FusionTech

SolidX

Security Assessment

www.solidx.tech

TABLE OF CONTENTS

Summary

Overview

Project Summary

Audit Summary

Vulnerability Summary

Audit Scope

Findings

01 : Centralization Risk in _hasBeenLiqAdded() Function

02 : Initial Token Distribution

03 : PotentialSandwich Attacks

04 : Redundant Code

05 : Typos In The Contract

06 : Function and Variable Naming Doesn’t Match the Operating Environment

07 : Potential ResourceExhaustion

08 : FullInlinerNonExpressionSplitArgumentEvaluationOrder

09 : StorageWriteRemovalBeforeConditionalTermination

10 : DelegateCallReturnValue

Appendix

Disclaimer

About

Summary

This report has been prepared for to discover issues and vulnerabilities in the source code of the project as well as any contract dependencies that were not part of an officially recognized library.A comprehensive examination has been performed, utilizing Static Analysisand Manual Reviewtechniques. The auditingprocess pays specialattention to the following considerations:

Testing the smart contracts against both commonand uncommon attackvectors. Assessing the codebase to ensure compliancewith current best practices and industry standards. Ensuring contract logic meets the specifications and intentions of the client. Cross referencing contract structure and implementation againstsimilar smart contracts produced by industryleaders. Thorough line-by-line manual review of the entirecodebase by industryexperts.

The security assessment resulted in findings that ranged from critical to informational. We recommendaddressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could betterserve the projectfrom the securityperspective:

Enhance general coding practices for better structures of source codes; Add enough unit tests to cover the possible use cases; Provide more comments per each function for readability, especiallycontracts that are verified in public; Provide more transparency on privileged activities once the protocolis live.

Project Summary

Project Name	SolidX - (https://solidx.tech/)
Platform	ETHEREUM
Language	Solidity
Code Base	https://etherscan.io/address/0x072382557067b36966dab6f5fb90be32c2da07eb
Commit	47674cccf80b356c2b0bf25b17f545ddf30e876fe6efb7399863f9aff3213339

Audit Summary

Delivery Date	October 19, 2023
Audit Methodology	Static Analysis, Manual Review
Key Components	SolidX

Vulnerability Summary

Vulnerability Summary	Total	ⓘ Pending	ⓘ Declined	ⓘ Acknowledged	ⓘ Partially Resolved	ⓘ Resolved
Critical	0	0	0	0	0	0
Major	0	0	0	0	0	0
Medium	0	0	0	0	0	0
Minor	0	0	0	0	0	0
Informational	0	0	0	0	0	0
Discussion	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
CKP	contract.sol	f79198f1e334d2889b0de0d9507c2bf3e16e6299f37d30102d9496b69c383809

Overview

External Dependencies

The contract serves as the underlying entity to interact with third-party protocols (token- wrapping). The scope of the audit treats third-party entities as blackboxes and assumes their functional correctness. However, in the real world, third parties can be compromised and this may lead to lost or stolen assets.

Privileged Functions

The contract contains the following privileged functions that are restricted by role with the modifier. Since the contract is the owner cannot modify the contract configurations and address attributes.

Overview

External Dependencies

The contract serves as the underlying entity to interact with third-party protocols (token- wapping). The scope of the audit treats third-party entities as blackboxes and assumestheir functional correctness. However, in the real world, third parties can be compromised and this may lead to lost or stolenassets.

Privileged Functions

The contract contains the following privileged functions are restricted to gain access by the modifier/_owner. They are used to modify the contract configurations and address attributes. We grouped these functions below.

01 | Centralization Risk in Function

Description

The addLiquidity()_hasLiqBeenAdded() function calls the UniswapV2Router. addLiquidityETH function with the to() address specified as owner() for acquiring the generated LP tokens from the corresponding pool. As a result, over time the _owner address will accumulate a significant portionof LP tokens. If _owner the is an EOA (Externally Owned Account),mishandling of its private key can have devastating consequences to the projectas a whole.

Recommendation

We advise to() address of the UniswapV2Router. addLiquidityETH() function call to be replaced by the contract() itself, i.e. address(this) , and to restrict the management of the LP tokens within the scope of the contract's business logic. This will also protect the LP tokens from being stolen if the _owner() account is compromised. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or via smart-contract based accounts with enhanced security practices, f.e.Multisignature wallets().

