Configure DNS on Routers

Contents

Introduction

Prerequisites

- Requirements
- Components Used
- Conventions

Set Up a Router to Use DNS Lookups

- Troubleshoot

You Can Ping a Web Server, But You Cannot View the HTML Pages

Router Queries Multiple Name Servers

Related Information

Introduction

This document describes how to configure Domain Name System (DNS) for Cisco routers.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco IOS®Command Line Interface (CLI)
- General DNS behavior

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

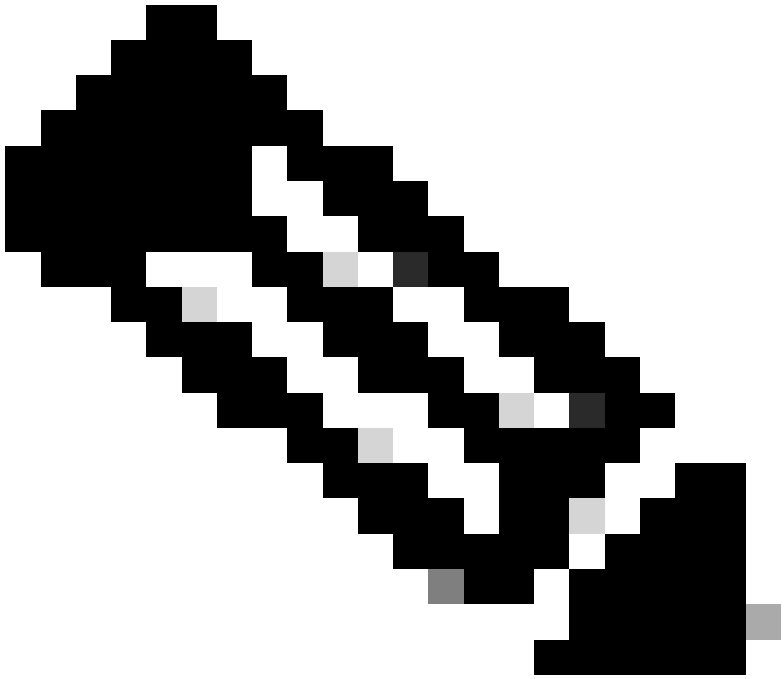
Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

Set Up a Router to Use DNS Lookups

Your router can be configured to use DNS lookups if you wish to use the ping or **traceroute** commands with a host name rather than an IP address. Use these commands to do so:

Command	Description

ip domain lookup	Enables DNS-based host name-to-address translation. This command is enabled by default.
ip name-server	Specifies the address of one or more name servers.
ip domain list	Defines a list of domains, each to be tried in turn.  **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.
ip domain name	Defines a default domain name that the Cisco IOS software uses to complete unqualified host names (names without a dotted-decimal domain name). Do not include the initial period that separates an unqualified name from the domain name.
ip ospf name-lookup	Configures Open Shortest Path First (OSPF) to look up DNS names for use in all OSPF **show EXEC** command displays. This feature makes it easier to identify a router because the router is displayed by name rather than by its router ID or neighbor ID.

This example shows a sample configuration on a router configured for basic DNS lookup:

Sample Basic DNS Lookup Configuration

```
<#root>

Router#

show running-config

Building configuration...
```

```
Current configuration : 3922 bytes
!
! Last configuration change at 16:24:57 UTC Fri May 12 2023
!
version 17.3
service timestamps debug datetime msec
service timestamps log datetime msec
! Call-home is enabled by Smart-Licensing.
service call-home
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
platform console serial
!
hostname Router
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
!
!
!
!
!
!
!
ip name-server 192.168.1.1



!--- Configures the IP address of the name server.
!--- Domain lookup is enabled by default.

!
!
interface GigabitEthernet1
 ip address 192.168.1.10 255.255.255.0
 negotiation auto
 no mop enabled
 no mop sysid
!
!

!--- Output Suppressed.

end
```

<#root>

Router#

ping www.cisco.com

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.37.145.84, timeout is 2 seconds:
```

```
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Router#
```

Troubleshoot

Under rare conditions, you can see one of these error conditions:

```
<#root>

Router#

debug ip udp

UDP packet debugging is on
Router#

ping www.cisco.com

*Mar  8 06:26:41.732: UDP: sent src=10.69.16.66(5476), dst=

10.250.35.250(53)

, length=59
*Mar  8 06:26:44.740: UDP: sent src=10.69.16.66(5476), dst=10.250.35.250(53), length=59
*Mar  8 06:26:47.744: UDP: sent src=10.69.16.66(5476), dst=10.250.35.250(53), length=59

% Unrecognized host or address, or protocol not running.


