QuillAudits

# Audit Report
# August, 2021

PROXY

# Contents

## Scope of Audit

The scope of this audit was to analyze and document the Relay smart contract codebase for quality, security, and correctness.

## Checked Vulnerabilities

We have scanned the smart contract for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that we considered:

- Re-entrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Use of tx.origin
- Exception disorder
- Gasless send
- Balance equality
- Byte array
- Transfer forwards all gas
- ERC20 API violation
- Malicious libraries
- Compiler version not fixed
- Redundant fallback function
- Send instead of transfer
- Style guide violation
- Unchecked external call
- Unchecked math
- Unsafe type inference
- Implicit visibility level

# Techniques and Methods

Throughout the audit of smart contract, care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments match logic and expected behaviour.
- Token distribution and calculations are as per the intended behaviour mentioned in the whitepaper.
- Implementation of ERC-20 token standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods and tools were used to review all the smart contracts.

**Structural Analysis**
In this step we have analyzed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.
SmartCheck.

**Static Analysis**
Static Analysis of Smart Contracts was done to identify contract vulnerabilities. In this step a series of automated tools are used to test security of smart contracts.

**Code Review / Manual Analysis**
Manual Analysis or review of code was done to identify new vulnerability or verify the vulnerabilities found during the static analysis. Contracts were completely manually analyzed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of automated analysis were manually verified.

**Gas Consumption**
In this step we have checked the behaviour of smart contracts in production. Checks were done to know how much gas gets consumed and possibilities of optimization of code to reduce gas consumption.

**Tools and Platforms used for Audit**
Remix IDE, Truffle, Truffle Team, Ganache, Solhint, Mythril, Slither, SmartCheck.

# Issue Categories

Every issue in this report has been assigned with a severity level. There are four levels of severity and each of them has been explained below.

## High severity issues

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality and we recommend these issues to be fixed before moving to a live environment.

## Medium level severity issues

The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems and they should still be fixed.

## Low level severity issues

Low level severity issues can cause minor impact and or are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

## Informational

These are severity four issues which indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

## Number of issues per severity

| Type | High | Medium | Low | Informational |
|---|---|---|---|---|
| Open | 0 | 0 | 0 | 0 |
| Acknowledged | 0 | 1 | 0 | 0 |
| Closed | 1 | 0 | 3 | 5 |

# Introduction

During the period of **July 04, 2021 to July 07, 2021** - QuillAudits Team performed a security audit for Relay smart contract.

The code for the audit was taken from following the official link: https://github.com/Proxy-Protocol/Matic-Layer2/blob/proxy_relayer/proxy/Relay.sol
**Branch:** security_audit_v3

| Note | Date | Commit hash |
|---|---|---|
| Version 1 | July | 411c04b5893948d1249fac2c3395c39644716954 |
| Version 2 | July | d1da42f8bc164f5577b7590eef178fd1858c6a49 |
| Version 3 | July | 40a5c874f0f2a41f63388f8598296018f1503bc6 |

# Issues Found – Code Review / Manual Testing

## High severity issues

1. Unchecked complex calculations throughout the contract

   Description
   Throughout the contract, calculations are performed without verifying that the values have not overflowed or underflowed. Performing calculations without checking the outputs can lead to severe errors.

   An overflow/underflow happens when an arithmetic operation reaches the maximum or minimum size of a type. For instance, if a number is stored in the uint8 type, it means that the number is stored in an 8 bits unsigned number ranging from 0 to 2^8-1. In computer programming, an integer overflow occurs when an arithmetic operation attempts to create a numeric value that is outside of the range that can be represented with a given number of bits – either larger than the maximum or lower than the minimum representable value.

   **Remediation**
   Given the vital importance of accurate calculations in the protocol, please consider instead implementing calculations using SafeMath, so that underflows and overflows will revert instead of causing cascading errors.

   **Status:** Fixed
   This issue was found fixed in version 3.

## Medium severity issues

### 2. Costly loops leading to DOS attack

**Description**
The role owner has the authority to
update critical settings (processTx, proxessMin)

**Remediation**
We advise the client to handle the governance account carefully to avoid any potential hack. We also advise the client to consider the following solutions:

- with reasonable latency for community awareness on privileged operations;
- Multisig with community-voted 3rd-party independent co-signers;
- DAO or Governance module increasing transparency and community involvement;

**Status: Acknowledged by the Auditee**

## Low level severity issues

### 3. Missing Events For Critical Arithmetic Parameters

| Line | Code |
| --- | --- |
| 610 | function setMintCounter(uint id) external override returns (bool) {<br>require(id > uuid, ERR_MINT_COUNTER_INVALID);<br>uuid = id;<br>return true;<br>} |

**Description**
The missing event makes it difficult to track off-chain changes in the uuid. An event should be emitted for significant transactions calling the setMintCounter function.

**Remediation**
We recommend emitting an event to log the update of the uuid variable. involvement;

**Status:** Closed
This issue was found fixed in version 3.

## 4. Redundant variables

| Line | Code |
| --- | --- |
| 66-67 | uint[] allPendingMints;<br>uint[] allCompletedMints; |
| 126 | string constant ERR_NOT_EXTENSION = "Not extension of chain"; |

### Description
Unused state variables in the above table were found.

### Remediation
We recommend removing those unused variables in the code.

**Status:** Closed
This issue was found fixed in version 3.

## 5. Missing Range Check for Input Variables

### Description
The role can set the txFee variable arbitrarily large or small, causing potential risks in fees and anti whale.

