© Hexlant Inc.
340 Gangnam-daero
Gangnam-gu, Seoul.
Republic of Korea 06242
Hexlant.com

PANCAKE BUNNY-FINANCE SMART CONTRACT AUDIT REPORT

Audit Date
17 May 2020

Category
De-fi

Auditor
Hexlant Audit Team

This contract specifies that Hexlant Technical Team validates and notifies that it does not have any technical defects.

AUDIT OVERVIEW

PUBLISHED INFORMATION	
REPORT NUMBER	HEXLANT20210517
DATE	2021/05/17
AUDIT SCOPE	https://github.com/PancakeBunny-finance/Bunny/blob/c96b13e39774458779ee3e5c028e2f714d5742d3/contracts/vaults/VaultVenus.sol

PROJECT INFORMATION	
TITLE	PANCAKE BUNNY FINANCE
TYPE	DEFI
PLATFORM	BINANCE SMART CHAIN
CONTRACT ADDRESS	
REPOSITORY	https://github.com/PancakeBunny-finance/Bunny

VULNERABILITY ANALYSIS		
CRITICAL	0	No relevant provision
HIGH	0	No relevant provision
MEDIUM	2	Validation of address, Validation of amount
LOW	0	No relevant provision

CENTRALIZED FUNCTION	
harvest	Function for the auto compound. Accounts with keeper privileges can only execute
decreaseCollateral, increaseCollateral	Function to cope with the emergency of the vault. This function forcibly increase or decrease the Collateral
recoverToken	Function to withdraw wrongly deposited tokens into the contract

COMPANY PROPOSAL

Hexlant is a blockchain technology company that was founded in 2018. Security, network, and software experts from Samsung Electronics discovered security defects in smart contracts and blockchain protocols, so they established Hexlant to demonstrate the technical stability of the blockchain ecosystem.

Hexlant is building more than 20 blockchain mainnets directly to understand the blockchain operating system. Furthermore, we have developed vital security algorithms and mainnet monitoring technology. This method is applied and operated on all mainnet platforms owned by Hexlant, including Bitcoin, Ethereum, Polkadot, and Cardano(ADA).

Hexlant validates smart contract technology based on the experiences in technology operations above. In addition to error testing to detect bugs in smart contracts, we also provide blockchain technical guides to see problems in mainnet situations and continue to operate from a service perspective.

Hexlant's customers can receive services across blockchain technology, from vulnerability audits of contracts, owner key management, and blockchain wallet system establishment. Currently, more than 200 of our customers have started and operated blockchain businesses based on Hexlant's services and have achieved 12 trillion KRW in accumulated assets.

Initials for identification purposes:

For identification purposes
Hexlant.

CONTENTS

ANALYSIS PURPOSE

This report analyzes and summarizes the results of published contract codes to determine whether they meet the requirements and identify security vulnerabilities and problems that may arise in practice. This code analysis was conducted to verify the following factors:

- Proper operation of the implemented functions
- Security risks during the operation
- Preparation for the potential issues in off-chain transactions
- Readability and completeness of the contract codes

VULNERABILITY CLASSIFICATION

This vulnerability verification evaluates and classifies as below:

● Critical Severity

The critical-severity phase is a significant security flaw and causes fatal issues such as asset theft, freezing, and additional issuance. This defect must be corrected.

● High Severity

The high-severity phase is an item that can cause security defects due to special conditions and is strongly recommended for correction.

● Medium Severity

The medium-severity phase is not a security flaw but causes inefficient contract behavior. It is an item that is recommended modification to operate the contract efficiently.

● Low Severity

The low-severity phase is an item with no security issues but is recommended modifications to improve the contract structure.

BUNNY CONTRACT VULNERABILITY ANALYSIS		
● **CRITICAL**	0	No relevant provision
● **HIGH**	0	No relevant provision
● **MEDIUM**	2	Validation of address, Validation of amount
● **LOW**	0	No relevant provision

TEST RESULT

Code Coverage

Code coverage is a quantitative index of how much the written test has tested the functionality of the contact code.

It has additional calls that have not been made about the library and some of the features implemented in some contracts.

The coverage index below is the result that reflects the details above.

File Name	Statements	Functions	Lines
vaultVenus.sol	95.26%	100%	95.74%
vaultVenusBridge.sol	94.74%	100%	94.79%
safeVenus.sol	100%	100%	100%
vaultController.sol	100%	100%	100%

TEST CASE

Information below is a list of test cases that have actually applied. Ten categories and a total of 95 test scenarios have been used to examine.

