Problemas de Redes de Computadores.
Ingeniería Técnica en Informática de Gestión
Conjunto de problemas 2

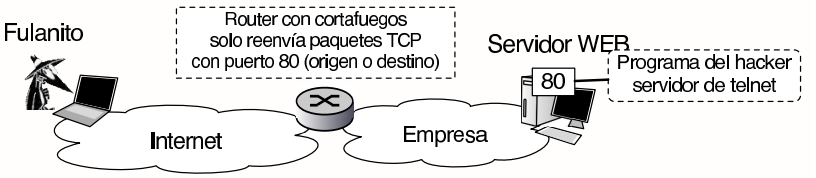
Pregunta 2.1: Fulanito el hacker consigue infiltrarse en el servidor Web de una empresa detrás de un cortafuegos que sólo deja pasar los paquetes que tienen puerto origen 80 o puerto destino 80. Para dejarse una puerta abierta, instala un servidor de telnet que escucha en el puerto 80 de ese ordenador. Pero cuando intenta conectarse a su servidor de telnet no le funciona... ¿Por qué no funciona?

a) Porque telnet sólo puede funcionar en el puerto 23 como manda el RFC-854

b) Porque los usuarios remotos no sabrán que el servidor está en el puerto 80

c) Porque no se puede tener dos aplicaciones TCP escuchando en el puerto 80

d) Porque telnet usa UDP y el cortafuegos sólo deja pasar los paquetes con el puerto 80 TCP



Pregunta 2.2: El hacker de la pregunta 2.1 quiere utilizar el servidor SMTP de la empresa en la que ha conseguido infiltrarse. Para ello pretende escribir un programa llamado `redirect`. El programa en cuestión utilizara 2 sockets, en uno de ellos (`sock_in`) escuchara una conexión en el puerto 80 que puede atravesar el cortafuegos y cuando la reciba establecerá una conexión desde el otro socket (`sock_out`) al servidor SMTP de la empresa. A partir de este punto reenviará lo que llega por cada conexión a la otra de forma que el programa externo pueda hablar con el servidor SMTP. Empieza a escribir el programa con la inicialización de los sockets

```
int main(int argc, char *argv[]) {
    int sock_in, sock_out;
    struct sockaddr_in dir;
    int ip_servidor_smtp;

    sock_in = socket(PF_INET,SOCK_STREAM,0);
    sock_out = socket(PF_INET,SOCK_STREAM,0);

    dir.sin_family=AF_INET;
    dir.sin_port=htons(80);
    dir.sin_addr.s_addr=htonl(INADDR_ANY);
    if ( bind( sock_in,(struct sockaddr*)&dir,sizeof(dir)) == 0 )
        sal_con_error("No puedo coger el puerto 80");

   ....
```

Señale los errores presentes en el programa

a) El socket sock_out al ser un socket cliente debe construirse con SOCK_DGRAM en lugar de SOCK_STREAM

b) dir.sin_family debe ser PF_INET al igual que el valor pasado al socket

c) El error se producirá cuando bind devuelve -1 en lugar de 0

d) En dir.sin_addr.s_addr hay que poner la dirección a la que queremos conectarnos es decir la dirección IP del servidor SMTP

e) El puerto 80 hay que convertirlo con ntohs() en lugar de con htons() porque queremos convertir del formato de red al de host

f) No hay errores

Pregunta 2.3: Sigue escribiendo el programa... (se supone que añade definiciones de variables al principio si es necesario)

```
    listen(sock_in,5);

    len=sizeof(dir);
    s1=accept(sock_in,(struct sockaddr *)&dir,&len);
    if (s1==-1) {
        printf("...");
        exit(-1);
    }
```

¿Qué error debería indicar en el prinf()?

a) El otro extremo ha rechazado la conexión

b) El puerto 80 no puede ser utilizado sin permiso de administrador

c) La conexión no ha conseguido atravesar el router con cortafuegos

d) Da igual, ese error no ocurre nunca... no merece la pena ni comprobarlo

Pregunta 2.4: Sigue escribiendo el programa...

```
ip_servidor_smtp=inet_addr("88.27.1.2");
dir.sin_port=htons(25);
dir.sin_addr.s_addr=ip_servidor_smtp;

s2=connect(sock_out,(struct sockaddr *)&dir,sizeof(dir));
if (s2==-1) {
    printf("Conexión rechazada por el servidor SMTP");
    exit(-1);
}
```

¿Qué errores ha cometido ahora?

