AlphBanX on Alephium

Independent security assessment report

inference

Report version: 1.1 / date: 24.09.2025

inference

Table of contents

Version / Date	Description
1.0 / 27.05.2025	Final version
1.1 / 24.09.2025	Review of 0% interest loan update

Summary

Inference AG was engaged to perform an independent security assessment of the AlphBanX's 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 the 25th of November 2024 and the 26th of May 2025. Feedback from the AlphBanX team was received and Inference performed a follow-up assessment.

In September 2025, Inference reviewed the changes enabling AlphBanX to issue 0% interest loans to stakeholders.

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 AlphBanX.

During our security assessments, the AlphBanX team resolved all our reported 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 known open security issues.	
Observations	
There are no known open observations.	

Project overview

Scope

Smart contract security assessment

The smart contracts covered in this security assessment include the following contracts, along with any extended contracts, implemented interfaces, and additional created contracts or subcontracts. Note: Contracts or subcontracts that were created are listed in parentheses:

- AbdToken / AbxToken
  - token/AbdToken.ral
  - token/AbxToken.ral
- StakeManager (Staker (LockInfo))
  - staking/StakeManager.ral
  - staking/StakeManagerV2.ral
- Vesting (Schedule, SortedList (ListNode))
  - vesting/Vesting.ral
- AuctionManager (AuctionPool (Bid), AuctionFarming, Bidder)
  - auction/AuctionManager.ral
- LoanManager (SortedLIst(ListNode), InterestPool, Loan)
  - loan/LoanManager.ral
  - loan/LoanManagerV2.ral
- BorrowerOperations
  - loan/BorrowerOperations.ral
  - loan/BorrowerOperationsV2.ral
- DIAAlphPriceAdapter
  - oracle/dia/DIAAlphPriceAdapter.ral
- PlatformSettings:
  - settings/PlatformSettings.ral

The files in scope were made available via a source code repo: https://github.com/FRAGSTARRR/Smart-Contracts---AlphBanX and our initial security assessment considered commit "c314120b262fc343f73a2da9767e4ee06a5693f5"[1].

Our follow-up assessment considered commit "8c44dfe722636a46abfb48430d1c65b7432692ac"[2].

For the 0% interest loan update, our assessment was initially based on commit "0b33339a37ae3de69f5d48f2b90dea35e74a51b2"[3], and the reassessment was conducted using commit "ee6ec0fa84183134db23b5b2ac413e925b623372"[4].

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 were 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.
- Price oracle configuration, data sources, data integrity, and potential manipulation or reliability issues related to the oracle were out of scope.

[1] The sha256sum hash of the repository's ".zip" file is:
0de4d294d3e089e8fccecad3853405dc7a412ba3c22870641f9608a81db75fb7
[2] The sha256sum hash of the repository's ".zip" file is:
e5622593a3e81912b5ae12fcb860d19f74a2aa460627b7def55d2bf489d48e86
[3] The sha256sum hash of the repository's ".zip" file is:
fef4934fa4a247c071a03ab0703dc7c5a5e8921e9715f84ba233aa94c35f68cf
[4] The sha256sum hash of the repository's ".zip" file is:
e34cd74a5aa6a408e049fbf8ac5a844a098d40b3fec28ec56e4e37c1de6bf90e

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. Follow-up assessments are conducted until all identified points, as defined by the team for resolution, have been appropriately addressed.

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 the 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 regard 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 AlphBanX 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 follow-up assessment 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
- Reviewing the documentation on GitBook to ensure it aligns with the reviewed code: https://alphbanx.gitbook.io/. A copy of the reviewed GitBook is attached to this report.

Our activities for the the 0% interest loan update

- Source code review of the changes applied to the smart contract code written in Ralph
- Reassessing security issues and observations from initial update assessment in case they are claimed to be resolved

Security issues

There are no known open security issues.

Observations

There are no known open observations.

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** Nothing identified.
As a normal user, execute restricted functionality.	**Ok** Nothing identified.
Exploit improper accounting to obtain more tokens.	**Ok** Nothing identified.
Exploit rounding and approximation errors to gain economic advantage.	**Ok** Nothing identified.
Prevent liquidation of an undercollateralized loan.	**Ok** Nothing identified.
Prevent the redemption of a loan.	**Ok** Nothing identified.
Pay a lower loan interest rate, while the loan is handled in an interest pool with higher interest rates.	**Ok** Nothing identified.
Place bids with a high discount rate to be preferred over bids with a lower discount rate.	**Ok** Nothing identified.
Place bids to be preferred over other bids of the same discount rate created earlier.	**Ok** Nothing identified.
As a normal user without having a stake to obtain an 0% interest loan.	**Ok** Nothing identified.
A regular stakeholder obtaining a larger 0% interest loan than the amount they are entitled to.	**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 an 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 in line 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. Website: https://docs.alephium.org/ralph/