CERTIK

Security Assessment

O3 Swap v2 (Interchange)

Aug 2nd, 2022

Table of Contents

Summary

Overview

- Project Summary
- Audit Summary
- Vulnerability Summary
- Audit Scope

Findings

- BCO-01 : Incompatibility With Deflationary Tokens
- BCO-02 : Incorrect initialization of the `isInitialized`
- BCO-03 : Potential Front-Running Risk
- BCO-04 : Lack of reasonable boundary
- BCO-05 : Discussion For Function `bridgeOutAndWithdraw()` and `bridgeOut()`
- BCO-06 : Discussion For Function `require()`
- COL-01 : Third Party Dependencies
- CPC-01 : Unchecked Low-level Call
- CPC-02 : Discussion For `withdrawAmount`
- CPO-01 : Bridge Fee logic is reimplemented in `CallProxy`
- CRO-01 : Approving zero amount is redundant
- CRO-02 : Usage of `transfer()` for sending Ether
- DVL-01 : Daily limit is zeroed only for one token
- DVL-02 : `1 days` can be used instead of 86400 magic number
- OLB-01 : Centralization Related Risks
- OLB-02 : Lack of Zero Address Validation
- OLU-01 : Redundant SafeMath Usage
- PTO-01 : Discussion For Function `burn()`
- WCO-02 : Discuss For Contract `Wrapper`

Optimizations

- OLB-03 : Function Visibility Optimization
- WCO-01 : Arguments Should Be `calldata`

Appendix

Disclaimer

About

Summary

This report has been prepared for O3 Swap v2 (Interchange) to discover issues and vulnerabilities in the source code of the O3 Swap v2 (Interchange) 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	O3 Swap v2 (Interchange)
Platform	Ethereum
Language	Solidity
Codebase	https://github.com/O3Labs/o3swap-v2-core
Commit	v1:71b3f8acb1af9ce26a6658aed49911ec1815d3aa v2:eba05b9dca567b00953afdf62bd5121f05453f57 v3:7ca432914ae07c2501ca0d8d604059e379368d2e

Audit Summary

Delivery Date	Aug 02, 2022 UTC
Audit Methodology	Static Analysis, Manual Review

Vulnerability Summary

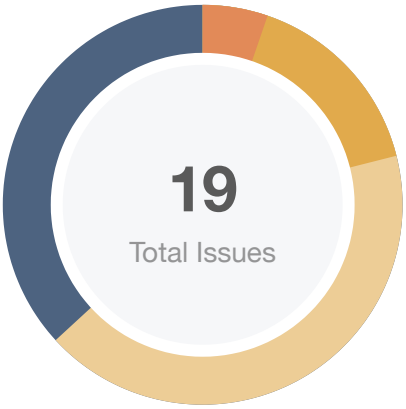
Vulnerability Level	Total	Pending	Declined	Acknowledged	Mitigated	Partially Resolved	Resolved
Critical	0	0	0	0	0	0	0
Major	1	0	0	1	0	0	0
Medium	3	0	0	0	0	0	3
Minor	8	0	0	3	0	0	5
Informational	7	0	0	6	0	0	1
Discussion	0	0	0	0	0	0	0

