Jan 5, 2023

ZAN

Security Assessment

Morpheus - Sleepee-smc

—

Express Service

Table of Contents

1. Overview
    - 1.1. Project Summary
    - 1.2. Assessment Summary
    - 1.3. Assessment Scope
2. Checklist
3. Findings
    - 3.1. H-01 | Unauthorized Token Mint
    - 3.2. M-02 | Incompliant to EIP-712 Signature Encoding
    - 3.3. I-03 | Function Visibility Can Be External
    - 3.4. I-04 | Code layout Conventions
    - 3.5. I-05 | Interface Defined But Not Inherited
    - 3.6. I-06 | Prefer uint256 For Variables
    - 3.7. I-07 | Variables Should Be Constants
    - 3.8. I-08 | No Check of Address Params with Zero Address
4. Disclaimer
5. Appendix

1. Overview

1.1. Project Summary

Project Name	Morpheus - Sleepee-smc
Platform	Ethereum
Language	Solidity
Code Repository	https://github.com/sagara11/Sleepee-smc
Commit	700ad6b7de6cbcc34cac8cf5c53673f838688b16

1.2. Assessment Summary

Delivery Date	Jan. 5th, 2023
Audit Methodology	Static Analysis, Formal Verification

1.3. Assessment Scope

ID	File
01	contracts/AnyswapV5ERC20.sol
02	contracts/IAnySwapV5ERC20.sol
03	contracts/withdrawWrap.sol

2. Checklist

2.1. General Vulnerability

Reentrancy	DelegateCall
Integer Overflow	Input Validation
Unchecked this.call	Frozen Money
Arbitrary External Call	Unchecked Owner Transfer
Do-while Continue	Right-To-Left-Override Character
Unauthenticated Storage Access	Risk For Weak Randomness
TxOrigin	Missing Checks for Return Values
Diamond Inheritance	ThisBalance
VarType Deduction	Array Length Manipulation
Uninitialized Variable	Shadow Variable
Divide Before Multiply	Function Not Working

2.2. Code Conventions

Compiler Version	Improper State Variable Modification
Function Visibility	Deprecated Function
Externally Controlled Variables	Code Style
Constant Specific	Event Specific
Return Value Unspecified	Nonexistent Error Message
Reference Variable Specification	Import Issue
Compare With Timestamp/Block Number/Blockhash	Constructor in Base Contract Not Implemented
Delete Struct Containing the Mapping Type	Usage of '=+'
Paths in the Modifier Not End with "_" or Revert	Non-payable Public Functions Use msg.value
SafeMath Issue	Compiler Error/Warning
ERC20/ERC721/ERC1155 Standard Specification	Anti-reentry Lock Specific
Nested Function Calls	Inheritance Issue
Signature Replay Risk	Missing Event

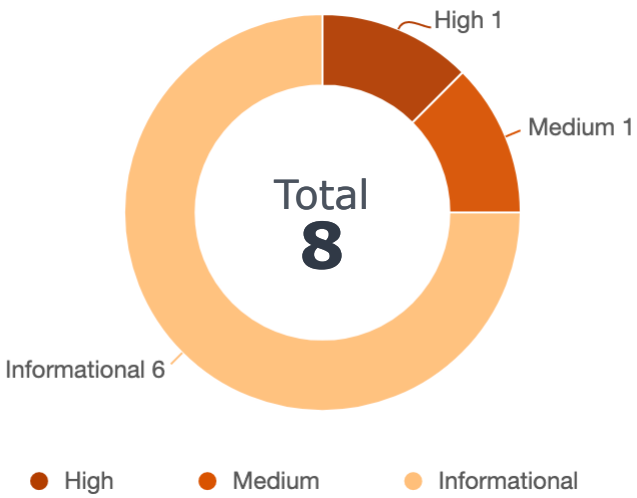
2.3. Gas Optimization

Tautology Issue	Loop Depends on Array Length
Redundant/Duplicated/Dead Code	Code Complexity/Code Inefficiency
Undeclared Resource	Optimizable Return Statement
Unused Resource	Duplicate Code