a) El htons() tiene que ser ntohs()

b) La direección 88.27.1.2 no puede ser convertida con inet_addr sino que hay que usar gethostbyname(). La linea sería ip_servidor_smtp=gethostbyname("88.27.1.2");

c) connect() no devuelve nunca -1 en realidad, la conexion es aceptada y posteriormente cuando vaya a leer del socket dará error el write()

d) No hay ningun error en ese trozo

Pregunta 2.5: Finalmente escribe un bucle en el que se lee lo que llega de cada socket y se reenvia por el otro

```
while (1) {
    FD_ZERO(&aleer);
    FD_SET(sock_in,&aleer);
    FD_SET(sock_out,&aleer);
    select(10,&aleer,NULL,NULL,NULL);
    if ( FD_ISSET(sock_in,&aleer) ) {
        len=read(sock_in,buf,5000);
        if (len>0) write(sock_out,buf,5000);
    }
    if ( FD_ISSET(sock_out,&aleer) ) {
        len=read(sock_out,buf,5000);
        if (len>0) write(sock_in,buf,len);
    }

}
```

¿Qué errores hay ahora en el último trozo de código?

a) El 10 del select deberií ser un 2

b) Todas las veces que aparece sock_in debería aparecer en su lugar s1

c) El primer write como tercer argumento tiene 5000 y debería ser len

d) El segundo write como tercer argumento tiene len y debería ser 5000

e) El conjunto de ficheros se vacía cada vez con FD_ZERO y sería mejor que se vaciara sólo una vez al principio pero fuera del bucle

f) No hay ningún error en ese codigo

Pregunta 2.6: Una vez terminado el programa el hacker se pregunta como puede usarlo dentro de la red de la empresa. Señale cuales son ciertas

a) Cada vez que se lance el programa redirect acepta una conexión única por el puerto 80 y la redirige al servidor SMTP. Para redirigir otra conexión hace falta matar el programa y lanzarlo otra vez

b) Cada vez que se lance el programa redirect acepta una conexión única por el puerto 80 y la redirige al servidor SMTP. Pero puedo redirigir varias conexiones desde el puerto 80 si lanzo el programa varias veces en el mismo ordenador

c) El programa tiene un bucle while que le permite redirigir simultaneamente varias conexiones. No es problema redirigir simultáneamente varias conexiones porque TCP permite que haya varias conexiones a un mismo puerto destino. De lo contrario no funcionarían los servidores web

d) El programa tiene un bucle que le permite redirigir varias conexiones pero NO SIMULTANEAMENTE. Cuando una conexion finaliza volvemos al accept que vuelve a aceptar la siguiente conexión

Pregunta 2.7: Pero el servidor SMTP de la empresa guarda registro de los clientes que se han conectado al servidor SMTP para enviar correo. ¿Cómo quedan registrados los envíos del hacker? ¿Qué dirección IP será registrada como cliente de SMTP cuando utilizo el programa redirect?

a) La dirección IP del ordenador del hacker H1 por lo que pueden detectar que alguien está enviando desde fuera

b) La dirección IP del router R1 en el exterior

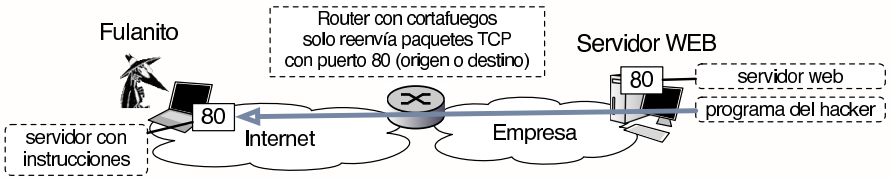
c) La dirección IP del router R1 en la red interna de la empresa, que es la más cercana al servidor y por tanto la que ha introducido el paquete en la red de la empresa

d) La dirección IP del ordenador en que corre redirect H2

e) La propia dirección IP del servidor SMTP H3 debido al reflejo de la conexión a través redirect

Problema 2.8: El host H1 se encuentra en una red en la que se filtran los paquetes que entran y sólo se permiten paquetes de conexiónes Web. El propietario de H1 quiere utilizar un programa peer-to-peer que utiliza normalmente el puerto TCP 6881 (aunque puede configurarse otro) pero no le funciona porque se eliminan los paquetes al no ser de conexiones web. ¿Cómo puede lograr que funcione el programa a pesar del filtro? ¿Puede conseguirlo si en H1 tiene activado el servidor Web? ¿Cómo cambia esto si el programa peer-to-peer utiliza UDP?

Problema 2.9: En una universidad el servidor oficial de correo se encuentra en el servidor S1. El administrador de la red de la universidad intenta evitar que se utilicen otros servidores de correo distintos de S1 en su red. Para ello, dado que el router de salida R1 tiene funcionalidades de firewall y permite aplicar reglas sobre los paquetes, añade una regla: R1 no reenviará paquetes TCP al exterior si tienen el puerto destino 25, salvo si su dirección IP origen es S1. ¿Evitará esto el uso de otros servidores?. Si un usuario del departamento B coloca un servidor de SMTP en un H3. Puede utilizarlo para enviar correo fuera sin usar el servidor S1? ¿Puede usarlo para recibir correo sin usar el servidor S1? Razone las respuestas. ¿Puede un usuario de la red B consultar su cuenta de correo de un servidor externo?¿Por que?