Router#undebug all
All possible debugging has been turned off

Router#

ping www.cisco.com

Translating "www.cisco.com"...domain server (172.16.249.4) ¡¦
Not process



Router#

ping www.cisco.com


*May 12 16:48:36.302: Reserved port 43478 in Transport Port Agent for UDP IP type 1
*May 12 16:48:36.302: UDP: sent src=0.0.0.0(43478), dst=

255.255.255.255(53)

, length=50
*May 12 16:48:37.303: Reserved port 56191 in Transport Port Agent for UDP IP type 1
*May 12 16:48:37.303: UDP: sent src=0.0.0.0(56191), dst=255.255.255.255(53), length=50
*May 12 16:48:37.304: Released port 43478 in Transport Port Agent for IP type 1
*May 12 16:48:37.304: Released port 43478 in Transport Port Agent for IP type 1%

Unrecognized host or address, or protocol not running.
```

Complete these steps to troubleshoot this problem:

1. Ensure the router can reach the DNS server. Ping the DNS server from the router with its IP address, and make sure that the **ip name-server** command is used to configure the IP address of the DNS server on the router.

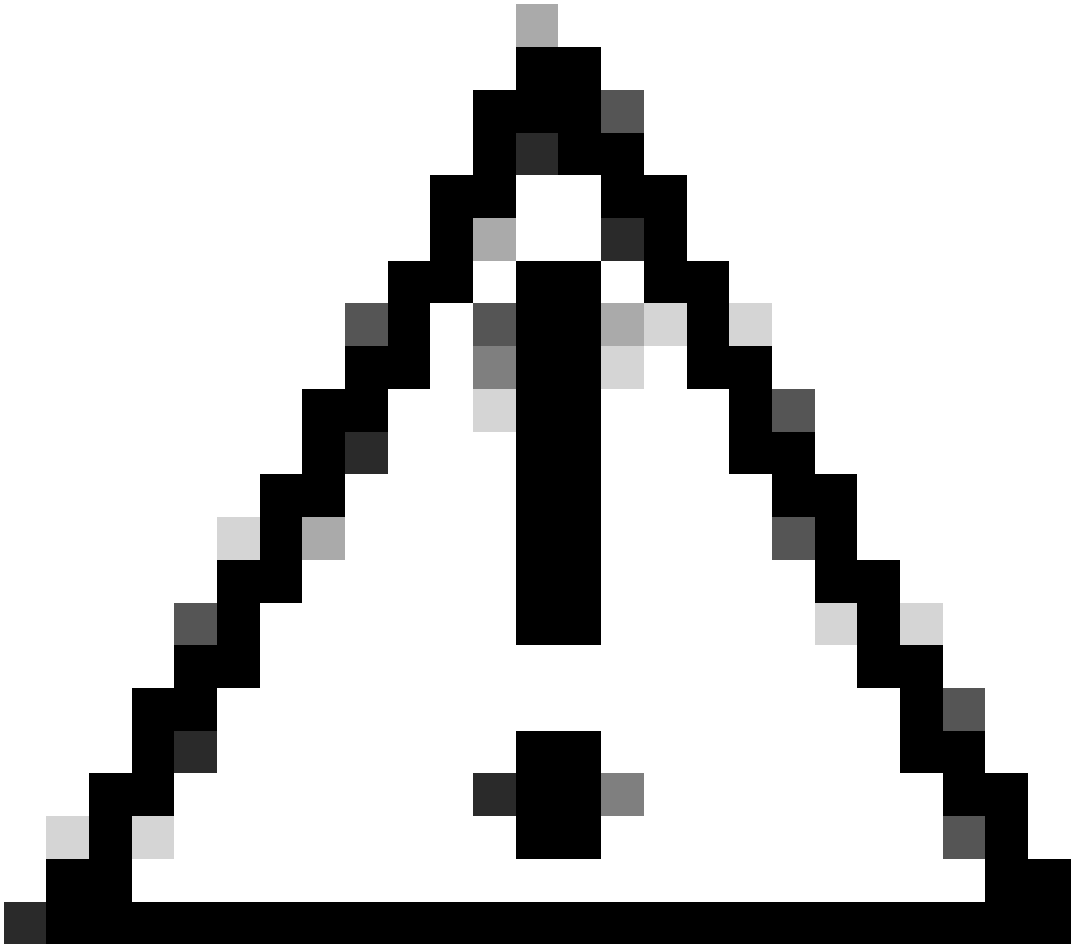
2. Use these steps to ensure that the router forwards the lookup requests:

   a. Define an access control list (ACL) that matches on DNS packets:

   ```
   <#root>

