Smart Contract Security Audit Report

[2021]

Table Of Contents

1 Executive Summary

On 2021.04.19, the SlowMist security team received the SOLV team's security audit application for SOLV 2.0, 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

Solv is a DeFi protocol purposed to create an on-chain market for an emerging category of crypto assets – Financial NFTs. Solv designed a novel and powerful non-fungible token standard called versatile NFT (vNFT for short) to standardize and simplify the expression and programming of advanced financial instruments. Solv is also developing a decentralized platform to facilitate the transactions of such assets. Solv has deployed the Investment Certificate as the first Financial NFTs to reveal the potential and flexibility of Financial NFTs.

This audit report covers Solv 2.0, which is a new version formed by code refactoring on the basis of Solv 1.0 and the addition of a series of contracts such as Solv IC Market. All the code has been completely re-audited, so this report does not include the audit information of Solv 1.0.

audit version code：

https://github.com/solv-finance-dev/solv-ictoken/tree/f49719d67ab48a32a3767c7dec1ff16a1296d282

https://github.com/solv-finance-dev/solver/tree/18d6887d1cd7351ed09a76d3de6f13fc12f1795d

https://github.com/solv-finance-dev/solv-token/tree/f7e7c3bbc0c4c6dea76499dce53297d0dfd1b24b

https://github.com/solv-finance-dev/solv-vnft-core/tree/d2ab317f09f84a6c44e868d09630d14fb93b5c1d

https://github.com/solv-finance-dev/solv-icmarket/tree/9956907a9ece5c4da21bc0713c42fa676c8766ac

fixed version code:

https://github.com/solv-finance/solv-v2-voucher/tree/8821fcc8bd47d1bae6fb0071ebc36312d39ea041

https://github.com/solv-finance/solv-v2-helper/tree/a6568b839e963a265bd80fc21439a52e5035b5dc

https://github.com/solv-finance/solv-v2-market/tree/f54e1934a2b4e25ee8ba1be5b3c08a1ccdb4050f

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Recommendations for coding specification enhancements	Others	Suggestion	Confirmed
N2	Redundant code	Others	Suggestion	Fixed
N3	Excessive authority issue	Authority Control Vulnerability	Low	Confirmed

4 Code Overview

4.1 Contracts Description

The main network address of the contract is as follows:

smart contract	BSC mainnet address
solver-ProxyAdmin	0xc605c3d26e665dB4114ab8D4CBB1255eaDb4db79

smart contract	BSC mainnet address
solver-Impl	0x949ab02cA3297D5BCf3C19B1e08BbDfe9E9E767C
solver-Proxy	0x7922dba230FdEd829c913e07588326E6f7cE7f1A
voucher-ProxyAdmin	0x6be9B5DCb835478E68a6C06a66eeA1c0c16E74Bb
voucher-avIDIA VestingPool-Impl	0x256F2d67e52fE834726D2DDCD8413654F5Eb8b53
voucher-avIDIA VestingPool-Proxy	0x67D48Ce0E776147B0d996e1FaCC0FbAA91b1CBC4
voucher-avIDIA-Proxy	0x0c491ac26d2cDDa63667DF65b43b967B9293161c
voucher market-ProxyAdmin	0xfdcdE28359Db316957534e825327d99D9f4a5d17
voucher market-Impl	0xba85C3AF2DEa9a5Fb1541AC68B92711E19764537
voucher market-Proxy	0x758ae0c48ff013D18b3D63Ca79408e70A977d496

smart contract	ethereum mainnet address
solver-ProxyAdmin	0xc605c3d26e665dB4114ab8D4CBB1255eaDb4db79
solver-Impl	0x949ab02cA3297D5BCf3C19B1e08BbDfe9E9E767C
solver-Proxy	0x7922dba230FdEd829c913e07588326E6f7cE7f1A
SOLV erc20 token	0x256F2d67e52fE834726D2DDCD8413654F5Eb8b53
voucher-ProxyAdmin	0x039Bb4b13F252597a69fA2e6ad19034E3CCbbF1C
voucher-icSOLVVestingPool-Impl	0xEF675b68DC40825B3E0c67a981f03e22F54dCE9d
voucher-icSOLVVestingPool-Proxy	0x7D0C93DcAD6f6B38C81431d7262CF0E48770B81a
voucher-icSOLV-Impl	0xe973CdBe071D30977D42Aefc6720A5D65E406fB5
voucher-icSOLV-Proxy	0xfdcdE28359Db316957534e825327d99D9f4a5d17

