SMART CONTRACT AUDIT (HTTPS://BLOG.COINFABRIK.COM/CATEGORY/SMART-CONTRACTS/SMART-CONTRACT-AUDIT-SMART-CONTRACTS/)

RCN Smart Contracts Audit v2

Pablo Yabo (https://blog.coinfabrik.com/author/pyabo/)

March 22, 2018 (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/rcn-smart-contracts-audit-v2/)

Ripio Credit Network

Contents

Summary

The smart contracts that have been audited were taken from the RCN repository at: https://github.com/ripio/rcn-network/tree/v2 (https://github.com/ripio/rcn-network/tree/v2). The audit is based on the commit ***3ded36151ad55543d16c354e70161852de4061d0***, which was updated to reflect changes at: ***052e5fd4d77301e854d0ecdaadbd785dd91950ce***.

The audited files are:

- NanoLoanEngine.sol: Lending engine
- examples/BasicScoreProvider.sol: Example
- examples/ReferenceCosigner.sol: Example of a cosigner
- examples/ReferenceOracle.sol: Example of an oracle
- interfaces/Cosigner.sol: Interface of cosigners

- interfaces/Engine.sol: Interface of loan engine
- interfaces/ERC721.sol: Interface ERC721 non-fungible token
- interfaces/Oracle.sol: Interface of oracles
- interfaces/Token.sol: Interface of ERC20 fungible token
- utils/BytesUtils.sol: Utility to access bytes array
- utils/Delegable.sol: Permission with valid date ranges for contract users
- utils/Ownable.sol: Defines a contract's owner
- utils/RpSafeMath.sol: Math operations with overflow check
- utils/SimpleDelegable.sol: Simple permission for contract users
- utils/TokenLockable.sol: Withdrawal of arbitrary tokens, with lock/unlock operations

The contracts are RCN lending protocol implementations. RCN is a protocol based on smart contracts which standardize credit lending through blockchain technology. Now, the NanoLoanEngine loans are treated as ERC721 non-fungible tokens.

We did not find any critical issues. However, there is one medium severity issue involving a function that does not validate the input data; a couple of functions can also cause problems if they are used in a contract.

Detailed findings

Medium severity

Read arbitrary data

The function *readBytes32* in BytesUtil.sol reads data from memory, but it can return data past the valid length of the bytes array.

```
function readBytes32(bytes data, uint256 index) internal constant returns (bytes32 o) {
     assembly {
         o := mload(add(data, add(32, mul(32, index))))
     }
 }
```

The assembly function *mload* reads 32 bytes from an arbitrary memory direction, it doesn't check that the direction is valid.

We recommend that the calculated memory direction is inside the data.

For example, in the function *requestCosign* in ReferenceCosigner.sol it uses *readBytes32* from a user-supplied bytes array. But it does not check that the data supplied is of the expected size.

This issue was fixed at commit 35b79591c13bf72fc6a0967901b241e21236927e.

Minor severity

Risk of running out of gas if large iterations take place

The functions *tokenOfOwnerByIndex* and *tokensOfOwner* can potentially iterate over all the loans in the contract.

```
        uint256 totalLoans = totalSupply();
        uint256 loanId;

        // This can iterate over all created loans
        for (loanId = 0; loanId <= totalLoans; loanId++) {
            ..
        }
```

These functions are not used by other contracts in the project, but if they are used by a third party contract there is the possibility of a lock caused by consuming more gas than the current block gas limit.

It was reported to us these functions are not designed to be used from other contracts and they will document the risk out of gas due to high gas consumption.

Observations

- Use of the *constant* modifier is deprecated in favor of *view* or *pure* in recent versions of solidity.

This was fixed at commit *047605361e3f1f95db2984cdd15921d1684ac2cd*

Conclusion

We did not find any serious issue in the analyzed contracts. However, we found one medium severity and one minor issue with a couple of functions.

The medium severity issue was caused by a function that did not validate input data and could be forced to return erroneous data. But it should be easy to fix to verify the length of the input. The minor issues can cause problems only if there are lots of loans and if they are called from within a badly programmed contract.

