Lecture 8 — Port numbers and Sockets

Sending a packet to a host isn't enough. When the destination host gets the packet, what program should it send it to? (Web server? Email server? Telnet?) Part of the layer 4 header includes a port number to identify which program should receive the packet and which one sent the packet. These are 16 bit values. (Example: a web browser with two windows open. You click a line on one, switch to the other and click a different link. Each browser window's HTTP request packet will use a different source port number so the replies will be sent to the correct window.)

When a host receives a packet, the kernel will check the port number to see which process to send it to. So how does a client (say a web browser) know which port number corresponds to a server? The servers listen for a particular port number that all agree on (IANA).

The standard servers use well known Port numbers in the range 0-1023. Which service (and its application level protocol) uses which port number is documented in the /etc/services file. (This is not a configuration file!) There is a similar file /etc/protocols with human-readable names for the various protocol numbers.

Well-known ports are reserved for public services. Many certification exams (and employers) expect you to memorize a number of these, such as:

- FTP (20 for data and 21 for control), ssh (22), telnet (23), SMTP (25), DNS (53), BOOTP/DHCP (67 for server, 68 for client), HTTP (80), POP (110), portmap (111), NNTP (119, used for Netnews), NTP (123), NetBIOS (139), IMA P (143), SNMP (161), and HTTPS (443), IPP (631), IMAPS (993), POPS (995). Use grep on /etc/services to lookup port numbers quickly.

Using specific ports for specific services makes it easy for clients; to contact your (web) server; the client will send the request packet to your IP address and destination port 80.

Note that on a Unix system root privileges are needed to listen in on a well-known Port. (This prevents a user from crashing your web server and then starting their own, fooling people who visit your web site!)

The range 1024-49151 is for User (Registered) Ports, used for other public services (such as Unix rlogin or the w3c SSL services). These are used because today there are well over 1023 application protocols registered. These are also registered by IANA (as a public service.)

The Dynamic and/or Private Ports are those from 49152 through 65535. Clients will use any available port number higher than 1024; the kernel keeps track of which are in use. (Note: you can use a telnet application to connect to any port: debugging.)

Sockets

A socket is the combination of an IP address, type, and port number.

socket number

read/recv  write/send

Stream sockets

Datagram sockets

User (Registered) Ports

rlogin    SSL

Dynamic and/or Private Ports
raw socket

Example pseudo-code of client using TCP/IP socket to connect to server:

```c
sock = create_socket( PF_INET, SOCK_STREAM );
sock.connect( dest_addr, dest_port );
if ( sock.is_connected() ) {
    sock.send( request_data );
    sock.recv( response_data );
}
sock.close();
```

**inetd**

**xinetd**

**daemons**

**spawn**

```
/etc/inetd.conf
```

```
lsof -i | grep LISTEN
```

Scan all ports

```bash
# Shows what program is listening on what port:
# Show all connections, like lsof
# Show LISTENING (services) only
```

**Question**

**Answer:**

**RPC**

**RPC**
Each service is identified by a unique RPC program number.

These are listed in /etc/rpc (format: service-name program-number [aliases]).

Each service also has a version number so it is possible to run multiple versions of the same service. Finally, each service binds one or more functions (a.k.a. procedures or methods) with procedure numbers.

A remote RPC client can query the portmap daemon with a program number, a version number, and a transport protocol (e.g., TCP or UDP). It receives a port number for that service, plus a list of procedures and their numbers.

Historically RPC has had many security problems. With dynamic port number assignment it is difficult to open firewall holes for just the services your host provides.

You should configure services to use static port number assignments if possible and limit the firewall holes needed. RPC should be turned off completely on your server unless you are using RPC dependent services. These include NFS and rlogin.

`rpcinfo` is a utility that reports which RPC ports are currently assigned (using `-p`) and can be used to examine, test, and debug RPC services such as NFS or samba. See the man page for more info.

NFS    rlogin

With dynamic port number assignment it is difficult to open firewall holes for just the services your host provides.