Audit Scope

ID	Repo	Commit	File	SHA256 Checksum
IOO	O3Labs/o3swap-v2-core	71b3f8a	assets/interfaces/IO3.sol	92f5cf705ce59cb82fae65d9eda61fd193bb06d5f104d961c74ea8582d09083e
IPT	O3Labs/o3swap-v2-core	71b3f8a	assets/interfaces/IPToken.sol	8c8ff13ae2ff5ea3f50289a11c460cde634b71a8971a0e2c89695ff45b662794
IWE	O3Labs/o3swap-v2-core	71b3f8a	assets/interfaces/IWETH.sol	8f71b1317f7ecc065d60719ba01019a507dea750da0184b15e7a09111c29b3f4
PTO	O3Labs/o3swap-v2-core	71b3f8a	assets/PToken.sol	9d933603f4c90f97674de04bfdc003ee7bd9cb722071bceb6e878922891f93b8
IBC	O3Labs/o3swap-v2-core	71b3f8a	crossChain/interfaces/IBridge.sol	45b83f008e2a66ca3deec370c5989d46fe2f9a98559ccf0d03ea4d340a79f867
ICP	O3Labs/o3swap-v2-core	71b3f8a	crossChain/interfaces/ICallProxy.sol	356e35899dc5d6677b1c7df48aac116fa90f3f8a8e3ccf33346dac6a59a9206e
IDV	O3Labs/o3swap-v2-core	71b3f8a	crossChain/interfaces/IDailyVolumeLimiter.sol	d6e82a52699498bd30cd57161af3bfd2122c96105ee7b319805a13ad9ab5cb4c
IEC	O3Labs/o3swap-v2-core	71b3f8a	crossChain/interfaces/IEthCrossChainManager.sol	ea7bfe8601ad1d00ae50df205ab5d746a68aaefa5631bee1fc5ff669f2d946bd
IEM	O3Labs/o3swap-v2-core	71b3f8a	crossChain/interfaces/IEthCrossChainManagerProxy.sol	a31e4a2cdc75fd48c45e15a9f76df49e7c9c6449a678344785a273c220ef655c
IWC	O3Labs/o3swap-v2-core	71b3f8a	crossChain/interfaces/IWrapper.sol	92740261b67648304b147550be8f035e82b187be9c6bc6a349126dbcd2925a06
BCO	O3Labs/o3swap-v2-core	71b3f8a	crossChain/Bridge.sol	2693cd59346366126d936141f6844e88b69f467ed9ab0ef5f9df821d4c26206c
CPC	O3Labs/o3swap-v2-core	71b3f8a	crossChain/CallProxy.sol	1cee36b6e8e31721f4370b2363b60ddec1149a55654d0ca4dbc8b1c751540fc5
DVL	O3Labs/o3swap-v2-core	71b3f8a	crossChain/DailyVolumeLimiter.sol	09d3fa12c8e6d451715102ed3d239e5217f5a460b34440562a7439d1412e65a4
UCO	O3Labs/o3swap-v2-core	71b3f8a	crossChain/Utils.sol	5a05511b03b28037d002bf038cffe2ad470c6e8ff70ecaffb08060b2a4937842

ID	Repo	Commit	File	SHA256 Checksum
WCO	O3Labs/o3swap-v2-core	71b3f8a	crossChain/Wrapper.sol	9ddc8189ef7a1a3a200f3de35cdf8a2208898d687938e43b61c769d9897b9284

Findings



Critical	0 (0.00%)
Major	1 (5.26%)
Medium	3 (15.79%)
Minor	8 (42.11%)
Informational	7 (36.84%)
Discussion	0 (0.00%)

ID	Title	Category	Severity	Status
BCO-01	Incompatibility With Deflationary Tokens	Logical Issue	Medium	Resolved
BCO-02	Incorrect Initialization Of The `isInitialized`	Logical Issue	Medium	Resolved
BCO-03	Potential Front-Running Risk	Volatile Code	Minor	Resolved
BCO-04	Lack Of Reasonable Boundary	Volatile Code	Minor	Resolved
BCO-05	Discussion For Function `bridgeOutAndWithdraw()` And `bridgeOut()`	Logical Issue	Informational	Acknowledged
BCO-06	Discussion For Function `require()`	Logical Issue	Informational	Acknowledged
COL-01	Third Party Dependencies	Volatile Code	Minor	Acknowledged
CPC-01	Unchecked Low-level Call	Control Flow	Minor	Acknowledged
CPC-02	Discussion For `withdrawAmount`	Logical Issue	Informational	Acknowledged
CPO-01	Bridge Fee Logic Is Reimplemented In `CallProxy`	Inconsistency	Minor	Acknowledged
CRO-01	Approving Zero Amount Is Redundant	Inconsistency	Minor	Resolved
CRO-02	Usage Of `transfer()` For Sending Ether	Volatile Code	Minor	Resolved
DVL-01	Daily Limit Is Zeroed Only For One Token	Volatile Code	Medium	Resolved

ID	Title	Category	Severity	Status
DVL-02	`1 days` Can Be Used Instead Of 86400 Magic Number	Magic Numbers	Informational	Resolved
OLB-01	Centralization Related Risks	**Centralization / Privilege**	**Major**	Acknowledged
OLB-02	Lack Of Zero Address Validation	Volatile Code	Minor	Resolved
OLU-01	Redundant SafeMath Usage	Language Specific	Informational	Acknowledged
PTO-01	Discussion For Function `burn()`	Logical Issue	Informational	Acknowledged
WCO-02	Discuss For Contract `Wrapper`	Logical Issue	Informational	Acknowledged

BCO-01 | Incompatibility With Deflationary Tokens

Category	Severity	Location	Status
Logical Issue	Medium	crossChain/Bridge.sol (v1): 146	Resolved

Description

When transferring standard ERC20 deflationary tokens, the input amount may not be equal to the received amount due to the charged transaction fee.

