The /etc/services File

Most UNIX network services are provided by individual programs called *servers*. For a server to operate, it must be assigned a protocol, e.g. TCP or UDP, be assigned a port number, and somehow be started.

As we know, most Internet services are assigned a specific port for their exclusive use. When a client opens a connection across the network to a server, the client uses the port to specify which service it wishes to use. These ports are called *well-known ports* because they need to be known in advance by both the client and the server. UNIX uses the */etc/services* file as a small local database. For each service this file specifies the service's well-known port number and notes whether the service is available as a TCP or UDP service. The */etc/services* file is distributed as part of the UNIX operating system. A typical */etc/services* file that comes with the Solaris distribution is

```
$ cat /etc/services
#
# Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident  "@(#)services   1.34    08/11/19 SMI"
#
# Network services, Internet style
#
tcpmux          1/tcp
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
daytime         13/tcp
daytime         13/udp
netstat         15/tcp
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp             21/tcp
ssh             22/tcp                          # Secure Shell
telnet          23/tcp
smtp            25/tcp          mail
time            37/tcp          timserver
time            37/udp          timserver
name            42/udp          nameserver
whois           43/tcp          nicname         # usually to sri-nic
domain          53/udp
domain          53/tcp
bootps          67/udp                          # BOOTP/DHCP server
bootpc          68/udp                          # BOOTP/DHCP client
kerberos        88/udp          kdc             # Kerberos V5 KDC
kerberos        88/tcp          kdc             # Kerberos V5 KDC
```

```
hostnames       101/tcp         hostname        # usually to sri-nic
pop2            109/tcp         pop-2           # Post Office Protocol - V2
pop3            110/tcp                         # Post Office Protocol -
Version 3
sunrpc          111/udp         rpcbind
sunrpc          111/tcp         rpcbind
imap            143/tcp         imap2           # Internet Mail Access Protocol
v2
ldap            389/tcp                         # Lightweight Directory Access
Protocol
ldap            389/udp                         # Lightweight Directory Access
Protocol
dhcpv6-client   546/udp         dhcpv6c         # DHCPv6 Client (RFC 3315)
dhcpv6-server   547/udp         dhcpv6s         # DHCPv6 Server (RFC 3315)
submission      587/tcp                         # Mail Message Submission
submission      587/udp                         #    see RFC 2476
ldaps           636/tcp                         # LDAP protocol over TLS/SSL
(was sldap)
ldaps           636/udp                         # LDAP protocol over TLS/SSL
(was sldap)
#
# Host specific functions
#
tftp            69/udp
rje             77/tcp
finger          79/tcp
link            87/tcp          ttylink
supdup          95/tcp
iso-tsap        102/tcp
x400            103/tcp                         # ISO Mail
x400-snd        104/tcp
csnet-ns        105/tcp
pop-2           109/tcp                         # Post Office
uucp-path       117/tcp
nntp            119/tcp         usenet          # Network News Transfer
ntp             123/tcp                         # Network Time Protocol
ntp             123/udp                         # Network Time Protocol
netbios-ns      137/tcp                         # NETBIOS Name Service
netbios-ns      137/udp                         # NETBIOS Name Service
netbios-dgm     138/tcp                         # NETBIOS Datagram Service
netbios-dgm     138/udp                         # NETBIOS Datagram Service
netbios-ssn     139/tcp                         # NETBIOS Session Service
netbios-ssn     139/udp                         # NETBIOS Session Service
NeWS            144/tcp         news            # Window System
slp             427/tcp         slp             # Service Location Protocol, V2
slp             427/udp         slp             # Service Location Protocol, V2
mobile-ip       434/udp         mobile-ip       # Mobile-IP
cvc_hostd       442/tcp                         # Network Console
ike             500/udp         ike             # Internet Key Exchange
```

```
uuidgen         697/tcp                         # UUID Generator
uuidgen         697/udp                         # UUID Generator
#
# UNIX specific services
#
# these are NOT officially assigned
#
exec            512/tcp
login           513/tcp
shell           514/tcp         cmd             # no passwords used
printer         515/tcp         spooler         # line printer spooler
courier         530/tcp         rpc             # experimental
uucp            540/tcp         uucpd           # uucp daemon
biff            512/udp         comsat
who             513/udp         whod
syslog          514/udp
talk            517/udp
route           520/udp         router routed
ripng           521/udp
klogin          543/tcp                         # Kerberos authenticated rlogin
kshell          544/tcp         cmd             # Kerberos authenticated remote
shell
new-rwho        550/udp         new-who         # experimental
rmonitor        560/udp         rmonitord       # experimental
monitor         561/udp                         # experimental
pcserver        600/tcp                         # ECD Integrated PC board srvr
sun-dr          665/tcp                         # Remote Dynamic
Reconfiguration
kerberos-adm    749/tcp                         # Kerberos V5 Administration
kerberos-adm    749/udp                         # Kerberos V5 Administration
kerberos-iv     750/udp                         # Kerberos V4 key server
krb5_prop       754/tcp                         # Kerberos V5 KDC propogation
ufsd            1008/tcp        ufsd            # UFS-aware server
ufsd            1008/udp        ufsd
cvc             1495/tcp                        # Network Console
ingreslock      1524/tcp
www-ldap-gw     1760/tcp                        # HTTP to LDAP gateway
www-ldap-gw     1760/udp                        # HTTP to LDAP gateway
listen          2766/tcp                        # System V listener port
nfsd            2049/udp        nfs             # NFS server daemon (clts)
nfsd            2049/tcp        nfs             # NFS server daemon (cots)
eklogin         2105/tcp                        # Kerberos encrypted rlogin
lockd           4045/udp                        # NFS lock daemon/manager
lockd           4045/tcp
ipsec-nat-t     4500/udp                        # IPsec NAT-Traversal
dtspc           6112/tcp                        # CDE subprocess control
fs              7100/tcp                        # Font server
apocd           38900/udp
#[swat] The swat service is added by the SUNWsmbar package.
```

```
#[swat] Removing the swat service manually while SUNWsmbar
#[swat] package is installed in the system can cause issues
#[swat] with smf(5) stability or with zones(5) installation.
swat            901/tcp                         # Samba Web Adm.Tool
servicetag      6481/udp
servicetag      6481/tcp
snmpd           161/udp        snmp             # SMA snmp daemon
$
```

