



Security Assessment

Xave Finance 2nd audit

May 6th, 2022

Table of Contents

Summary

Overview

[Project Summary](#)

[Audit Summary](#)

[Vulnerability Summary](#)

[Audit Scope](#)

Findings

[GLOBAL-01 : Usage of Decentralized Oracle](#)

[HDA-01 : SafeMath Not Used](#)

[HDA-02 : Usage of SafeCast](#)

[HDA-03 : Declaration Naming Convention](#)

[HDA-04 : Function Should Be Declared External](#)

[HDA-05 : Variables That Could Be Declared as Immutable](#)

Appendix

Disclaimer

About

Summary

This report has been prepared for Xave Finance to discover issues and vulnerabilities in the source code of the Xave Finance project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Addendum: This report has been updated to match the updated protocol name. The original audit remains dependent on the original code.

Overview

Project Summary

| | |
|--------------|---|
| Project Name | Xave Finance 2nd audit |
| Platform | Polygon |
| Language | Solidity |
| Codebase | https://github.com/xave-finance/lending-market-price-oracles/tree/df1d2e4de4490f7b4ed33e29d7fc439a195ae3f0 |
| Commit | df1d2e4de4490f7b4ed33e29d7fc439a195ae3f0 |

Audit Summary

| | |
|-------------------|--------------------------------|
| Delivery Date | Jun 30, 2022 UTC |
| Audit Methodology | Static Analysis, Manual Review |

Vulnerability Summary

| Vulnerability Level | Total | Pending | Declined | Acknowledged | Mitigated | Partially Resolved | Resolved |
|------------------------------|-------|---------|----------|--------------|-----------|--------------------|----------|
| ● Critical | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ● Major | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ● Medium | 2 | 0 | 0 | 0 | 0 | 0 | 2 |
| ● Minor | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ● Optimization | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ● Informational | 4 | 0 | 0 | 4 | 0 | 0 | 0 |
| ● Discussion | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Audit Scope

| ID | File | SHA256 Checksum |
|-----|----------------------------------|--|
| HLP | contracts/HLPPriceFeedOracle.sol | 790fb3e8dfcde23aa5bfd9e8d0c0eb2ce14f9371641afaeaedad1138854e3294 |
| PFB | contracts/PriceFeed.sol | a07dd30671ad489e8379c59bb76876f5123b4dcf42c0ef3ed2f0f40563230662 |

Findings



| | |
|---------------|------------|
| Critical | 0 (0.00%) |
| Major | 0 (0.00%) |
| Medium | 2 (33.33%) |
| Minor | 0 (0.00%) |
| Informational | 4 (66.67%) |
| Discussion | 0 (0.00%) |

| ID | Title | Category | Severity | Status |
|---------------------------|---|------------------|-----------------|----------------|
| GLOBAL-01 | Usage Of Decentralized Oracle | Coding Style | ● Informational | ⓘ Acknowledged |
| HDA-01 | SafeMath Not Used | Coding Style | ● Medium | ✓ Resolved |
| HDA-02 | Usage Of SafeCast | Coding Style | ● Medium | ✓ Resolved |
| HDA-03 | Declaration Naming Convention | Coding Style | ● Informational | ⓘ Acknowledged |
| HDA-04 | Function Should Be Declared External | Gas Optimization | ● Informational | ⓘ Acknowledged |
| HDA-05 | Variables That Could Be Declared As Immutable | Gas Optimization | ● Informational | ⓘ Acknowledged |

GLOBAL-01 | Usage Of Decentralized Oracle

| Category | Severity | Location | Status |
|--------------|-----------------|----------|----------------|
| Coding Style | ● Informational | | ⓘ Acknowledged |

Description

Currently, the data is being queried from chainlink as per the testing files. Chainlink provides a reliable manipulation resistant price feed for blockchain data. It should be noted that if the data is queried from another oracle such as a liquidity pool the data may no longer be manipulation resistant.

Recommendation

We recommend updating the documentation to include where data originates from. So that users can understand the different possible risks associated to this oracle feed.

HDA-01 | SafeMath Not Used

| Category | Severity | Location | Status |
|--------------|----------|---|------------|
| Coding Style | ● Medium | contracts/HLPPriceFeedOracle.sol (base): 43, 52, 54; contracts/PriceFeed.sol (base): 36, 45, 47 | 👍 Resolved |

Description

Usage of `SafeMath` library prevents the arithmetic attacks in solidity versions released before Version 0.8.0. The listed lines of code have arithmetic statements that are vulnerable to underflow and overflow attacks.

