CoinFabrik

AlexGo DAO Audit

Alex DAO & Executor DAO

February 2022

By CoinFabrik

Introduction

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

Scope

The contracts audited are from the https://github.com/MarvinJanssen/executor-dao and https://github.com/alexgo-io/alex-dao git repositories. The audit is based on the commits `aa748ceecbad201c226ecbba4d7111995851e7f5` and `9f99127e3aa7487c620982b6f7017c0541a6790d`, respectively. alex-dao was forked from executor-dao. The fixes were applied only on alex-dao's repository and checked on `010c9cf33dae53d4fa2af23dafb5c1b457e44d4c`.

The audited contracts are:

- `executor-dao:`
    - `contracts/executor-dao.clar`: Core contract, which stores the extensions and executes the proposals.
    - `contracts/extensions/ede000-governance-token.clar`: Extension which adds a governance token.
    - `contracts/extensions/ede001-proposal-voting.clar`: This extension adds a voting system for proposals.
    - `contracts/extensions/ede002-proposal-submission.clar`: Extension with mechanism to let users submit proposals.
    - `contracts/extensions/ede003-emergency-proposals.clar`: Extension with a designated team to make emergency proposals.
    - `contracts/extensions/ede004-emergency-execute.clar`: This extension has a set of addresses allowed to execute proposals without voting on them.
    - `contracts/proposals/edp000-bootstrap.clar`: Proposal for an initial construction of the DAO with the basic extensions.
- `alex-dao:`
    - `contracts/executor-dao.clar`
    - `contracts/extensions/age000-governance-token.clar`
    - `contracts/extensions/age001-proposal-voting.clar`
    - `contracts/extensions/age002-emergency-proposals.clar`
    - `contracts/extensions/age003-emergency-execute.clar`
    - `traits/extension-trait.clar`

- `traits/governance-token-trait.clar`
- `traits/ownable-trait.clar`
- `traits/proposal-trait.clar`
- `traits/sip010-ft-trait.clar`

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
- Incorrect handling of cryptographic signatures
- Centralization and upgradeability

Summary of Findings

Two medium issues were found. Also, an enhancement was proposed.

A medium issue was fixed while the other was acknowledged. The enhancement was not implemented.

Security Issues

ID	Title	Severity	Status
ME-01	Lack of Quorum Requirement	Medium	Acknowledged
ME-02	Insecure Authentication through tx-sender	Medium	Resolved

Privileged Roles

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

executor-dao

executor-dao.clar

DAO Deployer

The deployer of `executor-dao.clar` contract is the only address capable of calling `executor-dao::construct()`, which executes any proposal. Once this function is called, the deployer address is overwritten by executor-dao's address, becoming the only which can call the function again.

DAO and its extensions

The DAO and its extensions are allowed to:

- Add, replace, enable and disable extensions.
- Execute any proposal.

ede000-governance-token.clar

DAO and its extensions

The DAO and its extensions are allowed to:

- Mint, burn, transfer, lock and unlock governance tokens without holder consent.
- Set a new token name, symbol, amount of decimals and URI.

ede001-proposal-voting.clar

DAO and its extensions

This role can add new proposals and replace the governance token used in the contract for a new one.

Governance-token holders

Holders can vote for and against the proposals, locking the tokens used for the voting. When the proposal is concluded, they can reclaim their tokens to unlock them.

ede002-proposal-submission.clar

DAO and its extensions

The DAO can modify this extension’s parameters: proposal factor, proposal duration, minimum and maximum delay to start voting on a proposal.

Governance-token majority holders

The holders with more tokens than the proposal factor can submit new proposals to `ede001-proposal-voting.clar` extension.

ede003-emergency-proposals.clar

DAO and its extensions

They can modify the emergency proposal duration, add new emergency proposers and modify the block number when no more emergency proposals will be submitted since then.

Emergency proposers

This role can submit emergency proposals, which do not have a start delay and do not require a minimum holding amount.

ede004-emergency-execute.clar

DAO and its extensions

They can add new emergency executors, modify the amount of signals required to execute a proposal, and modify the block number when no more signals will be emitted since then.

Emergency executors

They can add proposals to the contract and emit signals to execute them when a threshold is reached.

alex-dao

executor-dao.clar

This contract does not include new roles besides the ones described for the contract with the same name in executor-dao repository.

age000-governance-token.clar

This contract includes the “DAO and its extensions” role described for its equivalent `ede000-governance-token.clar` in executor-dao repository, keeping already mentioned entitlements.

