80 - RCP - Remote Control Procedure

The portmapping is done through the daemon `/sbin/portmap`
Many programs(daemons) are using the RPC to communicate with other hosts.
The RPC procedure is done as follows:(here we use nfs services as an example)

nfs server daemon starts and registers its port with portmap .
nfs server daemon now listens to that port.
The nfs client asks the `portmap`(port 111) for the port of the NFS server.
He gets the port number of nfs server and starts communicating with it.

When a client wants to call a procedure from an rpc service it needs to know the version number of the rpc server so it makes the call appropriately. This way old and new versions of rpc services can communicate between each other properly.

To get the list of rpc services available on a server the following command can be used:

`rpcinfo -p <servername>`	List of rpc services on the server
`rpcinfo -p`	List of rpc services on localhost

The files `/etc/hosts.allow` and `/etc/hosts.deny` contains a list of hosts that are either allowed or denied to connect to the server through an rpc service.
examples:

```
/etc/hosts.allow
      portmap: your.sub.net.number/your.sub.net.mask

/etc/hosts.deny
      portmap: ALL
             -----------------------------------------------
```

RPC.PORTMAP

result of
`man portmap`

NAME

portmap - DARPA port to RPC program number mapper

SYNOPSIS

portmap [-dv]

DESCRIPTION

Portmap is a server that converts RPC program numbers into DARPA protocol port numbers. It must be running in order to make RPC calls.

When an RPC server is started, it will tell portmap what port number it is listening to, and what RPC program numbers it is prepared to serve. When a client wishes to make an RPC call to a given program number, it will first contact portmap on the server machine to determine the port number where RPC packets should be sent.

Portmap must be started before any RPC servers are invoked.
Normally portmap forks and dissociates itself from the terminal like any other daemon. Portmap then logs errors using syslog(3).(/var/log/messages)

Option available:

-d (debug) prevents portmap from running as a daemon, and causes er rors and debugging information to be printed to the standard er ror output.

-v (verbose) causes portmap to give more logging information to sys logd(8.)

Access control

By default, host access control is enabled. However, the host that runs the portmapper is always considered authorized. The host access control tables are never consulted with requests from the local system itself; they are always consulted with requests from other hosts.

In order to avoid deadlocks, the portmap program does not attempt to look up the remote host name or user name, nor will it try to match NIS netgroups. The upshot of all this is that only network number patterns will work for portmap access control.

Sample entries for the host access-control files are:

/etc/hosts.allow:

```
portmap: your.sub.net.number/your.sub.net.mask
portmap: 255.255.255.255 /0.0.0.0
```

/etc/hosts.deny

```
portmap: ALL
```

The syntax of the access-control files is described in the `hosts_access(5)` and `hosts_options(5)` manual page that comes with the tcp wrapper (log_tcp) sources. The safe_finger command comes with later wrapper releases.

The first line in the `hosts.allow` file permits access from all systems within your own subnet. Some rpc services rely on broadcasts and will contact your portmapper anyway; and once an intruder has access to your local network segment you're already in deep trouble.

The second line in the `hosts.allow` file may be needed if there are any

PC-NFS systems on your network segment.

For security reasons, the portmap process drops root privilegs after initialization. The access control files should therefore be readable for group or world.

SEE ALSO

inetd.conf(5), rpcinfo(8), inetd(8), syslogd(8), hosts_access(5), hosts_options(5)

BUGS

If portmap crashes, all servers must be restarted.

HISTORY

The portmap command appeared in 4.3BSD, March 16, 1991