HashEx

Avalaunch

smart contracts audit report

Prepared for:

avalaunch.app

Authors: HashEx audit team

September 2021

Contents

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and HashEx and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers, and other representatives) (HashEx) owe no duty of care towards you or any other person, nor does HashEx make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties, or other terms of any kind except as set out in this disclaimer, and HashEx hereby excludes all representations, warranties, conditions, and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, HashEx hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against HashEx, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of the use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Introduction

HashEx was commissioned by the Avalaunch team to perform an audit of their smart contracts. The audit was conducted between September 09 and September 13, 2021.

The code located in the github repository @avalaunch-app/xava-protocol was audited after the commit `e781789`. We were asked to track only commits after the audit by CoinFabrik [1], i.e. after the `ac00d9c` commit. The code was provided without documentation. The repository contains tests, but the last commit was pushed August 09, 2 weeks prior to the initial audit.

The purpose of this audit was to achieve the following:

- Identify potential security issues with smart contracts.
- Formally check the logic behind given smart contracts.

Information in this report should be used to understand the risk exposure of smart contracts, and as a guide to improving the security posture of smart contracts by remediating the issues that were identified.

Update. The Avalaunch team has responded to this report. Individual responses were added after each item in the section. The updated code is located in the same repository after the 5b74cd commit. Updated contracts are deployed to the Avalanche C-Chain: `0x8498a16B04E754f873198434063c8684d506A967` AvalaunchSale.

Contracts overview

`AvalaunchSale.sol`

Launchpad contract with vesting. Minor modifications after the last audit.

`AllocationStaking.sol`

Staking contract with rewards in XAVA tokens. Minor modifications after the last audit.

`KuCoinVestingContract.sol`

Simple vesting contract. Minimal modifications after the last audit.

`ParticipationVestingPrivate.sol & ParticipationVestingSeed.sol`

First rounds closed sales. Minimal modifications after the last audit.

`XavaToken.sol, SaleFactory.sol, DevToken.sol, FarminXava.sol, utils & interfaces`

Unmodified since the last audit.

Found issues

ID	Title	Severity	Response
01	AllocationStaking: owner can block withdrawals and redistribute user's balances	High	Responded
02	AvalaunchSale: price may be updated after sale start	Medium	Acknowledged
03	AvalaunchSale: maximum vesting time shift is unlimited	Medium	Fixed
04	AvalaunchSale: vesting percents may be set to zero	Medium	Fixed
05	AvalaunchSale: _portionVestingPrecision can't be set to 100	Low	Fixed

#01 AllocationStaking: owner can block withdrawals and redistribute user's balances — High

The `setSalesFactory()` function in L105 of the AllocationStaking contract allows the owner to bypass the `onlyVerifiedSales` modifier in `redistributeXava()` and `setTokensUnlockTime()` functions. It makes it possible for the owner to reduce staking balances by redistributing them to rewards and burns, or to block withdrawals by setting `user.tokensUnlockTime`.

Recommendation: restrict the owner by transferring the ownership to the Timelock contract with a minimum delay of 1 day.

Team reponse: currently, owner is our cold-storage wallet. We're aiming to migrate it soon to the Governance / Multi-sig contract, thus the functions will be much more safeguarded.

#02	AvalaunchSale: price may be updated after sale start	Medium

`updateTokenPriceInAVAX()` function in L362 of the AvalaunchSale contract was changed since the last audit so it doesn't check the current timestamp and can be called by the owner after the sale's start.

Recommendation: We recommend explicitly mentioning this behavior in the documentation.

Update: function's description was updated: '...to update token price before sale..., will be updated with an oracle during the sale every N minutes'.

#03	AvalaunchSale: maximum vesting time shift is unlimited	Medium

The `setVestingParams()` function in L150 of the AvalaunchSale contract updates the `maxVestingTimeShift` variable without checking the input value and could be called only once. Setting it to an unreasonably high value may lock the vested funds. Error message in L181 suggests that this parameter was intended to be limited from above.

Recommendation: We recommend capping the maximum time shift.

Update: the issue was fixed.

#04	AvalaunchSale: vesting percents may be set to zero	Medium

The `setVestingParams()` function in L150 of the AvalaunchSale contract could be called before the `setSaleParams()` function and therefore with `portionVestingPrecision = 0` in L172.

Recommendation: add a `require` check `portionVestingPrecision > 0` to avoid mistakenly calling the function with uninitialized parameters.

Update: the issue was fixed.

#05	AvalaunchSale: _portionVestingPrecision can't be set to 100	Low

The `setSaleParams()` function of the AvalaunchSale contract should require the variable `_portionVestingPrecision` to be greater than or equal to 100 in L219.

Recommendation: add suggested `require` check.

Update: the issue was fixed.

Conclusion

1 high severity issue was found in the modifications since the last audited commit ac00d9c. It should be noted that the contracts are highly dependent on the owner's account. Users using the project have to trust the owner and that the owner's account is properly secured.

Audit includes recommendations on the code improving and preventing potential attacks.

Update. The Avalaunch team has responded to this report. Regarding the high severity issue the team said that the owner is a cold wallet and later and later ownership will be transferred to a Timelock or Governance contract. Once this is done, the issue will be marked as fixed.
Individual responses were added after each item in the section. The updated code is located in the same repository after the 5b74cd commit. Updated contracts are deployed to the Avalanche C-Chain: 0x8498a16B04E754f873198434063c8684d506A967 .

References

1. Audit by CoinFabrik

Appendix A. Issues' severity classification

We consider an issue critical, if it may cause unlimited losses or break the workflow of the contract and could be easily triggered.

High severity issues may lead to limited losses or break interaction with users or other contracts under very specific conditions.

Medium severity issues do not cause the full loss of functionality but break the contract logic.

Low severity issues are typically nonoptimal code, unused variables, errors in messages. Usually, these issues do not need immediate reactions.

Appendix B. List of examined issue types

Business logic overview

Functionality checks

Following best practices

Access control and authorization

Reentrancy attacks

Front-run attacks

DoS with (unexpected) revert

DoS with block gas limit

Transaction-ordering dependence

ERC/BEP and other standards violation

Unchecked math

Implicit visibility levels

Excessive gas usage

Timestamp dependence

Forcibly sending ether to a contract

Weak sources of randomness

Shadowing state variables

Usage of deprecated code