Open Computing ‘‘Hands-On’’ Tutorial: July 1994

Network Information Services Plus

Sunsoft’s naming service helps deal with managing large networks of NIS-compliant systems

By Arshad Noor

To cope with the phenomenal growth of networks and the need to provide services that are easily managed by system administrators, Sunsoft Inc. has introduced the Network Information Services Plus (NIS+) with Solaris 2.x. NIS+ is a naming service that provides programs with information such as home directory, network information, remote procedure call (RPC) program name and number, socket number and protocol type.

A standalone machine looks up this information in the `/etc` directory in files such as `/etc/passwd` and `/etc/rpc`. On a small network (five to six machines), it is less effort to keep these files synchronized manually than to implement a name service. But on large networks where much of the information that needs to be shared is identical to groups of machines and people, a name service provides consistent information with less administrative effort.

NIS+ inherits characteristics of Domain Name Service (DNS) and Network Information Services (NIS), formerly known as Yellow Pages. Like NIS, NIS+ provides name services for all kinds of information--`bootparams`, `passwd`, and others--and like DNS, it allows you to provide this service in a hierarchical manner, where you can delegate administrative control to subdomains. Unlike either, it provides fine-grained security and the ability to determine what should be the primary authority for a given request, whether local files, NIS, DNS, or NIS+.

At the time of writing [Spring 1994], only machines running Solaris 2.x may be NIS+ servers because only Solaris includes NIS+. The Common Open Software Environment (COSE) group--Sun Microsystems Inc., IBM Corp., Hewlett-Packard Co., Novell Inc., and The Santa Cruz Operation Inc.--have endorsed ONC+ (Open Network Computing Plus), which includes NIS+, but have not committed to release dates. Workstations and servers running SunOS 4.x may be NIS+ clients because the distribution includes SunOS 4.x NIS+ binaries. Machines running Unix from vendors such as IBM and HP, which support NIS, may be NIS+ clients too if the NIS+ server is running in Yellow-Pages-compatibility mode.

A major change between NIS and NIS+ is the extensive security built around NIS+. By default, it uses Secure RPC, with data encryption standards (DES) encryption and Diffie-Hellmann key exchange for authentication. And it does extensive authorization checks before allowing any NIS+ operation. This procedure ensures that when NIS+ is set up and used as recommended, it is a secure environment.

In this tutorial, you will set up NIS+ for the Zoo Corp. The domain structure has a root domain called zoo.com and two subdomains, mammals.zoo.com and birds.zoo.com. All clients will be assigned to either the mammals or birds subdomain.

In Solaris 2.3, three shell scripts simplify setting up NIS+: `nisserver`, `nisclient`, and `nispopulate`. Although convenient, they are not the best way to start setting up NIS+. They hide the details of setup and thus diminish understanding. It is beneficial to set up NIS+ initially through the detailed steps so you become familiar with the utilities. Then use the shell scripts when necessary. Knowing the utilities is essential because when you need to make granular changes in NIS+, you have to use the commands that the shell scripts themselves use.

NIS+ Planning

First identify the root domain for the name space. An organization must have one domain that is the parent for the entire NIS+ hierarchy. This domain, known as the root domain, must have a name that consists of two parts separated by a period, such as xyz.com or pqr.edu. The naming convention follows DNS standards, using such extensions as ‘‘.com’’ for commercial or ‘‘.edu’’ for educational organizations. In our example, the name of the root domain will be zoo.com. A domain is analogous to a directory in a Unix file system. In fact, NIS+ describes domains as directory objects, and many of the NIS+ commands are similar to Unix commands.

Once the root domain has been named--in our example, zoo.com--recursively identify the subdomains within the parent. Each subdomain name must be unique within its parent. Most NIS+ environments will have at least one subdomain, otherwise they may as well just use NIS if security were not an issue. Having only one domain in NIS+ is, conceptually, the same as having a flat NIS domain.

Most organizations will use the root domain as a place holder for subdomains, and with few exceptions, clients (machines and human users) will be assigned to subdomains rather than the root domain. It is more logical to assign staff to sales.widgets.com or finance.widgets.com rather than to widgets.com. In our example, the subdomains are mammals.zoo.com and birds.zoo.com.

