Configuring DNS

The Domain Name System (DNS) is a distributed database in which you can map hostnames to IP addresses through the DNS protocol from a DNS server. Each unique IP address can have an associated hostname. The Cisco IOS software maintains a cache of hostname-to-address mappings for use by the **connect**, **telnet**, and **ping** EXEC commands, and related Telnet support operations. This cache speeds the process of converting names to addresses.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring DNS

To use DNS, you must have a DNS name server on your network.

Information About DNS

DNS Overview

If your network devices require connectivity with devices in networks for which you do not control name assignment, you can assign device names that uniquely identify your devices within the entire internetwork. The global naming scheme of the Internet, the DNS, accomplishes this task. This service is enabled by default. The following sections summarize DNS concepts and function.

Hostnames for Network Devices

Each unique IP address can have an associated hostname. DNS uses a hierarchical scheme for establishing hostnames for network nodes. This allows local control of the segments of the network through a client-server scheme. The DNS system can locate a network device by translating the hostname of the device into its associated IP address.

Domains Names for Groups of Networks

IP defines a naming scheme that allows a device to be identified by its location in the IP. This is a hierarchical naming scheme that provides for *domains*. On the Internet, a domain is a portion of the naming hierarchy tree that refers to general groupings of networks based on organization type or geography. Domain names are pieced together with periods (.) as the delimiting characters. For example, Cisco is a commercial organization that the IP identifies by a *com* domain name, so its domain name is *cisco.com*. A specific device in this domain, the File Transfer Protocol (FTP) system, for example, is identified as *ftp.cisco.com*.

Name Servers

To keep track of domain names, IP has defined the concept of a *name server*. Name servers are programs that have complete information about their namespace portion of the domain tree and may also contain pointers to other name servers that can be used to lead to information from any other part of the domain tree. Name servers know the parts of the domain tree for which they have complete information. A name server may also store information about other parts of the domain tree. Before domain names can be mapped to IP addresses, you must first identify the hostnames, then specify a name server, and enable the DNS service.

Cache

To speed the process of converting names to addresses, the name server maintains a database, called a *cache*, of hostname-to-address mappings for use by the **connect**, **telnet**, and **ping** EXEC commands, and related Telnet support operations. The cache stores the results from previous responses. Upon receiving a client-issued DNS query, the name server will check this local storage to see if the answer is available locally.

Name Resolvers

Name resolvers are programs that extract information from name servers in response to client requests. Resolvers must be able to access at least one name server. The resolver either uses that name server's information

to answer a query directly or pursues the query using referrals to other names servers. A resolver will typically be a system routine that is directly accessible to user programs. Therefore, no protocol is necessary between the resolver and the user program.

Zones

The domain namespace is divided into areas called zones that are points of delegation in the DNS tree. A zone contains all domains from a certain point downward, except those for which other zones are authoritative.

Authoritative Name Servers

A name server is said to be an authority for the parts of the domain tree for which it has complete information. A zone usually has an authoritative name server, often more than one. An authoritative name server has been configured with host table information or has acquired host table information though a zone transfer (the action that occurs when a secondary DNS server starts up and updates itself from the primary server).

DNS Operation

An organization can have many name servers, but Internet clients can query only those that the root name servers know. The other name servers answer internal queries only.

A name server handles client-issued queries to the DNS server for locally defined hosts within a particular zone as follows:

- An authoritative name server responds to DNS user queries for a domain name that is under its zone of authority by using the permanent and cached entries in its own host table. If the query is for a domain name that is under its zone of authority but for which it does not have any configuration information, the authoritative name server simply replies that no such information exists.
- A name server that is not configured as the authoritative name server responds to DNS user queries by using information that it has cached from previously received query responses. If no router is configured as the authoritative name server for a zone, queries to the DNS server for locally defined hosts will receive nonauthoritative responses.

Name servers answer DNS queries (forward incoming DNS queries or resolve internally generated DNS queries) according to the forwarding and lookup parameters configured for the specific domain.

