Suricata - Optimization #3314
rust: integrate log crate with suricata logging
11/05/2019 09:23 AM - Victor Julien

<table>
<thead>
<tr>
<th>Status:</th>
<th>Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Pierre Chifflier</td>
</tr>
<tr>
<td>Category:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td>TBD</td>
</tr>
<tr>
<td>Effort:</td>
<td></td>
</tr>
<tr>
<td>Difficulty:</td>
<td></td>
</tr>
</tbody>
</table>

Description
Make sure macros like debug!, info!, warn! etc. are useable and plugged to suricata's logging functions.

History
#1 - 11/05/2019 09:23 AM - Victor Julien
Pierre do you have an idea about how this can be done?

#2 - 11/05/2019 01:39 PM - Jason Ish
There are some issues we would run into here. The Rust std log implements:
- error
- warn
- info
- debug
- trace

All these follow the same call pattern, and do not give us a place to pass the code in error and warn. We could certainly intercept all these and pass them to our logging system with some default error code where required.

A pattern I've seen is an implementation of these so you get output when call them, but then application specific macros (ie: scwarn!, scerror!) where a lot more application specific context was added to meet the business requirements.

I'm not sure if looking into larger log implementations is worth it at this time as we're still best off passing to the C implementation of our logger for the foreseeable future.

#3 - 11/05/2019 03:16 PM - Danny Browning
Log crate doesn't actually provide a logger implementation, so we would have to implement a logger that can receive them, and pass them to the C implementation.

For error and warn, can use a generic error code.

#4 - 11/05/2019 03:19 PM - Victor Julien
The error code concern might go away depending how we deal with #3160.

#5 - 11/12/2019 09:56 PM - Jason Ish
I think I need a little reminder as to why this is necessary. I've created a PR that implements a logger here:
https://github.com/OISF/suricata/pull/4365

The only benefit I see is that we can now see the output from external crates that may log diagnostic information using the standard log macros. There is not much benefit for us to use it in tree.

Idea: Rename macros like SCLogNotice to simply scnotice!() (scinfo, scerror, etc) if that would help make this more usable, feel more Rusty.