### Remediation
We recommend setting ranges and check the txFee input variable where it’s set by setData function.

**Status:** Closed
This issue was found fixed in version 3.

## Informational

### 6. Conformance to Solidity naming conventions

#### Description

Follow the Solidity naming convention. To favor explicitness and readability, several parts of the contracts may benefit from better naming. Our suggestions are to rename:

- txhex to txHex
- toaddress to toAddress
- txid to txId
- relaydata to relayData
- requiredconfirmations to requiredConfirmations

**Status:** Closed

This issue was found fixed in version 2.

### 7. Comparison to boolean constants

| Line | Code |
| --- | --- |
| 507 | require(found == true, ERR_MINTID_NOT_FOUND); |

#### Description

Boolean constants of the variable found can be used directly and do not need to be compared to true or false.

#### Remediation

We recommend changing the comparison to require(found, ERR_MINTID_NOT_FOUND);

**Status:** Closed

This issue was found fixed in version 2.

## 8. State Variable Default Visibility

| Line | Code |
| --- | --- |
| 64-69 | mapping(uint => Txdata) txsInformation;<br>mapping(address => uint[]) pendingMints;<br>uint[] allPendingMints;<br>uint[] allCompletedMints;<br>mapping(address => uint[]) completedMints;<br>mapping(bytes => uint) relaysMapped;<br>mapping(bytes => bytes) relayDataMapped; |
| 118-142 | string constant ERR_INVALID_HEADER_SIZE = "Invalid block header size";<br>string constant ERR_INVALID_GENESIS_HEIGHT = "Invalid genesis height";<br>string constant ERR_INVALID_HEADER_BATCH = "Invalid block header batch";<br>string constant ERR_DUPLICATE_BLOCK = "Block already stored";<br>string constant ERR_PREVIOUS_BLOCK = "Previous block hash not found";<br>string constant ERR_LOW_DIFFICULTY = "Insufficient difficulty";<br>string constant ERR_DIFF_TARGET_HEADER = "Incorrect difficulty target";<br>string constant ERR_DIFF_PERIOD = "Invalid difficulty period";<br>string constant ERR_NOT_EXTENSION = "Not extension of chain";<br>string constant ERR_BLOCK_NOT_FOUND = "Block not found";<br>string constant ERR_CONFIRMS = "Insufficient confirmations";<br>string constant ERR_VERIFY_TX = "Incorrect merkle proof";<br>string constant ERR_INVALID_TXID = "Invalid tx identifier";<br>string constant ERR_MINIMUM_CONFIRMATION = "Minimum 3 confirmations are required to be set";<br>string constant ERR_INVALID_AMOUNT = "Invalid amount";<br>string constant ERR_RELAY_MAPPED = "Relay data is already mapped";<br>string constant ERR_MINTID_NOT_FOUND = "Corresponding minting id not found";<br>string constant ERR_MINTID_INVALID = "Invalid Mint Id";<br>string constant ERR_MINT_COUNTER_INVALID = "Counter must be above the current value";<br>string constant ERR_INVALID_ADDRESS = "Invalid address";<br>string constant ERR_TX_PROCESSED = "Transaction is already processed";<br>string constant ERR_TX_NOT_FOUND = "Transaction not found";<br>string constant ERR_INVALID_RELAY_DATA = "Invalid relay data";<br>string constant ERR_BTCPX_ADDRESS = "BTCpx MRC20 contract address is not set";<br>string constant ERR_BTCPX_TOKEN_ISSUENCE = "Issuing BTCpx MRC20 tokens failed"; |

**Description**
The Visibility of the above variables is not defined. Labeling the visibility explicitly makes it easier to catch incorrect assumptions about who can access the variable.

The default is internal for state variables, but it should be made explicit.

**Remediation**
We recommend adding the visibility for the above variables. Variables can be specified as being public, internal, or private.

**Status:** Closed
This issue was found fixed in version 2.

## 9. Inconsistent coding style

**Description**
Deviations from the Solidity Style Guide were identified throughout the entire codebase. Taking into consideration how much value a consistent coding style adds to the project's readability, enforcing a standard coding style with the help of linter tools such as Solhint is recommended.

**Status:** Closed
This issue was found fixed in version 3.

## 10. Avoiding Initial Values in Field Declarations

**Description**
The state variable uuid is declared outside the initialize() function. This is equivalent to setting these values in the constructor, and as such, will not work for upgradeable contracts.

**Remediation**
As recommended by the Openzeppelin, make sure that all initial values are set in an initializer function.

