passed	No Issue
4	queueOrRevertInternal	internal	Passed	No Issue
5	execute	external	Passed	No Issue
6	cancel	external	Passed	No Issue
7	getActions	external	Passed	No Issue
8	getReceipt	external	Passed	No Issue
9	state	read	Passed	No Issue
10	castVote	external	Passed	No Issue
11	castVoteWithReason	external	Passed	No Issue

12	castVoteBySig	external	Passed	No Issue
13	castVoteInternal	internal	Passed	No Issue
14	_setVotingDelay	external	Passed	No Issue
15 **16**	_setVotingPeriod _setProposalThreshold	external external	Passed Passed	No Issue No Issue
17 **18**	_initiate _setPendingAdmin	external external	Passed Passed	No Issue No Issue
19	_acceptAdmin	external	Passed	No Issue
20	add256	internal	Passed	No Issue
21	sub256	internal	Passed	No Issue
22	getChainIdInternal	internal	Passed	No Issue

FarmingInterface.sol

Sl.	Functions	Type	Observation	Conclusion
1	quorumVotes	write	Passed	No Issue
2	proposalThreshold	write	Passed	No Issue
3	proposalMaxOperations	write	Passed	No Issue
4	votingDelay	write	Passed	No Issue
5	votingPeriod	write	Passed	No Issue
6	propose	write	Passed	No Issue
7	queue	write	Passed	No Issue

8	_queueOrRevert	internal	Passed	No Issue
9	execute	write	Passed	No Issue
10	cancel	write	Passed	No Issue
11	getActions	read	Passed	No Issue
12	getReceipt	read	Passed	No Issue
13	state	read	Passed	No Issue
14	castVote	write	Passed	No Issue
15	castVoteBySig	write	Passed	No Issue
16	_castVote	internal	Passed	No Issue
17	__acceptAdmin	write	Passed	No Issue
18	__abdicate	write	Passed	No Issue
19	__queueSetTimelockPendingAdmin	write	Passed	No Issue
20	__executeSetTimelockPendingAdmin	write	Passed	No Issue
21	add256	internal	Passed	No Issue
22	sub256	internal	Passed	No Issue
23	getChainId	internal	Passed	No Issue

Comp.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	allowance	external Passed	No Issue
3	approve	external Passed	No Issue
4	balanceOf	external Passed	No Issue
5	transfer	external Passed	No Issue
6	transferFrom	external Passed	No Issue
7	delegate	write Passed	No Issue
8	delegateBySig	write Passed	No Issue
9	getCurrentVotes	external Passed	No Issue
10	getPriorVotes	read Passed	No Issue
11	_delegate	internal Passed	No Issue
12	_transferTokens	internal Passed	No Issue
13	_moveDelegates	internal Passed	No Issue
14	_writeCheckpoint	internal Passed	No Issue
15	safe32	internal Passed	No Issue
16	safe96	internal Passed	No Issue
17	add96	internal Passed	No Issue

18	sub96	internal Passed	No Issue
19	getChainId	internal Passed	No Issue

WhitePaperInterestRateModel.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	utilizationRate	write Passed	No Issue
3	getBorrowRate	read Passed	No Issue
4	getSupplyRate	read Passed	No Issue

CompoundLens.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	cTokenMetadata	write Passed	No Issue
3	cTokenMetadataAll	write Passed	No Issue
4	cTokenBalances	write Passed	No Issue
5	cTokenBalancesAll	write Passed	No Issue
6	cTokenUnderlyingPrice	write Passed	No Issue
7	cTokenUnderlyingPriceAll	external Passed	No Issue
8	getAccountLimits	write Passed	No Issue
9	getGovReceipts	read Passed	No Issue
10	getGovBravoReceipts	read Passed	No Issue
11	setProposal	internal Passed	No Issue
12	getGovProposals	external Passed	No Issue
13	setBravoProposal	internal Passed	No Issue
14	getGovBravoProposals	external Passed	No Issue
15	getCompBalanceMetad ata	external Passed	No Issue
16	getCompBalanceMetad ataExt	external Passed	No Issue

17	getCompVotes	external Passed	No Issue
18	compareStrings	internal Passed	No Issue
19	add	internal Passed	No Issue
20	sub	internal Passed	No Issue

BaseJumpRateModelV2.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	updateJumpRateModel	external Passed	No Issue
3	utilizationRate	write Passed	No Issue
4	getBorrowRateInternal	internal Passed	No Issue
5	getSupplyRate	read Passed	No Issue
6	updateJumpRateModelI nternal	internal Passed	No Issue

CarefulMath.sol

(1) Functions

Sl.	Functions	Type Observation	Conclusion
1	mulUInt	internal Passed	No Issue
2	divUInt	internal Passed	No Issue
3	subUInt	internal Passed	No Issue
4	addUInt	internal Passed	No Issue
5	addThenSubUInt	internal Passed	No Issue

