RD AUDITORS

VAULTY SMART CONTRACT, CODE REVIEW AND SECURITY ANALYSIS REPORT

Customer: Vaulty
Prepared on: 26 August 2021
Platform: Binance Smart Chain
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 THE DECISION OF THE CUSTOMER.

Document

Name	Smart Contract Code Review and Security Analysis Report for Vaulty
Platform	BSC / Solidity
File 1	HolviNft.sol
MD5 hash	3960EC4CAFFC4D1D1FF06C327179F730
SHA256 hash	F238F591FFB80672B4436E1637A3B825B2045359AD21134AA78FA27AB074A991
File 2	LanttiPoolProxy.sol
MD5 hash	FFDFF5CFF23BBDE6B78590A17F0BC74B
SHA256 hash	2221BAC8D300CD38DDC6C2F96A97A65AB6A7E9398FFA1C464365D8FCEFE0C7CD
File 3	NftHub.sol
MD5 hash	10E186F49032AA4DA5CBCFD651D9D242
SHA256 hash	87D4BC96A1A7245D57FEC93DB3C46976DED76982703A5841F2BE621148E1888D
File 4	NFTHubMock.sol
MD5 hash	93C2FCCFAA585FE881C6F1AD09D45119
SHA256 hash	E69DC903F8932FE552D4E7C1DD91B3D76FB07610A0DC4CD684BA4BD07DAD9913
File 5	NFTHubProxy.sol
MD5 hash	E8C913D44B9CDC9CE36CC114E2239292

SHA256 hash	83E5BEF734FF26D1743816109152F9A4C7DD44B87BC234E93B5635D2244B851D
File 6	NFTMarket.sol
MD5 hash	64DE0EA604B4D869595779D41EB6D263
SHA256 hash	83E5BEF734FF26D1743816109152F9A4C7DD44B87BC234E93B5635D2244B851D
File 7	NFTMarketProxy.sol
MD5 hash	C97B7BF9A0F3A2D1ADD8C698B0012BF2
SHA256 hash	E901B87743FF44738DD6D9BA25C37352919DEDC00D73B6007D8E26023CCB3833
File 8	BeltMultiStrategy.sol
MD5 hash	C629816E613185C78540049CDCD783AA
SHA256 hash	3F4F38A77F6B59EC144DE6D13BFD59A6AA4BB110EF4DE57B14EF3BEC40AF0B9F
File 9	BeltMultiStrategy_4Belt.sol
MD5 hash	C629816E613185C78540049CDCD783AA
SHA256 hash	3F4F38A77F6B59EC144DE6D13BFD59A6AA4BB110EF4DE57B14EF3BEC40AF0B9F
File 10	BeltPStrategy_BELT_BNB.sol
MD5 hash	0D4DB1454B7C269A29FE1B67FA0262F3
SHA256 hash	28FD58043798A6CE3CA1DF814422A8366E09F8AB8A4E7325C8AD14C1706F5E3D
File 11	BeltSingleStrategy_BTCB.sol
MD5 hash	B4CB311CB0059CDB5AF03F3EC055140C

SHA256 hash	783D0DA076F1E1225CCF371C1DEAFACC76DE56E11A782EB861168EB9232B2D3B
File 12	BeltSingleStrategy_ETH.sol
MD5 hash	FAFCB03EC7F79B6D37F5577E3754FB2B
SHA256 hash	12D6F7ACD1D03B7183CE197929885256CE678B8F0B9E7ECB3B441DAE5A7A4F33
File 13	BeltSingleStrategy.sol
MD5 hash	6083E92AC5D8AF1CE98B18DE89485B2C
SHA256 hash	6088812CF21F5B557C41CD8D8A1910B610510C0A0306123AB85F77E09A20243C
File 14	GeneralMasterchefStrategyNewRouter.sol
MD5 hash	9BD2DFC1591EA48AC7A7718FABCF2811
SHA256 hash	FD1CE359628E3F8EAE24DFF987297805804AA36A08527EDD9112ED3FBABA683E
File 15	AlpacaALPACAStrategy.sol
MD5 hash	DC48D82A493E55C3EE3DD285C4364A27
SHA256 hash	95791B0176430C4FE0C746CCF3DC387604F15B4AFAF8676A97A99DB9E1CED1DF
File 16	AlpacaBTCBStrategy.sol
MD5 hash	F60702E107C565EC1CB2AFA26E858C46
SHA256 hash	613F43F5BC6B61686EF0B7A61BE5C4DE2A4413DB899E9E02DA3461450B88D976
File 17	AlpacaBUSDStrategy.sol
MD5 hash	29C09E3043E9B366E47EDDB4DCA534F8

SHA256 hash	83D26E2351BF68271F14362B7F3E7847EB8B68A3F1F17DEBDE08178CEB9C86A5
File 18	AlpacaBaseStrategy.sol
MD5 hash	9C70EC664928DFB2915EFF4F0317601E
SHA256 hash	A1ED3D1F27F5DF3D73C956D0454FFDD6096A86D69443717EA78F79F7A1F096FF
File 19	AlpacaETHStrategy.sol
MD5 hash	CFF80DDD0D732E4C99C2FEE81CC2DDBE
SHA256 hash	AEC8AAD088E63BE691E76247C80ADFB18E4D8AFD731CDB1AD57880D535C3A3D3
File 20	AlpacaUSDTStrategy.sol
MD5 hash	004C61DE45195A2B90BAD2BAD7AA4A6E
SHA256 hash	19DD26B86DF776EE9DBB3AC8A19D95A4872D1BEBEB0B81BF4C9291DC4525BD32
File 21	TimelockController.sol
MD5 hash	4FBE97EB8C327063A91E67E348F892C6
SHA256 hash	A0E1A7167139B589E141F58ABD16308DCAAEBEE87FEE2026477479025EAE62A7
File 22	BEP20.sol
MD5 hash	D389CAE22ACACDAF69D6D48793B68BFA
SHA256 hash	D3227AAD9EEA873529A29C3ABE18D66679FD73B72ACD88408A9741CC424FCBAF
File 23	ERC1155.sol
MD5 hash	768C3A11866F0AD605C2ADF11E22E501