The information in the */etc/services* file is derived from Internet RFCs and other sources. Some of the services listed in the */etc/services* file are no longer in wide-spread use. Nevertheless, their names still appear in the file. Each line gives the canonical name of the service, the port number and protocol, and any aliases for the service name. As you can see, the SMTP service uses TCP on port 25 and also goes by the alias "mail."

Trusted Ports

On UNIX systems, TCP and UDP ports in the range of 0-1023 are sometimes referred to as *trusted ports*. UNIX requires that a process have superuser privileges to be able to start listening for incoming connections on such a port or to originate connections to a remote server using one of these ports as the source port. Note that any user can connect to a trusted port from an untrusted port.

Trusted ports were intended to prevent a regular user from obtaining privilege information. For example, if a regular user could write a program that listened to port 23, that program could masquerade as a *telnet server*, receive connections from unsuspecting users, and obtain their passwords.

This idea of a trusted port is a UNIX convention. It is *not* part of the Internet standard, and manufacturers of other TCP/IP implementations are not bound to observe this protocol. In particular, there are no restrictions that prohibit non-privileged users and processes on Windows-based machines from originating or accepting connections on so-called trusted ports.

Ports Cannot Be Trusted

It is important to remember that port assignments are standards, but they are not set in stone. Servers can be run on ports that are unassigned or are assigned to other protocols. This is especially problematic for organizations that wish to block some kinds of protocols from leaving their organizations while allowing others through. If you allow the packets for any specific IP port to travel unrestricted from the inside of your organization to the outside, then a malicious insider can effectively use that hole to tunnel any protocol through your defenses.

Story

Because the SSL protocol cannot be effectively proxied, many organizations allow TCP connections on port 443 to travel from inside their organization to outside their organization. This is because attempts to proxy the SSL protocol are effectively man-in-the-middle attacks and are specifically detected by the SSL protocol. I one time had the opportunity to spend a couple of days on a DoD base. Their firewall was configured to allow packets through on port 443 but not packets on port 22, i.e. ssh. The reason was "security." The network administrator had made a determination that ssh was too dangerous a protocol to allow from on base to off base. To get around this minor inconvenience, I telephoned one of my students at Clemson and asked her to set up an SSH server running on port 443. A few moments later, I used the ssh command on my laptop to connect to that server on port 443. On top of this SSH connection, I tunneled a variety of other protocols, including POP, SMTP, IMAP, HTTP and X. So much for the restrictive firewall!