Suricata

- Network Traffic Analysis
- Intrusion Detection System
- Network Security Monitoring
- Community driven
- Code:
  - Open Source
  - Mixed of C and Rust
Suricata on live traffic

- Sniffer mode
- Get raw traffic
- Use AF_PACKET socket
- Like tcpdump

```
suricata -af-packet=$(wtfiseth0) port 80 and host 10.0.0.1
```

Note: alias wtfiseth0='ip r l | grep default | cut -d " " -f 5'
Suricata on selected live traffic (hipster style)

- BPF is your grand mother technology
- Linux has eBPF
  - Extended Berkeley Packet Filter
  - Filters can be developed in C
- Shared data structure
  - Between kernel and user space
  - Data can be managed by external tool
Simple eBPF filter

```c
#ifdef LINUX_VERSION_CODE 263682

struct bpf_map_def SEC("maps") ipv4_drop = {
    .type = BPF_MAP_TYPE_PERCPU_HASH,
    .key_size = sizeof(__u32),
    .value_size = sizeof(__u32),
    .max_entries = 32768,
};

struct vlan_hdr {
    __u16   h_vlan_TCI;
    __u16   h_vlan_encapsulated_proto;
    __u32   *value;
    __u32   ip = 0;
}

nhoff = skb->cb[0];

ip = load_word(skb, nhoff + offsetof(struct iphdr, saddr));
value = bpf_map_lookup_elem(&ipv4_drop, &ip);
if (value) {
    *value = *value + 1;
    return 0;
}

ip = load_word(skb, nhoff + offsetof(struct iphdr, daddr));
value = bpf_map_lookup_elem(&ipv4_drop, &ip);
if (value) {
```

Introducing bpfctrl

- Wrapper on top of bpftool
- LGPLv2
- Manage pinned maps
- Currently support
  - IPv4 with counters
  - Single integer
Demonstration

- Verify Suricata sees traffic
- Add IP to drop list
- Nothing anymore
THANK YOU!

https://github.com/stamusNetworks/bpfctrl

Eloïse Brocas
eloise@brocas.org

Éric Leblond
el@stamus-networks.com