2.4. Compiler Bug

Affected by Compiler Bug

3. Findings



ID	Title	Category	Severity
H-01	Unauthorized Token Mint	General Vulnerability	High
M-02	Incompliant to EIP-712 Signature Encoding	General Vulnerability	Medium
I-03	Function Visibility Can Be External	Code Conventions	Informational
I-04	Code layout Conventions	Code Conventions	Informational
I-05	Interface Defined But Not Inherited	Code Conventions	Informational
I-06	Prefer uint256 For Variables	Gas Optimization	Informational
I-07	Variables Should Be Constants	Gas Optimization	Informational
I-08	No Check of Address Params with Zero Address	Code Conventions	Informational

H-01 | Unauthorized Token Mint

High : General Vulnerability

File Location : contracts/AnyswapV5ERC20.sol:331

Description

Function mint is open to all users without authentication.

```
331  function mint(address to, uint256 amount) external returns (bool) {
332      _mint(to, amount);
333      return true;
334  }
```

Recommendation

Add onlyAuth() modifier to mint function.

M-02 | Incompliant to EIP-712 Signature Encoding

Medium : General Vulnerability

File Location : contracts/AnyswapV5ERC20.sol:728

Description

According to EIP-712 (https://eips.ethereum.org/EIPS/eip-712#transactions-and-bytestrings), the personal sign should be encoded as:
encode(b : $\mathbb{B}^{8n}$) = "\x19Ethereum Signed Message:\n" ‖ len(b) ‖ b where len(b) is the ascii-decimal encoding of the number of bytes in b.
However, the verifyPersonalSignfunction uses a constant 32 as length to encode 64 bytes message, which violates the EIP-712 specification.

```
726  keccak256(
727      abi.encodePacked(
728          "\x19Ethereum Signed Message:\n32",
729          DOMAIN_SEPARATOR,
730          hash
731      )
732  );
```

Recommendation

We recommend to strictly follow the EIP-712 specification and change length to 64 as follows:

```
1  keccak256(
2      abi.encodePacked(
3          "\x19Ethereum Signed Message:\n64",
4          DOMAIN_SEPARATOR,
5          hash
6      )
7  );
```

I-03 | Function Visibility Can Be External

Informational : Code Conventions

File Location : contracts/withdrawWrap.sol:107,111,115
contracts/AnyswapV5ERC20.sol:260,323,346,356

Description

Functions that are not called should be declared as external.

contracts/withdrawWrap.sol

```
function updateThreshold(uint256 _threshold) public onlyOwner {

function updateAuthorize(address _owner, bool _allowance) public onlyOwner {

function updateFee(feeSystem calldata _fee) public onlyOwner {
```

contracts/AnyswapV5ERC20.sol

```
 function owner() public view returns (address) {

 function changeMPCOwner(address newVault) public onlyVault returns (bool) {

 function Swapin(
     bytes32 txhash,
     address account,
     uint256 amount
 ) public onlyAuth returns (bool) {

 function Swapout(uint256 amount, address bindaddr) public returns (bool) {
```

Recommendation

Change function visibility as external.

I-04 | Code layout Conventions

Informational : Code Conventions

File Location : contracts/IAnySwapV5ERC20.sol:4, contracts/AnyswapV5ERC20.sol:14,203

Description

In the solidity document(https://docs.soliditylang.org/en/v0.8.17/style-guide.html), there are the following conventions for code layout:
Layout contract elements in the following order: 1. Pragma statements, 2. Import statements, 3. Interfaces, 4. Libraries, 5. Contracts.
Inside each contract, library or interface, use the following order: 1. Type declarations, 2. State variables, 3. Events, 4. Modifiers, 5. Functions.
Functions should be grouped according to their visibility and ordered: 1. constructor, 2. receive function (if exists), 3. fallback function (if exists), 4. external, 5. public, 6. internal, 7. private.

Recommendation

Recommended to follow code layout conventions.