Each domain and all its objects are ‘‘owned’’ by the individual that first created it, just as Unix files are owned by their creators. This arrangement can be inconvenient over time because only the owner has administrative privileges for the domain. One solution is to create an NIS+ group that has administrative privileges on the domain and then add or remove individuals to or from this group. The concept is similar to Unix groups. The use of the name ‘‘admin’’ for administrative groups is recommended, so we’ll use admin.zoo.com, admin.mammals.zoo.com, and admin.birds.zoo.com.

Decide on the machines that will serve as masters and replicas for each domain. Each NIS+ domain must have only one master server and, optionally, one or more replica servers that back it up. Have at least one replica server per domain. A server may serve more than one domain, but if you can, use separate machines to serve as masters for distinct domains. A replica either must be in the same domain as the master or in a parent domain.

In our example, we will use the machines shown in Table 1. With the exception of the machines that will become servers for subdomains and replicas, the next step--identifying the clients of each domain--may be deferred until the domain hierarchy is set up. Remember that NIS+ servers of a subdomain are clients of its parent domain (with the exception of the root master, which is a client of its own domain). NIS+ clients can be machines or human users. All names must be unique within a domain and must conform to the Unix naming standards. Clients to be set up are also shown in Table 1.

NIS+ Security

Based on the type of client machines that will be installed, decide on the security policy to be used for each domain. NIS+ allows for three modes of security when running the daemon (`rpc.nisd`) that provides NIS+ services. These are levels 0, 1, and 2. Level 0 is used only for setup, Level 1 for debugging purposes, and Level 2 as the default mode with full security, which is the recommended operating mode. This mode can be used even when running in NIS-compatibility mode with non-Solaris 2.x machines.

An NIS+ object, such as a table or a column within a table, has four groups of privileges, each group consisting of four privileges. The four groups are User, Group, World, and Nobody. The four privileges are Read, Modify, Create, and Delete. The privileges work in exactly the same way as Unix file system privileges. The privileges on an object are depicted as `rmcdrmcdrmcdrmcd` where the first group of ''rmcd'' is for Nobody and the remaining three groups of rmcd are for User, Group, and World respectively. The default permissions on an object in NIS+ are `----rmcdrmcdr---`, which indicates that any unauthenticated request (Nobody) has no privileges on the object, while User and Group have Read, Modify, Create, and Delete privileges, and World has only Read permissions.

If you have a mixed Unix environment--HP, IBM, Digital Equipment Corp., and others--that supports NIS, then you must create the NIS+ objects and run `rpc.nisd` in NIS-compatibility mode. This step has the effect of diminishing the default security of the NIS+ environment--not by reducing the security level of `rpc.nisd`, but by making all objects in NIS+ readable by the entity Nobody or any unauthenticated request. The effective permissions of NIS+ objects in NIS-compatibility mode are now `r---rmcdrmcdr---`.

Setting up NIS+ in NIS-compatibility mode also forces the display of encrypted passwords from the passwd table so that NIS clients may validate their log-ins. This act reverses the secure policy that the shadow password file (`/etc/shadow`) imposed on System V Release 3.2 and System V Release 4 machines. In a pure NIS+ environment, the password field is indeed hidden as with the conventional `/etc/shadow` file.

NIS+ Root Server Implementation

Once you've taken care of the planning tasks, you can continue with the implementation:

Begin by establishing the domain name of the root server by creating the file `/etc/defaultdomain` and populate it with the domain name. You can set the domain name by running the command line: `domainname zoo.com`. You ensure using the correct `/etc/nsswitch.conf` file by copying the `/etc/nsswitch.nisplus` file over `/etc/nsswitch.conf`. Then make certain your `/var/nis` directory is empty (unless you have information from a previous attempt that you want to preserve). Set the `NIS_GROUP` environment variable, which establishes group ownership of the objects.

Add the `/usr/lib/nis` directory to your path so that your shell finds all NIS+ executables. Next you initialize NIS+ by creating the root directory of the NIS+ name space using `nisinit -r`. This command creates the files in `/var/nis` that establish the root domain. You have to start up the NIS+ server with security turned off (the `-S 0` option). The security must be disabled while you are setting up authentication (Listing 1).