SHA256 hash	30658F7728B5E90039C227483F9A6EE1D058C87593B020AE28822B13D95ABA2D
File 24	Latti.sol
MD5 hash	D3785534FBEA2A3FD95642317B4E53EE
SHA256 hash	BF585CD8433EF73EEDD6C22307A53FFC83D977BD19F6D3D4D935E40B32A78ABF
File 25	ProxyRegistery.sol
MD5 hash	5A2EA550CB00018CEC69B85E57AFD2D9
SHA256 hash	AF3EC03D63FDDAF4A8E3F05FBD6906C3A2D4C37BEACE642690A07B0F06135105
File 26	RewardToken.sol
MD5 hash	0766B1F80C2D9CAE8C11537DF4A2FA71
SHA256 hash	3C70B8069C6CB27E98768BFB98377CEC4D8416666343D0225EA7E807929BBC5A
File 27	BaseProxyStorage.sol
MD5 hash	BA5FF4739F416644BAEC5FFAD8885B1D
SHA256 hash	2662300E2F786CC07FD78BED6794BF1E5B3F7BC8C3594EA68B4E2D3F4BF9C68F
File 28	BaseUpgradeableStrategy.sol
MD5 hash	5BF31DFDC222F9CA39894777677C74B6
SHA256 hash	25EAE13F48BB23FBBAC14C7F9796CF5F74C21B52C3AE9CD3C0F5BB1061FFE05F
File 29	BaseUpgradeableStrategyStorage.sol
MD5 hash	5BE14FD62070EDC061D9065FFFD00B60

SHA256 hash	F072589A49F6347C8CA079A2562DCB64748E89C1DBE5A129B444DC5BEF7902F1
File 30	StrategyProxy.sol
MD5 hash	3F10416ACF339381EFEEB7DBD64A3F7B
SHA256 hash	C4B9743518E3F429E7B99974D2528E2BC5C182E8A388330B41BFB466AAB0F4B8
File 31	Controllable.sol
MD5 hash	B4D5E3A1734E3F5975A60190B5A35807
SHA256 hash	DF154459642E9BDEC7ACDAA14FBAD1F952FD2A34CF5BD7EF150AB8EDFC7AA450
File 32	ControllableInit.sol
MD5 hash	F20E28C046A58BAB4BFC7BC7B2B91DC5
SHA256 hash	E511B5DD48CE47924ED19EA5F72FCF5766422361CF46309D88400B42ED615A0D
File 33	ExclusiveRewardPool.sol
MD5 hash	51DE5BBECA6B9BCED029E3FE1AD11B3D
SHA256 hash	3F9D43396D0A6BD463A3B75C9F8D83084D988F8F64EC1AFD963FC652A93DADB1
File 34	GovernableInit.sol
MD5 hash	E6339712097B4AA886074ACDDA9580BA
SHA256 hash	0A526A96F74D4D777630D9F4F38848FE8D75CF4F5CECB014B54AC0FF1E3AA55D
File 35	Governable.sol
MD5 hash	C51D6F52C805D3FEE235535C94D5596D

SHA256 hash	8D092C6F73064FBF4CF1AD1EBEDFA75EA1D0FB7BE807A6E40E190C285EF1D052
File 36	GovernanceStaking.sol
MD5 hash	1F1C08F2E5F21DA46C2067C6EDE48CC6
SHA256 hash	8A215513495BE79E0C0DCD7A2C1B070925BDEA60B99AF447EA325E487EFCE0BC
File 37	LPTokenWrapper.sol
MD5 hash	9E3729AD14C307F73423ED200F506DCF
SHA256 hash	80A06C4BE387FE46E1F40F206B45377DACA518F05F44CD481C50EA96E249E7CF
File 38	NoMintRewardPool.sol
MD5 hash	C25771180834316CF1D5B7DD20A5A79D
SHA256 hash	8EFB7C0F524D4DEC35C1194BC8B307BBA3C4A8401C2E99C85AAC878B93D18FC2
File 39	NotifyHelper.sol
MD5 hash	E8FF4BABB3BBE86FBFEE4924826E5554
SHA256 hash	2AA7894323797BC7A9C0AC04A5AC6856759432794EB0E27DA67A908FA499CEFE
File 40	RewardDistributionRecipient.sol
MD5 hash	A81B398736CA01F06AE3B02ECC260183
SHA256 hash	EFE411FE1DFA46010FD3D2291C4127490222D4198E1117EB3046A4A40BCECC57
File 41	Storage.sol
MD5 hash	F51A9CD407221DB4C2FDC7D38CCFE0C8

SHA256 hash	A7566CFF9D7845D5FF01DD149E8F3172869C5B53472625D523A662444E49F9E1
File 42	TokenVesting.sol
MD5 hash	27FB3729A915818B034A650FA0CE0172
SHA256 hash	C2D9B7EC596037E9D2E95B4FE5B5471CA20A3FA5ACD86D6BA464CDDA82A69F16
File 43	Vault.sol
MD5 hash	50496AA597188369896EB5FDC2344F90
SHA256 hash	40C7EE4341C751B7F2F0538C25ED21E432979E164DBC742D82E5B8F971F22E72
File 44	VaultFactory.sol
MD5 hash	49871677A1FCDDAEF8CFD298540277E3
SHA256 hash	D83B42CDB78EBE9588601BCE163E805D191ABE99919BD90A6047DEC028170C93
File 45	VaultProxy.sol
MD5 hash	A46B51EAAEEC9C62D11D09221D8AEE62
SHA256 hash	22461A5FD6B934541C75B5616975C57B0174B76B9946871961F37707FC11FEF6
File 46	VaultStorage.sol
MD5 hash	FEC773CE9D7EE8FE35ED0D1BF829AF72
SHA256 hash	4D45D2ADFE0E5F54504A1D9A83954F490C38F1483341E80890DB0092C875D687
File 47	VenusWBNBFoldStrategy.sol
MD5 hash	4168B97130A129D2659C920EC64F80AA

SHA256 hash	E0FA3CD6CB191364652E73402F97C6F5E7411E678F982E6A41C1C7D0BD7AC5ED
File 48	VenusInteractorInitializable.sol
MD5 hash	4168B97130A129D2659C920EC64F80AA
SHA256 hash	E0FA3CD6CB191364652E73402F97C6F5E7411E678F982E6A41C1C7D0BD7AC5ED
File 49	VenusInteractor.sol
MD5 hash	754FDA1CD26058E98057FA1B498A2206
SHA256 hash	F4BEB9C6644319089B65FBEB40AA12ADED2CF3628D383C0FC4C431E8ECEEDD83
File 50	VenusFoldStrategy.sol
MD5 hash	3FAE0D9F8EC62FB8EFF23914D9E705C5
SHA256 hash	A0A8EDB8D1843034A09D20DCA8B60657EAB8DA8FBC6189A6C8D6EC2312CC56B5
File 51	PanCakeMasterchefLPStrategy.sol
MD5 hash	B56998B9648EF38037714255ACDC5DB9
SHA256 hash	67230332489DEEFA0F465006F45D0240E87C58543485283C08D023DAD92FEE3B
Date	26/08/2021

Introduction

RD Auditors (Consultant) were contracted by Vaulty (Customer) to conduct a Smart Contracts Code Review and Security Analysis. This report represents the findings of the security assessment of the customer`s smart contracts and its code review conducted between 8 - 26 August 2021.
This contract consists of 51 main and other supporting files.