Recommendation

We advise the client to regulate the set of pool tokens supported and add necessary mitigation mechanisms to keep track of accurate balances if there is a need to support deflationary tokens.

Alleviation

`[Client]`: Source code of new tokens will be reviewed before adding to the system, deflationary tokens are not very suitable for cross-chain so these tokens will not be accepted by default. Also, we've updated the code to get the accuracy balances(commit).

BCO-02 | Incorrect Initialization Of The `isInitialized`

Category	Severity	Location	Status
Logical Issue	Medium	crossChain/Bridge.sol (v1): 31	Resolved

Description

If the variable `isInitialized` is set to true, then the function `initialize` will fail when invoked.

Recommendation

We advise the client to recheck the function.

Alleviation

The client removed the code and resolved this issue.

BCO-03 | Potential Front-Running Risk

Category	Severity	Location	Status
Volatile Code	Minor	crossChain/Bridge.sol (v1): 63	Resolved

Description

Malicious hackers may observe the pending transaction which will execute the `initialize` function, and launch a similar transaction but with the hacker's address of `owner` and gain the ownership of the contract.

Recommendation

We advise the client to design functionality to only allow a specific user to execute the `initialize` function.

Alleviation

The client removed the code and resolved this issue.

BCO-04 | Lack Of Reasonable Boundary

Category	Severity	Location	Status
Volatile Code	Minor	crossChain/Bridge.sol (v1): 67	Resolved

Description

The variable `bridgeFeeRate` does not have reasonable boundaries, so they can be given arbitrary values after deploying.

Recommendation

We recommend adding reasonable upper and lower boundaries to all the configuration variables

Alleviation

`[Client]`: Max bridge fee rate constant was added(commit).

BCO-05 | Discussion For Function `bridgeOutAndWithdraw()` And `bridgeOut()`

Category	Severity	Location	Status
Logical Issue	Informational	crossChain/Bridge.sol (v1): 157, 177	Acknowledged

Description

The functions `bridgeOutAndWithdraw()` and `bridgeOut()` can be called by anyone. While both functions perform a similar role, the function `bridgeOut()` charges a fee. Therefore users may prefer to call `bridgeOutAndWithdraw()`.

Recommendation

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

Alleviation

`[Client]`: Yes this aligns the design, bridgeOut() charges a bridge fee, bridgeOutAndWithdraw() charges a withdraw fee, additionally, an extra bridge fee will be charged if withdraw ptokens to non-entrance chains(the flag '_depositWithdrawEnabled' is only enabled on its entrance chain).

BCO-06 | Discussion For Function `require()`

Category	Severity	Location	Status
Logical Issue	Informational	crossChain/Bridge.sol (v1): 245	Acknowledged

Description

Line 245 checks that the contract address of the source `fromChainId` is `fromContractAddr`, but according to the function `bindBridge()` setting, it is all mapped from the `toChainId` to the `targetBridge`.

Recommendation

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

Alleviation

`[Client]`: Yes this aligns the design. Bridge bindings are all two-way bindings.

COL-01 | Third Party Dependencies

Category	Severity	Location	Status
Volatile Code	Minor	crossChain/Bridge.sol (v1): 225, 282; crossChain/CallProxy.sol (v1): 69, 135; crossChain/Wrapper.sol (v1): 124, 161	Acknowledged

Description

The contract is serving as the underlying entity to interact with third-party protocols. The scope of the audit would treat those 3rd party entities as black boxes and assume their functional correctness. However, in the real world, 3rd parties may be compromised and lead to assets being lost or stolen.

Recommendation

We understand that the business logic of the contract requires interaction with the aforementioned protocols. We encourage the team to constantly monitor the status of 3rd parties to mitigate side effects when unexpected activities are observed.