**Status:** Closed
This issue was found fixed in version 2.

# Functional test

| Function Names | Testing results |
|---|---|
| processMint | Passed |
| processTx | Passed |
| setMintCounter | Passed |
| submitBlockHeader | Passed |
| submitBlockHeaderBatch | Passed |
| verifyTx | Passed |
| initialize | Passed |
| reinitialize | Passed |
| getBlockHeight() | Passed |
| getBlockHash() | Passed |
| getBestBlock() | Passed |
| getCurrentBlock() | Passed |
| getPendingMints() | Passed |
| getCompletedMints() | Passed |
| getTransactionById() | Passed |
| getCompletedTransactions() | Passed |
| getPendingTransactions() | Passed |

# Automated Testing

## Slither

```
INFO:Detectors:
BTCUtils.retargetAlgorithm(uint256,uint256,uint256) (BTCUtils.sol#643-666) performs a multiplication on the result of a division:
    -_elapsedTime = RETARGET_PERIOD.div(4) (BTCUtils.sol#652)
    -_adjusted = _previousTarget.div(65536).mul(_elapsedTime) (BTCUtils.sol#664)
BTCUtils.retargetAlgorithm(uint256,uint256,uint256) (BTCUtils.sol#643-666) performs a multiplication on the result of a division:
    -_adjusted.div(RETARGET_PERIOD).mul(65536) (BTCUtils.sol#665)
BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) performs a multiplication on the result of a division:
    -sstore(uint256,uint256)(_preBytes,fslot_concatStorage_asm_0 + mload(uint256)(_postBytes + 0x20) / 0x100 ** 32 - mlength_concatStorage_asm_0 * 0x100 ** 32 - newlength_concatStorage_asm_0 + mlength_concatStorage_asm_0 * 2) (BytesLib.sol#135-160)
BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) performs a multiplication on the result of a division:
    -sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (BytesLib.sol#209)
BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) performs a multiplication on the result of a division:
    -sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (BytesLib.sol#243)
BytesLib.equalStorage(bytes,bytes) (BytesLib.sol#342-405) performs a multiplication on the result of a division:
    -fslot_equalStorage_asm_0 = fslot_equalStorage_asm_0 / 0x100 * 0x100 (BytesLib.sol#362)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply
INFO:Detectors:
Parser.decryptData(bytes).v_scope_2 (Parser.sol#71) is a local variable never initialized
Parser.decryptData(bytes).witness (Parser.sol#55) is a local variable never initialized
Parser.readVarInt(bytes,uint256).vi (Parser.sol#120) is a local variable never initialized
Parser.readVarInt(bytes,uint256).vi_scope_0 (Parser.sol#131) is a local variable never initialized
Parser.readValueInt(bytes,uint256).result (Parser.sol#157) is a local variable never initialized
Parser.readVarInt(bytes,uint256).vi_scope_2 (Parser.sol#142) is a local variable never initialized
Relay.find(uint256,address).index (Relay.sol#483) is a local variable never initialized
Parser.readUInt32(bytes,uint256).result (Parser.sol#86) is a local variable never initialized
```

```
Parser.decryptData(bytes).v (Parser.sol#56) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
Parser.decryptData(bytes).witness (Parser.sol#55) shadows:
    - Parser.witness(uint256,uint256,uint256) (Parser.sol#104-112) (function)
Relay.getAmount(Parser.vout[],bytes).outs (Relay.sol#495) shadows:
    - Parser.outs (Parser.sol#25) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
Relay.setMintCounter(uint256) (Relay.sol#524-528) should emit an event for:
    - uuid = id (Relay.sol#526)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (BytesLib.sol#167)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before declaration: sc_concatStorage_asm_0 = keccak256(uint256,uint256)(0x0,0x20) + slength_concatStorage_asm_0 / 32 (BytesLib.sol#215)
Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (BytesLib.sol#181)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before declaration: submod_concatStorage_asm_0 = 32 - slengthmod_concatStorage_asm_0 (BytesLib.sol#224)
Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (BytesLib.sol#181)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before declaration: mc_concatStorage_asm_0 = _postBytes + submod_concatStorage_asm_0 (BytesLib.sol#225)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (BytesLib.sol#182)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before declaration: mc_concatStorage_asm_0 = _postBytes + submod_concatStorage_asm_0 (BytesLib.sol#225)
Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (BytesLib.sol#183)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before declaration: end_concatStorage_asm_0 = _postBytes + mlength_concatStorage_asm_0 (BytesLib.sol#226)
Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (BytesLib.sol#181)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** submod_concatStorage_asm_0 - 1 (BytesLib.sol#227)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (BytesLib.sol#184)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** submod_concatStorage_asm_0 - 1 (BytesLib.sol#227)
```

```
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (BytesLib.sol#167)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (BytesLib.sol#229)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (BytesLib.sol#184)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (BytesLib.sol#229)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (BytesLib.sol#182)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (BytesLib.sol#229)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (BytesLib.sol#167)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before declaration: sc_concatStorage_asm_0 = sc_concatStorage_asm_0 + 1 (BytesLib.sol#232)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (BytesLib.sol#182)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before declaration: mc_concatStorage_asm_0 = mc_concatStorage_asm_0 + 0x20 (BytesLib.sol#233)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (BytesLib.sol#182)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before declaration: mc_concatStorage_asm_0 < end_concatStorage_asm_0 (BytesLib.sol#234)
Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (BytesLib.sol#183)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before declaration: mc_concatStorage_asm_0 < end_concatStorage_asm_0 (BytesLib.sol#234)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (BytesLib.sol#184)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (BytesLib.sol#241)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (BytesLib.sol#182)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (BytesLib.sol#241)
Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (BytesLib.sol#183)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (BytesLib.sol#241)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (BytesLib.sol#167)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before declaration:
sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0)
```

```
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0
(BytesLib.sol#184)' in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially
used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)
(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0)
(BytesLib.sol#243)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (BytesLib.sol#182)'
in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before
declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)
(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0)
(BytesLib.sol#243)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (BytesLib.sol#167)' in
BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before
declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)
(mc_concatStorage_asm_0)) (BytesLib.sol#238)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (BytesLib.sol#182)'
in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before
declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)
(mc_concatStorage_asm_0)) (BytesLib.sol#238)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (BytesLib.sol#167)' in
BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before
declaration: sc_concatStorage_asm_0 = sc_concatStorage_asm_0 + 1 (BytesLib.sol#235)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (BytesLib.sol#182)'
in BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) potentially used before
declaration: mc_concatStorage_asm_0 = mc_concatStorage_asm_0 + 0x20
(BytesLib.sol#236)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-
usage-of-local-variables
INFO:Detectors:
BytesLib.concat(bytes,bytes) (BytesLib.sol#40-109) uses assembly
    - INLINE ASM (BytesLib.sol#43-106)
BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) uses assembly
    - INLINE ASM (BytesLib.sol#112-245)
BytesLib.slice(bytes,uint256,uint256) (BytesLib.sol#248-273) uses assembly
    - INLINE ASM (BytesLib.sol#255-272)
BytesLib.toAddress(bytes,uint256) (BytesLib.sol#275-285) uses assembly
    - INLINE ASM (BytesLib.sol#280-282)
BytesLib.toUint(bytes,uint256) (BytesLib.sol#287-297) uses assembly
    - INLINE ASM (BytesLib.sol#292-294)
BytesLib.equal(bytes,bytes) (BytesLib.sol#299-340) uses assembly
    - INLINE ASM (BytesLib.sol#302-337)
BytesLib.equalStorage(bytes,bytes) (BytesLib.sol#342-405) uses assembly
    - INLINE ASM (BytesLib.sol#345-402)
BytesLib.toBytes32(bytes) (BytesLib.sol#407-415) uses assembly
    - INLINE ASM (BytesLib.sol#412-414)
BytesLib.keccak256Slice(bytes,uint256,uint256) (BytesLib.sol#417-424) uses assembly
```

```
    - INLINE ASM (BytesLib.sol#421-423)
Utils.concat(bytes,bytes) (Utils.sol#5-81) uses assembly
    - INLINE ASM (Utils.sol#15-78)
Utils.slice(bytes,uint256,uint256) (Utils.sol#83-138) uses assembly
    - INLINE ASM (Utils.sol#88-135)
Utils.toBytes32(bytes) (Utils.sol#140-149) uses assembly
    - INLINE ASM (Utils.sol#146-148)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Relay.removeByValue(uint256,address) (Relay.sol#505-509) compares to a boolean constant:
    -require(bool,string)(found == true,ERR_MINTID_NOT_FOUND) (Relay.sol#507)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality
INFO:Detectors:
BTCUtils.determineInputLength(bytes) (BTCUtils.sol#208-217) is never used and should be
removed
BTCUtils.determineOutputLength(bytes) (BTCUtils.sol#307-323) is never used and should be
removed
BTCUtils.determineVarIntDataLength(bytes) (BTCUtils.sol#31-47) is never used and should be
removed
BTCUtils.extractDifficulty(bytes) (BTCUtils.sol#583-585) is never used and should be removed
BTCUtils.extractHash(bytes) (BTCUtils.sol#390-435) is never used and should be removed
BTCUtils.extractInputAtIndex(bytes,uint256) (BTCUtils.sol#153-177) is never used and should
be removed
BTCUtils.extractInputTxIdLE(bytes) (BTCUtils.sol#287-289) is never used and should be
removed
BTCUtils.extractOpReturnData(bytes) (BTCUtils.sol#378-384) is never used and should be
removed
BTCUtils.extractOutpoint(bytes) (BTCUtils.sol#279-281) is never used and should be removed
BTCUtils.extractOutputAtIndex(bytes,uint256) (BTCUtils.sol#330-354) is never used and
should be removed
BTCUtils.extractScriptSig(bytes) (BTCUtils.sol#244-250) is never used and should be removed
BTCUtils.extractScriptSigLen(bytes) (BTCUtils.sol#191-202) is never used and should be
removed
BTCUtils.extractSequenceLELegacy(bytes) (BTCUtils.sol#223-229) is never used and should
be removed
BTCUtils.extractSequenceLEWitness(bytes) (BTCUtils.sol#261-263) is never used and should
be removed
BTCUtils.extractSequenceLegacy(bytes) (BTCUtils.sol#235-239) is never used and should be
removed
BTCUtils.extractSequenceWitness(bytes) (BTCUtils.sol#269-273) is never used and should be
removed
BTCUtils.extractTxIndexLE(bytes) (BTCUtils.sol#295-297) is never used and should be
removed
BTCUtils.extractValue(bytes) (BTCUtils.sol#368-372) is never used and should be removed
BTCUtils.extractValueLE(bytes) (BTCUtils.sol#360-362) is never used and should be removed
```

