Protocolos de transporte

Revisión: 2025-05-21

Resumen

El objetivo de esta práctica es comprender cómo funcionan ciertos aspectos de la capa de transporte de TCP/IP y UDP.

E72. Suponiendo que no tiene acceso a un host, determinar desde otro host si están activos los siguientes servicios:

a. daytime
b. time
c. smtp
d. telnet

E73. *{W}* Elegir un servicio de los anteriores que se encuentre inactivo y analizar los paquetes que se generan e intercambian cuando se realiza un intento de conexión.

E74. {W} Utilizando netcat transfiera un archivo entre dos hosts utilizando TCP, UDP, y SCTP. Prestar atención a las diferencias de uso (UDP) y analizar con wireshark los diferentes flujos de información.

E75. {W} Exponer un servidor `echo TCP` utilizando **xinetd**. Debe asegurarse que el servidor `echo UDP` no se encuentra habilitado. De ser posible ejecutar el cliente (**netcat**) desde un host diferente (pero con conectividad). Analizar los paquetes en la interfaz del red donde el cliente deja los datagramas cuando:

a. Se realiza una conexión al servicio echo.

b. Se envía un único caracter.

c. Se envían varios caracteres dejando un pequeño intervalo entre cada uno (por ejemplo 1 segundo).

d. Se envían varios caracteres simultáneamente (por ejemplo escribiendo rápidamente o manteniendo presionada una tecla).

e. Se desconecta el servicio (se mata el proceso servidor)

También analice los cambios de estados que publica la salida de **netstat** en las diferentes etapas. Puede servirse de la Figura 1, “Maquina de estados de TCP” para el seguimiento.

E76. {W} Exponer un servicio `echo UDP` utilizando **xinetd**. Debe asegurarse que el servidor echo TCP no se encuentra habilitado. De ser posible ejecutar el cliente (**netcat**) desde un host diferente (pero con conectividad). Analizar los paquetes en la interfaz del red donde el cliente deja los datagramas cuando:

a. Se ejecuta `nc -u <direccion IP> echo`

b. Se envía un único caracter (presione **Enter**)

c. Se envían varios caracteres

d. Se cierra abruptamente alguno de los procesos (cliente o del servidor)

¿Por qué se muestra en la salida estándar de **netcat** el mismo contenido que se envió? Describa el mecanismo.

E77. {W} Utilizar **nmap** para determinar cuales son los servicios (TCP y UDP) para alguna de las direcciones IP que resultan de resolver el nombre `pampero.itba.edu.ar`. Analizar los paquetes generados.

a. ¿Cómo funciona el escaneo TCP? Verificar empíricamente de forma directa las diferencia el *TCP SYN scan* y el *TCP connect scan*. ¿En que situación es conveniente usar cada uno?

b. ¿Cómo funciona el escaneo UDP?

E78. {W} Utilizar **nmap** para determinar estimativamente el sistema operativo de varios hosts. De ser posible probar con impresoras de red, celulares, "*access points*" hogareños, etc.

