Solaris UNIX: How to Setup DNS Client configuration / name server

by nixcraft on January 28, 2005 · 9 comments

So how do you set DNS namserver under Sun Solaris UNIX system? The resolver is a set of routines in the C library that provide access to the Internet Domain Name System (DNS). The resolver configuration file contains information that is read by the resolver routines the first time they are invoked by a process. The file is designed to be human readable and contains a list of keywords with values that provide various types of resolver information.

Internet address (in dot notation) of a name server that the resolver should query. Up to MAXNS (currently 3, see) name servers may be listed, one per keyword. If there are multiple servers, the resolver library queries them in the order listed. To configure Solaris nameserver edit or modify /etc/resolv.conf file.

DNS Client Setup

First Create the file called /etc/resolv.conf - which includes the primary and secondary DNS server IP address for Solaris system, it can be your own DNS server or your ISPs DNS server:
# touch /etc/resolv.conf

Open /etc/resolv.conf using vi text editor:

```
# vi /etc/resolv.conf
```

Add the following lines to it:

```
search nixcraft.in
nameserver 202.54.1.30
nameserver 202.54.1.18
```

Where,

- nameserver IP : It is IP of first DNS server
- search nixcraft.in : Default domain name to search. For example, if you type command nslookup www, it will search it as www.nixcraft.in

2) Enable the name resolving using DNS as follows:

```
# cp /etc/nsswitch.dns /etc/nsswitch.conf
```

3) Test new name server by running the nslookup command as follows:

```
# nslookup nixcraft.in
```

OR

```
# /usr/sfw/sbin/nslookup nixcraft.in
```

Output:

```
Server:  gwbsd2vsnl.nixcraft.in
Address:  192.168.1.202

Non-authoritative answer:
Name:    nixcraft.in
Address:  202.71.128.225
```

/etc/nsswitch.conf -- Name Service Switch File

The name service switch file determines which name services a system uses to search for information, and in which order the name services are searched. All Solaris systems use the /etc/nsswitch.conf file as the name service switch file. The nsswitch.conf file is loaded with the contents of a template file during the installation of the Solaris, depending on the name service that is selected.

The /etc/nsswitch.conf file includes a list of databases that are sources of information about IP addresses, users, and groups. Data for these can come from a variety of sources. For example, host names and host addresses, are located in the /etc/hosts file, NIS, NIS+, LDAP, or DNS. Each database has zero or more sources; the sources and their lookup order are specified in the /etc/nsswitch.conf file.

Note: The default name service during installation of the Solaris 9 is the /etc/nsswitch.nisplus template that configures the name service for NIS+.

The following example is the /etc/nsswitch.conf file configured to support the NIS name service using the /etc/nsswitch.nis template.

```
# /etc/nsswitch.nis:
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# uses NIS (YP) in conjunction with files.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet" transports.
# the following two lines obviate the "+" entry in /etc/passwd and /etc/group.
passwd: files nis
group: files nis
# consult /etc "files" only if nis is down.
hosts: nis [NOTFOUND=return] files dns
ipnodes: files
```

… … …

There are five templates that are used during Solaris installation, or when you need to change name service:

1. Local files /etc/nsswitch.files
2. DNS /etc/nsswitch.dns
3. NIS /etc/nsswitch.nis
4. NIS+ /etc/nsswitch.nisplus
5. LDAP /etc/nsswitch.ldap

Database Sources

There is an entry in the /etc/nsswitch.conf file for each database. Some typical examples of these entries are:

```
ipnodes: files
passwd: files nis
hosts: nis [NOTFOUND=return] files
```

There might be a single information source listed, in which case the search terminates if the information is not found. If two or more sources are listed, the first listed source is searched before moving on to the next listed source.

- files Specifies that entries be obtained from a file stored in the client’s /etc directory. For example, /etc/hosts.
- nisplus Specifies that entries be obtained from an NIS+ table. For example, the hosts table.
- nis Specifies that entries be obtained from an NIS map. For example, the hosts map.
- dns Specifies that host information be obtained from DNS.
- ldap Specifies that entries be obtained from the LDAP directory.
- user Specifies that printer information be obtained from the ${HOME}/.printers file

When multiple information sources are specified, it is sometimes necessary to define precisely the circumstances under which each source is searched. When a name service is referenced, the attempt to search this source can return one of the following status codes:

- SUCCESS The requested entry was found in the specified source.
- UNAVAIL The source is not configured on this system and cannot be used. In other words, the NIS or NIS+ processes could not be found or contacted.

- NOTFOUND The source responded with No such entry. In other words, the table, map, or file was accessed, but it did not contain the needed information.
- TRYAGAIN The source is busy. It might respond if tried again. In other words, the name service is running and was contacted but could not service the request at that moment.

Actions

For each status code, two actions are possible. When the action is not explicitly specified, the default action is to continue the search using the next specified information source, as follows:

- SUCCESS = return
- UNAVAIL = continue
- NOTFOUND = continue
- TRYAGAIN = continue

Examples:

1. **Example 1**

   ipnodes: files

   In this example, the /etc/inet/ipnodes file is searched for the first entry that matches the requested host name. If no matches are found, an appropriate error is returned, and no further information sources are searched.

2. **Example 2**: passwd: files nis

   In this example, the appropriate files in the /etc directory are searched for the corresponding password entry. If the entry is not found, the NIS maps are searched for the entry. If no entry is found in the NIS maps, an appropriate error is returned, and no further information sources are searched.

   - return Stop looking for the information.
   - continue Try the next source, if there is one.

