SBP Automated Security Assessment for *peaq*

Security Research Labs

2022-11-22

Introduction and Scope

As part of the Substrate Builders Program, the blockchain security team of SRLabs has been tasked to provide a semi-automated security assessment for peaq.

This assessment was conducted on the week of the 19th of September 2022, targeting specifically the `peaq-node-runtime` runtime on branch `parity_test_1.0`.

The automatic analysis is capable of identifying the following security vulnerabilities:

- Reachable runtime panic conditions
- Integer overflows
- Underweighted extrinsics
- Unbounded allocations
- Unbenchmarked extrinsics
- Exponential complexity in weights
- Use of twox_64_concat hasher for user provided keys
- Unsafe code
- Lack of storage deposits
- Wrongly configured ExistentialDeposit

Logic bugs are still the most widespread and impactful security bug category identified during blockchain security assurance projects provided by SRLabs. Finding a logic bug is a complex process that requires the audit team to have detailed knowledge of the involved protocols and business logic. Please note that this cannot be covered via the semi-automated tooling used for this assessment.

Findings

We have found a total of 3 vulnerabilities during this review.

[High] Extrinsics with default weights can cause block production timeouts

Summary

A number of extrinsics throughout the Peaq codebase are assigned default weights, e.g. 10_000 which can lead to overweight blocks and cause block production timeouts.

Issue details

Default weights are present in the following pallets and extrinsics:

peaq-pallet-did

- add_attribute
- update_attribute
- read_attribute
- remove_attribute

peaq-pallet-transaction

- service_requested
- service_delivered

Risk

Each of the extrinsics must have a weight that is calculated based on the computational complexity and database access of the extrinsic. Underweighted extrinsics enable attackers to create overweight blocks that could subsequently cause block production timeouts. This can slow down transaction processing and potentially stall the chain if all collators miss their block production slots.

Mitigation

Make sure that all the extrinsics are assigned benchmarked weights, in accordance with their computational complexity and database access.

[High] Extrinsics with missing storage deposits/fees could clutter the blockchain storage

Summary

Storage deposit fees are missing for pallet peaq-pallet-did. An attacker could perform a Denial of Service attack against the Peaq blockchain by calling extrinsics that save data into the blockchain database many times to clutter the underlying storage.

Issue details

The following extrinsics are affected:

peaq-pallet-did

- The extrinsics `add_attribute` and `update_attribute` are missing storage deposit for the items: `AttributeStore`.

Risk

As a best practice, every call that writes data into the storage database should be charged a storage deposit or fee. In the case of missing storage deposits for an extrinsic, an attacker could call the extrinsic multiple times to cheaply fill up the blockchain storage. This would result in overloading the database and making the blockchain harder to operate.

Mitigation

Implement extra fees or deposits for all affected extrinsics that save data to the blockchain storage. Deposits should be returned to the caller of the extrinsic once the data is removed from the storage database.

[High] Integer overflows in peaq-pallet-did

Summary

In the `peaq-pallet-did` pallet, the extrinsic add_attribute has an optional `valid_for` parameter, that adds itself to the current block number which can overflow if the provided value is sufficiently high. The same issue also exists for the create in a similar fashion.

Risk

By exploiting this integer overflow, an attacker could:

- Crash any node compiled in debug mode or with overflow checks enabled.
- On nodes that are compiled without overflow checks, this will lead to unexpected behaviors and logic inconsistencies.

Mitigation

Use safe math functions or perform overflow checks while doing math operations.