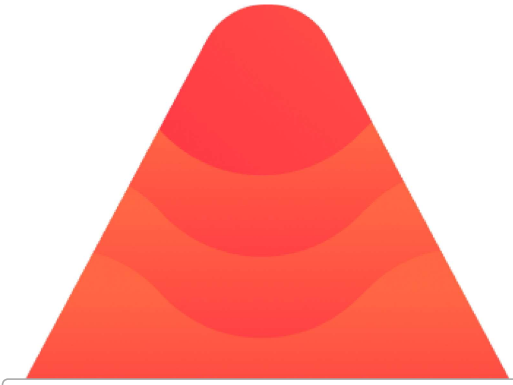
Avalaunch XAVA Protocol Audit

Heloisa Ceni (https://blog.coinfabrik.com/author/heloisa-ceni/)

April 20, 2022 (https://blog.coinfabrik.com/smart-contract-audit/avalaunch-xava-protocol-audit/)



Contents

1. Introduction
1.1. Scope
1.2. Analyses
2. Summary of Findings
2.1. Security Issues
3. Privileged Roles
4. Security Issues Found
4.1. Severity Classification
4.2. Issues Status
4.3. Critical Severity Issues
4.3.1. CR-01 Unlimited Permits Issued for Admin
4.4. Medium Severity Issues
4.5. Minor Severity Issues
4.5.1. MI-01 No Booster Round Is Possible
4.5.2. MI-02 Floating Pragma
4.5.3. MI-03 Inconsistent Parameters in AvalaunchSale and SalesFactory
4.5.4. MI-04 Prefer Importing Libraries from NPM
5. Enhancements
5.1. Table
5.2. Details
5.2.1. EN-01 Outdated Solidity Compiler Version
5.2.2. EN-02 Unused Flattened Library
5.2.3. EN-03 tokenPriceInUSD Is Unused and Misleading
5.2.4. EN-04 dexalotPortfolio May Be Set Before Sale Is Created
5.2.5. EN-05 admin Array Not Checked For Repetitions During Construction
5.2.6. EN-06 Replace Integers by Constants
5.2.7. EN-07 Two safeMath Libraries
5.2.8. EN-08 Documentation
6. Changelog

Introduction

CoinFabrik was asked to audit the contracts for Avalaunch's XAVA Protocol project. First we will provide a summary of our discoveries and then we will show the details of our findings.

Scope

The contracts audited are from the https://github.com/avalaunch-app/xava-protocol/ (https://github.com/avalaunch-app/xava-protocol/) git repository. The audit is based on the commit `fd9b97ccd963819a282fa5c21bf0545d180f8797`. Revisions were made based on commit `8c9f3c021f0a3366e9f8cfc3fd74163f31b57b40.`

The audited contracts are:

- contracts/airdrop/Airdrop.sol
- contracts/airdrop/AirdropAVAX.sol
- contracts/airdrop/AirdropSale.sol
- contracts/farming/DevToken.sol
- contracts/farming/FarmingXava.sol
- contracts/interfaces/IAdmin.sol
- contracts/interfaces/IAllocationStaking.sol
- contracts/interfaces/IAvalaunchSale.sol
- contracts/interfaces/ICollateral.sol
- contracts/interfaces/IDexalotPortfolio.sol
- contracts/interfaces/IERC20Metadata.sol
- contracts/interfaces/ISalesFactory.sol
- contracts/sales/AvalaunchSale.sol
- contracts/sales/SalesFactory.sol
- contracts/utils/Context.sol
- contracts/Admin.sol
- contracts/AllocationStaking.sol
- contracts/AvalaunchColateral.sol
- contracts/IERC20.sol
- contracts/XavaToken.sol

The scope of the audit is limited to those files. No other files in this repository were audited. Its dependencies are assumed to work according to their documentation. Also, no tests were reviewed for this audit.