smart contract	ethereum mainnet address
voucher-icPRQVestingPool-Impl	0xfB923B66B1c152e253dd5b078c723888C6342DC2
voucher-icPRQVestingPool-Proxy	0x3A076c72219eCaC060A09Ce1006f4194007a7C0f
voucher-icPRQ-Impl	0xd8e8ad3ee2B5B78a994e472DDdEdDEaCE0a66079
voucher-icPRQ-Proxy	0x705d6b0c912a33A8643b5F7E42B5c7CA386A3b32
voucher-icDODOVestingPool-Impl	0xfB923B66B1c152e253dd5b078c723888C6342DC2
voucher-icDODOVestingPool-Proxy	0x4A6c7ba0Cda9Bdc51Bd088Be20664926225BeEbF
voucher-icDODO-Impl	0x3ebceb9989630b3f4E4Eb0C7263A07E8c515bE23
voucher-icDODO-Proxy	0xaDB9AB302ECd551264a718d43aE6B3C255a8afa5
voucher-icSFIVestingPool-Impl	0xf8F24538c4DF984b1dFf64267fa9299601135F2e
voucher-icSFIVestingPool-Proxy	0x87A6F77eEC56f37c56a625Ab690294274a6116b6
voucher-icSFI-Impl	0xb4CF4Bc604740D6bD946B3E8BF89f01399296ec2
voucher-icSFI-Proxy	0x884F3b6d68DcC64bad97c84b5e32b80cE0EBBe7d
voucher-icAuctionVestingPool-Impl	0xf8F24538c4DF984b1dFf64267fa9299601135F2e
voucher-icAuctionVestingPool-Proxy	0xf30e8dF9C5EC87edfCfC9E15A349dE8061Dfe8dB
voucher-icAuction-Impl	0x7B4b2e785d434bf2C013F1DF2641a25A2466a0Aa
voucher-icAuction-Proxy	0xE995Aa82Ed3E0a3E86097D3d2914aA6AA20777Ab
voucher-avPROM VestingPool-Impl	0xBF3d2a3a073923306f946DE360298d7872F920b7
voucher-avPROM VestingPool-Proxy	0x84A4734da8Ea31e81cbe52eB547A3097f641a6d3
voucher-avPROM-Impl	0x7Eb516621DEf07981b98Fa779D53B86d8bDB89F3
voucher-avPROM-Proxy	0xb62d5ceb3668Fd7919e3ad860779B6c4AefFbE70

smart contract	ethereum mainnet address
voucher market-ProxyAdmin	0xd4A4025Ac285796A237aF820452F569cf9fC92a1
voucher market-Impl	0x04a2C0258c9C13E4B3a56aFFE6804Cc6C5c1587A
voucher market-Proxy	0xD91A208995bfBde9D133c39417FBD352e595650b

4.2 Visibility Description

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

Solver			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can Modify State	initializer
isSolver	External	-	-
_setTransferGuuardianPause	Public	Can Modify State	onlyAdmin
_setDepositGuuardianPause	Public	Can Modify State	onlyAdmin
_setWithdrawGuuardianPause	Public	Can Modify State	onlyAdmin
_setConvertUnsafeTransferContracts	Public	Can Modify State	onlyAdmin
_setRejectUnsafeTransferContracts	Public	Can Modify State	onlyAdmin
depositAllowed	External	Can Modify State	-
depositVerify	External	Can Modify State	-
withdrawAllowed	External	Can Modify State	-
withdrawVerify	External	Can Modify State	-
transferFromAllowed	External	Can Modify State	-

