QuillAudits

Audit Report
June, 2021

DEFI 11

Contents

Introduction

This Audit Report highlights the overall security of the DeFi 11 Smart Contract. With this report, we have tried to ensure the reliability of their smart contract by a complete assessment of their system's architecture and the smart contract codebase.

Auditing Approach and Methodologies applied

The QuillAudits team has performed thorough testing of the project, starting with analysing the code design patterns in which we reviewed the smart contract architecture to ensure it is structured and safe use of third party smart contracts and libraries.

Our team then performed a formal line by line inspection of the Smart Contract to find any potential issue like race conditions, transaction-ordering dependence, timestamp dependence, and denial of service attacks.

In the Unit testing Phase, we coded/conducted Custom unit tests written for each function in the contract to verify that each function works as expected. In Automated Testing, We tested the Smart Contract with our in-house developed tools to identify vulnerabilities and security flaws.

The code was tested in collaboration with our multiple team members, and this included -

- Testing the functionality of the Smart Contract to determine proper logic has been followed throughout the process.
- Analysing the complexity of the code by thorough, manual review of the code, line-by-line.
- Deploying the code on testnet using multiple clients to run live tests
- Analysing failure preparations to check how the Smart Contract performs in case of bugs and vulnerabilities.
- Checking whether all the libraries used in the code are on the latest version.
- Analysing the security of the on-chain data.

Audit Details

Project Name: DeFi 11
Github Commit: e4c57beedc8e0bdc6b935ec45dda898069bfa5c5
Correction commit: dcda9e53c68df097d14ead5e4a8a95b05004c3f4
Languages: Solidity (Smart contract), Javascript (Unit Testing)
Platforms and Tools: Remix IDE, Truffle, Truffle Team, Ganache, Slither, Surya

Summary of Smart Contract

QuillAudits conducted a security audit of a smart contract of DeFi 11. DeFi 11 contracts are used to create smart wallets, registry contract to register wallet addresses and forwardProxy contract.

And some advanced features other than essential functions.

- Wallet
- Forwards signed transactions to the user's wallet for executing logic.
- Deploy wallets and track

Audit Goals

The focus of the audit was to verify that the smart contract system is secure, resilient and working according to its specifications. The audit activities can be grouped into the following three categories:

Security
Identifying security related issues within each contract and the system of contract.

Sound Architecture
Evaluation of the architecture of this system through the lens of established smart contract best practices and general software best practices.

Code Correctness and Quality

A full review of the contract source code. The primary areas of focus include:

- Accuracy
- Readability
- Sections of code with high complexity
- Quantity and quality of test coverage

Security Level references

Every issue in this report was assigned a severity level from the following:

High level severity issues

Issues on this level are critical to the smart contract's performance/ functionality and should be fixed before moving to a live environment.

Medium level severity issues

Issues on this level could potentially bring problems and should eventually be fixed.

Low level severity issues

Issues on this level are minor details and warnings that can remain unfixed but would be better fixed at some point in the future.

Number of issues per severity

	Low	Medium	High	Recommendations
Open	1	0	0	0
Closed	3	0	0	0

Manual Audit

High level severity issues

No high severity issues

Medium level severity issues

No medium severity issues

Low level severity issues

1. **Solidity version is not locked**

   ForwardRegistry Contract use pragma solidity ^0.4.24;

   Always use a locked version of solidity pragma solidity 0.4.24;
   So that development and deployment will always have the same version.
   Please change it to the latest version as well with locks.

   Status: Fixed by Developer

2. **Solidity version is not the same**

   ForwardRegistry Contract use pragma solidity ^0.4.24;

   Other contract uses different version pragma solidity ^0.7.0;

   Registry contract uses pragma solidity ^0.7.0;

   Use a locked version with the same in all files.
   We would suggest you use the same solidity version in all the files and make necessary changes based on solidity version syntax.
   The version has been updated with initial fixes but not locked.

   Status: Fixed by Developer

3. **Function that can be declared external**

   Smart Wallet

   SmartWallet.onERC721Received (SmartWallet.sol#) should be declared **external**

SmartWallet.onERC1155Received (SmartWallet.sol#) should be declared **external**
SmartWallet.onERC1155BatchReceived (SmartWallet.sol#) should be declared **external**

MultiOwnable Contract

MultiOwnable.setOwners (multiOwnable.sol#41-46) should be declared **external**
MultiOwnable.transferOwnership (multiOwnable.sol#62-64) should be **declared external**
MultiOwnable.voteToRemoveOwner (multiOwnable.sol#79-84) should be declared **external**
MultiOwnable.updateOwner (multiOwnable.sol#91-99) should be declared **external**

Status: Fixed by Developer

4. **Try not to Use Experimental features in live deployment**

pragma experimental ABIEncoderV2;

There is an array that contains elements with a size less than 32 bytes that have elements that share a storage slot or members of type bytesNN shorter than 32 bytes.