Analyses

Without being limited to them, the audit process included the following analyses:

- Arithmetic errors
- Outdated version of Solidity compiler
- Race conditions

- Reentrancy attacks

(https://blog.coinfabrik.com/)

- Misuse of block timestamps
- Denial of service attacks

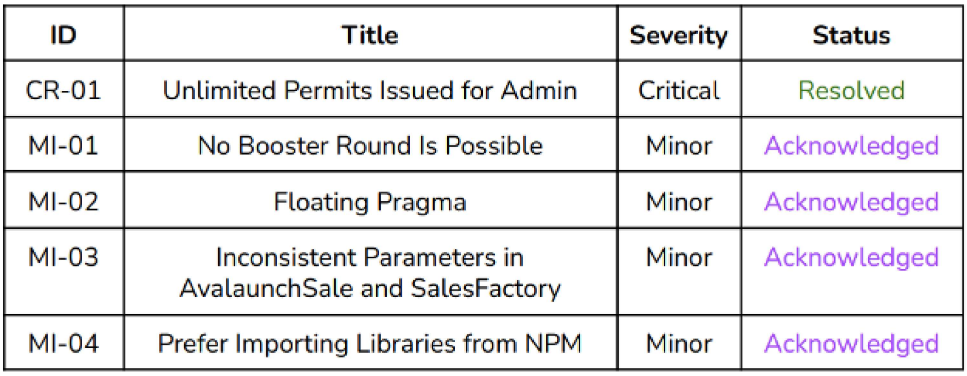
(https://blog.coinfabrik.com/)

- Excessive gas usage
- Missing or misused function qualifiers
- Needlessly complex code and contract interactions
- Poor or nonexistent error handling
- Insufficient validation of the input parameters
- Incorrect handling of cryptographic signatures
- Centralization and upgradeability

Summary of Findings

We found one critical issue and four minor issues. Also, some enhancements were proposed.

Security Issues



ID	Title	Severity	Status
CR-01	Unlimited Permits Issued for Admin	Critical	Resolved
MI-01	No Booster Round Is Possible	Minor	Acknowledged
MI-02	Floating Pragma	Minor	Acknowledged
MI-03	Inconsistent Parameters in AvalaunchSale and SalesFactory	Minor	Acknowledged
MI-04	Prefer Importing Libraries from NPM	Minor	Acknowledged

(https://blog.coinfabrik.com/wp-content/uploads/2022/04/avalaunch-11.png)

Privileged Roles

The audit encompasses 20 contracts including interfaces and libraries. Contracts like AllocationStaking and FarmingXava are Ownable. The owner is responsible for setting sale parameters in the first case, and adding pools, setting allocation points in the second case.

Other contracts, like AllocationStaking, AvalaunchBadgeFactory, AvalaunchCollateral, the airdrop contracts (Airdrop, AirdropAVAX, AirdropSale), AvalaunchSale and SaleFactory make use of the Admin interface, setting an administrator at construction.

The new AvalaunchCollateral contact further introduces the role of the moderator who is responsible for approving sales to have collateral as per this contract.

Security Issues Found

Severity Classification

(https://blog.coinfabrik.com/)

Security risks are classified as follows:

(https://blog.coinfabrik.com/)

● **Critical**: These are issues that we manage to exploit. They compromise the system seriously. They must be fixed immediately.

● **Medium:** These are potentially exploitable issues. Even though we did not manage to exploit them or their impact is not clear, they might represent a security risk in the near future. We suggest fixing them as soon as possible.

● **Minor:** These issues represent problems that are relatively small or difficult to take advantage of but can be exploited in combination with other issues. These kinds of issues do not block deployments in production environments. They should be taken into account and be fixed when possible.

Issues Status

An issue detected by this audit can have four distinct statuses:

● Unresolved: The issue has not been resolved.

● Acknowledged: The issue remains in the code but is a result of an intentional decision.

● Resolved: Adjusted program implementation to eliminate the risk.

● Partially resolved: Adjusted program implementation to eliminate part of the risk. The other part remains in the code but is a result of an intentional decision.

● Mitigated: Implemented actions to minimize the impact or likelihood of the risk.

