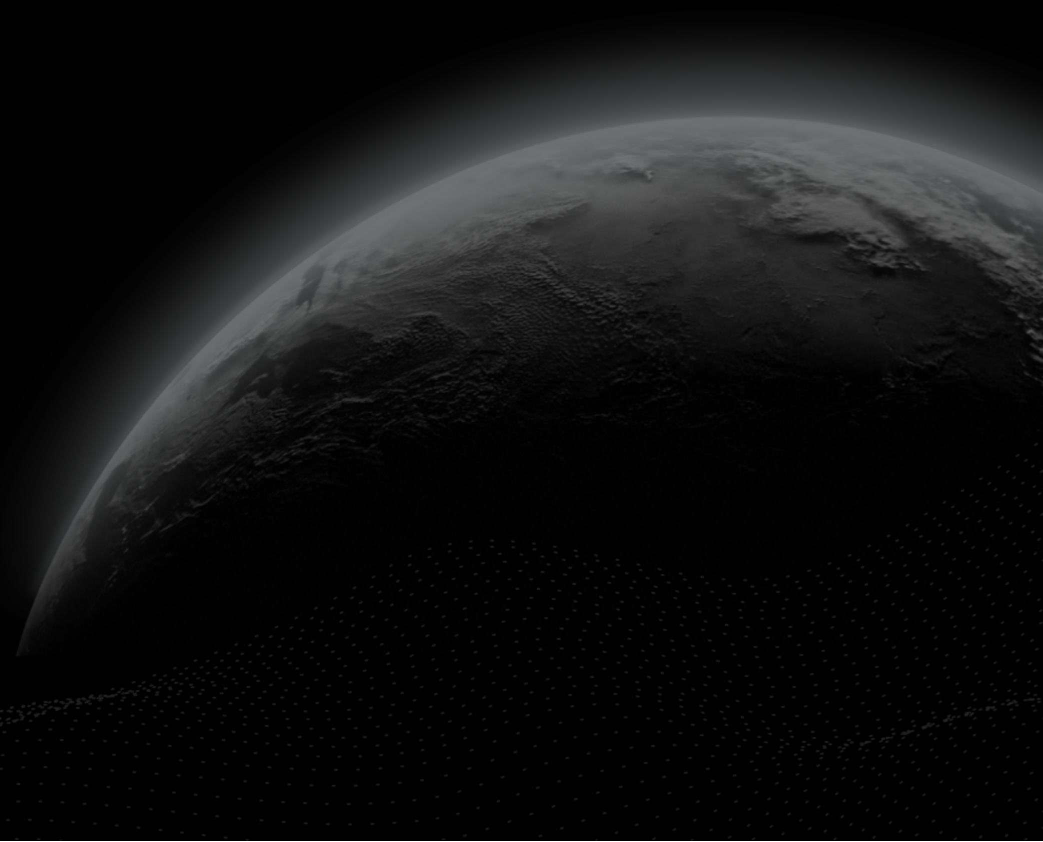
CERTIK

Security Assessment

Colend - Audit

CertiK Assessed on Apr 23rd, 2024



CertiK Assessed on Apr 23rd, 2024

Colend - Audit

The security assessment was prepared by CertiK, the leader in Web3.0 security.

Executive Summary

TYPES	ECOSYSTEM	METHODS
DeFi	EVM Compatible	Manual Review, Static Analysis

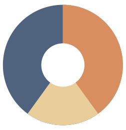
LANGUAGE	TIMELINE	KEY COMPONENTS
Solidity	Delivered on 04/23/2024	dad86c82e625236135125fc410352d43fdc6fcc7

CODEBASE

https://github.com/Colend-Protocol/aave-v3-core/tree/feat/pyth-oracle

View All in Codebase Page

Vulnerability Summary



5	0	0	1	4	0
Total Findings	Resolved	Mitigated	Partially Resolved	Acknowledged	Declined