CCompLikeDelegate.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	_delegateCompLikeTo	external Passed	No Issue

CDaiDelegate.sol

Sl.	Functions	Type Observation	Conclusion
1	_becomeImplementatio n	write Passed	No Issue
2	_becomeImplementatio n	internal Passed	No Issue
3	_resignImplementation	write Passed	No Issue
4	accrueInterest	write Passed	No Issue
5	getCashPrior	internal Passed	No Issue
6	doTransferIn	internal Passed	No Issue
7	doTransferOut	internal Passed	No Issue
8	add	internal Passed	No Issue
9	mul	internal Passed	No Issue

CErc20.sol

Sl.	Functions	Type Observation	Conclusion
1	initialize	write Passed	No Issue
2	mint	external Passed	No Issue
3	redeem	external Passed	No Issue
4	redeemUnderlying	external Passed	No Issue
5	borrow	external Passed	No Issue
6	repayBorrow	external Passed	No Issue
7	repayBorrowBehalf	external Passed	No Issue
8	liquidateBorrow	external Passed	No Issue
9	sweepToken	external Passed	No Issue
10	_addReserves	external Passed	No Issue
11	getCashPrior	internal Passed	No Issue

12	doTransferIn	internal Passed	No Issue
13	doTransferOut	internal Passed	No Issue

CErc20Delegate.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	_becomeImplementatio n	write Passed	No Issue
3	_resignImplementation	write Passed	No Issue

CErc20Delegator.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	_setImplementation	write Passed	No Issue
3	mint	external Passed	No Issue
4	redeem	external Passed	No Issue
5	redeemUnderlying	external Passed	No Issue
6	borrow	external Passed	No Issue
7	repayBorrow	external Passed	No Issue
8	repayBorrowBehalf	external Passed	No Issue
9	liquidateBorrow	external Passed	No Issue
10	transfer	external Passed	No Issue
11	transferFrom	external Passed	No Issue
12	approve	external Passed	No Issue

13	allowance	external Passed	No Issue
14	balanceOf	external Passed	No Issue
15	balanceOfUnderlying	external Passed	No Issue
16	getAccountSnapshot	external Passed	No Issue
17	borrowRatePerBlock	external Passed	No Issue
18	supplyRatePerBlock	external Passed	No Issue
19	totalBorrowsCurrent	external Passed	No Issue
20	borrowBalanceCurrent	external Passed	No Issue
21	borrowBalanceStored	read Passed	No Issue
22	exchangeRateCurrent	write Passed	No Issue
23	exchangeRateStored	read Passed	No Issue
24	getCash	external Passed	No Issue
25	accrueInterest	write Passed	No Issue
26	seize	external Passed	No Issue
27	sweepToken	external Passed	No Issue
28	_setPendingAdmin	external Passed	No Issue
29	_setComptroller	write Passed	No Issue
30	_setReserveFactor	external Passed	No Issue

31	_acceptAdmin	external Passed	No Issue
32	_addReserves	external Passed	No Issue
33	_reduceReserves	external Passed	No Issue
34	_setInterestRateModel	write Passed	No Issue
35	delegateTo	internal Passed	No Issue
36	delegateToImplementati on	write Passed	No Issue
37	delegateToViewImplem entation	read Passed	No Issue

CErc20Immutable.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	_becomeImplementatio n	write Passed	No Issue
3	_resignImplementation	write Passed	No Issue

CErc20Immutable.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue

CEther.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	mint	external Passed	No Issue

3	redeem	external Passed	No Issue
4	redeemUnderlying	external Passed	No Issue
5	borrow	external Passed	No Issue
6	repayBorrow	external Passed	No Issue
7	repayBorrowBehalf	external Passed	No Issue
8	liquidateBorrow	external Passed	No Issue
9	getCashPrior	internal Passed	No Issue
10	doTransferIn	internal Passed	No Issue
11	doTransferOut	internal Passed	No Issue
12	requireNoError	internal Passed	No Issue

Comptroller.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	checkMembership	external Passed	No Issue
3	getAssetsIn	external Passed	No Issue
4	enterMarkets	write Passed	No Issue
5	addToMarketInternal	internal Passed	No Issue
6	exitMarket	external Passed	No Issue
7	mintAllowed	external Passed	No Issue
8 **9**	mintVerify redeemAllowed	external Passed external Passed	No Issue No Issue
10 **11**	redeemAllowedInternal redeemVerify	internal Passed external Passed	No Issue No Issue