I-05 | Interface Defined But Not Inherited

Informational : Code Conventions

File Location : contracts/IAnySwapV5ERC20.sol:4, contracts/AnyswapV5ERC20.sol:203

Description

Interface IAnySwapV5ERC20 is defined but not inherited. Meanwhile, contract AnyswapV5ERC20 inherits interface IAnyswapV3ERC20and used as IAnySwapV5ERC20 in withdrawWrap.
This inconsistency may cause incorrect function signature when calling contract using interface.

contracts/IAnySwapV5ERC20.sol

```
4   interface IAnySwapV5ERC20 {
5       function transfer(address to, uint256 value) external returns (bool);
6
7       function balanceOf(address account) external view returns (uint256);
8
9       event Transfer(address indexed from, address indexed to, uint256 value);
10  }
```

contracts/AnyswapV5ERC20.sol

```
203  contract AnyswapV5ERC20 is IAnyswapV3ERC20 {
```

contracts/withdrawWrap.sol

```
70  function withdrawWrapper(
71      bytes memory _requestId,
72      address[] calldata _to,
73      uint256[] calldata _amounts,
74      uint256 _totalAmount
75  ) external {
76      require(authorizes[msg.sender], "Invalid Authorizer");
77      require(_to.length == _amounts.length, "Invalid Parameter");
78
79      uint256 balance = IAnySwapV5ERC20(AnyswapV5ERC20Address).balanceOf(
80          address(this)
81      );
82      ......
83  }
```

Recommendation

It is recommended to correctly inherit interface to keep consistency between interface and implementation.

I-06 | Prefer uint256 For Variables

Informational : Gas Optimization

File Location : contracts/AnyswapV5ERC20.sol:207

Description

It is recommended to replace integer types that are not 32 bytes in size and cannot be combined with other storage with uint256 to avoid the gas overhead caused by filling 32 bytes in operation.

```
207  uint8 public immutable override decimals;
```

Recommendation

Change variable decimals to uint256.

I-07 | Variables Should Be Constants

Informational : Gas Optimization

File Location : contracts/AnyswapV5ERC20.sol:232

Description

There are unchanging state variables delay can be declared as constant to save gas.

```
232  uint256 public delay = 2 * 24 * 3600;
```

Recommendation

Change variables delay to constant.

I-08 | No Check of Address Params with Zero Address

Informational : Code Conventions

File Location : contracts/withdrawWrap.sol:59, 111,
contracts/AnyswapV5ERC20.sol:275,285,290,307,393,582

Description

The input parameter of the address type in the function should check against the zero address.

contracts/withdrawWrap.sol

```
59  constructor(
60      address _AnyswapV5ERC20Address,
61      uint256 _threshold,
62      feeSystem memory _fee
63   ) {
64      ......
65  }
```

contracts/AnyswapV5ERC20.sol

```
1   function initVault(address _vault) external onlyVault {
2       ......
3   }
4
5   function setMinter(address _auth) external onlyVault {
6       ......
7   }
8
9   function setVault(address _vault) external onlyVault {
10      ......
11  }
12
13  function revokeMinter(address _auth) external onlyVault {
14      ......
15  }
16
17  constructor(
18      string memory _name,
19      string memory _symbol,
20      uint8 _decimals,
21      address _underlying,
22      address _vault
23  ) {
24      ......
25  }
26
27  function approve(address spender, uint256 value)
28      external
29      override
30      returns (bool)
31  {
32      ......
33  }
```

Recommendation

It is recommended to perform zero address verification on the input parameters of the address type.

4. Disclaimer

No description, statement, recommendation or conclusion in this report shall be construed as endorsement, affirmation or confirmation of the project. The security assessment is limited to the scope of work as stipulated in the Statement of Work.

This report is prepared in response to source code, and based on the attacks and vulnerabilities in the source code that already existed or occurred before the date of this report, excluding any new attacks or vulnerabilities that exist or occur after the date of this report. The security assessment are solely based on the documents and materials provided by the customer, and the customer represents and warrants documents and materials are true, accurate and complete.

CONSULTANT DOES NOT MAKE AND HEREBY DISCLAIMS ANY REPRESENTATIONS OR WARRANTIES OF ANY KIND, WHETHER EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, REGARDING THE SERVICES, DELIVERABLES, OR ANY OTHER MATTER PERTAINING TO THIS REPORT.

