NAME

ypserv - NIS database server

SYNOPSIS

ypserv [**-n**] [**-d**] [**-P** *port*] [**-p** *path*]

DESCRIPTION

NIS is an RPC-based service designed to allow a number of UNIX-based machines to share a common set of configuration files. Rather than requiring a system administrator to update several copies of files such as */etc/hosts*, */etc/passwd* and */etc/group*, which tend to require frequent changes in most environments, NIS allows groups of computers to share one set of data which can be updated from a single location.

The **ypserv** utility is the server that distributes NIS databases to client systems within an NIS *domain*. Each client in an NIS domain must have its domainname set to one of the domains served by **ypserv** using the domainname(1) command. The clients must also run ypbind(8) in order to attach to a particular server, since it is possible to have several servers within a single NIS domain.

The databases distributed by **ypserv** are stored in */var/yp/[domainname]* where *domainname* is the name of the domain being served. There can be several such directories with different domainnames, and you need only one **ypserv** daemon to handle them all.

The databases, or *maps* as they are often called, are created by */var/yp/Makefile* using several system files as source. The database files are in db(3) format to help speed retrieval when there are many records involved. In FreeBSD, the maps are always readable and writable only by root for security reasons. Technically this is only necessary for the password maps, but since the data in the other maps can be found in other world-readable files anyway, it does not hurt and it is considered good general practice.

The **ypserv** utility is started by */etc/rc.d/ypserv* if it has been enabled in */etc/rc.conf*.

SPECIAL FEATURES

There are some problems associated with distributing a FreeBSD password database via NIS: FreeBSD normally only stores encrypted passwords in */etc/master.passwd*, which is readable and writable only by root. By turning this file into an NIS map, this security feature would be completely defeated.

To make up for this, the FreeBSD version of **ypserv** handles the *master.passwd.byname* and *master.passwd.byuid* maps in a special way. When the server receives a request to access either of these two maps (or in fact either of the *shadow.byname* or *shadow.byuid* maps), it will check the TCP port from which the request originated and return an error if the port number is greater than 1023. Since only

the superuser is allowed to bind to TCP ports with values less than 1024, the server can use this test to determine whether or not the access request came from a privileged user. Any requests made by non-privileged users are therefore rejected.

Furthermore, the getpwent(3) routines in the FreeBSD standard C library will only attempt to retrieve data from the *master.passwd.byname* and *master.passwd.byuid* maps for the superuser: if a normal user calls any of these functions, the standard *passwd.byname* and *passwd.byuid* maps will be accessed instead. The latter two maps are constructed by */var/yp/Makefile* by parsing the *master.passwd* file and stripping out the password fields, and are therefore safe to pass on to unprivileged users. In this way, the shadow password aspect of the protected *master.passwd* database is maintained through NIS.

NOTES

Setting Up Master and Slave Servers

ypinit(8) is a convenient script that will help setup master and slave NIS servers.

Limitations

There are two problems inherent with password shadowing in NIS that users should be aware of:

1. The 'TCP port less than 1024' test is trivial to defeat for users with unrestricted access to machines on your network (even those machines which do not run UNIX-based operating systems).

2. If you plan to use a FreeBSD system to serve non-FreeBSD clients that have no support for password shadowing (which is most of them), you will have to disable the password shadowing entirely by uncommenting the *UNSECURE=True* entry in */var/yp/Makefile*. This will cause the standard *passwd.byname* and *passwd.byuid* maps to be generated with valid encrypted password fields, which is necessary in order for non-FreeBSD clients to perform user authentication through NIS.

Security

In general, any remote user can issue an RPC to **ypserv** and retrieve the contents of your NIS maps, provided the remote user knows your domain name. To prevent such unauthorized transactions, **ypserv** supports a feature called *securenets* which can be used to restrict access to a given set of hosts. At startup, **ypserv** will attempt to load the securenets information from a file called */var/yp/securenets*. (Note that this path varies depending on the path specified with the **-p** option, which is explained below.) This file contains entries that consist of a network specification and a network mask separated by white space. Lines starting with "#" are considered to be comments. A sample securenets file might look like this:

```
# allow connections from local host -- mandatory
```

```
127.0.0.1     255.255.255.255
# allow connections from any host
# on the 192.168.128.0 network
192.168.128.0 255.255.255.0
# allow connections from any host
# between 10.0.0.0 to 10.0.15.255
10.0.0.0      255.255.240.0
```

If **ypserv** receives a request from an address that matches one of these rules, it will process the request normally. If the address fails to match a rule, the request will be ignored and a warning message will be logged. If the */var/yp/securenets* file does not exist, **ypserv** will allow connections from any host.

