Networking with Windows Server 2016

Exam Ref 70-741

Andrew Warren

Microsoft

Exam Ref 70-741 Networking with Windows Server 2016

Andrew Warren

Exam Ref 70-741 Networking with Windows Server 2016

Published with the authorization of Microsoft Corporation by:
Pearson Education, Inc.

ISBN-13: 978-0-7356-9742-3
ISBN-10: 0-7356-9742-6

Library of Congress Control Number: 2016959968

First Printing December 2016

Editor-in-Chief	Greg Wiegand
Acquisitions Editor	Trina MacDonald
Development Editor	Rick Kughen
Managing Editor	Sandra Schroeder
Senior Project Editor	Tracey Croom
Editorial Production	Backstop Media, Troy Mott
Copy Editor	Kristin Dudley
Indexer	Julie Grady
Proofreader	Christina Rudloff
Technical Editor	Byron Wright
Cover Designer	Twist Creative, Seattle

Contents at a glance

This page intentionally left blank

Contents

This page intentionally left blank

Introduction

The 70-741 exam focuses on the networking features and functionality available in Windows Server 2016. It covers DNS, DHCP, and IPAM implementations as well as remote access solutions such as VPN and Direct Access. It also covers DFS and branch cache solutions, high performance network features and functionality, and implementation of Software Defined Networking (SDN) solutions such as Hyper-V Network Virtualization (HNV) and Network Controller.

The 70-741 exam is geared toward network administrators that are looking to reinforce their existing skills and learn about new networking technology changes and functionality in Windows Server 2016.

This book covers every major topic area found on the exam, but it does not cover every exam question. Only the Microsoft exam team has access to the exam questions, and Microsoft regularly adds new questions to the exam, making it impossible to cover specific questions. You should consider this book a supplement to your relevant real-world experience and other study materials. If you encounter a topic in this book that you do not feel completely comfortable with, use the "Need more review?" links you'll find in the text to find more information and take the time to research and study the topic. Great information is available on MSDN, TechNet, and in blogs and forums.

Organization of this book

This book is organized by the "Skills measured" list published for the exam. The "Skills measured" list is available for each exam on the Microsoft Learning website: *https://aka.ms/examlist*. Each chapter in this book corresponds to a major topic area in the list, and the technical tasks in each topic area determine a chapter's organization. If an exam covers six major topic areas, for example, the book will contain six chapters.

Microsoft certifications

Microsoft certifications distinguish you by proving your command of a broad set of skills and experience with current Microsoft products and technologies. The exams and corresponding certifications are developed to validate your mastery of critical competencies as you design and develop, or implement and support, solutions with Microsoft products and technologies both on-premises and in the cloud. Certification brings a variety of benefits to the individual and to employers and organizations.

MORE INFO **ALL MICROSOFT CERTIFICATIONS**

For information about Microsoft certifications, including a full list of available certifications, go to *https://www.microsoft.com/learning*.

Acknowledgments

Andrew Warren Writing a book is a collaborative effort, and so I would like to thank my editor, Trina MacDonald, for her guidance. I'd also like to thank my wife, Naomi, and daughter, Amelia, for their patience while I spent the summer locked away in my office following that guidance.

Free ebooks from Microsoft Press

From technical overviews to in-depth information on special topics, the free ebooks from Microsoft Press cover a wide range of topics. These ebooks are available in PDF, EPUB, and Mobi for Kindle formats, ready for you to download at:

https://aka.ms/mspressfree

Check back often to see what is new!

Microsoft Virtual Academy

Build your knowledge of Microsoft technologies with free expert-led online training from Microsoft Virtual Academy (MVA). MVA offers a comprehensive library of videos, live events, and more to help you learn the latest technologies and prepare for certification exams. You'll find what you need here:

https://www.microsoftvirtualacademy.com

Quick access to online references

Throughout this book are addresses to webpages that the author has recommended you visit for more information. Some of these addresses (also known as URLs) can be painstaking to type into a web browser, so we've compiled all of them into a single list that readers of the print edition can refer to while they read.

Download the list at *https://aka.ms/examref741/downloads*.

The URLs are organized by chapter and heading. Every time you come across a URL in the book, find the hyperlink in the list to go directly to the webpage.

Errata, updates, & book support

We've made every effort to ensure the accuracy of this book and its companion content. You can access updates to this book—in the form of a list of submitted errata and their related corrections—at:

https://aka.ms/examref741/errata

If you discover an error that is not already listed, please submit it to us at the same page.

If you need additional support, email Microsoft Press Book Support at *mspinput@microsoft.com*.

Please note that product support for Microsoft software and hardware is not offered through the previous addresses. For help with Microsoft software or hardware, go to *https://support.microsoft.com*.

We want to hear from you

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

https://aka.ms/tellpress

We know you're busy, so we've kept it short with just a few questions. Your answers go directly to the editors at Microsoft Press. (No personal information will be requested.) Thanks in advance for your input!

Stay in touch

Let's keep the conversation going! We're on Twitter: *http://twitter.com/MicrosoftPress*.

This page intentionally left blank

Important: How to use this book to study for the exam

Certification exams validate your on-the-job experience and product knowledge. To gauge your readiness to take an exam, use this Exam Ref to help you check your understanding of the skills tested by the exam. Determine the topics you know well and the areas in which you need more experience. To help you refresh your skills in specific areas, we have also provided "Need more review?" pointers, which direct you to more in-depth information outside the book.

The Exam Ref is not a substitute for hands-on experience. This book is not designed to teach you new skills.

We recommend that you round out your exam preparation by using a combination of available study materials and courses. Learn more about available classroom training at *https://www.microsoft.com/learning*. Microsoft Official Practice Tests are available for many exams at *https://aka.ms/practicetests*. You can also find free online courses and live events from Microsoft Virtual Academy at *https://www.microsoftvirtualacademy.com*.

This book is organized by the "Skills measured" list published for the exam. The "Skills measured" list for each exam is available on the Microsoft Learning website: *https://aka.ms/examlist*.

Note that this Exam Ref is based on this publicly available information and the author's experience. To safeguard the integrity of the exam, authors do not have access to the exam questions.

This page intentionally left blank

CHAPTER 1

Implement Domain Name System

Typically, users and computers use host names rather than Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) network addresses to communicate with other hosts and services on networks. A Windows Server 2016 service, known as the Domain Name System (DNS) server role, resolves these names into IPv4 or IPv6 addresses.

Since many important apps and services rely on the DNS server role, it is important that you know how to install and configure Windows Server 2016 name resolution using the DNS server role. As a result, the 70-741 Networking Windows Server 2016 exam covers how to install and configure the DNS server role on Windows Server 2016.

The 70-741 Networking Windows Server 2016 exam also covers how to implement zones and Domain Name System records using the DNS server role. It is therefore important that you know how to create and manage DNS zones using the Windows Server 2016 DNS server role, and how to create and manage host and service-related records within these zones.

IMPORTANT

Have you read page xv?

It contains valuable information regarding the skills you need to pass the exam.

Skills in this chapter:

- Install and configure DNS servers
- Create and configure DNS zones and records

Skill 1.1: Install and configure DNS servers

Windows Server 2016 provides the DNS server role to enable you to provide name resolution services to devices and computers in your organization's network infrastructure. The first stage to provide name resolution is to deploy the DNS server role on Windows Server 2016 server computers.

Overview of name resolution

Although IP addressing is not especially complex, it is easier for users to work with host names rather than with the IPv4 or IPv6 addresses of hosts, such as websites, to which they want to connect. When an application, such as Microsoft Edge, references a website name, the name in the URL is converted into the underlying IPv4 or IPv6 address using a process known as name resolution. Windows 10 and Windows Server 2016 computers can use two types of names. These are:

- **Host names** A host name, up to 255 characters in length, contains only alpha-numeric characters, periods, and hyphens. A host name is an alias combined with a DNS domain name. For example, the alias *computer1,* is prefixed to the domain name, Contoso.com, to create the host name, or Fully Qualified Domain Name (FQDN), *computer1*.contoso.com.
- **NetBIOS names** Less relevant today, NetBIOS names use a nonhierarchical structure based on a 16-character name. The sixteenth character identifies a particular service running on the computer named by the preceding 15 characters. Thus, LON-SVR1[20h] is the NetBIOS server service on the computer named LON-SVR1.

The method in which a Windows 10 or Windows Server 2016 computer resolves names varies based on its configuration, but it typically works as shown in Figure 1-1.