🟥 0	Critical		Critical risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
🟧 2	Major	1 Partially Resolved, 1 Acknowledged	Major risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
🟨 0	Medium		Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
🟨 1	Minor	1 Acknowledged	Minor risks can be any of the above, but on a smaller scale. They generally do not compromise the overall integrity of the project, but they may be less efficient than other solutions.
🟦 2	Informational	2 Acknowledged	Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

TABLE OF CONTENTS | COLEND - AUDIT

CODEBASE | COLEND - AUDIT

Repository

https://github.com/Colend-Protocol/aave-v3-core/tree/feat/pyth-oracle

AUDIT SCOPE | COLEND - AUDIT

8 files audited ● 3 files with Acknowledged findings ○ 5 files without findings

ID	Repo	File	SHA256 Checksum
● ECP	Colend-Protocol/aave-v3-core	contracts/protocol/libraries/helpers/Errors.sol	f04e7936e4a32b86dead5e7b973934c66d44364b58c2ff0a5a2037563a8e198b
● PAP	Colend-Protocol/aave-v3-core	contracts/protocol/configuration/PoolAddressesProvider.sol	e11ccebb0e91715e6b62d2e8f972a2e84f41be8554e764248b690bc40d69bf53
● AOC	Colend-Protocol/aave-v3-core	contracts/misc/AaveOracle.sol	457668dcfe844e2f51aa022e9c300d4dbca5423be1fbb6207c8a25eead5d10b1
○ IPA	Colend-Protocol/aave-v3-core	contracts/interfaces/IPoolAddressesProvider.sol	639f0ac55ef43aadf47a23a31d9bdb6d454c0fee179069d1804a361163aac51f
○ IAO	Colend-Protocol/aave-v3-core	contracts/interfaces/IAaveOracle.sol	59a9814ce5c41c0d0472f0662e66a3d4939fe32a8ac63f26fda53fb1afd127a9
○ IPC	Colend-Protocol/aave-v3-core	contracts/dependencies/pyth/IPyth.sol	8588ed8a28374b474390c8182549e4973b46b2259e50a7e9a148384c5eeb2420
○ IPE	Colend-Protocol/aave-v3-core	contracts/dependencies/pyth/IPythEvents.sol	06580966cfb3cdf1357a960450d7549c801f649c1df5f2ddfa33ff7386685e0e
○ PSC	Colend-Protocol/aave-v3-core	contracts/dependencies/pyth/PythStructs.sol	46b5413a78f67cd4ef527ebad1fd913b572a67ac9201567dd67a84db4c544e60

APPROACH & METHODS | COLEND - AUDIT

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

- Testing the smart contracts against both common and uncommon attack vectors;
- 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.

REVIEW NOTES | COLEND - AUDIT

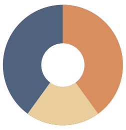
Differential Audit

The main branch of the project, `https://github.com/Colend-Protocol/aave-v3-core/tree/feat/pyth-oracle` , is forked from the AAVE project: `aave-v3-core v1.19.2` at `https://github.com/aave/aave-v3-core/releases/tag/v1.19.2` .

The code modification made by the client integrates the Pyth Price Oracle and ensures compatibility with the Core blockchain network.

The scope of the audit encompasses the differences between the Colend and AAVE (aave-v3-core v1.19.2) code, and these are detailed in the report.

FINDINGS | COLEND - AUDIT



5	0	2	0	1	2
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for Colend - Audit. Through this audit, we have uncovered 5 issues ranging from different severity levels. Utilizing the techniques of Manual Review & Static Analysis to complement rigorous manual code reviews, we discovered the following findings:

