

**NAME**

**bootpd**, **bootpgw** - Internet Boot Protocol server/gateway

**SYNOPSIS**

```
bootpd [-i | -s] [-c chdir-path] [-d level] [-h hostname] [-t timeout] [bootptab [dumpfile]]  
bootpgw [-i | -s] [-d level] [-h hostname] [-t timeout] server
```

**DESCRIPTION**

The **bootpd** utility implements an Internet Bootstrap Protocol (BOOTP) server as defined in RFC951, RFC1532, and RFC1533. The **bootpgw** utility implements a simple BOOTP gateway which can be used to forward requests and responses between clients on one subnet and a BOOTP server (i.e. **bootpd**) on another subnet. While either **bootpd** or **bootpgw** will forward BOOTREPLY packets, only **bootpgw** will forward BOOTREQUEST packets.

One host on each network segment is normally configured to run either **bootpd** or **bootpgw** from inetd(8) by including one of the following lines in the file */etc/inetd.conf*:

```
bootps dgram udp wait root /usr/libexec/bootpd bootpd /etc/bootptab  
bootps dgram udp wait root /usr/libexec/bootpgw bootpgw server
```

This mode of operation is referred to as "inetd mode" and causes **bootpd** (or **bootpgw**) to be started only when a boot request arrives. If it does not receive another packet within fifteen minutes of the last one it received, it will exit to conserve system resources. The **-t** option controls this timeout (see OPTIONS).

It is also possible to run **bootpd** (or **bootpgw**) in "standalone mode" (without inetd(8)) by simply invoking it from a shell like any other regular command. Standalone mode is particularly useful when **bootpd** is used with a large configuration database, where the start up delay might otherwise prevent timely response to client requests. (Automatic start up in standalone mode can be done by invoking **bootpd** from within */etc/rc.local*, for example.) Standalone mode is less useful for **bootpgw** which has very little start up delay because it does not read a configuration file.

Either program automatically detects whether it was invoked from inetd or from a shell and automatically selects the appropriate mode. The **-s** or **-i** option may be used to force standalone or inetd mode respectively (see OPTIONS).

**OPTIONS**

The following options are available:

**-a** Skip ARP table modifications.

**-t** *timeout*

Specify the *timeout* value (in minutes) that a **bootpd** or **bootpgw** process will wait for a BOOTP packet before exiting. If no packets are received for *timeout* minutes, then the program will exit. A timeout value of zero means "run forever". In standalone mode, this option is forced to zero.

**-d** *debug-level*

Set the *debug-level* variable that controls the amount of debugging messages generated. For example, **-d4** or **-d 4** will set the debugging level to 4. For compatibility with older versions of **bootpd**, omitting the numeric parameter (i.e., just **-d**) will simply increment the debug level by one.

**-c** *chdir-path*

Set the current directory used by **bootpd** while checking the existence and size of client boot files. This is useful when client boot files are specified as relative pathnames, and **bootpd** needs to use the same current directory as the TFTP server (typically */tftpboot*). This option is not recognized by **bootpgw**.

**-h** *hostname*

Specify the hostname corresponding to the IP address to listen on. By default, **bootpd** listens on the IP address corresponding to the machine's hostname, as returned by `gethostname(3)`.

**-i** Force inetd mode. This option is obsolete, but remains for compatibility with older versions of **bootpd**.

**-s** Force standalone mode. This option is obsolete, but remains for compatibility with older versions of **bootpd**.

*bootptab*

Specify the name of the configuration file from which **bootpd** loads its database of known clients and client options (**bootpd** only).

*dumpfile*

Specify the name of the file that **bootpd** will dump its internal database into when it receives a SIGUSR1 signal (**bootpd** only). This option is only recognized if **bootpd** was compiled with the `-DDEBUG` flag.

*server* Specify the name of a BOOTP server to which **bootpgw** will forward all BOOTREQUEST packets it receives (**bootpgw** only).

**OPERATION**

Both **bootpd** and **bootpgw** operate similarly in that both listen for any packets sent to the *bootps* port, and both simply forward any BOOTREPLY packets. They differ in their handling of BOOTREQUEST packets.

When **bootpgw** is started, it determines the address of a BOOTP server whose name is provided as a command line parameter. When **bootpgw** receives a BOOTREQUEST packet, it sets the "gateway address" and "hop count" fields in the packet and forwards the packet to the BOOTP server at the address determined earlier. Requests are forwarded only if they indicate that the client has been waiting for at least three seconds.

When **bootpd** is started it reads a configuration file, (normally */etc/bootptab*) that initializes the internal database of known clients and client options. This internal database is reloaded from the configuration file when **bootpd** receives a hangup signal (SIGHUP) or when it discovers that the configuration file has changed.

When **bootpd** receives a BOOTREQUEST packet, it looks for a database entry matching the client request. If the client is known, **bootpd** composes a BOOTREPLY packet using the database entry found above, and sends the reply to the client (possibly using a gateway). If the client is unknown, the request is discarded (with a notice if debug > 0).

If **bootpd** is compiled with the -DDEBUG option, receipt of a SIGUSR1 signal causes it to dump its internal database to the file */tmp/bootpd.dump* or the dumpfile specified as a command line parameter.

During initialization, both programs determine the UDP port numbers to be used by calling `getservbyname(3)` (which normally uses */etc/services*). Two service names (and port numbers) are used:

bootps BOOTP Server listening port  
bootpc BOOTP Client destination port

If the port numbers cannot be determined using `getservbyname(3)` then the values default to `bootps=67` and `bootpc=68`.

## FILES

*/etc/bootptab* Database file read by **bootpd**.  
*/tmp/bootpd.dump* Debugging dump file created by **bootpd**.  
*/etc/services* Internet service numbers.  
*/tftpboot* Current directory typically used by the TFTP server and **bootpd**.

## SEE ALSO

bootptab(5), inetd(8), tftpd(8)

DARPA Internet Request For Comments:

RFC951 Bootstrap Protocol

RFC1532 Clarifications and Extensions for the Bootstrap Protocol

RFC1533 DHCP Options and BOOTP Vendor Extensions

## AUTHORS

This distribution is currently maintained by Walter L. Wimer <*walt+@cmu.edu*>.

The original BOOTP server was created by Bill Croft at Stanford University in January 1986.

The current version of **bootpd** is primarily the work of David Kovar, Drew D. Perkins, and Walter L. Wimer, at Carnegie Mellon University.

Enhancements and bug-fixes have been contributed by:

(in alphabetical order)

Danny Backx <*db@sunbim.be*>

John Brezak <*brezak@ch.hp.com*>

Frank da Cruz <*fdc@cc.columbia.edu*>

David R. Linn <*drl@vuse.vanderbilt.edu*>

Jim McKim <*mckim@lerc.nasa.gov*>

Gordon W. Ross <*gwr@mc.com*>

Jason Zions <*jazz@hal.com*>.

## BUGS

Individual host entries must not exceed 1024 characters.