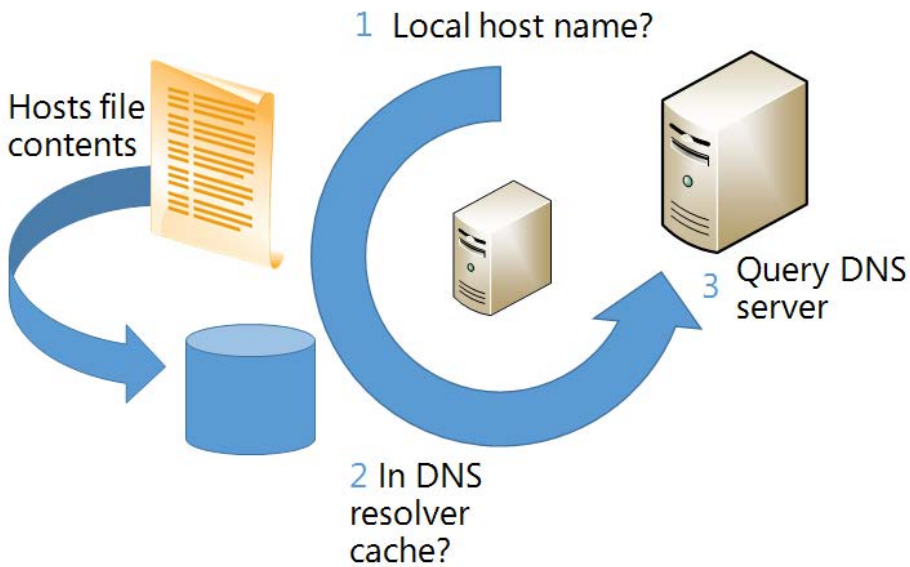

FIGURE 1-1 Typical stages of name resolution in a Windows Server computer

The following process identifies the typical stages of name resolution for a Windows 10 or Windows Server 2016 computer.

1. Determine whether the queried host name is the same as the local host name.

2. Search the local DNS resolver cache for the queried host name. The cache is updated when records are successfully resolved. In addition, the content of the local Hosts file is added to the resolver cache.
3. Petition a DNS server for the required host name.

> ***NEED MORE REVIEW?*** **IPV4 NAME RESOLUTION**
>
> **To review further details about IPv4 name resolution, refer to the Microsoft TechNet website at *https://technet.microsoft.com/library/dd379505(v=ws.10).aspx*.**

Of course, name resolution in Windows Server 2016 does more than just provide for simple name to IP mapping. The DNS server role is also used by computers to locate services within the network infrastructure. For example, when a computer starts up, the user must sign-in to the Active Directory Domain Services (AD DS) domain and perhaps open Microsoft Office Outlook. This means that the client computer must locate a server that can provide authentication services in the local AD DS site, and furthermore, locate the appropriate Microsoft Exchange mailbox server for the user. These processes require DNS.

Determine DNS installation requirements

Before you can install the DNS server role, you must verify that your server computer meets the installation requirements of the role.

The DNS server role installation requirements are:

- **Security** You must sign in on the server computer as a member of the local Administrators group.
- **IP configuration** The server must have a statically assigned IPv4 and/or IPv6 configuration. This ensures that client computers can locate the DNS server role by using its IP address.

In addition to these server requirements, you must also be prepared to answer questions that relate to your organization's network infrastructure. These organizational questions pertain to your Internet presence, and the registered domain names that you intend to use publicly. Although you need not define these domain names during DNS role installation, you must provide this information when you configure the DNS role.

Install the DNS server role

You can install the DNS server role by using Server Manager, or by using Windows PowerShell.

Installing DNS with Server Manager

To install the DNS server role with Server Manager, use the following procedure:

1. Sign in to the target server as a local administrator.
2. Open Server Manager.

3. In Server Manager, click Manage and then click Add Roles And Features.
4. In the Add Roles And Features Wizard's Before You Begin page, click Next.
5. On the Select Installation Type page, click Role-Based or Feature-Based Installation, and click Next.
6. On the Select Destination Server page, select the server from the Server Pool list, and click Next.
7. In the Roles list on the Select Server Roles page, select the DNS Server (see Figure 1-2).
8. In the Add Roles And Features Wizard pop-up dialog box, click Add Features, and then click Next.
9. On the Select features page, click Next.
10. On the DNS Server page, click Next.
11. On the Confirm Installation Selections page, click Install. When the installation is complete, click Close.

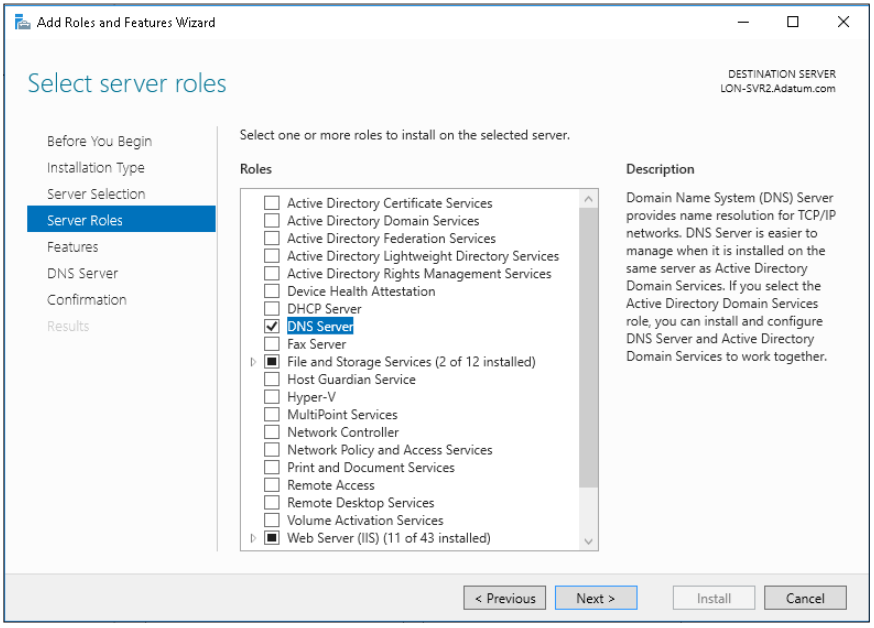

FIGURE 1-2 Installing the DNS Server role by using Server Manager

Installing DNS with Windows PowerShell

Although using Server Manager to install server roles and features is simple, it is not always the quickest method. To install the DNS server role and all related management tools by using Windows PowerShell, use the following procedure:

1. Sign in to the target server as a local administrator.

2. Open an elevated Windows PowerShell window.
3. At the Windows PowerShell prompt, as shown in Figure 1-3, type the following command and press Enter:

```
Add-WindowsFeature DNS -IncludeManagementTools
```

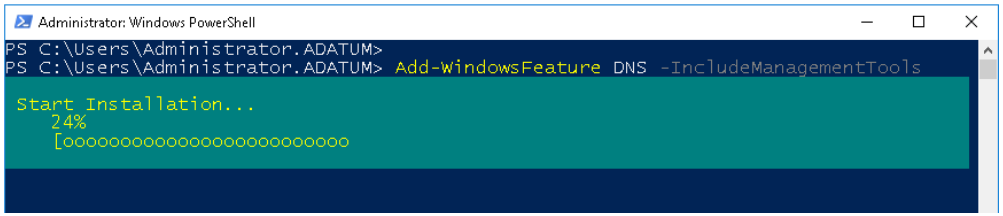

FIGURE 1-3 Installing the DNS Server with Windows PowerShell

Determine supported DNS deployment scenarios on Nano Server

Nano Server is a new Windows Server 2016 deployment option. It is similar to Windows Server Core, but has much smaller hardware requirements. Nano Server also has very limited local sign-in capabilities and local administration function, and supports only 64-bit apps, agents, and tools.

There are a number of situations when you should consider choosing Nano Server over other Windows Server deployment options. For example, Nano Server provides a good platform for a web server running Internet Information Services (IIS). Also, Nano Server is ideally suited to run the DNS server role.

> ***NEED MORE REVIEW?*** **GETTING STARTED WITH NANO SERVER**
>
> To review further details about working with Nano Server, refer to the Microsoft TechNet website at *https://technet.microsoft.com/windows-server-docs/compute/nano-server/getting-started-with-nano-server.*

To install the DNS server role on Nano Server, you can use one of the following two strategies.

- **Install the DNS server role as part of the Nano Server deployment** When you deploy Nano Server with the New-NanoServerImage cmdlet, you can use the -Packages Microsoft-NanoServer-DNS-Package parameter to install the DNS server role.
- **Add the role after deployment** After you have deployed Nano Server, you can add the DNS server role by using either Server Manager or Windows PowerShell. However, since Nano Server is a headless server platform with very little local management capability, you must remotely manage the server.

