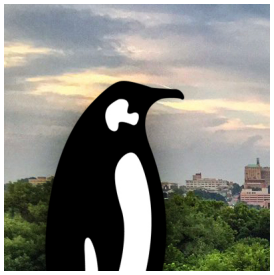

Contribution ID: **301** Type: **not specified**

Global BPF sockets iterator and rethinking conntrack in Cilium BPF datapath

This talk will cover two main topics around the complexities around network namespaces and rethinking some aspects of conntrack in the cilium BPF datapath in the context of cloud-native environments -

(1) Global sockets iterator and entering host-wide netns: Cilium has a few use cases where it needs to enter network namespaces on a host. There are certain challenges involved in containerised cloud-native environments like Kubernetes where it's not trivial for Cilium to gain access to other network namespaces. In general, there is no correlation between userspace (netns fd) and kernel network namespace constructs (netns cookie) that makes it difficult to tackle our use cases.
To that end, we propose a global sockets iterator in BPF to solve one of the use cases. The BPF sockets iterator is currently network namespace aware, and hence requires iterating over the sockets hash table in every network namespace. We'll discuss APIs and corresponding upstream kernel changes based on the initial discussions at LSF/MM/BPF '23. On a related note, it's worthwhile to discuss the kernel APIs involving namespace file descriptors [0], and circle back on some of the syscalls originally proposed in the patch set pre Kubernetes-era.

(2) Rethinking Conntrack: Cilium has a native connection tracking implementation in its BPF datapath for load-balancing and policy enforcement use cases. It follows a conventional approach where flows are tracked based on a 5-tuple metadata (source/destination {ip, port} and protocol). The main advantage of this map based approach is that it allows data sharing between TC and XDP programs. However, we've had various issues with this approach: LRU map scale issues at high packet rate, unbalanced load-balancing due to conntrack entry collisions, inefficient conntrack map lookups for hairpin traffic, out of band garbage collection of the conntrack map in userspace becomes complex.
We'll discuss the limitations with the existing approach, and alternate approach to some aspects of conntrack where we can store flow context in native BPF socket storage construct. The alternate approach also has some challenges with network namespaces as described in topic (1). We hope our experiences rethinking conntrack in the Cilium datapath drive discussions around kernel extensions in the upstream community, and help others who might be looking to adopt similar approaches.

[0] https://lwn.net/Articles/407495/

Primary author: GHAG, Aditi (Isovalent)

Presenter: GHAG, Aditi (Isovalent)

Session Classification: eBPF & Networking

Track Classification: eBPF & Networking Track