02 | Initial Token Distribution

Category	Severity	Location	Status
Logical Issue	Minor	projects/contract.sol (98a012): 497	Acknowledged

Description

All of the tokens are sent to the contract deployer when deploying the contract. This could be a centralization risk as the deployer can distribute those tokens without obtaining the consensus of the community.

Recommendation

We recommend the team to be transparent regarding the initial token distribution process.

```
}

function changeLimits(uint256 _buy, uint256 _trans, uint256 _wallet) external onlyOwner {
        uint256 newTx = (totalSupply() * _buy) / 100;
        uint256 newTransfer = (totalSupply() * _trans) / 100;
        uint256 newWallet = (totalSupply() * _wallet) / 100;
        _maxTxAmountPercent = _buy;
        _maxTransferPercent = _trans;
        _maxWalletPercent = _wallet;
        uint256 limit = totalSupply().mul(5).div(1000);
         require(newTx >= limit && newTransfer >= limit && newWallet >= limit, "Max TXs and Max
Wallet cannot be less than .5%");

}
function changeTaxReceiverAddresses(address _liquidity_receiver, address _marketing_receiver,
address _development_receiver) external onlyOwner {
        liquidity_receiver = _liquidity_receiver;
        marketing_receiver = _marketing_receiver;
        development_receiver = _development_receiver;
    }
```

03 | Potential Sandwich Attacks

Description

A sandwich attack might happen when an attacker observes a transaction swapping tokens or adding liquidity without setting restrictions on slippage or minimum output amount. The attacker can manipulate the exchange rate by frontrunning (beforethe transaction being attacked) a transaction to purchase one of the assets and make profitsby backrunning (afterthe transaction beingattacked) a transaction to sell the asset.

The following functionsare called withoutsetting restrictions on slippage or minimum outputamount, so transactions triggering these functions are vulnerable to sandwich attacks, especially when the input amount is large:

Recommendation

We recommend setting reasonable minimum output amounts, instead of 0, based on token prices when calling the fore mentioned functions.

```
}

interface IFactory{
        function createPair(address tokenA, address tokenB) external returns (address pair);
        function getPair(address tokenA, address tokenB) external view returns (address pair);
}

interface IRouter {
    function factory() external pure returns (address);
    function WETH() external pure returns (address);
    function addLiquidityETH(
        address token,
        uint amountTokenDesired,
        uint amountTokenMin,
        uint amountETHMin,
        address to,
        uint deadline
    ) external payable returns (uint amountToken, uint amountETH, uint liquidity);

}
```