12 **13**	borrowAllowed borrowVerify	external Passed external Passed	No Issue No Issue
14 **15**	repayBorrowAllowed repayBorrowVerify	external Passed external Passed	No Issue No Issue
16	liquidateBorrowAllowed	external Passed	No Issue
17	liquidateBorrowVerify	external Passed	No Issue
18	seizeAllowed	external Passed	No Issue
19	seizeVerify	external Passed	No Issue
20	transferAllowed	external Passed	No Issue
21	transferVerify	external Passed	No Issue
22	getAccountLiquidity	write Passed	No Issue
23	getAccountLiquidityInter nal	internal Passed	No Issue
24	getHypotheticalAccount Liquidity	write Passed	No Issue
25	getHypotheticalAccount LiquidityInternal	internal Passed	No Issue
26	liquidateCalculateSeize Tokens	external Passed	No Issue
27	_setPriceOracle	write Passed	No Issue
28	_setCloseFactor	external Passed	No Issue

29 **30**	_setCollateralFactor _setLiquidationIncentive	external Passed external Passed	No Issue No Issue
31 **32**	_supportMarket _addMarketInternal	external Passed internal Passed	No Issue No Issue
33 **34**	_setMarketBorrowCaps _setBorrowCapGuardia n	external Passed external Passed	No Issue No Issue
35	_setPauseGuardian	write Passed	No Issue
36	_setMintPaused	write Passed	No Issue
37	_setBorrowPaused	write Passed	No Issue
38	_setTransferPaused	write Passed	No Issue
39	_setSeizePaused	write Passed	No Issue
40	_become	write Passed	No Issue
41	adminOrInitializing	internal Passed	No Issue
42	setCompSpeedInternal	internal Passed	No Issue
43	updateCompSupplyInde x	internal Passed	No Issue
44	updateCompBorrowInd ex	internal Passed	No Issue
45	distributeSupplierComp	internal Passed	No Issue
46	distributeBorrowerCom p	internal Passed	No Issue

47	updateContributorRewards	write Passed	No Issue
48	claimComp	write Passed	No Issue
49	claimComp	write Passed	No Issue
50	claimComp	write Passed	No Issue
51	grantCompInternal	Internal Passed	No Issue
52	_grantComp	write Passed	No Issue
53	_setCompSpeed	write Passed	No Issue
54	_setContributorCompSpeed	write Passed	No Issue
55	getAllMarkets	read Passed	No Issue
56	getBlockNumber	read Passed	No Issue
57	getCompAddress	read Passed	No Issue

ComptrollerG1.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	getAssetsIn	external Passed	No Issue
3	checkMembership	external Passed	No Issue
4	enterMarkets	write Passed	No Issue
5	exitMarket	external Passed	No Issue
6	mintAllowed	external Passed	No Issue

7	mintVerify	external Passed	No Issue
8	redeemAllowed	external Passed	No Issue
9 **10**	redeemAllowedInternal redeemVerify	internal Passed external Passed	No Issue No Issue
11 **12**	borrowAllowed borrowVerify	external Passed external Passed	No Issue No Issue
13 **14**	repayBorrowAllowed repayBorrowVerify	external Passed external Passed	No Issue No Issue
15 **16**	liquidateBorrowAllowed liquidateBorrowVerify	external Passed external Passed	No Issue No Issue
17	seizeAllowed	external Passed	No Issue
18	seizeVerify	external Passed	No Issue
19	transferAllowed	external Passed	No Issue
20	transferVerify	external Passed	No Issue
21	getAccountLiquidity	read Passed	No Issue
22	getAccountLiquidityInter nal	internal Passed	No Issue
23	getHypotheticalAccount LiquidityInternal	internal Passed	No Issue

24	liquidateCalculateSeize Tokens	external Passed	No Issue
25	_setPriceOracle	write Passed	No Issue
26	_setCloseFactor	external Passed	No Issue
27	_setCollateralFactor	external Passed	No Issue
28	_setMaxAssets	external Passed	No Issue
29	_setLiquidationIncentive	external Passed	No Issue
30	_supportMarket	external Passed	No Issue
31	_become	write Passed	No Issue
32	adminOrInitializing	internal Passed	No Issue

ComptrollerG2.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	getAssetsIn	external Passed	No Issue
3	checkMembership	external Passed	No Issue
4	enterMarkets	write Passed	No Issue
5	addToMarketInternal	internal Passed	No Issue

6	exitMarket	external Passed	No Issue
7	mintAllowed	external Passed	No Issue
8 **9**	mintVerify redeemAllowed	external Passed external Passed	No Issue No Issue
10 **11**	redeemAllowedInternal redeemVerify	internal Passed external Passed	No Issue No Issue
12 **13**	borrowAllowed borrowVerify	external Passed external Passed	No Issue No Issue
14 **15**	repayBorrowAllowed repayBorrowVerify	external Passed external Passed	No Issue No Issue
16	liquidateBorrowAllowed	external Passed	No Issue
17	liquidateBorrowVerify	external Passed	No Issue
18	seizeAllowed	external Passed	No Issue
19	seizeVerify	external Passed	No Issue
20	transferAllowed	external Passed	No Issue
21	transferVerify	external Passed	No Issue
22	getAccountLiquidity	read Passed	No Issue