BTCUtils.hash160(bytes) (BTCUtils.sol#132-134) is never used and should be removed
BTCUtils.isLegacyInput(bytes) (BTCUtils.sol#183-185) is never used and should be removed
BTCUtils.lastBytes(bytes,uint256) (BTCUtils.sol#122-126) is never used and should be removed
BTCUtils.parseVarInt(bytes) (BTCUtils.sol#54-65) is never used and should be removed
BTCUtils.reverseUint256(uint256) (BTCUtils.sol#85-102) is never used and should be removed
BTCUtils.validateVin(bytes) (BTCUtils.sol#446-478) is never used and should be removed
BTCUtils.validateVout(bytes) (BTCUtils.sol#484-516) is never used and should be removed
BytesLib.concat(bytes,bytes) (BytesLib.sol#40-109) is never used and should be removed
BytesLib.concatStorage(bytes,bytes) (BytesLib.sol#111-246) is never used and should be removed
BytesLib.equal(bytes,bytes) (BytesLib.sol#299-340) is never used and should be removed
BytesLib.equalStorage(bytes,bytes) (BytesLib.sol#342-405) is never used and should be removed
BytesLib.keccak256Slice(bytes,uint256,uint256) (BytesLib.sol#417-424) is never used and should be removed
BytesLib.toAddress(bytes,uint256) (BytesLib.sol#275-285) is never used and should be removed
BytesLib.toUint(bytes,uint256) (BytesLib.sol#287-297) is never used and should be removed
Utils.concat(bytes,bytes) (Utils.sol#5-81) is never used and should be removed
ValidateSPV.calculateTxId(bytes,bytes,bytes,bytes) (ValidateSPV.sol#69-77) is never used and should be removed
ValidateSPV.getErrBadLength() (ValidateSPV.sol#27-29) is never used and should be removed
ValidateSPV.getErrInvalidChain() (ValidateSPV.sol#31-33) is never used and should be removed
ValidateSPV.getErrLowWork() (ValidateSPV.sol#35-37) is never used and should be removed
ValidateSPV.validateHeaderChain(bytes) (ValidateSPV.sol#83-115) is never used and should be removed
ValidateSPV.validateHeaderPrevHash(bytes,bytes32) (ValidateSPV.sol#131-140) is never used and should be removed
ValidateSPV.validateHeaderWork(bytes32,uint256) (ValidateSPV.sol#121-124) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Parameter BTCUtils.determineVarIntDataLength(bytes)._flag (BTCUtils.sol#31) is not in mixedCase
Parameter BTCUtils.parseVarInt(bytes)._b (BTCUtils.sol#54) is not in mixedCase
Parameter BTCUtils.reverseEndianness(bytes)._b (BTCUtils.sol#71) is not in mixedCase
Parameter BTCUtils.reverseUint256(uint256)._b (BTCUtils.sol#85) is not in mixedCase
Parameter BTCUtils.bytesToUint(bytes)._b (BTCUtils.sol#108) is not in mixedCase
Parameter BTCUtils.lastBytes(bytes,uint256)._b (BTCUtils.sol#122) is not in mixedCase
Parameter BTCUtils.lastBytes(bytes,uint256)._num (BTCUtils.sol#122) is not in mixedCase
Parameter BTCUtils.hash160(bytes)._b (BTCUtils.sol#132) is not in mixedCase
Parameter BTCUtils.hash256(bytes)._b (BTCUtils.sol#140) is not in mixedCase
Parameter BTCUtils.extractInputAtIndex(bytes,uint256)._vin (BTCUtils.sol#153) is not in mixedCase

