# Smart Contract Security Audit Report

[2021]

# Table Of Contents

# 1 Executive Summary

On 2021.05.31, the SlowMist security team received the Mars team's security audit application for Mars Ecosystem, developed the audit plan according to the agreement of both parties and the characteristics of the project, and finally issued the security audit report.

The SlowMist security team adopts the strategy of "white box lead, black, grey box assists" to conduct a complete security test on the project in the way closest to the real attack.

The test method information:

| Test method | Description |
|---|---|
| Black box testing | Conduct security tests from an attacker's perspective externally. |
| Grey box testing | Conduct security testing on code modules through the scripting tool, observing the internal running status, mining weaknesses. |
| White box testing | Based on the open source code, non-open source code, to detect whether there are vulnerabilities in programs such as nodes, SDK, etc. |

The vulnerability severity level information:

| Level | Description |
|---|---|
| Critical | Critical severity vulnerabilities will have a significant impact on the security of the DeFi project, and it is strongly recommended to fix the critical vulnerabilities. |
| High | High severity vulnerabilities will affect the normal operation of the DeFi project. It is strongly recommended to fix high-risk vulnerabilities. |
| Medium | Medium severity vulnerability will affect the operation of the DeFi project. It is recommended to fix medium-risk vulnerabilities. |
| Low | Low severity vulnerabilities may affect the operation of the DeFi project in certain scenarios. It is suggested that the project party should evaluate and consider whether these vulnerabilities need to be fixed. |
| Weakness | There are safety risks theoretically, but it is extremely difficult to reproduce in engineering. |

| Level | Description |
|---|---|
| Suggestion | There are better practices for coding or architecture. |

## 2 Audit Methodology

The security audit process of SlowMist security team for smart contract includes two steps:

Smart contract codes are scanned/tested for commonly known and more specific vulnerabilities using automated analysis tools.

Manual audit of the codes for security issues. The contracts are manually analyzed to look for any potential problems.

Following is the list of commonly known vulnerabilities that was considered during the audit of the smart contract:

- Reentrancy Vulnerability
- Replay Vulnerability
- Reordering Vulnerability
- Short Address Vulnerability
- Denial of Service Vulnerability
- Transaction Ordering Dependence Vulnerability
- Race Conditions Vulnerability
- Authority Control Vulnerability
- Integer Overflow and Underflow Vulnerability
- TimeStamp Dependence Vulnerability
- Uninitialized Storage Pointers Vulnerability
- Arithmetic Accuracy Deviation Vulnerability
- tx.origin Authentication Vulnerability

- "False top-up" Vulnerability
- Variable Coverage Vulnerability
- Gas Optimization Audit
- Malicious Event Log Audit
- Redundant Fallback Function Audit
- Unsafe External Call Audit
- Explicit Visibility of Functions State Variables Aduit
- Design Logic Audit
- Scoping and Declarations Audit

# 3 Project Overview

## 3.1 Project Introduction

Project: Mars Ecosystem,

Type: DeFi

Module: Bondingcurve + core + genesis + oracle + pcv + refs + redemption

Code lines: 2206

Complexity: Medium

Workload： 11 working days, smart contract security audit standards, related instructions and certificate query:

https://www.slowmist.com/service-smart-contract-security-audit.html

Source code:

https://github.com/MarsEcosystem/mars-swap

Initial audit commit: 313bf2b4ad943f7c9216aa7c33abfb33cda88551

Last audit commit: c0dad9789a9713ccdf6abfe56d4bddc344616746

## 3.2 Vulnerability Information

The following is the status of the vulnerabilities found in this audit:

| NO | Title | Category | Level | Status |
|---|---|---|---|---|
| N1 | initialize function doesn't check the role that may be called by others perviously | Authority Control Vulnerability | Low | Fixed |
| N2 | purchase/deposit functions has payable maybe lock user's assets | Unsafe External Call Audit | Suggestion | Fixed |
| N3 | redeem may be called by others | Authority Control Vulnerability | Suggestion | Ignored |
| N4 | Suggest that add call authority check to the launch function | Authority Control Vulnerability | Suggestion | Fixed |
| N5 | PCVController role has too much authority that can withdraw assets from the contracts | Authority Control Vulnerability | Medium | Fixed |
| N6 | getUSDMAmountGovernance loop maybe call DoS | Others | Suggestion | Ignored |