Solver			
transferFromVerify	External	Can Modify State	-
mergeAllowed	External	Can Modify State	-
mergeVerify	External	Can Modify State	-
splitAllowed	External	Can Modify State	-
splitVerify	External	Can Modify State	-
publishFixedPriceAllowed	External	Can Modify State	-
publishDecliningPriceAllowed	External	Can Modify State	-
publishVerify	External	Can Modify State	-
buyAllowed	External	Can Modify State	-
buyVerify	External	Can Modify State	-
removeAllow	External	Can Modify State	-
needConvertUnsafeTransfer	Public	-	-
needRejectUnsafeTransfer	Public	-	-
_setPendingAdmin	Public	Can Modify State	-
_acceptAdmin	Public	Can Modify State	-

VNFTCore			
Function Name	Visibility	Mutability	Modifiers
_initialize	Internal	Can Modify State	-
_setContractURI	Internal	Can Modify State	-

VNFTCore			
contractURI	Public	-	-
_safeTransferUnitsFrom	Internal	Can Modify State	-
_transferUnitsFrom	Internal	Can Modify State	-
_merge	Internal	Can Modify State	-
_splitUnits	Internal	Can Modify State	-
approve	Public	Can Modify State	-
_mintUnits	Internal	Can Modify State	-
_exists	Internal	-	-
_burn	Internal	Can Modify State	-
_burnUnits	Internal	Can Modify State	-
_approveUnits	Internal	Can Modify State	-
allowance	Public	-	-
unitsInToken	Public	-	-
balanceOfSlot	Public	-	-
tokenOfSlotByIndex	Public	-	-
slotOf	Public	-	-
isValid	Public	-	-
_checkOnVNFTReceived	Internal	Can Modify State	-

SOLV

SOLV			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	ERC20

VestingPool			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can Modify State	-
isVestingPool	External	-	-
_setManager	Public	Can Modify State	onlyAdmin
mint	External	Payable	onlyManager
_mint	Internal	Can Modify State	-
claim	External	Can Modify State	onlyManager
claimableAmount	Public	-	-
_claim	Internal	Can Modify State	-
transferVesting	Public	Can Modify State	onlyManager
splitVesting	Public	Can Modify State	onlyManager
mergeVesting	Public	Can Modify State	onlyManager
units2amount	Public	-	-
amount2units	Public	-	-
totalAmount	Public	-	-
getVestingSnapshot	Public	-	-

VestingPool			
underlying	Public	-	-
_setPendingAdmin	Public	Can Modify State	-
_acceptAdmin	Public	Can Modify State	-
_add	Internal	-	-
_sub	Internal	-	-

ICToken			
Function Name	Visibility	Mutability	Modifiers
initialize	External	Can Modify State	-
owner	External	-	-
setContractURI	External	Can Modify State	onlyAdmin
setBaseURI	External	Can Modify State	onlyAdmin
mint	External	Payable	-
_mint	Internal	Can Modify State	nonReentrant
claim	External	Can Modify State	-
claimAll	External	Can Modify State	-
claimableAmount	Public	-	-
_claim	Internal	Can Modify State	nonReentrant
split	External	Can Modify State	-
_splitUnits	Internal	Can Modify State	-

ICToken			
merge	External	Can Modify State	-
_merge	Internal	Can Modify State	-
transferFrom	Public	Can Modify State	-
transferFrom	Public	Can Modify State	-
safeTransferFrom	External	Can Modify State	-
safeTransferFrom	External	Can Modify State	-
_transferUnitsFrom	Internal	Can Modify State	-
generateTokenId	Internal	Can Modify State	-
getSlot	Internal	-	-
getSnapshot	Public	-	-
getUnderlyingAmount	External	-	-
getUnits	External	-	-
underlying	External	-	-
totalUnderlyingAmount	External	-	-
_setSolver	Public	Can Modify State	onlyAdmin
_setVestingPool	Public	Can Modify State	onlyAdmin
_setPendingAdmin	External	Can Modify State	-
_acceptAdmin	External	Can Modify State	-
_sub	Internal	-	-

