ルータでの DNS の設定

内容

はじめに

このドキュメントでは、シスコルータのドメインネームシステム（DNS）を設定する方法について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Cisco IOS®コマンドラインインターフェイス(CLI)
- 一般的な DNS 動作

使用するコンポーネント

このドキュメントの内容は、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

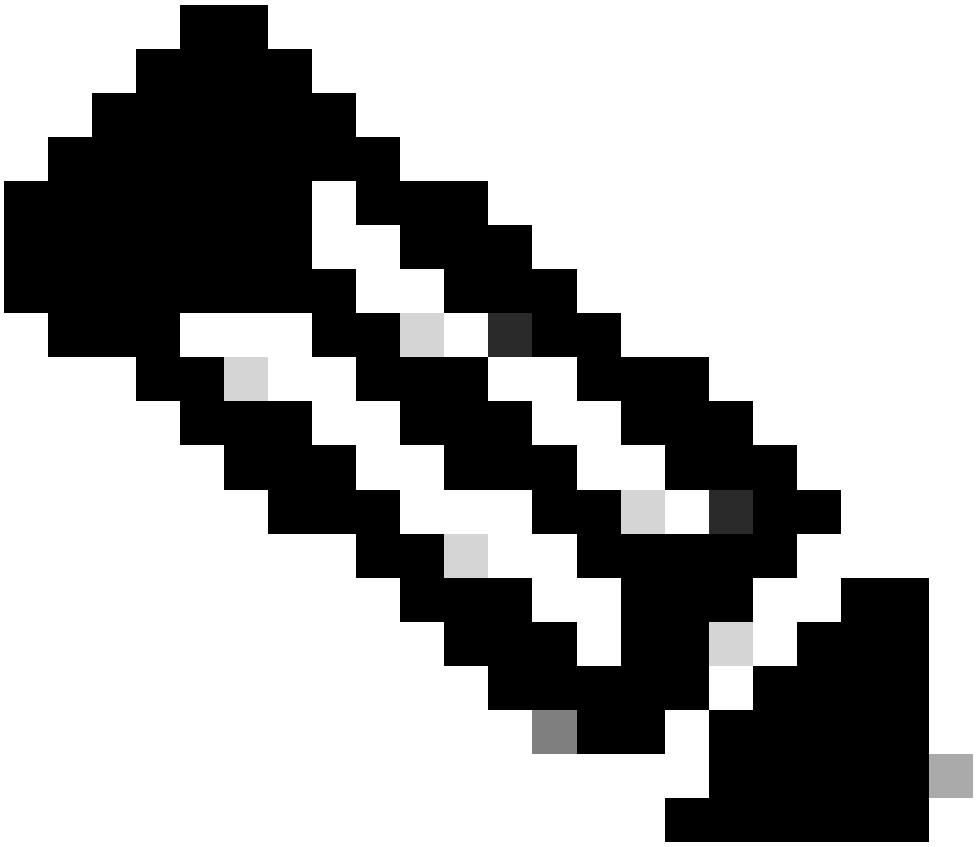
表記法

表記法の詳細については、『シスコ テクニカル ティップスの表記法』を参照してください。

DNSルックアップを使用するためのルータの設定

IPアドレスではなくホスト名でping または**traceroute** コマンドを使用する場合は、DNSルックアップを使用するようにルータを設定できます。これを実行するには、次のコマンドを使用します。

コマンド	説明
ip domain lookup	DNS ベースでのホスト名からアドレスへの変換が有効になります。このコマンドはデフォルトで有効になっています。
ip name-server	1 つ以上のネーム サーバのアドレスが指定されます。
ip domain list	順番に試行される、ドメインのリストが定義されます。

	 **注：**ドメインリストが存在しない場合は、ip domain-nameグローバルコンフィギュレーションコマンドで指定したドメイン名が使用されます。 ドメイン リストが存在する場合、このデフォルトのドメイン名は使用されません。
ip domain name	非修飾ホスト名（ドット付き 10 進のドメイン名を持たない名前）を完全修飾ドメイン名に変更するために、Cisco IOS ソフトウェアによって使用されるデフォルトのドメイン名が定義されます。非修飾名をドメイン名から区切る最初のピリオドを含めないでください。
ip ospf name-lookup	show EXEC コマンドによって表示される、すべての OSPF 内で使用される DNS 名を検索するために、Open Shortest Path First（OSPF）を設定します。この機能を使用すると、ルータの識別が容易になります。その理由は、ルータがそのルータ ID または隣接 ID ではなく、名前によって表示されるからです。

