DNS Basics

The Domain Name System (DNS) is an application-layer protocol that is part of the standard TCP/IP protocol suite. This protocol implements the DNS name service, which is the name service used on the Internet.

This section introduces the basic DNS concepts. It assumes that you have some familiarity with network administration, particularly TCP/IP, and some exposure to other name services, such as NIS+ and NIS.

Refer to *Solaris Naming Setup and Configuration Guide* for information regarding initial setup and configuration of DNS.

Note -

DNS, NIS+, NIS, and FNS provide similar functionality and sometimes use the same terms to define different entities. Thus, this chapter takes care to define terms like domain and name server according to their DNS functionality, a very different functionality than NIS+ and NIS domains and servers.

Name-to-Address Resolution

Though it supports the complex, world-wide hierarchy of computers on the Internet, the basic function of DNS is actually very simple: providing **name-to-address resolution** for TCP/IP-based networks. Name-to-address resolution, also referred to as "mapping," is the process of finding the IP address of a computer in a database by using its host name as an index.

Name-to-address mapping occurs when a program running on your local machine needs to contact a remote computer. The program most likely will know the host name of the remote computer but may not know how to locate it, particularly if the remote machine is in another company, miles from your site. To get the remote machine's address, the program requests assistance from the DNS software running on your local machine, which is considered a **DNS client**.

Your machine sends a request to a **DNS name server**, which maintains the distributed DNS database. The files in the DNS database bear little resemblance to the NIS+ `host` or `ipnodes` Table or even the local `/etc/hosts` or `/etc/inet/ipnodes` file, though they maintain similar information: the host names, the ipnode names, IPv4 and IPv6 addresses, and other information about a particular group of computers. The name server uses the host name your machine sent as part of its request to find or "resolve" the IP address of the remote machine. It then returns this IP address to your local machine IF the host name is in its DNS database.

Figure 28-1 shows name-to-address mapping as it occurs between a DNS client and a name server, probably on the client's local network.

Figure 28-1 Name to Address Resolution

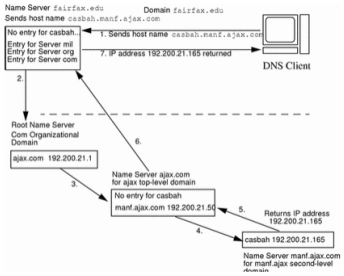
If the host name is not in that name server's DNS database, this indicates that the machine is outside of its authority, or, to use DNS terminology, outside the **local administrative domain**. Thus, each name server is spoken of as being "authoritative" for its local administrative domain.

Fortunately, the local name server maintains a list of host names and IP addresses of **root domain name servers**, to which it will forward the request from your machine. These root name servers are authoritative for huge organizational domains, as explained fully in "DNS Hierarchy and the Internet". These hierarchies resemble UNIX file systems, in that they are organized into an upside-down tree structure.

Each root name server maintains the host names and IP address of top level domain name servers for a company, a university, or other large organizations. The root name server sends your request to the top-level name servers that it knows about. If one of these servers has the IP address for the host you requested, it will return the information to your machine. If the top-level servers do not know about the host you requested, they pass the request to second-level name servers for which they maintain information. Your request is then passed on down through the vast organizational tree. Eventually, a name server that has information about your requested host in its database will return the IP address back to your machine.

Figure 28-2 shows name-to-address resolution outside the local domain.

Figure 28-2 Name to Address Resolution for a Remote Host



DNS Administrative Domains

From a DNS perspective, an **administrative domain** is a group of machines that are administered as a unit. Information about this domain is maintained by at least two name servers; they are "authoritative" for the domain. The DNS domain is a purely logical grouping of machines. It could correspond to a physical grouping of machines, such as all machines attached to the Ethernet in a small business. But a local DNS domain just as likely could include all machines on a vast university network that belong to the computer science department or to university administration.

For example, suppose the Ajax company has two sites, one in San Francisco and one in Seattle. The `Retail.Sales.Ajax.com.` domain might be in Seattle and the `Wholesale.Sales.Ajax.com.` domain might be in San Francisco. One part of the `Sales.Ajax.com.` domain would be in one city, the other part in the second city.

Each administrative domain must have its own unique subdomain name. Moreover, if you want your network to participate in the Internet, the network must be part of a registered administrative domain. The section "Joining the Internet" has full details about domain names and domain registration.

`in.named` and DNS Name Servers

As mentioned previously, name servers in an administrative domain maintain the DNS database. They also run the `in.named` daemon, which implements DNS services, most significantly, name-to-address mapping. `in.named` is a public domain TCP/IP program and included with the Solaris operating environment.

Note -

The `in.named` daemon is also called the Berkeley Internet Name Domain service, or BIND, because it was developed at the University of California at Berkeley.

There are three types of DNS name servers:

- Primary server
- Secondary server
- Cache-only server

Each domain must have one primary server and should have at least one secondary server to provide backup. "Zones" explains primary and secondary servers in detail.

DNS Clients and the Resolver

To be a DNS client, a machine must run the **resolver**. The resolver is neither a daemon nor a single program; rather, it is a set of dynamic library routines used by applications that need to know machine names. The resolver's function is to resolve users' queries. To do that, it queries a name server, which then returns either the requested information or a referral to another server. Once the resolver is configured, a machine can request DNS service from a name server.

When a machine's `/etc/nsswitch.conf` file specifies `hosts: dns` (or any other variant that includes `dns` in the `hosts` line), the resolver libraries are automatically used. If the `nsswitch.conf` file specifies some other name service before `dns`, that name service is consulted first for host information and only if that name service does not find the host in question are the resolver libraries used.

For example, if the hosts line in the `nsswitch.conf` file specifies `hosts: nisplus dns`, the NIS+ name service will first be searched for host information. If the information is not found in NIS+, then the DNS resolver is used. Since name services such as NIS+ and NIS only contain information about hosts in their own network, the effect of a `hosts:nisplus dns` line in a switch file is to specify the use of NIS+ for local host information and DNS for information on remote hosts out on the Internet.

There are two kinds of DNS clients:

IPR2025-01400
CrowdStrike EX1030 Page 1

There are two kinds of DNS clients:

- *Client-only*. A client-only DNS client does not run `in.named`; instead, it consults the resolver. The resolver knows about a list of name servers for the domain, to which queries are then directed.
- *Client-server*. A client-server uses the services provided by `in.named` to resolve queries forwarded to it by client-machine resolvers.

The Solaris operating environment includes the dynamic library routines that make up the resolver. *Solaris Naming Setup and Configuration Guide*, contains instructions for setting up a host as a DNS client.

- *Previous*: Part V Administering FNS
- *Next*: Introducing the DNS Namespace

- © 2010, Oracle Corporation and/or its affiliates

Cookie Preferences | Ad Choices