Alleviation

`[Client]`: We will constantly monitor the status of 3rd parties, we also cooperate with many security teams to construct a better eco-system.

CPC-01 | Unchecked Low-level Call

Category	Severity	Location	Status
Control Flow	Minor	crossChain/CallProxy.sol (v1): 119	Acknowledged

Description

The low-level `call` function returns the status of the call as first variable in the returned tuple. The status of the `call` is not asserted to be `true`, which would treat the low-level call as a success even when it reverted.

```
119                 callee.call(data);
```

Recommendation

We advise to check the return value of a low-level call or log it.

Alleviation

`[Client]`: This aligns the design. The external call is open to use if the flag is enabled. The call can be used to execute the destination chain aggregation swap or other customized logic. If the call failed, all the ptokens will be sent to the receiver.

CPC-02 | Discussion For `withdrawAmount`

Category	Severity	Location	Status
Logical Issue	Informational	crossChain/CallProxy.sol (v1): 104	Acknowledged

Description

If the variable `withdrawAmount` is equal to 0, then the user is exempt from the charge and gets all `PToken`.

Recommendation

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

Alleviation

`[Client]`: This aligns the design. No matter if the withdraw amount matches the total amount, this type of transactions is more expensive than the direct ptoken cross-chain transactions (withdraw transactions use more gas to execute logic). So it's user's loss if they modified the amount manually and unable to get all the underlying token.

CPO-01 | Bridge Fee Logic Is Reimplemented In `CallProxy`

Category	Severity	Location	Status
Inconsistency	Minor	crossChain/CallProxy.sol (v2): 107~110	Acknowledged

Description

`CallProxy` contract uses the knowledge about the Bridge Fee and calculates it itself. In particular, the `FEE_DENOMINATOR` constant is redeclared. This breaks the encapsulation principle and reduces the code maintainability.

Recommendation

We recommend introducing the `getBridgeFee()` method to the `Bridge` contract and use it in `CallProxy`.

Alleviation

`[Client]`: Since this does not affect the logic and the actual bridge fee is zero in mainnet, we will apply the update in future versions.

CRO-01 | Approving Zero Amount Is Redundant

Category	Severity	Location	Status
Inconsistency	Minor	crossChain/CallProxy.sol (v2): 119~120, 137~138; crossChain/Wrapper.sol (v2): 97~98, 125~126, 133~134, 162~163, 170~171, 202~203, 234~235, 260~261	Resolved

Description

```
121             IERC20(tokenFrom).safeApprove(poolAddress, 0);
122             IERC20(tokenFrom).safeApprove(poolAddress, dx);
```

Approving 0 amount before the desired is useless. No Approval Race is possible during contract execution.

Recommendation

We recommend removing of redundant statements.

CRO-02 | Usage Of `transfer()` For Sending Ether

Category	Severity	Location	Status
Volatile Code	Minor	crossChain/CallProxy.sol (v2): 89~90; crossChain/Wrapper.sol (v2): 72~73	Resolved

Description

After EIP-1884 was included in the Istanbul hard fork, it is not recommended to use `.transfer()` or `.send()` for transferring ether as these functions have a hard-coded value for gas costs making them obsolete as they are forwarding a fixed amount of gas, specifically `2300`. This can cause issues in case the linked statements are meant to be able to transfer funds to other contracts instead of EOAs.

Recommendation

We advise that the linked `.transfer()` and `.send()` calls are substituted with the utilization of the `sendValue()` function from the `Address.sol` implementation of OpenZeppelin either by directly importing the library or copying the linked code.

DVL-01 | Daily Limit Is Zeroed Only For One Token

Category	Severity	Location	Status
Volatile Code	Medium	crossChain/DailyVolumeLimiter.sol (v2): 36~38	Resolved

Description

```
35          if (_day != day) {
36              dailyVolumeMap[_token] = 0;
37              day = _day;
38          }
```

When the new `day` comes, the `dailyVolumeMap[_token]` is zeroed. However, for all other tokens the `dailyVolumeMap` is left intact. The malicious actor can trigger the first daily transaction with one specific token. All other token limits will be depleted and the bridge unusable.

Recommendation

We recommend using of

```
      mapping(address => uint256) public dayMap;          // dayMap[token] is the day when
dailyVolumeMap[token] was last updated
```

instead of `uint256 day`.