Project Scope

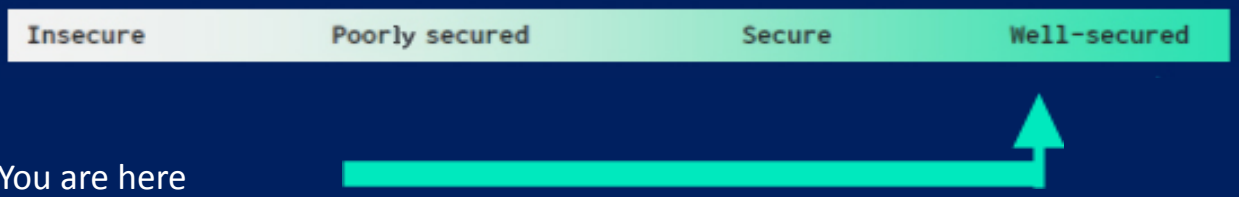
The scope of the project is a smart contract.

We have scanned this smart contract for commonly known and more specific vulnerabilities, below are those considered (the full list includes but is 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 smart contract is well-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, 0 high, 0 medium, 0 low and 0 very low level issues.

Code Quality

The libraries within this smart contract are part of a 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 to a specific address and its properties/methods can be reused many times by other contracts.

The Vaulty development team has not provided unit test scripts, which fortify functionality and security of the contract, which also helps us to determine the integrity of the code in an automated way.

Overall, the code is well commented. Commenting can provide rich documentation for functions, return variables and more. Use of the Ethereum Natural Language Specification Format (NatSpec) for commenting is recommended.

Documentation

We were given the Vaulty contract as a github link:
https://github.com/VaultyFinance/contracts

The hash of that file is mentioned in the table. As mentioned above, It's well commented 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 that are written well and systematically.

AS-IS Overview

Vaulty

Vaulty is defi diversified investment and reward with a pancake interface using vaulty token.

File And Function Level Report

File: HolviNft.sol

Contract: HolviNft
Import: ERC1155
Inherit: ERC1155, Mintable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	endMinting	write	Passed	All Passed	No Issue	Passed
2	burnFt	write	Passed	All Passed	No Issue	Passed
3	burnNft	write	Passed	All Passed	No Issue	Passed
4	airdropFt	write	Passed	All Passed	No Issue	Passed
5	airdropNft	write	Passed	All Passed	No Issue	Passed

File: LantiPoolProxy.sol

Contract: LantiPoolsProxy
Inherit: UpgradeableProxy
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

File: LantiPools.sol

Contract: LantiPools
Import: IBEP20, SafeBEP20, SafeMath, Governable, Lantti, INftHub, IUpgradeSource, ControllableInit, BaseProxyStorage
Inherit: BaseProxyStorage, controllableInit, IUpgradeSource
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	LantiPools	write	Passed	All Passed	No Issue	Passed
2	initialize	read	Passed	All Passed	No Issue	Passed
3	PoolLength	write	Passed	All Passed	No Issue	Passed
4	add	write	Passed	All Passed	No Issue	Passed
5	setMaxStake	write	Passed	All Passed	No Issue	Passed
6	setLanttiPerDay	read	Passed	All Passed	No Issue	Passed
7	_UserPoolState	read	Passed	All Passed	No Issue	Passed
8	PendingLantti	read	Passed	All Passed	No Issue	Passed
9	totalPendingLantti	write	Passed	All Passed	No Issue	Passed
10	rugPull	write	Passed	All Passed	No Issue	Passed
11	deposit	write	Passed	All Passed	No Issue	Passed
12	withdraw	write	Passed	All Passed	No Issue	Passed
13	emergencyWithdraw	write	Passed	All Passed	No Issue	Passed
14	UpdateNftHubAddress	write	Passed	All Passed	No Issue	Passed

15	ScheduleUpgrade	write	Passed	All Passed	No Issue	Passed
16	shouldUpgrade	read	Passed	All Passed	No Issue	Passed
17	finalizeUpgrade	write	Passed	All Passed	No Issue	Passed

File: NFTHub.sol

Contract: NftHub
Import: Governable.sol, SafeMath, Lantti, HolviNft, LanttiPools, INftHub.sol, IUpgradeSource.sol, controllableInit, BaseProxyStorage
Inherit: controllableInit, BaseProxyStorage, IUpgradeSource, INftHub
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	initialize	write	Passed	All Passed	No Issue	Passed
2	setLanttiPools	write	Passed	All Passed	No Issue	Passed
3	setMultiplierofAddress	write	Passed	All Passed	No Issue	Passed
4	_isInArray	read	Passed	All Passed	No Issue	Passed
5	getNftStakedofAddress	read	Passed	All Passed	No Issue	Passed
6	getNftIdListOfSet	read	Passed	All Passed	No Issue	Passed
7	getBoostersOfNft	read	Passed	All Passed	No Issue	Passed
8	getFullSetofAddress	read	Passed	All Passed	No Issue	Passed
9	getNumOfNftsStakedForSet	read	Passed	All Passed	No Issue	Passed
10	getNumOfNftsStakedByAddress	read	Passed	All Passed	No Issue	Passed
11	totalPendingLanttiOfAddress	read	Passed	All Passed	No Issue	Passed
12	getBoosterForUser	read	Passed	All Passed	No Issue	Passed
13	addNftSet	write	Passed	All Passed	No Issue	Passed
14	SetLanttiRateOfSets	write	Passed	All Passed	No Issue	Passed
15	SetBonusLanttiMultiplierOfSets	write	Passed	All Passed	No Issue	Passed
16	removeNftSet	write	Passed	All Passed	No Issue	Passed
17	harvest	write	Passed	All Passed	No Issue	Passed