PriceManager			
Function Name	Visibility	Mutability	Modifiers
price	Internal	-	-
setFixedPrice	Internal	Can Modify State	-
setDecliningPrice	Internal	Can Modify State	-

SolvICMarket			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
initialize	Public	Can Modify State	-
publishFixedPrice	External	Can Modify State	-
publishDecliningPrice	External	Can Modify State	-
_publish	Internal	Can Modify State	-
buyByAmount	External	Payable	-
buyByUnits	External	Payable	-
_buy	Internal	Can Modify State	-
remove	Public	Can Modify State	-
_getFee	Internal	-	-
getPrice	Public	-	-
totalSalesOfICToken	Public	-	-
saleIdOfICTokenByIndex	Public	-	-

SolvICMarket			
_gernateNextSaleId	Internal	Can Modify State	-
_gernateNextTradeId	Internal	Can Modify State	-
_addMarket	Public	Can Modify State	onlyAdmin
_removeMarket	Public	Can Modify State	onlyAdmin
_setCurrency	Public	Can Modify State	onlyAdmin
_withdrawFee	Public	Can Modify State	onlyAdmin
_addAllowAddress	External	Can Modify State	onlyAdmin
_removeAllowAddress	External	Can Modify State	onlyAdmin
_setSolver	Public	Can Modify State	onlyAdmin
_setPendingAdmin	Public	Can Modify State	-
_acceptAdmin	Public	Can Modify State	-

SolvICMarketV2			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
initialize	Public	Can Modify State	-
publishFixedPrice	External	Can Modify State	-
publishDecliningPrice	External	Can Modify State	-
_publish	Internal	Can Modify State	-
buyByAmount	External	Payable	-

SolvICMarketV2			
buyByUnits	External	Payable	-
_buy	Internal	Can Modify State	-
remove	Public	Can Modify State	-
_getFee	Internal	-	-
getPrice	Public	-	-
totalSalesOfICToken	Public	-	-
saleIdOfICTokenByIndex	Public	-	-
_gernateNextSaleId	Internal	Can Modify State	-
_gernateNextTradeId	Internal	Can Modify State	-
_addMarket	Public	Can Modify State	onlyAdmin
_removeMarket	Public	Can Modify State	onlyAdmin
_setCurrency	Public	Can Modify State	onlyAdmin
_withdrawFee	Public	Can Modify State	onlyAdmin
_addAllowAddress	External	Can Modify State	onlyAdmin
_removeAllowAddress	External	Can Modify State	onlyAdmin
_setSolver	Public	Can Modify State	onlyAdmin
_setPendingAdmin	Public	Can Modify State	-
_acceptAdmin	Public	Can Modify State	-

4.3 Vulnerability Summary

[N1] [Suggestion] Recommendations for coding specification enhancements

Category: Others

Content

The following functions of the SolvicMarket contract are allowed to be called from outside, but are named in the same style as functions called from inside, starting with an underscore.

```
_addMarket , _removeMarket , _setCurrency , _withdrawFee , _addAllowAddress ,
_removeAllowAddress , _setSolver , _setPendingAdmin , _acceptAdmin
```

The same issues:

The following functions of the VestingPool contract：

```
_setSolver , _setVestingPool , _setPendingAdmin , _acceptAdmin
```

The following functions of the ICToken contract:

```
_setManager , _setPendingAdmin , _acceptAdmin
```

Solution

It is recommended that externally available functions should not be named with an underscore to avoid confusion in subsequent code iterations.

Status

Confirmed; After communication and feedback, the issue will not be fixed.

[N2] [Suggestion] Redundant code

Category: Others

Content

The function uses payable, but the code doesn't use the value of msg.value, so the payable is redundant code.