この例では、基本的な DNS lookup 用に設定されたルータの設定例を示しています。

基本的な DNS lookup 設定例

```
     <#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#
```

トラブルシュート

まれに、次のいずれかの工ラー状態が表示されることがあります。

<#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.

この問題をトラブルシューティングするには、次のステップを実行します。

1. ルータが DNS サーバに到達可能であることを確認します。ルータからそのIPアドレスでDNSサーバにpingを実行し、ルータ上でDNSサーバのIPアドレスを設定するために**ip name-server**コマンドが使用されていることを確認します。

2.

次の手順で、ルックアップ要求がルータによって転送されることを確認します。

a.

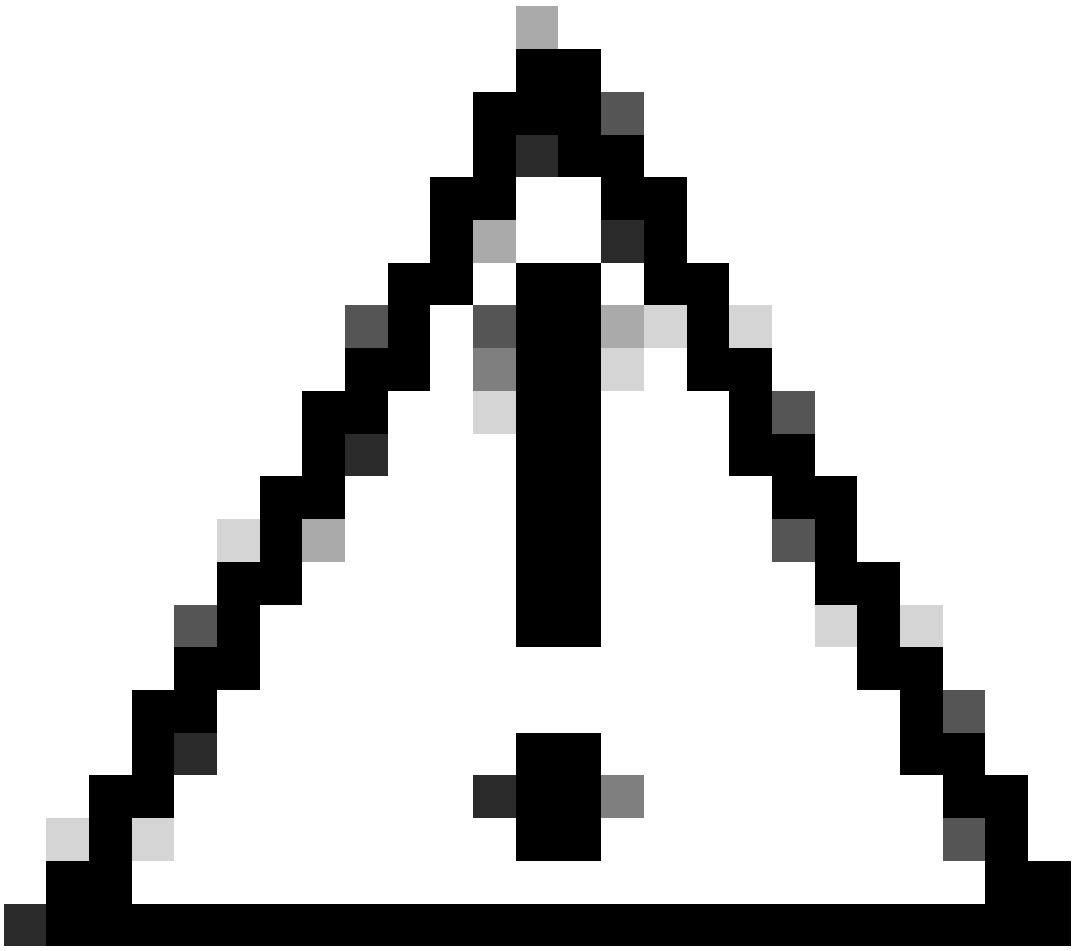
DNS パケットに照合する ACL を定義します。

<#root>

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

b.

