CityCoins Audit

Contracts V2

May 2022

By CoinFabrik

Introduction

CoinFabrik was asked to audit the contracts for the CityCoins project. First we will provide a summary of our discoveries, and then we will show the details of our findings.

Scope

The audited files are from the git repository located at https://github.com/citycoins/contracts. The audit is based on the commit `878e4e3d83ef4d50fd0e983dde0f58a8595db35f`.

The audited files are:

- `contracts/cities/mia/mainnet/miamicoin-auth-v2.clar`: Administrative functions for the approvers and city wallet roles.
- `contracts/cities/mia/mainnet/miamicoin-core-v1-patch.clar`: Core contract with functionalities disabled and only a function for CityCoin v1 token burning.
- `contracts/cities/mia/mainnet/miamicoin-core-v2.clar:` Contains mining, staking and reward claiming functions.
- `contracts/cities/mia/mainnet/miamicoin-token-v2.clar:` Token implementation.
- `contracts/cities/nyc/mainnet/newyorkcitycoin-auth-v2.clar`: Administrative functions for the approvers and city wallet roles.
- `contracts/cities/nyc/mainnet/newyorkcitycoin-core-v1-patch.clar`: Core contract with functionalities disabled and only a function for CityCoin v1 token burning.
- `contracts/cities/nyc/mainnet/newyorkcitycoin-core-v2.clar:` Contains mining, staking and reward claiming functions.
- `contracts/cities/nyc/mainnet/newyorkcitycoin-token-v2.clar:` Token implementation.
- `contracts/vrf/mainnet/citycoin-vrf-v2.clar:` Contains the required functions to generate an integer from the specified block's VRF seed.

The scope of the audit is limited to those files. No other files in this repository were audited. Its dependencies are assumed to work according to their documentation. Also, no tests were reviewed for this audit.

Analyses

Without being limited to them, the audit process included the following analyses:

- Arithmetic errors
- Race conditions
- Misuse of block timestamps
- Denial of service attacks
- Excessive gas usage
- Missing or misused function qualifiers
- Needlessly complex code and contract interactions
- Poor or nonexistent error handling
- Insufficient validation of the input parameters
- Centralization and upgradeability

Summary of Findings

We found a critical issue and a minor issue. Also, several enhancements were proposed.

Security Issues

ID	Title	Severity	Status
CR-01	Authorized Remote Code Execution	Critical	Unresolved
MI-01	Inconsistent State After Using a non-Active Core	Minor	Unresolved

Security Issues Found

Severity Classification

Security risks are classified as follows:

- **Critical:** These are issues that we manage to exploit. They compromise the system seriously. They must be fixed **immediately**.
- **Medium:** These are potentially exploitable issues. Even though we did not manage to exploit them or their impact is not clear, they might represent a security risk in the near future. We suggest fixing them **as soon as possible**.
- **Minor:** These issues represent problems that are relatively small or difficult to take advantage of, but can be exploited in combination with other issues. These kinds of issues do not block deployments in production environments. They should be taken into account and be fixed **when possible**.

Issues Status

An issue detected by this audit can have four distinct statuses:

- **Unresolved**: The issue has not been resolved.
- **Acknowledged**: The issue remains in the code, but is a result of an intentional decision.
- **Resolved**: Adjusted program implementation to eliminate the risk.
- **Partially resolved**: Adjusted program implementation to eliminate part of the risk. The other part remains in the code, but is a result of an intentional decision.
- **Mitigated**: Implemented actions to minimize the impact or likelihood of the risk

Critical Severity Issues

CR-01 Authorized Remote Code Execution

Location:

- `contracts/cities/mia/mainnet/miamicoin-auth-v2.clar`,
- `contracts/cities/nyc/mainnet/newyorkcitycoin-auth-v2.clar`

In `set-token-uri()`, `update-coinbase-thresholds()`, `execute-update-coinbase-thresholds()`, `update-coinbase-amounts()`,

`execute-update-coinbase-amounts()`, external code is executed with the auth contract as `tx-sender`. Therefore, if it is called with a malicious contract as argument, this contract could call any of the CityCoins contracts with system authorization.

Moreover, `execute-update-coinbase-thresholds()` and `execute-update-coinbase-amounts()` make these calls based on proposals which were approved without specifying the callee address. Even when the proposal is genuine, a malicious approver could front-run the call to execute it and use malicious contracts as arguments.

