CERTIK

Code Security Assessment

Aboard

Jan 14th, 2022

Table of Contents

Summary

This report has been prepared for Aboard to discover issues and vulnerabilities in the source code of the Aboard 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.

Overview

Project Summary

Project Name	Aboard
Description	Aboard
Platform	BSC, Arbitrum
Language	Solidity
Codebase	https://github.com/AboardGroup/Aboard-Contracts
Commit	24bcbf9d40a5239707c296877e80a4100abdcf83

Audit Summary

Delivery Date	Jan 14, 2022
Audit Methodology	Static Analysis, Manual Review

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
Critical	0	0	0	0	0	0
Major	5	0	0	2	1	2
Medium	0	0	0	0	0	0
Minor	1	0	0	0	0	1
Informational	3	0	0	1	0	2
Discussion	0	0	0	0	0	0

Audit Scope

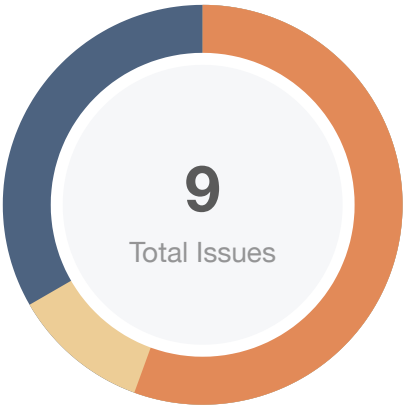
ID	File	SHA256 Checksum
AVI	external/chainlink/AggregatorV3Interface.sol	ba60f727e9c27b67e0f3460a762735d65e7a9f4cac98117962e1adf1fc92a4af
AAC	protocol/lib/Adminable.sol	0519d135baa739826aa3688f1bdfc9d9c5a47bfea014a92adc4e1e4eabfbfdc0
BMA	protocol/lib/BaseMath.sol	10eaae11bd5f2377c22a1f168c28dcb0903386aa55ec936d52d1e6e960a47b86
MAC	protocol/lib/Math.sol	72a5ba674b4ac46fbc552e9890f594a6575c87ebbd26078ed705e34e9ed6fe3e
RGA	protocol/lib/ReentrancyGuard.sol	fd96851f3cf7b3decffec0497a24b350b61f8409c50a511bf5cfdcb61bb82ae2
SCA	protocol/lib/SafeCast.sol	7f257a6ac80a4ef3022ad11c0c372a1ceaff718057d084a63796e704db0c3884
SMA	protocol/lib/SignedMath.sol	15606f9e2ec6b63fcd1b719d6e986eab7971ba6be73cab2184dcec576295394c
SAC	protocol/lib/Storage.sol	239b28358b99d1837d1c41a381284e34c6004280b54b561cb7e3826815b7b15e
PAA	protocol/v1/impl/P1Admin.sol	fc8828b5527343047e82ca9c5b7539870305f14eaf48064e208f24a1fe09c7c1
PGA	protocol/v1/impl/P1Getters.sol	a3ea0b12639659631f099739b60ae5ea3fcc5fd442005eb291b728d269a1ff8e
PMA	protocol/v1/impl/P1Margin.sol	a347c49d61f683a5da5be03966ced385ee67f5d8c7474fdc24eeb57066f86681
POA	protocol/v1/impl/P1Operator.sol	dc179fc74a0e0d9bbda74be4445058e7aa68a2527432362626ce6f89100c6a7a
PSA	protocol/v1/impl/P1Settlement.sol	aa410fb9612284ddbd3225c86f2548db20b198d9de1cb9170bbe4fac489536da
PSC	protocol/v1/impl/P1Storage.sol	c80640de2a2d330221e97f47e341b57bb47d1ca47dc5ea83ddf746bfaa833da0

