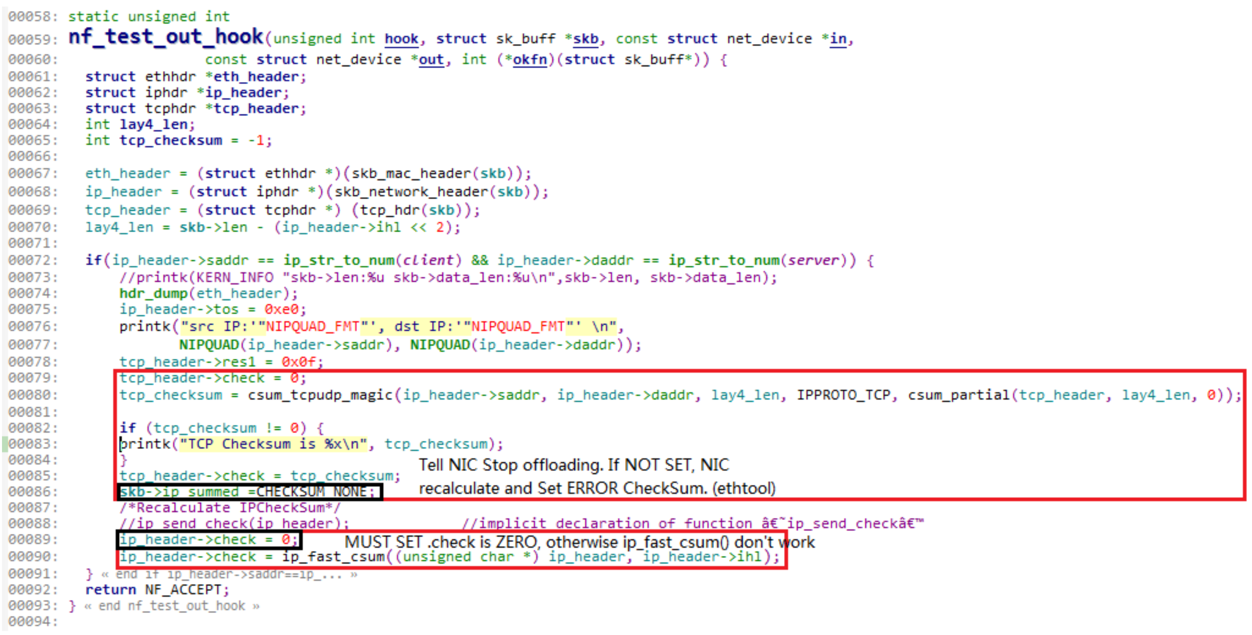
# How to Write a Netfilter Module

1. Experiments Environments: SmartOS KVM Virtual Machines.
2. All codes are in Netfilter_TCPIPChecksum (https://github.com/jinhao2826/Netfilter_TCPIPChecksum.git)
3. The most important is to EXPAIN following Codes about CHECKSUM OF TCP AND IP



```
00058: static unsigned int
00059: nf_test_out_hook(unsigned int hook, struct sk_buff *skb, const struct net_device *in,
00060:                  const struct net_device *out, int (*okfn)(struct sk_buff*)) {
00061:   struct ethhdr *eth_header;
00062:   struct iphdr *ip_header;
00063:   struct tcphdr *tcp_header;
00064:   int lay4_len;
00065:   int tcp_checksum = -1;
00066:
00067:   eth_header = (struct ethhdr *)(skb_mac_header(skb));
00068:   ip_header = (struct iphdr *)(skb_network_header(skb));
00069:   tcp_header = (struct tcphdr *) (tcp_hdr(skb));
00070:   lay4_len = skb->len - (ip_header->ihl << 2);
00071:
00072:   if(ip_header->saddr == ip_str_to_num(client) && ip_header->daddr == ip_str_to_num(server)) {
00073:       //printk(KERN_INFO "skb->len:%u skb->data_len:%u\n",skb->len, skb->data_len);
00074:       hdr_dump(eth_header);
00075:       ip_header->tos = 0xe0;
00076:       printk("src IP:'"NIPQUAD_FMT"', dst IP:'"NIPQUAD_FMT"' \n",
00077:              NIPQUAD(ip_header->saddr), NIPQUAD(ip_header->daddr));
00078:       tcp_header->res1 = 0x0f;
00079:       tcp_header->check = 0;
00080:       tcp_checksum = csum_tcpudp_magic(ip_header->saddr, ip_header->daddr, lay4_len, IPPROTO_TCP, csum_partial(tcp_header, lay4_len, 0));
00081:
00082:       if (tcp_checksum != 0) {
00083:       printk("TCP Checksum is %x\n", tcp_checksum);
00084:       }
00085:       tcp_header->check = tcp_checksum;
00086:       skb->ip_summed =CHECKSUM_NONE;
00087:       /*Recalculate IPCheckSum*/
00088:       //ip_send_check(ip_header);          //implicit declaration of function â€˜ip_send_checkâ€™
00089:       ip_header->check = 0;
00090:       ip_header->check = ip_fast_csum((unsigned char *) ip_header, ip_header->ihl);
00091:   } « end if ip_header->saddr==ip_... »
00092:   return NF_ACCEPT;
00093: } « end nf_test_out_hook »
00094:
```

