Contribution ID: **164** Type: **not specified**

ELF relocation for static data in BPF

Thursday, November 15, 2018 9:50 AM (20 minutes)

BPF program writers today who build and distribute their programs as ELF objects typically write their programs using one of a small set of (mostly) similar headers that establish norms around ELF section definitions. One such norm is the presence of a "maps" section which allows maps to be referenced within BPF instructions using virtual file descriptors. When a BPF loader (eg, iproute2) opens the ELF, it loads each map referred in this section, creates a real file descriptor for that map, then updates all BPF instructions which refer to the same map to specify the real file descriptor. This allows symbolic referencing to maps without requiring writers to implement their own loaders or recompile their programs every time they create a map.

This discussion will take a look at how to provide similar symbolic referencing for static data. Existing implementations already templatize information such as MAC or IP addresses using C macros, then invoke a compiler to replace such static data at load time, at a cost of one compilation per load. By extending the support for static variables into ELF sections, programs could be written and compiled once then reloaded many times with different static data.

Presenters: STRINGER, Joe (Cilium); BORKMANN, Daniel (Cilium)

Session Classification: BPF MC