When DNS queries are forwarded to name servers for resolution, some memory space is held for the corresponding DNS query until an appropriate response is received or until there is timeout. To avoid the free I/O memory from getting exhausted when handling queries at high rate, configure the maximum size for the queue.

How to Configure DNS

Mapping Hostnames to IP Addresses

Perform this task to map hostnames to IP addresses.

A name server is used to keep track of information associated with domain names. A name server can maintain a database of hostname-to-address mappings. Each name can map to one or more IP addresses. In order to use this service to map domain names to IP addresses, you must specify a name server.

The name lookup system can be statically configured using the commands described in this task. Some other functions in Cisco IOS software, such as DHCP, can dynamically modify the state of the name lookup system. Use the **show hosts** command to display the cached hostnames and the DNS configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip host** *name* [*tcp-port-number*] *address1* [*address2 ... address8*]
4. Do one of the following:
   - **ip domain name** *name*
   -
   - **ip domain list** *name*
5. **ip name-server** *server-address1* [*server-address2 ... server-address6*]
6. **ip domain lookup** [**source-interface** *interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip host** *name* [*tcp-port-number*] *address1* [*address2 ... address8*] **Example:** `Router(config)# ip host cisco-rtp 192.168.0.148`	Defines a static hostname-to-address mapping in the hostname cache. • Typically, it is easier to refer to network devices by symbolic names rather than numerical addresses (services such as Telnet can use hostnames or addresses). Hostnames and IP addresses can be associated with one another through static or dynamic means. • Manually assigning hostnames to addresses is useful when dynamic mapping is not available.
Step 4	Do one of the following: • **ip domain name** *name*	(Optional) Defines a default domain name that the Cisco IOS software will use to complete unqualified hostnames. or

	Command or Action	Purpose

	•
	• **ip domain list** *name*
	
	Example:
	`Router(config)# ip domain name cisco.com`
	
	Example:
	
	Example:
	`Router(config)# ip domain list cisco1.com`
		(Optional) Defines a list of default domain names to complete unqualified hostnames.
		• You can specify a default domain name that the Cisco IOS software will use to complete domain name requests. You can specify either a single domain name or a list of domain names. Any hostname that does not contain a complete domain name will have the default domain name you specify appended to it before the name is looked up.
		
		Note If there is no domain list, the domain name that you specified with the **ip domain name** global configuration command is used. If there is a domain list, the default domain name is not used. The **ip domain list** command is similar to the **ip domain name** command, except that with the **ip domain list** command you can define a list of domains, each to be tried in turn until the system finds a match.

Step 5	**ip name-server** *server-address1* [*server-address2 ... server-address6*] **Example:** `Router(config)# ip name-server 172.16.1.111 172.16.1.2`	Specifies one or more hosts (up to six) that can function as a name server to supply name information for DNS.
Step 6	**ip domain lookup** [**source-interface** *interface-type interface-number*] **Example:** `Router(config)# ip domain lookup`	(Optional) Enables DNS-based address translation. • DNS is enabled by default. Use this command if DNS has been disabled.

Customizing DNS

Perform this task to customize your DNS configuration.

In a multiple server configuration without the DNS round-robin functionality, many programs will use the first host server/IP address for the whole time to live (TTL) of the cache and use the second and third host servers/IP addresses only in the event of host failure. This behavior presents a problem when a high volume of users all arrive at the first host during the TTL time. For example, the network access server (NAS) sends out a DNS query. The DNS servers reply with a list of the configured IP addresses to the NAS. The NAS then caches these IP addresses for a given time (for example, five minutes). All users that dial in during the five minute TTL time will land on one host, the first IP address in the list.

In a multiple server configuration with the DNS round-robin functionality, the DNS server returns the IP address of all hosts to rotate between the cache of hostnames. During the TTL of the cache, users are distributed

among the hosts. This functionality distributes calls across the configured hosts and reduces the number of DNS queries.