Tell NIC Stop offloading. If NOT SET, NIC recalculate and Set ERROR CheckSum. (ethtool)

MUST SET .check is ZERO, otherwise ip_fast_csum() don't work

A. Line75: modify the tos field (dscp) to 0xe0
B. Line78: modify the res1 field in TCP header
C. MUST remember that first step to calculate the TCP and UDP pseudoheader, then calculate IP checksum because Pseudoheader contains IP saddr and daddr.

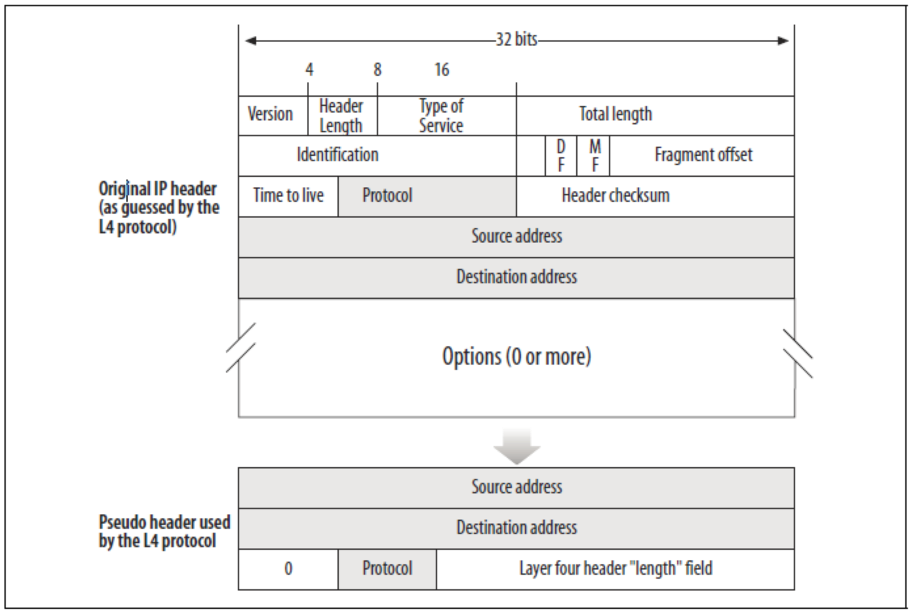


*Figure 18-14. Pseudoheader used by TCP and UDP while computing the checksum*

D. ip_fast_csum(): Given an IPheader and length, computes and returns the IPchecksum. It can be used both to validate an input packet and to compute the checksum of an outgoing packet. You can consider ip_fast_csum a variation of ip_compute_csum optimized for IP headers.

E. csum_tcpudp_magic(): Compute the checksum on the TCP and UDP pseudoheader (see Figure 18-14).

Reference: Understanding Linux Network Internals Part: Checksums Page:434