debug ip packet 101コマンドを使用します。



注意:ACLを指定していることを確認してください。ACLを使用せずに**debug ip packet**コマンドを有効にすると、コンソールに大量の出力が生成され、デバイスへのアクセスに影響を及ぼす場合があります。

3.

ルータで**ip domain**-lookupcommandコマンドが有効になっていることを確認します。

Webサーバにping できるが、HTML ページを表示できない場合

まれに、特定のWebサイトに名前でアクセスできない場合があります。この問題は通常、アドレスがスプーフィングされていないことを確認するために送信元IPアドレスで逆DNSルックアップを実行するアクセス不能サイトに起因します。誤ったエントリが返された場合、またはエントリが返されない場合（つまり、IP範囲に関連する名前がない場合）は、HTTP要求をブロックできます。

インターネットドメイン名を取得する場合は、inaddr.arpaドメインにも申請する必要があります。この特殊なドメインは、逆ドメインと呼ばれることもあります。この逆ドメインによって、数値の IP アドレスがドメイン名にマッピングされます。ISPがネームサーバを提供している場合、またはISPが自身のアドレスのブロックからアドレスを割り当てている場合は、自分でin-addr.arpaドメインを申請する必要はありません。ISP にお問い合せください。

次に、www.cisco.comを使用した例を示します。次の出力は、UNIXワークステーションでキャプチャしたものです。nslookup プログラムとdigプログラムを使用します。出力内の違いに注意してください。

```
<#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.
```

dig プログラムでは、DNS パケットからより詳細な情報が出力されます。

```
    <#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
```

ルータによって複数のネーム サーバにクエリーが実行される

ネットワークのアクティビティレベルに応じて、ルータは設定にリストされている複数のネームサーバにクエリーを送信できます。次に、**debug ip domain detail**の出力例を示します。

```
    <#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 result
*May 12 17:56:32.724: DNS: detail: val_operate: validator[module 0] operate: extstate:module_state_init
*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_init
*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_modul
*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 result
*May 12 17:56:32.726: DNS: detail: val_operate: validator[module 0] operate: extstate:module_state_init
*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_init

***May 12 17:56:32.726: DNS: info: log_nametypeclass: resolving test002. AAAA IN *May 12 17:56:32.726: DNS**

```
*May 12 17:56:32.726: DNS: detail: cdns_set_udp_source_interface: using source interface GigabitEtherne
*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 GigabitEtherne
*May 12 17:56:34.726: DNS: detail: iter_operate: iterator[module 1] operate: extstate:module_wait_reply
*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 GigabitEtherne
*May 12 17:56:35.729: DNS: detail: iter_operate: iterator[module 1] operate: extstate:module_wait_reply
*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:

*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

*May 12 17:56:35.730: DNS: detail: cdns_set_udp_source_interface: using source interface GigabitEtherne
*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_reply
*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
```

この動作は予測されたものであり、ルータで DNS サーバ用に Address Resolution Protocol（ARP; アドレス解決プロトコル）エントリを作成する必要がある場合に発生します。デフォルトでは、ARP エントリはルータで 4 時間維持されます。低いアクティビティの期間に、ルータは ARP エントリの作成を完了し、次に DNS クエリーを実行する必要があります。DNSサーバのARPエントリがルータのARPテーブルにない場合、DNSクエリーを1つだけ送信すると障害が発生します。したがって、2 つのクエリーが送信されることになります。1 つは必要に応じて ARP エントリを取得するためのもので、もう 1 つが実際に DNS クエリーを実行するものです。TCP/IP アプリケーションでは、この動作は一般的です。

関連情報

- **IPアドレッシングのサポート**
- **IPルーティングのサポート**
- シスコのテクニカルサポートとダウンロード

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。