# 4 Code Overview

## 4.1 Contracts Description

The main network address of the contract is as follows:

| Module | Address |
|---|---|

| Module | Address |
|---|---|
| Core | 0x00789Cfb69499c65ac9A3a68fb4917c9b4FcA2a7 |
| XMS | 0x7859B01BbF675d67Da8cD128a50D155cd881B576 |
| IMO | 0x243DDd2E42CEb93349E726e2367EDeC6339aba75 |
| MarsSwapFactory | 0x6f12482D9869303B998C54D91bCD8bCcba81f3bE |
| MarsSwapRouter | 0xb68825C810E67D4e444ad5B9DeB55BA56A66e72D |
| MarsStake | 0x3b550BBFaC32Ec434F858a8135fa17C40636583B |
| AirDrop | 0x01D152fF991E76b6cb310387c07cAfdFda790a25 |
| Governor (timelock) | 0xC35a8BdBB93abFAb362aF6dC3383cD2c6aEA6cBc |

## 4.2 Visibility Description

The SlowMist Security team analyzed the visibility of major contracts during the audit, the result as follows:

| BondingLCurve | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | internal | can modify state | - |
| setIncentiveAmount | external | can modify state | onlyGovernor |
| setMaxBasisPointsFromPegLP | external | can modify state | onlyGovernor |
| setAllocation | external | can modify state | onlyGovernor |
| setOracle | external | can modify state | onlyGovernor |
| allocate | external | can modify state | postGenesis,whenNotPaused,validPriceRange |

**BondingLCurve**

| | | | |
|---|---|---|---|
| getCurrentPrice | public | - | - |
| getAmountOut | public | - | - |
| getTotalPCVHeld | public | - | - |
| _purchase | internal | can modify state | - |
| _incentivize | internal | can modify state | - |
| _isValidPriceRange | internal | - | - |
| _ignoreUSDMSupplyCap | internal | - | - |

**BUSDBondingLCurve**

| Function Name | Visibility | Mutability | Modifiers |
|---|---|---|---|
| constructor | public | can modify state | - |
| initialize | external | can modify state | initializer |
| setOracle | external | can modify state | onlyGovernor |
| getCurrentPrice | public | - | - |
| getAmountOut | public | - | - |
| getAmountIn | public | - | - |
| setFee | public | can modify state | onlyGovernor |
| purchase | external | payable | postGenesis,whenNotPaused,ensure |
| getTotalPCVHeld | public | - | - |
| _allocateSingle | internal | can modify state | - |

| BUSDBondingLCurve | | | |
|---|---|---|---|
| _isValidPriceRange | internal | - | - |
| _ignoreUSDMSupplyCap | internal | - | - |
| setChainlink | external | can modify state | onlyGovernor |

| BNBBondingLCurve | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | public | can modify state | - |
| initialize | external | can modify state | initializer |
| getCurrentPrice | public | - | - |
| getAmountOut | public | - | - |
| getAmountIn | public | - | - |
| setFee | public | can modify state | onlyGovernor |
| purchase | external | payable | postGenesis,whenNotPaused,ensure |
| getTotalPCVHeld | public | - | - |
| _allocateSingle | internal | can modify state | - |
| _isValidPriceRange | internal | - | - |
| _ignoreUSDMSupplyCap | internal | - | - |
| setChainlink | external | can modify state | onlyGovernor |