Parameter BTCUtils.extractInputAtIndex(bytes,uint256)._index (BTCUtils.sol#153) is not in mixedCase
Parameter BTCUtils.isLegacyInput(bytes)._input (BTCUtils.sol#183) is not in mixedCase
Parameter BTCUtils.extractScriptSigLen(bytes)._input (BTCUtils.sol#191) is not in mixedCase
Parameter BTCUtils.determineInputLength(bytes)._input (BTCUtils.sol#208) is not in mixedCase
Parameter BTCUtils.extractSequenceLELegacy(bytes)._input (BTCUtils.sol#223) is not in mixedCase
Parameter BTCUtils.extractSequenceLegacy(bytes)._input (BTCUtils.sol#235) is not in mixedCase
Parameter BTCUtils.extractScriptSig(bytes)._input (BTCUtils.sol#244) is not in mixedCase
Parameter BTCUtils.extractSequenceLEWitness(bytes)._input (BTCUtils.sol#261) is not in mixedCase
Parameter BTCUtils.extractSequenceWitness(bytes)._input (BTCUtils.sol#269) is not in mixedCase
Parameter BTCUtils.extractOutpoint(bytes)._input (BTCUtils.sol#279) is not in mixedCase
Parameter BTCUtils.extractInputTxIdLE(bytes)._input (BTCUtils.sol#287) is not in mixedCase
Parameter BTCUtils.extractTxIndexLE(bytes)._input (BTCUtils.sol#295) is not in mixedCase
Parameter BTCUtils.determineOutputLength(bytes)._output (BTCUtils.sol#307) is not in mixedCase
Parameter BTCUtils.extractOutputAtIndex(bytes,uint256)._vout (BTCUtils.sol#330) is not in mixedCase
Parameter BTCUtils.extractOutputAtIndex(bytes,uint256)._index (BTCUtils.sol#330) is not in mixedCase
Parameter BTCUtils.extractValueLE(bytes)._output (BTCUtils.sol#360) is not in mixedCase
Parameter BTCUtils.extractValue(bytes)._output (BTCUtils.sol#368) is not in mixedCase
Parameter BTCUtils.extractOpReturnData(bytes)._output (BTCUtils.sol#378) is not in mixedCase
Parameter BTCUtils.extractHash(bytes)._output (BTCUtils.sol#390) is not in mixedCase
Parameter BTCUtils.validateVin(bytes)._vin (BTCUtils.sol#446) is not in mixedCase
Parameter BTCUtils.validateVout(bytes)._vout (BTCUtils.sol#484) is not in mixedCase
Parameter BTCUtils.extractMerkleRootLE(bytes)._header (BTCUtils.sol#528) is not in mixedCase
Parameter BTCUtils.extractTarget(bytes)._header (BTCUtils.sol#536) is not in mixedCase
Parameter BTCUtils.calculateDifficulty(uint256)._target (BTCUtils.sol#550) is not in mixedCase
Parameter BTCUtils.extractPrevBlockLE(bytes)._header (BTCUtils.sol#559) is not in mixedCase
Parameter BTCUtils.extractTimestampLE(bytes)._header (BTCUtils.sol#567) is not in mixedCase
Parameter BTCUtils.extractTimestamp(bytes)._header (BTCUtils.sol#575) is not in mixedCase
Parameter BTCUtils.extractDifficulty(bytes)._header (BTCUtils.sol#583) is not in mixedCase
Parameter BTCUtils.verifyHash256Merkle(bytes,uint256)._proof (BTCUtils.sol#600) is not in mixedCase
Parameter BTCUtils.verifyHash256Merkle(bytes,uint256)._index (BTCUtils.sol#600) is not in mixedCase

