StableSwap Audit

2023.03.12

by Formal Land

Table of Contents

1. Overview

1.1 Formal Land

Formal Land is an auditing and formal-verification company based in Paris, France, specialized in audits, security assessments, formal verification. We did verification of the implementation of the crypto-currency ꜩTezos (composed of around 100,000 lines of code).

1.2 Audit Methodology

1. The audit was scoped to the smart contract provided in the Github repository: https://github.com/saber-hq/stable-swap

2. The code is checked line by line for common vulnerabilities, code duplication, best practices and the code architecture.

3. Audit tools:

- cargo-audit
- rust-clippy

- cargo-deny check

- cargo upgrades

1.3 Review team

The audit has been performed with a total time donation of 1 business week.

The work was divided between the Chief Auditor and two auditors who performed manual code review.

Member's Name	Position
Guillaume Claret	Chief Auditor
Daniel Hilst Selli	Proof Engineer
Natalie Klaus	Proof Engineer

1.4 Disclaimer

We've put our best effort to find all vulnerabilities in the system, however our findings shouldn't be considered as a complete list of all existing issues.

2. Scope

The StableSwap is an automated market maker for mean-reverting trading pairs. The audit encompassed all code parts of StableSwap source code.

2.1 Review result: overview

The audit team reported a total of 18 findings, of which (with decreasing impact).

- 0 were critical
- 0 were high
- 0 were medium
- 10 were low
- 8 were informational

None of the issues requires immediate action.

3. Low level priority findings

Low severity issues are more comments and recommendations rather than security issues. We provide hints on how to improve code readability and follow best practices. Further actions depend on the development team decision.

3.1 Comments and documentation

During the first days of the audit we tried to receive general understanding of the whole system, however due to the low level of developer documentation, it was hard to understand how the program is supposed to work in general and the relations between the various components. Lack of documentation caused us to extend the general understanding period.

There is sufficient amount of comments for the functions on each file, however there is much more to be covered, there are some parts of code which lack comments completely. We suggest mandatory [rustdoc] annotations for all methods so documentation can be generated automatically.

3.2 TODOs

In https://github.com/saber-hq/stable-swap/tree/master/stable-swap-program there are `todos` left (mentioned in README.md): It is not recommended for production code to contain todos.

3.3 Cargo Audit

To audit dependencies of StableSwap for crates, and reveal security vulnerabilities, we used `cargo audit` tool. Cargo-audit is a command-line utility which inspects Cargo.lock files and compares them against the RustSec Advisory Database, a community database of security vulnerabilities maintained by the Rust Secure Code Working Group. The result of the cargo audit tool is the next:

error: 2 vulnerabilities found!

warning: 3 allowed warnings found

The result of the `cargo audit` command could be seen in Appendix A.

3.4 Cargo Clippy

The next standard tool we used was Cargo Clippy. Cargo Clippy is a collection of lints to catch common mistakes and improve Rust code.

All warnings were shown for the stable-swap-anchor package, while for stable-swap-program, stable-swap-client, stable-swap-fuzz, stable-swap-math and stable-swap-sim, cargo-clippy didn’t show any warnings.

The result of the `cargo-clippy` command could be seen in Appendix B.

3.5 Cargo Upgrades

Cargo Upgrades shows which dependencies in Cargo.toml can be upgraded to a newer version. In total 2 dependencies can be upgraded: the [borsh], from the [stable-swap-math] and [pyo3] from the [stable-swap-sim]. The result of the `cargo-upgrades` command could be seen in Appendix C.

3.6 Cargo deny

Cargo-deny is a cargo plugin that helps to lint project's dependency graph to ensure all projects dependencies conform to requirements. The result of running the tool was the next:

- 2 checks failed : advisories, licenses
- 2 passed : bans, sources

Example of Errors and Warnings:

```
error[rejected]: failed to satisfy license requirements
  ┌─ arrayref 0.3.6 (registry+https://github.com/rust-lang/crates.io-
index):4:12
  │              license expression retrieved via Cargo.toml `license`
  │              rejected: not explicitly allowed
warning[duplicate]: found 2 duplicate entries for crate 'bs58'
```

The part of result of the [cargo deny check] command could be seen in Appendix C.

5. Informational Findings

1. In README.md file, in documentation section, the link Saber developer documentation website, Leads to “Page not found”.

2. In README.md file, audit section https://www.bramah.systems/ website is down (server is not responding).

3. Rust Crate → Package → Stable-Swap-Anchor → README.md: https://github.com/saber-hq/stable-swap/tree/master/stable-swap-anchor, in the Documentation section, the link Saber developer documentation website, Leads to “Page not found”.

