4.4 패킷 스푸핑

- 패킷의 일부 중요한 정보가 위조되면 이를 패킷 스푸핑(packet spoofing)이라고 함.
- 패킷 스푸핑에는 두 가지 주요 단계:
  - **패킷 구성하기**
  - **패킷 내 보내기**
- 많은 네트워크 공격은 패킷 스푸핑에 의존.
- 스푸핑 없이 패킷을 보내는 방법(udp_client.c).

UDP 소켓을 이용한 일반 패킷 전송

```c
#include <unistd.h>
// POSIX 운영 체제 API 접근을 위한 헤더 파일
#include <stdio.h>
// 표준 입출력 함수들을 위한 헤더 파일
#include <string.h>
// 문자열 처리 함수들을 위한 헤더 파일
#include <sys/socket.h>
// 소켓 프로그래밍을 위한 헤더 파일
#include <netinet/ip.h>
// 인터넷 프로토콜(IP) 관련 구조체를 위한 헤더 파일
#include <arpa/inet.h>
// 인터넷 주소 변환 함수를 위한 헤더 파일

void main()
{
  struct sockaddr_in dest_info;
  // 목적지 주소 정보를 저장할 구조체

  char *data = "UDP message\n";
  // 전송할 데이터를 저장할 문자열

  // Step 1: Create a network socket
  /*
  IPv4 주소 체계(AF_INET)와 UDP 소켓(SOCK_DGRAM)을 사용하여 소켓을 생성.
  IPPROTO_UDP는 UDP 프로토콜을 의미
  */
  int sock = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);

```

udp_client.c 설명

일반 패킷 전송에서는 IP헤더를 직접 설정 또는 변경할 수 없음.

- 테스트: netcat(nc) 명령을 사용하여 10.9.0.5에서 UDP 서버를 실행.
- 그런 다음 다른 컴퓨터에서 왼쪽의 프로그램을 실행.
- 메시지가 서버 컴퓨터로 전달된 것을 확인:

```
seed@Server(10.0.2.5):$ nc -luv 9090
Connection from 10.0.2.6 port 9090 [udp/*] accepted
UDP message
```

원시 소켓을 이용하여 스푸핑된 UDP 패킷 보내기

udp_spoof.c 설명

Open udpspoof.c ~/Downloads/ch04_lab/volumes

```
 98 void send_raw_ip_packet(struct ipheader* ip)
 99 {
100   struct sockaddr_in dest_info;
101   int enable = 1;
102   // 로우 소켓을 생성
103   int sock = socket(AF_INET,SOCK_RAW,IPPROTO_RAW);
104
105   // 소켓 옵션을 설정하는 함수인 setsockopt을 사용하여 소켓의 동작 방식을 제어.
106   // 여기서 이 호출은 소켓 옵션 중 하나인 IP_HDRINCL 옵션을 설정.
107   setsockopt(sock, IPPROTO_IP, IP_HDRINCL, &enable, sizeof(enable));
108   // (1) socket: 옵션을 설정할 소켓의 파일 디스크립터
109   // (2) level: 옵션의 프로토콜 레벨 (예: SOL_SOCKET, IPPROTO_IP, IPPROTO_TCP 등)
110   // (3) option_name: 설정할 옵션의 이름
111   // (4) option_value: 옵션 값에 대한 포인터
112   // (5) option_len: 옵션 값의 크기
113   /*
114   IP_HDRINCL 옵션은 IP 레벨에서 사용,
115  이 옵션을 설정하면 소켓을 통해 전송되는 패킷의 IP 헤더를 애플리케이션이 직접 구성.
116  기본적으로 소켓을 사용하여 데이터를 전송할 때, IP 헤더는 운영체제가 자동으로 생성.
117  그러나 IP_HDRINCL 옵션을 활성화하면 애플리케이션이 IP 헤더를 직접 생성하고 구성.
118  여기서는 level 매개변수로 IPPROTO_IP를 사용하여 IP 프로토콜 레벨에서 설정
119
120   IP_HDRINCL 옵션을 설정하면 IP 헤더를 올바르게 구성해야 하며,
121  잘못된 헤더를 구성하면 패킷 전송이 실패하거나 네트워크 문제를 일으킬 수 있음.
122  또한, 로우 소켓을 사용하여 IP 패킷을 직접 전송하는 행위는 보안 문제를 일으킬 수 있으므로,
123  적절한 권한이 필요하며 네트워크 관리자의 승인이 필요.
```

원시 소켓을 이용하여 스푸핑된 ICMP 패킷 보내기

spoof_icmp.c 설명

```c
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>

// 패킷의 최대 길이를 1500 바이트로 정의
#define PACKET_LEN   1500
// 위조할 출발지 IP 주소를 "1.2.3.4"로 정의
#define SRC_IP   "1.2.3.4"
// 목적지 IP 주소를 "10.9.0.5"로 정의
#define DEST_IP  "10.9.0.5"

/* IP Header */
// IP 헤더의 각 필드를 정의하는 구조체
struct ipheader {
        unsigned char      iph_ihl:4, iph_ver:4; //IP Header length & Version.
        unsigned char      iph_tos; //Type of service
        unsigned short int iph_len; //IP Packet length (Both data and header)
        unsigned short int iph_ident; //Identification
        unsigned short int iph_flag:3, iph_offset:13; //Flags and Fragmentation offset
        unsigned char      iph_ttl; //Time to Live
        unsigned char      iph_protocol; //Type of the upper-level protocol
        unsigned short int iph_chksum; //IP datagram checksum
        struct  in_addr    iph_sourceip; //IP Source address (In network byte order)
        struct  in_addr    iph_destip;//IP Destination address (In network byte order)
};

```

실습

- UDP 소켓을 이용한 일반 패킷 전송
- 원시 소켓을 이용하여 스푸핑된 UDP 패킷 보내기
- 원시 소켓을 이용하여 스푸핑된 ICMP 패킷 보내기