Function execute : bytes[] calldata datas

Try to fit it using either way as it will not arise vulnerability:

- if all your structs or arrays only use uint256 or int256 types
- if you only use integer types (that may be shorter) and only encode at most one array at a time

Status: The DeFi11 team considered this Exhibit and chose to leave the code as it is to avoid significant code changes in the codebase at its current state.

Functional test

We did functional tests for different contracts as well manually. Below is the report.

SmartWallet.sol

- registry provide registry address
  -- > PASS

- Owner provide owner address
  --> PASS

- nonce transaction count
  --> PASS

- execute executes multiple delegate calls
  --> PASS

- getHash return hash
  --> PASS

- executeMetaTransaction execute meta transaction for user
  --> PASS

- executeMetaTransactionUsingForwardProxy execute meta transaction coming from forwardProxy
  --> PASS

- recover Recovers user address from the hash and signature
  --> PASS

ForwardProxy.sol

- getHash return hash
  -- > PASS

- forward forwards transaction to user's smart wallet
  --> PASS

DefiRegistry.sol

- setFee will set fees
  --> PASS

- changeFeeRecipient set new fees recipient
  --> PASS

- withdraw withdraw all the ERC20 and ethereum from contract to address
  --> PASS

multiOwnable.sol

- setOwners will add owner by manager only
  --> PASS

- transferOwnership it will transfer ownership to another address and can be called by current owner only
  --> PASS

- voteToRemoveOwner only owners can vote to remove owner from owners list
  --> PASS

- updateOwner update owner only if voting eligibility meet
  --> PASS

Automated Testing

Slither Tool Result

MultiOwnable.sol

```
MultiOwnable.setOwners (multiOwnable.sol#41-46) should be declared external
MultiOwnable.transferOwnership (multiOwnable.sol#62-64) should be declared external
MultiOwnable.voteToRemoveOwner (multiOwnable.sol#79-84) should be declared external
MultiOwnable.updateOwner (multiOwnable.sol#91-99) should be declared external
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#public-function-that-could-be-declared-as-external
INFO:Detectors:
Detected issues with version pragma in multiOwnable.sol:
        - pragma solidity0.5.16 (multiOwnable.sol#1): it allows old versions
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#incorrect-version-of-solidity
INFO:Detectors:
Parameter '_owners' of MultiOwnable.setOwners (multiOwnable.sol#41) is not in mixedCase
Function 'MultiOwnable._setOwners' (multiOwnable.sol#49-56) is not in mixedCase
Parameter '_owners' of MultiOwnable._setOwners (multiOwnable.sol#49) is not in mixedCase
Function 'MultiOwnable._transferOwnership' (multiOwnable.sol#67-73) is not in mixedCase
Parameter '_owner' of MultiOwnable.voteToRemoveOwner (multiOwnable.sol#79) is not in mixedCase
Parameter '_owner' of MultiOwnable.updateOwner (multiOwnable.sol#91) is not in mixedCase
Parameter '_newOwner' of MultiOwnable.updateOwner (multiOwnable.sol#91) is not in mixedCase
Constant 'MultiOwnable.maxOwners' (multiOwnable.sol#7) is not in UPPER_CASE_WITH_UNDERSCORES
Constant 'MultiOwnable.minVotesNeeded' (multiOwnable.sol#15) is not in UPPER_CASE_WITH_UNDERSCORES
A Terminal : https://github.com/trailofbits/slither/wiki/Detectors-Documentation#conformance-to-solidity-naming-conventions
```