4. Rust Crate → Package → Stable-Swap-Client → README.md: https://github.com/saber-hq/stable-swap/tree/master/stable-swap-client, in the Documentation section, the link Saber developer documentation website, Leads to “Page not found”.

5. Rust Crate → Package → Stable-Swap-Math → README.md: https://github.com/saber-hq/stable-swap/tree/master/stable-swap-math, in the Documentation section, the link Saber developer documentation website, Leads to “Page not found”.

6. Rust Crate → Package → Stable-Swap-Program → README.md: https://github.com/saber-hq/stable-swap/tree/master/stable-swap-program, in the StableSwap Program section, the link: Curve's StableSwap invariant is not working.

7. Rust Crate → Package → Stable-Swap-Program → README.md: https://github.com/saber-hq/stable-swap/tree/master/stable-swap-program, in the StableSwap Program section, the link to live demo on the Solana testnet is not working.

8. Rust Crate → Package → Stable-Swap-Program → README.md: https://github.com/saber-hq/stable-swap/tree/master/stable-swap-program, in the Documentation section, the link Saber developer documentation website, Leads to “Page not found”.

6. Conclusion

We have discovered a number of a low level severity issues, and a few more informational. None of the issues requires immediate action.

We believe the project lacks technical leadership with clear rules and guidelines for development, commit messages, comments and documentation. This would definitely help future auditors (or developers) better understand the code and be able to focus on finding single issues.

This report summarizes the engagement, and findings. It also contains detailed descriptions of the discovered vulnerabilities, steps the Formal Land Team took to identify each issue.

APPENDIX A

The result of the cargo audit command:

```
error: 2 vulnerabilities found!

warning: 3 allowed warnings found

Fetching advisory database from `https://github.com/RustSec/advisory-db.git`

      Loaded 517 security advisories (from /home/natalie/.cargo/advisory-db)

    Updating crates.io index

    Scanning Cargo.lock for vulnerabilities (215 crate dependencies)

Crate:      remove_dir_all

Version:    0.5.3

Title:      Race Condition Enabling Link Following and Time-of-check Time-of-use (TOCTOU)
```

Date: 2023-02-24

ID: RUSTSEC-2023-0018

URL: https://rustsec.org/advisories/RUSTSEC-2023-0018

Solution: Upgrade to >=0.8.0

Dependency tree:

```
remove_dir_all 0.5.3
└── tempfile 3.3.0
    ├── rusty-fork 0.3.0
    │   └── proptest 1.0.0
    │       └── stable-swap-math 1.8.1
    │           └── stable-swap 1.8.1
    │               └── stable-swap-fuzz 1.8.1
    └── proptest 1.0.0
```

Crate: time

Version: 0.1.43

Title: Potential segfault in the time crate

Date: 2020-11-18

ID: RUSTSEC-2020-0071

URL: https://rustsec.org/advisories/RUSTSEC-2020-0071

Solution: Upgrade to >=0.2.23

Dependency tree:

```
time 0.1.43
└── chrono 0.4.22
    ├── stable-swap-fuzz 1.8.1
    └── solana-sdk 1.9.18
        └── stable-swap 1.8.1
            └── stable-swap-fuzz 1.8.1
```

Crate: failure

Version: 0.1.8

Warning: unmaintained

Title: failure is officially deprecated/unmaintained

Date: 2020-05-02

ID: RUSTSEC-2020-0036

URL: https://rustsec.org/advisories/RUSTSEC-2020-0036

Dependency tree:

```
failure 0.1.8
├── ed25519-dalek-bip32 0.1.1
│   └── solana-sdk 1.9.18
│       └── stable-swap 1.8.1
│           └── stable-swap-fuzz 1.8.1
```

```
└── derivation-path 0.1.3
    ├── solana-sdk 1.9.18
    └── ed25519-dalek-bip32 0.1.1
```

Crate: cpufeatures

Version: 0.2.2

Warning: yanked

Dependency tree:

```
cpufeatures 0.2.2
└── sha2 0.9.9
    ├── solana-sdk 1.9.18
    |   └── stable-swap 1.8.1
    |       └── stable-swap-fuzz 1.8.1
    ...
```

