Contribution ID: **143** Type: **not specified**

Building Socket-aware BPF Programs

Tuesday, November 13, 2018 11:00 AM (35 minutes)

Over the past several years, BPF has steadily become more powerful in multiple ways: Through building more intelligence into the verifier which allows more complex programs to be loaded, and through extension of the API such as by adding new map types and new native BPF function calls. While BPF has its roots in applying filters at the socket layer, the ability to introspect the sockets relating to traffic being filtered has been limited.

To build such awareness into a BPF helper, the verifier needs the ability to track the safety of the calls, including appropriate reference counting upon the underlying socket. This talk walks through extensions to the verifier to perform tracking of references in a BPF program. This allows BPF developers to extend the UAPI with functions that allocate and release resources within the execution lifetime of a BPF program, and the verifier will validate that the resources are released exactly once prior to program completion.

Using this new reference tracking ability in the verifier, we add socket lookup and release function calls to the BPF API, allowing BPF programs to safely find a socket and build logic upon the presence or attributes of a socket. This can be used to load-balance traffic based on the presence of a listening application, or to implement stateful firewalling primitives to understand whether traffic for this connection has been seen before. With this new functionality, BPF programs can integrate more closely with the networking stack's understanding of the traffic transiting the kernel.

Presenter: STRINGER, Joe (Cilium)

Session Classification: Networking Track