Recommendation

It is recommended to update the statements using `SafeMath` functions to prevent arithmetic attacks or update the Solidity compiler to 0.8.0 or later. The compiler of the later versions has a built-in feature to prevent underflows and overflows.

Alleviation

The smart contracts have been updated to the Solidity Version 0.8.4. This compiler version has underflow and overflow errors built in.

HDA-02 | Usage Of SafeCast

| Category | Severity | Location | Status |
|--------------|----------|---|------------|
| Coding Style | ● Medium | contracts/HLPPriceFeedOracle.sol (base); contracts/PriceFeed.sol (base) | ✓ Resolved |

Description

There are type conversions from `uint256` to `int256` that could cause overflows.

Recommendation

To safely downcast, it is recommended to use the SafeCast Library. This library allows for safe type cast from `uint256` to `int256` which reverts if there is an overflow. More information can be found here:

[SafeCast](#).

Alleviation

Type casting has been updated to use the SafeCast Library.

HDA-03 | Declaration Naming Convention

| Category | Severity | Location | Status |
|--------------|-----------------|---|----------------|
| Coding Style | ● Informational | contracts/HLPPriceFeedOracle.sol (base): 7, 12; contracts/PriceFeed.sol (base): 6 | ⓘ Acknowledged |

Description

One or more declarations do not conform to the [Solidity style guide](#) with regards to its naming convention.

Particularly:

- `camelCase`: Should be applied to function names, argument names, local and state variable names, modifiers
- `UPPER_CASE`: Should be applied to `constant` variables
- `CapWords`: Should be applied to contract names, struct names, event names and enums

File: contracts/HLPPriceFeedOracle.sol (Line 12)

```
contract hlpPriceFeedOracle {
```

- Contract `hlpPriceFeedOracle` is not in `CapWords`.

File: contracts/HLPPriceFeedOracle.sol (Line 7)

```
interface hlpContract {
```

- Contract `hlpContract` is not in `CapWords`.

File: contracts/PriceFeed.sol (Line 6)

```
contract fxPriceFeed {
```

- Contract `fxPriceFeed` is not in `CapWords`.

Recommendation

We recommend adjusting those variable and function names in conformance to Solidity's naming convention. It is not a violation. It can be considered as a suggestion to improve readability and consistency in the code.

HDA-04 | Function Should Be Declared External

| Category | Severity | Location | Status |
|------------------|-----------------|---|----------------|
| Gas Optimization | ● Informational | contracts/HLPPriceFeedOracle.sol (base): 33; contracts/PriceFeed.sol (base): 25 | ⓘ Acknowledged |

Description

The functions which are never called internally within the contract should have external visibility for gas optimization.

File: contracts/HLPPriceFeedOracle.sol (Line 33, Contract `hlpPriceFeedOracle`)

```
function latestAnswer() public view returns (int256) {
```

File: contracts/PriceFeed.sol (Line 25, Contract `fxPriceFeed`)

```
function latestAnswer() public view returns (int256) {
```

Recommendation

We advise to change the visibility of the aforementioned functions to `external`.

HDA-05 | Variables That Could Be Declared As Immutable

| Category | Severity | Location | Status |
|------------------|-----------------|---|----------------|
| Gas Optimization | ● Informational | contracts/HLPPriceFeedOracle.sol (base): 20; contracts/PriceFeed.sol (base): 12 | ① Acknowledged |

Description

The linked variables assigned in the constructor can be declared as `immutable`. Immutable state variables can be assigned during contract creation but will remain constant throughout the lifetime of a deployed contract. A big advantage of immutable variables is that reading them is significantly cheaper than reading from regular state variables since they will not be stored in storage.

Recommendation

We recommend declaring these variables as immutable. Please note that the `immutable` keyword only works in Solidity version `v0.6.5` and up.

Appendix

Finding Categories

Gas Optimization

Gas Optimization findings 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.

Coding Style

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

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you (“Customer” or the “Company”) in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK’s prior written consent in each instance.

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is 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 assessment. This report does 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.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing 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.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED “AS IS” AND

“AS AVAILABLE” AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED “AS IS” AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

About

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.

