Polygon zkEVM Security Review

zkEVM ROM June Upgrade features review

Auditors

Pawel Bylica, Lead Security Researcher

Andrei Maiboroda, Lead Security Researcher

Report prepared by: Pablo Misirov

June 20, 2023

Contents

1 About Spearbit

Spearbit is a decentralized network of expert security engineers offering reviews and other security related services to Web3 projects with the goal of creating a stronger ecosystem. Our network has experience on every part of the blockchain technology stack, including but not limited to protocol design, smart contracts and the Solidity compiler. Spearbit brings in untapped security talent by enabling expert freelance auditors seeking flexibility to work on interesting projects together.

Learn more about us at spearbit.com

2 Introduction

Polygon zkEVM is a new zk-rollup that provides Ethereum Virtual Machine (EVM) equivalence (opcode-level compatibility) for a transparent user experience and existing Ethereum ecosystem and tooling compatibility.

Disclaimer: This security review does not guarantee against a hack. It is a snapshot in time of the zkevm-rom fix-RLP-push0 feature according to the specific commit. Any modifications to the code will require a new security review.

3 Risk classification

Severity level	Impact: High	Impact: Medium	Impact: Low
Likelihood: high	Critical	High	Medium
Likelihood: medium	High	Medium	Low
Likelihood: low	Medium	Low	Low

3.1 Impact

- High - leads to a loss of a significant portion (>10%) of assets in the protocol, or significant harm to a majority of users.
- Medium - global losses <10% or losses to only a subset of users, but still unacceptable.
- Low - losses will be annoying but bearable--applies to things like griefing attacks that can be easily repaired or even gas inefficiencies.

3.2 Likelihood

- High - almost certain to happen, easy to perform, or not easy but highly incentivized
- Medium - only conditionally possible or incentivized, but still relatively likely
- Low - requires stars to align, or little-to-no incentive

3.3 Action required for severity levels

- Critical - Must fix as soon as possible (if already deployed)
- High - Must fix (before deployment if not already deployed)
- Medium - Should fix
- Low - Could fix

4 Executive Summary

Over the course of 1 day in total, Polygon engaged with Spearbit to review the PUSH0 feature and a change to RLP parsing. In this period of time a total of **8** issues were found.

Summary

Project Name	Polygon
Repository	0xPolygonHermez
Commit	fix-RLP-push0
Type of Project	Assembly, zkEVM
Audit Timeline	May 30 - May 31
Two week fix period	May 31 - June 14

Issues Found

Severity	Count	Fixed	Acknowledged
Critical Risk	0	0	0
High Risk	2	2	0
Medium Risk	0	0	0
Low Risk	1	1	0
Gas Optimizations	0	0	0
Informational	5	4	1
Total	**8**	**7**	**1**

5 Findings

5.1 High Severity

5.1.1 Lack of ZK counters check in PUSH0

Severity: *High Risk*

Context: stack-operations.zkasm#L11

Description: PUSH0 implementation does not update ZK counters.

Recommendation: Check for out of counters as in other PUSH instructions.

Polygon zkEVM: Fixed by PR 290.

Spearbit: Fixed.

5.1.2 Lack of airth ZK counters check for effective gas calculations

Severity: *High Risk*

Context: process-tx.zkasm#L88

Description: Effective gas implementation should account for arith ZK counters, because it uses `ARITH` twice.

Recommendation: Add arith counters check.

Polygon zkEVM: Fixed by PR 290.

Spearbit: Fixed.

5.2 Low Severity

5.2.1 Lack of documentation comment for PUSH0

Severity: *Low Risk*

Context: stack-operations.zkasm#L1-L9

Description: Comment above `opPUSH0` applies to `opPUSH1` and further.

Recommendation: Move `opPUSH0` above this comment and add its own documentation comment.

Polygon zkEVM: Fixed by PR 290.

Spearbit: Fixed.

5.3 Informational

5.3.1 Documentation about RLP fix logic adviced

Severity: *Informational*

Context: utils.zkasm#L931

Description: No documentation about why RLP error processing is different during RLP parsing and later.

Recommendation: Add an explanation comment.

Polygon zkEVM: Fixed by PR 290.

Spearbit: Fixed.

5.3.2 General feedback about RLP fix approach

Severity: *Informational*

Context: vars.zkasm#L51

Description: In general it feels like fixing the problem of managing one global thing (ZK counters, behaving differently in different contexts), with adding more global data. Overall this increases complexity and potential implicit interdependencies and difficulty of reasoning about code.

Recommendation: Ideally I think it would be better to make this dependency explicit by e.g. having an extra flag parameter to those utils that are called from RLP processing and should process counter errors differently.

I can see that this might be impractical if those utils are also called from many other places. Another less radical suggestion to consider could be: do the branching based on global isLoadingRLP only inside required utils instead of lower-level handleBatchError. This way the fix would affect only some limited scope of functionality, comparing to affecting every batch error processing like now. And it would still be more explicit which utils depend on this global state.

(Downside of this approach could be if RLP processing changes to using some new utils, you have to remember to check isLoadingRLP in each one)

Polygon zkEVM: We decided to go with the `isLoadingRLP` approach since it provides better security when adding new code to the RLP parsing since it is handled globally and it is not function specific. Also, RLP error where just removed and all zk-counters are handled via `handleBatchError`.

Spearbit: Acknowledged.

5.3.3 Redundant comment

Severity: *Informational*

Context: process-tx.zkasm#L74

Description: Comment on line 74 is duplicated on line 83.

Recommendation: Remove comment on line 74.

Polygon zkEVM: Fixed by PR 290.

Spearbit: Fixed.

5.3.4 Confusing notation in comments

Severity: *Informational*

Context: process-tx.zkasm#L76 process-tx.zkasm#L78

Description: Comment notation with => is confusing, because meaning is opposite of zkasm syntax.

Recommendation: Use = like `A = gasPrice` or reverse the sides like `gasPrice => A`.

Polygon zkEVM: Fixed by PR 290.

Spearbit: Fixed.

5.3.5 Inconsistent comment

Severity: *Informational*

Context: process-tx.zkasm#L79

Description: The range `[0, 255]` in comment corresponds to `B` value before the operation, while in other places comments explain the effect of operation.

Recommendation: Replace with `[1, 256]`.

Polygon zkEVM: Fixed by PR 290.

Spearbit: Fixed.