Overall: We found that the contracts are well written and extensively documented. Also, we noticed while we were performing the audit, that RCN was updating continuously to comply with new solidity compiler version once there was a new version released.

Do you want to know what is Coinfabrik Auditing Process?

Check our A-Z Smart Contract Audit Guide (https://blog.coinfabrik.com/smart-contract-audits-ultimate-security-guide/) or you could request a quote (https://www.coinfabrik.com#contact_us) for your project.

Related Posts

(https://blog.coinfabrik.com/smart-contracts/etherparty-token-smart-contract-security-audit-coinfabrik/)

EtherParty Smart Contract Security Audit (https://blog.coinfabrik.com/smart-contracts/etherparty-token-smart-contract-security-audit-coinfabrik/)

Coinfabrik team has been hired to audit Etherparty smart contracts. Firstly, we will provide a…

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/rcn-basiccosigner-smart-contract-security-audit/)

RCN BasicCosigner Smart Contract Security Audit (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/rcn-basiccosigner-smart-contract-security-audit/)

Coinfabrik security audit's team was asked to audit the BasicCosigner contract. This contract is part…

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/ripio-credit-

RCN Network Smart Contract Audit (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/ripio-credit-network-rcn-smart-contract-audit/)

Coinfabrik's smart contract audit team was commissioned to conduct a security

network-
rcn-
smart-
contract-
audit/)

Coinfabrik's smart contract audit team was commissioned to conduct a security audit of the contracts…

Tags:

- audit (https://blog.coinfabrik.com/tag/audit/)
- ERC20 (https://blog.coinfabrik.com/tag/erc20/)
- RCN (https://blog.coinfabrik.com/tag/rcn/)
- Ripio (https://blog.coinfabrik.com/tag/ripio/)
- Token (https://blog.coinfabrik.com/tag/token/)

SHARE ON

(https://www.facebook.com/sharer/sharer.php?u=https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/rcn-smart-contracts-audit-v2/)
(https://twitter.com/intent/tweet?text=RCN%20Smart%20Contracts%20Audit%20v2&url=https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/rcn-smart-contracts-audit-v2/)
(https://pinterest.com/pin/create/button/?url=&media=https://blog.coinfabrik.com/wp-content/uploads/2018/03/rcn.png&description=RCN+Smart+Contracts+Audit+v2)
(https://www.linkedin.com/shareArticle?mini=true&url=https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/rcn-smart-contracts-audit-v2/&title=RCN%20Smart%20Contracts%20Audit%20v2&source=CoinFabrik%20Blog)

← PREVIOUS ARTICLE

Security Audit – Send (SDT) Token Sale ICO Smart Contract (https://blog.coinfabrik.com/smart-contracts/security-audit-send-sdt-token-sale-ico-smart-contract/)

NEXT ARTICLE →

DreamTeam Smart Contracts Audit for Players' Compensation (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/dreamteam-smart-contract-for-players-compensation/)

You may also like

(https://blog.coinfabrik.com/smart-contracts/magic-bridge-audit/)

Magic Bridge Audit (https://blog.coinfabrik.com/smart-contracts/magic-bridge-audit/)

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/mintingfactoryv2-baseupgradablemarketplace-kodav3upgradablegatedmarketplace/)

2 months ago

Smart Contract Audit (https://blog.coinfabrik.com/category/smart-contracts/smart-contract-audit-smart-contracts/)

MintingFactoryV2, BaseUpgradableMarketplace & KODAV3UpgradableGatedMarketplace (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/mintingfactoryv2-baseupgradablemarketplace-kodav3upgradablegatedmarketplace/)

(https://blog.coinfabrik.com/feed/)

(https://ar.linkedin.com/company/coinfabrik)

(https://twitter.com/coinfabrik)

(https://www.youtube.com/channel/UC2GmjCr7aEz-il31kqOy9aw)

(https://www.facebook.com/CoinFabrik/)

(https://www.reddit.com/r/CoinFabrik/)

(https://github.com/coinfabrik)

© 2021 CoinFabrik - All Rights Reserved.

Made with ❤ in Buenos Aires, Argentina.