Quantstamp Security Assessment Certificate

QUANTSTAMP VERIFIED · SECURITY CERTIFICATE

September 18th 2021 — Quantstamp Verified

AStarNetwork: Custom-Signature

This security assessment was prepared by Quantstamp, the leader in blockchain security

Executive Summary

Type	Substrate-Based Blockchain
Auditors	Poming Lee, Research Engineer Souhail Mssassi, Research Engineer
Timeline	2021-08-31 through 2021-09-18
Languages	Rust
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	None
Documentation Quality	Undetermined
Test Quality	Undetermined

Source Code

Repository	Commit
Astar	7bb088b
Astar	ac131c7

Total Issues	4 (4 Resolved)
High Risk Issues	1 (1 Resolved)
Medium Risk Issues	2 (2 Resolved)
Low Risk Issues	1 (1 Resolved)
Informational Risk Issues	0 (0 Resolved)
Undetermined Risk Issues	0 (0 Resolved)

High Risk	The issue puts a large number of users' sensitive information at risk, or is reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
Medium Risk	The issue puts a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or is reasonably likely to lead to moderate financial impact.
Low Risk	The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low-impact in view of the client's business circumstances.
Informational	The issue does not post an immediate risk, but is relevant to security best practices or Defence in Depth.
Undetermined	The impact of the issue is uncertain.

Unresolved	Acknowledged the existence of the risk, and decided to accept it without engaging in special efforts to control it.
Acknowledged	The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings).
Resolved	Adjusted program implementation, requirements or constraints to eliminate the risk.
Mitigated	Implemented actions to minimize the impact or likelihood of the risk.

Summary of Findings

During auditing, we found 4 potential issues of various levels of severity: 1 high-severity, 2 medium-severity, and 1 low-severity issues. We highly recommend addressing the findings before going live.

2021-09-18: during this reaudit, the admin team has either brought all the status of findings into fixed or mitigated.

ID	Description	Severity	Status
QSP-1	Overflow on The Libsecp256k1	⌃⌃ High	Fixed
QSP-2	Lack of Validation in the `what` Parameter	⌃ Medium	Mitigated
QSP-3	Cross Chain Replay Attack is Possible	⌃ Medium	Fixed
QSP-4	Order Logic In Nonce Increment	⌄ Low	Fixed

Quantstamp Audit Breakdown

Quantstamp's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.

Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow / underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Methodology

The Quantstamp auditing process follows a routine series of steps:

1. Code review that includes the following
   i. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
   ii. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
   iii. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Quantstamp describe.
2. Testing and automated analysis that includes the following:
   i. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
   ii. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

Toolset

The notes below outline the setup and steps performed in the process of this audit.

Setup

Tool Setup:

- Rust Audit v0.15.0
- Rust-Clippy Latest

Steps taken to run the tools:

```
cargo install cargo-audit cargo audit rustup component add clippy cargo clippy
```

Findings

QSP-1 Overflow on The Libsecp256k1

Severity: *High Risk*

Status: Fixed

File(s) affected: `frame/custom-signatures/Cargo.toml`

Description: In `frame/custom-signatures/Cargo.toml` (L23): Libsecp256k1 accepts signatures whose R or S parameter is larger than the secp256k1 curve order, which differs from other implementations. This could lead to invalid signatures being verified. This error is resolved in 0.5.0 by adding a check_overflow flag.

Recommendation: Update the Libsecp Library to the latest version (Greater than 0.5.0).

QSP-2 Lack of Validation in the `what` Parameter

Severity: *Medium Risk*

Status: Mitigated

File(s) affected: `frame/custom-signatures/src/ethereum.rs`

Description: In `frame/custom-signatures/src/ethereum.rs` (L44): the `signable_message` function takes as parameters the variable `what` and then it executes a loop N times such that N is the length of the variable `what`. The problem here is that there is no limit on the length of this variable, which can cause a denial of service during the execution of this function call.

Recommendation: Enforce a limitation on the size of the `what` parameter.

Update: 2021-09-17: The original issue was solved by using a double-hashing approach. However, the admin team should be aware that this approach increases the probability of collision and could thus introduce additional risk to the system.

QSP-3 Cross Chain Replay Attack is Possible

Severity: *Medium Risk*

Status: Fixed

File(s) affected: `frame/custom-signatures/src/lib.rs`

Description: Cross chain replay attack is possible for the current custom signature design because there is no information in the signature which blockchain a signature is intended for.

Recommendation: Include the `chainId` into the signature to distinguish a signature for a specific blockchain from the other blockchains. Reference: Signing Data with MetaMask.

Update: 2021-09-17: The admin team stated that ChainId will be used to fill in the magic number field.

QSP-4 Order Logic In Nonce Increment

Severity: *Low Risk*

Status: Fixed

File(s) affected: `frame/custom-signatures/src/lib.rs`

Description: In `frame/custom-signatures/src/lib.rs` (L113): in the call function, the first thing verified is the validity of the transaction with the help of the nonce, once the verification has been done, this nonce will be incremented and if the following lines have some problems or errors, the nonce will be incremented and the call will not be executed properly.

Recommendation: Increment the nonce after the validation and verification of the signature.

Automated Analyses

Rust Audit

RUSTSEC-2021-0076: libsecp256k1: libsecp256k1 allows overflowing signatures › RustSec Advisory Database https://rustsec.org/advisories/RUSTSEC-2021-0076

Crate: libsecp256k1 Version: 0.3.5 Title: libsecp256k1 allows overflowing signatures Date: 2021-07-13 ID: RUSTSEC-2021-0076 URL: https://rustsec.org/advisories/RUSTSEC-2021-0076 Solution: Upgrade to >=0.5.0

This security finding has been added as a finding in the finding section.