| Permissions | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |

| Permissions | | | |
|---|---|---|---|
| constructor | public | can modify state | - |
| createRole | external | can modify state | onlyGovernor |
| grantMinter | external | can modify state | onlyGovernor |
| grantBurner | external | can modify state | onlyGovernor |
| grantPCVController | external | can modify state | onlyGovernor |
| grantGovernor | external | can modify state | onlyGovernor |
| grantGuardian | external | can modify state | onlyGovernor |
| revokeMinter | external | can modify state | onlyGovernor |
| revokeBurner | external | can modify state | onlyGovernor |
| revokePCVController | external | can modify state | onlyGovernor |
| revokeGovernor | external | can modify state | onlyGovernor |
| revokeGuardian | external | can modify state | onlyGovernor |
| revokeOverride | external | can modify state | onlyGuardian |
| isMinter | external | - | - |
| isBurner | external | - | - |
| isPCVController | external | - | - |
| isGovernor | public | - | - |
| isGuardian | public | - | - |
| _setupGovernor | internal | can modify state | - |
| _setupMinter | internal | can modify state | - |

| Permissions | | | |
|---|---|---|---|
| _setupBurner | internal | can modify state | - |

| Core | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| init | external | can modify state | initializer |
| setXMSSupportRatio | external | can modify state | onlyGovernor |
| setUSDM | external | can modify state | onlyGovernor |
| setXMS | external | can modify state | onlyGovernor |
| setGenesisGroup | external | can modify state | onlyGovernor |
| allocateXMS | external | can modify state | onlyGovernor |
| allocateToken | public | can modify state | onlyGovernor |
| approveXMS | public | can modify state | onlyGovernor |
| approveToken | public | can modify state | onlyGovernor |
| completeGenesisGroup | external | can modify state | - |
| getApprovedPairsLength | public | - | - |
| getApprovedContractsLength | public | - | - |
| setApprovedPairAndContract | public | can modify state | onlyGovernor |
| removeApprovedPairAndContract | public | can modify state | onlyGovernor |
| _setXMSSupportRatio | internal | can modify state | - |
| _setUSDM | internal | can modify state | - |

| Core | | | |
|---|---|---|---|
| _setXMS | internal | can modify state | - |

| IDO | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | public | can modify state | - |
| deploy | external | can modify state | onlyGenesisGroup |
| removeLiquidity | external | can modify state | onlyGuardianOrGovernor |

| BUSDGenesisGroup | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | public | can modify state | - |
| initGenesis | external | can modify state | onlyGovernor |
| purchase | external | payable | duringTime |
| redeem | external | can modify state | - |
| launch | external | can modify state | nonContract,afterTime |
| emergencyExit | external | can modify state | - |
| getAmountsToRedeem | public | - | postGenesis |
| getAmountOut | public | - | - |
| _burnFrom | internal | can modify state | - |
| _usdmXMSExchangeRate | internal | - | - |
| _getEffectiveMGEN | internal | - | - |

| BUSDGenesisGroup | | | |
|---|---|---|---|
| setChainlink | external | can modify state | onlyGovernor |

| CombinationOracle | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | internal | can modify state | - |
| initialize | external | can modify state | initializer |
| isStale | public | - | - |
| canUpdate | public | - | - |
| update | external | can modify state | - |
| consult | public | - | - |
| setRouter | public | can modify state | onlyGovernor |
| getRouter | public | - | - |
| _initialize | internal | can modify state | - |
| _canUpdate | internal | - | - |
| _isStale | internal | - | - |
| _update | internal | can modify state | - |
| _consult | internal | - | - |

| BNBLastPriceOracle | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | public | can modify state | - |

| **BNBLastPriceOracle** | | | |
|---|---|---|---|
| getLatestPrice | public | - | - |

| **BUSDLastPriceOracle** | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | public | can modify state | - |
| getLatestPrice | public | - | - |

| **OracleIncentives** | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | public | can modify state | - |
| updateXMSForUSDMMROracle | public | can modify state | whenNotPaused |
| updateXMSForUSDMSupplyCapOracle | public | can modify state | whenNotPaused |
| _incentivize | internal | can modify state | - |
| setIncentiveAmount | external | can modify state | onlyGovernor |