- solv-ictoken/contracts/ICToken.sol

```
 function mint(
        uint64 term_, /*seconds*/
        uint256 amount_,
        uint64[] calldata maturities_,
        uint32[] calldata percentages_,
        string memory originalInvestor_
    ) external payable virtual override returns (uint256, uint256) {
        (uint256 slot, uint256 tokenId) =
        _mint(msg.sender, term_, amount_, maturities_, percentages_,
originalInvestor_);
        return (slot, tokenId);
    }
```

- solv-ictoken/contracts/ICToken.sol

```
function _mint(
        address minter_,
        uint64 term_,
        uint256 amount_,
        uint64[] memory maturities_, /*seconds*/
        uint32[] memory percentages_,
        string memory originalInvestor_
    ) internal virtual nonReentrant returns (uint256, uint256) {
        MintLocalVar memory vars;
        vars.tokenId = generateTokenId();
        vars.claimType = maturities_.length > 1 ? 2 : term_ == 0 ? 1 : 0;

        uint256 err =
        solver.depositAllowed(
            address(this),
            minter_,
            term_,
            amount_,
            maturities_
        );
        require(err == 0, "solver not allowed");
        /*
        if (vestingPool.underlying() == EthAddressLib.ethAddress()) {
            address(vestingPool).call{value : amount_}
(abi.encodeWithSignature("mint(uint8,address,uint256,uint64,uint256,uint64[],uint32[]
,string) payable returns (uint256)",
```

```
            vars.claimType,
            minter_,
            vars.tokenId,
            term_,
            amount_,
            maturities_,
            percentages_,
            originalInvestor_));
    } else {
        vars.mintUnits =
        vestingPool.mint(
            vars.claimType,
            minter_,
            vars.tokenId,
            term_,
            amount_,
            maturities_,
            percentages_,
            originalInvestor_
        );
    }
    */
    vars.mintUnits =
        vestingPool.mint(
            vars.claimType,
            minter_,
            vars.tokenId,
            term_,
            amount_,
            maturities_,
            percentages_,
            originalInvestor_
        );


    vars.slot = getSlot(vars.claimType, maturities_, percentages_, term_);

    VNFTCore._mintUnits(minter_, vars.tokenId, vars.slot, vars.mintUnits);

    solver.depositVerify(
        address(this),
        minter_,
        amount_,
        vars.tokenId,
        term_,
```

```
            maturities_
        );

        return (vars.slot, vars.tokenId);
    }
```

Solution

It is recommended that remove the payable redundant code.

Status

Fixed; The issue has been fixed in the fixed version code.

[N3] [Low] Excessive authority issue

Category: Authority Control Vulnerability

Content

The contract structure adopts an upgradable model (Proxy-Impl). Currently, the owner of the solver-ProxyAdmin and voucher market-ProxyAdmin contracts is 0x21bc9179d5c529b52e3ee8f6ecf0e63fa231d16c (EOA address). The owner can upgrade the contract. If the upgraded new contract does not pass a security audit, it may have security risks. Therefore, the owner role of the ProxyAdmin contract has an issue of excessive authority.

SolvICMarket, ICToken's Admin can configure contract parameters and modify the external contract address called in the project. The current Admin is 0x21bc9179d5c529b52e3ee8f6ecf0e63fa231d16c (EOA address). The new external contract address may have security risks. Therefore, ICToken's Admin role has the issue of excessive authority.

The manager of SolvICMarket can configure contract parameters. Modifying and configuring contract parameters will affect the operation of the project. Therefore, the manager role of SolvICMarket has the issue of excessive authority.

Solution

It is recommended to transfer the authority of the Owner and Admin roles to the governance contract or timelock contract.

Status

Confirmed; After communication and feedback with the project party, in the initial stage of the project, in order to ensure the reliable and stable operation of the project, the project party needs certain management authority, and after the subsequent stable operation of the project, it will consider governance or timelock.

5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0x002104260002	SlowMist Security Team	2021.04.19 - 2021.04.26	Low Risk

Summary conclusion: The SlowMist security team uses a manual and SlowMist team's analysis tool to audit the project, during the audit work we found 1 low risk, 2 suggestion vulnerabilities. And 1 low risk, 1 suggestion vulnerabilities were confirmed; All other findings were fixed. the excessive authority issue has not been 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.

Official Website

www.slowmist.com

E-mail

team@slowmist.com

Twitter

@SlowMist_Team

Github

https://github.com/slowmist