Suricata has had support for Lua for a long time. Originally we only supported luajit, but later this was extended to regular lua. Luajit was giving us lots of issues and some design limitations. Due to luajit targeting lua 5.1, we've worked with that for the regular lua as well. Lua 5.1 is starting to be considered legacy, and recent versions of distributions are not including it anymore.

One problem that we never really considered is that lua versions include breaking changes. So it is tricky to support all versions of lua in a single script (or maybe impossible). This means that a generic way to ship lua rules, for example as part of a ruleset, is also going to be difficult or even impossible. Unless we are willing to drop support for luajit, we can't move away from the lua 5.1 'standard'.

It looks like one of the main use cases for lua is gaming engines and it seems that these generally vendor lua in, so that they have a single version to work with. We could consider something like this. It would bring overhead and I could imagine distro's not being excited about this.

A drawback of lua is the fragile ecosystem in which many modules exist, but quality of them varies wildly. Also various methods for installing those modules exist. For new users this is quite confusing. Despite these drawbacks the modules have made some really interesting script possible, either for real time inspection or more focused at offline pcap analysis.

At Suricon 2019 it was suggested that perhaps a more modern technology like WASM would be a better fit for us.

When considering the future of our lua integration, it might be interesting to keep an eye on [https://github.com/dibyendumajumdar/ravi](https://github.com/dibyendumajumdar/ravi)