ZKFair

Audit Report

Wed Dec 20 2023

✉ contact@scalebit.xyz https://twitter.com/scalebit_

ScaleBit

ZKFair Audit Report

1 Executive Summary

1.1 Project Information

Description	ZKFair is the ZK-Rollup on ethereum based on Polygon CDK and Celestia DA. The contracts included in the audit scope are the differences between commit 34e2e9e7e2e70e7cb44988335465fdd8b4781c29 and cecd53e0b1e39cd9df1a79215eedbbb636b4e0a7.
Type	L2
Auditors	ScaleBit
Timeline	Mon Dec 18 2023 - Wed Dec 20 2023
Languages	Solidity
Platform	Ethereum
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/ZKFair/zkfair-cdk-validium-contracts
Commits	34e2e9e7e2e70e7cb44988335465fdd8b4781c29

1.2 Files in Scope

The following are the SHA1 hashes of the original reviewed files.

ID	File	SHA-1 Hash
DCO	contracts/lib/DepositContract.sol	fa95ec31f9921eac4b1726504ba95f193c2d12d6
EMA	contracts/lib/EmergencyManager.sol	e5535f95b992de4b8d974bee0b1d194718d33be5
PZEVMB	contracts/PolygonZkEVMBridge.sol	0f1ed9323f870ce88c5992e510eb84fd7b5e523e

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	4	0	4
Informational	3	0	3
Minor	1	0	1
Medium	0	0	0
Major	0	0	0
Critical	0	0	0

1.4 ScaleBit Audit Breakdown

ScaleBit aims to assess repositories for security-related issues, code quality, and compliance with specifications and best practices. Possible issues our team looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Integer overflow/underflow
- Number of rounding errors
- Unchecked External Call
- Unchecked CALL Return Values
- Functionality Checks
- Reentrancy
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic issues
- Gas usage
- Fallback function usage
- tx.origin authentication
- Replay attacks
- Coding style issues

1.5 Methodology

The security team adopted the **"Testing and Automated Analysis"**, **"Code Review"** and **"Formal Verification"** strategy to perform a complete security test on the code in a way that is closest to the real attack. The main entrance and scope of security testing are stated in the conventions in the "Audit Objective", which can expand to contexts beyond the scope according to the actual testing needs. The main types of this security audit include:

(1) Testing and Automated Analysis

Items to check: state consistency / failure rollback / unit testing / value overflows / parameter verification / unhandled errors / boundary checking / coding specifications.

(2) Code Review

The code scope is illustrated in section 1.2.

(3) Audit Process

- Carry out relevant security tests on the testnet or the mainnet;
- If there are any questions during the audit process, communicate with the code owner in time. The code owners should actively cooperate (this might include providing the latest stable source code, relevant deployment scripts or methods, transaction signature scripts, exchange docking schemes, etc.);
- The necessary information during the audit process will be well documented for both the audit team and the code owner in a timely manner.

2 Summary

This report has been commissioned by ZKFair to identify any potential issues and vulnerabilities in the source code of the ZKFair smart contract, as well as any contract dependencies that were not part of an officially recognized library. In this audit, we have utilized various techniques, including manual code review and static analysis, to identify potential vulnerabilities and security issues.

During the audit, we identified 4 issues of varying severity, listed below.

ID	Title	Severity	Status
IPZ-1	Inaccurate Error Code	Informational	Acknowledged
PZE-1	Lack of Events Emit	Minor	Acknowledged
PZE-2	Unused Params	Informational	Acknowledged
PZE-3	Unnecessary Check	Informational	Acknowledged

3 Participant Process

Here are the relevant actors with their respective abilities within the ZKFair Smart Contract:

Admin

- The admin has the capability to invoke the `setBridgeSettingsFee` function to change `feeAddress` addresses.

User

- Users can invoke the `bridgeAsset` function to deposit token to bridge.
- Users can invoke the `claimAsset` function to claim token from other network.
- Users can utilize the `getTokenWrappedAddress` function to get the address of a wrapper using the token information if already exist
- Users have the option to call the `precalculatedWrapperAddress` function to get the precalculated address of a wrapper using the token information.

4 Findings

IPZ-1 Inaccurate Error Code

Severity: Informational

Status: Acknowledged

Code Location:

contracts/interfaces/IPolygonZkEVMBridge.sol#19

Descriptions:

Code in lines 197 and 205 correspond to different situations, the `AmountDoesNotMatchMsgValue` error code aborts when the amount does not match `msg.value`, but code in 205 compares the value of `bridgeFee` with `msg.value`, so it recommended to use other error codes to abort.

Suggestion:

It is recommended to use other error codes to abort, such as `AmountDoesNotMatchBridgeFee`.

PZE-1 Lack of Events Emit

Severity: Minor

Status: Acknowledged

Code Location:

contracts/PolygonZkEVMBridge.sol#682

Descriptions:

The smart contract lacks appropriate events for monitoring sensitive operations, which could make it difficult to track sensitive actions or detect potential issues.

Suggestion:

It is recommended to emit events for those sensitive functions.

PZE-2 Unused Params

Severity: Informational

Status: Acknowledged

Code Location:

contracts/PolygonZkEVMBridge.sol#181

Descriptions:

Due to changes in the `bridgeAsset` function, the parameter `permitData` is not used, it is recommended to remove the unused parameter in the function. The constants `_PERMIT_SIGNATURE` and `_PERMIT_SIGNATURE_DAI` are also redundant.

Suggestion:

It is recommended to remove the unused parameter in the function if there is no further design.

Resolution:

The client replied that they have retained this section of the code to maintain consistency with the previous version.

PZE-3 Unnecessary Check

Severity: Informational

Status: Acknowledged

Code Location:

contracts/PolygonZkEVMBridge.sol#290

Descriptions:

In the `bridgeAsset` function, the `feeAddress` will never be set to 0 address, so the `if` condition `feeAddress ! = address(0)` is redundant.

Suggestion:

It is suggested to remove unnecessary `if` check statements.

Appendix 1

Issue Level

- **Informational** issues are often recommendations to improve the style of the code or to optimize code that does not affect the overall functionality.
- **Minor** issues are general suggestions relevant to best practices and readability. They don't post any direct risk. Developers are encouraged to fix them.
- **Medium** issues are non-exploitable problems and not security vulnerabilities. They should be fixed unless there is a specific reason not to.
- **Major** issues are security vulnerabilities. They put a portion of users' sensitive information at risk, and often are not directly exploitable. All major issues should be fixed.
- **Critical** issues are directly exploitable security vulnerabilities. They put users' sensitive information at risk. All critical issues should be fixed.

Issue Status

- **Fixed:** The issue has been resolved.
- **Partially Fixed:** The issue has been partially resolved.
- **Acknowledged:** The issue has been acknowledged by the code owner, and the code owner confirms it's as designed, and decides to keep it.

Appendix 2

Disclaimer

This report is based on the scope of materials and documents provided, with a limited review at the time provided. Results may not be complete and do not include all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your own risk. A report does not imply an endorsement of any particular project or team, nor does it guarantee its security. These reports should not be relied upon in any way by any third party, including for the purpose of making any decision to buy or sell products, services, or any other assets. TO THE FULLEST EXTENT PERMITTED BY LAW, WE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, IN CONNECTION WITH THIS REPORT, ITS CONTENT, RELATED SERVICES AND PRODUCTS, AND YOUR USE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NOT INFRINGEMENT.