ID	File	SHA256 Checksum
PTA	protocol/v1/impl/P1Trade.sol	410afa16ec12801008d50d7a10001f9beb59d9d0c722994324ad5ff517150565
IPF	protocol/v1/intf/I_P1Funder.sol	34c2aef33d21509586c1c10e5da4f6cf32ae50182e4a3056e30e882f2481fd3f
IPO	protocol/v1/intf/I_P1Oracle.sol	c0740785719c499d35d41e7cfa00185ab959de87dc776ac4bcee522bf52860a1
IPT	protocol/v1/intf/I_P1Trader.sol	e0394aa7b671d806c7e56d1290ed3f057f9236e12a1bf72fdc7d5203f9fb385a
IPV	protocol/v1/intf/I_PerpetualV1.sol	33626b91749cf4a4d7b03e6df91601a01f6598f9986f1b2057a4e51261437b9c
PTC	protocol/v1/lib/P1Types.sol	ade23c1424242f214938126b39da438ebebd69daad5e6e96aac3453b8f921fde
PCO	protocol/v1/oracles/P1ChainlinkOracle.sol	a940f028357d0dcd32a8d4b3cca24c52b57440863d292f0ea8bf57ae5e740fb3
PLA	protocol/v1/traders/P1Liquidation.sol	94c5aa859ed046494906b7a1f7359b920784717d73374c29641dd92c6194112c
POC	protocol/v1/traders/P1Orders.sol	99ff9cc80f3ca2d2d4c0cf8ee8337c736cbbf5b93c036492f8fde4add473725d
PTK	protocol/v1/traders/P1TraderConstants.sol	eeaf28e21898099cd0ca571fc658c8d9ac273b060b50de4e9905a129369d6754
PVA	protocol/v1/PerpetualV1.sol	9d2b1ad55cd5cc7e476d7734dc276ef9bfe8bf2f83489567552c5ebf349f39bc
PPA	protocol/PerpetualProxy.sol	ea5036a7ccf9b00e3026e78e0f4bccbbb7aeaf7cde8da880ce1506d7e87119e0

Financial Models

Financial models of blockchain protocols need to be resilient to attacks. It needs to pass simulations and verifications to guarantee the security of the overall protocol. Financial models are not in the scope of the audit.

Findings



Critical	**0** (0.00%)
Major	**5** (55.56%)
Medium	**0** (0.00%)
Minor	**1** (11.11%)
Informational	**3** (33.33%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
GLOBAL-01	Incorrect naming convention utilization	Coding Style	Informational	Acknowledged
PCO-01	Centralization related risks	**Centralization / Privilege**	**Major**	Resolved
PCO-02	Third party price oracle	Volatile Code	Minor	Resolved
PCO-03	Missing emit events	Coding Style	Informational	Resolved
PMA-01	Centralization related risks	**Centralization / Privilege**	**Major**	Resolved
PPA-01	Centralization Related Risks	**Coding Style, Centralization / Privilege**	**Major**	Partially Resolved
PTA-01	Centralization related risks	**Centralization / Privilege**	**Major**	Acknowledged
PTA-02	Unknown implementation of order handler	Logical Issue	Major	Acknowledged
PTA-03	Function visibility optimization	Gas Optimization	Informational	Resolved

GLOBAL-01 | Incorrect naming convention utilization

Category	Severity	Location	Status
Coding Style	Informational	Global	Acknowledged

Description

Variable P1Storage.*TOKEN_SYMBOL* (protocol/v1/impl/P1Storage.sol#38) is not in mixedCase

Variable P1Storage.*BALANCES* (protocol/v1/impl/P1Storage.sol#40) is not in mixedCase

Variable P1Storage.*GLOBAL_OPERATORS* (protocol/v1/impl/P1Storage.sol#42) is not in mixedCase

Variable P1Storage.*LOCAL_OPERATORS* (protocol/v1/impl/P1Storage.sol#43) is not in mixedCase

Variable P1Storage.*TOKEN* (protocol/v1/impl/P1Storage.sol#45) is not in mixedCase

