**NATIONAL UNIVERSITY**
**of Computer & Emerging Sciences, Lahore**

Department of Computer Science

# CS411 – Network Security
**May 31st 2020**
**Assignment 4 – Weight 8 %**
**Group assignment**

1. Make groups of 2 (from the same section). You can retain the same group members from the previous assignment or make new ones. You are responsible for the groups dynamics therefore select your group partner wisely

2. Read the instructions provided in “Attacks on the TCP Protocol” (Chapter 13 (or 16 in the newer version) of A Hands-on Approach by Wenliang Du). Pdf of Chapter 16 from the newer version is **available on SLATE**.

3. **Launch the SYN Flooding attack (section 13.2.4)** using the VM from SEED LABS and your own code (screenshot attached below). Summarize the countermeasures presented in 13.2.5. (2 points)

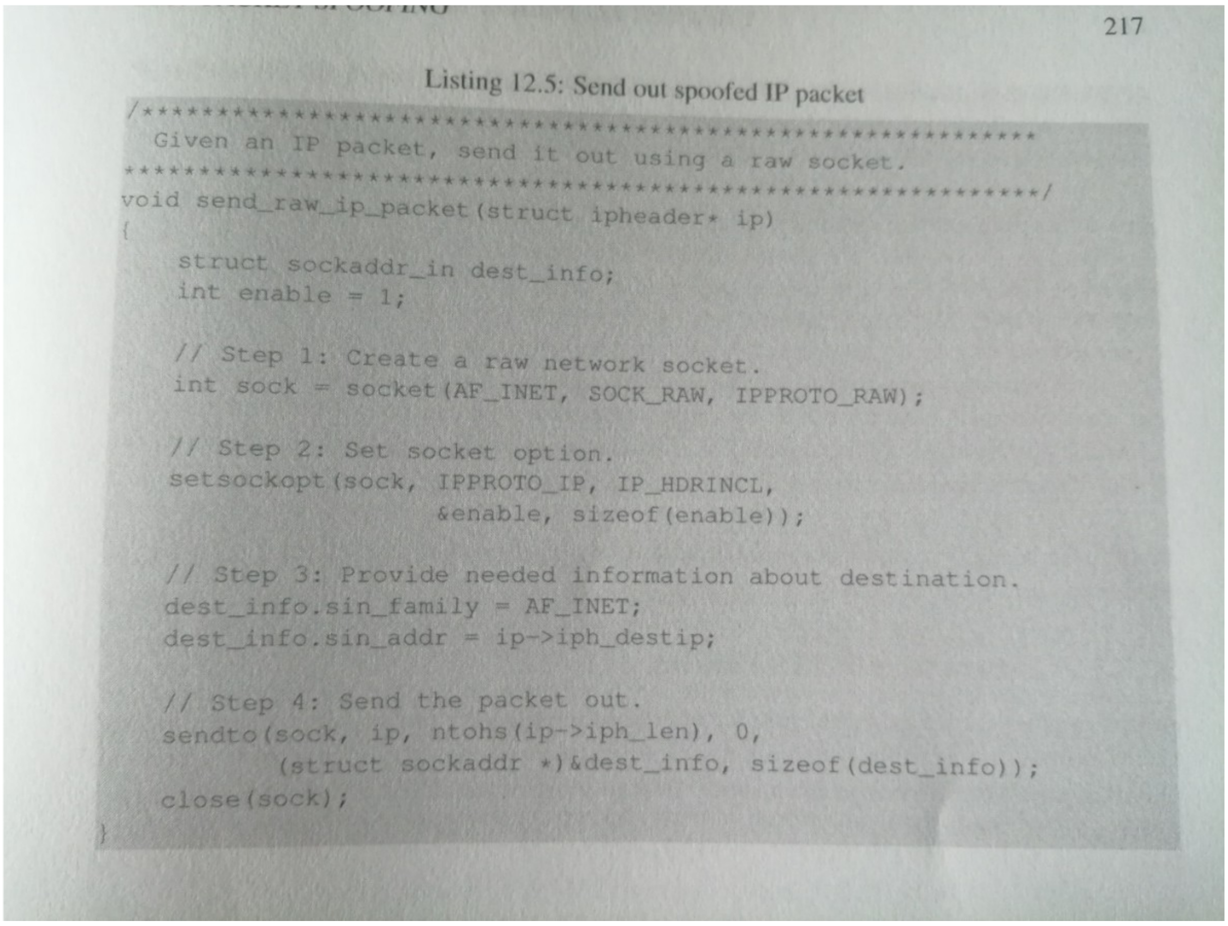
4. **Launch the TCP Reset attack (13.3) on telnet (13.3.4), SSH (13.3.5) and video streaming (13.3.6)** connections. **Discuss the differences** in all three attacks. **Why** are there differences? (3 points)