Pregunta 2.10: Fulanito el hacker consigue infiltrarse en el servidor Web de una empresa detrás de un cortafuegos que sólo deja pasar los paquetes que tienen puerto origen 80 o puerto destino 80. Para dejarse una puerta abierta, instala un programa que cada cierto tiempo establece una conexión con el puerto 80 de un servidor externo controlado por él y se descarga instrucciones por HTTP ¿Qué problema tiene esto?

a) Que HTTP no puede funcionar en el puerto 80 porque el puerto está reservado para la web

b) No tiene ningún problema y debería funcionar

c) Que no se puede tener una conexión TCP al puerto 80 y a la vez escuchar conexiones en el puerto 80

d) Que HTTP usa UDP y el cortafuegos sólo deja pasar los paquetes con el puerto 80 TCP

Problema 2.11: ¿Cuáles de estas funciones provoca el envío de algún paquete a la red?

`socket()` `connect()` `recvfrom()` `sendto()` `bind()` `listen()` `accept()`

Problema 2.12: ¿Como se detecta desde un programa que los datos entregados a un socket TCP pueden haber sufrido errores y no ser correctos?

Problema 2.13: La siguiente traza ha sido capturada en la red de la universidad

```
 1 0.963491 IP 130.206.168.45.60905 > 193.252.23.108.110: S 2357731200:2357731200(0) win 65535
 2 0.964072 IP 193.252.23.108.110 > 130.206.168.45.60905: S 1061601894:1061601894(0) ack 2357731201 win 5792
 3 0.964129 IP 130.206.168.45.60905 > 193.252.23.108.110: . ack 1061601895 win 65535
 4 1.111168 IP 193.252.23.108.110 > 130.206.168.45.60905: P 1061601895:1061601927(32) ack 2357731201 win 5792
 5 1.111753 IP 130.206.168.45.60905 > 193.252.23.108.110: P 2357731201:2357731218(17) ack 1061601927 win 65535
 6 1.112349 IP 193.252.23.108.110 > 130.206.168.45.60905: . ack 2357731218 win 5792
 7 1.200422 IP 193.252.23.108.110 > 130.206.168.45.60905: P 1061601927:1061601956(29) ack 2357731218 win 5792
 8 1.200834 IP 130.206.168.45.60905 > 193.252.23.108.110: P 2357731218:2357731232(14) ack 1061601956 win 65535
 9 1.201287 IP 193.252.23.108.110 > 130.206.168.45.60905: . ack 2357731232 win 5792
10 1.711614 IP 193.252.23.108.110 > 130.206.168.45.60905: P 1061601956:1061601991(35) ack 2357731232 win 5792
11 1.712040 IP 130.206.168.45.60905 > 193.252.23.108.110: P 2357731232:2357731238(6) ack 1061601991 win 65535
12 1.712630 IP 193.252.23.108.110 > 130.206.168.45.60905: . ack 2357731238 win 5792
13 1.861177 IP 193.252.23.108.110 > 130.206.168.45.60905: P 1061601991:1061602000(9) ack 2357731238 win 5792
14 1.861596 IP 130.206.168.45.60905 > 193.252.23.108.110: P 2357731238:2357731244(6) ack 1061602000 win 65535
15 1.862059 IP 193.252.23.108.110 > 130.206.168.45.60905: . ack 2357731244 win 5792
16 2.064350 IP 193.252.23.108.110 > 130.206.168.45.60905: P 1061602000:1061602005(5) ack 2357731244 win 5792
17 2.065276 IP 130.206.168.45.60905 > 193.252.23.108.110: F 2357731244:2357731244(0) ack 1061602005 win 65535
18 2.065890 IP 193.252.23.108.110 > 130.206.168.45.60905: . ack 2357731245 win 5792
19 2.065894 IP 193.252.23.108.110 > 130.206.168.45.60905: F 1061602005:1061602005(0) ack 2357731245 win 5792
20 2.065957 IP 130.206.168.45.60905 > 193.252.23.108.110: . ack 1061602006 win 65535
```

a) Indique a qué aplicación pertenece y qué acción del usuario ha causado esos paquetes

b) Indique cual es el cliente y cual el servidor en esta acción (con sus direcciones IP)

c) Haga una tabla para el cliente y otra para el servidor, indicando en qué estado de conexión estaba TCP al principio y en que estado ha quedado después de enviar o recibir cada paquete mostrado.