Variable P1Storage.*ORACLE* (protocol/v1/impl/P1Storage.sol#46) is not in mixedCase

Parameter P1Admin.setToken(address).token_address (protocol/v1/impl/P1Admin.sol#83) is not in mixedCase

Parameter P1Admin.setTokenSymbolInitial(string[]).symbol_array (protocol/v1/impl/P1Admin.sol#126) is not in mixedCase

Function P1Settlement.toBytes32_deposit_withdraw(address,SignedMath.Int) (protocol/v1/impl/P1Settlement.sol#188-203) is not in mixedCase

Function P1Settlement.toBytes32_fee(uint256,bool) (protocol/v1/impl/P1Settlement.sol#208-221) is not in mixedCase

Parameter P1Settlement.toBytes32_fee(uint256,bool).is_neg_fee (protocol/v1/impl/P1Settlement.sol#210) is not in mixedCase

Function P1Settlement.toBytes32_funding(SignedMath.Int) (protocol/v1/impl/P1Settlement.sol#226-238) is not in mixedCase

Contract I_P1Funder (protocol/v1/intf/I_P1Funder.sol#30-49) is not in CapWords

Function P1Margin.withdraw_apply(address,address,uint256) (protocol/v1/impl/P1Margin.sol#106-119) is not in mixedCase

Variable P1Orders.*PERPETUAL_V1* (protocol/v1/traders/P1Orders.sol#62) is not in mixedCase

Variable ReentrancyGuard.*STATUS* (protocol/lib/ReentrancyGuard.sol#34) is not in mixedCase

Contract I_P1Trader (protocol/v1/intf/I_P1Trader.sol#32-51) is not in CapWords

Variable P1ChainlinkOracle.*ORACLE* (protocol/v1/oracles/P1ChainlinkOracle.sol#45) is not in mixedCase

Variable P1ChainlinkOracle.*READER* (protocol/v1/oracles/P1ChainlinkOracle.sol#48) is not in mixedCase

Variable P1ChainlinkOracle.*ADJUSTMENT* (protocol/v1/oracles/P1ChainlinkOracle.sol#51) is not in mixedCase

Variable P1ChainlinkOracle.*MAPPING* (protocol/v1/oracles/P1ChainlinkOracle.sol#54) is not in mixedCase

Function P1Trade.margin_position(address,address,P1Types.TradeResult,string) (protocol/v1/impl/P1Trade.sol#182-247) is not in mixedCase

Contract I_P1Oracle (protocol/v1/intf/I_P1Oracle.sol#30-45) is not in CapWords

Contract I_PerpetualV1 (protocol/v1/intf/I_PerpetualV1.sol#32-253) is not in CapWords

Function I_PerpetualV1.withdraw_apply(address,address,uint256) (protocol/v1/intf/I_PerpetualV1.sol#77-82) is not in mixedCase

Variable P1Liquidation.*PERPETUAL_V1* (protocol/v1/traders/P1Liquidation.sol#66) is not in mixedCase

Alleviation

The team acknowledged this issue and they will leave it as it is for now.

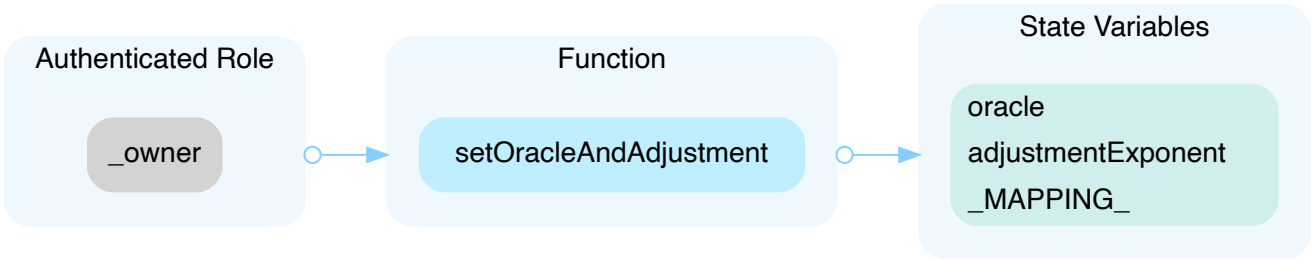
PCO-01 | Centralization related risks

Category	Severity	Location	Status
Centralization / Privilege	Major	protocol/v1/oracles/P1ChainlinkOracle.sol: 116~128	Resolved