Approved contracts

Set of addresses which can mint and burn governance tokens without holder consent.

age001-proposal-voting.clar

This contract does not include new roles besides the ones described for its equivalent `ede001-proposal-voting.clar` in executor-dao repository.

age002-emergency-proposals.clar

This contract does not include new roles besides the ones described for its equivalent `ede003-emergency-proposals.clar` in executor-dao repository.

age003-emergency-execute.clar

This contract does not include new roles besides the ones described for its equivalent `ede004-emergency-execute.clar` in executor-dao repository.

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.
- **Mitigated**: Implemented actions to minimize the impact or likelihood of the risk

Critical Severity Issues

No issues found.

Medium Severity Issues

ME-01 Lack of Quorum Requirement

Location:

- `executor-dao/contracts/extensions/ede001-proposal-voting.clar:124-137`
- `alex-dao/contracts/extensions/age001-proposal-voting.clar:125-138`

Proposals require more votes for than votes against to pass and be executed, without a required minimum participation. This condition might lead to speculation regarding when a proposal would not be noticed (e.g. holidays) or when the token holders would be less likely to vote (e.g. high lock rates and expensive gas fees due to network congestion).

For example, a multiple submission attack can be made with different malicious contracts which can harm the system. Attackers might overload the DAO with these proposals and wait for the last block to vote on the ones which could pass.

Also, emergency proposals, which initially are shorter than the others, can be submitted while another proposal is being voted on. Therefore, the emergency proposal will conclude before the regular one and holders will have not participated because their tokens were locked.

Recommendation

Define and set a quorum requirement for a proposal to pass in the `conclude()` function.

Status

Acknowledged. This is a design decision and, since the recommendation is a mitigation that does not completely address the issue, the project team decides to continue iterating on the proposal system.

ME-02 Insecure Authentication through tx-sender

Location:

- `executor-dao/contracts/ede004-emergency-execute.clar:86`

- `alex-dao/contracts/age003-emergency-execute.clar:87`

Global variable `tx-sender` returns the original sender of the current transaction, or if `as-contract` was called to modify the sending context, it returns that contract principal. Using this variable for authentication is not secure. Actors in the system could be targeted for phishing.

This issue affects especially the contracts with emergency executors because, initially, only one signal is required to execute a proposal.

Recommendation

Prefer `contract-caller` to `tx-sender` for authentication. `contract-caller` returns the caller of the current contract context.

Status

Resolved. Fixed according to the recommendation.

Minor Severity Issues

No issues found.

Enhancements

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

Table

ID	Title	Status
EN-01	Governance Token in Extensions might Differ	Not implemented

Details

EN-01 Governance Token in Extensions might Differ

Location:

- `executor-dao/contracts/extensions/ede002-proposal-submission.clar:45-50`

The proposal submission extension has a function to set the governance token as well as the proposal voting extension. Therefore, if the token is replaced, the new principal should be set in both contracts. Since the submission extension depends on the voting extension, the governance tokens must be synchronized.

Recommendation

Instead of the extension having the token principal stored and a setter, `ede002-proposal-submission.clar` should call the getter in `ede001-proposal-voting.clar` (`get-governance-token()`). Therefore, the value only needs to be updated in the voting extension.

Status

Not implemented. Since this enhancement would increase the runtime cost, the development team has not implemented it.

Other Considerations

DAO and Approved Contracts can burn any token

The DAO and its extensions, and also a set of approved contracts in the case of alex-dao, are authorized to mint and burn any token, even if it is in users' possession.

Centralization

As it was mentioned in the Privileged Roles section, the deployer of `executor-dao.clar` can call `construct()` and execute any proposal, but only once.

Also, a designated team can add emergency proposals with shorter duration, and another team can execute any proposal without voting on it. These privileges are limited on time, but new proposals can extend it forever.

Upgradeability

Through the implementation of extensions with proposals, the DAO can upgrade its functionalities, adding new extensions and disabling previous ones.

Changelog

- 2022-02-10 – Initial report based on commits `9f99127e3aa7487c620982b6f7017c0541a6790d` (alex-dao) and `aa748ceecbad201c226ecbba4d7111995851e7f5` (executor-dao).
- 2022-02-21 – Fixes checked on commit `010c9cf33dae53d4fa2af23dafb5c1b457e44d4c` (alex-dao).

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