Defi11Registry.sol

```
LogicRegistry.updateProxy (Defi11Registry.sol#58-60) should be declared external
Defi11Registry.setFee (Defi11Registry.sol#104-107) should be declared external
Ownable.owner (Ownable.sol#27-29) should be declared external
Ownable.renounceOwnership (Ownable.sol#53-56) should be declared external
Ownable.transferOwnership (Ownable.sol#62-64) should be declared external
SmartWallet.onERC721Received (SmartWallet.sol#198-207) should be declared external
SmartWallet.onERC1155Received (SmartWallet.sol#208-218) should be declared external
SmartWallet.onERC1155BatchReceived (SmartWallet.sol#219-230) should be declared external
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#public-function-that-could-be-declared-as-external
INFO:Detectors:
Detected issues with version pragma in Defi11Registry.sol:
        - pragma solidity^0.5.16 (Defi11Registry.sol#1): it allows old versions
        - pragma solidity^0.5.16 (IERC20.sol#1): it allows old versions
        - pragma solidity^0.5.16 (IRegistry.sol#1): it allows old versions
        - pragma solidity^0.5.16 (Ownable.sol#1): it allows old versions
        - pragma solidity^0.5.16 (SmartWallet.sol#1): it allows old versions
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#incorrect-version-of-solidity
INFO:Detectors:
Low level call in SmartWallet.executeMetaTransaction (SmartWallet.sol#127-136):
        -(success) = target.call(data) SmartWallet.sol#132
Low level call in SmartWallet.executeMetaTransactionUsingForwardProxy (SmartWallet.sol#142-145):
        -address(this).call(data) SmartWallet.sol#144
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#low-level-calls
INFO:Detectors:
Parameter '_logicAddress' of LogicRegistry.logic (Defi11Registry.sol#25) is not in mixedCase
Parameter '_logicAddress' of LogicRegistry.enableLogic (Defi11Registry.sol#31) is not in mixedCase
Parameter '_logicAddresses' of LogicRegistry.enableLogicMultiple (Defi11Registry.sol#39) is not in mixedCase
Parameter '_logicAddress' of LogicRegistry.disableLogic (Defi11Registry.sol#50) is not in mixedCase
Parameter '_newProxy' of LogicRegistry.updateProxy (Defi11Registry.sol#58) is not in mixedCase
Parameter '_fee' of Defi11Registry.setFee (Defi11Registry.sol#104) is not in mixedCase
Parameter '_feeRecipient' of Defi11Registry.changeFeeRecipient (Defi11Registry.sol#113) is not in mixedCase
Function 'Ownable._transferOwnership' (Ownable.sol#69-73) is not in mixedCase
Parameter '_registry' of SmartWallet. (SmartWallet.sol#65) is not in mixedCase
Parameter '_user' of SmartWallet. (SmartWallet.sol#65) is not in mixedCase
Function 'SmartWallet._execute' (SmartWallet.sol#76-99) is not in mixedCase
Parameter '_target' of SmartWallet._execute (SmartWallet.sol#76) is not in mixedCase
Parameter '' of SmartWallet.onERC721Received (SmartWallet.sol#199-200) is not in mixedCase
Parameter '_scope_0' of SmartWallet.onERC721Received (SmartWallet.sol#200-201) is not in mixedCase
```

```
INFO:Detectors:
SmartWallet.recover (SmartWallet.sol#152-197) is declared view but contains assembly code
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#constant-functions-changing-the-state
INFO:Detectors:
Contract locking ether found in Defi11Registry.sol:
        Contract SmartWallet has payable functions:
         - execute (SmartWallet.sol#106-113)
         - executeMetaTransaction (SmartWallet.sol#127-136)
         - executeMetaTransactionUsingForwardProxy (SmartWallet.sol#142-145)
        But does not have a function to withdraw the ether
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#contracts-that-lock-ether
INFO:Detectors:
Reentrancy in SmartWallet.executeMetaTransaction (SmartWallet.sol#127-136):
        External calls:
        - (success) = target.call(data) (SmartWallet.sol#132)
        State variables written after the call(s):
        - nonce (SmartWallet.sol#135)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#reentrancy-vulnerabilities-1
INFO:Detectors:
Defi11Registry.withdraw (Defi11Registry.sol#128-140) does not use the value returned by external calls:
        -erc20Contract.transfer(recipient,amount) (Defi11Registry.sol#138)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#unused-return
INFO:Detectors:
SmartWallet._execute uses assembly (SmartWallet.sol#76-99)
        - SmartWallet.sol#81-98
SmartWallet.recover uses assembly (SmartWallet.sol#152-197)
        - SmartWallet.sol#166-170
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#assembly-usage
INFO:Detectors:
Different versions of Solidity is used in Defi11Registry.sol:
        - Version used: ['ABIEncoderV2', '^0.5.16']
        - Defi11Registry.sol#1 declares pragma solidity^0.5.16
        - IERC20.sol#1 declares pragma solidity^0.5.16
        - IRegistry.sol#1 declares pragma solidity^0.5.16
        - Ownable.sol#1 declares pragma solidity^0.5.16
        - SmartWallet.sol#1 declares pragma solidity^0.5.16
        - SmartWallet.sol#2 declares pragma experimentalABIEncoderV2
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#different-pragma-directives-are-used
```