| **SwapMiningOracle** | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | public | can modify state | - |
| getPairsLength | external | - | - |
| setFactory | external | can modify state | onlyGovernor |
| setPeriod | external | can modify state | onlyGovernor |

| **SwapMiningOracle** | | | |
|---|---|---|---|
| addPair | external | can modify state | onlyGovernor |
| removePair | external | can modify state | onlyGovernor |
| update | external | can modify state | - |
| consult | external | - | - |

| **MarsSwapPairCombOracle** | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | public | can modify state | - |
| setFactory | external | can modify state | onlyGovernor |
| setPeriod | external | can modify state | onlyGovernor |
| setConsultLeniency | external | can modify state | onlyGovernor |
| setAllowStaleConsults | external | can modify state | onlyGovernor |
| _initialize | internal | can modify state | - |
| _canUpdate | internal | - | - |
| _isStale | internal | - | - |
| _update | internal | can modify state | - |
| _consult | internal | - | - |

| **PCVController** | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | internal | can modify state | - |

| PCVController | | | |
|---|---|---|---|
| forceWithdraw | external | can modify state | onlyGovernor |
| setPCVDeposit | external | can modify state | onlyGovernor |
| _withdraw | internal | can modify state | - |

| BUSDUniswapPCVController | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | public | can modify state | - |
| depositLpMining | external | can modify state | onlyGovernor |
| harvest | external | can modify state | onlyGovernor |
| withdrawLpMining | external | can modify state | onlyGuardianOrGovernor |
| removeLiquidity | external | can modify state | onlyGuardianOrGovernor |
| _removeLiquidity | internal | can modify state | - |
| _harvest | internal | can modify state | - |
| _depositLpMining | internal | can modify state | - |
| _withdrawLpMining | internal | can modify state | - |
| setLpMiningMaster | external | can modify state | onlyGovernor |

| PCVVenusDeposit | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | internal | can modify state | - |
| withdraw | external | can modify state | onlyPCVController |

| PCVVenusDeposit | | | |
|---|---|---|---|
| leaveSupply | external | can modify state | onlyPCVController |
| harvest | external | can modify state | onlyPCVController |
| _supply | internal | can modify state | - |
| _leaveSupply | internal | can modify state | - |
| _harvest | internal | can modify state | - |
| _transferWithdrawn | internal | can modify state | - |

| BNBVenusPCVDeposit | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | public | can modify state | - |
| receive | external | payable | - |
| deposit | external | payable | postGenesis,whenNotPaused |
| _supply | internal | can modify state | - |
| _leaveSupply | internal | can modify state | - |
| _harvest | internal | can modify state | - |
| _transferWithdrawn | internal | can modify state | - |

| PCVSplitter | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | internal | can modify state | - |
| checkAllocation | public | - | - |

| PCVSplitter | | | |
|---|---|---|---|
| getAllocation | public | - | - |
| _allocateSingle | internal | can modify state | - |
| _setAllocation | internal | can modify state | - |
| _allocate | internal | can modify state | - |

| PCVUniswapDeposit | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | internal | can modify state | - |
| withdraw | external | can modify state | onlyPCVController |
| removeLiquidity | external | can modify state | onlyPCVController |
| harvest | external | can modify state | onlyPCVController |
| depositLpMining | external | can modify state | onlyPCVController |
| withdrawLpMining | external | can modify state | onlyPCVController |
| _addLiquidity | internal | can modify state | - |
| _removeLiquidity | internal | can modify state | - |
| _harvest | internal | can modify state | - |
| _depositLpMining | internal | can modify state | - |
| _withdrawLpMining | internal | can modify state | - |
| _getLpMiningPid | internal | - | - |
| _transferWithdrawn | internal | can modify state | - |

| PCVUniswapDeposit | | | |
|---|---|---|---|
| setLpMiningMaster | external | can modify state | onlyPCVController |

