MixBytes()

Open Enterprise Template Smart Contract Audit Report

Introduction

General provisions

Aragon is software allowing to freely organize and collaborate without borders or intermediaries. Create global, bureaucracy-free organizations, companies, and communities.

Autark is an Aragon Network organization building open source tools that serve digital cooperatives and aims to revolutionize work by leveraging the corresponding challenges.

With this in mind, MixBytes team was willing to contribute to Aragon ecosystem development by providing security assessment of the Open Enterprise Template smart contract created by Autark.

Scope of the audit

Code written by: Autark

Audited code:

- BaseTemplate.sol version 0e0df6e
- TokenCache.sol version 297a950
- BaseOEApps version 1502373
- OpenEnterpriseTemplate version 1502373

The initial commits for the contracts were reviewed by MixBytes while they were in a work-in-progress stage.

Security Assessment Principles

Classification of Issues

- CRITICAL: Bugs that enable theft of ether/tokens, lock access to funds without possibility to restore it, or lead to any other loss of ether/tokens to be transferred to any party (for example, dividends).
- MAJOR: Bugs that can trigger a contract failure, with further recovery only possible through manual modification of the contract state or contract replacement altogether.
- WARNINGS: Bugs that can break the intended contract logic or enable a DoS attack on the contract.
- COMMENTS: All other issues and recommendations.

Security Assessment Methodology

The audit was performed with triple redundancy by three auditors.

Stages of the audit were as follows:

- “Blind” manual check of the code and model behind the code
- “Guided” manual check of the code
- Check of adherence of the code to requirements of the client
- Automated security analysis using internal solidity security checker
- Automated security analysis using public analysers
- Manual by-checklist inspection of the system
- Discussion and merge of independent audit results
- Report execution

Detected Issues

CRITICAL

Not found

MAJOR

1. OpenEnterpriseTemplate.sol#L177

BaseTemplate.sol#L305

We advise to prohibit the burning of tokens, otherwise `Rewards` will not function properly. As it has not been done since the previous `Rewards` contract audit, we still recommend doing so.

Client: Acknowledged. We can't change this with respect to the template we have here, but will look into providing warnings in the frontend UI when creating merit rewards.

2. OpenEnterpriseTemplate.sol#L114

OpenEnterpriseTemplate.sol#L143

A repeated attempt to get the token from the cache will fail, because the token is removed from the cache during the first call. We recommend abandoning the caches altogether and passing the token in function arguments.

Fixed at OpenEnterpriseTemplate.sol#L143

Client: the contract was in a work-in-progress state, as the review occurred in parallel to finalizing the development.

3. OpenEnterpriseTemplate.sol#L160

This call will not be valid because the current contract is not the `_vault.TRANSFER_ROLE()` permission manager (Voting has already been assigned here OpenEnterpriseTemplate.sol#L172). You can initially set the template as a permission manager, then call `_grantVaultPermissions` and then pass the control to Voting.

Fixed at OpenEnterpriseTemplate.sol#L197

Client: the contract was in a work-in-progress state, as the review occurred in parallel to finalizing the development.

WARNINGS

1. BaseOEApps.sol#L72

The parameters of the `Allocations.initialize` call do not match those in the `Allocations` from the npm-repository as of September 27th. We suggest using the versioning mechanics to ensure that these parameters are consistent.

Fixed at BaseOEApps.sol#L69

2. OpenEnterpriseTemplate.sol#L78

BaseOEApps.sol#L93

DotVoting.sol#L97-L98

There is a type mismatch. It seems that the settings were copied from the `Voting` initialization. We advise checking the code and making explicit type casts.

Fixed at OpenEnterpriseTemplate.sol#L57

3. OpenEnterpriseTemplate.sol#L193

Only the Voting app is able to create DotVoting vote, i.e. DAO members will first have to vote for creating a DotVoting vote. We recommend making sure that this is the desired behavior. As an alternative, any DAO members may be granted a permission to create a DotVoting vote (as it is done in Voting).

Fixed at OpenEnterpriseTemplate.sol#L219

COMMENTS

1. OpenEnterpriseTemplate.sol#L13

As DAO participants are given one token and the `decimals` equals 0, the token as such turns into a boolean flag of the address that belongs to the DAO. In this case, a DotVoting vote is senseless, because there is no way to distribute a vote (i.e. tokens) between several candidates. Additional tokens can be generated, but this will require the DAO to vote. We recommend making sure that this is the desired behavior.

2. BaseOEApps.sol#L64

The `UPDATE_ENTRY_ROLE` permission is not configured.

3. BaseOEApps.sol#L76

The `EXECUTE_ALLOCATION_ROLE`, `EXECUTE_PAYOUT_ROLE`, `CHANGE_PERIOD_ROLE`, and `CHANGE_BUDGETS_ROLE` permissions are not configured.

4. BaseOEApps.sol#L138

The `REMOVE_ISSUES_ROLE`, `FUND_OPEN_ISSUES_ROLE`, and `UPDATE_BOUNTIES_ROLE` permissions are not configured.

5. BaseOEApps.sol#L109

The `ROLE_MODIFY_QUORUM` and `ROLE_MODIFY_CANDIDATE_SUPPORT` permissions are not configured.

6. OpenEnterpriseTemplate.sol#L173-L174

BaseTemplate.sol#L270-L277

The `CHANGE_PERIOD_ROLE`, `CHANGE_BUDGETS_ROLE` permissions are not configured.

7. OpenEnterpriseTemplate.sol#L207-L208

The checks are redundant as they always return the `true` value.

8. BaseOEApps.sol#L41

OpenEnterpriseTemplate.sol#L33

To increase the code readability, you can set individual parameters instead of an array.

CONCLUSION

In case DAO tokens are burned, Rewards app may issue rewards equal to 0. The client regards this as expected behaviour.

The fixed contracts don’t have any vulnerabilities according to our analysis.