ForwardProxy.sol

```
INFO:Slither:Defi11Registry.sol analyzed (10 contracts), 44 result(s) found
abhi@crypticocean:~/Downloads/defi11-1/defi-audit$ ls
Defi11Registry.sol  ForwardProxy.sol  IERC20.sol  IRegistry.sol  Ownable.sol  SafeMath.sol  SmartWallet.sol
abhi@crypticocean:~/Downloads/defi11-1/defi-audit$ slither ForwardProxy.sol
INFO:Detectors:
ForwardProxy.recover (ForwardProxy.sol#72-116) is declared view but contains assembly code
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#constant-functions-changing-the-state
INFO:Detectors:
Reentrancy in ForwardProxy.forward (ForwardProxy.sol#48-56):
        External calls:
        - require(bool,string)(address(IRegistry(registry).wallets(recover(_hash,sig))) != address(0),Can only be used by defi11 users) (Forwa
rdProxy.sol#51)
        - require(bool,string)(address(IRegistry(registry).wallets(recover(_hash,sig))) == destination,User can only send txn to his own walle
t) (ForwardProxy.sol#52)
        State variables written after the call(s):
        - nonce (ForwardProxy.sol#54)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#reentrancy-vulnerabilities-1
INFO:Detectors:
ForwardProxy.executeCall uses assembly (ForwardProxy.sol#64-68)
        - ForwardProxy.sol#65-67
ForwardProxy.recover uses assembly (ForwardProxy.sol#72-116)
        - ForwardProxy.sol#86-90
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#assembly-usage
INFO:Detectors:
ForwardProxy.forward (ForwardProxy.sol#48-56) should be declared external
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#public-function-that-could-be-declared-as-external
INFO:Detectors:
Detected issues with version pragma in ForwardProxy.sol:
        - pragma solidity^0.5.16 (ForwardProxy.sol#1): it allows old versions
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#incorrect-version-of-solidity
INFO:Detectors:
Parameter '_registry' of ForwardProxy. (ForwardProxy.sol#30) is not in mixedCase
Parameter '' of ForwardProxy.getHash (ForwardProxy.sol#37) is not in mixedCase
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#conformance-to-solidity-naming-conventions
INFO:Slither:ForwardProxy.sol analyzed (2 contracts), 8 result(s) found
```