CONSULTANT SHALL NOT BE RESPONSIBLE FOR AND HEREBY DISCLAIMS MERCHANTABILITY, FITNESS FOR PURPOSE, TITLE, NON-INFRINGEMENT OR NON-APPROPRIATION OF INTELLECTUAL PROPERTY RIGHTS OF A THIRD PARTY, SATISFACTORY QUALITY, ACCURACY, QUALITY, COMPLETENESS, TIMELINESS, RESPONSIVENESS, OR PRODUCTIVITY OF THE SERVICES OR DELIVERABLES.

CONSULTANT EXCLUDES ANY WARRANTY THAT THE SERVICES AND DELIVERABLES WILL BE UNINTERRUPTED, ERROR FREE, FREE OF SECURITY DEFECTS OR HARMFUL COMPONENTS, REVEAL ALL SECURITY VULNERABILITIES, OR THAT ANY DATA WILL NOT BE LOST OR CORRUPTED.

CONSULTANT SHALL NOT BE RESPONSIBLE FOR (A) ANY REPRESENTATIONS MADE BY ANY PERSON REGARDING THE SUFFICIENCY OR SUITABILITY OF SERVICES AND DELIVERABLES IN ANY ACTUAL APPLICATION, OR (B) WHETHER ANY SUCH USE WOULD VIOLATE OR INFRINGE THE APPLICABLE LAWS, OR (C) REVIEWING THE CUSTOMER MATERIALS FOR ACCURACY.

5. Appendix

5.1 Visibility

Contract	FuncName	Visibility	Mutability	Modifiers
AnyswapV5ERC20	owner	public	N	
AnyswapV5ERC20	mpc	public	N	
AnyswapV5ERC20	setVaultOnly	external	Y	onlyVault
AnyswapV5ERC20	initVault	external	Y	onlyVault
AnyswapV5ERC20	setMinter	external	Y	onlyVault
AnyswapV5ERC20	setVault	external	Y	onlyVault
AnyswapV5ERC20	applyVault	external	Y	onlyVault
AnyswapV5ERC20	applyMinter	external	Y	onlyVault
AnyswapV5ERC20	revokeMinter	external	Y	onlyVault
AnyswapV5ERC20	getAllMinters	external	N	
AnyswapV5ERC20	changeVault	external	Y	onlyVault
AnyswapV5ERC20	changeMPCOwner	public	Y	onlyVault
AnyswapV5ERC20	mint	external	Y	
AnyswapV5ERC20	burn	external	Y	onlyAuth
AnyswapV5ERC20	Swapin	public	Y	onlyAuth
AnyswapV5ERC20	Swapout	public	Y	

Contract	FuncName	Visibility	Mutability	Modifiers
AnyswapV5ERC20	_CTOR_	public	Y	
AnyswapV5ERC20	totalSupply	external	N	
AnyswapV5ERC20	depositWithPermit	external	Y	
AnyswapV5ERC20	depositWithTransferPermit	external	Y	
AnyswapV5ERC20	deposit	external	Y	
AnyswapV5ERC20	deposit(uint256 amount)	external	Y	
AnyswapV5ERC20	deposit(uint256 amount, address to)	external	Y	
AnyswapV5ERC20	depositVault	external	Y	onlyVault
AnyswapV5ERC20	_deposit	internal	Y	
AnyswapV5ERC20	withdraw	external	Y	
AnyswapV5ERC20	withdraw(uint256 amount)	external	Y	
AnyswapV5ERC20	withdraw(uint256 amount, address to)	external	Y	
AnyswapV5ERC20	withdrawVault	external	Y	onlyVault
AnyswapV5ERC20	_withdraw	internal	Y	
AnyswapV5ERC20	_mint	internal	Y	
AnyswapV5ERC20	_burn	internal	Y	