   access-list 101 permit udp any any eq domain
   access-list 101 permit udp any eq domain any
   ```

   b. Use the**debug ip packet 101**command.

   Caution: Ensure that you specify the ACL. If you enable the **debug ip packet** command without an ACL, it can produce a large amount of output to the console and affect access to the device.

   ---

3. Ensure you have the**ip domain-lookup**command enabled on the router.

You Can Ping a Web Server, But You Cannot View the HTML Pages

In rare cases, you are unable to access particular Web sites by name. This problem typically results from the inaccessible sites that perform a reverse DNS lookup on the source IP address to verify that the address is not spoofed. If an incorrect entry or no entry returns (in other words, there is no associated name for the the IP range) then the HTTP request can be blocked.

When you obtain your Internet domain name, you must also apply for an inaddr.arpa domain. This special domain is sometimes called a reverse domain. The reverse domain maps numeric IP addresses into domain names. If your ISP provides your name server or your ISP assigned you an address from a block of its own addresses, you cannot apply for an in-addr.arpa domain on your own. Check with your ISP.

Here is an example that uses www.cisco.com. This next output was captured from a UNIX workstation. The nslookupprogram and the dig program are used. Note the differences in the output:

```
<#root>

sj-cse-280%

nslookup www.cisco.com


Note:  nslookup is deprecated and can be removed from future releases.
Consider with the 'dig' or 'host' programs instead.  Run nslookup with
the '-sil[ent]' option to prevent this message from appearing.
Server:         172.16.226.120
Address:        172.16.226.120#53
Name:   www.cisco.com
Address: 192.168.219.25

sj-cse-280%

nslookup 192.168.219.25


Note:  nslookup is deprecated and can be removed from future releases.
Consider with the 'dig' or 'host' programs instead.  Run nslookup with
the '-sil[ent]' option to prevent this message from appearing.
Server:         172.16.226.120
Address:        172.16.226.120#53
10.219.133.198.in-addr.arpa     name = www.cisco.com.
```

The dig program prints more detailed information from the DNS packets:

```
<#root>

sj-cse-280%

dig 192.168.219.25



; <<>> DiG 9.0.1 <<>> 192.168.219.25
;; global options:  printcmd
;; Got answer:
```

```
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 5231
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;192.168.219.25.                        IN      A

;; AUTHORITY SECTION:
.                       86400   IN      SOA
A.ROOT-SERVERS.NET. nstld.verisign-grs.com.
( 2002031800 1800 900 604800 86400 )

;; Query time: 135 msec
;; SERVER: 172.16.226.120#53(172.16.226.120)
;; WHEN: Mon Mar 18 09:42:20 2002
;; MSG SIZE  rcvd: 107
```

Router Queries Multiple Name Servers

Dependant upon on the network activity level, the router can query multiple name servers listed in the configuration. This is an example from the output **debug ip domain detail**:

<#root>

Router#

show run | section name-server

```
ip name-server 192.168.1.1 10.0.0.2
Router#
Router#
```

debug ip domain detail

Router#

test002

```
*May 12 17:56:32.723: DNS: detail: cdns_name_verify_internal: Checking if hostname is valid or not
*May 12 17:56:32.723: DNS: info: cdns_name_verify_internal: Hostname is valid
*May 12 17:56:32.723: DNS: detail: cdns_get_rr_type: converting name kind 2000 to type 28
*May 12 17:56:32.723: DNS: detail: read_forwards: Forward zone server list:
*May 12 17:56:32.723: DNS: info: delegpt_log: DelegationPoint<.>: 0 names (0 missing), 2 addrs (0
*May 12 17:56:32.724: DNS: detail: val_operate: validator[module 0] operate: extstate:module_state
*May 12 17:56:32.724: DNS: info: log_nametypeclass: validator operate: query test002. AAAA IN
*May 12 17:56:32.724: DNS: detail: iter_operate: iterator[module 1] operate: extstate:module_state
*May 12 17:56:32.724: DNS: info: log_nametypeclass: resolving test002. AAAA IN
*May 12 17:56:32.724: DNS: detail: error_response: return error response NXDOMAIN
*May 12 17:56:32.724: DNS: detail: val_operate: validator[module 0] operate: extstate:module_wait_
*May 12 17:56:32.724: DNS: info: log_nametypeclass: validator operate: query test002. AAAA IN
*May 12 17:56:32.725: DNS: detail: cdns_get_rr_type: converting name kind 2000 to type 28
*May 12 17:56:32.725: DNS: detail: read_forwards: Forward zone server list:
*May 12 17:56:32.725: DNS: info: delegpt_log: DelegationPoint<.>: 0 names (0 missing), 2 addrs (0
*May 12 17:56:32.726: DNS: detail: val_operate: validator[module 0] operate: extstate:module_state
*May 12 17:56:32.726: DNS: info: log_nametypeclass: validator operate: query test002. AAAA IN
*May 12 17:56:32.726: DNS: detail: iter_operate: iterator[module 1] operate: extstate:module_state
```