23	getAccountLiquidityInter nal	internal Passed	No Issue
24	getHypotheticalAccount Liquidity	write Passed	No Issue
25	getHypotheticalAccount LiquidityInternal	internal Passed	No Issue
26	liquidateCalculateSeize Tokens	external Passed	No Issue
27	_setPriceOracle	write Passed	No Issue
28	_setCloseFactor	external Passed	No Issue
29	_setCollateralFactor	external Passed	No Issue
30	_setMaxAssets	external Passed	No Issue
31	_setLiquidationIncentive	external Passed	No Issue
32	_supportMarket	external Passed	No Issue
33	_setPauseGuardian	write Passed	No Issue
34	_setMintPaused	write Passed	No Issue
35 **36**	_setBorrowPaused _setTransferPaused	write Passed write Passed	No Issue No Issue
37 **38**	_setSeizePaused	write Passed	No Issue

	_become	write Passed	No Issue

ComptrollerG3.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	getAssetsIn	external Passed	No Issue
3	checkMembership	external Passed	No Issue
4	enterMarkets	write Passed	No Issue
5	addToMarketInternal	internal Passed	No Issue
6	exitMarket	external Passed	No Issue
7	mintAllowed	external Passed	No Issue
8	mintVerify	external Passed	No Issue
9	redeemAllowed	external Passed	No Issue
10	redeemAllowedInternal	internal Passed	No Issue
11	redeemVerify	external Passed	No Issue
12	borrowAllowed	external Passed	No Issue
13	borrowVerify	external Passed	No Issue
14	repayBorrowAllowed	external Passed	No Issue
15	repayBorrowVerify	external Passed	No Issue

16	liquidateBorrowAllowed	external Passed	No Issue
17	liquidateBorrowVerify	external Passed	No Issue
18	seizeAllowed	external Passed	No Issue
19	seizeVerify	external Passed	No Issue
20	transferAllowed	external Passed	No Issue
21	transferVerify	external Passed	No Issue
22	getAccountLiquidity	write Passed	No Issue
23	getAccountLiquidityInter nal	internal Passed	No Issue
24	getHypotheticalAccount Liquidity	write Passed	No Issue
25 **26** **27**	getHypotheticalAccount LiquidityInternal liquidateCalculateSeize Tokens _setPriceOracle	internal Passed external Passed write Passed	No Issue No Issue No Issue
28	_setCloseFactor	external Passed	No Issue
29	_setCollateralFactor	external Passed	No Issue
30	_setMaxAssets	external Passed	No Issue
31	_setLiquidationIncentive	external Passed	No Issue

32	_supportMarket	external Passed	No Issue
33	_addMarketInternal	internal Passed	No Issue
34	_setPauseGuardian	write Passed	No Issue
35	_setMintPaused	write Passed	No Issue
36	_setBorrowPaused	write Passed	No Issue
37	_setTransferPaused	write Passed	No Issue
38	_setSeizePaused	write Passed	No Issue
39	_become	write Passed	No Issue
40	_becomeG3	write Passed	No Issue
41	adminOrInitializing	internal Passed	No Issue
42	refreshCompSpeeds	write Passed	No Issue
43	updateCompSupplyInde x	internal Passed	No Issue
44	updateCompBorrowInd ex	internal Passed	No Issue
45	distributeSupplierComp	internal Passed	No Issue
46	distributeBorrowerCom p	internal Passed	No Issue

47	transferComp	internal Passed	No Issue
48	claimComp	write Passed	No Issue
49	_setCompRate	write Passed	No Issue
50	_addCompMarkets	write Passed	No Issue
51	_addCompMarketIntern al	internal Passed	No Issue
52 **53**	_dropCompMarket getAllMarkets	write Passed read Passed	No Issue No Issue
54 **55**	getBlockNumber getCompAddress	read Passed read Passed	No Issue No Issue

ComptrollerG4.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	getAssetsIn	external Passed	No Issue
3	checkMembership	external Passed	No Issue
4	enterMarkets	write Passed	No Issue
5	addToMarketInternal	internal Passed	No Issue
6	exitMarket	external Passed	No Issue
7	mintAllowed	external Passed	No Issue
8	mintVerify	external Passed	No Issue