Description

In the contract `P1ChainlinkOracle`, the role `_owner` has the authority over the functions shown in the diagram below.

Any compromise to the privileged account which has access to `_owner` may allow the hacker to take advantage of this.



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., multi-signature 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

The team heeded our advice and removed the file `P1ChainlinkOracle.sol` in commit `https://github.com/AboardGroup/sc_audit_for_certik/tree/6a4e974fe66ba1fb49cb63eba0ccd9beca66f682`.

PCO-02 | Third party price oracle

Category	Severity	Location	Status
Volatile Code	Minor	protocol/v1/oracles/P1ChainlinkOracle.sol	Resolved

Description

The contract is serving as the underlying entity to interact with third party `Chainlink` protocol. The scope of the audit treats 3rd party entities as black boxes and assumes their functional correctness. However, in the real world, 3rd parties can be compromised and this may lead to lost or stolen assets. In addition, upgrades of 3rd parties can possibly create severe impacts.

Recommendation

We understand that the business logic of oracle requires interaction with `Chainlink`, etc. We encourage the team to constantly monitor the statuses of 3rd parties to mitigate the side effects when unexpected activities are observed.

Alleviation

The team heeded our advice and removed the file `P1ChainlinkOracle.sol` in commit `https://github.com/AboardGroup/sc_audit_for_certik/tree/6a4e974fe66ba1fb49cb63eba0ccd9beca66f682`.

PCO-03 | Missing emit events

Category	Severity	Location	Status
Coding Style	Informational	protocol/v1/oracles/P1ChainlinkOracle.sol: 116~128	Resolved

Description

There should always be events emitted in the sensitive functions that are controlled by centralization roles.

Recommendation

It is recommended emitting events for the sensitive functions that are controlled by centralization roles.

Alleviation

The team heeded our advice and removed the file `P1ChainlinkOracle.sol` in commit `https://github.com/AboardGroup/sc_audit_for_certik/tree/6a4e974fe66ba1fb49cb63eba0ccd9beca66f682`.