| BUSDUniswapPCVDeposit | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | public | can modify state | - |
| deposit | external | payable | postGenesis,whenNotPaused |
| _getAmountUSDMToDeposit | internal | - | - |
| _addLiquidity | internal | can modify state | - |
| _removeLiquidity | internal | can modify state | - |
| _harvest | internal | can modify state | - |
| _depositLpMining | internal | can modify state | - |
| _withdrawLpMining | internal | can modify state | - |
| _transferWithdrawn | internal | can modify state | - |
| _getLpMiningPid | internal | - | - |
| setChainlink | external | can modify state | onlyGovernor |

| RedemptionUnit | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | internal | can modify state | - |
| fee | external | - | - |
| feePrecision | external | - | - |

| **RedemptionUnit** | | | |
|---|---|---|---|
| _purchase | internal | can modify state | - |
| purchase | external | payable | postGenesis,whenNotPaused,ensure |

| **XMSRedemptionUnit** | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | public | can modify state | - |
| getCurrentPrice | public | - | - |
| getAmountOut | public | - | - |
| getAmountIn | public | - | - |
| _purchase | internal | can modify state | - |
| getTotalAssetHeld | public | - | - |
| setFee | public | can modify state | onlyGovernor |

| **UniAndOracleRef** | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | internal | can modify state | - |
| setPair | external | can modify state | onlyGovernor |
| setRouter | external | can modify state | onlyGovernor |
| setFactory | external | can modify state | onlyGovernor |
| token | public | - | - |
| getReserves | public | - | - |

| UniAndOracleRef | | | |
|---|---|---|---|
| liquidityOwned | public | - | - |
| _approveToken | internal | can modify state | - |
| _setupPair | internal | can modify state | - |
| _setupRouter | internal | can modify state | - |
| _setupFactory | internal | can modify state | - |
| _isPair | internal | - | - |
| _getUniswapPrice | internal | - | - |

| CoreRef | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | internal | can modify state | - |
| setCore | external | can modify state | onlyGovernor |
| pause | public | can modify state | onlyGuardianOrGovernor |
| unpause | public | can modify state | onlyGuardianOrGovernor |
| core | public | - | - |
| usdm | public | - | - |
| xms | public | - | - |
| usdmBalance | public | - | - |
| xmsBalance | public | - | - |
| getUSDMAmountGovernance | public | - | - |

| CoreRef | | | |
|---|---|---|---|
| _burnUSDMHeld | internal | can modify state | - |
| _mintUSDM | internal | can modify state | - |

| OracleRef | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | internal | can modify state | - |
| setXMSForUSDMMROracle | external | can modify state | onlyGovernor |
| setXMSForUSDMSupplyCapOracle | external | can modify state | onlyGovernor |
| invert | public | - | - |
| getXMSPrice | public | - | - |
| getUSDMSupplyCap | public | - | - |
| _setXMSForUSDMMROracle | internal | can modify state | - |
| _setXMSForUSDMSupplyCapOracle | internal | can modify state | - |

| UniRef | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| constructor | internal | can modify state | - |
| setPair | external | can modify state | onlyGovernor |
| setRouter | external | can modify state | onlyGovernor |
| setFactory | external | can modify state | onlyGovernor |
| token | public | - | - |

| UniRef | | | |
|---|---|---|---|
| getReserves | public | - | - |
| liquidityOwned | public | - | - |
| _approveToken | internal | can modify state | - |
| _setupPair | internal | can modify state | - |
| _setupRouter | internal | can modify state | - |
| _setupFactory | internal | can modify state | - |
| _isPair | internal | - | - |
| _getUniswapPrice | internal | - | - |

## 4.3 Vulnerability Summary

**[N1] [Low] initialize function doesn't check the role that may be called by others perviously**

**Category: Authority Control Vulnerability**

**Content**

contracts/bondingcurve/BNBBondingLCurve.sol#53-59,

```
function initialize(address _wbnb, address _chainlink)
    external
    initializer
{
    wbnb = IERC20(_wbnb);
    chainlink = IChainlinkLastPriceOracle(_chainlink);
}
```

contracts/bondingcurve/BUSDBondingLCurve.sol#55-61,

```
function initialize(address _busd, address _chainlink)
    external
    initializer
{
    busd = IERC20(_busd);
    chainlink = IChainlinkLastPriceOracle(_chainlink);
}
```