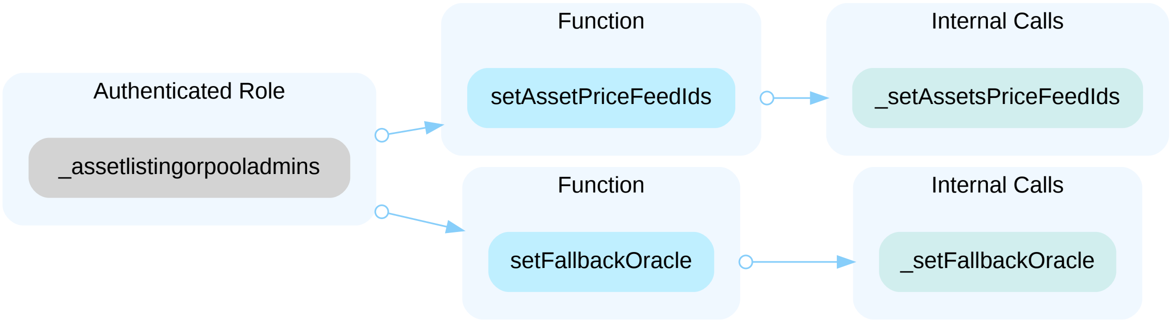
ID	Title	Category	Severity	Status
CPB-01	**Centralization Related Risks**	**Centralization**	**Major**	● **Acknowledged**
RLC-01	Protocol Can Be Attacked Due To The Potential Rounding Issue If Total Supply Of A Market Is Empty	Logical Issue	Major	● Partially Resolved
AOC-01	Third-Party Dependency Usage	Design Issue	Minor	● Acknowledged
AOC-02	Unuse Confidence Intervals	Design Issue	Informational	● Acknowledged
ECP-01	Unused Constant Variable	Coding Style	Informational	● Acknowledged

CPB-01 | CENTRALIZATION RELATED RISKS

Category	Severity	Location	Status
Centralization	● **Major**	**contracts/misc/AaveOracle.sol: 65, 73; contracts/protocol/configuration/PoolAddressesProvider.sol: 48, 58, 65, 81, 93, 105, 117, 129, 141, 153, 165**	● **Acknowledged**