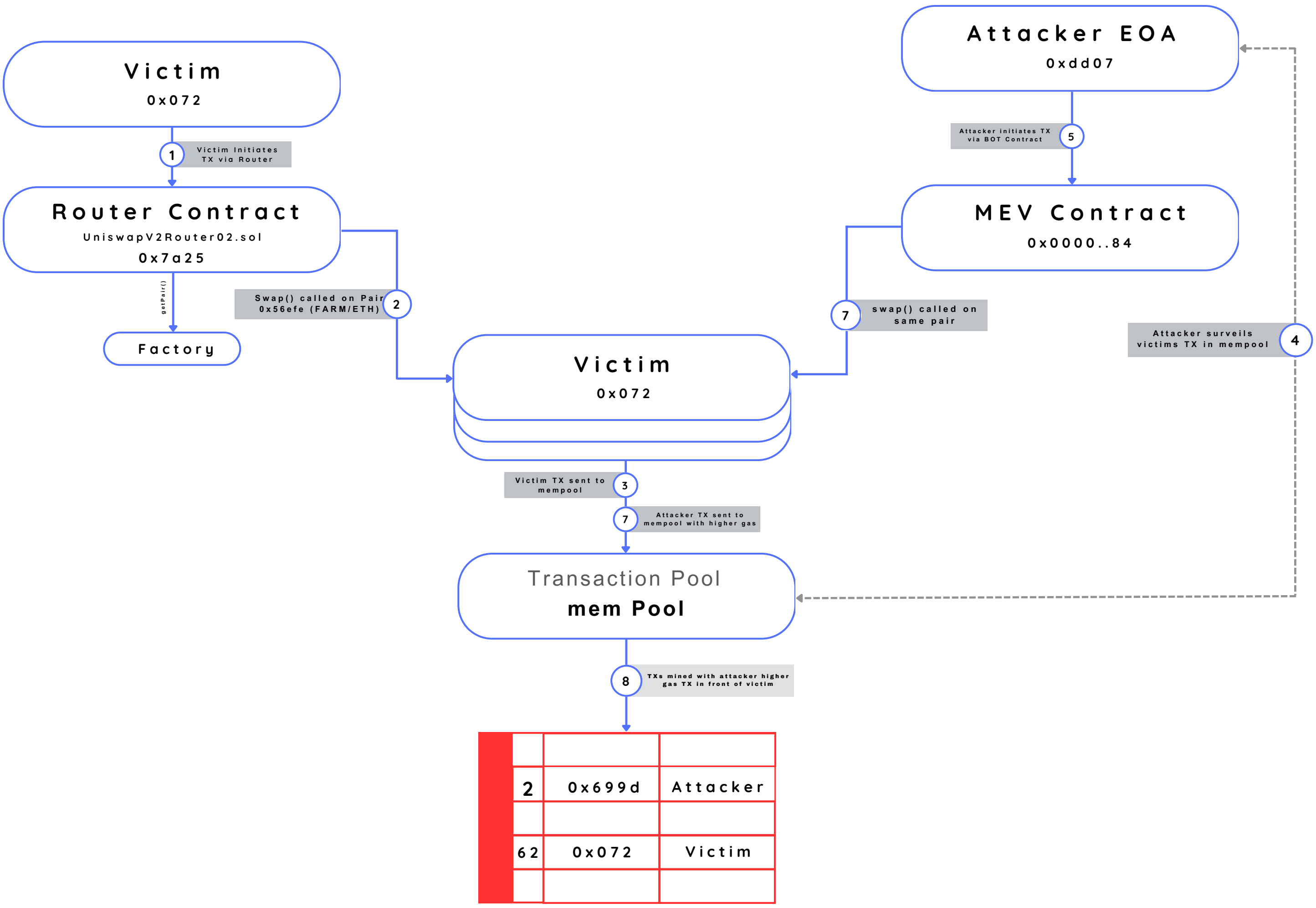
04 | Redundant Code

Category	Severity	Location	Status
Logical Issue	• Informational	projects/contract.sol (98a012): 862	ⓘ Acknowledged

Description

The condition! _isExcluded[sender] & !_isExcluded[recipient]
can be included in else .

```
...else if (!_isExluded[sender] && !_isExcluded[recipent]) {
    _transferStandard(sender, recipient, amount);
} ...
```



05 | Typos In The Contract

Category	Severity	Location	Status
Coding Style	Informational	projects/contract.sol (98a012): 470, 670	Acknowledged

Description

There are several typos in the code and comments.

1. In the following code snippet, tokensIntoLiquidity() should be tokensIntoLiquidity()

```
event SwapAndLiquify(
        uint256 tokensSwapped,
        uint256 ethReceived,
        uint256 tokens IntoLiqudity
  );
```

2. recieve() should be recieve() _swapping() should be _swapping() in the line of comment //to _recieve ETH from UniswapV2Router when swaping() .

06 | Function and Variable Naming

Description

_maxTransferAmount()

_maxTxAmount()

_maxWalletToken()

_allowance()

_balanceOf()

_decimals()

_getOwner()

_isFeeExempt()

_name()

_pair()

_swapThreshold()

_symbol()

_totalSupply()

07 | Potential Resource Exhaustion

Category	Severity	Location	Status
Logical Issue	Informational	projects/contract.sol (98a012): 614, 709	Acknowledged

Description

The farloop() within functions and _getCurrentSupply() takes the variable _excluded.length(), as the maximal iteration times. If the size of the array is very large, it could exceed the gas limit to execute the functions. In this case, the contract might suffer from DoS (Denial of Service) situation.

Recommendation

We recommend the team review the design and ensure investors that this would not cause loss to the project.

08 | FullInlinerNonExpressionSplitArgumentEvaluationOrder

Description

Function call arguments in Yul are evaluated right to left. This order matters when the argument expressions have side-effects, and changing it may change contract behavior. FullInliner is an optimizer step that can replace a function call with the body of that function. The transformation involves assigning argument expressions to temporary variables, which imposes an explicit evaluation order. FullInliner was written with the assumption that this order does not necessarily have to match usual argument evaluation order because the argument expressions have no side-effects. In most circumstances this assumption is true because the default optimization step sequence contains the ExpressionSplitter step. ExpressionSplitter ensures that the code is in *expression-split form*, which means that function calls cannot appear nested inside expressions, and all function call arguments have to be variables. The assumption is, however, not guaranteed to be true in general. Version 0.6.7 introduced a setting allowing users to specify an arbitrary optimization step sequence, making it possible for the FullInliner to actually encounter argument expressions with side-effects, which can result in behavior differences between optimized and unoptimized bytecode. Contracts compiled without optimization or with the default optimization sequence are not affected. To trigger the bug the user has to explicitly choose compiler settings that contain a sequence with FullInliner step not preceded by ExpressionSplitter.

