Security Amendment Statement

To Whom It May Concern,

This statement serves to confirm that Hexens, a reputable cybersecurity provider, has successfully conducted a security amendment for RISC Zero. The purpose of this endeavor was to enhance the cybersecurity measures of RISC Zero and address identified vulnerabilities within its zero-knowledge-proof implementation software systems.

The assessment was conducted by one Senior Security Engineer, with a total hours spent:

- C++ EDSL Circuit - 1200 hours during 10 engineering weeks with a start on 8 November 2023
- Stark-to-Snark Circuit - 480 hours during 4 engineering weeks with a start on 12 February 2024

Through rigorous analysis, various security issues were identified.
The following is a list of the issues identified during the security assessment and consequently resolved by the RISC Zero engineering team:

C++ EDSL Circuit:

Scope:
zirgen/circuit/rv32im
zirgen/components

- Issue 1. Missing Binary Constraints

Reg checkDirty, Reg isNewAddr fields are missing binary constraints, change to the Bit type
Severity: low
Status: Fixed
RISC Zero Comment:
The checkDirty constraint should be safe with the condition that 0 means terminate and is the only case in which it's OK for the memory argument to skip checking dirty bits. Any other non-zero value is safe since it will conservatively force a dirty check. Using a Bit for checkDirty complicates the circuit and causes a performance regression that is unacceptable.

All other recommendations for using Bit have been applied.

- Issue2: Refactor/Improvement Components Register Type

OneHot encoding component uses ```std::vector<Reg> bits``` for each of bit. It would be more clear to change it to ```std::vector<Bit> bits```.
DivideCycle component uses ```Reg isSigned, Reg onesComplement``` binary types. It would be more clear to change it to ```Bit isSigned, Bit onesComplement``` for safety

Severity: low
Status: Partially fixed (OneHot not fixed, DivideCycle fixed)
RISC Zero Comment: OneHot is constrained to using a Bit, which causes the recursion circuit to break the existing trusted setup. It feels like this is a low risk issue but this one is open to discussion. If it turns out it is higher risk, we could try to split the rv32im and recursion code bases to at least get the fix in for rv32im (unfortunately we can't make a change to recursion without requiring a new trusted setup ceremony).

- Issue 3: Constraint Documentation

RamPlonkVerifier, BytesPlonkVerifier, WomPlonkVerifier components verifications are using ```back``` to get old value from the trace. Document or add assertions that passed cycles are more than ```back```.

Severity: Informational
Status: Acknowledged

- Issue 4: Missing constraint on guest address range check-in BigInt component

RISC Zero VM reads the addresses of Big Int syscall numbers from registers. There are missing constraint on writing/reading data from addresses to be from guest memory range.

Severity: Critical
Status: Acknowledged/Acceptable Risk
Risc0 Comment: Due to limitations in both degree and columns, adding this constraint is infeasible. We recommend that users use the approved host-side and guest-side crates that check that addresses are within range.

- Issue 5: Missing constraint on guest address range check-in ECall

There are missing constraints on writing/reading data from addresses to be from guest memory range.

Severity: critical

Status: Acknowledged/Acceptable Risk
Comment: same as 4

- Issue 6: Missing constraint on division by zero in the Division component

Severity: Critical

RISC Zero VM's executor in case of division checks whether denomerator is zero return quotient 0xffffffff and reminder is numerator.

Severity: critical
Status: Fixed

Stark-to-Snark Circuit:

- Issue 1: Missing constraint on count_high in Valid Baby Bear component

Severity: Low
Resolution: Fixed

- Issue 2: Missing constraints in NormalizeImpl component

Severity: High
Resolution: Fixed

- Issue 3: Optimize power raising operation

Severity: Informational
Resolution: Acknowledged

This statement attests to the successful collaboration between Hexens and RISC Zero in bolstering cybersecurity defenses and signifies our commitment to delivering excellence in shipping emerging technologies developed by RISC Zero.

Should further inquiries arise regarding this matter, please do not hesitate to contact the undersigned for additional clarification or information.

Sincerely,

Konstantin Andriotis
COO
Hexens Cyber Security Ltd.

DocuSigned by:
Konstantin Andriotis
68BA3539DE8E4C5...