In a scheduling algorithm, processes are activated in a fixed cyclic order. Processes that are waiting for other events, like termination of a child process or an input or output operation, cannot proceed and hence they return control to the scheduler. If the TTL of the process times out just before the event (for which it was waiting) occurs, then the event will not be handled until all the other processes are activated.

Note The DNS round-robin functionality is applicable only for the DNS lookups on a router and is not applicable to another client pointing to the router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip domain timeout** *seconds*
4. **ip domain retry** *number*
5. **ip domain round-robin**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip domain timeout** *seconds* **Example:** `Router(config)# ip domain timeout 17`	(Optional) Specifies the amount of time to wait for a response to a DNS query. • If the **ip domain timeout** command is not configured, the Cisco IOS software will wait 3 seconds for a response to a DNS query.
Step 4	**ip domain retry** *number* **Example:** `Router(config)# ip domain retry 10`	(Optional) Specifies the number of times to retry sending DNS queries. • If the **ip domain retry** command is not configured, the Cisco IOS software will retry DNS queries twice.

	Command or Action	Purpose
Step 5	**ip domain round-robin** **Example:** `Router(config)# ip domain round-robin`	(Optional) Enables round-robin functionality on DNS servers.

Configuring DNS Spoofing

Perform this task to configure DNS spoofing.

DNS spoofing is designed to allow a router to act as a proxy DNS server and "spoof" replies to any DNS queries using either the configured IP address in the **ip dns spoofing** *ip-address* command or the IP address of the incoming interface for the query. This feature is useful for devices where the interface toward the Internet service provider (ISP) is not up. Once the interface to the ISP is up, the router forwards DNS queries to the real DNS servers.

This feature turns on DNS spoofing and is functional if any of the following conditions are true:

- The **no ip domain lookup** command is configured.
- IP name server addresses are not configured.
- There are no valid interfaces or routes for sending to the configured name server addresses.

If these conditions are removed, DNS spoofing will not occur.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dns server**
4. **ip dns spoofing** [*ip-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dns server** **Example:** `Router(config)# ip dns server`	Activates the DNS server on the router.
Step 4	**ip dns spoofing** [*ip-address*] **Example:** `Router(config)# ip dns spoofing 192.168.15.1`	Configures DNS spoofing. • The router will respond to the DNS query with the configured *ip-address* when queried for any hostname other than its own. • The router will respond to the DNS query with the IP address of the incoming interface when queried for its own hostname.

Configuring the Router as a DNS Server

Perform this task to configure the router as a DNS server.

A Cisco IOS router can provide service to DNS clients, acting as both a caching name server and as an authoritative name server for its own local host table.

When configured as a caching name server, the router relays DNS requests to other name servers that resolve network names into network addresses. The caching name server caches information learned from other name servers so that it can answer requests quickly, without having to query other servers for each transaction.

When configured as an authoritative name server for its own local host table, the router listens on port 53 for DNS queries and then answers DNS queries using the permanent and cached entries in its own host table.

An authoritative name server usually issues zone transfers or responds to zone transfer requests from other authoritative name servers for the same zone. However, the Cisco IOS DNS server does not perform zone transfers.

When it receives a DNS query, an authoritative name server handles the query as follows:

- If the query is for a domain name that is not under its zone of authority, the authoritative name server determines whether to forward the query to specific back-end name servers based on whether IP DNS-based hostname-to-address translation has been enabled via the **ip domain lookup** command.
- If the query is for a domain name that is under its zone of authority and for which it has configuration information, the authoritative name server answers the query using the permanent and cached entries in its own host table.

- If the query is for a domain name that is under its zone of authority but for which it does not have any configuration information, the authoritative name server does not forward the query elsewhere for a response; instead the authoritative name server simply replies that no such information exists.

