CERTIK

Avalaunch

Xava Protocol

Security Assessment

May 31st, 2021

[Preliminary Report]

Disclaimer

CertiK reports are not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security review.

CertiK Reports do not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

CertiK Reports should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.

CertiK Reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

What is a CertiK report?

- A document describing in detail an in depth analysis of a particular piece(s) of source code provided to CertiK by a Client.
- An organized collection of testing results, analysis and inferences made about the structure, implementation and overall best practices of a particular piece of source code.
- Representation that a Client of CertiK has completed a round of auditing with the intention to increase the quality of the company/product's IT infrastructure and or source code.

Overview

Project Summary

Project Name	Avalaunch - Xava Protocol
Description	A typical ERC20 implementation with enhanced features and a staking pool.
Platform	Ethereum; Solidity, Yul
Codebase	GitHub Repository
Commits	1. 8291eac7077e1bf65684352316e753b05fa3b42e

Audit Summary

Delivery Date	May 31st, 2021
Method of Audit	Static Analysis, Manual Review
Consultants Engaged	1
Timeline	May 26th, 2021 - May 31st, 2021

Vulnerability Summary

Total Issues	6
● **Total Critical**	0
● **Total Major**	1
● **Total Medium**	0
● **Total Minor**	2
● **Total Informational**	3

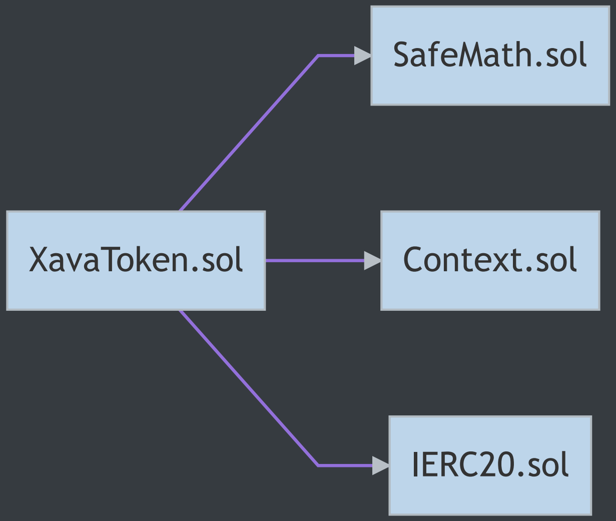
Executive Summary

We were tasked with auditing the Xava project's source code and in particular their token and staking implementation. We were able to find major issues in the farming implementation which we strongly recommend are fixed as soon as possible to ensure the project comes to a deploy-able state. Additionally, we strongly urge the Xava team to expand the test cases of their contracts as the issues highlighted by the report would have been identified by extensive test cases.