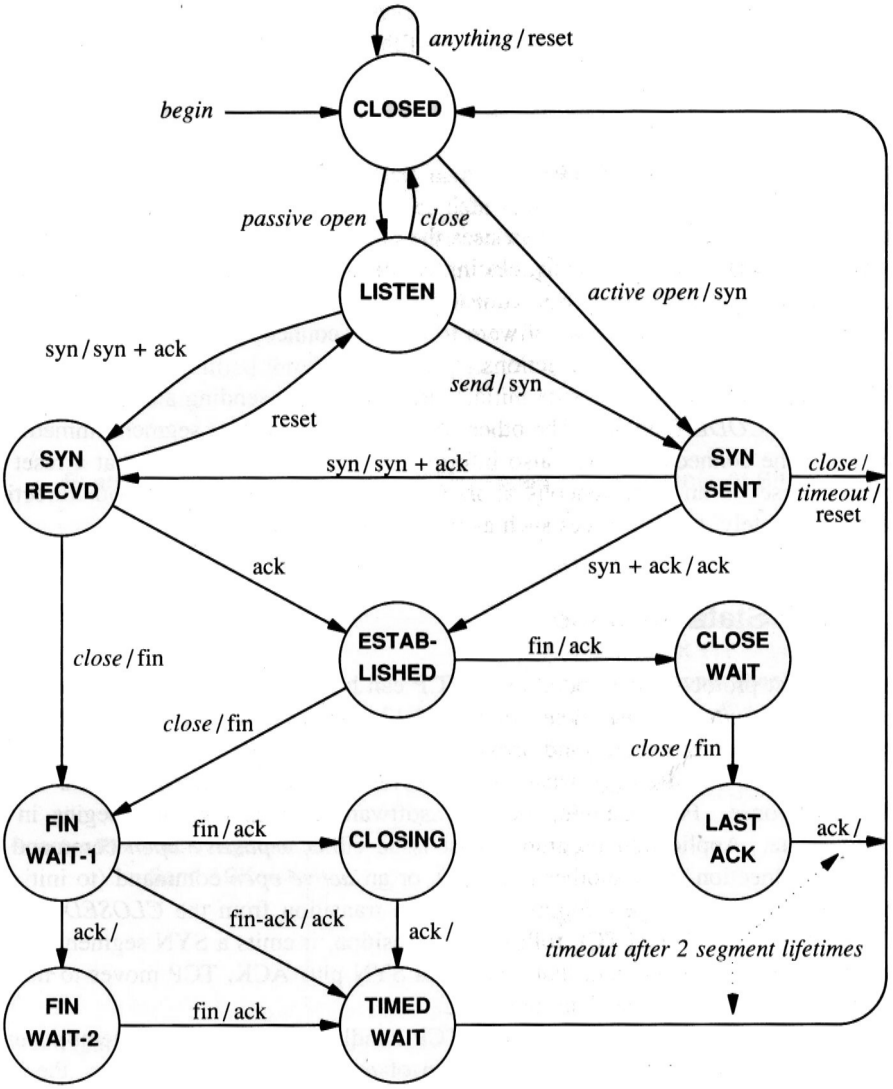

Figura 1. Maquina de estados de TCP

Bibliografía

[xinetd(8)] ***xinetd*** *- the extended Internet services daemon*.

> **xinetd** performs the same function as **inetd**: it starts programs that provide Internet services. Instead of having such servers started at system initialization time, and be dormant until a connection request arrives, xinetd is the only daemon process started and it listens on all service ports for the services listed in its configuration file. When a request comes in, xinetd starts the appropriate server. Because of the way it operates, xinetd (as well as inetd) is also referred to as a super-server.

[services(7)] ***services*** *- Internet network services list*.

services is a plain ASCII file providing a mapping between human- friendly textual names for internet services, and their underlying assigned port numbers and protocol types. Every networking program should look into this file to get the port number (and protocol) for its service. The C library routines getservent(3), getservbyname(3), getservbyport(3), setservent(3), and endservent(3) support querying this file from programs.

[nc(1)] *nc — arbitrary TCP and UDP connections and listens*. BSD General Commands Manual. February 7, 2012.

The nc (or netcat) utility is used for just about anything under the sun involving TCP, UDP, or UNIX-domain sockets. It can open TCP connections, send UDP packets, listen on arbitrary TCP and UDP ports, do port scan# ning, and deal with both IPv4 and IPv6. Unlike telnet(1), nc scripts nicely, and separates error messages onto standard error instead of send# ing them to standard output, as telnet(1) does with some.

[nmap(1)] *Network exploration tool and security / port scanner*.

Nmap (“Network Mapper”) is an open source tool for network exploration and security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. While Nmap is commonly used for security audits, many systems and network administrators find it useful for routine tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime.

[netstat(8())] *Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships*.

Netstat prints information about the Linux networking subsystem. The type of information printed is controlled by the first argument, as follows:...