PROXIMA
Proxima Interactive Agency

Audit Report for:

NewsCrypto

October 15, 2021

Contents

1 Introduction

NewsCrypto[1] is a comprehensive educational, social and informative platform offering both beginners and expert traders everything they need in order to learn about crypto and further improve their trading skills.

Their platform is an already tested product used by the world's top traders that serves as an all-in-one crypto suite. It provides a comprehensive set of tools for all users, regardless of their current level of knowledge. The platform offers everything from automatic charting tools that help beginners understand the basics of technical analysis to advanced proprietary indicators for expert traders.

Furthermore, they wish to implement a bridge between two platforms, Stellar[2] and Polygon[3]. Stellar is an open network which is used to create, send, and trade digital representations of all forms of money. Polygon is a protocol and a framework for building and connecting Ethereum-compatible blockchain networks.

2 Overview

The project consists of the following components:

- **Redis**

  Redis[4] is an in-memory data structure store, used as a database. The project uses it in order to store transaction memos and payment addresses.

- **RabbitMQ**

  RabbitMQ[5] is used as a communicaton service between components. It is primarily used to alert the TokenSender component about a received transaction.

- **TokenSender**

  This component consumes RabbitMQ messages in order to perform transactions. If it gets alerted by StellarListener about a received Stellar payment, then TokenSender consumes the message and then sends a transaction to the Polygon contract (and vice-versa in a similar manner).

- **StellarListener**

  The main Stellar service which listens for Stellar payments to the main wallet. When a payment gets processed, it publishes a message to RabbitMQ, which then gets consumed by the TokenSender.

- **PolygonListener**

  The main Polygon service. It listens to emitted Withdraw events on the Polygon side. When an event gets processed, it publishes a message to RabbitMQ, which then gets consumed by the TokenSender.

- **PolygonCotract**

  Contains the smart contract for the token which is deployed on the Polygon network. The smart contract itself is the base ERC20 token with Burn, Deposit and Withdraw functions. (Similar to ETH-Polygon bridge).

3 Smart Contract

3.1 Issues

- **Missing NatSpec annotation for the following functions:**
  - `requestLockedTokenData`
  - `fulfill`

  Applied fix: added NatSpec annotation

3.2 Recommendations

- **The following variables should be set to private:**
  - `snapshotLocked`
  - `snapshotSupply`
  - `snapshotTimestamp`
  - `oracle`
  - `jobId`
  - `fee`
  - `stellarUrl`
  - `linkAddress`

  Applied fix: adjusted variable visibility to private

- **Avoid using hardcoded values for the following variables**
  - `linkAddress`
  - `oracle`
  - `jobId`
  - `stellarUrl`

Applied fixes:

- added more arguments to the constructor (the listed variables can be set dynamically on contract initialization)
- added a function that the admin can call to update ChainLink parameters

```
/**
 * @notice called when updating parameters for creating Chainlink requests
 * @dev admin can update the link token address, oracle address, job ID, Stellar
 * account URL and fee amount
 * @param link_ link token address
 * @param oracle_ oracle address
 * @param jobId_ job ID
 * @param stellarUrl_ Stellar account URL
 * @param fee_ Chainlink request payment fee
 */
function setChainlinkParameters(
    address link_,
    address oracle_,
    string calldata jobId_,
    string calldata stellarUrl_,
    uint256 fee_
) public only(DEFAULT_ADMIN_ROLE) {
    _oracle = oracle_;
    _jobId = jobId_;
    _fee = fee_;
    _linkAddress = link_;
    _stellarUrl = stellarUrl_;
    setChainlinkToken(_linkAddress);
    setChainlinkOracle(_oracle);
}
```

- **The constructor should have the public visibility keyword removed**

  Applied fix: removed the "public" keyword

- **Move the `ProofOfReserve` Smart Contract inside the existing Polygon contract**

  Applied fixes:

  - moved the `ProofOfReserve` Smart Contract inside the Polygon Smart Contract directory
  - Polygon SC now inherits `ProofOfReserve` to improve code readability
    * all ChainLink related code is now in `ProofOfReserve`

- **Missing function for withdrawing LINK tokens from the Smart Contract**

  Applied fix: added function for withdrawing LINK tokens and the according tests for it

```
/**
 * @notice called when updating parameters for creating Chainlink requests
 * @dev admin can update the link token address, oracle address, job ID, Stellar
 * account URL and fee amount
 * @param link_ link token address
 * @param oracle_ oracle address
 * @param jobId_ job ID
 * @param stellarUrl_ Stellar account URL
 * @param fee_ Chainlink request payment fee
 */
function setChainlinkParameters(
    address link_,
    address oracle_,
    string calldata jobId_,
    string calldata stellarUrl_,
    uint256 fee_
) public only(DEFAULT_ADMIN_ROLE) {
    _oracle = oracle_;
    _jobId = jobId_;
    _fee = fee_;
    _linkAddress = link_;
    _stellarUrl = stellarUrl_;
    setChainlinkToken(_linkAddress);
    setChainlinkOracle(_oracle);
}
```

References

[1] *NewsCrypto*. 2021. URL: https://newscrypto.io/ (visited on 09/07/2021).

[2] *Stellar*. 2021. URL: https://www.stellar.org/ (visited on 09/07/2021).

[3] *Polygon*. 2021. URL: https://polygon.technology/ (visited on 09/07/2021).

[4] *Redis*. 2021. URL: https://redis.io/ (visited on 09/07/2021).

[5] *RabbitMQ*. 2021. URL: https://www.rabbitmq.com/ (visited on 09/07/2021).