Note Unless Distributed Director is enabled, the TTL on locally defined resource records will always be ten seconds, regardless of any authority record parameters that may have been specified for the DNS name server by the use of the **ip dns primary** command.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dns server**
4. **ip name-server** *server-address1* [*server-address2... server-address6*]
5. **ip dns server queue limit** {**forwarder** *queue-size-limit* | **director** *queue-size-limit*}
6. **ip host** [**vrf** *vrf-name*] [**view** *view-name*] *hostname* {*address1* [*address2 ... address8*] | **additional** *address9* [*address10 ... addressn*]}
7. **ip dns primary** *domain-name* **soa** *primary-server-name mailbox-name* [*refresh-interval* [*retry-interval* [*expire-ttl* [*minimum-ttl*]]]]
8. **ip host** *domain-name* **ns** *server-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dns server** **Example:** `Router(config)# ip dns server`	Enables the DNS server.
Step 4	**ip name-server** *server-address1* [*server-address2... server-address6*]	(Optional) Configures other DNS servers: • Cisco IOS resolver name servers

	Command or Action	Purpose
	Example: `Router(config)# ip name-server 192.168.2.120 192.168.2.121`	• DNS server forwarders **Note** If the Cisco IOS name server is being configured to respond only to domain names for which it is authoritative, there is no need to configure other DNS servers.
Step 5	**ip dns server queue limit** {**forwarder** *queue-size-limit* | **director** *queue-size-limit*} **Example:** `Router(config)# ip dns server queue limit forwarder 10`	(Optional) Configures a limit to the size of the queues used by the DNS server processes. • The **director** keyword was removed in Cisco IOS Release 12.4(24)T.
Step 6	**ip host** [**vrf** *vrf-name*] [**view** *view-name*] *hostname* {*address1* [*address2 ... address8*] | **additional** *address9* [*address10 ... addressn*]} **Example:** `Router(config)# ip host user1.example.com 192.168.201.5 192.168.201.6`	(Optional) Configures local hosts.
Step 7	**ip dns primary** *domain-name* **soa** *primary-server-name mailbox-name* [*refresh-interval* [*retry-interval* [*expire-ttl* [*minimum-ttl*]]]] **Example:** `Router(config)# ip dns primary example.com soa ns1.example.com mb1.example.com`	Configures the router as the primary DNS name server for a domain (zone) and as the start of authority (SOA) record source (which designates the start of a zone). **Note** Unless Distributed Director is enabled, the TTL on locally defined resource records will always be ten seconds.
Step 8	**ip host** *domain-name* **ns** *server-name* **Example:** `Router(config)# ip host example.com ns ns1.example.com`	(Optional) Configures the router to create an name server (NS) resource record to be returned when the DNS server is queried for the associated domain. • This configuration is needed only if the zone for which the system is authoritative will also be served by other name servers.

Examples

This section provides examples of debugging output that is logged when a router is configured as an authoritative name server for its own local host table and the **debug domain** command is in effect:

Note For DNS-based X.25 routing, the **debug x25 events** command supports functionality to describe the events that occur while the X.25 address is being resolved to an IP address using a DNS server. The **debug domain** command can be used along with **debug x25 events** to observe the whole DNS-based X.25 routing data flow.

Debugging Output for Relaying a DNS Query to Another Name Server Example

The following is sample output from the **debug domain** command that corresponds to relaying a DNS query to another name server when the router is configured as an authoritative name server for its own local host table:

```
Apr  4 22:18:32.183: DNS: Incoming UDP query (id#18713)
Apr  4 22:18:32.183: DNS: Type 1 DNS query (id#18713) for host 'ns1.example.com' from
192.0.2.120(1283)
Apr  4 22:18:32.183: DNS: Re-sending DNS query (type 1, id#18713) to 192.0.2.121
Apr  4 22:18:32.211: DNS: Incoming UDP query (id#18713)
Apr  4 22:18:32.211: DNS: Type 1 response (id#18713) for host <ns1.example.com> from
192.0.2.121(53)
Apr  4 22:18:32.215: DOM: dom2cache: hostname is ns1.example.com, RR type=1, class=1,
ttl=86400, n=4
Apr  4 22:18:32.215: DNS: Forwarding back A response - no director required
Apr  4 22:18:32.215: DNS: Finished processing query (id#18713) in 0.032 secs
Apr  4 22:18:32.215: DNS: Forwarding back reply to 192.0.2.120/1283
```