Recommendation
Verify the contracts passed as arguments are active, or registered, contracts.

Status
Unresolved.

Medium Severity Issues

No issues found.

Minor Severity Issues

MI-01 Inconsistent State After Using a non-Active Core

Location:
- `contracts/cities/mia/mainnet/miamicoin-auth-v2.clar`,
- `contracts/cities/nyc/mainnet/newyorkcitycoin-auth-v2.clar`

In `upgrade-core-contract()`, a core other than the active one can be passed as the `oldContract`. Therefore, two contracts can have `STATE_ACTIVE`. The same for `execute-upgrade-core-contract-job()`.

In `update-coinbase-thresholds()`, active token thresholds can be updated and a core other than the active one will update its values. However, the active core will remain with the previous thresholds. The same applies for `execute-update-coinbase-thresholds-job()`, `update-coinbase-amounts()`, and `execute-update-coinbase-amounts-job()`.

Recommendation
Verify the state of the core on each function.

Status
Unresolved.

Enhancements

These items do not represent a security risk. They are best practices that we suggest implementing.

Table

ID	Title	Status
EN-01	Unused Constants	Not implemented
EN-02	Unnecessary Stored Values	Not implemented
EN-03	Unnecessary Assertions	Not implemented

Details

EN-01 Unused Constants

Location:
- `contracts/cities/mia/mainnet/miamicoin-core-v2.clar`,
- `contracts/cities/nyc/mainnet/newyorkcitycoin-core-v2.clar`

`CONTRACT_OWNER` and `ERR_UNABLE_TO_FIND_CITY_WALLET` are constants defined in the core contract, but they are not used anywhere on the contract.

Recommendation
Remove unused constants.

Status
Not implemented.

EN-02 Unnecessary Stored Values

Location:
- `contracts/cities/mia/mainnet/miamicoin-core-v2.clar`,
- `contracts/cities/nyc/mainnet/newyorkcitycoin-core-v2.clar`

The `amount` value in the `MiningStatsAtBlock` mapping could be computed based on `amountToCity` and `amountToStackers`. Removing it from the tuple would make writing and reading operations cheaper. The same applies to the `ustx` value in the `MinersAtBlock` mapping, which can be calculated with `highValue` and `lowValue`.

Recommendation
Remove unnecessary storage values.

Status
Not implemented.

EN-03 Unnecessary Assertions

Location:

- `contracts/cities/mia/mainnet/miamicoin-auth-v2.clar`,
- `contracts/cities/nyc/mainnet/newyorkcitycoin-auth-v2.clar`

In `upgrade-core-contract()` and `execute-upgrade-core-contract-job()`, the new core contract's address is compared to the previous one and it reverts if it is equal. However, since the functions get the old core contract from the `CoreContracts` mapping and check if the new address is already stored, the comparison is not necessary. If the old contract's address is in the mapping and the new contract's address is not, they are different.

Recommendation
Remove unnecessary assertions.

Status
Not implemented.

Other Considerations

The considerations stated in this section are not right or wrong. We do not suggest any action to fix them. But we consider that they may be of interest for other stakeholders of the project, including users of the audited contracts, owners or project investors.

Centralization

The auth contract governs the system and it works through a voting system where few designated addresses can participate in (approvers) or through an address (city wallet), which is supposed to be a multisig wallet.

Upgradeability

The active core contract can be changed by the auth contract, through a proposal voted by the approvers or a transaction made by the city wallet. Changing the core contract, the current core-related functionalities could be modified or eliminated. Also, new features could be added.

Privileged Roles

These are the privileged roles that we identified on each of the audited contracts.

miamicoin-auth-v2.clar and newyorkcitycoin-auth-v2.clar

Approvers

Approvers can make proposals and vote on them. Through proposals, approvers can change the active core contract, set a new city wallet, update the coinbase thresholds and amounts, and replace an approver.

City Wallet

The city wallet can change the active core contract, set a new city wallet, set a new token URI, and update coinbase thresholds and amounts.

Changelog

- 2022-05-19 – Initial report based on commit `878e4e3d83ef4d50fd0e983dde0f58a8595db35f`.

Disclaimer: This audit report is not a security warranty, investment advice, or an approval of the CityCoins project since CoinFabrik has not reviewed its platform. Moreover, it does not provide a smart contract code faultlessness guarantee.