Parameter BTCUtils.retargetAlgorithm(uint256,uint256,uint256)._previousTarget (BTCUtils.sol#644) is not in mixedCase
Parameter BTCUtils.retargetAlgorithm(uint256,uint256,uint256)._firstTimestamp (BTCUtils.sol#645) is not in mixedCase
Parameter BTCUtils.retargetAlgorithm(uint256,uint256,uint256)._secondTimestamp (BTCUtils.sol#646) is not in mixedCase
Parameter BytesLib.concat(bytes,bytes)._preBytes (BytesLib.sol#40) is not in mixedCase
Parameter BytesLib.concat(bytes,bytes)._postBytes (BytesLib.sol#40) is not in mixedCase
Parameter BytesLib.concatStorage(bytes,bytes)._preBytes (BytesLib.sol#111) is not in mixedCase
Parameter BytesLib.concatStorage(bytes,bytes)._postBytes (BytesLib.sol#111) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._bytes (BytesLib.sol#248) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._start (BytesLib.sol#248) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._length (BytesLib.sol#248) is not in mixedCase
Parameter BytesLib.toAddress(bytes,uint256)._bytes (BytesLib.sol#275) is not in mixedCase
Parameter BytesLib.toAddress(bytes,uint256)._start (BytesLib.sol#275) is not in mixedCase
Parameter BytesLib.toUint(bytes,uint256)._bytes (BytesLib.sol#287) is not in mixedCase
Parameter BytesLib.toUint(bytes,uint256)._start (BytesLib.sol#287) is not in mixedCase
Parameter BytesLib.equal(bytes,bytes)._preBytes (BytesLib.sol#299) is not in mixedCase
Parameter BytesLib.equal(bytes,bytes)._postBytes (BytesLib.sol#299) is not in mixedCase
Parameter BytesLib.equalStorage(bytes,bytes)._preBytes (BytesLib.sol#342) is not in mixedCase
Parameter BytesLib.equalStorage(bytes,bytes)._postBytes (BytesLib.sol#342) is not in mixedCase
Parameter BytesLib.toBytes32(bytes)._source (BytesLib.sol#407) is not in mixedCase
Parameter BytesLib.keccak256Slice(bytes,uint256,uint256)._bytes (BytesLib.sol#417) is not in mixedCase
Parameter BytesLib.keccak256Slice(bytes,uint256,uint256)._start (BytesLib.sol#417) is not in mixedCase
Parameter BytesLib.keccak256Slice(bytes,uint256,uint256)._length (BytesLib.sol#417) is not in mixedCase
Event IRelaytxProcessed(address,bytes,uint256) (IRelay.sol#30) is not in CapWords
Parameter Ownable.transferOwnership(address)._newOwner (Ownable.sol#38) is not in mixedCase
Struct Parser.vin (Parser.sol#10-16) is not in CapWords
Struct Parser.vout (Parser.sol#18-22) is not in CapWords
Variable Relay._headers (Relay.sol#59) is not in mixedCase
Variable Relay._chain (Relay.sol#61) is not in mixedCase
Variable Relay._forks (Relay.sol#69) is not in mixedCase
Variable Relay._epochStartTarget (Relay.sol#76) is not in mixedCase
Variable Relay._epochEndTarget (Relay.sol#77) is not in mixedCase
Variable Relay._epochStartTime (Relay.sol#79) is not in mixedCase
Variable Relay._epochEndTime (Relay.sol#80) is not in mixedCase
Variable Relay._bestHeight (Relay.sol#83) is not in mixedCase

```
Variable Relay._bestBlock (Relay.sol#84) is not in mixedCase
Variable Relay._currHeight (Relay.sol#85) is not in mixedCase
Variable Relay._currBlock (Relay.sol#86) is not in mixedCase
Parameter SafeMath.mul(uint256,uint256)._a (SafeMath.sol#40) is not in mixedCase
Parameter SafeMath.mul(uint256,uint256)._b (SafeMath.sol#40) is not in mixedCase
Parameter SafeMath.div(uint256,uint256)._a (SafeMath.sol#56) is not in mixedCase
Parameter SafeMath.div(uint256,uint256)._b (SafeMath.sol#56) is not in mixedCase
Parameter SafeMath.sub(uint256,uint256)._a (SafeMath.sol#66) is not in mixedCase
Parameter SafeMath.sub(uint256,uint256)._b (SafeMath.sol#66) is not in mixedCase
Parameter SafeMath.add(uint256,uint256)._a (SafeMath.sol#74) is not in mixedCase
Parameter SafeMath.add(uint256,uint256)._b (SafeMath.sol#74) is not in mixedCase
Parameter Utils.concat(bytes,bytes)._preBytes (Utils.sol#6) is not in mixedCase
Parameter Utils.concat(bytes,bytes)._postBytes (Utils.sol#7) is not in mixedCase
Parameter Utils.slice(bytes,uint256,uint256)._bytes (Utils.sol#83) is not in mixedCase
Parameter Utils.slice(bytes,uint256,uint256)._start (Utils.sol#83) is not in mixedCase
Parameter Utils.slice(bytes,uint256,uint256)._length (Utils.sol#83) is not in mixedCase
Parameter ValidateSPV.prove(bytes32,bytes32,bytes,uint256)._txid (ValidateSPV.sol#47) is
not in mixedCase
Parameter ValidateSPV.prove(bytes32,bytes32,bytes,uint256)._merkleRoot
(ValidateSPV.sol#48) is not in mixedCase
Parameter ValidateSPV.prove(bytes32,bytes32,bytes,uint256)._intermediateNodes
(ValidateSPV.sol#49) is not in mixedCase
Parameter ValidateSPV.prove(bytes32,bytes32,bytes,uint256)._index (ValidateSPV.sol#50) is
not in mixedCase
Parameter ValidateSPV.calculateTxId(bytes,bytes,bytes,bytes)._version (ValidateSPV.sol#70)
is not in mixedCase
Parameter ValidateSPV.calculateTxId(bytes,bytes,bytes,bytes)._vin (ValidateSPV.sol#71) is not
in mixedCase
Parameter ValidateSPV.calculateTxId(bytes,bytes,bytes,bytes)._vout (ValidateSPV.sol#72) is
not in mixedCase
Parameter ValidateSPV.calculateTxId(bytes,bytes,bytes,bytes)._locktime (ValidateSPV.sol#73)
is not in mixedCase
Parameter ValidateSPV.validateHeaderChain(bytes)._headers (ValidateSPV.sol#83) is not in
mixedCase
Parameter ValidateSPV.validateHeaderWork(bytes32,uint256)._digest (ValidateSPV.sol#121) is
not in mixedCase
Parameter ValidateSPV.validateHeaderWork(bytes32,uint256)._target (ValidateSPV.sol#121) is
not in mixedCase
Parameter ValidateSPV.validateHeaderPrevHash(bytes,bytes32)._header (ValidateSPV.sol#131)
is not in mixedCase
Parameter ValidateSPV.validateHeaderPrevHash(bytes,bytes32)._prevHeaderDigest
(ValidateSPV.sol#131) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-
solidity-naming-conventions
```

```
INFO:Detectors:
BTCUtils.reverseUint256(uint256) (BTCUtils.sol#85-102) uses literals with too many digits:
    - v = ((v >> 32) &
0x00000000FFFFFFFF00000000FFFFFFFF00000000FFFFFFFF00000000FFFFFFFF) |
((v &
0x00000000FFFFFFFF00000000FFFFFFFF00000000FFFFFFFF00000000FFFFFFFF) <<
32) (BTCUtils.sol#95-96)
BTCUtils.reverseUint256(uint256) (BTCUtils.sol#85-102) uses literals with too many digits:
    - v = ((v >> 64) &
0x0000000000000000FFFFFFFFFFFFFFFF0000000000000000FFFFFFFFFFFFFFFF) |
((v &
0x0000000000000000FFFFFFFFFFFFFFFF0000000000000000FFFFFFFFFFFFFFFF) <<
64) (BTCUtils.sol#98-99)
BTCUtils.slitherConstructorConstantVariables() (BTCUtils.sol#11-667) uses literals with too
many digits:
    - DIFF1_TARGET =
0xffff0000000000000000000000000000000000000000000000000000
(BTCUtils.sol#16)
BytesLib.toAddress(bytes,uint256) (BytesLib.sol#275-285) uses literals with too many digits:
    - tempAddress = mload(uint256(_bytes + 0x20 + _start) /
0x1000000000000000000000000 (BytesLib.sol#281)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
Relay.allPendingMints (Relay.sol#52) is never used in Relay (Relay.sol#15-605)
Relay.allCompletedMints (Relay.sol#53) is never used in Relay (Relay.sol#15-605)
Relay.ERR_NOT_EXTENSION (Relay.sol#103) is never used in Relay (Relay.sol#15-605)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-
variables
INFO:Detectors:
transferOwnership(address) should be declared external:
    - Ownable.transferOwnership(address) (Ownable.sol#38-40)
decode(bytes) should be declared external:
    - Parser.decode(bytes) (Parser.sol#178-183)
slice(bytes,uint256,uint256) should be declared external:
    - Utils.slice(bytes,uint256,uint256) (Utils.sol#83-138)
bytesToUint(bytes) should be declared external:
    - Utils.bytesToUint(bytes) (Utils.sol#157-163)
flip32Bytes(bytes32) should be declared external:
    - Utils.flip32Bytes(bytes32) (Utils.sol#166-168)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-
that-could-be-declared-external
```

