Práctica 1. Uso básico de servicios cliente-servidor

SCS, 2010/11

21 de septiembre de 2010

Índice

1. Utilidades de línea de comandos

Se repasará el uso de herramientas de línea de comandos de GNU/linux que pueden resultar útiles para experimentar y/o depurar aplicaciones cliente-servidor, especialmente las desarrolladas directamente sobre el interfaz de SOCKETS.

1.1. Comando `nc/netcat`

netcat(nc) es un comando que permite acceder a puertos TCP o UDP de la propia máquina o de otras máquinas remotas. También permite quedar a la escucha en un puerto dado (TCP o UDP) de la máquina local.

Funcionamiento

- Cuando funciona como cliente, **nc** crea un socket para conectarse al puerto indicado de la máquina destino.

```
$ nc maquina_destino puerto_destino
```

    - La conexión permenecrá abierta mientras no la finalice el servidor o el cliente **nc** (CONTROL+C)

- Cuando funciona como servidor (modo escucha [opción **-l**, *listen*]), abre un socket en la máquina local que queda a la escucha en el puerto indicado

  ```
  $ nc -l -p puerto_escucha
  ```

- En ambos casos, una vez establecida la conexión, **nc** envía a través del socket creado todo lo que reciba por la entrada estándar y envía a la salida estándar lo que le llegue por el socket.

Opciones interesantes:

- **-u** usa el modo UDP (por defecto son conexiones TCP)
- **-c comando** / **-e ejecutable** ejecuta un comando/programa una vez iniciada la conexión cuyas entrada y salida estándar están redirigidas a la conexión establecida

Más información:

- GNU netcat
- `man nc`

1.1.1. Pruebas a realizar

- Abrir una sesión **nc** en modo escucha desde un terminal en un puerto no privilegiado (¿1024). En otro terminal abrir otra sesión **nc** que se conecte al primero. Comprobar que el funcionamiento es el descrito.
- Repetir el ejercicio entre dos máquinas del laboratorio (usar las direcciones IP para identificar los equipos)

1.2. Comando `netstat`

netstat es un comando que ofrece información (uso, estado, etc) sobre las conexiones de la máquina propia, tanto entrantes como salientes, además de otras informaciones sobre la red.

Informa tanto de conexiones del dominio de Internet (locales y remota) como de conexiones del dominio UNIX (entre procesos locales).

La información típica que muestra para cada conexión de red es:

- tipo de protocolo
- dirección+puerto local
- dirección+puerto remoto
- estado de la conexión (para TCP): establecida, cerrada, esperando cierre, en espera, etc ...

Opciones interesante:

- **-a** muestra información sobre todas las conexiones/sockets (tanto TCP y UDP)
- **-p** muestra los procesos que están usando las conexiones (sólo si lo ejecuta el superusuario)
- **-l** muestra información sobre las conexiones/sockets en modo escucha
- **-e** información en formato extendido
- **-t** sólo conexiones TCP, **-u** sólo conexiones UDP

1.2.1. Pruebas a realizar

- Comprobar los distintos parámetros de **netstat** e identificar las conexiones abiertas en la máquina local
- Comprobar las conexiones abiertas en los ejemplos de comunicación entre sesiones **nc** de la sección anterior, identificar los puertos cliente de la sesión **nc**.

2. Ejemplos de servicios con estado y sin estado

2.1. Servicio HTTP

Ejemplo típico de servicio sin estado, implementa un esquema *una petición → una respuesta*.

Más información sobre el protocolo HTTP (incluido formato de los mensajes).

- protocolo HTTP (español)
- protocolo HTTP (inglés)

Cada petición HTTP recibida por el servidor es independiente de las anteriores.

- La respuesta enviada no depende de respuestas previas ni de las peticiones realizadas por el cliente (el servidor no almacena información de las conexiones previas de un cliente).
- El servidor trata todas la peticiones de la misma forma, independientemente del cliente que se la envía.
- El mecanismo de *cookies* permite dotar a los servidores HTTP de un comportamiento *con estado*, mediante el almacenamiento de un identificado único en los clientes, que será adjuntado como parámetro en cada petición enviada por el cliente.

  Mas información sobre cookies

2.1.1. Acceso a un servidor WEB

Comprobar el diálogo HTTP desde el punto de vista de un cliente.

1. Ejecutar **nc** en modo conexión con el puerto 80 de un servidor web accesible.

   ```
   $ nc ccia.ei.uvigo.es 80
   ```

2. Escribir la siguiente petición (copiar y pegar tal cual, incluida la línea en blanco final)

   ```
   GET /docencia/SCS/index.html HTTP/1.1
   Host: ccia.ei.uvigo.es

   ```

3. Comprobar que sucede si se envía sólo un mensaje GET (copiar y pegar tal cual, incluida la línea en blanco final)

   ```
   GET /docencia/SCS/index.html HTTP/1.1

   ```

2.1.2. Simulación de un servidor WEB en el puerto 8080

Comprobar el diálogo HTTP desde el punto de vista del servidor.