This function only has one initializer modifier, but the initializer only check whether the contract was intialized, so it can be called by others.

the same points:

contracts/core/Core.sol#40-42,

```
function init() external override initializer {
    _setupGovernor(msg.sender);
}
```

contracts/oracle/CombinationOracle.sol#40-45,

```
function initialize(address _oracle, address[] memory _path)
    external
    initializer
{
    setRouter(_oracle, _path);
}
```

**Solution**

add onlyOwner / onlyGovernor check

**Status**

Fixed

**[N2] [Suggestion] purchase/deposit functions has payable maybe lock user's assets**

**Category: Unsafe External Call Audit**

**Content**

contracts/genesis/BUSDGenesisGroup.sol#76-87,

```
function purchase(address to, uint256 value)
    external









    payable
    override
    duringTime
{
    require(value != 0, "BUSDGenesisGroup::purchase: No value sent");
    busd.transferFrom(msg.sender, address(this), value);
    _mint(to, value);

    emit Purchase(to, value);
}
```

Has payable but never use it.

contracts/pcv/BNBVenusPCVDeposit.sol#39,

```
receive() external payable {}
```

contracts/pcv/BUSDUniswapPCVDeposit.sol#40-54,

```
function deposit(uint256 busdAmount)
    external
    payable
    override
    postGenesis
    whenNotPaused
{
    busdAmount = busd.balanceOf(address(this)); // Include any BUSD dust from prior
LP

    _addLiquidity(busdAmount);

    _burnUSDMHeld(); // Burn any USDM dust from LP
```

```
        emit Deposit(msg.sender, busdAmount);
    }
```

**Solution**

remove payable.

**Status**

Fixed;

### [N3] [Suggestion] redeem may be called by others

**Category: Authority Control Vulnerability**

**Content**

contracts/genesis/BUSDGenesisGroup.sol#91-132,

```
function redeem(address to) external override {
    (uint256 usdmAmount, uint256 genesisXMS, uint256 busdAmount) =
        getAmountsToRedeem(to);
    require(
        block.number > launchBlock,
        "BUSDGenesisGroup::redeem: No redeeming in launch block"
    );

    // Burn MGEN
    uint256 amountIn = balanceOf(to);
    _burnFrom(to, amountIn);

    // Send USDM and XMS and BUSD
    if (usdmAmount != 0) {
        usdm().transfer(to, usdmAmount);
    }
    if (genesisXMS != 0) {
        uint256 genesis20Percent = genesisXMS.mul(2).div(10);
        xms().transfer(to, genesis20Percent);
        address tokenTimelockDelegator =
            address(
                new StraightTokenTimelockDelegator(
                    address(xms()),
                    to,
                    3600 * 24 * 30 * 12
                )
            );
        tokenTimelockDelegators[to] = tokenTimelockDelegator;
        xms().transfer(
            tokenTimelockDelegator,
            genesisXMS.sub(genesis20Percent)
        );
        ILinearTokenTimelock(tokenTimelockDelegator).initialize(
            launchTimestamp
        );
    }
    if (busdAmount != 0) {
        busd.transfer(to, busdAmount);
    }

    emit Redeem(to, amountIn, usdmAmount, genesisXMS);
}
```

The redeem function is that redeem the assets of to address, it can by called by others, maybe it's not 'to' address's desire.

**Solution**

Check if the sender is 'to' address

**Status**

Ignored; This mechanism has already been tested by other well-known protocols.