18	stake	write	Passed	All Passed	No Issue	Passed
19	UnStake	write	Passed	All Passed	No Issue	Passed
20	stateAction	write	Passed	All Passed	No Issue	Passed
21	emergencyUnstake	write	Passed	All Passed	No Issue	Passed
22	UpdateLanttiPoolAddress	write	Passed	All Passed	No Issue	Passed
23	extractType	read	Passed	All Passed	No Issue	Passed
24	OnERC1155Received	write	Passed	All Passed	No Issue	Passed
25	OnERC1155BatchReceived	write	Passed	All Passed	No Issue	Passed
26	SupportsInterface	read	Passed	All Passed	No Issue	Passed
27	ScheduleUpgrade	write	Passed	All Passed	No Issue	Passed
28	shouldUpgrade	read	Passed	All Passed	No Issue	Passed
29	finalizeUpgrade	write	Passed	All Passed	No Issue	Passed

File: NFTHubMock.sol

Contract: NFTHubMock
Inherit: INftHub
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	getBoosterForUser	read	Passed	All Passed	No Issue	Passed

File: NFTHubProxy.sol

Contract: NftHub
Inherit: UpgradeableProxy
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

File: NFTMarket.sol

Contract: NftHub
Inherit: Governable, Lantti, HolviNft, safeMath, IUpgradeSource, ControllableInit, BaseProxyStorage
Import: ControllableInit, BaseProxyStorage, IUpgradeSource
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	initialize	write	Passed	All Passed	No Issue	Passed
2	getSet	read	Passed	All Passed	No Issue	Passed
3	removeSet	write	Passed	All Passed	No Issue	Passed
4	addToSet	write	Passed	All Passed	No Issue	Passed
5	CreateSet	write	Passed	All Passed	No Issue	Passed
6	openSetFor	write	Passed	All Passed	No Issue	Passed
7	nextRandom	write	Passed	All Passed	No Issue	Passed
8	OnERC1155Received	write	Passed	All Passed	No Issue	Passed
9	OnERC1155BatchReceived	write	Passed	All Passed	No Issue	Passed
10	SupportsInterface	read	Passed	All Passed	No Issue	Passed
11	ScheduleUpgrade	write	Passed	All Passed	No Issue	Passed
12	ShouldUpgrade	read	Passed	All Passed	No Issue	Passed
13	finalizeUpgrade	write	Passed	All Passed	No Issue	Passed

File: NFTMarketProxy.sol

Contract: NftMarketProxy
Inherit: UpgradeableProxy
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

File: BeltMultiStrategy_4Belt.sol

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

File:BeltMultiStrategy.sol

Contract: BeltMultiStrategy
Inherit: IMasterBelt, IDepositor, MathUpgradeable, safeMath, BaseUpgradeableStrategy, IPancakeRouter02
Import: BaseUpgradeableStrategy
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	depositArbCheck	read	Passed	All Passed	No Issue	Passed
2	rewardPoolBalance	read	Passed	All Passed	No Issue	Passed
3	unsalvagableTokens	read	Passed	All Passed	No Issue	Passed
4	depositArbCheck	read	Passed	All Passed	No Issue	Passed
5	rewardPoolBalance	read	Passed	All Passed	No Issue	Passed
6	UnSalvagableTokens	read	Passed	All Passed	No Issue	Passed
7	emergencyExit	write	Passed	All Passed	No Issue	Passed
8	ContinueInvesting	write	Passed	All Passed	No Issue	Passed
9	SetLiquidationPath	write	Passed	All Passed	No Issue	Passed
10	WithdrawAllToVault	write	Passed	All Passed	No Issue	Passed
11	WithdrawToVault	write	Passed	All Passed	No Issue	Passed
12	investedUnderlyingBalance	read	Passed	All Passed	No Issue	Passed
13	Salvage	write	Passed	All Passed	No Issue	Passed
14	setSellFloor	write	Passed	All Passed	No Issue	Passed
15	PoolId	read	Passed	All Passed	No Issue	Passed
16	depositor	read	Passed	All Passed	No Issue	Passed

File: BeltPStrategy_BELT_BNB.sol

Contract: BeltPStrategy_BELT_BNB
Inherit: GeneralMasterchefStrategyNewRouter
Import: GeneralMasterchefStrategyNewRouter
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

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

File: BeltSingleStrategy_ETH.sol

Contract: BeltSingleStrategy_BeltETH
Inherit: BeltSingleStrategy
Import: BeltSingleStrategy
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

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

File: BeltSingleStrategy.sol

Contract: BeltSingleStrategy
Inherit: BaseUpgradeableStrategy
Import: IMasterBelt, IMultiStrategyToken, IBEP20, SafeBEP20, MathUpgradeable, safeMath, BaseUpgradeableStrategy, IPancakeRouter02
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	initialize	write	Passed	All Passed	No Issue	Passed
2	depositArbCheck	read	Passed	All Passed	No Issue	Passed
3	UnsalvagableTokens	read	Passed	All Passed	No Issue	Passed
4	emergencyExit	write	Passed	All Passed	No Issue	Passed
5	ContinueInvesting	write	Passed	All Passed	No Issue	Passed
6	SetliquidationPath	write	Passed	All Passed	No Issue	Passed
7	WithdrawAllToVault	write	Passed	All Passed	No Issue	Passed
8	WithdrawToVault	write	Passed	All Passed	No Issue	Passed
9	investedUnderlyingBalance	read	Passed	All Passed	No Issue	Passed
10	salvage	write	Passed	All Passed	No Issue	Passed
11	doHardwork	write	Passed	All Passed	No Issue	Passed
12	setSell	write	Passed	All Passed	No Issue	Passed
13	SetSellFloor	write	Passed	All Passed	No Issue	Passed
14	PoolId	read	Passed	All Passed	No Issue	Passed
15	Depositor	read	Passed	All Passed	No Issue	Passed
16	depositorUnderlying	read	Passed	All Passed	No Issue	Passed
17	finalizeUpgrade	write	Passed	All Passed	No Issue	Passed

File: BeltSingleStrategy_BTCB.sol

Contract: BeltSingleStrategy
Inherit: BeltSingleStrategy
Import: BeltSingleStrategy
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