Rust-Clippy

```
For Clippy Rust
warning: this expression borrows a reference (`&[u8]`) that is immediately dereferenced by the compiler
  --> src/ethereum.rs:64:48
   |
64 |         let msg = keccak_256(&signable_message(&msg.get()));
   |                                                ^^^^^^^^^^ help: change this to: `msg.get()`
   |
   = note: `#[warn(clippy::needless_borrow)]` on by default
   = help: for further information visit https://rust-lang.github.io/rust-clippy/master/index.html#needless_borrow
warning: you seem to be trying to use `&Box<T>`. Consider using just `&T`
   --> src/lib.rs:146:19
    |
146 |             call: &Box<<T as Config>::Call>,
    |                   ^^^^^^^^^^^^^^^^^^^^^^^^^ help: try: `&<<T as Config>::Call`
    |
    = note: `#[warn(clippy::borrowed_box)]` on by default
    = help: for further information visit https://rust-lang.github.io/rust-clippy/master/index.html#borrowed_box
warning: using `clone` on type `<T as frame_system::Config>::Index` which implements the `Copy` trait
   --> src/lib.rs:151:55
    |
151 |             let payload = (T::CallMagicNumber::get(), nonce.clone(), call.clone());
    |                                                       ^^^^^^^^^^^^^ help: try dereferencing it: `*nonce`
    |
    = note: `#[warn(clippy::clone_on_copy)]` on by default
    = help: for further information visit https://rust-lang.github.io/rust-clippy/master/index.html#clone_on_copy
warning: 3 warnings emitted
```

Test Results

Test Suite Results

All tests pass.

```
running 6 tests
test tests::__construct_runtime_integrity_test::runtime_integrity_tests ... ok
test ethereum::verify_should_works ... ok
test tests::call_fixtures ... ok
test tests::eth_sign_works ... ok
test tests::invalid_signature ... ok
test tests::balance_transfer ... ok

test result: ok. 6 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.66s

   Doc-tests pallet-custom-signatures

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

Code Coverage

Rust: Test coverage (only Line Coverage) of the Rust code was tried to calculated with `tarpaulin`. Errors were encountered for this project and thus the coverage score could not be obtained. Errors are attached as below.

```
Sep 13 08:55:06.695 INFO cargo_tarpaulin::process_handling::linux: Launching test Sep 13 08:55:06.695 INFO cargo_tarpaulin::process_handling: running /tmp/20210913-1445/202109B-Astar-7bb088b-real-1st-audit/target/debug/deps/pallet_custom_signatures-bfb22441634835db Sep 13 08:55:06.696 ERROR cargo_tarpaulin: Failed to run tests: ASLR disable failed: EPERM: Operation not permitted Error: "Failed to run tests: ASLR disable failed: EPERM: Operation not permitted" Sep 13 08:55:10.674 ERROR cargo_tarpaulin: Failed to get test coverage! Error: Failed to run tests: Unexpected signal when starting test Error: "Failed to get test coverage! Error: Failed to run tests: Unexpected signal when starting test"
```

Appendix

File Signatures

The following are the SHA-256 hashes of the reviewed files. A file with a different SHA-256 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different SHA-256 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review.

Contracts

`cb208e81798ed5a97a12167e614db17573a0ae55b6434dd57f8ea7fbca53f0be ./custom-signatures/src/ethereum.rs`

`6f684755c9098b73d76ae389befdb1f39c7b20c2d3cf03f66c9c8c661df2118b ./custom-signatures/src/lib.rs`

Tests

`8fdb05e3c91a65b8a0f4d020ca8755c26c316c9ffaa2227f01547f672cdad707 ./custom-signatures/src/tests.rs`

Changelog

- 2021-09-14 - Initial report
- 2021-09-18 - final report

About Quantstamp

Quantstamp is a Y Combinator-backed company that helps to secure blockchain platforms at scale using computer-aided reasoning tools, with a mission to help boost the adoption of this exponentially growing technology.

With over 1000 Google scholar citations and numerous published papers, Quantstamp's team has decades of combined experience in formal verification, static analysis, and software verification. Quantstamp has also developed a protocol to help smart contract developers and projects worldwide to perform cost-effective smart contract security scans.

To date, Quantstamp has protected $5B in digital asset risk from hackers and assisted dozens of blockchain projects globally through its white glove security assessment services. As an evangelist of the blockchain ecosystem, Quantstamp assists core infrastructure projects and leading community initiatives such as the Ethereum Community Fund to expedite the adoption of blockchain technology.

Quantstamp's collaborations with leading academic institutions such as the National University of Singapore and MIT (Massachusetts Institute of Technology) reflect our commitment to research, development, and enabling world-class blockchain security.

Timeliness of content

The content contained in the report is current as of the date appearing on the report and is subject to change without notice, unless indicated otherwise by Quantstamp; however, Quantstamp does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following publication.

Notice of confidentiality

This report, including the content, data, and underlying methodologies, are subject to the confidentiality and feedback provisions in your agreement with Quantstamp. These materials are not to be disclosed, extracted, copied, or distributed except to the extent expressly authorized by Quantstamp.

Links to other websites

You may, through hypertext or other computer links, gain access to web sites operated by persons other than Quantstamp, Inc. (Quantstamp). Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites' owners. You agree that Quantstamp are not responsible for the content or operation of such web sites, and that Quantstamp shall have no liability to you or any other person or entity for the use of third-party web sites. Except as described below, a hyperlink from this web site to another web site does not imply or mean that Quantstamp endorses the content on that web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the report. Quantstamp assumes no responsibility for the use of third-party software on the website and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any outcome generated by such software.

Disclaimer

This report is based on the scope of materials and documentation provided for a limited review at the time provided. Results may not be complete nor inclusive of 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 sole risk. Blockchain technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.