You can add the role to Nano server using one of the following methods:

- From Server Manager, use the Add Other Servers To Manage option to add the Nano Server as a manageable server. Then add the DNS Server role to the server using the procedure outlined earlier in this chapter (see "Installing DNS with Server Manager").
- Establish a Windows PowerShell remoting session with the Nano Server by using the Enter-PSSession cmdlet. You can then use Windows PowerShell cmdlets to install the DNS server role, as described earlier in this chapter. For example, to add the DNS role to a Nano Server from a Windows PowerShell remote session, use the following command:

```
Enable-WindowsOptionalFeature -Online -FeatureName DNS-Server-Full-Role
```

EXAM TIP

Active Directory integrated DNS is not supported on Nano Server, which means that you can implement file-based DNS only on Nano Server.

NEED MORE REVIEW? **ENABLE AND USE REMOTE COMMANDS IN WINDOWS POWERSHELL**

To review further details about using Windows PowerShell remoting, refer to the Microsoft TechNet website at *https://technet.microsoft.com/magazine/ff700227.aspx*.

Configure forwarders, root hints, recursion, and delegation

After you have installed the DNS server role on your Windows Server 2016 server computer, you must configure it. This involves configuring forwarding, root hints, recursion, and delegation.

Configure forwarders

DNS forwarding enables you to define what happens to a DNS query when the petitioned DNS server is unable to resolve that DNS query. For example, you can configure and use DNS forwarding to control the flow of DNS requests throughout your organization so that only specific DNS servers are used to handle Internet DNS queries.

With DNS forwarding, you can:

- Configure a DNS server only to respond to those queries that it can satisfy by reference to locally stored zone information. For all other requests, the petitioned DNS server must forward the request to another DNS server.
- Define the forwarding behavior for specific DNS domains by configuring DNS conditional forwarding. In this scenario, if the DNS query contains a specific domain name, for example Contoso.com, then it is forwarded to a specific DNS server.

To configure forwarding, use the following procedure:

1. In Server Manager, click Tools, and then click DNS.
2. In DNS Manager, right-click the DNS server in the navigation pane and click Properties.
3. In the Server Properties dialog box, on the Forwarders tab, click Edit.
4. In the IP Address list located in the Edit Forwarders dialog box, enter the IP address of the server to which you want to forward all DNS queries, and then click OK. You can configure several DNS servers here; those servers are petitioned in preference order. You can also set a timeout value, in seconds, after which the query is timed out
5. In the Server Properties dialog box on the Forwarders tab you can view and edit the list of DNS forwarders, as shown in Figure 1-4. You can also determine what happens when no DNS forwarders can be contacted. By default, when forwarders cannot be contacted, root hints are used. Root hints are discussed in the next section. Click OK to complete configuration.

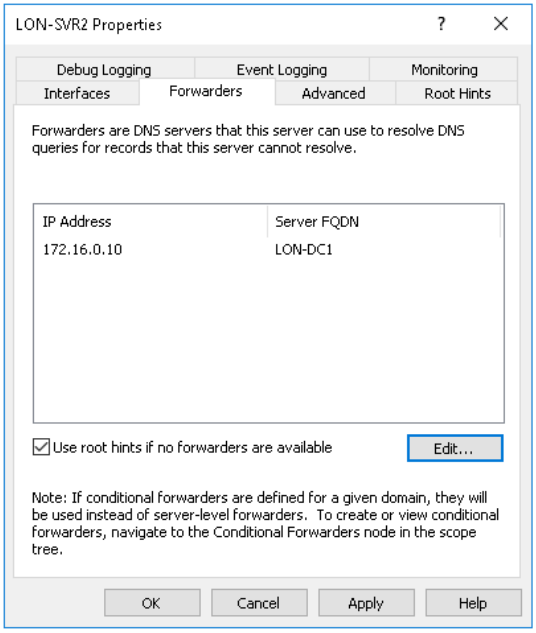

FIGURE 1-4 Configuring DNS forwarding

EXAM TIP

You can also configure forwarding by using the Add-DnsServerForwarder Windows PowerShell cmdlet.

To enable and configure conditional forwarding, use the following procedure:

1. In DNS Manager, right-click the Conditional Forwarders node in the navigation pane, and then click New Conditional Forwarder.

2. On the New Conditional Forwarder dialog box, in the DNS Domain box, type the domain name for which you want to create a conditional forward, as shown in Figure 1-5. Next, in the IP address of the master servers list, enter the IP address of the server to use as a forwarder for this domain; press Enter.
3. Optionally, specify the Number of Seconds Before Forward Queries Time Out value. The default value is 5 seconds.
4. Click OK.

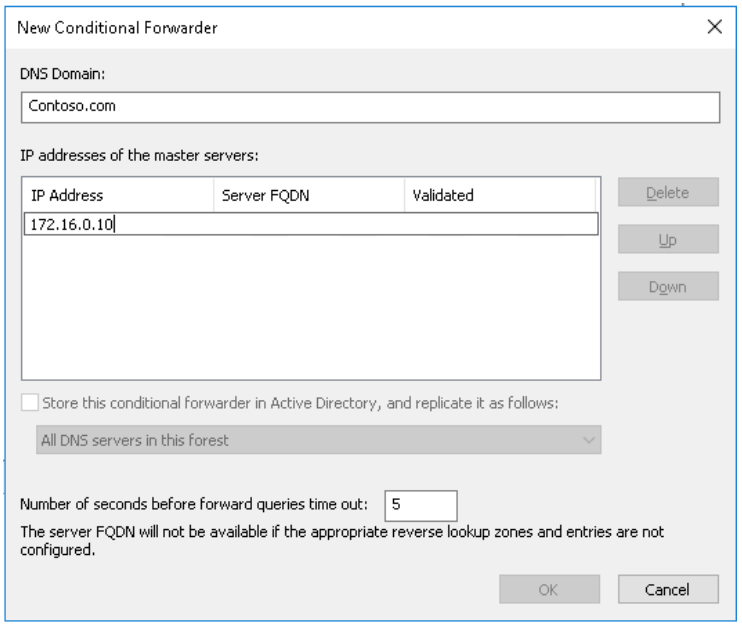

FIGURE 1-5 Configuring conditional DNS forwarding

EXAM TIP

You can use the Add-DnsServerConditionalForwarderZone Windows PowerShell cmdlet to configure conditional forwarding.

Configure root hints

If you do not specify DNS forwarding, then when a petitioned DNS server is unable to satisfy a DNS query, it uses root hints to determine how to resolve it. Before we look at root hints, it is important that you understand how an Internet DNS query is handled.

HOW AN INTERNET DNS QUERY IS HANDLED

A client app, such as Microsoft Edge, wants to resolve a name (like www.contoso.com) to the relevant IPv4 address. This app is referred to as a DNS client. The process used to resolve this name is described next and is shown in Figure 1-6.

1. The DNS client petitions its configured DNS server for the required record (for example, www.contoso.com) using a recursive query.

EXAM TIP

When a DNS server receives a recursive query, it either returns the required result, or it returns an error; the DNS server does not refer the DNS client to another server.

   - The petitioned DNS server checks to see if it is authoritative for the required record. If it is, it returns the requested information.
   - If it is not authoritative, the DNS server checks its local cache to determine if the record was recently resolved. If the record exists in cache, it is returned to the petitioning client.

2. If the record is not cached, then the DNS server uses a series of iterative queries to other DNS servers in which it requests the petitioned record. It starts with the root server.

EXAM TIP

When a DNS server receives an iterative query, it either returns the required result, or it returns a referral to another server that might be authoritative for the requested record.

3. The record returns it if the root server is authoritative for the requested record. Otherwise, the root server returns the IP address of a DNS server authoritative for the next down-level domain, in this instance .com.
4. The original DNS server petitions the specified .com DNS server using another iterative query.
5. The .com DNS server is not authoritative, and so returns the IP address of the Contoso.com DNS server.
6. The original DNS server petitions the specified Contoso.com DNS server using another iterative query.
7. The Contoso.com DNS server is authoritative, and so returns the required information—in this case, the IPv4 address for www.contoso.com.
8. The original DNS server caches the record and returns the requested information to the DNS client.