9	redeemAllowed	external Passed	No Issue
10	redeemAllowedInternal	internal Passed	No Issue
11	redeemVerify	external Passed	No Issue
12	borrowAllowed	external Passed	No Issue
13	borrowVerify	external Passed	No Issue
14	repayBorrowAllowed	external Passed	No Issue
15	repayBorrowVerify	external Passed	No Issue
16	liquidateBorrowAllowed	external Passed	No Issue
17	liquidateBorrowVerify	external Passed	No Issue
18	seizeAllowed	external Passed	No Issue
19	seizeVerify	external Passed	No Issue
20	transferAllowed	external Passed	No Issue
21	transferVerify	external Passed	No Issue
22	getAccountLiquidity	write Passed	No Issue
23	getAccountLiquidityInter nal	internal Passed	No Issue
24	getHypotheticalAccount Liquidity	write Passed	No Issue
25	getHypotheticalAccount	internal Passed	No Issue
26	LiquidityInternal liquidateCalculateSeize Tokens	external Passed	No Issue
27	_setPriceOracle	write Passed	No Issue

28	_setCloseFactor	external Passed	No Issue
29	_setCollateralFactor	external Passed	No Issue
30	_setMaxAssets	external Passed	No Issue
31	_setLiquidationIncentive	external Passed	No Issue
32	_supportMarket	external Passed	No Issue
33	_addMarketInternal	internal Passed	No Issue
34	_setPauseGuardian	write Passed	No Issue
35	_setMintPaused	write Passed	No Issue
36	_setBorrowPaused	write Passed	No Issue
37	_setTransferPaused	write Passed	No Issue
38	_setSeizePaused	write Passed	No Issue
39	_become	write Passed	No Issue
40	adminOrInitializing	internal Passed	No Issue
41	refreshCompSpeeds	write Passed	No Issue
42	refreshCompSpeedsInt ernal	internal Passed	No Issue
43	updateCompSupplyInde x	internal Passed	No Issue
44	updateCompBorrowInd ex	internal Passed	No Issue

45	distributeSupplierComp	internal Passed	No Issue
46	distributeBorrowerCom p	internal Passed	No Issue
47 **48**	transferComp claimComp	internal Passed write Passed	No Issue No Issue
49 **50**	_setCompRate _addCompMarkets	write Passed write Passed	No Issue No Issue
51 **52**	_addCompMarketIntern al _dropCompMarket	internal Passed write Passed	No Issue No Issue
53	getAllMarkets	read Passed	No Issue
54	getBlockNumber	read Passed	No Issue
55	getCompAddress	read Passed	No Issue

ComptrollerG5.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	getAssetsIn	external Passed	No Issue
3	checkMembership	external Passed	No Issue
4	enterMarkets	w Passed	No Issue
5	addToMarketInternal	internal Passed	No Issue
6	exitMarket	external Passed	No Issue

7	mintAllowed	external Passed	No Issue
8	mintVerify	external Passed	No Issue
9	redeemAllowed	external Passed	No Issue
10	redeemAllowedInternal	internal Passed	No Issue
11	redeemVerify	external Passed	No Issue
12	borrowAllowed	external Passed	No Issue
13	borrowVerify	external Passed	No Issue
14	repayBorrowAllowed	external Passed	No Issue
15	repayBorrowVerify	external Passed	No Issue
16 **17**	liquidateBorrowAllowed liquidateBorrowVerify	external Passed external Passed	No Issue No Issue
18 **19**	seizeAllowed seizeVerify	external Passed external Passed	No Issue No Issue
20 **21**	transferAllowed transferVerify	external Passed external Passed	No Issue No Issue
22	getAccountLiquidity	write Passed	No Issue
23	getAccountLiquidityInter nal	internal Passed	No Issue
24	getHypotheticalAccount Liquidity	write Passed	No Issue

25	getHypotheticalAccount LiquidityInternal	internal Passed	No Issue
26	liquidateCalculateSeize Tokens	external Passed	No Issue
27	_setPriceOracle	write Passed	No Issue
28	_setCloseFactor	external Passed	No Issue
29	_setCollateralFactor	external Passed	No Issue
30	_setMaxAssets	external Passed	No Issue
31	_setLiquidationIncentive	external Passed	No Issue
32	_supportMarket	external Passed	No Issue
33	_addMarketInternal	internal Passed	No Issue
34	_setMarketBorrowCaps	external Passed	No Issue
35	_setBorrowCapGuardia n	external Passed	No Issue
36	_setPauseGuardian	write Passed	No Issue
37	_setMintPaused	write Passed	No Issue
38	_setBorrowPaused	write Passed	No Issue
39	_setTransferPaused	write Passed	No Issue
40	_setSeizePaused	write Passed	No Issue
41	_become	write Passed	No Issue

42	adminOrInitializing	internal Passed	No Issue
43	refreshCompSpeeds	write Passed	No Issue
44	refreshCompSpeedsInt ernal	internal Passed	No Issue
45	updateCompSupplyInde xQ	internal Passed	No Issue
46	updateCompBorrowInd ex	internal Passed	No Issue
47	distributeSupplierComp	internal Passed	No Issue
48	distributeBorrowerCom p	internal Passed	No Issue
49	transferComp	internal Passed	No Issue
50	claimComp	write Passed	No Issue
51	_setCompRate	write Passed	No Issue
52	_addCompMarkets	write Passed	No Issue
53	_addCompMarketIntern al	internal Passed	No Issue
54	_dropCompMarket	write Passed	No Issue
55	getAllMarkets	read Passed	No Issue
56	getBlockNumber	read Passed	No Issue
57	getCompAddress	read Passed	No Issue