3. **Example 3**:

   hosts: nis [NOTFOUND=return] files

   In this example, the NIS maps are searched for the entry. If the source (NIS) is not running, the system returns the status UNAVAIL, and continues to search the

/etc/inet/hosts file. If the entry returns the status NOTFOUND, an appropriate error is returned, and the search is terminated without searching the /etc/inet/hosts file.

nsswitch.conf - System Databases and Name Service Switch configuration file

Various functions in the C Library need to be configured to work correctly in the local environment. Traditionally, this was done by using files (e.g., `/etc/passwd'), but other nameservices (like the Network Information Service (NIS) and the Domain Name Service (DNS)) became popular, and were hacked into the C library, usually with a fixed search order.

The Linux libc5 with NYS support and the GNU C Library 2.x (libc.so.6) contain a cleaner solution of this problem. It is designed after a method used by Sun Microsystems in the C library of Solaris 2. We follow their name and call this scheme "Name Service Switch" (NSS). The sources for the "databases" and their lookup order are specified in the **/etc/nsswitch.conf** file.

The following databases are available in the NSS:

aliases
Mail aliases, used by **sendmail**(8). Presently ignored.
ethers
Ethernet numbers.
group
Groups of users, used by **getgrent**(3) functions.
hosts
Host names and numbers, used by **gethostbyname**(3) and similar functions.
netgroup
Network wide list of hosts and users, used for access rules. C libraries before glibc 2.1 only support netgroups over NIS.
network
Network names and numbers, used by **getnetent**(3) functions.
passwd
User passwords, used by **getpwent**(3) functions.
protocols
Network protocols, used by **getprotoent**(3) functions.
publickey
Public and secret keys for Secure_RPC used by NFS and NIS+.
rpc
Remote procedure call names and numbers, used by **getrpcbyname**(3) and similar functions.

services
Network services, used by getservent(3) functions.
shadow
Shadow user passwords, used by getspnam(3).

An example **/etc/nsswitch.conf** (namely, the default used when **/etc/nsswitch.conf** is missing):

```
passwd:
compat
group:
compat
shadow:
compat
hosts:
dns [!UNAVAIL=return] files
networks:
nis [NOTFOUND=return] files
ethers:
nis [NOTFOUND=return] files
protocols:
nis [NOTFOUND=return] files
rpc:
nis [NOTFOUND=return] files
services:
nis [NOTFOUND=return] files
```

The first column is the database. The rest of the line specifies how the lookup process works. You can specify the way it works for each database individually.

The configuration specification for each database can contain two different items:

```
* The service specification like `files', `db', or `nis'.
* The reaction on lookup result like `[NOTFOUND=return]'.
```

For libc5 with NYS, the allowed service specifications are `files', `nis', and `nisplus'. For hosts, you could specify `dns' as extra service, for passwd and group `compat', but not for shadow.

For glibc, you must have a file called **/lib/libnss_SERVICE.so.***X* for every SERVICE you are using. On a standard installation, you could use `files', `db', `nis', and `nisplus'. For hosts, you could specify `dns' as extra service, for passwd, group, and shadow `compat'. These services will not be used by libc5 with NYS. The version number *X* is 1 for glibc 2.0 and 2 for glibc 2.1.

The second item in the specification gives the user much finer control on the lookup process. Action items are placed between two service names and are written within brackets. The general form is

`[' ( `!'? STATUS `=' ACTION )+ `]'

where

```
STATUS => success | notfound | unavail | tryagain
ACTION => return | continue
```

The case of the keywords is insignificant. The STATUS values are the results of a call to a lookup function of a specific service. They mean:

success
No error occurred and the wanted entry is returned. The default action for this is `return'.
notfound
The lookup process works ok but the needed value was not found. The default action is `continue'.
unavail
The service is permanently unavailable. This can either mean the needed file is not available, or, for DNS, the server is not available or does not allow queries. The default action is `continue'.
tryagain
The service is temporarily unavailable. This could mean a file is locked or a server currently cannot accept more connections. The default action is `continue'.

Interaction with +/- syntax (compat mode)

Linux libc5 without NYS does not have the name service switch but does allow the user some policy control. In **/etc/passwd** you could have entries of the form +user or +@netgroup (include the specified user from the NIS passwd map), -user or -@netgroup (exclude the specified user), and + (include every user, except the excluded ones, from the NIS passwd map). Since most people only put a + at the end of **/etc/passwd** to include everything from NIS, the switch provides a faster alternative for this case (`passwd: files nis') which doesn't require the single + entry in **/etc/passwd**, **/etc/group**, and **/etc/shadow**. If this is not sufficient, the NSS `compat' service provides full +/- semantics. By default, the source is `nis', but this may be overriden by specifying `nisplus' as source for the pseudo-databases **passwd_compat, group_compat** and **shadow_compat.** This pseudo-databases are only available in GNU C Library.

A service named SERVICE is implemented by a shared object library named **libnss_SERVICE.so.***X* that resides in */lib*.

/etc/nsswitch.conf configuration file
/lib/libnss_compat.so.*X*
implements `compat' source for glibc2
/lib/libnss_db.so.*X*
implements `db' source for glibc2
/lib/libnss_dns.so.*X*
implements `dns' source for glibc2
/lib/libnss_files.so.*X*
implements `files' source for glibc2
/lib/libnss_hesiod.so.*X*
implements `hesiod' source for glibc2
/lib/libnss_nis.so.*X*
implements `nis' source for glibc2
/lib/libnss_nisplus.so.2
implements `nisplus' source for glibc 2.1

NOTES

Within each process that uses **nsswitch.conf**, the entire file is read only once; if the file is later changed, the process will continue using the old configuration.

With Solaris, it isn't possible to link programs using the NSS Service statically. With Linux, this is no problem.