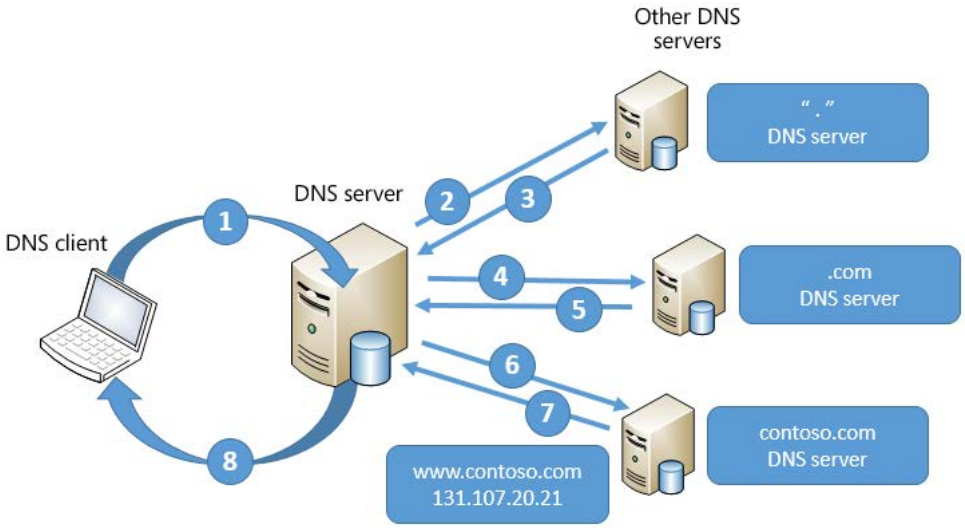

FIGURE 1-6 How Internet DNS queries work

HOW ROOT HINTS ARE USED

As you can see in the preceding explanation and diagram, if a DNS server is not authoritative and holds no cache for that DNS domain, it petitions a root server to start the process of determining which server is authoritative for the petitioned record. However, without the IP address of the root name servers, this process cannot begin.

Root hints are used by DNS servers to enable them to navigate the DNS hierarchy on the Internet, starting at the root. Microsoft DNS servers are preconfigured with the relevant root hint records. However, you can modify the list of root hint servers by using the DNS Manager console or by using Windows PowerShell.

EXAM TIP

By default, the DNS Server service implements root hints by using a file, CACHE.DNS, that is stored in the %systemroot%\System32\dns folder on the server computer.

You might consider editing the root hints information if you want to configure the flow of DNS query traffic within your internal network. This is also useful between your internal network and the boundary network, which sits between your internal network and the Internet.

EDITING ROOT HINTS

To modify the root hints information using DNS Manager, use the following procedure:

1. In Server Manager, click Tools, and then click DNS.
2. In the DNS Manager console, locate the appropriate DNS server. Right-click the server and click Properties.

3. In the server Properties dialog box, click the Root Hints tab, as shown in Figure 1-7.
4. You can then add new records, or edit or remove any existing records. You can also click Copy From Server to import the root hints from another online DNS server. Click OK when you have finished editing root hints.

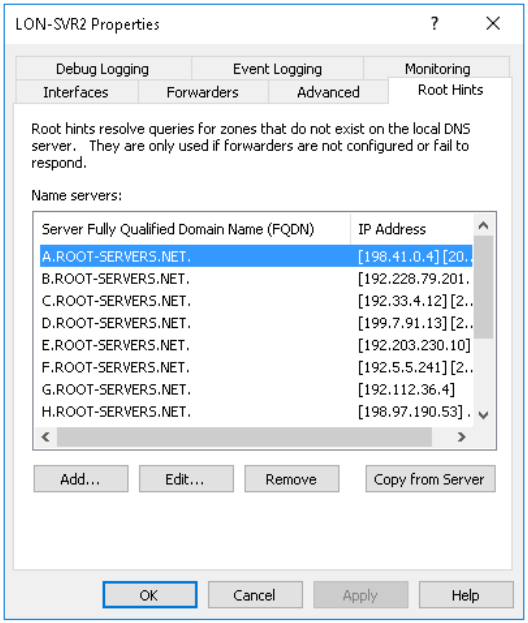

FIGURE 1-7 Configuring root hints

Also, you can use Windows PowerShell to modify the root hints information on your DNS server. The following cmdlets are available to manage root hints:

- **Add-DnsServerRootHint** Enables you to add new root hints records.
- **Remove-DnsServerRootHint** Enables you to delete root hints records.
- **Set-DnsServerRootHint** Enables you to edit existing root hints records. You can also use the Get-DnsServerRootHint cmdlet to retrieve the required record for editing.
- **Import-DnsServerRootHint** Enables you to copy the root hints information from another online DNS server.

For example, to update the value for the root hints assigned to H.Root-servers.adatum.com, use the following two Windows PowerShell commands:

```
$hint = (Get-DnsServerRootHint | Where-Object {$_.NameServer.RecordData.NameServer
-eq "H.Root-Servers.Adatum.com."} )

$hint.IPAddress[0].RecordData.Ipv4address = "10.24.60.254"
```

The first command obtains the H.Root-servers.adatum.com root hint and assigns it to the variable `$hint`. The Get-DnsServerRootHint cmdlet obtains the list of all root hints, and the Where-Object cmdlet filters the results to get only the root hint for H.Root-servers.adatum.com.

Configure recursion

Recursion is the name resolution process when a petitioned DNS server queries other DNS servers to resolve a DNS query on behalf of a requesting client. The petitioned server then returns the answer to the DNS client. By default, all DNS servers perform recursive queries on behalf of their DNS clients and other DNS servers that have forwarded DNS client queries to them.

However, since malicious people can use recursion as a means to attempt a denial of service attack on your DNS servers, you should consider disabling recursion on any DNS server in your network that is not intended to receive recursive queries.

To disable recursion, use the following procedure:

1. From Server Manager, click Tools, and then click DNS.
2. In the DNS Manager console, right-click the appropriate server, and then click Properties.
3. Click the Advanced tab, and then in the Server options list, select the Disable Recursion (Also Disables Forwarders) check box, as shown in Figure 1-8, and then click OK.

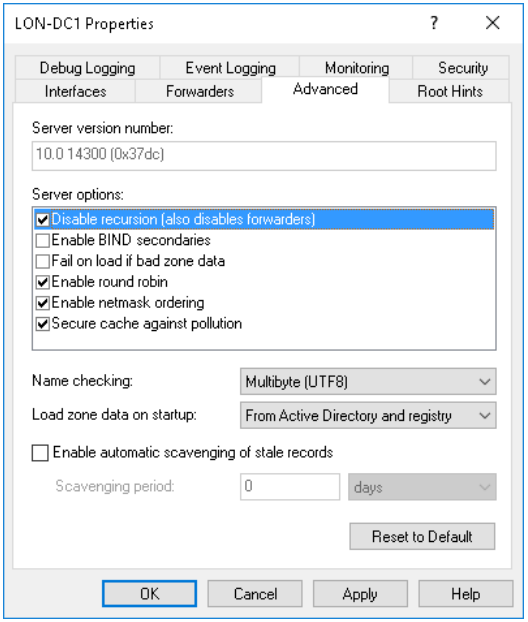

FIGURE 1-8 Disabling recursion

RECURSION SCOPES

While it might seem like a good idea to disable recursion, there are servers that must perform recursion for their clients and other DNS servers. However, these are still at risk from malicious network attacks. Windows Server 2016 supports a feature known as *recursion scopes*, which allow you to control recursive query behavior. To do this, you must use DNS Server Policies.

For example, you might have a DNS server that should be able to perform recursive queries for internal clients within the Adatum.com domain, but should not accept any recursive queries from Internet-based computers. To configure this behavior, open Windows PowerShell and then run the following two commands:

```
Set-DnsServerRecursionScope -Name . -EnableRecursion $False

Add-DnsServerRecursionScope -Name "InternalAdatumClients" -EnableRecursion $True
```

The first command disables recursion for the default recursion scope, which as a result, turns off recursion. The default scope consists of the server-level recursion and forwarding settings that we previously discussed (see "Configure forwarders, root hints, recursion, and delegation," in this chapter).

The second command creates a new recursion scope called InternalAdatumClients. Recursion is enabled for clients in this scope. Next, you must define which clients are part of the recursion scope. Use the following Windows PowerShell command to achieve this:

```
Add-DnsServerQueryResolutionPolicy -Name "RecursionControlPolicy" -Action ALLOW
-ApplyOnRecursion -RecursionScope "InternalAdatumClients" -ServerInterfaceIP
"EQ,10.24.60.254"
```

In this example, client requests received on the DNS server interface with the IP 10.24.60.254 are evaluated as belonging to InternalAdatumClients, and recursion is enabled. For client requests received on other server interfaces, recursion is disabled.