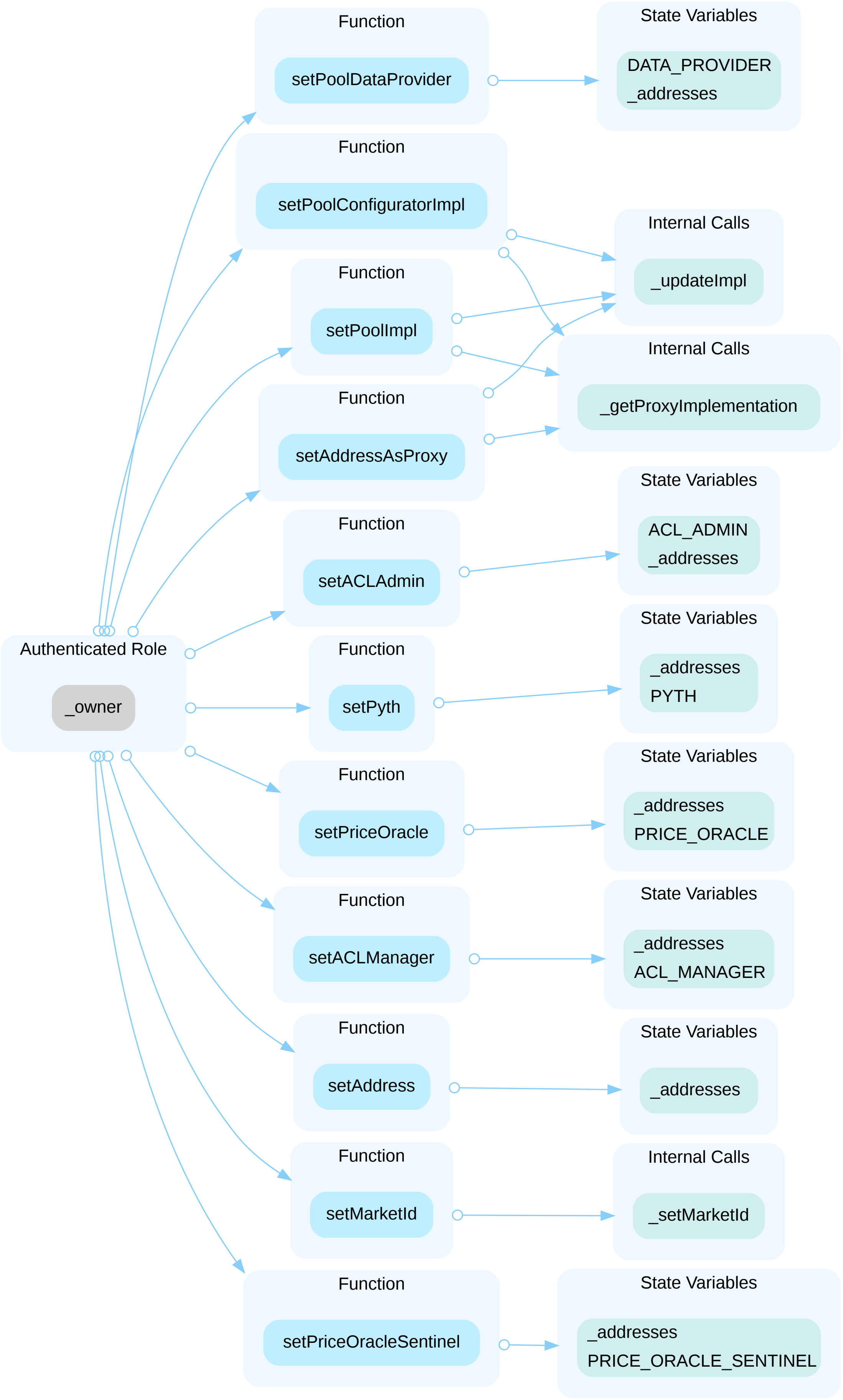
Description

In the contract `AaveOracle` the role `_assetlistingorpooladmins` has authority over the functions shown in the diagram below. Any compromise to the `_assetlistingorpooladmins` account may allow the hacker to take advantage of this authority and set asset price feed IDs and set fallback oracle.



In the contract `PoolAddressesProvider` the role `_owner` has authority over the functions shown in the diagram below. Any compromise to the `_owner` account may allow the hacker to take advantage of this authority and

- set market ID
- set address
- set the address as a proxy
- set pool implement
- set pool configuration implement
- set price oracle address
- set Pyth oracle address
- set ACL manager
- set ACL admin
- set price Oracle sentinel
- set pool data provider


Authenticated Role
_owner
Function
setPoolDataProvider
State Variables
DATA_PROVIDER
_addresses
Function
setPoolConfiguratorImpl
Internal Calls
_updateImpl
Function
setPoolImpl
Internal Calls
_getProxyImplementation
Function
setAddressAsProxy
Function
setACLAdmin
State Variables
ACL_ADMIN
_addresses
Function
setPyth
State Variables
_addresses
PYTH
Function
setPriceOracle
State Variables
_addresses
PRICE_ORACLE
Function
setACLManager
State Variables
_addresses
ACL_MANAGER
Function
setAddress
State Variables
_addresses
Function
setMarketId
Internal Calls
_setMarketId
Function
setPriceOracleSentinel
State Variables
_addresses
PRICE_ORACLE_SENTINEL

Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., multisignature wallets. Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign (⅔, ⅗) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles.
  OR
- Remove the risky functionality.

Alleviation

[Colend Team, 04/22/2024]: The team acknowledged the issue and adopted the multisign solution to ensure the private key management process at the current stage. The `PoolAddressProvider` contract has transferred the ownership to a Gnosis Safe contract with 2/3 signers in the sensitive function signing process.

