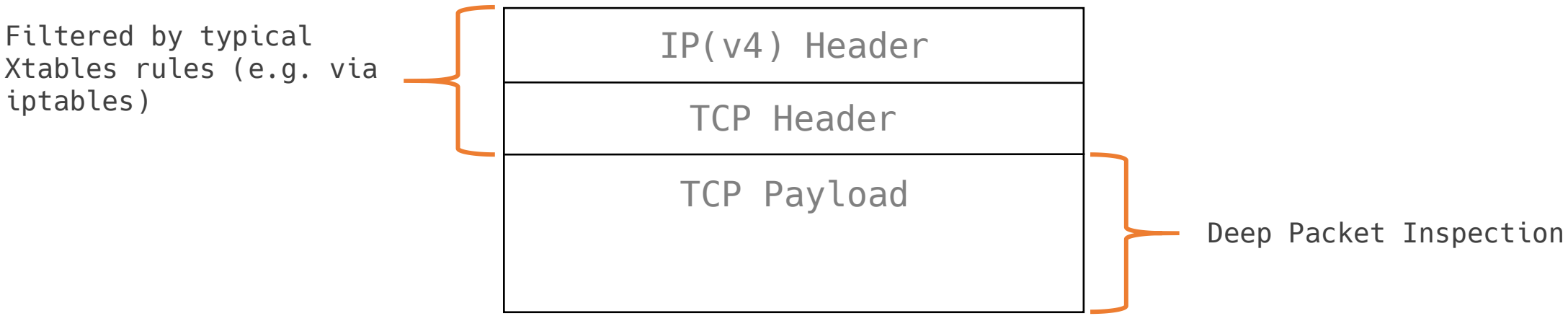
Deep Packet Inspection using Linux Netfilter

Leonard Seibold

Deep Packet Inspection

- concept of filtering Packets processed by the Operating System by their payload before they get passed along
- enables advanced firewalls that are able to analyze payload data, not just meta information from IP/TCP/UDP Headers
- firewall rules often application layer protocol specific



The Linux Netfilter Framework

- allows Loadable Kernel Modules (LKMs) to register hooks that get called when a packet is processed
- those hooks can then decide whether to accept or to drop the packet
- existing firewalls are built on this framework (iptables, nf_tables,...)

```
unsigned int hook_func(void *priv,
                       struct sk_buff *skb,
                       const struct nf_hook_state *state) {
    struct iphdr *iph;
    struct tcphdr *tcph;

    iph = ip_hdr(skb);
    if (iph->protocol == IPPROTO_TCP) {
        tcph = tcp_hdr(skb);
        // [...] Inspect packet payload
        return NF_DROP;
    }
     return NF_ACCEPT;
}
```

Project Goals

- Create a LKM that uses the Netfilter Framework to perform Deep Packet Inspection on an example protocol
- Possible Questions to investigate
  - Can certain attacks (e.g. Application Layer DoS) be mitigated?
  - How practicable is DPI for realtime applications where every milisecond counts?
  - Can we move the actual inspection code to user space for easier extensibility?