Test Case 1. Individual Function (Test of 92 items)

DEPOSIT

1.1. deposit	Result	
Prevent execution, when staking token is wbnb	[✓ PASS]	FAIL
Allows execution when not paused	[✓ PASS]	FAIL
Prevents execution when paused	[✓ PASS]	FAIL
Returns zero to the related table when no deposit	[✓ PASS]	FAIL
Updates related table when deposit	[✓ PASS]	FAIL
Transfer deposited asset to venusBridge, increase vault's available value	[✓ PASS]	FAIL
Prevents unnecessary operation when the amount is zero	PASS	[✓ FAIL]

1.2. depositAll	Result	
Prevents execution, when staking token is wbnb	[✓ PASS]	FAIL
Allows execution when has some balance	[✓ PASS]	FAIL
Prevents unnecessary operation when the amount is zero	PASS	[✓ FAIL]

1.3. depositBNB	Result	
Prevents execution, when staking token is not wbnb	[✓ PASS]	FAIL
Allows execution when not paused	[✓ PASS]	FAIL
Prevents execution when paused	[✓ PASS]	FAIL
Returns zero to the related table when no deposit	[✓ PASS]	FAIL
Updates related table when deposit	[✓ PASS]	FAIL
Transfer deposited asset to venusBridge, increase vault's available value	[✓ PASS]	FAIL
Prevents unnecessary operation when the amount is zero	PASS	[✓ FAIL]

HARVEST

2.1. harvest	Result	
Prevents execution if not keeper	PASS	FAIL
Nothing changes, when no deposit	PASS	FAIL
Updates venus supply & borrow when having enough deposit	PASS	FAIL
Returns zero to supply when mintable usd value is less than 1e18	PASS	FAIL
Returns zero to borrow when borrowable value is less than 1 szabo	PASS	FAIL
Increase staked asset amount when protocol reward exists on venusBridge	PASS	FAIL
Prevents borrow when compound depth is zero	PASS	FAIL

WITHDRAW

3.1. withdrawAll	Result	
Returns zero when no deposit	PASS	FAIL
Returns deposited amount without withdrawal fee when no harvest & before 3 days	PASS	FAIL
Returns deposited amount when no harvest & after 3 days	PASS	FAIL
Returns deposited amount without withdrawal fee & profit(bunny) which is a value above performance fee when harvesting & before 3 days	PASS	FAIL
Returns deposited amount of bnb & profit(bunny) which is a value above performance fee when harvesting & after 3 days	PASS	FAIL
Returns deposited amount of bep20 & profit(bunny) which is a value above performance fee when harvesting & after 3 days	PASS	FAIL

3.2. withdrawUnderlying	Result	
Returns zero when no deposit	PASS	FAIL
Returns max. deposited amount without withdrawal fee when input exceeded deposit amount & before 3 days	PASS	FAIL
Returns max. deposited amount when input exceeded deposit amount & after 3 days	PASS	FAIL
Returns specified withdraw amount without withdrawal fee when before 3 days	PASS	FAIL
Decrease value of related table	PASS	FAIL
Prevents unnecessary operations when the amount is zero	PASS	FAIL

3.3. withdraw	Result	
Revert	PASS	FAIL

REWARD

4.1. getReward	Result	
Returns zero when no deposit	PASS	FAIL
Earns zero when deposit & no harvesting	PASS	FAIL
Returns zero and delete dust share table value when shares are less than dust	PASS	FAIL
Earns allotted interest value when harvesting	PASS	FAIL

RECOVER

5.1. recoverToken(VaultVenus)	Result	
Prevents execution if not owner	PASS	FAIL
Prevents recovery of staking token	PASS	FAIL
Recovers wrongly deposited token	PASS	FAIL

5.2. recoverToken(VaultVenusBridge)	Result	
Prevents execution if not owner	PASS	FAIL
Prevents recovery if related to all market’s token	PASS	FAIL
Allows recovery if related to all market’s staking token	PASS	FAIL
Prevents recovery when the amount exceeds the value without the market’s available value	PASS	FAIL
Recovers wrongly deposited tokens unrelated to market	PASS	FAIL
Allows wBNB recovery	PASS	FAIL

AUTH SET

6.1. VaultVenus	Result	
setReserveRatio. Prevents reserve ratio when more than the specified limit	PASS	FAIL
setReserveRatio. Prevents execution if not owner	PASS	FAIL
setReserveRatio. Allows reserve ratio when less than the specified limit	PASS	FAIL
setReserveRatio. Returns balance of the specified reserve percentage	PASS	FAIL
setCollateralFactors. Prevents when collateralFactors is higher than or equal to the specified emergency ratio	PASS	FAIL
setCollateralFactors. Prevents execution if not owner	PASS	FAIL
setCollateralFactors. Allows valid collateralFactors	PASS	FAIL