- Grant Role transaction hash for Gnosis Safe: https://scan.coredao.org/tx/0x9e738a493f615d48c459aa23d4f23e88f097a8da7beb92a8e8c2a326f6595808#overview

- The three multisign addresses:

    1. EOA:0x0B4F1B818144fca1f56191F1d50Dc4584503AD5f
    2. EOA:0xA9cB975efd9E4De4BFCb57018B0fA205A953cc2C
    3. EOA:0x07C41661eC79E3354134A0cB35Ae47C139B2670e

[CertiK, 04/22/2024]: While this strategy has indeed reduced the risk, it's crucial to note that it has not completely eliminated it. CertiK strongly encourages the project team to periodically revisit the private key security management of all the above-listed addresses.

RLC-01 PROTOCOL CAN BE ATTACKED DUE TO THE POTENTIAL ROUNDING ISSUE IF TOTAL SUPPLY OF A MARKET IS EMPTY

Category	Severity	Location	Status
Logical Issue	● Major	contracts/protocol/libraries/logic/ReserveLogic.sol: 118~130	● Partially Resolved

Description

The value of the `liquidityIndex` is calculated proportionally to the result of protocol revenue divided by the total liquidity amount. When the total liquidity is extremely small (e.g., value equal to 1), this can cause the `liquidityIndex` to become an extremely large value. This situation can occur when a pool has 1 wei deposited token and generates a large amount of revenue from the income of a flash loan operation.

```
uint256 result = (amount.wadToRay().rayDiv(totalLiquidity.wadToRay()) +
WadRayMath.RAY).rayMul(
      reserve.liquidityIndex
    );
```

The `liquidityIndex` variable is used in the collateral withdrawal process when calculating the share amount to burn. Due to the inflated `liquidityIndex` value, it enables the rounding error originating from the "WadRayMath" math library to be used in the burning of the share token and withdrawal calculations from the pool. When the "totalShare" of the pool is 2, and when the user is required to return a value larger than 1 (e.g., 1.5) share, the value will round down to 1, which allows the user to withdraw 1.5 shares worth of collateral and only burn 1 share. By repeating deposits to create a total share of 2 and withdrawing more than 1 share of collateral tokens, the attacker can withdraw more collateral than they are supposed to and drain the pool.

```
// In the "withdraw" function
IAToken(reserveCache.aTokenAddress).burn(
      msg.sender,
      params.to,
      amountToWithdraw,
      reserveCache.nextLiquidityIndex
    );

// In the burn function:
uint256 amountScaled = amount.rayDiv(index);
```

This specific attack vector is only possible when a token pool is empty or has an extremely small amount of liquidity.

Reference: https://medium.com/@RadiantCapital/post-mortem-report-radiant-capital-aea46cb985ae

Recommendation

When adding a new market to the protocol, the auditor recommends that, within the same transaction that enables the market, the project team sets the LTV (Loan to Value) value to zero, deposits a small initial amount of funds into the pool, and burns the share tokens. Afterward, the team can change the LTV back to a non-zero normal value. This procedure can help mitigate the attack vector against a newly deployed empty market.

Alleviation

[Colend Team, 04/16/2024]:

Supply Assets sent to 0x...dead

- https://scan.coredao.org/tx/0xa6b3091248b5202dbc0c8ffed346f0eb3ac48a855bcca384ab8549633effa0aa
- https://scan.coredao.org/tx/0xb1eb6db1f90dd7358061e3c2fce25c9a807606ddf55218cf8a924b71df5baf67
- https://scan.coredao.org/tx/0x3ab076baaa34f91195fc0fbd7c9ac50410d69bf706fb3c835311f9ce7b2e63ae
- https://scan.coredao.org/tx/0xc85b65867ca93f609196cae4dee8a6ab5a7856ef8c0c28b366a654fb60cdc407

The Client supplies the `WCORE` , `COREBTC` , `USDT` , and `USDC` assets to the protocol and burns the share tokens.

[CertiK Team, 04/16/2024]:

While the above pools are protected, please ensure that new pools also implement the same security measures in the future.