ComptrollerG6.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue

2	getAssetsIn	external Passed	No Issue
3	checkMembership	external Passed	No Issue
4	enterMarkets	write Passed	No Issue
5	addToMarketInternal	internal Passed	No Issue
6	exitMarket	external Passed	No Issue
7	mintAllowed	external Passed	No Issue
8 **9**	mintVerify redeemAllowed	external Passed external Passed	No Issue No Issue
10 **11**	redeemAllowedInternalredeem Verify	internal Passed external Passed	No Issue No Issue
12 **13**	borrowAllowed borrowVerify	external Passed external Passed	No Issue No Issue
14 **15**	repayBorrowAllowed repayBorrowVerify	external Passed external Passed	No Issue No Issue
16	liquidateBorrowAllowed	external Passed	No Issue
17	liquidateBorrowVerify	external Passed	No Issue
18	seizeAllowed	external Passed	No Issue
19	seizeVerify	external Passed	No Issue

20	transferAllowed	external Passed	No Issue
21	transferVerify	external Passed	No Issue
22	getAccountLiquidity	read Passed	No Issue
23	getAccountLiquidityInternal	internal Passed	No Issue
24	getHypotheticalAccountLiquidity	read Passed	No Issue
25	getHypotheticalAccountLiquidityInternal	internal Passed	No Issue
26	liquidateCalculateSeizeTokens	external Passed	No Issue
27	_setPriceOracle	write Passed	No Issue
28	_setCollateralFactor	external Passed	No Issue
29	_setLiquidationIncentive	external Passed	No Issue
30	_supportMarket	external Passed	No Issue
31	_addMarketInternal	internal Passed	No Issue
32 **33** **34**	_setMarketBorrowCaps _setBorrowCapGuardian _setPauseGuardian	external Passed external Passed write Passed	No Issue No Issue No Issue
35 **36**	_setMintPaused _setBorrowPaused	write Passed write Passed	No Issue No Issue

37	_setTransferPaused	write Passed	No Issue
38	_setSeizePaused	write Passed	No Issue
39	_become	write Passed	No Issue
40	adminOrInitializing	internal Passed	No Issue
41	refreshCompSpeeds	write Passed	No Issue
42	refreshCompSpeedsInternal	internal Passed	No Issue
43	updateCompSupplyIndex	internal Passed	No Issue
44	updateCompBorrowIndex	internal Passed	No Issue
45	distributeSupplierComp	internal Passed	No Issue
46	distributeBorrowerComp	internal Passed	No Issue
47	transferComp	internal Passed	No Issue
48	updateContributorRewards	write Passed	No Issue
49	claimComp	write Passed	No Issue
50	grantCompInternal	internal Passed	No Issue
51	_grantComp	write Passed	No Issue

52	_setContributorCompSpeed	write Passed	No Issue
53	_setCompRate	write Passed	No Issue
54	_addCompMarkets	write Passed	No Issue
55	_addCompMarketInternal	internal Passed	No Issue
56	_dropCompMarket	write Passed	No Issue
57	getAllMarkets	read Passed	No Issue
58	getBlockNumber	read Passed	No Issue
59	getCompAddress	read Passed	No Issue

ComptrollerInterface.sol

(1) Functions

Sl.	Functions	Type Observation	Conclusion
1	enterMarkets	external Passed	No Issue
2	exitMarket	external Passed	No Issue
3	mintAllowed	external Passed	No Issue
4	mintVerify	external Passed	No Issue
5	redeemAllowed	external Passed	No Issue

6	redeemVerify	external Passed	No Issue
7	borrowAllowed	external Passed	No Issue
8 **9**	borrowVerify repayBorrowAllowed	external Passed external Passed	No Issue No Issue
10 **11**	liquidateBorrowAllowed liquidateBorrowVerify	external Passed external Passed	No Issue No Issue
12 **13**	seizeAllowed seizeVerify	external Passed external Passed	No Issue No Issue
14	transferAllowed	external Passed	No Issue
15	transferVerify	external Passed	No Issue
16	liquidateCalculateSeize Tokens	external Passed	No Issue

ComptrollerStorage.sol

Sl.	Functions	Type Observation	Conclusion
1	initialize	write Passed	No Issue
2	transferTokens	write Passed	No Issue
3	transfer	external Passed	No Issue
4	transferFrom	external Passed	No Issue
5	approve	external Passed	No Issue
6	allowance	external Passed	No Issue
7	balanceOf	external Passed	No Issue