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

File:GeneralMasterchefStrategyNewRouter.sol

Contract: GeneralMasterchefStrategyNewRouter
Inherit: BaseUpgradeableStrategy
Import: IMasterChef, IStrategy, IBEP20, SafeBEP20, MathUpgradeable, SafeMath, IVault, BaseUpgradeableStrategy, IPanCakePair, IPanCakeRouter02
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	initialize	write	Passed	All Passed	No Issue	Passed
2	depositArbCheck	write	Passed	All Passed	No Issue	Passed
3	UnsalvagableTokens	read	Passed	All Passed	No Issue	Passed
4	emergencyExit	write	Passed	All Passed	No Issue	Passed

5	ContinueInvesting	write	Passed	All Passed	No Issue	Passed
6	SetliquidationPath	write	Passed	All Passed	No Issue	Passed
7	WithdrawAllToVault	write	Passed	All Passed	No Issue	Passed
8	WithdrawToVault	write	Passed	All Passed	No Issue	Passed
9	investUnderlyingBalance	read	Passed	All Passed	No Issue	Passed
10	salvage	write	Passed	All Passed	No Issue	Passed
11	doHardwork	write	Passed	All Passed	No Issue	Passed
12	setSell	write	Passed	All Passed	No Issue	Passed
13	SetSellFloor	write	Passed	All Passed	No Issue	Passed
14	PoolId	read	Passed	All Passed	No Issue	Passed
15	isLpAsset	read	Passed	All Passed	No Issue	Passed
16	finalizeUpgrade	write	Passed	All Passed	No Issue	Passed

File: AlpacaBaseStrategy.sol

Contract: AlpacaBaseStrategy
Inherit: BaseUpgradeableStrategy
Import: IFairLaunch, IVault, IBEP20, SafeBEP20, MathUpgradeable, safeMath, BaseUpgradeableStrategy, IPancakeRouter02
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	initialize	write	Passed	All Passed	No Issue	Passed
2	depositArbCheck	read	Passed	All Passed	No Issue	Passed
3	rewardPoolBalance	read	Passed	All Passed	No Issue	Passed
4	UnSalvagableTokens	read	Passed	All Passed	No Issue	Passed
5	emergencyExit	write	Passed	All Passed	No Issue	Passed
6	ContinueInvesting	write	Passed	All Passed	No Issue	Passed
7	SetLiquidationPath	write	Passed	All Passed	No Issue	Passed
8	WithdrawAllToVault	write	Passed	All Passed	No Issue	Passed
9	WithdrawToVault	write	Passed	All Passed	No Issue	Passed
10	investUnderlyingBalance	read	Passed	All Passed	No Issue	Passed
11	Salvage	write	Passed	All Passed	No Issue	Passed
12	doHardWork	write	Passed	All Passed	No Issue	Passed

13	SetSell	write	Passed	All Passed	No Issue	Passed
14	SetSellFloor	write	Passed	All Passed	No Issue	Passed
15	PoolId	read	Passed	All Passed	No Issue	Passed
16	depositor	read	Passed	All Passed	No Issue	Passed
17	depositorUnderlying	read	Passed	All Passed	No Issue	Passed
18	finalizeUpgrade	write	Passed	All Passed	No Issue	Passed

Contract: AlpacaETHStrategy
Inherit: AlpacaBaseStrategy
Import: AlpacaBaseStrategy
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

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

Contract: AlpacaUSDTStrategy
Inherit: AlpacaBaseStrategy
Import: AlpacaBaseStrategy
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

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

File:TimelockController

Contract: TimelockController
Inherit: AccessControl
Import: AccessControl
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	isOperation	read	Passed	All Passed	No Issue	Passed
2	isOperationPending	read	Passed	All Passed	No Issue	Passed
3	isOperationReady	read	Passed	All Passed	No Issue	Passed
4	isOperationDone	read	Passed	All Passed	No Issue	Passed
5	getTimeStamp	read	Passed	All Passed	No Issue	Passed
6	getMinDelay	read	Passed	All Passed	No Issue	Passed
7	hashOperation	read	Passed	All Passed	No Issue	Passed
8	hashOperationBatch	read	Passed	All Passed	No Issue	Passed
9	Schedule	write	Passed	All Passed	No Issue	Passed
10	ScheduleBatch	write	Passed	All Passed	No Issue	Passed
11	_Schedule	write	Passed	All Passed	No Issue	Passed
12	cancel	write	Passed	All Passed	No Issue	Passed
13	execute	write	Passed	All Passed	No Issue	Passed
14	executeBatch	write	Passed	All Passed	No Issue	Passed
15	_beforeCall	read	Passed	All Passed	No Issue	Passed
16	_afterCall	write	Passed	All Passed	No Issue	Passed
17	_call	write	Passed	All Passed	No Issue	Passed
18	UpdateDelay	write	Passed	All Passed	No Issue	Passed

File:BEP20.sol

Contract: BEP20
Inherit: Context, IBEP20, Ownable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	getOwner	read	Passed	All Passed	No Issue	Passed
2	name	read	Passed	All Passed	No Issue	Passed
3	decimals	read	Passed	All Passed	No Issue	Passed
4	symbol	read	Passed	All Passed	No Issue	Passed
5	totalSupply	read	Passed	All Passed	No Issue	Passed
6	balanceOf	read	Passed	All Passed	No Issue	Passed
7	transfer	write	Passed	All Passed	No Issue	Passed
8	allowance	read	Passed	All Passed	No Issue	Passed
9	approve	read	Passed	All Passed	No Issue	Passed
10	transferFrom	read	Passed	All Passed	No Issue	Passed
11	increaseAllowance	write	Passed	All Passed	No Issue	Passed
12	decreaseAllowance	write	Passed	All Passed	No Issue	Passed

File:ERC1155

Contract: ERC1155
Inherit: IERC165
Import: Ownable, IBEP20, Context, SafeMath, Address, IERC165, IERC1155TokenReceiver, minterRole, Whitelist, AdminRole, Strings, ProxyRegistery
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	getNonFungibleIndex	read	Passed	All Passed	No Issue	Passed
2	getNonFungibleBaseType	read	Passed	All Passed	No Issue	Passed
3	OwnerOf	read	Passed	All Passed	No Issue	Passed
4	isNonFungible	read	Passed	All Passed	No Issue	Passed
5	SafeTransferFrom	write	Passed	All Passed	No Issue	Passed
6	SafeBatchTransferFrom	write	Passed	All Passed	No Issue	Passed
7	SetApprovalForAll	write	Passed	All Passed	No Issue	Passed