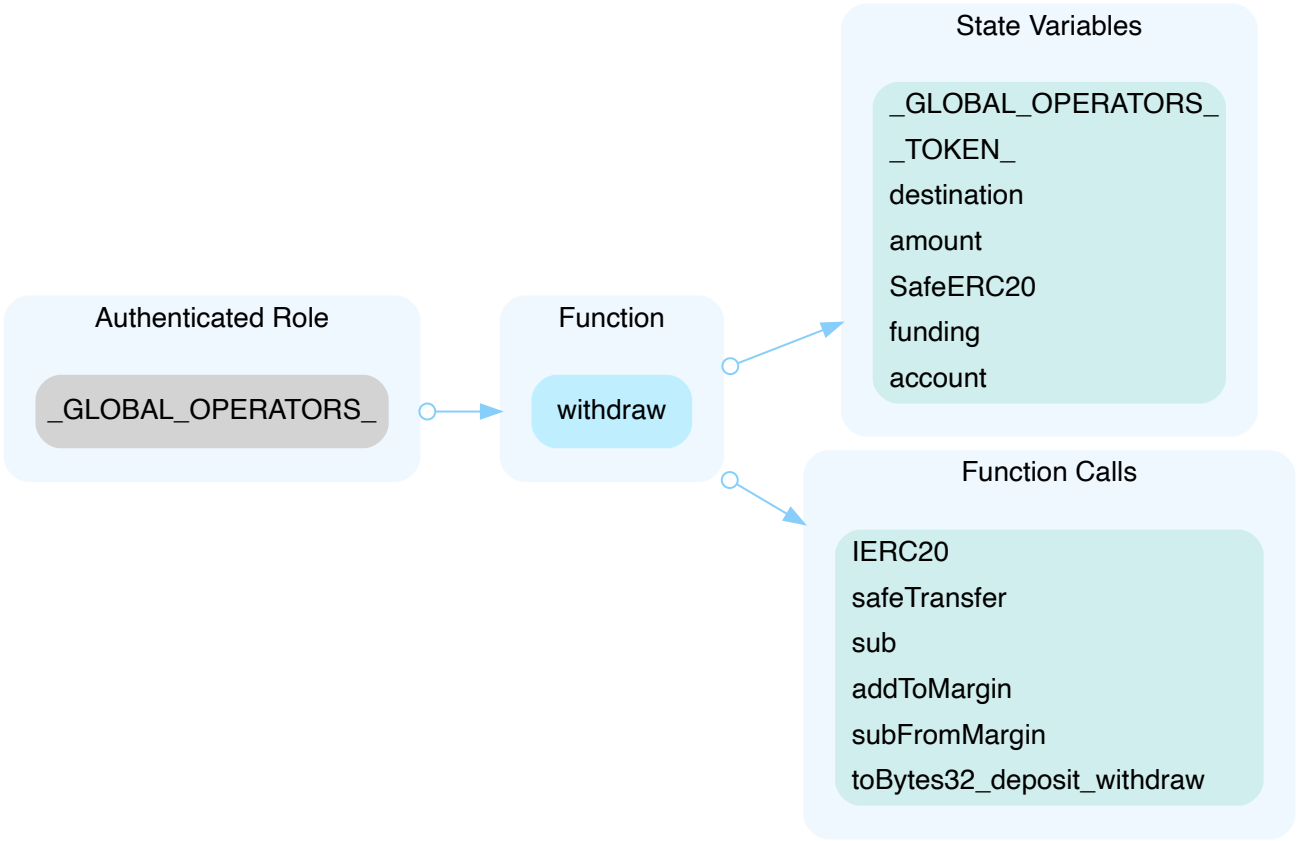
PMA-01 | Centralization related risks

Category	Severity	Location	Status
Centralization / Privilege	Major	protocol/v1/impl/P1Margin.sol: 129~164	Resolved

Description

In the contract, `P1Margin`, the role, `_GLOBAL_OPERATORS_`, has the authority over the functions shown in the diagram below.

Any compromise to the privileged account which has access to `_GLOBAL_OPERATORS_` may allow the hacker to take advantage of this.



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., multi-signature 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

The team heeded our advice and changed the authority to the user in the commit `https://github.com/AboardGroup/sc_audit_for_certik/tree/6a4e974fe66ba1fb49cb63eba0ccd9beca66`

`f682`.

PPA-01 | Centralization Related Risks

Category	Severity	Location	Status
Coding Style, Centralization / Privilege	**Major**	protocol/PerpetualProxy.sol	Partially Resolved

Description

In the contract `PerpetualProxy`, the role `admin` has the authority over the following function:

- `upgradeTo()/upgradeToAndCall()`: change the implementation of `PerpetualProxy` with any contracts,
- `changeAdmin()`: change the `admin` of the contract,

Any compromise to the `admin` account may allow the hacker to take advantage of this and users' assets may suffer loss.

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., multi-signature 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

The team acknowledged the issue and adopted the timelock solution to delay-sensitive operations at the current stage, and adopted the multi-sig solution to ensure the private key management process at the current stage. The `PerpetualProxy` contract has transferred the ownership to a multi-sig plus timelock contract with 2/3 signers and 48 hours of current delay in the sensitive function signing process. However, the delay can be changed without a minimum limit.