DVL-02 | `1 days` Can Be Used Instead Of 86400 Magic Number

Category	Severity	Location	Status
Magic Numbers	Informational	crossChain/DailyVolumeLimiter.sol (v2): 34~35, 51~52	Resolved

Description

Time Units can be used instead of magic numbers.

Recommendation

We recommend replacing the 86400 constant with `1 days` to improve the code readability.

OLB-01 | Centralization Related Risks

Category	Severity	Location	Status
Centralization / Privilege	Major	assets/PToken.sol (v1): 59, 64, 104, 109, 114, 119, 124; crossChain/Bridge.sol (v1): 67, 73, 78, 83, 92, 98, 104, 113, 123, 127, 237; crossChain/CallProxy.sol (v1): 36, 41, 46, 51, 56; crossChain/DailyVolumeLimiter.sol (v1): 28, 71, 76, 81, 90; crossChain/Wrapper.sol (v1): 44, 48, 52, 57, 62, 67, 71	Acknowledged

Description

To bridge the gap in trust between the administrators need to express a sincere attitude regarding the consideration of the administrator team's anonymity.

The `Owner` of `PToken` has the responsibility to notify users about the following capabilities:

- set `_authorizedCaller` through `setAuthorizedCaller()` and `removeAuthorizedCaller()`
- set `_depositWithdrawEnabled` through `enableDepositWithdraw()` and `disableDepositWithdraw()`
- set `_withdrawFeeRate` and `_feeCollector` through `setWithdrawFee()`

The `AuthorizedCaller` of `PToken` has the responsibility to notify users about the following capabilities:

- mint uncapped tokens for anyone through `mint()`
- burn tokens through `burn()`

The `Owner` of `Wrapper` has the responsibility to notify users about the following capabilities:

- pause the contract through `pause()`
- unpause the contract through `unpause()`
- set `bridge` through `setBridgeContract()`
- set `feeCollector` through `setFeeCollector()`
- set `wethAddress` through `setWETHAddress()`
- withdraw contract's tokens through `rescueFund()`

The `FeeCollector` of `Wrapper` has the responsibility to notify users about the following capabilities:

- withdraw contract's ETH/BNB through `extractFee()`

The `Owner` of `DailyVolumeLimiter` has the responsibility to notify users about the following capabilities:

- set `_authorizedCallers` through `setAuthorizedCaller()`

- set `volumeLimitMap` through `setLimit()`
- set `volumeLimitMap` through `setLimitBatch()`
- set `dailyVolumeMap` through `updateVolume()`

The `AuthorizedCaller` of `DailyVolumeLimiter` has the responsibility to notify users about the following capabilities:

- accumulate `dailyVolumeMap` through `accumulate()`

The `Owner` of `Bridge` has the responsibility to notify users about the following capabilities:

- set `bridgeFeeRate` and `bridgeFeeCollector` through `setBridgeFee()`
- set `callProxy` through `setCallProxy()`
- set `managerProxyContract` through `setManagerProxy()`
- set `volumeLimiter` through `setVolumeLimiter()`
- set `bridgeHashMap` through `bindBridge()`
- set `assetHashMap` through `bindAssetHash()`
- set `bridgeHashMap` through `bindBridgeBatch()`
- set `assetHashMap` through `bindAssetHashBatch()`
- pause the contract through `pause()`
- unpause the contract through `unpause()`

The `ManagerContract` of `Bridge` has the responsibility to notify users about the following capabilities:

- get tokens for `toAddress` through `bridgeIn()`

The `Owner` of `CallProxy` has the responsibility to notify users about the following capabilities:

- set `wethAddress` through `setWETH()`
- set `bridgeAddress` through `setBridge()`
- set `externalCallEnabled` through `enableExternalCall()` and `disableExternalCall()`

The `Bridge` of `CallProxy` has the responsibility to notify users about the following capabilities:

- swap tokens for `receiver` through `proxyCall()`

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

`[Client]`: We are doing more researches to upgrade the whole protocol to a more decentralized state. We are considering set up a DAO to execute admin functions.