8	isApproveForAll	read	Passed	All Passed	No Issue	Passed
9	balanceOf	read	Passed	All Passed	No Issue	Passed
10	balanceOfBatch	read	Passed	All Passed	No Issue	Passed
11	SupportsInterface	read	Passed	All Passed	No Issue	Passed

Contract: ERC1155Metadata
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	Uri	read	Passed	All Passed	No Issue	Passed

Contract: ERC1155Mintable
Inherit: ERC1155, ERC1155MintBurn, ERC1155MetaData, Ownable, MinterRole, WhitelistAdminRole
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	removeWhitelistAdmin	write	Passed	All Passed	No Issue	Passed
2	removeMinter	write	Passed	All Passed	No Issue	Passed
3	totalSupply	read	Passed	All Passed	No Issue	Passed
4	maxSupply	read	Passed	All Passed	No Issue	Passed
5	SetBaseMetadataURI	write	Passed	All Passed	No Issue	Passed
6	Create	write	Passed	All Passed	No Issue	Passed
7	mintFt	write	Passed	All Passed	No Issue	Passed
8	mintNft	write	Passed	All Passed	No Issue	Passed
9	isApprovedForAll	read	Passed	All Passed	No Issue	Passed
10	_exists	read	Passed	All Passed	No Issue	Passed

File:Latti.sol

Contract: **Latti**
Inherit: BEP20, Governable, MinterRole
Import: BEP20, Governable, MinterRole, SafeMath
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	mint	write	Passed	All Passed	No Issue	Passed
2	burn	write	Passed	All Passed	No Issue	Passed

File:ProxyRegistery.sol

Contract: ProxyRegistry
Inherit: Governable
Import: Governable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	addOperator	write	Passed	All Passed	No Issue	Passed
2	removeOperator	write	Passed	All Passed	No Issue	Passed

File: RewardToken.sol

Contract: RewardToken
Inherit: BEP20, MinterRole
Import: BEP20, MinterRole
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

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

File: BaseProxyStorage.sol

Contract: BaseProxyStorage
Inherit: Initializable
Import: Initializable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	nextImplementation	read	Passed	All Passed	No Issue	Passed
2	nextImplementationTimestamp	read	Passed	All Passed	No Issue	Passed
3	nextImplementationDelay	read	Passed	All Passed	No Issue	Passed

File: BaseUpgradeableStrategy.sol

Contract: BaseUpgradeableStrategy
Inherit: initializable, BaseUpgradeableStrategy Storage, ControllableInit
Import: initializable, BaseUpgradeableStrategy, StorageControllableInit, IController, IBEP20, SafeMath, SafeBEP20

Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	nextImplementation	read	Passed	All Passed	No Issue	Passed
2	nextImplementationTimestamp	read	Passed	All Passed	No Issue	Passed
3	nextImplementationDelay	read	Passed	All Passed	No Issue	Passed

File: BaseUpgradeableStrategyStorage.sol

Contract: BaseUpgradeableStrategyStorage
Inherit: initializable
Import: initializable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

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

3	rewardToken	read	Passed	All Passed	No Issue	Passed
4	Vault	read	Passed	All Passed	No Issue	Passed
5	Sell	read	Passed	All Passed	No Issue	Passed
6	PausedInvesting	read	Passed	All Passed	No Issue	Passed
7	SellFloor	read	Passed	All Passed	No Issue	Passed
8	ProfitSharingNumerator	read	Passed	All Passed	No Issue	Passed
9	ProfitSharingDenominator	read	Passed	All Passed	No Issue	Passed
10	nextImplementationTimeStamp	read	Passed	All Passed	No Issue	Passed
11	nextImplementationDelay	read	Passed	All Passed	No Issue	Passed

File: StrategyProxy.sol

Contract: StrategyProxy
Inherit: BaseUpgradeability
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

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

File: Controllable.sol

Contract: Controllable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	Controller	read	Passed	All Passed	No Issue	Passed

File: ControllableInit.sol

Contract: ControllableInit
Import: GovernableInit
Inherit: GovernableInit
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

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

File: ExclusiveRewardPool.sol

Contract: ExclusiveRewardPool
Import: LpTokenWrapper, RewardDistributionRecipient, Controllable, IController, AddressUpgradeable, IBEP20
Inherit: IPTokenWrapper, RewardDistributRecipient, Controllable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	SetWithdrawalDelay	write	Passed	All Passed	No Issue	Passed
2	SetWithdrawalFee	write	Passed	All Passed	No Issue	Passed
3	lastTimeRewardApplicable	read	Passed	All Passed	No Issue	Passed
4	rewardPerToken	read	Passed	All Passed	No Issue	Passed
5	earned	read	Passed	All Passed	No Issue	Passed

6	initExclusive	write	Passed	All Passed	No Issue	Passed
7	Stake	write	Passed	All Passed	No Issue	Passed
8	Withdraw	write	Passed	All Passed	No Issue	Passed
9	exit	write	Passed	All Passed	No Issue	Passed
10	PushReward	write	Passed	All Passed	No Issue	Passed
11	getReward	write	Passed	All Passed	No Issue	Passed
12	notifyRewardAmount	write	Passed	All Passed	No Issue	Passed

File: GovernableInit.sol

Contract: GovernableInit
Import: Initializable, Storage
Inherit: Initializable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	initialize	write	Passed	All Passed	No Issue	Passed
2	_setStorage	write	Passed	All Passed	No Issue	Passed
3	setStorage	write	Passed	All Passed	No Issue	Passed
4	governance	read	Passed	All Passed	No Issue	Passed

File: Governable.sol

Contract: Governable
Import: Storage
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

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

File: GovernanceStaking.sol

Contract: GovernanceStaking
Import: Governable, NoMintRewardPool, IBEP20, SafeBEP20
Inherit: Governable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	depositTokens	write	Passed	All Passed	No Issue	Passed
2	WithdrawTokens	write	Passed	All Passed	No Issue	Passed
3	Stake	write	Passed	All Passed	No Issue	Passed
4	Unstake	read	Passed	All Passed	No Issue	Passed