AOC-01 | THIRD-PARTY DEPENDENCY USAGE

Category	Severity	Location	Status
Design Issue	● Minor	contracts/misc/AaveOracle.sol: 109	● Acknowledged

Description

The contract is serving as the underlying entity to interact with one or more third-party protocols. The scope of the audit treats third-party entities as black boxes and assumes their functional correctness. However, in the real world, third parties can be compromised and this may lead to lost or stolen assets. In addition, upgrades of third parties can possibly create severe impacts, such as increasing fees of third parties, migrating to new LP pools, etc.

```
109        PythStructs.Price memory priceStruct = IPyth(ADDRESSES_PROVIDER.getPyth()
).getPrice(
110        priceFeedId
111      );
```

- The contract `AaveOracle` interacts with the third-party contract with `IPyth` interface via `ADDRESSES_PROVIDER.getPyth()` .

Recommendation

The auditors understood that the business logic requires interaction with third parties. It is recommended for the team to constantly monitor the statuses of third parties to mitigate the side effects when unexpected activities are observed.

Alleviation

[Colend Team, 04/12/2024]: Issue acknowledged. I won't make any changes for the current version. We currently have monitor solution in place to constantly compare the on-chain Pyth price vs off-chain aggregate price.

AOC-02 | UNUSE CONFIDENCE INTERVALS

Category	Severity	Location	Status
Design Issue	● Informational	contracts/misc/AaveOracle.sol: 113	● Acknowledged

Description

The function `getPrice()` can get the latest price and confidence interval for the requested price feed ID.

At every point in time, Pyth publishes both a price and a confidence interval for each product.

In a Pyth feed, each publisher specifies an interval (p_i-c_i, p_i+c_i) in the form of their price and confidence submission. This interval is intended to achieve 95% coverage, i.e. the publisher expresses the belief that this interval contains the “true” price with 95% probability. The resulting aggregate interval (μ-σ, μ+σ), where μ represents the aggregate price and σ represents the aggregate confidence, is a good estimate of a range in which the true price lies.

1. It can use a discounted price in the direction favorable to it. For example, a lending protocol valuing a user’s collateral can use the lower valuation price μ-σ. When valuing an outstanding loan position consisting of tokens a user has borrowed from the protocol, it can use the higher end of the interval by using the price μ+σ. This allows the protocol to be conservative with regard to its own health and safety when making valuations.
2. It can decide that there is too much uncertainty when σ/μ exceeds some threshold and choose to pause any new activity that depends on the price of this asset.

Refer: https://docs.pyth.network/price-feeds/best-practices

Recommendation

We would like to confirm with the client if the current implementation aligns with the original project design.

Alleviation

[Colend Team, 04/12/2024]: Issue acknowledged. I won't make any changes for the current version.

interesting point that we haven't looked into for now. We are going to analyze this and see if we won't utilize this.

ECP-01 | UNUSED CONSTANT VARIABLE

Category	Severity	Location	Status
Coding Style	● Informational	contracts/protocol/libraries/helpers/Errors.sol: 100	● Acknowledged

Description

The constant variable `UNREGISTERED_ASSET_FOR_PRICE_FEED` is declared but never used in the project.

Recommendation

We recommend removing unused constant variable if it is not going to be used.

Alleviation

The client acknowledged this finding.

APPENDIX | COLEND - AUDIT

Finding Categories

Categories	Description
Coding Style	Coding Style findings may not affect code behavior, but indicate areas where coding practices can be improved to make the code more understandable and maintainable.
Logical Issue	Logical Issue findings indicate general implementation issues related to the program logic.
Centralization	Centralization findings detail the design choices of designating privileged roles or other centralized controls over the code.
Design Issue	Design Issue findings indicate general issues at the design level beyond program logic that are not covered by other finding categories.

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 | CERTIK

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.

CertiK | **Securing** the **Web3** World

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.