The **ypserv** utility also has support for Wietse Venema's *tcpwrapper* package. This allows the administrator to use the tcpwrapper configuration files (*/etc/hosts.allow* and */etc/hosts.deny*) for access control instead of */var/yp/securenets*.

Note: while both of these access control mechanisms provide some security, they, like the privileged port test, are both vulnerable to "IP spoofing" attacks.

NIS v1 compatibility

This version of **ypserv** has some support for serving NIS v1 clients. The FreeBSD NIS implementation only uses the NIS v2 protocol, however other implementations include support for the v1 protocol for backwards compatibility with older systems. The ypbind(8) daemons supplied with these systems will try to establish a binding to an NIS v1 server even though they may never actually need it (and they may persist in broadcasting in search of one even after they receive a response from a v2 server). Note that while support for normal client calls is provided, this version of **ypserv** does not handle v1 map transfer requests; consequently, it cannot be used as a master or slave in conjunction with older NIS servers that only support the v1 protocol. Fortunately, there probably are not any such servers still in use today.

NIS servers that are also NIS clients

Care must be taken when running **ypserv** in a multi-server domain where the server machines are also NIS clients. It is generally a good idea to force the servers to bind to themselves rather than allowing them to broadcast bind requests and possibly become bound to each other: strange failure modes can result if one server goes down and others are dependent upon on it. (Eventually all the clients will time out and attempt to bind to other servers, but the delay involved can be considerable and the failure mode is still present since the servers might bind to each other all over again).

Refer to the ypbind(8) man page for details on how to force it to bind to a particular server.

OPTIONS

The following options are supported by **ypserv**:

-n This option affects the way **ypserv** handles yp_match requests for the *hosts.byname* and *hosts.byaddress* maps. By default, if **ypserv** cannot find an entry for a given host in its hosts maps, it will return an error and perform no further processing. With the **-n** flag, **ypserv** will go one step further: rather than giving up immediately, it will try to resolve the hostname or address using a DNS nameserver query. If the query is successful, **ypserv** will construct a fake database record and return it to the client, thereby making it seem as though the client's yp_match request succeeded.

This feature is provided for compatibility with SunOS 4.1.x, which has brain-damaged resolver functions in its standard C library that depend on NIS for hostname and address resolution. The FreeBSD resolver can be configured to do DNS queries directly, therefore it is not necessary to enable this option when serving only FreeBSD NIS clients.

-d Cause the server to run in debugging mode. Normally, **ypserv** reports only unusual errors (access violations, file access failures) using the syslog(3) facility. In debug mode, the server does not background itself and prints extra status messages to stderr for each request that it receives. Also, while running in debug mode, **ypserv** will not spawn any additional subprocesses as it normally does when handling yp_all requests or doing DNS lookups. (These actions often take a fair amount of time to complete and are therefore handled in subprocesses, allowing the parent server process to go on handling other requests.) This makes it easier to trace the server with a debugging tool.

-h *addr*

Specify a specific address to bind to for requests. This option may be specified multiple times. If no **-h** option is specified, **ypserv** will bind to default passive address (e.g. INADDR_ANY for IPv4) for each transport.

-P *port*

Force ypserv to bind to a specific TCP/UDP port, rather than selecting its own.

-p *path*

Normally, **ypserv** assumes that all NIS maps are stored under */var/yp*. The **-p** flag may be used to specify an alternate NIS root path, allowing the system administrator to move the map files to a different place within the file system.

FILES

/var/yp/[domainname]/[maps]	the NIS maps
/etc/nsswitch.conf	name switch configuration file
/var/yp/securenets	host access control file

SEE ALSO

ypcat(1), db(3), hosts_access(5), rpc.yppasswdd(8), yp(8), ypbind(8), ypinit(8), yppush(8), ypxfr(8)

HISTORY

This version of **ypserv** first appeared in FreeBSD 2.2.

AUTHORS

Bill Paul <*wpaul@ctr.columbia.edu*>