Critical Severity Issues

CR-01 Unlimited Permits Issued for Admin

Location:

- `contracts/AvalaunchCollateral.sol:135-176,`
- `contracts/AvalaunchCollateral.sol:192-226`

The admin can call autoParticipate() or boostParticipation() in the contract AvalaunchCollateral and act on behalf of a user, paying with this user's collateral. The user only allows the sale (address), but cannot limit amounts,

Recommendation

Include an amount (or maximum amount) that the user allows the admin to use.

Status

Resolved. Developers informed us that this is the expected use. Users must therefore trust that the admin will select the amount safely in their interests.

Medium Severity Issues

No issues found.

Minor Severity Issues

MI-01 No Booster Round Is Possible

In `AvalaunchSale.setSalesParams()` we set the boosterRoundId as

```
boosterRoundId = _stakingRoundId.add(1);
```

after the following check

```
require(_stakingRoundId > 0, "Invalid staking round id.");
```

However, since stakingRoundId may be equal to roundIds.length-1 and boosterRoundId could point to a nonexistent round, thus rendering the booster round unavailable to all practical purposes.

Recommendation

Validate that `_stakingRoundId` is not the last index. Alternatively, revisit the order in which parameters are set to prevent inconsistencies between number of rounds, and specific rounds.

Status

Acknowledged. Developers explained that this is an expected behavior and the booster round is optional.

MI-02 Floating Pragma

Location:

- `contracts/AvalaunchCollateral.sol:2`
- `contracts//DevToken.sol:2`
- `contracts/interfaces/IAvalaunchSale.sol:2`
- `contracts/interfaces/IDexalotPortfolio.sol:2`
- `contracts/interfaces/ICollateral.sol:2`
- `contracts/interfaces/IERC20Metadata.sol:3`
- `contracts/utils/Context.sol:3`

Contracts should be deployed with the same compiler version that they have been tested with. Locking the pragma helps to ensure that contracts do not accidentally get deployed using, for example, an outdated compiler version that might introduce bugs that affect the contract system negatively.

Recommendation

Lock the pragma version, replacing pragma solidity `^0.8.0;` with a specific patch, preferring the most updated version. For example, pragma solidity `0.8.14;`.

Status

Acknowledged.

MI-03 Inconsistent Parameters in AvalaunchSale and SalesFactory

There is a problem if the collateral parameter set in SalesFactory is different to that set in AvalaunchSales.

Recommendation

Set the parameter in only one contract.

Status

Acknowledged.

MI-04 Prefer Importing Libraries from NPM

Library `IERC20Metadata.sol` is stored locally and imported, instead of using NPM to import it from openzeppelin's repository. This may lead to errors in copying, or lose an update.

Recommendation

Import all externally-generated code through NPM.

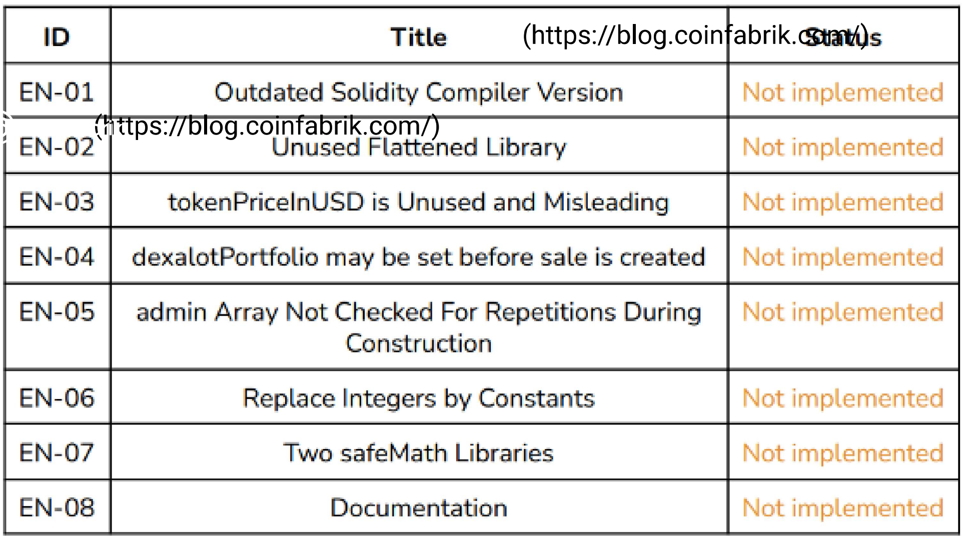
Status

Acknowledged.