1. Ejecutar **nc** en modo escucha en el puerto 8080. (o en cualquier otro puerto ¿1024)

   ```
   $ nc -l -p 8080
   ```

2. Desde un navegador web escribir `http://localhost:8080/index.html`
   - **nc** muestra la petición HTTP realizada por el navegador
3. Responder a la petición desde **nc** (copiar y pegar tal cual, incluidas las líneas en blanco)

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 56

<html><body> hola <img src="imagen.gif"> </body> </html>
```

   - Comprobar lo que sucede en la sesión **nc** que simula el servidor web.
4. Realizar otra petición desde el navegador (recargar página) y enviarle una nueva respuesta desde **nc** (copiar y pegar tal cual, incluidas las líneas en blanco)

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 70

<html><body> hola <a href="principal.html"> enlace </a> </body> </html>
```

   - Pinchar en el enlace y comprobar que sucede

2.2. Servicios POP3 y SMTP

Los protocolos POP3 (servicio de correo entrante) y SMTP (servicio de correo saliente) son ejemplos de servidores **con estado**. En función de los mensajes previos enviados por el cliente, como parte de una conexión establecida previamente, las respuestas del servidor varían.

Típicamente, el comportamiento de este tipo de servicios con estadoestará gobernado por una máquina de estados (o un mecanismo equivalente) exclusiva para cada cliente. Esa información permite que el servidor pueda llevar traza de las peticiones recibidas y del **estado** en el que se encuentra el **diálogo con el cliente**.

Más información sobre el protocolo SMTP.

- protocolo SMTP (español)
- protocolo SMTP (inglés)

Más información sobre el protocolo POP3.

- protocolo POP3 (español)
- protocolo POP3 (inglés)

2.2.1. Conectar con un servidor SMTP (puerto 25)

Elegir un servidor SMTP donde el alumno tenga acceso y simular el diálogo de un cliente SMTP a la hora de enviar un e-mail.

1. Ejecutar **nc** en modo conexión con el puerto 25 del servidor SMTP

   ```
   $ nc aaa.aaa.com 25
   ```

2. Obtener información sobre el servidor SMTP.
   Escribir el siguiente mensaje en la sesión **nc**:

```
HELO  <nombre o IP del cliente>
```

3. Enviarse un e-mail a si mismo.

   Escribir los siguientes mensajes en la sesión **nc**:

   ```
   MAIL FROM: login@aaa.aaa.com
   RCPT TO:login@aaa.aaa.com
   DATA
   Subject: asunto del mensaje
   mensaje a enviar
   mensaje a enviar
   .
   ```

   a) "`MAIL FROM:`" indica quien envía el mensaje

   b) "`RCPT TO:`" indica quien es el destinatario (si fueran varios se repite para cada uno de ellos)

   c) "`DATA`" indica el comienzo del contenido del mensaje.

   d) El fin del mensaje se marca con "`.`" (punto "solitario" + retorno de carro)

4. Cerrar la conexión con el servidor SMTP

   Escribir el siguiente mensaje en la sesión **nc**:

   ```
   QUIT
   ```

5. Comprobar la recepción del mensaje

2.2.2. Conectar con un servidor POP3 (puerto 110)

Elegir un servidor POP3 donde el alumno tenga acceso y simular el diálogo de un cliente POP3 a la hora de descargar los mensajes del buzón del usuario.

1. Ejecutar **nc** en modo conexión con el puerto 110 del servidor POP3

   ```
   $ nc aaa.aaa.com 110
   ```

2. Autenticarse frente al servidor.

   Escribir los siguientes 2 mensajes en la sesión **nc** (envío del nombre de usuario + envío de la clave)

   ```
   USER miusuario
   PASS mipassword
   ```

   Aviso: el tráfico irá en claro y el login y el password serán visibles para cualquiera que escuche en ese enlace

3. Gestionar los mensajes del buzón

   Escribir la siguiente secuencia de mensajes en la sesión **nc** (ver estado del buzón, obtener la lista de mensajes pendientes de descargar, descargar el primer mensaje, cerrar la conexión POP3)

   ```
   STAT
   LIST
   RETR 1
   QUIT
   ```

   - Comprobar las respuestas y resultados mostrados en el cliente **nc**

3. Documentación a entregar

Descripción breve de las pruebas realizadas (comandos ejecutados, parámetros utilizados, etc) en las dos secciones anteriores (herramientas de red y servicios con y sin estado).

Explicar los resultados obtenidos en cada una de las pruebas realizadas.

(Límite: 2-3 páginas)