> ***NEED MORE REVIEW?*** **ADD-DNSSERVERQUERYRESOLUTIONPOLICY**
>
> **For more information about using Windows PowerShell to configure recursion scopes, visit the TechNet website at *https://technet.microsoft.com/library/mt126273.aspx*.**

Configure delegation

This content is covered in Chapter 1, Implement Domain Name System: "Configure delegation."

Configure advanced DNS settings

Configuring forwarding, recursion, and root hints enables you to control the fundamentals of how DNS queries are processed within your organization. After you have configured these settings, you can move on to enable and configure more advanced settings.

Configure DNSSEC

DNSSEC is a security setting for DNS that enables all the DNS records in a DNS zone to be digitally signed so DNS clients are able to verify the identity of the DNS server. DNSSEC helps ensure that the DNS client is communicating with a genuine DNS server.

> ***NOTE*** **DNS ZONES**
>
> **Creating and managing DNS zones is covered in "Create DNS Zones."**

When a client queries a DNS server that has been configured with DNSSEC, the server returns any DNS results along with a digital signature. To ensure that the signature is valid, the DNS client obtains the public key of the public/private key pair associated with this signature from a *trust anchor*. In order for this to work, you must configure your DNS clients with a trust anchor for the signed DNS zone.

TRUST ANCHORS

To implement DNSSEC, you must create a TrustAnchors zone. This zone is used to store public keys associated with specific DNS zones. You must create a trust anchor from the secured zone on every DNS server that hosts the zone.

NAME RESOLUTION POLICY TABLE

Additionally, you must create, configure, and distribute a Name Resolution Policy Table (NRPT). A DNSSEC rule in the NRPT is used by clients to determine DNS client behavior and is used by DNSSEC to instruct the client to request validation through the use of a signature.

EXAM TIP

It is usual in Active Directory Domain Services (AD DS) environments to use Group Policy Objects (GPOs) to distribute the NRPT.

IMPLEMENTING DNSSEC

After installing Windows Server 2016 and deploying the DNS server role to the server, use the following procedure to implement DNSSEC:

1. Launch the DNSSEC Configuration Wizard from the DNS Manager console to sign the DNS zone. In DNS Manager, right-click the desired zone, point to DNSSEC, and then click Sign The Zone. When you sign the zone, as shown in Figure 1-9, you can choose between three options.

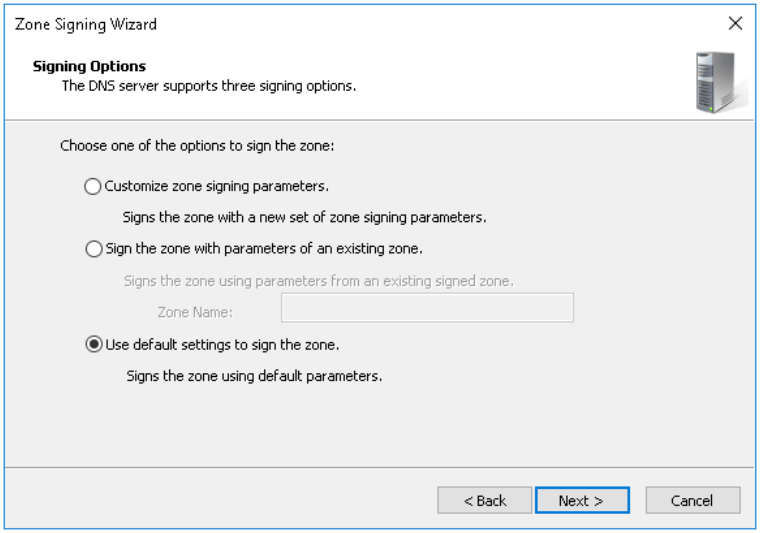

FIGURE 1-9 Signing a DNS zone

- **Customize Zone Signing Parameters** Enables you to configure all values for the Key Signing Key (KSK) and the Zone Signing Key (ZSK).
- **Sign The Zone With Parameters Of An Existing Zone** Enables you to use the same values and options as an existing signed zone.
- **Use Default Settings To Sign The Zone** Signs the zone using default values.

2. Configure Trust Anchor Distribution Points You can choose this option if you select the Customize Zone Signing Parameters option above. Otherwise, after you have signed the zone, use the following procedure to configure trust anchor distribution points:

   A. In DNS Manager, right-click the desired zone, point to DNSSEC, and then click Properties.

   B. In the DNSSEC Properties For Selected Zone dialog box, on the Trust Anchor tab, as shown in Figure 1-10, select the Enable The Distribution Of Trust Anchors For This Zone check box, and click OK. When prompted, click Yes, and then click OK.

   C. Verify that the Trust Points node exists and contains the relevant DNS KEY (DNSKEY) records. To do this, in DNS Manager, expand the Server node and then expand Trust Points. It contains sub nodes for your DNS zones, which contain two DNS KEY (DNSKEY) records.

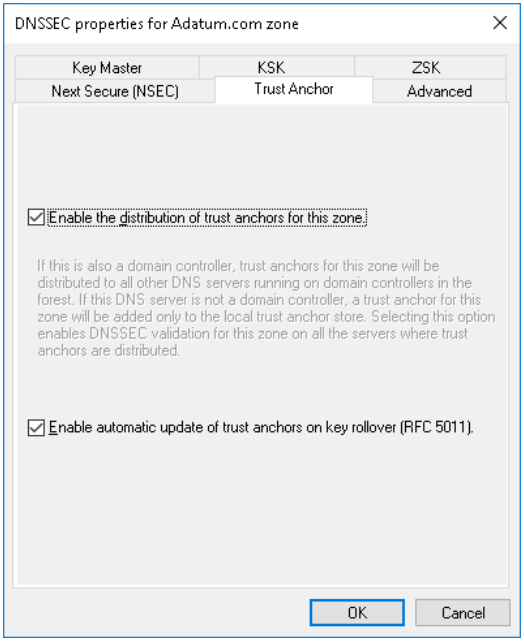

FIGURE 1-10 Enabling trust anchor distribution

3. Configure the NRPT on the client computers You must distribute the NRPT to all client computers so that they know to request validation using DNSSEC. The easiest way to achieve this is to use GPO distribution:
   - **A.** Open Group Policy Management and locate the Default Domain Policy.
   - **B.** Open this policy for editing and navigate to Computer Configuration / Policies / Windows Settings / Name Resolution Policy, as shown in Figure 1-11.
   - **C.** In the Create Rules section, type the name of your domain (for example, Adatum.com) in the Suffix text box; doing so applies the rule to the suffix of that namespace.
   - **D.** Select the Enable DNSSEC in this Rule check box, select the Require DNS Clients To Check That The Name And Address Data Has Been Validated By The DNS Server check box, and then click Create.

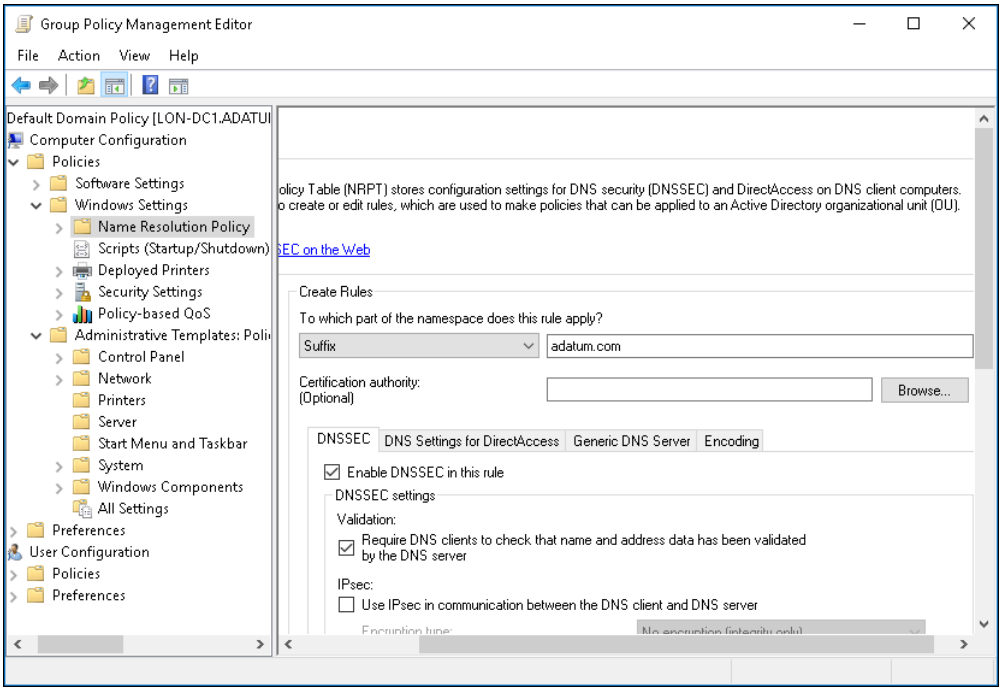

FIGURE 1-11 Creating the NRPT GPO

> ***NEED MORE REVIEW?*** **STEP-BY-STEP: DEMONSTRATE DNSSEC IN A TEST LAB**
>
> For more information about implementing DNSSEC, refer to the Microsoft TechNet website at *https://technet.microsoft.com/library/hh831411(v=ws.11).aspx*.