### [N4] [Suggestion] Suggest that add call authority check to the launch function

**Category: Authority Control Vulnerability**

**Content**

contracts/genesis/BUSDGenesisGroup.sol#135-180,

```
function launch() external override nonContract afterTime {
    // Complete Genesis
    core().completeGenesisGroup();
    launchBlock = block.number;
    launchTimestamp = block.timestamp;
```

```
        (totalEffectiveMGEN, supersuper) = _getEffectiveMGEN(totalSupply());

        uint256 endOfTime = uint256(-1);
        // Bonding curve purchase and PCV allocation
        bondingCurve.purchase(address(this), totalEffectiveMGEN, 0, endOfTime);
        bondingCurve.allocate();

        ido.deploy(_usdmXMSExchangeRate());

        // solhint-disable-next-line not-rely-on-time
        emit Launch(block.timestamp);
    }
```

It can be called by others, may exhibit unexpected behavior, especially the launch function has ido.deploy operation.

**Solution**

Add call authority check.

**Status**

Fixed;

### [N5] [Medium] PCVController role has too much authority that can withdraw assets from the contracts

**Category: Authority Control Vulnerability**

**Content**

contracts/pcv/PCVVenusDeposit.sol#27-34,

```
    function withdraw(address to, uint256 amountUnderlying)
        external
        override
        onlyPCVController
    {
        _transferWithdrawn(to, amountUnderlying);
        emit Withdrawal(msg.sender, to, amountUnderlying);
    }
```

contracts/pcv/BNBVenusPCVDeposit.sol#87-90,

```
function _transferWithdrawn(address to, uint256 amount) internal override {
    (bool success, ) = to.call{value: amount}("");
    require(success, "BNBVenusPCVDeposit::_transferWithdrawn: Transfer failed");
}
```

There is onlyPCVController check, but the role has too much authority that can withdraw assets from the contracts especially when the private key was stolen.

**Solution**

Move the authority to the governor, add timelock.

**Status**

Fixed; PCVController is default empty, onlyGovernor can change it, and onlyGovernor is a Timelock contract now.

### [N6] [Suggestion] getUSDMAmountGovernance loop maybe call DoS

**Category: Others**

**Content**

contracts/refs/CoreRef.sol#157-176,

```
function getUSDMAmountGovernance()
    public
    view
    override
    returns (uint256 usdmAmount)
{
    address pair;
    address _contract;
    for (uint256 i; i < core().getApprovedPairsLength(); i++) {
        pair = core().approvedPairs(i);
        for (uint256 j; j < core().getApprovedContractsLength(pair); j++) {
            _contract = core().approvedContracts(pair, j);
            usdmAmount += core()
                .usdm()
                .balanceOf(pair)
                .mul(IERC20(pair).balanceOf(_contract))
                .div(IERC20(pair).totalSupply());
        }
        }
    }
```

The function calls external contracts in the loop, maybe failed when core().getApprovedPairsLength() is too long.

**Solution**

Control getApprovedPairsLength() is in the range, or execute in batches.

**Status**

Ignored; The types of USDM LP tokens that are managed by the protocol are limited, currently there are only two liquidity pairs — XMS/USDM and BUSD/USDM.

## 5 Audit Result

| Audit Number | Audit Team | Audit Date | Audit Result |
|---|---|---|---|
| 0X002106110004 | SlowMist Security Team | 2021.05.31 - 2021.06.14 | Passed |

Summary conclusion: The SlowMist security team use a manual and SlowMist team's analysis tool to audit the project, during the audit work we found 1 medium risk, 1 low risk, 4 suggestion vulnerabilities. And 2 suggestion vulnerabilities were ignored; All other findings were fixed.

# 6 Statement

SlowMist issues this report with reference to the facts that have occurred or existed before the issuance of this report, and only assumes corresponding responsibility based on these.

For the facts that occurred or existed after the issuance, SlowMist is not able to judge the security status of this project, and is not responsible for them. The security audit analysis and other contents of this report are based on the documents and materials provided to SlowMist by the information provider till the date of the insurance report (referred to as "provided information"). SlowMist assumes: The information provided is not missing, tampered with, deleted or concealed. If the information provided is missing, tampered with, deleted, concealed, or inconsistent with the actual situation, the SlowMist shall not be liable for any loss or adverse effect resulting therefrom. SlowMist only conducts the agreed security audit on the security situation of the project and issues this report. SlowMist is not responsible for the background and other conditions of the project.