The Hyperscan API makes the use of specialized instruction sets relatively straightforward. It provides `hs_populate_platform()` which returns an `hs_platform_info`, which can in turn be fed into the various `hs_compile` calls.

It looks like the HS module in Suricata is currently feeding NULL into these functions eg: `hs_compile_multi()`

```c
err = hs_compile_ext_multi((const char *const *)cd->expressions, cd->flags, cd->ids, (const hs_expr_ext_t *const *)cd->ext, cd->pattern_cnt, HS_MODE_BLOCK, NULL, &pd->hs_db, &compile_err);
```

I'm no C programmer, but this looks like an improvement I should be able to add with context from issue 2010. I've assigned it to myself for now and will see if I can get it going. Reading over this presentation makes it sound like AVX2/512 will be beneficial.

Please advise if this is obviously more complicated than it appears to me.