setCollateralFactors. Returns increased or decreased collateral ratio after set	PASS	FAIL
setVenusBridge. Prevents execution if not owner	PASS	FAIL
setVenusBridge. Prevents non-venusBridge contract	PASS	FAIL
setVenusBridge. Migrates BNB asset	PASS	FAIL
setVenusBridge. Migrates BEP20 asset	PASS	FAIL
setKeeper. Prevents execution if not keeper	PASS	FAIL
setMinter. Prevents execution if not owner	PASS	FAIL
setMinter. Prevents non-bunny minter contract	PASS	FAIL
setBunnyChef. Prevents execution if not owner	PASS	FAIL
setBunnyChef. Prevents non-bunny chef contract	PASS	FAIL
setWhitelist. Prevent execution if not owner	PASS	FAIL
disableWhitelist. Prevent execution if not owner	PASS	FAIL
transferOwnership. Prevents execution if not owner	PASS	FAIL
renounceOwnership. Prevents execution if not owner	PASS	FAIL
setPaused. Prevents execution if not owner	PASS	FAIL

6.2. VaultVenusBridge	**Result**	
migrateTo. Prevents execution if not whitelisted account	PASS	FAIL
deposit. Prevents execution if not whitelisted account	PASS	FAIL
withdraw. Prevents execution if not whitelisted account	PASS	FAIL

6.3. BunnyMinterV2	**Result**	
setWithdrawalFee. Prevents execution if not owner	PASS	FAIL
setPerformanceFee. Prevents execution if not owner	PASS	FAIL
setWithdrawalFeeFreePeriod. Prevents execution if not owner	PASS	FAIL

COLLATERAL

7.1. decreaseCollateral	**Result**	
BNB, Removes all collateral when max. value & supply are not injected	PASS	FAIL
BNB, Removes all collateral & returns supply value when max. value & supply are injected	PASS	FAIL
BEP20, Removes all collateral when max. value & supply are not injected	PASS	FAIL
BEP20, Removes all collateral, return supply value when max. value & supply are injected	PASS	FAIL
Prevent execution if not keeper	PASS	FAIL

7.2. increaseCollateral	Result	
BNB, Returns increased collateral	PASS	FAIL
BEP20, Returns increased collateral	PASS	FAIL
Prevents execution if not keeper	PASS	FAIL

UTILIZE

8.1. SafeVenus	Result	
safeMintAmount, Returns increased mintable value when deposit	PASS	FAIL
safeMintAmount, Returns zero mintable value when no deposit	PASS	FAIL
safeBorrowAndRedeemAmount, Returns zero when deposit & no harvesting	PASS	FAIL
safeBorrowAndRedeemAmount, Returns increased borrowable value, redeemable value when having enough deposit & harvesting	PASS	FAIL
venusBorrowAndSupply, Returns zero when deposit & no harvesting	PASS	FAIL
venusBorrowAndSupply, Returns increased supply & borrow when deposit & harvesting	PASS	FAIL

Test Case 2. Complex environment (Test of three items)

MULTI_PLAYERS

9.1. Multi players	Result	
Returns allotted deposit, reward & withdrawal amount	PASS	FAIL

REENTRANT

10.1. Withdraw	Result	
Prevents withdrawAll	PASS	FAIL
Prevents withdrawUnderlying	PASS	FAIL

VULNERABILITY ANALYSIS

A total of two items need to be modified.

➢ PANCAKE BUNNY-FINANCE - 01 : setBunnyChef

Type	Severity	Location
Validation of address	● **Medium**	VaultVenus.sol L211

· Description : VaultVenus.setBunnyChef can be saved only once and cannot be changed after. The bunnyChef contract plays an important role in collecting additional bunny payments and events.

· Recommendation :

1. Additional validation of the bunnyChef contract

2. Change to hard-coded
   The bunnyChef contract seems unchanged. You can initialize the vaultVenus contract by hard-coding the bunnyChef address prior to deployment.

3. Change to be modifiable

➢ PANCAKE BUNNY-FINANCE - 02 : deposit, depositBNB, withdrawUnderlying

Type	Severity	Location
Validation of amount	● **Medium**	VaultVenus.sol L266 VaultVenus.sol L292 VaultVenus.sol L365

· Description : These lines are respectively deposit(uint amount), depositBNB(), withdrawUnderlying(). If the amount or msg.value is 0, it is necessary to prevent operation and events occurring.

· Recommendation :

1. Add require(amount > 0) or require(msg.value > 0) before proceeding main logic

Declare

The report is based on Hexlant's smart contract security audit results. This report does not warrant the suitability of the business model, legal regulation, or investment opinion. In addition to the problems described in the report, there may be undiscovered problems, including mainnet technology or virtual machines. This report is intended for discussion purposes only.

Hexlant.

-

contact@hexlant.com
www.hexlant.com