09 | StorageWriteRemovalBeforeConditionalTermination

Description

A call to a Yul function that conditionally terminates the external EVM call could result in prior storage writes being incorrectly removed by the Yul optimizer. This used to happen in cases in which it would have been valid to remove the store, if the Yul function in question never actually terminated the external call, and the control flow always returned back to the caller instead. Conditional termination within the same Yul block instead of within a called function was not affected. In Solidity with optimized via-IR code generation, any storage write before a function conditionally calling ``return(...)`` or ``stop()`` in inline assembly, may have been incorrectly removed, whenever it would have been valid to remove the write without the ``return(...)`` or ``stop()``. In optimized legacy code generation, only inline assembly that did not refer to any Solidity variables and that involved conditionally-terminating user-defined assembly functions could be affected.

10 | DelegateCallReturnValue

Description

The return value of the low-level .delegatecall() function is taken from a position in memory, where the call data or the return data resides. This value is interpreted as a boolean and put onto the stack. This means if the called function returns at least 32 zero bytes, .delegatecall() returns false even if the call was successful.

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanismto relocate funds.

Logical Issue

Logical Issue findingsdetail a fault in the logic of the linked code, such as an incorrect notion on how block.times tamp works.

Volatile Code

Volatile Code findingsrefer to segmentsof code that behave unexpectedly on certain edge cases that may resultin a vulnerability.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather commenton how to make the codebase more legible and, as a result, easily maintainable.

Inconsistency

Inconsistency findings referto functions that should seemingly behave similarly yet contain different code, such as a constructor assignment imposing different require statements on the input variables than a setterfunction.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexa-decimal encoded and is the same as the output of the Linux "sha256sum" commandagainst the target file.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimerand limitation of liability) set forth in the ServicesAgreement, or the scope of services, and terms and conditions providedto you (“Customer” or the “Company”) in connection with the Agreement. This report providedin connection with the Servicesset forth in the Agreementshall be used by the Company only to the extent permittedunder the terms and conditions set forth in the Agreement.
This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes,nor may copiesbe delivered to any otherperson other than the Company,without FusionTech prior written consentin each instance.

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts FusionTech to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, businessmodel or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor shouldbe leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. FusionTech’s position is that each companyand individual are responsible for their own due diligenceand continuous security.
FusionTech’s goal is to help reduce the attack vectorsand the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by FusionTech is subject to dependencies and under continuing development. You agree that your access and/or use, includingbut not limited to any services, reports,and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokensare emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon,multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTSOR RESULTS OF THE USE THEREOF ARE PROVIDED “AS IS” AND “AS

AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUTWARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, FusionTech HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, FusionTech SPECIFICALLY DISCLAIMS ALL IMPLIEDWARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, FusionTech MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE,ACCURATE, COMPLETE, FREE OF HARMFULCODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, FusionTech PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDSOR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHERFusionTech NOR ANY OF FusionTech'SAGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. FusionTech WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTINGFROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES,ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED"AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THETHIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSENOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIESBE DELIVERED TO, ANY OTHERPERSON WITHOUT FusionTech'SPRIOR WRITTEN CONSENTIN EACH INSTANCE.

NOTHIRD PARTY OR ANYONE ACTINGON BEHALF OF ANY THEREOF,SHALL BE A THIRD PARTYOR OTHER BENEFICIARY OF SUCH SERVICES,ASSESSMENT REPORT, AND ANY ACCOMPANYING

FORAVOIDANCE OF DOUBT, THE SERVICES,INCLUDING ANY ASSOCIATEDASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

About

Founded in 2022 by leading academics in the field of Computer Science, FusionTech is going to be a leading blockchain security company that serves to verify the security and correctness of smart contracts KYC and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of ourclients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.

FusionTech

https://fusiontech.live

https://twitter.com/fusiontechh

https://t.me/fusiontechofficial

https://github.com/fusiontechofficial