Network namespace unaware BPF sockets iterator

Aditi Ghag, Isovalent

Cilium use cases to iterate over host-wide sockets

- Cilium needs to iterate and filter client sockets (e.g., connected to deleted service backends) [1]

- Invoke bpf-sock-destroy from BPF iterator program

[1] https://lpc.events/event/16/contributions/1358/

Current: TCP/UDP sockets iterator match netns

```
static bool seq_sk_match(struct seq_file *seq, const struct sock *sk)
{
     unsigned short family = seq_file_family(seq);

     /* AF_UNSPEC is used as a match all */
     return ((family == AF_UNSPEC || family == sk->sk_family) &&
          net_eq(sock_net(sk), seq_file_net(seq)));
}
```

Inefficient to enter all network namespaces to retrieve host-wide sockets!

Proposed extensions to the tcp,udp iterators

- Override allow options

  (1) Host netns

  - Won't work for nested envs

  (2) SYS_CAP_NET

- Opt-in flag to override netns checks
  - Plumb global flag via `bpf_iter_attach_opts`:

```
union bpf_iter_link_info {
      struct {
          …
      } map;
      :
      struct {
          bool global
      } socket; (or should this be separate targets tcp, udp?)
};
```

  - Extend `struct bpf_iter_reg` for tcp and udp with `attach_targe`

    `and fill_link_info` callbacks

  - Extend `bpf_iter_aux_info` with `global` flag
  - Pass flag to tcp/udp iterator init callbacks via `bpf_iter_aux_info`