Configure DNS socket pool

You can use the DNS socket pool to enable a DNS server to use a random source port when issuing DNS queries. If you enable DNS socket pool the DNS server selects a source port from a pool of available sockets when the DNS service starts. This means that the DNS server avoids using well-known ports. This can help to secure the DNS server because a malicious person must guess both the source port of a DNS query and a random transaction ID to successfully run a malicious attack.

You can use the DNSCMD.exe command-line tool to configure the DNS socket pool size.

From an elevated command prompt, run the dnscmd /Config /SocketPoolSize <value> command and then restart the DNS server. You can configure the socket pool size from 0 through 10,000. The default pool size is 2,500.

Configure cache locking

When a DNS client queries a recursive DNS server, the server caches the result so that it can respond more quickly to other DNS clients querying the same information. The amount of time that a record resides in cache is determined by the Time To Live (TTL) value of the record.

During the TTL, a record can be overwritten if more recent data is available for the record. However, this potentially exposes a security issue. A malicious person might be able to overwrite the record in cache with information that could redirect clients to a site containing unsafe content.

To mitigate this risk in Windows Server 2016, you can use cache locking to determine when information in the DNS resolver cache can be overwritten. When you enable cache locking, the DNS server does not allow updates to cached records until the TTL expires.

To configure cache locking, on your DNS server, run the Set-DnsServerCache –LockingPercent <value> Windows PowerShell command. The <value> you enter is a percentage of the TTL. For example, if you type 75, then the DNS server does not allow updates to the cached record until at least 75 percent of the TTL has expired.

EXAM TIP

By default, the cache locking percentage value is 100, which means that cached entries cannot be overwritten for the entire duration of the TTL.

Enable response rate limiting

Another security feature you can use in Windows Server 2016 is response rate limiting, which is as a defense against DNS denial-of-service attacks. One common DNS denial-of-service attack is to fool DNS servers into sending large amounts of DNS traffic to particular DNS servers, thus overloading the target servers.

When a configured DNS server with response rate limiting identifies potentially malicious requests, it ignores them instead of propagating them. The DNS server can identify potentially malicious requests because many identical requests in a short time period from the same source are suspicious.

By default, response rate limiting is disabled. To enable response rate limiting, run the Set-DnsServerResponseRateLimiting Windows PowerShell command. This enables response rate limiting using the default values. You can also supply command parameters to customize response rate limiting.

NEED MORE REVIEW? **SET-DNSSERVERRESPONSERATELIMITING**

For more information about configuring DNS response rate limiting, refer to the Microsoft TechNet website at *https://technet.microsoft.com/library/mt422603.aspx*.

Configure DNS-based authentication of named entities

Windows Server 2016 supports a new feature known as DNS-Based Authentication of Named Entities (DANE). This feature relies on using Transport Layer Security Authentication (TLSA) and can help reduce man-in-the-middle type attacks on your network.

DANE works by informing DNS clients requesting records from your domain from which Certification Authority (CA) they must expect digital certificates to be issued. For example, suppose a DNS client requests the IPv4 address relating to the record https://www.adatum.com. The DNS server provides the requested IPv4 address and related information. However, the DNS server also provides information that the certificate used to authenticate the identity of the webserver www.adatum.com is provided by a particular CA.

Administering DNS

It is important that you know how to administer your DNS servers. You can use tools such as Windows PowerShell and the DNS Manager console to interactively administer the DNS servers in your organization. However, in large enterprise environments, it can be difficult to keep on top of administration of such a critical service. In these circumstances, you can consider implementing DNS policies, delegating DNS administration to a specialist team, and using DNS logging as an indicator of potential problems with DNS.

Implement DNS policies

DNS Policy is a new feature in Windows Server 2016 that enables you to control how a DNS server behaves in a particular set of circumstances. For example, we have already seen how you can implement recursion scopes to control DNS recursion based on certain factors; this is an example of a DNS policy in action.

You can create one or several DNS policies as your organizational needs dictate. However, common reasons for implementing DNS policies include:

- **Application high availability** The DNS server redirects clients to the healthiest endpoint for an application based, for example, on high availability factors in a failover cluster.
- **Traffic management** The DNS server redirects clients to the nearest server or datacenter.
- **Split-brain DNS** The DNS server responds to clients based on whether the client is external or internal to your organization's intranet.
- **Filtering** The DNS server blocks DNS queries if they are from malicious hosts.
- **Forensics** The DNS server redirects malicious DNS clients to a sinkhole instead of the host they are attempting to reach.
- **Time-of-day based redirection** The DNS server redirects clients to servers or datacenters based on the time.

To implement DNS policies, you must use Windows PowerShell commands. However, you must first be able to classify groups of records in a DNS zone, DNS clients on a specific network, or other characteristics that can help identify the DNS clients. You can use the following DNS objects to characterize your DNS clients:

- **Client subnet** The IPv4 or IPv6 subnet containing the DNS clients.
- **Recursion scope** The unique instances of a group of settings that control DNS server recursion.
- **Zone scopes** Contains its own set of DNS resource records. A record can exist in several scopes, each with a different IP address depending on the scope. DNS zones can have multiple zone scopes.

To implement DNS policies, you must first define one or more of the above objects to classify your DNS clients and scopes.

1. For example, to create a subnet for DNS clients in New York, use the following command:

```
Add-DnsServerClientSubnet -Name "NYCSubnet" -IPv4Subnet "172.16.0.0/24"
```

2. You need to create multiple client subnet objects based on the IPv4 or IPv6 subnet address.
3. Next, you create a DNS zone scope for New York DNS clients by using the following command:

```
Add-DnsServerZoneScope -ZoneName "Adatum.com" -Name "NYCZoneScope"
```

4. Again, you would need to create multiple zone scopes based on your requirements.
5. Next, to create a specific IP address record for clients in the New York City zone scope, run the following command:

```
Add-DnsServerResourceRecord -ZoneName "Adatum.com" -A -Name "www" -IPv4Address
"172.16.0.41" -ZoneScope "NYCZoneScope"
```

6. Finally, you create the policy that instructs the DNS server to respond based upon the previously defined factors:

```
Add-DnsServerQueryResolutionPolicy -Name "NYCPolicy" -Action ALLOW -ClientSubnet
"eq,NYCSubnet" -ZoneScope "NYCZoneScope,1" -ZoneName "Adatum.com"
```

Now, if a client in the New York subnet petitions a DNS server for the IPv4 address of the www.adatum.com host, the DNS server responds with the IP address 172.16.0.41. If you create other subnets and zone scopes for other locations, you could instruct the DNS server to respond with a different IP address for client queries from other locations.

NEED MORE REVIEW? **DNS POLICIES OVERVIEW**

For more information about configuring DNS policies, refer to the Microsoft TechNet website at *https://technet.microsoft.com/windows-server-docs/networking/dns/deploy/dns-policies-overview.*

Configure delegated administration

By default, the following groups have administrative capabilities over your organization's DNS servers:

- **Domain Admins** Has full permissions to manage all aspects of the DNS server in its home domain.
- **Enterprise Admins** Has full permissions to manage all aspects of all DNS servers in any domain in your AD DS forest.
- **DnsAdmins** Can view and modify all DNS data, settings, and configurations of DNS servers in their home domain.

In a small to medium network, it is generally acceptable to use these defaults. However, in large network environments, it can be beneficial to delegate administration for aspects of DNS management to different teams.

If you decide to delegate DNS Server administration to a different user or group, you can add that user or group to the DnsAdmins group for a given domain in the forest. To modify membership of this group, you can use Active Directory Users and Computers or the Windows PowerShell Add-ADGroupMember cmdlet.

To configure DNS administrative permissions, right-click the appropriate DNS server or DNS zone in the DNS Manager console, , and then click Properties. n the Server Properties or Zone Properties dialog box, on the Security tab, you can view and modify permissions for the server or zone, as shown in Figure 1-12.