## Mythril

```
The analysis was completed successfully. No issues were detected.
```

## THEO

```
Scanning for exploits in contract: 0xDD42AF0bdF4dD56eC56Fb5158330438C7BfeEbB7
Connecting to HTTP: http://127.0.0.1:8545.
No exploits found. You're going to need to load some exploits.

Tools available in the console:
- `exploits` is an array of loaded exploits found by Mythril or read from a file
- `w3` an initialized instance of web3py for the provided HTTP RPC endpoint
- `dump()` writing a json representation of an object to a local file

Check the readme for more info:
https://github.com/cleanunicorn/theo

Theo version v0.8.2.
```

## THEO

```
Relay.sol
  181:20    error      Only use indent of 16 spaces.              indentation
  182:20    error      Only use indent of 16 spaces.              indentation
  183:20    error      Only use indent of 16 spaces.              indentation
  184:20    error      Only use indent of 16 spaces.              indentation
  185:20    error      Only use indent of 16 spaces.              indentation
  186:20    error      Only use indent of 16 spaces.              indentation
  442:4     warning    Line exceeds the limit of 145 characters    max-len
```

## Mythril

```
Relay.sol
   3:1  error    Compiler version ^0.6.12 does not satisfy the ^0.5.8 semver requirement
compiler-version
  15:1  warning  Contract has 21 states declarations but allowed no more than 15         max-
states-count
  50:5  warning  Explicitly mark visibility of state                                     state-visibility
  51:5  warning  Explicitly mark visibility of state                                     state-visibility
  52:5  warning  Explicitly mark visibility of state                                     state-visibility
  53:5  warning  Explicitly mark visibility of state                                     state-visibility
  54:5  warning  Explicitly mark visibility of state                                     state-visibility
  55:5  warning  Explicitly mark visibility of state                                     state-visibility
  56:5  warning  Explicitly mark visibility of state                                     state-visibility
  95:5  warning  Explicitly mark visibility of state                                     state-visibility
  96:5  warning  Explicitly mark visibility of state                                     state-visibility
  97:5  warning  Explicitly mark visibility of state                                     state-visibility
  98:5  warning  Explicitly mark visibility of state                                     state-visibility
  99:5  warning  Explicitly mark visibility of state                                     state-visibility
 100:5  warning  Explicitly mark visibility of state                                     state-visibility
 101:5  warning  Explicitly mark visibility of state                                     state-visibility
 102:5  warning  Explicitly mark visibility of state                                     state-visibility
 103:5  warning  Explicitly mark visibility of state                                     state-visibility
 104:5  warning  Explicitly mark visibility of state                                     state-visibility
 105:5  warning  Explicitly mark visibility of state                                     state-visibility
 106:5  warning  Explicitly mark visibility of state                                     state-visibility
 107:5  warning  Explicitly mark visibility of state                                     state-visibility
 108:5  warning  Explicitly mark visibility of state                                     state-visibility
 109:5  warning  Explicitly mark visibility of state                                     state-visibility
 110:5  warning  Explicitly mark visibility of state                                     state-visibility
 111:5  warning  Explicitly mark visibility of state                                     state-visibility
 112:5  warning  Explicitly mark visibility of state                                     state-visibility
 113:5  warning  Explicitly mark visibility of state                                     state-visibility
 114:5  warning  Explicitly mark visibility of state                                     state-visibility
 115:5  warning  Explicitly mark visibility of state                                     state-visibility
 116:5  warning  Explicitly mark visibility of state                                     state-visibility
```

# Closing Summary

In this report, we have considered the security of the Relay platform. We performed our audit according to the procedure described above.

The audit showed several high, medium, low, and informational severity issues. In the end, the majority of the issues were fixed or acknowledged by the Auditee.

# Disclaimer

Quillhash audit is not a security warranty, investment advice, or an endorsement of the Relay platform. This audit does not provide a security or correctness guarantee of the audited smart contracts. The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them. Securing smart contracts is a multistep process. One audit cannot be considered enough. We recommend that the Relay Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.


PROXY

QuillAudits

Canada, India, Singapore and United Kingdom

audits.quillhash.com

audits@quillhash.com