SmartWallet.sol

```
SmartWallet.onERC721Received (SmartWallet.sol#198-207) should be declared external
SmartWallet.onERC1155Received (SmartWallet.sol#208-218) should be declared external
SmartWallet.onERC1155BatchReceived (SmartWallet.sol#219-230) should be declared external
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#public-function-that-could-be-declared-as-external
INFO:Detectors:
Detected issues with version pragma in SmartWallet.sol:
        - pragma solidity^0.5.16 (IERC20.sol#1): it allows old versions
        - pragma solidity^0.5.16 (IRegistry.sol#1): it allows old versions
        - pragma solidity^0.5.16 (SmartWallet.sol#1): it allows old versions
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#incorrect-version-of-solidity
INFO:Detectors:
Low level call in SmartWallet.executeMetaTransaction (SmartWallet.sol#127-136):
        -(success) = target.call(data) SmartWallet.sol#132
Low level call in SmartWallet.executeMetaTransactionUsingForwardProxy (SmartWallet.sol#142-145):
        -address(this).call(data) SmartWallet.sol#144
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#low-level-calls
INFO:Detectors:
Parameter '_registry' of SmartWallet. (SmartWallet.sol#65) is not in mixedCase
Parameter '_user' of SmartWallet. (SmartWallet.sol#65) is not in mixedCase
Function 'SmartWallet._execute' (SmartWallet.sol#76-99) is not in mixedCase
Parameter '_target' of SmartWallet._execute (SmartWallet.sol#76) is not in mixedCase
Parameter '' of SmartWallet.onERC721Received (SmartWallet.sol#199-200) is not in mixedCase
Parameter '_scope_0' of SmartWallet.onERC721Received (SmartWallet.sol#200-201) is not in mixedCase
Parameter '_scope_1' of SmartWallet.onERC721Received (SmartWallet.sol#201-202) is not in mixedCase
Parameter '_scope_2' of SmartWallet.onERC721Received (SmartWallet.sol#202-203) is not in mixedCase
Parameter '' of SmartWallet.onERC1155Received (SmartWallet.sol#209-210) is not in mixedCase
Parameter '_scope_0' of SmartWallet.onERC1155Received (SmartWallet.sol#210-211) is not in mixedCase
Parameter '_scope_1' of SmartWallet.onERC1155Received (SmartWallet.sol#211-212) is not in mixedCase
Parameter '_scope_2' of SmartWallet.onERC1155Received (SmartWallet.sol#212-213) is not in mixedCase
Parameter '_scope_3' of SmartWallet.onERC1155Received (SmartWallet.sol#213-214) is not in mixedCase
Parameter '' of SmartWallet.onERC1155BatchReceived (SmartWallet.sol#220-221) is not in mixedCase
Parameter '_scope_0' of SmartWallet.onERC1155BatchReceived (SmartWallet.sol#221-222) is not in mixedCase
Parameter '_scope_1' of SmartWallet.onERC1155BatchReceived (SmartWallet.sol#222-223) is not in mixedCase
Parameter '_scope_2' of SmartWallet.onERC1155BatchReceived (SmartWallet.sol#223-224) is not in mixedCase
Parameter '_scope_3' of SmartWallet.onERC1155BatchReceived (SmartWallet.sol#224-225) is not in mixedCase
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#conformance-to-solidity-naming-conventions

INFO:Detectors:
SmartWallet.recover (SmartWallet.sol#152-197) is declared view but contains assembly code
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#constant-functions-changing-the-state
INFO:Detectors:
Contract locking ether found in SmartWallet.sol:
        Contract SmartWallet has payable functions:
         - execute (SmartWallet.sol#106-113)
         - executeMetaTransaction (SmartWallet.sol#127-136)
         - executeMetaTransactionUsingForwardProxy (SmartWallet.sol#142-145)
        But does not have a function to withdraw the ether
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#contracts-that-lock-ether
INFO:Detectors:
Reentrancy in SmartWallet.executeMetaTransaction (SmartWallet.sol#127-136):
        External calls:
        - (success) = target.call(data) (SmartWallet.sol#132)
        State variables written after the call(s):
        - nonce (SmartWallet.sol#135)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#reentrancy-vulnerabilities-1
INFO:Detectors:
SmartWallet._execute uses assembly (SmartWallet.sol#76-99)
        - SmartWallet.sol#81-98
SmartWallet.recover uses assembly (SmartWallet.sol#152-197)
        - SmartWallet.sol#166-170
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#assembly-usage
INFO:Detectors:
Different versions of Solidity is used in SmartWallet.sol:
        - Version used: ['ABIEncoderV2', '^0.5.16']
        - IERC20.sol#1 declares pragma solidity^0.5.16
        - IRegistry.sol#1 declares pragma solidity^0.5.16
        - SmartWallet.sol#1 declares pragma solidity^0.5.16
        - SmartWallet.sol#2 declares pragma experimentalABIEncoderV2
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#different-pragma-directives-are-used
INFO:Detectors:
SmartWallet.onERC721Received (SmartWallet.sol#198-207) should be declared external
SmartWallet.onERC1155Received (SmartWallet.sol#208-218) should be declared external
SmartWallet.onERC1155BatchReceived (SmartWallet.sol#219-230) should be declared external
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#public-function-that-could-be-declared-as-external
```

```
Defi11Registry.sol  ForwardProxy.sol  IERC20.sol  IRegistry.sol  Ownable.sol  SafeMath.sol  SmartWallet.sol
abhi@crypticocean:~/Downloads/defi11-1/defi-audit$ slither SmartWallet.sol
INFO:Slither:Compilation warnings/errors on SmartWallet.sol:
SafeMath.sol:14:1: Warning: Source file does not specify required compiler version! Consider adding "pragma solidity ^0.5.16;"
library SafeMath {
^ (Relevant source part starts here and spans across multiple lines).
SmartWallet.sol:2:1: Warning: Experimental features are turned on. Do not use experimental features on live deployments.
pragma experimental ABIEncoderV2;
^-----------------------------^
SmartWallet.sol:144:7: Warning: Return value of low-level calls not used.
      address(this).call(data);
      ^----------------------^
SmartWallet.sol:198:5: Warning: Function state mutability can be restricted to pure
    function onERC721Received(
    ^ (Relevant source part starts here and spans across multiple lines).
SmartWallet.sol:208:5: Warning: Function state mutability can be restricted to pure
    function onERC1155Received(
    ^ (Relevant source part starts here and spans across multiple lines).
SmartWallet.sol:219:5: Warning: Function state mutability can be restricted to pure
    function onERC1155BatchReceived(
    ^ (Relevant source part starts here and spans across multiple lines).

INFO:Detectors:
SmartWallet.recover (SmartWallet.sol#152-197) is declared view but contains assembly code
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#constant-functions-changing-the-state
INFO:Detectors:
Contract locking ether found in SmartWallet.sol:
        Contract SmartWallet has payable functions:
         - execute (SmartWallet.sol#106-113)
         - executeMetaTransaction (SmartWallet.sol#127-136)
         - executeMetaTransactionUsingForwardProxy (SmartWallet.sol#142-145)
        But does not have a function to withdraw the ether
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#contracts-that-lock-ether
INFO:Detectors:
Reentrancy in SmartWallet.executeMetaTransaction (SmartWallet.sol#127-136):
        External calls:
        - (success) = target.call(data) (SmartWallet.sol#132)
        State variables written after the call(s):
```