APPENDIX B

The result of the cargo clippy command:

```
warning: the following explicit lifetimes could be elided: 'a, 'b, 'c

  --> stable-swap-anchor/src/instructions.rs:15:1

   |

15 | / pub fn initialize<'a, 'b, 'c, 'info>(

16 | |     ctx: CpiContext<'a, 'b, 'c, 'info, Initialize<'info>>,

17 | |     nonce: u8,

18 | |     amp_factor: u64,

19 | |     fees: stable_swap_client::fees::Fees,

20 | | ) -> Result<()> {

   | |_____________^

   |
```

```
help: replace with `'_` in generic arguments such as here
  --> stable-swap-anchor/src/instructions.rs:16:21
   |
16 |     ctx: CpiContext<'a, 'b, 'c, 'info, Initialize<'info>>,
   |                     ^^
   = help: for further information visit https://rust-lang.github.io/rust-clippy/master/index.html#needless_lifetimes
   = note: `#[warn(clippy::needless_lifetimes)]` on by default


warning: the `Err`-variant returned from this function is very large
  --> stable-swap-anchor/src/instructions.rs:20:6
   |
20 | ) -> Result<()> {
   |      ^^^^^^^^^^ the `Err`-variant is at least 160 bytes
```

```
   |

   = help: try reducing the size of `anchor_lang::error::Error`, for example
by boxing large elements or replacing it with
`Box<anchor_lang::error::Error>`

   = help: for further information visit https://rust-lang.github.io/rust-
clippy/master/index.html#result_large_err

   = note: `#[warn(clippy::result_large_err)]` on by default


warning: the following explicit lifetimes could be elided: 'a, 'b, 'c

  --> stable-swap-anchor/src/instructions.rs:70:1

   |

70 | / pub fn deposit<'a, 'b, 'c, 'info>(

71 | |     ctx: CpiContext<'a, 'b, 'c, 'info, Deposit<'info>>,

72 | |     token_a_amount: u64,

73 | |     token_b_amount: u64,
```

```
74 | |      min_mint_amount: u64,
75 | | ) -> Result<()> {
   | |_____________^
   |
help: replace with `'_` in generic arguments such as here
   |                    ^^
   = help: for further information visit https://rust-lang.github.io/rust-clippy/master/index.html#needless_lifetimes

warning: the `Err`-variant returned from this function is very large
   --> stable-swap-anchor/src/instructions.rs:332:6
   |
...
```

APPENDIX C

The result of the [cargo upgrades] command:

```
1. stable-swap-math: /home/natalie/stable_swap/stable-swap-math/Cargo.toml

      borsh matches 0.9.3; latest is 0.10.2


2. stable-swap-sim:
/home/natalie/stable_swap/stable-swap-math/sim/Cargo.toml

      pyo3 matches 0.16.6; latest is 0.18.1
```

APPENDIX D

The result of the [cargo deny check] command:

```
error[rejected]: failed to satisfy license requirements
  ┌─ addr2line 0.17.0 (registry+https://github.com/rust-lang/crates.io-index):4:12
  │
4 │ license = "Apache-2.0 OR MIT"
  │           ^^^^^^^^^^^----^^^
  │           │          │
  │           │          rejected: not explicitly allowed
  │           license expression retrieved via Cargo.toml `license`
  │           rejected: not explicitly allowed
  │
```

```
= addr2line v0.17.0
    └── backtrace v0.3.65
        └── failure v0.1.8
            ├── derivation-path v0.1.3
            │   ├── ed25519-dalek-bip32 v0.1.1
            │   │   └── solana-sdk v1.9.18
            │   │       └── (dev) stable-swap v1.8.1
            │   │           └── stable-swap-fuzz v1.8.1
            │   └── solana-sdk v1.9.18 (*)
            └── ed25519-dalek-bip32 v0.1.1 (*)
```

```
error[rejected]: failed to satisfy license requirements
  ┌─ adler 1.0.2 (registry+https://github.com/rust-lang/crates.io-index):4:12
  │
```

```
4 | license = "0BSD OR MIT OR Apache-2.0"
  |           ^^^^---^^^---^^^^^^^^^^^
  |           |    |     |
  |           |    |     rejected: not explicitly allowed
  |           |    rejected: not explicitly allowed
  |           license expression retrieved via Cargo.toml `license`
  |           rejected: not explicitly allowed
  |

= adler v1.0.2
  └── miniz_oxide v0.5.1
      └── backtrace v0.3.65
          └── failure v0.1.8
              ├── derivation-path v0.1.3
              │   ├── ed25519-dalek-bip32 v0.1.1
```

```
                │   │   └── solana-sdk v1.9.18
                │   │       └── (dev) stable-swap v1.8.1
                │   │           └── stable-swap-fuzz v1.8.1
                │   └── solana-sdk v1.9.18 (*)
                └── ed25519-dalek-bip32 v0.1.1 (*)
```

```
error[rejected]: failed to satisfy license requirements
  ┌─ ahash 0.7.6 (registry+https://github.com/rust-lang/crates.io-index):4:12
  │
4 │ license = "MIT OR Apache-2.0"
...
```