Currently the codecov support works for C code only.

Getting code coverage for the Rust to work is complex.

There are 2 ways to instrument Rust code:

- gcc compatible gcov support
- llvm native mode

I can't get the gcov support to work. When its compiled in (using -Zprofile) Suricata SEGVs on the simplest commands (like suricata -V). Maybe this is due to different llvm versions in use. I compiled with clang-10 while using rustc nightly which probably uses llvm 11 or a higher version.

With the native llvm instrumentation things compile, and a default.profraw is generated, but extracting actual coverage info from that has failed until now. The tools require an instrumented binary, where it rejects both the suricata binary as well as the libsuricata.a. Possibly this is another case where using different llvm versions is an issue.

So I think the next step should be to try if compiling both the C and Rust code with the same llvm backend makes things better.

Issues with rustc llvm coverage are
- no optimization https://github.com/rust-lang/rust/issues/82144
- no brotli https://github.com/rust-lang/rust/issues/82875