Suricata - Task #4221

Build Suricata into a static and shared library

12/16/2020 03:21 PM - Jason Ish

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Jason Ish</td>
</tr>
<tr>
<td>Category:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td>7.0rc1</td>
</tr>
<tr>
<td>Effort:</td>
<td></td>
</tr>
<tr>
<td>Difficulty:</td>
<td></td>
</tr>
</tbody>
</table>

Label:

Description
Part of libsuricata is simply building Suricata as a library that can be re-used in other projects. Ideally static and shared libraries should be made possible.

Related issues:
Related to Task #2693: tracking: libsuricata

History

#1 - 12/16/2020 03:21 PM - Jason Ish
- Related to Task #2693: tracking: libsuricata added

#2 - 12/16/2020 03:36 PM - Jason Ish
I've spent some time looking at ways to build a Suricata library. The biggest barrier is build tools like automake and libtool.

Currently with "--disable-shared" we build the Rust code into a static library and the C code into a static library. A final main binary can be built from these 2 static libraries without issue.

Without "--disabled-shared", libtool will build the C sources into shared and static libraries, however, coercing it to link against the static C libraries does not appear to work. So we end up with a binary that links against the static Rust code, but the dynamic C code, which works but is just very odd.

We can update the "--disabled-shared" to work against a shared library of Rust code, but as this is not generated by libtool there are some difficulties. Libtool sets the rpath in the binary to the final location where the library should be found. So an install Suricata built this way works, but the "./src/suricata" libtool wrapper script cannot find the Rust code shared library, as this library is not a libtool library. The LD_LIBRARY_PATH can be set, which would require our own custom wrapper script and then things get a little complicated, but I suppose its an option.

Given the dual shared library approach, CMake handles this a little different. While in the "build" tree, CMake can encode the RPATH in the binary to find the in-tree location of the shared libraries. On "make install" it can rewrite the RPATH to the final location. This gives us a little more flexibility, but a change to CMake would be huge.

The other thing I tried was using Cargo to generate the libraries. The build process needs to change a little for this:
- Go into Rust src and run cbindgen
- Go into C source and build static library from C source. Want to enable -fPIC no matter what.
- Go back into Rust src and build library, with some rustc flags to pull in the static C source code library.

The result is a single library, a .a and a .so. I have successfully linked with the .a to create a runnable Suricata, but have issues with the generated shared library. While "nm" does list symbols from the C code, I've been unable to actually build a binary that uses them. I get *undefined reference* errors during link. There are various other discussions and open issues around this, but I think it should be solvable. At least it looks promising as in the static case, its able to give us a single library file.

Note: Both autotools and CMake are unable to combine the code into a single library without some ugly hacks by us (exploding the Rust .a and C .a into object files and repacking them into a single .a). So we'd have to build the library somewhat manually, which differs on all platforms a little. Autotools seems to go out of its way to prevent you from manually building libraries without libtool.

#3 - 05/31/2021 04:59 PM - Jason Ish
- Status changed from New to Closed
- Assignee set to Jason Ish
- Target version set to 7.0rc1

Merged into master - https://github.com/OISF/suricata/pull/5866.

This is basic support for building Suricata into static and shared libraries with shared library support only on Linux at this time.
New Makefile targets were added:
- make install-headers (to install headers)
- make install-library (to install library files, and shared library on linux)