Set up the NIS+ objects using the `nissetup` shell script (using the `-Y` flag for NIS compatibility), which creates the Table objects in a Directory object called `org_dir.zoo.com` (`org_dir` is not a subdomain but a container for the Table objects). Next create the administrative group for the root domain and allow this group to have full privileges on the NIS+ objects. Then add the machine on which you are logged in to the administrative group. Create credentials for yourself and update the objects created so far with the credentials. Start the NIS+ cache daemon and, after killing the NIS+ server--`rpc.nisd`, which is currently running with no security--restart it with Level 2 security. This step sets up the root domain and its server.

Populating NIS+ Tables

The `nissetup` shell script only created empty tables in the NIS+ name space, which now have to be populated. This process can be done by using the `nisaddent` or the `nistbladm` utility. The former allows you to add information from files or from NIS maps, while the latter allows you to add it one record at a time (Listing 2). The `/etc/shadow` file is added to the `passwd.org_dir` table because no such table exists. You can also use the Solaris `admintool` to add information a record at a time.

Next add credentials for administrators who will have privileges on the NIS+ name space. See Listing 3A on how to add local and DES credentials for users. Bear in mind that adding credentials does not automatically create entries in the `passwd` table. The `cred` table only authenticates users after they are logged onto a Unix machine, for which a passwd entry is still required. This entry must be added separately. You can accomplish both at one time by using `admintool`. Now add the user to the administrative group that manages NIS+, using the `nisgrpadm` command (Listing 3B).

Creating NIS+ credentials for client machines is accomplished in the same way as setting up DES credentials for a user (machines do not have local credentials in NIS+), as shown in Listing 3C. Every time a client machine boots up, it authenticates itself to the NIS+ servers using these credentials.

Setting Up NIS+ Client Machines

You have now finished adding information to the NIS+ tables. Next you set up clients in the root domain that will become servers for the subdomains. The tasks are similar to setting up the root domain, but there are some differences. Repeat the tasks as for the root master in setting up the domain name, using the correct `/etc/nsswitch.conf` file and cleaning out the `/var/nis` directory.

NIS+ allows you to initialize clients using one of three methods: the broadcast method, which is the least secure of all and therefore not recommended; the host name method, which is more secure than broadcast and can be used if you know your network can be trusted, that is, you know nobody will be masquerading as the host name you specify; or the cold start method, which is the most secure and recommended method (Listing 4A).

You copy the `NIS_COLD_START` file from the `/var/nis` directory of the master server to the client and then initialize the client using this file. Kill and restart the `keyserv` process and explicitly `keylogin` with the `-r` option to save the private key of the client machine locally in `/etc/.rootkey`. Finally, reboot the client workstation. This process effectively sets up NIS+ clients. (See Listing 4B.)

NIS+ Subdomains

Now that the clients that will become servers of subdomains have been set up, the task of setting up subdomains can begin. This step is accomplished on the client that will become the master of the subdomain by first setting the `NIS_GROUP` and creating the subdomain with the `nismkdir` command. The necessity for this step happens only when the machine serving as root master serves more than one domain. In any other situation, the client becoming a subdomain master would be a separate machine. Use the `-m` option to identify the master server for the new domain.

You then create the tables within the subdomain using `nissetup` and the administrative group using `nisgrpadm`. After you grant the administrative group privileges on the subdomain, you add credentials for new administrative users to the administrative group, just as you did in the root domain. Start the `rpc.nisd` daemon on the subdomain master. (See Listing 5.)

NIS+ Replica Servers

Setting up replica servers can be accomplished only after a client has been set up and the client has been converted to a server. This step is accomplished by starting `rpc.nisd` on the machine. Once done, the replica is created using the `nismkdir` command with the `-s` option. Then propagate the tables and the information in it by ‘‘pinging’’ the replica with the `nisping` command (Listing 6).

Setting up clients within the various domains remains the only task. The `nisaddcred` command does this task as shown in Listing 3. More explanations and details can be found in the Administering NIS+ and DNS document that is part of the *Systems Administrator Reference AnswerBook* CD-ROM, or the hard-copy documentation for Solaris 2.3. This tutorial just begins to touch on NIS+. There are a whole host of services that can be provided by NIS+ with a little imagination and some time.

Go To Content   Search

Last Modified: Thursday, 30-Nov-95 16:53:10 PST