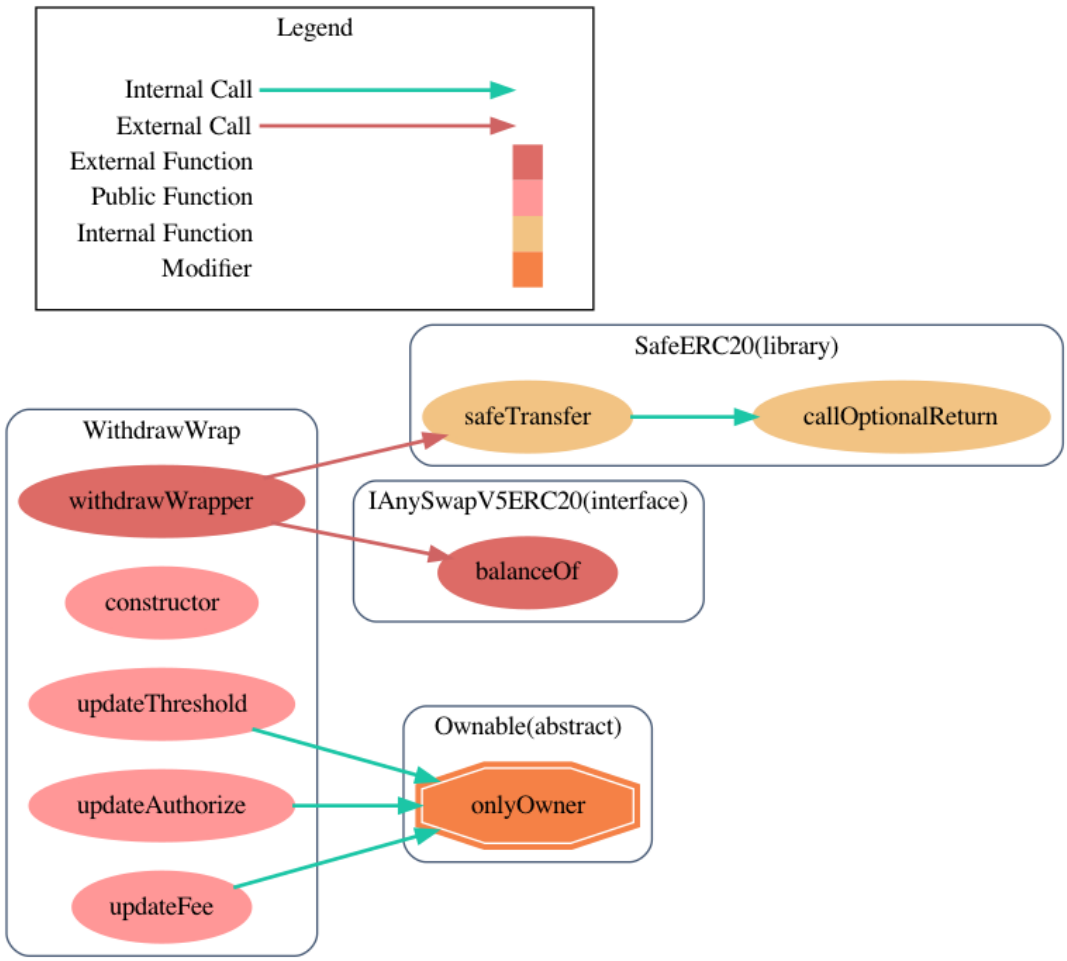
Contract	FuncName	Visibility	Mutability	Modifiers
AnyswapV5ERC20	approve	external	Y	
AnyswapV5ERC20	approveAndCall	external	Y	
AnyswapV5ERC20	permit	external	Y	
AnyswapV5ERC20	transferWithPermit	external	Y	
AnyswapV5ERC20	verifyEIP712	internal	N	
AnyswapV5ERC20	verifyPersonalSign	internal	N	
AnyswapV5ERC20	prefixed	internal	N	
AnyswapV5ERC20	transfer	external	Y	
AnyswapV5ERC20	transferFrom	external	Y	
AnyswapV5ERC20	transferAndCall	external	Y	
WithdrawWrap	_CTOR_	public	Y	
WithdrawWrap	withdrawWrapper	external	N	
WithdrawWrap	updateThreshold	public	Y	onlyOwner
WithdrawWrap	updateAuthorize	public	Y	onlyOwner
WithdrawWrap	updateFee	public	Y	onlyOwner

5. Appendix

5.2 Call Graph

contracts/AnyswapV5ERC20.sol

contracts/withdrawWrap.sol



5. Appendix

5.3 Inheritance Graph

contracts/AnyswapV5ERC20.sol

contracts/withdrawWrap.sol