8 **9**	balanceOfUnderlying getAccountSnapshot	external Passed external Passed	No Issue No Issue
10 **11**	getBlockNumber borrowRatePerBlock	internal Passed external Passed	No Issue No Issue
12 **13**	supplyRatePerBlock totalBorrowsCurrent	external Passed external Passed	No Issue No Issue
14 **15**	borrowBalanceCurrent borrowBalanceStored	external Passed read Passed	No Issue No Issue
16	borrowBalanceStoredInt ernal	internal Passed	No Issue
17	exchangeRateCurrent	write Passed	No Issue
18	exchangeRateStored	read Passed	No Issue
19	exchangeRateStoredInt ernal	internal Passed	No Issue
20	getCash	external Passed	No Issue
21	accrueInterest	write Passed	No Issue
22	mintInternal	internal Passed	No Issue
23	mintFresh	internal Passed	No Issue
24	redeemInternal	internal Passed	No Issue

25	redeemUnderlyingIntern al	internal Passed	No Issue
26	redeemFresh	internal Passed	No Issue
27	borrowInternal	internal Passed	No Issue
28	borrowFresh	internal Passed	No Issue
29	repayBorrowInternal	internal Passed	No Issue
30	repayBorrowBehalfInter nal	internal Passed	No Issue
31	repayBorrowFresh	internal Passed	No Issue
32	liquidateBorrowInternal	internal Passed	No Issue
33	liquidateBorrowFresh	internal Passed	No Issue
34	seize	external Passed	No Issue
35	seizeInternal	internal Passed	No Issue
36	_setPendingAdmin	external Passed	No Issue
37	_acceptAdmin	external Passed	No Issue
38	_setComptroller	write Passed	No Issue
39	_setReserveFactor	external Passed	No Issue
40	_setReserveFactorFres h	internal Passed	No Issue
41	_addReservesInternal	internal Passed	No Issue
42	_addReservesFresh	internal Passed	No Issue

43	_reduceReserves	external Passed	No Issue
44	_reduceReservesFresh	internal Passed	No Issue
45	_setInterestRateModel	write Passed	No Issue
46	_setInterestRateModelF resh	internal Passed	No Issue
47	getCashPrior	internal Passed	No Issue
48	doTransferIn	internal Passed	No Issue
49	doTransferOut	internal Passed	No Issue
50	nonReentrant	modifier Passed	No Issue

CTokenInterfaces.sol

Sl.	Functions	Type Observation	Conclusion
1	transfer	external Passed	No Issue
2	transferFrom	external Passed	No Issue
3	approve	external Passed	No Issue
4	allowance	external Passed	No Issue
5	balanceOf	external Passed	No Issue
6	balanceOfUnderlying	external Passed	No Issue

7	getAccountSnapshot	external Passed	No Issue
8 **9**	borrowRatePerBlock supplyRatePerBlock	external Passed external Passed	No Issue No Issue
10 **11**	totalBorrowsCurrent borrowBalanceCurrent	external Passed external Passed	No Issue No Issue
12 **13**	borrowBalanceStored exchangeRateCurrent	read Passed write Passed	No Issue No Issue
14 **15**	exchangeRateStored getCash	read Passed external Passed	No Issue No Issue
16	accrueInterest	external Passed	No Issue
17	seize	external Passed	No Issue
18	_setPendingAdmin	external Passed	No Issue
19	_setReserveFactor	external Passed	No Issue
20	_acceptAdmin	external Passed	No Issue
21	_setComptroller	write Passed	No Issue
22	_reduceReserves	external Passed	No Issue
23	_setInterestRateModel	write Passed	No Issue

DAIInterestRateModelV3.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	updateJumpRateModel	external Passed	No Issue
3	getSupplyRate	read Passed	No Issue
4	dsrPerBlock	read Passed	No Issue
5	poke	write Passed	No Issue

ErrorReporterx.sol

Sl.	Functions	Type Observation	Conclusion
1	fail	internal Passed	No Issue
2	failOpaque	internal Passed	No Issue

Exponential.sol

Sl.	Functions	Type Observation	Conclusion
1	addExp	internal Passed	No Issue
2	getExp	internal Passed	No Issue
3	subExp	internal Passed	No Issue
4	mulScalar	internal Passed	No Issue
5	mulScalarTruncate	internal Passed	No Issue
6	mulScalarTruncateAdd UInt	internal Passed	No Issue
7	divScalar	internal Passed	No Issue
8	divScalarByExp	internal Passed	No Issue
9	divScalarByExpTruncate	internal Passed	No Issue
10	mulExp	internal Passed	No Issue

11	mulExp	internal Passed	No Issue
12	mulExp3	internal Passed	No Issue
13	divExp	internal Passed	No Issue