Implementation Recommendations

Parameter '_registry' of SmartWallet. (SmartWallet.sol#65) is not in mixedCase

Parameter '_user' of SmartWallet. (SmartWallet.sol#65) is not in mixedCase

Function 'SmartWallet._execute' (SmartWallet.sol#76-99) is not in mixedCase

Parameter '_target' of SmartWallet._execute (SmartWallet.sol#76) is not in mixedCase

Parameter '' of SmartWallet.onERC721Received (SmartWallet.sol#199-200) is not in mixedCase

Parameter '_scope_0' of SmartWallet.onERC721Received (SmartWallet.sol#200-201) is not in mixedCase

Parameter '_scope_1' of SmartWallet.onERC721Received (SmartWallet.sol#201-202) is not in mixedCase

Parameter '_scope_2' of SmartWallet.onERC721Received (SmartWallet.sol#202-203) is not in mixedCase

Parameter '' of SmartWallet.onERC1155Received (SmartWallet.sol#209-210) is not in mixedCase

Parameter '_scope_0' of SmartWallet.onERC1155Received (SmartWallet.sol#210-211) is not in mixedCase

Parameter '_scope_1' of SmartWallet.onERC1155Received (SmartWallet.sol#211-212) is not in mixedCase

Parameter '_scope_2' of SmartWallet.onERC1155Received (SmartWallet.sol#212-213) is not in mixedCase

Parameter '' of SmartWallet.onERC1155Received (SmartWallet.sol#209-210) is not in mixedCase

Parameter '_scope_0' of SmartWallet.onERC1155Received (SmartWallet.sol#210-211) is not in mixedCase

Parameter '_scope_1' of SmartWallet.onERC1155Received (SmartWallet.sol#211-212) is not in mixedCase

Parameter '_scope_2' of SmartWallet.onERC1155Received (SmartWallet.sol#212-213) is not in mixedCase

Parameter '_scope_3' of SmartWallet.onERC1155Received (SmartWallet.sol#213-214) is not in mixedCase

Parameter '' of SmartWallet.onERC1155BatchReceived (SmartWallet.sol#220-221) is not in mixedCase

Parameter '_scope_0' of SmartWallet.onERC1155BatchReceived (SmartWallet.sol#221-222) is not in mixedCase

Parameter '_scope_1' of SmartWallet.onERC1155BatchReceived (SmartWallet.sol#222-223) is not in mixedCase

Parameter '_scope_2' of SmartWallet.onERC1155BatchReceived (SmartWallet.sol#223-224) is not in mixedCase

Parameter '_scope_3' of SmartWallet.onERC1155BatchReceived (SmartWallet.sol#224-225) is not in mixedCase

Disclaimer

Quillhash audit is not a security warranty, investment advice, or an endorsement of the DeFi 11 contract. Securing smart contracts is a multistep process; therefore, running a bug bounty program as a complement to this audit is strongly recommended.

Summary

The use case of the smart contract is very well designed and Implemented. Overall, the code is well written and demonstrates effective use of abstraction, separation of concerns, and modularity. The DeFi 11 development team demonstrated high technical capabilities, both in the design of the architecture and in the implementation.

Some low-severity issues have been reported and documented above, although the DeFi 11 team has now fixed most of them.

DEFI 11

QuillAudits

Canada, India, Singapore and United Kingdom

audits.quillhash.com

hello@quillhash.com