Enhancements

These items do not represent a security risk. They are best practices that we suggest implementing.

Table



(https://blog.coinfabrik.com/)

(https://blog.coinfabrik.com/)

ID	Title	Status
EN-01	Outdated Solidity Compiler Version	Not implemented
EN-02	Unused Flattened Library	Not implemented
EN-03	tokenPriceInUSD is Unused and Misleading	Not implemented
EN-04	dexalotPortfolio may be set before sale is created	Not implemented
EN-05	admin Array Not Checked For Repetitions During Construction	Not implemented
EN-06	Replace Integers by Constants	Not implemented
EN-07	Two safeMath Libraries	Not implemented
EN-08	Documentation	Not implemented

(https://blog.coinfabrik.com/wp-content/uploads/2022/04/avalaunch22.png)

Details

EN-01 Outdated Solidity Compiler Version

The audited contracts use the outdated version of solidity v0.6.12 (SWC-102 (https://swcregistry.io/docs/SWC-102)).

Recommendation

Consider updating the code to compile with the latest version.

Status

Not implemented.

EN-02 Unused Flattened Library

Flattened old version of OpenZeppelin's TransparentUpgradeableProxy. If not using the latest version, devs should specify what version they are using so we can check for reported issues.

Recommendation

Document the usage of this variable and allow modifications.

Status

Not implemented.

EN-03 tokenPriceInUSD Is Unused and Misleading

The variable tokenPriceInUSD in `AvalauncSale.sol` is misleading, since it is not used by any of the sale functions nor needs to have any relationship with `tokenPriceInAVAX`. Also tokenIpriceInAVAX may be changed but this cannot.

Recommendation

(https://blog.coinfabrik.com/)

Document the usage of this variable and allow modifications.

(https://blog.coinfabrik.com/)

Status

Not implemented.

EN-04 dexalotPortfolio May Be Set Before Sale Is Created

Ensure the sale is created in `setAndSupportDexalotPortfolio()`

EN-05 admin Array Not Checked For Repetitions During Construction

Location:

- **Admin.sol:18-24.**

In the constructor of Admin.sol it could happen that an address is repeated in the input array `_addresses.` If this happens, the same address would be pushed more than once to the array admins and may create inconsistencies, e.g., when calling `removeAdmin()` . We recommend you check i `sAdmin[_admins[i]]` = false before turning it to true.

EN-06 Replace Integers by Constants

In `FarmingXava.sol` replace occurrences of 1e18 and 1e36 by a call or a constant (e.g., `IERC20Metadata(address(token)).decimals()` as it is done in the `AvalaunchSale.sol` contract).

EN-07 Two safeMath Libraries

Some contracts import OpenZeppelin's safeMath library and others use a custom safeMath library (`math/safeMath.sol)` which is a copy of OpenZeppelin's safeMath with some modifications. Also, note that some contracts, like `AirdropSale.sol,` are importing safeMath but not using it. In that case, consider removing and saving gas.

EN-08 Documentation

Use Solidity's NatSpec format for documentation (link (https://docs.soliditylang.org/en/develop/natspec-format.html)), removing "todo"s and other development comments.

Changelog

- 2022-03-22 – Initial report based on commit f `d9b97ccd963819a282fa5c21bf0545d180f8797.`
- 2022-04-05 – Revision based on commit 8 `c9f3c021f0a3366e9f8cfc3fd74163f31b57b40.`

Disclaimer: This audit report is not a security warranty, investment advice, or an approval of the Avalaunch XAVA Protocol project since CoinFabrik has not reviewed its platform. Moreover, it does not provide a smart contract code faultlessness guarantee.