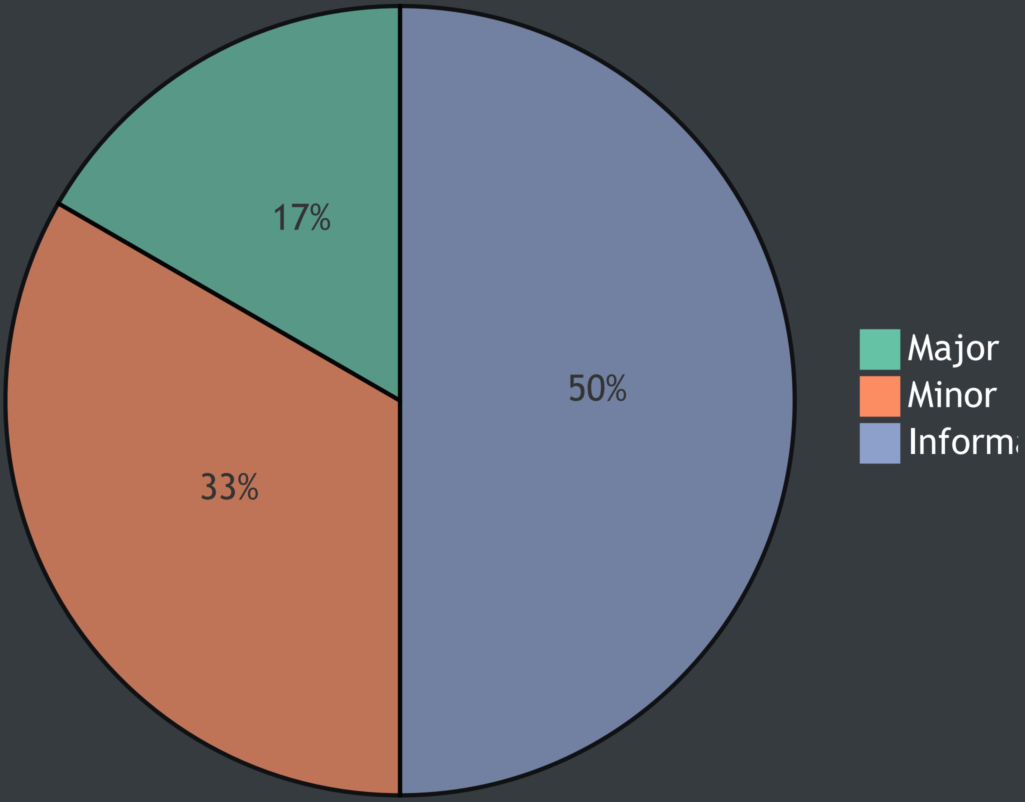
Files In Scope

ID	Contract	Location
XTN	XavaToken.sol	contracts/XavaToken.sol
FXA	FarmingXava.sol	contracts/farming/FarmingXava.sol

File Dependency Graph



Finding Summary



Manual Review Findings

ID	Title	Type	Severity	Resolved
XTN-01	Variable Mutability Optimization	Gas Optimization	🟢 Informational	⊙
FXA-01	Incorrect Accounting of `totalDeposits` Variable	Logical Issue	🟠 Major	⊙
FXA-02	Ill-Advised Renewal Pattern	Logical Issue	🔵 Minor	⊙
FXA-03	Improper Execution Path	Logical Issue	🔵 Minor	⊙
FXA-04	Redundant Value Initialization	Coding Style	🟢 Informational	⊙
FXA-05	Variable Mutability Optimization	Gas Optimization	🟢 Informational	⊙

XTN-01: Variable Mutability Optimization

Type	Severity	Location
Gas Optimization	● Informational	XavaToken.sol L18-L20, L22-L27

Description:

The `_name`, `_symbol` and `_decimals` variables are initialized at the `constructor` of the contract, however, they are meant to only represent the Xava token.

Recommendation:

We advise them to be set as `constant` and assigned to the actual values they will ultimately be set to as the implementation of the Xava token should not be generic.

FXA-01: Incorrect Accounting of `totalDeposits` Variable

Type	Severity	Location
Logical Issue	● Major	FarmingXava.sol L178-L192, L213-L220

Description:

The `totalDeposits` of a pool are not properly updated during a `deposit` or an `emergencyWithdrawal`, which will permanently lock tokens in the contract and will also cause improper rewards to be paid out.

Recommendation:

We advise the `totalDeposits` variable to be properly updated during a `deposit` and an `emergencyWithdraw`.

FXA-02: Ill-Advised Renewal Pattern

Type	Severity	Location
Logical Issue	● Minor	FarmingXava.sol L82-L86

Description:

The `fund` function is meant to provide more funds for the Xava farm by extending its `endTimestamp` by the amount deposited. However, it prevents deposits if the `endTimestamp` has been passed meaning that the pool cannot be renewed if improperly set in the `constructor` or if the initial transactions do not go through due to a gas spike.

Recommendation:

We strongly recommend a logic path to be introduced in the `fund` function that adjusts `startTimestamp` and `endTimestamp` according to the amount deposited if `endTimestamp` has been surpassed to ensure that the farm will be maintainable at any given point in time.

FXA-03: Improper Execution Path

Type	Severity	Location
Logical Issue	● Minor	FarmingXava.sol L128-L133

Description:

The `if` block within `pending` will execute even beyond a particular pool's duration because the evaluation of time in the conditional is done with `block.timestamp` which, if the pool has ended, will always be greater-than `pool.lastRewardTimestamp` thus performing multiple redundant zero value assignments within.

Recommendation:

We advise the `lastTimestamp` declaration to be set outside the `if` block and utilized in the `if` conditional to ensure proper execution of its inner statements.

FXA-04: Redundant Value Initialization

Type	Severity	Location
Coding Style	● Informational	FarmingXava.sol L49, L58

Description:

The linked contract level variables are explicitly set to `0` on declaration which is their default value.

Recommendation:

We advise the explicit assignments to be omitted from the codebase.

FXA-05: Variable Mutability Optimization

Type	Severity	Location
Gas Optimization	● Informational	FarmingXava.sol L47, L51, L61, L70-L72

Description:

The linked variables are assigned to only once during the contract's `constructor`.

Recommendation:

We advise them to be set as `immutable` greatly optimizing their gas cost.

Appendix

Finding Categories

Gas Optimization

Gas Optimization findings refer to exhibits that do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

Logical Issue

Logical Issue findings are exhibits that detail a fault in the logic of the linked code, such as an incorrect notion on how `block.timestamp` works.

Coding Style

Coding Style findings usually do not affect the generated byte-code and comment on how to make the codebase more legible and as a result easily maintainable.