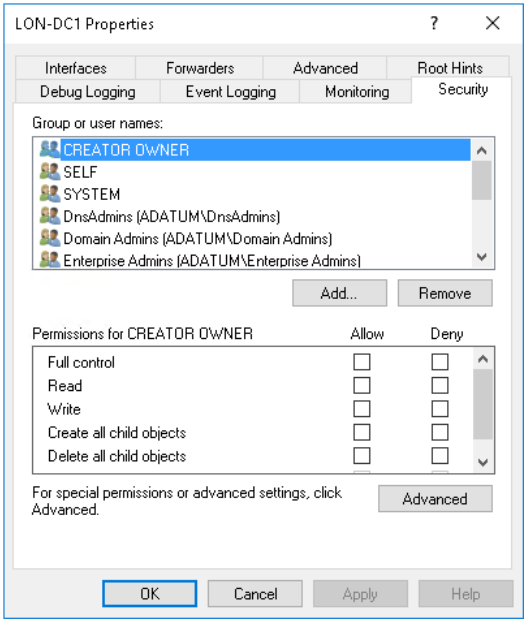

FIGURE 1-12 Delegating DNS administration

Configure DNS logging

Enabling logging can be very beneficial for proactive monitoring, especially when you are investigating poor performance or spurious and unexpected service behavior. By default, DNS records events into a DNS server log that you can review using Event Viewer. The DNS server log is located under the Application and Services Logs node, as shown in Figure 1-13.

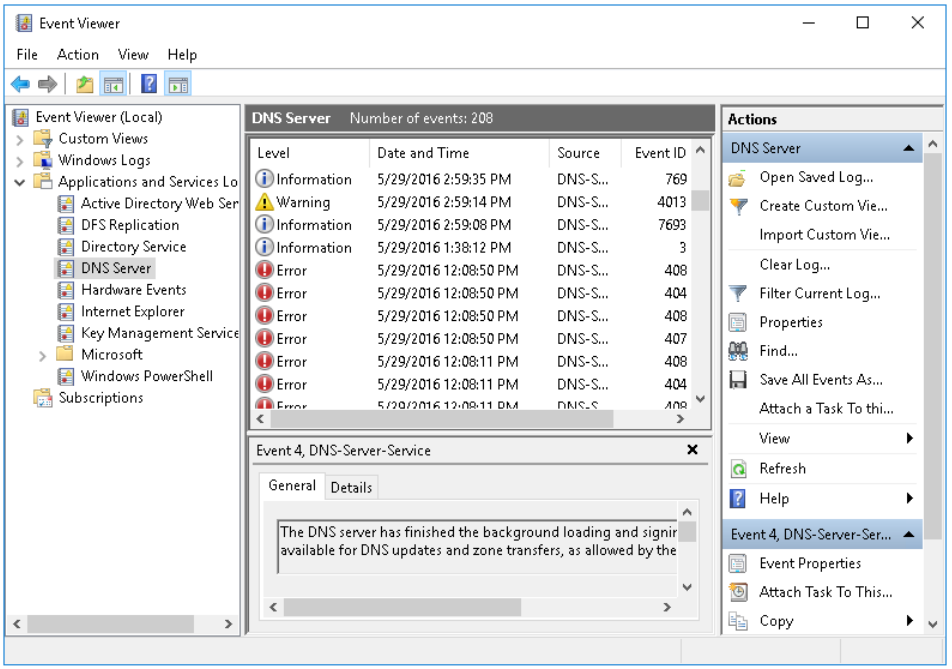

FIGURE 1-13 Viewing the DNS server event log

This log contains common DNS related events, such as service starts and stops, zone signing events, configuration changes, and common warnings and errors.

You can also enable more detailed logging with *debug* logging. However, you should exercise caution when enabling debug logging as it can impose load on the DNS server that might impact service delivery. Debug logging provides the following additional details:

- Packet direction (Outgoing or Incoming)
- Packet contents (Queries/Transfers, Updates, or Notifications)
- Transport protocol (UDP or TCP)
- Packet type (Request or Response)
- Filtering packets by IP address
- Name and location of the log file, which defaults to the %systemroot%\System32\DNS directory
- Log file maximum size limit

To enable debug logging, from the DNS Manager console:

1. Right-click the relevant DNS server, and then click Properties.
2. In the Server Properties dialog box, click the Debug Logging tab, as shown in Figure 1-14, select the Log Packets For Debugging check box, select the events for which you want the DNS server to record debug logging, and then click OK.

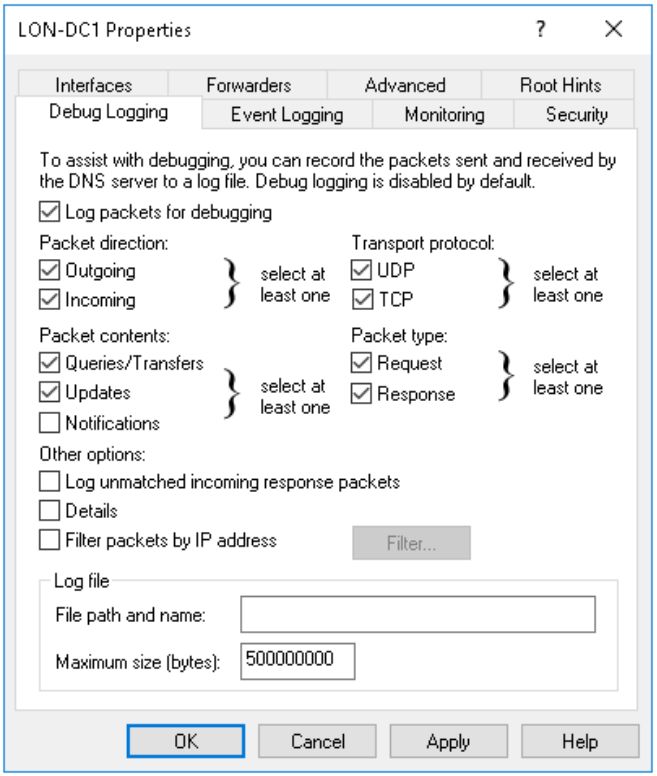

FIGURE 1-14 Configuring DNS Debug logging

Implement DNS performance tuning

The DNS server role, like other server roles and services, can be affected by the poor performance of your server. Poor performance is often caused by lack of server resources: memory, CPU, sufficient disk throughput, and network bandwidth. You can use general tools, such as Performance Monitor, to gauge whether these resources are sufficient in your server and to determine which resources are causing a bottleneck.

When any one or more of these resources is insufficient, a performance bottleneck is created. The solution is to identify which resource has the bottleneck, and to optimize that resource, often by adding more of that resource. The alternative is to distribute the load by adding additional DNS servers.

NEED MORE REVIEW? **WINDOWS PERFORMANCE MONITOR**

For more information about using Performance Monitor, refer to the Microsoft TechNet website at *https://technet.microsoft.com/library/cc749249(v=ws.11).aspx*.

The two key resources in the DNS server role are CPU and memory. The DNS tab in the Server Manager console provides a Performance pane that you can use to monitor these two critical resources, as shown in Figure 1-15.

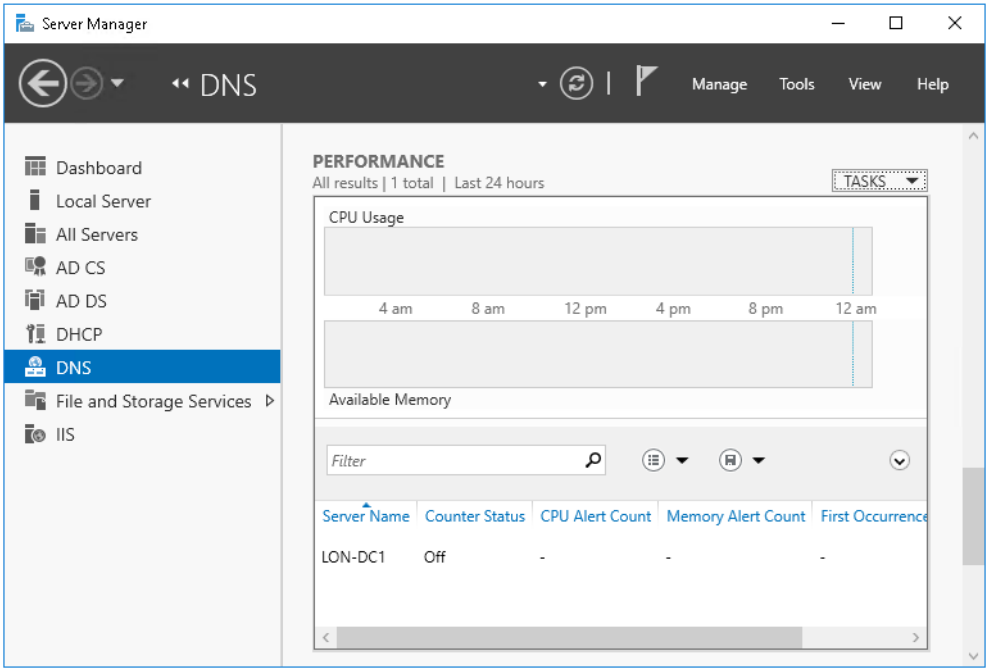

FIGURE 1-15 Monitoring DNS performance

To start monitoring these resources, click Tasks, and then click Configure Performance Alerts. In the DNS Server: Configure Performance Alerts dialog box, you can configure thresholds for alerts for both CPU (percent usage) and Memory (MB available) as shown in Figure 1-16. Click Save when you are ready.