Related Posts

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/emdx-protocol-audit/)

EMDX - Protocol Audit (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/emdx-protocol-audit/)

Introduction CoinFabrik was asked to audit the contracts for the EMDX project. First, we will…

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/rcn-smart-contracts-audit-v2/)

RCN Smart Contracts Audit v2 (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/rcn-smart-contracts-audit-v2/)

The smart contracts that have been audited were taken from the RCN repository at: https://github.com/ripio/rcn-network/tree/v2.…

Mokens League Audit (https://blog.coinfabrik.com/auditoria-smart-contracts/mokens-league-audit/)

(https://blog.coinfabrik.com/auditoria-smart-contracts/mokens-league-audit/)

Introduction CoinFabrik was asked to audit the contracts for the Mokens League project. First, we…

(https://blog.coinfabrik.com/)

(https://blog.coinfabrik.com/)

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/emdx-token-contracts-audit/)

EMDX - Token Contracts Audit (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/emdx-token-contracts-audit/)

Introduction CoinFabrik was asked to audit the contracts for the EMDX project. First, we will…

Tags:

- audit (https://blog.coinfabrik.com/tag/audit/)
- avalanche (https://blog.coinfabrik.com/tag/avalanche/)
- avalaunch (https://blog.coinfabrik.com/tag/avalaunch/)
- ethereum (https://blog.coinfabrik.com/tag/ethereum/)
- ethereum audit (https://blog.coinfabrik.com/tag/ethereum-audit/)
- Smart Contract (https://blog.coinfabrik.com/tag/smart-contract/)
- smart contract audits (https://blog.coinfabrik.com/tag/smart-contract-audits/)
- solidity audit (https://blog.coinfabrik.com/tag/solidity-audit/)
- xava protocol (https://blog.coinfabrik.com/tag/xava-protocol/)

SHARE ON

(https://www.facebook.com/sharer/sharer.php?u=https://blog.coinfabrik.com/smart-contract-audit/avalaunch-xava-protocol-audit/)

(https://twitter.com/intent/tweet?text=Avalaunch%20XAVA%20Protocol%20Audit&url=https://blog.coinfabrik.com/smart-contract-audit/avalaunch-xava-protocol-audit/)

(https://pinterest.com/pin/create/button/?url=&media=https://blog.coinfabrik.com/wp-content/uploads/2022/04/avalaunch-logo.png&description=Avalaunch+XAVA+Protocol+Audit)

(https://www.linkedin.com/shareArticle?mini=true&url=https://blog.coinfabrik.com/smart-contract-audit/avalaunch-xava-protocol-audit/&title=Avalaunch%20XAVA%20Protocol%20Audit&source=CoinFabrik%20Blog)

← PREVIOUS ARTICLE

(https://blog.coinfabrik.com/)

Mokens League Audit (https://blog.coinfabrik.com/auditoria-smart-contracts/mokens-league-audit/)

NEXT ARTICLE →

(https://blog.coinfabrik.com/)

AlexGo Audit Launchpad, Yield Vault and Collateral Rebalancing Pool (https://blog.coinfabrik.com/smart-contracts/alexgo-auditbrlaunchpad-yield-vault-and-collateral-rebalancing-pool/)

You may also like

(https://blog.coinfabrik.com/smart-contracts/magic-bridge-audit/)

Magic Bridge Audit (https://blog.coinfabrik.com/smart-contracts/magic-bridge-audit/)

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/mintingfactoryv2-baseupgradablemarketplace-kodav3upgradablegatedmarketplace/)

2 months ago

Smart Contract Audit (https://blog.coinfabrik.com/category/smart-contracts/smart-contract-audit-smart-contracts/)

MintingFactoryV2, BaseUpgradableMarketplace & KODAV3UpgradableGatedMarketplace (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/mintingfactoryv2-baseupgradablemarketplace-kodav3upgradablegatedmarketplace/)

(https://blog.coinfabrik.com/feed/)

(https://ar.linkedin.com/company/coinfabrik)

(https://blog.coinfabrik.com/)

(https://twitter.com/coinfabrik)

(https://blog.coinfabrik.com/)

(https://www.youtube.com/channel/UC2GmjCr7aEz-il31kqOy9aw)

(https://www.facebook.com/CoinFabrik/)

(https://www.reddit.com/r/CoinFabrik/)

(https://github.com/coinfabrik)

Made with ♥ in Buenos Aires, Argentina.