# MyOnions on Alephium

Independent security assessment report

inference

Report version: 1.0 / date: 10.03.2025

inference

# Table of contents


**Table of contents** **2**
**Summary** **4**
Overview on issues and observations 4
**Project overview** **6**
Scope 6
Smart contract security assessment 6
Scope limitations 7
Methodology 8
Objectives 8
Activities 8
**Security issues** **9**
**Observations** **10**
O-MOF-001: Missing slippage or deadline protection 10
O-MOF-002: Token core features are mutable 11
O-MOF-003: Users can pay for the creation of referrer accounts 12
**Disclaimer** **13**
**Appendix** **14**
Adversarial scenarios 14
Risk rating definition for smart contracts 15
Glossary 16

inference

| Version / Date | Description |
| --- | --- |
| 1.0 / 10.03.2025 | Final version. |

## Summary

Inference AG was engaged to perform an independent security assessment of the MyOnions' smart contracts used.

Inference AG performed the security assessment based on the agreed scope, following our approach and activities as outlined in the "Project overview" chapter between the 12th of February 2025 and the 27th of February 2025. Feedback from the MyOnion team was received and Inference performed a reassessment.

Based on our scope and our performed activities, our initial security assessment revealed several security issues rated from critical to low severity. Additionally, different observations were also made, which if resolved with appropriate actions, may improve the quality of MyOnions. During our assessment, the MyOnion team resolved all the reported security issues.

This report only shows remaining open or partly resolved security issues and observations.

## Overview on issues and observations

At Inference AG we separate the findings that we identify in our security assessments in two categories:

- Security issues represent risks to either users of the platform, owners of the contract, the environment of the blockchain, or one or more of these. For example, the possibility to steal funds from the contract, or to lock them in the contract, or to set the contract in a state that renders it unusable are all potential security issues;
- Observations represent opportunities to create a better performing contract, saving gas fees, integrating more efficiently into the existing environment, and creating a better user experience overall. For example, code optimizations that save execution time (and thus gas fees), better compliance to existing standards, and following secure coding best practices are all examples of observations.

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

| | Severity / Status |
|---|---|
| **Security issues** | |
| There are no open known security issues. | |
| **Observations** | |
| O-MOF-001: Missing slippage or deadline protection | - / open |
| O-MOF-002: Token core features are mutable | - / open |
| O-MOF-003: Users can pay for the creation of referrer accounts | - / open |

# Project overview

## Scope

### Smart contract security assessment

The scope of the smart contract security assessment was the following smart contracts:

- TokenLauncher:
  - Source code: contracts/launch/token_launcher.ral
- BondingPair
  - Source code: contracts/trade/bonding_pair.ral
- DexPair
  - Source code: contracts/trade/dex_pair.ral
- OnionRouter:
  - Source code: contracts/trade/onion_router.ral
- FeeCollector
  - Source code: contracts/onion/fee_collector.ral
- FeeHandler
  - Source code: contracts/onion/fee_handler.ral
- MemeToken
  - Source code: contracts/onion/token.ral
- CommentTracker
  - Source code: contracts/value_add/comment_tracker.ral
- ProfileTracker
  - Source code: contracts/value_add/profile_tracker.ral
- Upgradable
  - Source code: contracts/lib/upgradable/upgradable.ral
- FixedPointMath
  - Source code: contracts/lib/math/fixed_point_math.ral

The files in scope were made available via a source code repo: https://github.com/MyOnion-Fun/contracts and our initial security assessment considered commit “d951391e09f0f8a56df527f6465d0e30d4386d78”.

Our reassessment considered commit “b1ec4b51fbf140f1cc91ab252d3fcb50434d6a25”.

## 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.
- Deployment and initial configuration of the deployed smart contract was out of scope.
- The key management of associated secret keys has not been assessed.
- The entities owning privileged roles have not been reviewed, assessed, or vetted in any form.

## Methodology

Inference's methodology for security assessments comprises a source code review in the high-level language, followed by multiple rounds of Q&A with the development team to discuss findings and critical points that emerged during the first assessment. This process is iterated until a version where no new findings emerge is assessed.

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 protocols under review. Additionally, for smart contract security reviews, we apply checklists derived from good practices and commonly known issues to document our work and ensure good coverage.

Furthermore, Inference maintains regular communications with the development team to ensure a correct understanding of the 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 MyOnion team and checked during our security assessment.

## Activities

Our security assessment activities for the defined scope were:

- Source code review of smart contract code written in Ralph

Our activities for the reassessment were:

- Source code review of the changes applied to the smart contract code written in Ralph

- Reassessing security issues and observations from initial assessment in case they are claimed to be resolved
- Discussing the potential future-proofing activities to be conducted by the development team

## Security issues

There are no open known security issues.

# Observations

## O-MOF-001: Missing slippage or deadline protection

The BondingPair contract can only be called by the OnionRouter for token swaps, providing users with slippage and deadline protections. In contrast, the DexPair contract does not offer these protections, because users can call it directly instead of using the OnionRouter.

*Recommendation:*
We suggest implementing a consistent approach across the BondingPair and the DexPair.

*Comment from MyOnion team:*
This is ok, DEXPair is meant to be a low level contract, in case something happens with onion & its team, someone else can spin up a router and continue trading of existing pairs.

*Reassessment:*
We have decided to classify this issue as an observation and add it to the report to increase user caution and awareness when interacting with the DexPair.

## O-MOF-002: Token core features are mutable

Tokens can change their core features. Specifically, the following features can be altered at any time:

- Logo
- Description
- Socials

This can represent a risk to platform users, who may suddenly be associated with ideas they do not support and hold tokens whose nature can be completely altered.

*Recommendation:*
We suggest limiting the possible modifications to a token after its creation to ensure users are protected from potential damage to their reputations related to the tokens they hold.

*Comment from MyOnion team:*
This is acceptable as it allows tokens to evolve, or in case one of their socials gets hacked they can update it, if we notice exploitative behaviour we can show the old details along with the new

*Reassessment:*
We have decided to classify this issue as an observation and add it to the report to increase user awareness of the risks associated with tokens' mutable nature.

## O-MOF-003: Users can pay for the creation of referrer accounts

Referrers can save the cost of creating an account on the platform.
If a referrer delays the creation of their account and waits for another user to add them as referrers, the account creation cost for the referrer will be paid by the first user listing them as referrers.

*Recommendation:*
We recommend altering the “rewardReferrer” process to require non-registered referrers to pay their own fees without exploiting other users.

*Comment from MyOnion team:*
Yes they can avoid paying the storage fee, however implementing another flow would hurt user experience, the cost is negligible so I feel it’s an acceptable trade-off.

*Reassessment:*
We have decided to classify this issue as an observation and add it to the report to increase user awareness.

# Disclaimer

This security assessment report (“Report”) by Inference AG (“Inference”) is solely intended for the “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 and checked during our security assessment.

| Scenario | Assessment result |
|---|---|
| As a normal user, add myself as an owner. | **Ok**<br>Nothing identified. |
| As a normal user, execute restricted functionality. | **Ok**<br>Nothing identified. |
| Exploit improper accounting to obtain more tokens. | **Ok**<br>Nothing identified. |
| Exploit improper accounting to game the trading curve. | **Ok**<br>Nothing identified. |
| Exploit rounding and approximation errors in the math library to gain economical advantage. | **Ok**<br>Nothing identified. |
| Exploit the reward process to gain more XP than what is effectively due. | **Ok**<br>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 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 |
| --- | --- |
| Ralph | High level smart contract language.<br>Website: https://docs.alephium.org/ralph/ |