Debugging Output for Servicing a DNS Query from the Local Host Table Example

The following is sample output from the **debug domain** command that corresponds to servicing a DNS query from the local host table when the router is configured as an authoritative name server for its own local host table:

```
Apr  4 22:16:35.279: DNS: Incoming UDP query (id#8409)
Apr  4 22:16:35.279: DNS: Type 1 DNS query (id#8409) for host 'ns1.example.com' from
192.0.2.120(1279)
Apr  4 22:16:35.279: DNS: Finished processing query (id#8409) in 0.000 secs
```

Disabling DNS Queries for ISO CLNS Addresses

Perform this task to disable DNS queries for International Organization for Standardization (ISO) Connectionless Network Service (CLNS) addresses.

If your router has both IP and ISO CLNS enabled and you want to use ISO CLNS network service access point (NSAP) addresses, you can use the DNS to query these addresses, as documented in RFC 1348. This feature is enabled by default.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip domain lookup nsap**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no ip domain lookup nsap** **Example:** `Router(config)# no ip domain lookup nsap`	Disables DNS queries for ISO CLNS addresses.

Verifying DNS

Perform this task to verify your DNS configuration.

1. **enable**
2. **ping** *hosts*
3. **show hosts**

SUMMARY STEPS

1. **enable**
2. **ping** *hosts*
3. **show hosts**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**ping** *hosts* **Example:** `Router# ping cisco-rtp`	Diagnoses basic network connectivity. • After the DNS configuration is set, you can verify the DNS server by using a hostname to ping or telnet to a device.
Step 3	**show hosts** **Example:** `Router# show hosts`	Displays the default domain name, the style of name lookup service, a list of name server hosts, and the cached list of hostnames and addresses. • After a name is resolved using DNS, use the **show hosts** command to view the cached hostnames and the DNS configuration.

Configuration Examples for DNS

IP Addresses Example

The following example establishes a domain list with several alternate domain names:

```
ip domain list example.com
ip domain list example1.edu
ip domain list example2.edu
```

Mapping Hostnames to IP Addresses Example

The following example configures the hostname-to-address mapping process. IP DNS-based translation is specified, the addresses of the name servers are specified, and the default domain name is given.

```
! IP DNS-based hostname-to-address translation is enabled
ip domain lookup
! Specifies hosts 192.168.1.111 and 192.168.1.2 as name servers
ip name-server 192.168.1.111 192.168.1.2
! Defines cisco.com as the default domain name the router uses to complete
! Set the name for unqualified hostnames
ip domain name cisco.com
```

Customizing DNS Example

The following example allows a Telnet to company.example.com to connect to each of the three IP addresses specified in the following order: the first time the hostname is referenced, it would connect to 10.0.0.1; the second time the hostname is referenced, it would connect to 10.1.0.1; and the third time the hostname is

referenced, it would connect to 10.2.0.1. In each case, the other two addresses would also be tried if the first one failed; this is the normal operation of the Telnet command.

```
Router(config)# ip host company.example.com 10.0.0.1 10.1.0.1 10.2.0.1
Router(config)# ip domain round-robin
```

Configuring DNS Spoofing Example

In the following example, the router is configured to spoof replies to any DNS queries:

```
ip dns server
ip dns spoofing
no ip domain lookup
interface e3/1
 ip address 10.1.1.1 255.255.255.0
```

Additional References

Related Documents

Related Topic	Document Title
DNS commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this functionality.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 1348	*DNS NSAP RRs*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for DNS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for DNS

Feature Name	Releases	Feature Information
DNS Spoofing	12.3(2)T	This feature is designed to allow a router to act as a proxy DNS server and “spoof” replies to any DNS queries using either the configured IP address in the **ip dns spoofing** *ip-address* command or the IP address of the incoming interface for the query. The following command was introduced by this feature: **ip dns spoofing**.