

Programming assignment #2

CprE 530 Due Thursday November 9

Do programming problem number 1 from chapters 5, 6 & 7 in the book.

These have you print out Ethernet, IP, TCP, and UDP headers

NOTES:

Use the Kali VM you were provided.

If you are doing the programming assignment turn in a printout of a sample trace.

Notes:

For C and UNIX help see: <http://www.dougi.net/modules/index.html>

There are two places you need to add code:

First is in this routine:

```
void raw_print(u_char *user, const struct pcap_pkthdr *h, const
u_char *p)
{
    u_int length = h->len;
    u_int caplen = h->caplen;

    default_print(p, caplen);
}
```

The routine default_print prints the packet out in hex. The packet is contained in the array p where p[0] is the first byte of the HW destination address. The variable caplen is the length of the packet. To get you started this line of code will print the HW destination address out:

```
printf("DEST Address = %02X:%02X:%02X:%02X:%02X:%02X\n",
    p[0],p[1],p[2],p[3],p[4],p[5],p[6]);
```

This will print out the ethernet type:

```
uint16_t e_type; // this goes after u_int caplen = h->caplen;

e_type = ntohs((uint16_t) * &p[12]);
printf("E_Type = %04X ", e_type);
if (e_type == 0x800) printf(" -> IP\n");
if (e_type == 0x806) printf(" -> ARP\n");
```

The other routine you will need to modify is `program_ending`. This routine is called when you stop the program. So for example to print out the number of IP packets you would declare a global variable called `num_ip_packets` and in the `raw_print` routine you would increment the variable by one each time you saw an IP packet.

```
/* routine is executed on exit */
void program_ending(int signo)
{
    struct pcap_stat stat;

    if (pd != NULL && pcap_file(pd) == NULL) {
        (void)fflush(stdout);
        putc('\n', stderr);
        if (pcap_stats(pd, &stat) < 0)
            (void)fprintf(stderr, "pcap_stats: %s\n",
                           pcap_geterr(pd));
        else {
            (void)fprintf(stderr, "%d packets received
by filter\n",
                           stat.ps_recv);
            (void)fprintf(stderr, "%d packets dropped
by kernel\n",
                           stat.ps_drop);
        }
    }
    exit(0);
}
```