File: LPTokenWrapper.sol

Contract: GovernanceStaking
Import: SafeBEP20, IBEP20, SafeMath
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	depositTokens	write	Passed	All Passed	No Issue	Passed
2	WithdrawTokens	write	Passed	All Passed	No Issue	Passed
3	Stake	write	Passed	All Passed	No Issue	Passed
4	Unstake	read	Passed	All Passed	No Issue	Passed

File: NoMintRewardPool.sol

Contract: NoMintRewardPool
Import: LPTokenWrapper, RewardDistributionRecipient, Controllable, IController, AddressUpgradeable, IBEP20
Inherit: LPTokenWrapper, RewardDistributionRecipient, Controllable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	SetWithdrawDelay	write	Passed	All Passed	No Issue	Passed
2	SetWithdrawalFee	write	Passed	All Passed	No Issue	Passed
3	LastTimeRewardApplicable	read	Passed	All Passed	No Issue	Passed
4	rewardPerToken	read	Passed	All Passed	No Issue	Passed
5	earned	read	Passed	All Passed	No Issue	Passed
6	Stake	write	Passed	All Passed	No Issue	Passed
7	Withdraw	write	Passed	All Passed	No Issue	Passed
8	emit	write	Passed	All Passed	No Issue	Passed
9	PushReward	write	Passed	All Passed	No Issue	Passed
10	getReward	write	Passed	All Passed	No Issue	Passed
11	notifyRewardAmount	write	Passed	All Passed	No Issue	Passed

File: NotifyHelper.sol

Contract: NotifyHelper
Import: Controllable, NoMintRewardPool, WhitelistAdminRole, IBEP20, SafeBEP20, SafeMath
Inherit: WhitelistAdminRole, Controllable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	notifyPools	write	Passed	All Passed	No Issue	Passed
2	notifyPoolsIncludingProfitShare	write	Passed	All Passed	No Issue	Passed
3	notifyProfitSharing	write	Passed	All Passed	No Issue	Passed
4	SetFeeRewardForwarder	write	Passed	All Passed	No Issue	Passed

File: RewardDistributionRecipient.sol

Contract: RewardDistributionRecipient
Inherit: Ownable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	notifyRewardAmount	write	Passed	All Passed	No Issue	Passed
2	SetRewardDistribution	write	Passed	All Passed	No Issue	Passed

File: Storage.sol

Contract: **Storage**
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	Set Governance	write	Passed	All Passed	No Issue	Passed
2	Set Controller	write	Passed	All Passed	No Issue	Passed
3	isGovernance	read	Passed	All Passed	No Issue	Passed
4	isController	read	Passed	All Passed	No Issue	Passed

File: TokenVesting.sol

Contract: TokenVesting
Import: IERC20Upgradeable, SafeERC20Upgradeable, SafeMathUpgradeable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	addBeneficiaries	write	Passed	All Passed	No Issue	Passed
2	tokensToClaim	read	Passed	All Passed	No Issue	Passed
3	_tokensToClaim	read	Passed	All Passed	No Issue	Passed
4	ClaimFor	write	Passed	All Passed	No Issue	Passed
5	ClaimForSelf	write	Passed	All Passed	No Issue	Passed
6	ClaimForAll	write	Passed	All Passed	No Issue	Passed
7	endSetup	write	Passed	All Passed	No Issue	Passed
8	reclaimTokens	write	Passed	All Passed	No Issue	Passed

File: Vault.sol

Contract: Vault
Import: Mathupgradeable, SafeMathUpgradeable, SafeERC20Upgradeable, IERC20Upgradeable, ERC20Upgradeable, AddressUpgradeable, IStrategy, IVault, IController, IUpgradeSource, ControllableInit, VaultStorage
Inherit: ERC20Upgradeable, IVault, IUpgradeSource, ControllableInit, VaultStorage
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	initializeVault	write	Passed	All Passed	No Issue	Passed
2	Strategy	read	Passed	All Passed	No Issue	Passed
3	underlying	read	Passed	All Passed	No Issue	Passed
4	UnderlyingUnit	read	Passed	All Passed	No Issue	Passed
5	VaultFractionToInvestNumerator	read	Passed	All Passed	No Issue	Passed
6	nextImplementation	read	Passed	All Passed	No Issue	Passed
7	nextImplementationTimestamp	read	Passed	All Passed	No Issue	Passed
8	nextImplementationDelay	read	Passed	All Passed	No Issue	Passed
9	doHardWork	write	Passed	All Passed	No Issue	Passed
10	unlyingBalanceInVault	read	Passed	All Passed	No Issue	Passed
11	UnderlyingBalanceWithInvestmentForHolder	read	Passed	All Passed	No Issue	Passed
12	FutureStrategy	read	Passed	All Passed	No Issue	Passed
13	StrategyUpdateTime	read	Passed	All Passed	No Issue	Passed
14	StrategyTimeLock	read	Passed	All Passed	No Issue	Passed
15	CanUpdateStrategy	read	Passed	All Passed	No Issue	Passed
16	announceStrategyUpdate	write	Passed	All Passed	No Issue	Passed
17	finalizeStrategyUpdate	write	Passed	All Passed	No Issue	Passed
18	SetStrategy	write	Passed	All Passed	No Issue	Passed
19	SetVaultFractionToInvest	write	Passed	All Passed	No Issue	Passed
20	rebalance	write	Passed	All Passed	No Issue	Passed
21	availableToInvestOut	read	Passed	All Passed	No Issue	Passed
22	deposit	write	Passed	All Passed	No Issue	Passed
23	depositFor	write	Passed	All Passed	No Issue	Passed
24	WithdrawAll	write	Passed	All Passed	No Issue	Passed
25	Withdraw	write	Passed	All Passed	No Issue	Passed
26	ScheduleUpgrade	write	Passed	All Passed	No Issue	Passed
27	ShouldUpgrade	read	Passed	All Passed	No Issue	Passed
28	finalizeUpgrade	write	Passed	All Passed	No Issue	Passed

File: VaultFactory.sol

Contract: CreateVault
Import: VaultProxy, Vault, Controllable
Inherit: Controllable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

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

File: VaultProxy.sol

Contract: VaultProxy
Import: IUpgradeSource, BaseUpgradeabilityProxy
Inherit: BaseUpgradeabilityProxy
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