For Arbitrum:

- Grant role transaction hash for multi-sig contract:
  0xeafbc98a1a768b12b67f160a2e45b835b5b413585fcfacd65b25bd7fa2ec351f
- Set the implementation transaction hash:
  0x70ff935ad0ad4fcfe480910cce556881377a0a5bc3053c77318fcf1b7b3167af
- The 3 signers' addresses:
  1. EOA: 0x54ad8c33e6c7df7a6143d962a44be65ae1d4ab36
  2. EOA: 0x799ce6da8e5a251f5ab057518f57bb7edd25613c
  3. EOA: 0x48e6818d594a02f8b1e801c824eaa4ff23eb8b00

For BSC:

- Grant role transaction hash for multi-sig contract:
  0x5cc50322a301e2c3fa63bf4d96a839c1f8c86adbf8438005a5c9f21674e1da08
- Set the implementation transaction hash:
  0xcaffeb979a69e6f6c416e95f5d1642992fee81d61d0727d1a82fa35c6ba31447
- The 3 signers' addresses:
  1. EOA: 0x54ad8c33e6c7df7a6143d962a44be65ae1d4ab36
  2. EOA: 0x799ce6da8e5a251f5ab057518f57bb7edd25613c
  3. EOA: 0x48e6818d594a02f8b1e801c824eaa4ff23eb8b00

PTA-01 | Centralization related risks

Category	Severity	Location	Status
Centralization / Privilege	**Major**	protocol/v1/impl/P1Trade.sol: 85~136	Acknowledged

Description

In the contract, `P1Trade`, the role, `_GLOBAL_OPERATORS_`, has the authority over the functions shown below.

- `trade()`: submit trades between any number of accounts, which call the function `trade()` in the contracts `P1Liquidation` and `P1Orders`

Any compromise to the privileged account which has access to `_GLOBAL_OPERATORS_` may allow the hacker to take advantage of this.

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., multi-signature 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

The team acknowledged this issue and they stated:

"They will add a monitor on the event logs to verify every trading data."

PTA-02 | Unknown implementation of order handler

Category	Severity	Location	Status
Logical Issue	Major	protocol/v1/impl/P1Trade.sol: 85	Acknowledged

Description

All the data for the orders and liquidation are passed in via parameters from off-chain order handler.

This is not in the scope of the audit. The scope of the audit treats the off-chain order handler as black boxes and assumes their functional correctness.

Recommendation

We understand that the business logic of the function `trade()` requires interaction with off-chain order handler. We encourage the team to constantly monitor the statuses of off-chain order handler to mitigate the side effects when unexpected activities are observed.

Alleviation

The team acknowledged this issue and they stated:

"The off-chain module is an order matching system. And all its information is transferred on-chain by `_GATEWAY_`. The information is then stored and emitted event logs on-chain in P1Trade.sol, P1Orders.sol, P1Liquidation.sol and P1Deleveraging.sol. The Users can get all the information on client or on blockchain scanner(e.g. arbiscan.io and bscscan.com). And they will develop a module for users to verify their asserts by on-chain event logs easily."

PTA-03 | Function visibility optimization

Category	Severity	Location	Status
Gas Optimization	Informational	protocol/v1/impl/P1Trade.sol: 89	Resolved

Description

The function `trade()` is declared as `public`, contain array function arguments, and are not invoked in any of the contracts contained within the project's scope. The functions that are never called internally within the contract should have external visibility.

Recommendation

We advise that the functions' visibility specifiers are set to `external` and the array-based arguments change their data location from `memory` to `calldata`, optimizing the gas cost of the function.

Alleviation

The team headed our advice and resolved this issue in commit `https://github.com/AboardGroup/sc_audit_for_certik/tree/74a34e8336c2352312421080199f015380f32298`.

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

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.

Logical Issue

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

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

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.