Bids, sale, royalties, and transfer-manager smart contracts

Rarible

Independent security assessment report

inference

Report version: 1.0 / date: 07.11.2022

inference

Table of contents

Version / Date	Description
1.0 / 07.11.2022	Final version

Summary

Inference AG was engaged by Rarible to perform an independent security assessment of their bids, sale, royalties, and transfer-manager smart contracts.

Inference AG performed the security assessment based on the agreed scope, following our approach and activities as outlined in the “Project overview” chapter between 4th October 2022 and 31th October 2022. Feedback from Rarible was received during the course of the assessment and Inference performed a follow-up assessment.

Based on our scope and our performed activities, our security assessment revealed one critical and a few medium or low rate security issues. Additionally, different observations were also made which, if resolved appropriately, may improve the quality of Rarible’s smart contracts.

Based on our activities in our follow-up assessment, we can confirm our reported security issues have been resolved. Thus, this report only shows remaining open or partly resolved observations.

Overview on issues and observations

Details for each reported issue or observation can be obtained from the "Security issues" and "Observations" sections.

Row header	Severity / Status
Issues	
There are no open known security issues.	
Observations	
O-RBS-001: Gas exhaustion	- / open
O-RBS-002: Small or non-fungible token payment amounts	- / open

Project overview

Scope

The scope of the security assessment was:

- Sale smart contracts
  - onchain-sales/contracts/sales.arl
  - onchain-sales/contracts/sales-storage.arl
- Bids smart contracts
  - bids/contracts/bids.arl
  - bids/contracts/bids-storage.arl
- transfer-manager contract
  - transfer-manager/contracts/transfer-manager.arl
- Royalties contract
  - royalties-provider/contracts/royalties.arl

All files in scope were made available via a source code repo: https://github.com/rarible/tezos-protocol-contracts and our initial security assessment considered the commit “d52056f803339823faa272f6991b2c1dd94486b4”

Our reassessment considered commit: “c8c547cf79a27c9e10aefd8e26d3f3bc25e249e2” and the following deployed smart contracts on Tezos Mainnet:

- KT193nwmeEg3apR2966xJ9inaxW1b8g2Q438 for bids
- KT1PJcedA18TZmLwTZ6vSXWBYWgXShuVhg4C for bids-storage
- KT1Go3zVcWvpcgWhGWVsXr2GrKf6zGHci8zs for sales
- KT1FqQmyf7C4WuqQhd9APeRpeVr88Rb5E8rb for sales-storage
- KT1CYEoTycbFEuFfJG9Xxbt5XTiSVHopy5rg for transfer-manager
- KT1MVkp4Tb7dJj1GzVbLqaon3QvCM21WkYz6 for royalties

Scope limitations

Our security assessment is based on the following key assumptions and scope limitations:

- Any potential adversarial activities conducted by the administrator of the contract or operational errors by administrators were out of scope.
- Content of metadata was out of scope. For instance: any off-chain views have not been assessed.
- Key management of associated secret keys has not been assessed.
- Content of metadata was out of scope. For instance: any off-chain views have not been assessed.

Methodology

Inference's methodology for smart contract security assessments on Tezos is a combination of a source code review of the smart contract source code in the high-level language (e.g. Ligo or SmartPy), and the resulting compiled code in Michelson. This approach provides additional assurance that the compiler producing the Michelson code is correct, and does not introduce any security issues. Furthermore, this approach fosters a better understanding for smart contract users of what has been reviewed and what has been effectively deployed on-chain.

In order to ensure a high quality in our security assessments, Inference is using subject matter experts having a high adversarial scenario mindset to spot potential issues in smart contracts under review. Additionally, we apply checklists derived from good practices and commonly known issues based on the Tezos smart contract assessment checklist to document our work and to ensure good issue coverage.

Furthermore, Inference maintains regular communications with the smart contract development team to ensure a correct understanding of the smart contract solution and environment, but also to make teams aware of any observations as soon as possible.

Inference's internal quality assurance procedures ensure that results of security assessments are challenged for completeness and appropriateness by a second independent expert.

Objectives

The objectives are the identification of security issues with regards to the assessed smart contracts and their conceptual design and specification. The security assessment also focuses on adversarial scenarios on specific use cases which have been listed in appendix Adversarial scenarios. These were identified together with the Rarible developers and checked during our security assessment.

Activities

Our security assessment activities for the defined scope were:

- Source code review of smart contract code in Archetype

We applied the checklist for smart contract security assessments on Tezos, version 1.1 obtained from https://github.com/InferenceAG/TezosSecurityAssessmentChecklist. We applied the following security checklist tables:

- System / Platform
- Storage
- Gas issues and efficiency
- Code issues
- Transaction
- Entrypoint
- Admin / Operator functions
- Other topics & test cases

Our activities for the follow-up assessment were:

- Source code review of changes in the smart contract code in Archetype
- Source code review of deployed smart contracts in Michelson
- Reassessing security issues and observation from initial assessment in case they are claimed to be resolved

Security issues

There are no open known security issues.

Observations

O-RBS-001: Gas exhaustion

Some data is stored in data structures, which could lead to gas exhaustion in the case where these data structures are too large:

- "Fee receivers" are stored in a "map" data type structure in the transfer-manager contract.
- "Authorised contracts" are stored in a "set" data type structure in the transfer-manager contract.
- "Users" are stored in a "set" data type structure in the royalties smart contract.

This poses a risk that the smart contracts can no longer be executed in cases where these data structures grow too large.

However, since these data structures can only be manipulated by privileged users (admins/owners), this is not critical.