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

File: VaultStorage.sol

Contract: VaultStorage
Import: Initializable
Inherit: Initializable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	initialize	write	Passed	All Passed	No Issue	Passed
2	setAddress	write	Passed	All Passed	No Issue	Passed
3	SetUint256	write	Passed	All Passed	No Issue	Passed
4	getUint256	read	Passed	All Passed	No Issue	Passed

File: VenusFoldStrategy.sol

Contract: VenusFoldStrategy
Import: IBEP20, SafeBEP20, SafeMath, VenusInteractionInitializable, BaseUpgradeableStrategy, IVault, IPancakeRouter02
Inherit: BaseUpgradeableStrategy, VenusInteractorInitializable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	initializeStrategy	write	Passed	All Passed	No Issue	Passed
2	depositArbCheck	read	Passed	All Passed	No Issue	Passed
3	investAllUnderlying	write	Passed	All Passed	No Issue	Passed

4	withdrawAllToVault	write	Passed	All Passed	No Issue	Passed
5	emergencyExit	write	Passed	All Passed	No Issue	Passed
6	withdrawToVault	write	Passed	All Passed	No Issue	Passed
7	doHardWork	write	Passed	All Passed	No Issue	Passed
8	Salvage	write	Passed	All Passed	No Issue	Passed
9	investedUnderlyingBalance	read	Passed	All Passed	No Issue	Passed
10	SetAllowLiquidityShortage	write	Passed	All Passed	No Issue	Passed
11	SetCollateralFactorNumeerator	write	Passed	All Passed	No Issue	Passed
12	setFolds	write	Passed	All Passed	No Issue	Passed
13	setSellFloor	write	Passed	All Passed	No Issue	Passed
14	setSell	write	Passed	All Passed	No Issue	Passed

File: VenusInteractor.sol

Contract: VenusInteractor
Import: IBEP20, SafeBEP20, MathUpgradeable, SafeMath, ReentrancyGuardUpgradeable, IVBNB, CompleteVToken, WBNB
Inherit: ReentrancyGuardUpgradeable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

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

File: VenusInteractorInitializable .sol

Contract: VenusInteractorInitializable
Import: IBEP20, SafeBEP20, MathUpgradeable, SafeMath, GuardUpgradeable, IVBNB, CompleteVToken, WBNB,Initializable
Inherit: ReentrancyGuardUpgradeable, Initializable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	initialize	write	Passed	All Passed	No Issue	Passed
2	ClaimVenus	write	Passed	All Passed	No Issue	Passed
3	getLiquidity	read	Passed	All Passed	No Issue	Passed

File: VenusWBNBFoldStrategy .sol

Contract: VenusBNBFoldStrategy
Import: IBEP20, SafeBEP20, SafeMath, VenusInteractorInitializable, BaseuogradeableStrategy, IVault, IPancakeRouter02
Inherit: BaseUpgradeableStrategy, venusInteractorInitializable
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	InitializeStrategy	write	Passed	All Passed	No Issue	Passed
2	depositCheck	read	Passed	All Passed	No Issue	Passed
3	depositArbCheck	read	Passed	All Passed	No Issue	Passed

4	investAllUnderlying	write	Passed	All Passed	No Issue	Passed
5	WithdrawAllToVault	write	Passed	All Passed	No Issue	Passed
6	emergencyExit	write	Passed	All Passed	No Issue	Passed
7	WithdrawToVault	write	Passed	All Passed	No Issue	Passed
8	doHardWork	write	Passed	All Passed	No Issue	Passed
9	Salvage	write	Passed	All Passed	No Issue	Passed
10	investedUnderlyingBalance	read	Passed	All Passed	No Issue	Passed
11	setAllowLiquidityShortage	write	Passed	All Passed	No Issue	Passed
12	setBorrowMinThreshold	write	Passed	All Passed	No Issue	Passed
13	setCollateralFactorNumerator	write	Passed	All Passed	No Issue	Passed
14	setFolds	write	Passed	All Passed	No Issue	Passed
15	setSellFloor	write	Passed	All Passed	No Issue	Passed
16	setSell	write	Passed	All Passed	No Issue	Passed

File: PancakeMasterChefLPstrategy .sol

Contract: PanCakeMasterchefLPStrategy
Import: IMasterchef, IStrategy, MathUpgradeable, SafeMath, BaseUpgradeableStrategy, IVault, IPanCakeRouter02, IPancakePair
Inherit: BaseUpgradeableStrategy
Observation: Passed
Test Report: Passed
Score: Passed
Conclusion: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	initializeStrategy	write	Passed	All Passed	No Issue	Passed
2	depositArbCheck	read	Passed	All Passed	No Issue	Passed
3	rewardPoolBalance	read	Passed	All Passed	No Issue	Passed
4	UnSalvagableTokens	read	Passed	All Passed	No Issue	Passed
5	emergencyExit	write	Passed	All Passed	No Issue	Passed
6	ContinueVesting	write	Passed	All Passed	No Issue	Passed
7	SetLiquidationPathsOnPancake	write	Passed	All Passed	No Issue	Passed
8	WithdrawAllToVault	write	Passed	All Passed	No Issue	Passed
9	investedUnderlyingBalance	read	Passed	All Passed	No Issue	Passed

10	Salvage	write	Passed	All Passed	No Issue	Passed
11	doHardWork	write	Passed	All Passed	No Issue	Passed
12	setSell	write	Passed	All Passed	No Issue	Passed
13	setSellFloor	write	Passed	All Passed	No Issue	Passed
14	PoolId	read	Passed	All Passed	No Issue	Passed
15	finalizeUpgrade	write	Passed	All Passed	No Issue	Passed

Severity Definitions

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

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

No very low severity vulnerabilities were found.

Discussion

- Recommended solidity version 0.8.0 or above to be on the safe side, as it has many safety modifications.
- Hard coded values must be double checked before deploying.

Notice

'Emergency exit' and 'Contract upgradability' type functions exist within the smart contracts.

Update 16.10.21 - The Vaulty team is working on implementing a multisig wallet to mitigate potential risks.

Conclusion

We were given a contract file and have used all possible tests based on the given object. The contract is written systematically, so it is ready to go for production.

Since possible test cases can be unlimited and developer level documentation (code flow diagram with function level description) not provided, for such an extensive smart contract 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.

The security state of the reviewed contract is now “well 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 analyse 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 scrutinised 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 analysed 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, the audit makes no statements or warranties on the 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. 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

Email: info@rdauditors.com

Website: www.rdauditors.com