5. Launch the **TCP Session Hijacking Attack (13.4)**. Complete all the steps through section 13.4.5. (3 points)

6. Prepare a word document **reporting all the steps carried out**. Use screen shots to show your actions and their results. For each attack, **half the points are for successfully carrying out the attack and the other half are for properly summarizing, discussing, and documenting the steps.**

Notes
1. Names of both group members should be visible on every screen shot.
2. Title page of the assignment should clearly mention the names and IDs of both groups members
3. You are also allowed to collaborate with you friends as much as you like. However, copying their screen shots is .
4. Since we are at the end of the semester, the deadline is **FIRM**.

Deadline to submit: Sunday June 10th, 11:55 PM via SLATE.



Listing 12.5: Send out spoofed IP packet

```
/**************************************************************
  Given an IP packet, send it out using a raw socket.
**************************************************************/
void send_raw_ip_packet(struct ipheader* ip)
{
    struct sockaddr_in dest_info;
    int enable = 1;

    // Step 1: Create a raw network socket.
    int sock = socket(AF_INET, SOCK_RAW, IPPROTO_RAW);

    // Step 2: Set socket option.
    setsockopt(sock, IPPROTO_IP, IP_HDRINCL,
                     &enable, sizeof(enable));

    // Step 3: Provide needed information about destination.
    dest_info.sin_family = AF_INET;
    dest_info.sin_addr = ip->iph_destip;

    // Step 4: Send the packet out.
    sendto(sock, ip, ntohs(ip->iph_len), 0,
           (struct sockaddr *)&dest_info, sizeof(dest_info));
    close(sock);
}
```

**TCP and UDP checksum.** According to RFC 768 [Postel, 1980] and RFC 793 [Postel, 1981], TCP and UDP checksum is the 16-bit one's complement of the one's complement sum of a pseudo header, which consists of the information from IP header, TCP/UDP header, and data, padded with zero octets at the end (if necessary) to make a multiple of two octets. To compute the checksum, we need to first construct a pseudo header, and then use the above `in_cksum()` function to compute its checksum. The following example shows how to compute the checksum for TCP packets.

Listing 12.10: Calculating TCP Checksum

```
/*****************************************************************
  TCP checksum is calculated on the pseudo header, which includes
  the TCP header and data, plus some part of the IP header.
  Therefore, we need to construct the pseudo header first.
*****************************************************************/

/* Psuedo TCP header */
struct pseudo_tcp
{
        unsigned saddr, daddr;
        unsigned char mbz;
        unsigned char ptcl;
        unsigned short tcpl;
        struct tcpheader tcp;
        char payload[PACKET_LEN];
};
```

```
unsigned short calculate_tcp_checksum(struct ipheader *ip)
{
   struct tcpheader *tcp = (struct tcpheader *)((u_char *)ip +
                             sizeof(struct ipheader));

   int tcp_len = ntohs(ip->iph_len) - sizeof(struct ipheader);

   /* pseudo tcp header for the checksum computation */
   struct pseudo_tcp p_tcp;
   memset(&p_tcp, 0x0, sizeof(struct pseudo_tcp));

   p_tcp.saddr  = ip->iph_sourceip.s_addr;
   p_tcp.daddr  = ip->iph_destip.s_addr;
   p_tcp.mbz    = 0;
   p_tcp.ptcl   = IPPROTO_TCP;
   p_tcp.tcpl   = htons(tcp_len);
   memcpy(&p_tcp.tcp, tcp, tcp_len);

   return  (unsigned short) in_cksum((unsigned short *)&p_tcp,
                                     tcp_len + 12);
}
```

## 12.7 Summary