Gas exhaustion may also occur, if too many transactions are emitted due to long lists of stored and provided royalties and fee receivers. However, we rate this not as an issue, since lists can be adapted and also bids or sales can be cancelled. Thus, it does not lead to any locked assets.

Recommendation:
We recommend storing data in "big map" data structures, which causes data to be lazy-loaded.

Comment from Rarible:
In this situation, these elements are only handled by Rarible administrators. The content will always stay small, safe and trusted.

O-RBS-002: Small or non-fungible token payment amounts

If a bid or sale is created with a payment token which cannot be appropriately divided and distributed to the fee/royalties receivers, then fee/royalties receivers do not get any share of fee/royalties.

Recommendation:
We recommend analysing consequences and implementing appropriate measures such as a payment token whitelist. As a minimum contract users should be clearly informed about this behaviour and what the characteristics of an appropriate payment token are.

Comment from Rarible:
This is the intended behavior as Rarible will provide a whitelisted list of FA1.2/FA2 assets through Rarible.com. If users decide to use different tokens, it's their responsibility to use a safe and trusted one.

Disclaimer

This security assessment report (“Report”) by Inference AG (“Inference”) is solely intended for Rarible (“Client”) with respect to the Report’s purpose as agreed by the Client. The Report may not be relied upon by any other party than the Client and may only be distributed to a third party or published with the Client’s consent. If the Report is published or distributed by the Client or Inference (with the Client’s approval) then it is for information purposes only and Inference does not accept or assume any responsibility or liability for any other purpose or to any other party.

Security assessments of a software or technology cannot uncover all existing vulnerabilities. Even an assessment in which no weaknesses are found is not a guarantee of a secure system. Generally, code assessments enable the discovery of vulnerabilities that were overlooked during development and show areas where additional security measures are necessary. Within the Client’s defined time frame and engagement, Inference has performed an assessment in order to discover as many vulnerabilities of the technology or software analysed as possible. The focus of the Report’s security assessment was limited to the general items and code parts defined by the Client. The assessment shall reduce risks for the Client but in no way claims any guarantee of security or functionality of the technology or software that Inference agreed to assess. As a result, the Report does not provide any warranty or guarantee regarding the defect-free or vulnerability-free nature of the technology or software analysed.

In addition, the Report only addresses the issues of the system and software at the time the Report was produced. The Client should be aware that blockchain technology and cryptographic assets present a high level of ongoing risk. Given the fact that inherent limitations, errors or failures in any software development process and software product exist, it is possible that even major failures or malfunctions remain undetected by the Report. Inference did not assess the underlying third party infrastructure which adds further risks. Inference relied on the correct performance and execution of the included third party technology itself.

Appendix

Adversarial scenarios

The following adversarial scenarios have been identified together with Rarible developers and checked during our security assessment.

Scenario	Impact rating & descriptive	Assessment result
Sale: Buyer buys NFTs or NFT bundles, but without having to provide the full price.	Medium: Financial impact for seller, if scenario materialises.	**Ok** Nothing identified.
Sale: Buyer buys more NFTs or NFT bundles than the ones which are for sale.	Medium: Financial impact for seller, if scenario materialises.	**Ok** Nothing identified.
Sale: Buyer/Anyone can extract NFTs from sellers, which are not for sale.	Medium: Financial impact for seller, if scenario materialises.	**Ok** Nothing identified.
Sale: Buyer pays for NFT or NFT bundles but does not receive their bought items or wrong ones.	Medium: Financial impact for buyer, if scenario materialises.	**Ok** Nothing identified.
Sale: Anyone is able to withdraw assets handled by the contract.	High: Loss of assets.	**Ok** Nothing identified. Note: Assets are also not in custody of the sale contracts.
Bids: Bidder bids for an NFT or NFT bundle, but is able to extract more items or other items.	Medium: Financial impact for NFT owners, if scenario materialises.	**Ok** Nothing identified.
Bids: Bidder bids for a NFT or NFT bundle, but receives no items or the wrong ones.	Medium: Financial impact for bidder, if scenario materialises.	**Ok** Nothing identified.
Bids: Anyone is able to withdraw assets handled by the contract.	High: Loss of assets.	**Ok** Nothing identified.

Risk rating definition for smart contracts

Severities are quantified with two dimensions, roughly defined as follows, whereas the examples have to be regarded as indication only:

Probability of occurrence / materialisation of an issue
(bullets for a category are linked with each other with “and/or” condition.)

- Low:
  - A trusted / privileged role is required.
  - Contract may end up in the issue if other conditions, which are also unlikely to happen, are required.
- Medium:
  - A specific role or contract state is required to trigger the issue.
  - Contract may end up in the issue if another condition is fulfilled as well.
- High:
  - Anybody can trigger the issue.
  - Contract’s state will over the short or long term end up in the issue.

Impact:
(bullets for a category are linked with each other with “and/or” condition.)

- Low:
  - Non-compliance with TZIP standards
  - Unclear error messages
  - Confusing structures
- Medium:
  - A minor amount of assets can be withdrawn or destroyed.
- High:
  - Not inline with the specification
  - A non-minor amount of assets can be withdrawn or destroyed.
  - Entire or part of the contract becomes unusable.

Severity:

	Low impact	Medium impact	High impact
High probability	**High**	**Critical**	**Critical**
Medium probability	**Medium**	**High**	**Critical**
Low probability	**Low**	**Medium**	**High**

Glossary

Term	Description
Archetype	High level smart contract language. Website: https://archetype-lang.org/
Ligo	High level smart contract language. Website: https://ligolang.org/
Origination	Deployment of a smart contract
SmartPy	High level smart contract language. Website: https://smartpy.io/
TZIP	Tezos Improvement Proposal