```
*May 12 17:56:32.726: DNS: info: log_nametypeclass: resolving test002. AAAA IN
*May 12 17:56:32.726: DNS: info: log_nametypeclass: processQueryTargets: test002. AAAA IN
*May 12 17:56:32.726: DNS: info: log_nametypeclass: sending query: test002. AAAA IN
*May 12 17:56:32.726: DNS: detail: log_name_addr: sending to target: <.> 192.168.1.1#53
*May 12 17:56:32.726: DNS: detail: cdns_get_first_hop: dst 192.168.1.1, intf GigabitEthernet1

*May 12 17:56:32.726: DNS: detail: cdns_set_udp_source_interface: using source interface GigabitEt
*May 12 17:56:33.726: DNS: detail: cdns_get_first_hop: dst 192.168.1.1, intf GigabitEthernet1
*May 12 17:56:33.726: DNS: detail: cdns_set_udp_source_interface: using source interface GigabitEt
*May 12 17:56:34.726: DNS: detail: iter_operate: iterator[module 1] operate: extstate:module_wait_
*May 12 17:56:34.726: DNS: info: log_nametypeclass: iterator operate: query test002. AAAA IN
*May 12 17:56:34.726: DNS: info: log_nametypeclass: processQueryTargets: test002. AAAA IN
*May 12 17:56:34.727: DNS: info: log_nametypeclass: sending query: test002. AAAA IN
*May 12 17:56:34.727: DNS: detail: log_name_addr: sending to target: <.> 192.168.1.1#53
*May 12 17:56:34.727: DNS: detail: cdns_get_first_hop: dst 192.168.1.1, intf GigabitEthernet1
*May 12 17:56:34.727: DNS: detail: cdns_set_udp_source_interface: using source interface GigabitEt
*May 12 17:56:35.729: DNS: detail: iter_operate: iterator[module 1] operate: extstate:module_wait_
*May 12 17:56:35.729: DNS: info: log_nametypeclass: iterator operate: query test002. AAAA IN
*May 12 17:56:35.729: DNS: info: log_nametypeclass: response for test002. AAAA IN

*May 12 17:56:35.729: DNS: info: log_name_addr: reply from <.> 192.168.1.1#53
*May 12 17:56:35.729: DNS: info: processQueryResponse: query response was THROWAWAY

*May 12 17:56:35.729: DNS: info: log_nametypeclass: processQueryTargets: test002. AAAA IN

*May 12 17:56:35.729: DNS: info: log_nametypeclass: sending query: test002. AAAA IN
*May 12 17:56:35.729: DNS: detail: log_name_addr: sending to target: <.> 10.0.0.2#53
*May 12 17:56:35.730: DNS: detail: cdns_get_first_hop: dst 10.0.0.2, intf GigabitEthernet1

*May 12 17:56:35.730: DNS: detail: cdns_set_udp_source_interface: using source interface GigabitEt
*May 12 17:58:35.732: DNS: error: comm_point_tcp_handle_write: tcp connect: Connection refused
*May 12 17:58:35.732: DNS: detail: log_addr: remote address is ip4 10.0.0.2 port 53 (len 16)
*May 12 17:58:35.732: DNS: detail: outnet_tcp_cb: outnettcp got tcp error -1
*May 12 17:58:35.732: DNS: detail: log_addr: tcp error for address ip4 10.0.0.2 port 53 (len 16)
*May 12 17:58:35.732: DNS: detail: iter_operate: iterator[module 1] operate: extstate:module_wait_
*May 12 17:58:35.732: DNS: info: log_nametypeclass: iterator operate: query test002. AAAA IN
*May 12 17:58:35.732: DNS: info: log_nametypeclass: processQueryTargets: test002. AAAA IN
```

This behavior is expected and occurs when the router needs to create an Address Resolution Protocol (ARP) entry for the DNS server. By default, a router maintains an ARP entry for four hours. In periods of low activity, the router needs to complete the ARP entry and then perform the DNS query. If the ARP entry for the DNS server is not in the router ARP table, then you would get a failure if it sends only one DNS query. So, two queries are sent out, one to get the ARP entry, if needed, and the second to actually do the DNS query. This behavior is common with TCP/IP applications.

Related Information

- IP Addressing Support
- IP Routing Support
- Cisco Technical Support & Downloads