ExponentialNoError.sol

Sl.	Functions	Type Observation	Conclusion
1	truncate	internal Passed	No Issue
2	mul_ScalarTruncate	internal Passed	No Issue
3	mul_ScalarTruncateAdd UInt	internal Passed	No Issue
4	lessThanExp	internal Passed	No Issue
5	lessThanOrEqualExp	internal Passed	No Issue
6	greaterThanExp	internal Passed	No Issue
7	isZeroExp	internal Passed	No Issue
8	safe224	internal Passed	No Issue
9	Safe32	internal Passed	No Issue
10	add_	internal Passed	No Issue
11	sub_	internal Passed	No Issue
12	mul_	internal Passed	No Issue
13	div_	internal Passed	No Issue
14	fraction	internal Passed	No Issue

InterestRateModel.sol

(1) Functions

Sl.	Functions	Type Observation	Conclusion
1	getBorrowRate	external Passed	No Issue

2	getSupplyRate	external Passed	No Issue

JumpRateModel.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	read Passed	No Issue
2	utilizationRate	write Passed	No Issue
3	getBorrowRate	read Passed	No Issue
4	getSupplyRate	read Passed	No Issue

JumpRateModelV2.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	read Passed	No Issue
2	getBorrowRate	external Passed	No Issue

LegacyInterestRateModel.sol

Sl.	Functions	Type Observation	Conclusion
1	getBorrowRate	external Passed	No Issue
2	getSupplyRate	external Passed	No Issue

LegacyJumpRateModelV2.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	getBorrowRate	external Passed	No Issue

Maximillion.sol

Sl.	Functions	Type Observation	Conclusion
1	repayBehalfExplicit	write Passed	No Issue

2	repayBehalf	write Passed	No Issue
3	constructor	write Passed	No Issue

PriceOracle.sol

Sl.	Functions	Type Observation	Conclusion
1	getUnderlyingPrice	external Passed	No Issue

Reservoir.sol

Sl.	Functions	Type Observation	Conclusion
1	drip	write Passed	No Issue
2	constructor	write Passed	No Issue
3	add	internal Passed	No Issue
4	sub	internal Passed	No Issue
5	mul	internal Passed	No Issue
6	min	internal Passed	No Issue

SimplePriceOracle.sol

Sl.	Functions	Type Observation	Conclusion
1	getUnderlyingPrice	write Passed	No Issue
2	setUnderlyingPrice	write Passed	No Issue
3	setDirectPrice	write Passed	No Issue
4	assetPrices	external Passed	No Issue
5	compareStrings	internal Passed	No Issue

Timelock.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	receive	external Passed	No Issue
3	setDelay	write Passed	No Issue
4	acceptAdmin	write Passed	No Issue
5	setPendingAdmin	write Passed	No Issue
6	queueTransaction	write Passed	No Issue
7	cancelTransaction	write Passed	No Issue
8	executeTransaction	write Passed	No Issue
9	getBlockTimestamp	internal Passed	No Issue

Unitroller.sol

Sl.	Functions	Type Observation	Conclusion
1	constructor	write Passed	No Issue
2	_setPendingImplement ation	write Passed	No Issue
3	_acceptImplementation	write Passed	No Issue
4	_setPendingAdmin	write Passed	No Issue
5	_acceptAdmin	write Passed	No Issue

Audit Findings

Critical

No Critical severity vulnerabilities were found.

High

No High severity vulnerabilities were found.

Medium

No Medium severity vulnerabilities were found.

Low

No Low severity vulnerabilities were found.

Very Low / Discussion / Best practices:

(1) Use the latest solidity version - (All contracts)

```
pragma solidity ^0.5.16;
```

Using the latest solidity will prevent any compiler-level bugs. **Resolution**: Please use 0.8.7 which is the latest version.

Status: Open

(2) All functions which are not called internally, must be declared as external. It is more efficient as sometimes it saves some gas.

https://ethereum.stackexchange.com/questions/19380/external-vs-public-best-practices **Status:** Open

Centralization

These smart contracts have some functions which can be executed by Admin (Owner) only. If the admin wallet private key would be compromised, then it would create trouble. Following are Admin functions:

- updateMultiplier: The Master owner can update the multiplier Number. ● add: The Master Owner can add a new lp to the pool.

● set: The Master owner can update the given pool's XLD allocation point. ● setMigrator: The Master owner can set the migrator contract.

Conclusion

We were given a contract code. And we have used all possible tests based on given objects as files. We observed some issues, but they are not critical. So, **it's good to go to production**.

Since possible test cases can be unlimited for such smart contracts protocol, we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan everything.

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools. Smart Contract's high level description of functionality was presented in As-is overview section of the report.

Audit report contains all found security vulnerabilities and other issues in the reviewed code.
Security state of the reviewed contract, based on standard audit procedure scope, is **"Secured".**