OLB-02 | Lack Of Zero Address Validation

Category	Severity	Location	Status
Volatile Code	Minor	assets/PToken.sol (v1): 47; crossChain/Bridge.sol (v1): 69, 74, 79; crossChain/CallProxy.sol (v1): 37, 42; crossChain/Wrapper.sol (v1): 53, 58, 63	Resolved

Description

The given input is missing the check for the non-zero address.

Recommendation

We advise the client to add the check for the passed-in values to prevent unexpected errors.

Alleviation

The client heeded our advice and resolved this issue in commit.

OLU-01 | Redundant SafeMath Usage

Category	Severity	Location	Status
Language Specific	Informational	assets/PToken.sol (v2): 12~13; crossChain/Bridge.sol (v2): 21~22; crossChain/CallProxy.sol (v2): 17~18	Acknowledged

Description

Solidity version >=0.8.0 includes checked arithmetic operations and underflow/overflow by default, making SafeMath redundant.

Recommendation

We recommend removing the SafeMath library and use standard arithmetic operators to reduce code complexity.

Alleviation

`[Client]`: Since this does not affect the logic, we'll update it in future versions.

PTO-01 | Discussion For Function `burn()`

Category	Severity	Location	Status
Logical Issue	Informational	assets/PToken.sol (v1): 59, 64	Acknowledged

Description

The function `burn()` is called to burn `PToken`, yet the corresponding `_tokenUnderlying` is not taken out. Meanwhile, the contract has no function to take out the `_tokenUnderlying`, so the `_tokenUnderlying` in the contract and the number of PToken would not be the same. The function `mint()` has the same issue.

Recommendation

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

Alleviation

`[Client]`: This aligns the design. The burn() and mint() functions is logically splitted into different chains. Every ptoken has its own ‘entrance chain’ to store all the underlying token(all liquidity of this ptoken stored in one chain so users can safely whtidraw at any time). The bridge contract burns ptokens from source chain and mint the same amount of ptokens on destionation chain to complete the cross-chain process. Underlying tokens will be taken out only when users withdraw their liquidity from pool.

WCO-02 | Discuss For Contract `Wrapper`

Category	Severity	Location	Status
Logical Issue	Informational	crossChain/Wrapper.sol (v1): 76	Acknowledged

Description

The contract `Wrapper` wraps the contract `Bridge`. Functions of the `Wrapper` which have modifier `paybale` can accept ETH, such as the `bridgeOut()`, but the `Bridge` does not accept ETH, so the user can call functions in the `Bridge` directly without calling functions in the `Wrapper`.

Recommendation

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

Alleviation

`[Client]`: Network fees are required to complete to cross-chain process. The protocol receives network fees through wrapper contract, so if users call the bridge contract directly, the transaction can succeed on the source chain but the message will not be passed to the destination chain due to a lack of enough fees.

Optimizations

ID	Title	Category	Severity	Status
OLB-03	Function Visibility Optimization	Gas Optimization	Optimization	Resolved
WCO-01	Arguments Should Be `calldata`	Gas Optimization	Optimization	Resolved

OLB-03 | Function Visibility Optimization

Category	Severity	Location	Status
Gas Optimization	Optimization	assets/PToken.sol (v1): 55; crossChain/Bridge.sol (v1): 63, 67, 73, 78, 92, 98, 104, 113, 157, 237; crossChain/CallProxy.sol (v1): 36, 41, 46, 51, 142, 172, 196, 213, 228, 240; crossChain/Wrapper.sol (v1): 52, 57, 62, 67, 71, 76, 103, 139, 176, 208, 240	Resolved

Description

`public` functions that are never called by the contract could be declared `external`. When the inputs are arrays, `external` functions are more efficient than `public` functions.

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 client heeded our advice and resolved this issue in commit.

WCO-01 | Arguments Should Be `calldata`

Category	Severity	Location	Status
Gas Optimization	Optimization	crossChain/Wrapper.sol (v2): 84~86	Resolved

Description

Non changed arguments of external functions are declared as `memory`.

Recommendation

We recommend declaring the non changed arguments of external functions as `calldata` to save gas.

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.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functions being invoke-able by anyone under certain circumstances.

Volatile Code

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

Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code, such as a constructor assignment imposing different require statements on the input variables than a setter function.

Magic Numbers

Magic Number findings refer to numeric literals that are expressed in the codebase in their raw format and should otherwise be specified as constant contract variables aiding in their legibility and maintainability.

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.