

```

#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <string.h>
#include <fcntl.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <linux/in.h>
#include <linux/ip.h>
#include <linux/udp.h>
struct my_packet {
    struct iphdr ip;
    struct udphdr udp;
    unsigned char data[8];
};

unsigned short ip_cksum(unsigned short *buff, int len);
unsigned short udp_check(struct udphdr *th, unsigned short len,
    unsigned long saddr, unsigned long daddr);
unsigned long syn_daddr;
unsigned long syn_saddr;
int sockfd;

int main()
{
    struct my_packet out;
    unsigned char in[65000];
    int size;
    unsigned short local_port = htons(3000);
    unsigned short syn_port = htons(2000);
    struct sockaddr_in to={AF_INET,2000},from={AF_INET,3000};
    int one=1;
    syn_daddr = inet_addr("127.0.0.1");
    syn_saddr = inet_addr("127.0.0.1");
    to.sin_addr.s_addr=syn_daddr;
    from.sin_addr.s_addr=syn_saddr;

    if((sockfd = socket(AF_INET, SOCK_RAW, IPPROTO_UDP)) < 0)
    {
        printf("socket creation error\n");
        exit(-1);
    }
    bind(sockfd, (struct sockaddr *)&from, sizeof(from));
    bzero(&out, sizeof(out));
    strcpy(out.data, "testing");
    out.ip.ihl = 5;
    out.ip.version = 4;
    out.ip.tos = 0;
    out.ip.tot_len = htons(sizeof(out));
    out.ip.id = getpid();
    out.ip.frag_off = 0;
    out.ip.ttl = 255;
    out.ip.protocol = IPPROTO_UDP;
    out.ip.saddr = syn_saddr;
    out.ip.daddr = syn_daddr;
    out.ip.check = 0;

```

```

out.ip.check = ip_cksum( (unsigned short *) &out.ip, 20);
out.udp.source = local_port;
out.udp.dest = syn_port;
out.udp.len = htons(sizeof(struct udphdr)+8);
out.udp.check = 0;
out.udp.check = udp_check(&out.udp, sizeof(struct udphdr)+8 ,
                           out.ip.saddr, out.ip.daddr);
if(setsockopt(sockfd, IPPROTO_IP, IP_HDRINCL, &one, sizeof(one)))
    printf("setsockopt error\n");
if (sendto(sockfd, &out, sizeof(out), 0, (struct sockaddr *) &to,
           sizeof(to)) < 0)
    printf("sendto error\n");
printf("%d bytes packet sent to ...\n", sizeof(out));
if(recvfrom(sockfd, &in, 4096, 0, (struct sockaddr *) &from, &size) < 0)
    printf("recvfrom error \n");

return 0;
}
unsigned short ip_cksum(unsigned short *buff, int len)
{
    unsigned long sum = 0;
    while(len > 1)
    {
        sum += *buff++;
        len -= 2;
    }
    if (len == 1)
        sum += (*buff & 0xff);
    sum = (sum >> 16) + (sum & 0xffff);
    sum += (sum >> 16);
    sum = ~sum;
    return (sum & 0xffff);
}
unsigned short udp_check(struct udphdr *th, unsigned short len,
                        unsigned long saddr, unsigned long daddr)
{
    unsigned long sum = 0;
    unsigned short *buff;
    buff = (unsigned short *) &saddr;
    sum += *buff++;
    sum += *buff;
    buff = (unsigned short *) &daddr;
    sum += *buff++;
    sum += *buff;
    sum += IPPROTO_UDP * 256;
    sum += htons(len) & 0xffff;
    buff = (unsigned short *) th;
    while(len > 1) {
        sum += *buff++;
        len -= 2;
    }
    if (len == 1)
        sum += (*buff & 0xff);
    sum = (sum >> 16) + (sum & 0xffff);
    sum += (sum >> 16);
    return ( (~sum) & 0xffff);
}

```