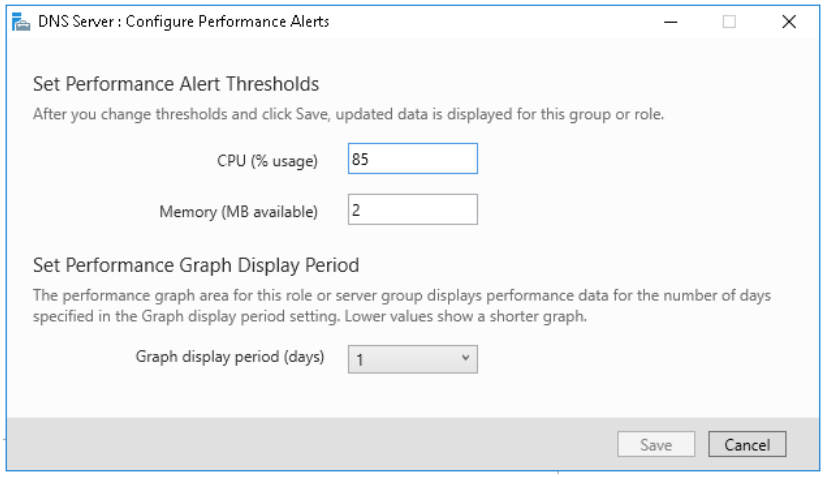

FIGURE 1-16 Configuring DNS performance alerts

Aside from these fundamental server performance characteristics, you can configure the DNS server to help to optimize DNS responsiveness. For example, allowing a DNS server to perform recursion involves imposing additional load on the DNS server when it is unable to provide an authoritative response to a client query. By disabling recursion, you can reduce the load on that DNS server, but at the cost of preventing it from using recursion. Similarly, removing root hints prevents a server from querying the Internet DNS tree on behalf of clients, which reduces workload.

Many of the performance-related decisions you make might have a functionality impact on the way name resolution works within your organization. That means you must consider that impact carefully. To help you plan DNS optimization, you should create a standard DNS server and then perform performance monitoring on the server while it is under a typical query load. You can use tools, such as the industry standard *dnsperf* tool, to help determine the optimum queries per second value for your standard server.

NEED MORE REVIEW? **NAME RESOLUTION PERFORMANCE OF AUTHORITATIVE WINDOWS DNS SERVER**

The following TechNet blog article contains a test procedure for optimizing Microsoft DNS servers at *https://blogs.technet.microsoft.com/networking/2015/08/13/name-resolution-performance-of-authoritative-windows-dns-server-2012-r2/*.

Implement DNS global settings using Windows PowerShell and configure global settings using Windows PowerShell

So far, throughout this chapter, you have seen that you can perform many of the implementation and configuration tasks on DNS servers by using Windows PowerShell. In Skill 1.2: Create and configure DNS zones and records, you explore more Windows PowerShell cmdlets for the DNS server role.

> ***NEED MORE REVIEW?*** **DOMAIN NAME SYSTEM (DNS) SERVER CMDLETS**
>
> To review a complete list of Windows PowerShell cmdlets for DNS server, refer to the Microsoft TechNet website at *https://technet.microsoft.com/library/jj649850.aspx*.

Skill 1.2: Create and configure DNS zones and records

Although DNS is based on the concept of domains and subdomains, you store information about these domains and subdomains and the relationship between them in DNS zones. You can consider a DNS zone to be one or more domains and subdomains from your DNS infrastructure.

For example, the domains Adatum.com and sales.adatum.com might both be stored in a DNS zone called Adatum.com, or sales.adatum.com might be stored in a delegated zone called sales.adatum.com, while the parent domain, Adatum.com, is stored in its own zone.

You can store the zone in files on the DNS server or in the Active Directory Domain Services (AD DS) database. It is important that you know how and when to create primary and secondary zones, delegated zones, AD DS–integrated zones, and stub zones.

Overview of DNS zones

Zones are used by DNS servers to resolve client DNS queries. Usually, clients perform forward lookup queries in which a hostname must be resolved into the corresponding Internet Protocol Version 4 (IPv4) or Internet Protocol Version 6 (IPv6) address. Forward lookup queries are resolved by reference to *forward lookup zones*.

Forward lookup zones contain a variety of DNS record type (discussed in the next section) include:

- Host (A) records
- Alias (CNAME) records
- Records that identify which server is hosting a service, such as service (SRV) records and Mail exchanger (MX) records.

Less often, a DNS client queries a DNS server for the name of a host when it has the IPv4 or IPv6 address of the host. This is called a reverse lookup, and is satisfied by reference to a *reverse lookup zone*. Reverse lookup zones contain pointer (PTR) records.

Before you create your zone, you must first determine whether the zone is a forward or reverse lookup zone. Then you must determine whether the zone is primary, secondary, or AD DS–integrated. Strictly speaking, it is not the zone that is primary or secondary. Instead, it is the local copy of the zone that is primary or secondary. In other words, for there to be a secondary zone for Adatum.com, there must already exist a primary zone for Adatum.com on another DNS server from which the secondary can obtain the zone data.

When you first deploy the DNS server role in Windows Server 2016, the DNS Manager console navigation pane contains the server node, and beneath this, nodes for Forward Lookup Zones, Reverse Lookup Zones, Trust Points, and Conditional Forwarders. These nodes are all empty until you start to create zones on the DNS server.

Configure DNS zones

Windows Server 2016 supports a number of different zone types. These include primary zones, secondary zones, and Active Directory integrated zones. It's important that you know how to create and configure these different types of zone..

Create primary zones

A primary zone is a writable copy of a DNS zone that exists on a DNS server. To create a primary zone, in the DNS Manager console, use the following procedure:

1. Right-click the Forward Lookup Zones node, and then click New Zone.
2. In the New Zone Wizard, on the Welcome To The New Zone Wizard page, click Next.
3. On the Zone Type page, select Primary Zone, as shown in Figure 1-17, and then click Next.

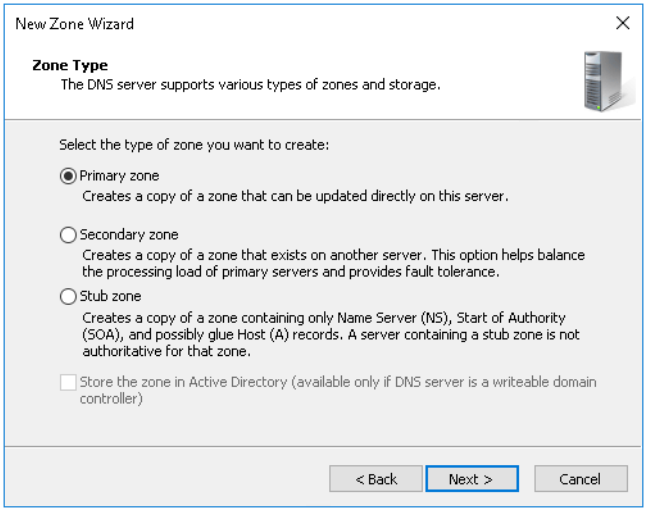

FIGURE 1-17 Creating a primary zone

4. On the Zone Name page, in the Zone name box, type the zone name. For example, type Contoso.com. Click Next.
5. On the Zone File page:
   - If you have a DNS zone file with which to populate your zone (for example, from another DNS server), click Use This Existing File, specify the path to the file, and then click Next.
   - If you do not have an existing zone file, click Create A New File With This File Name and click Next. Figure 1-18 shows the filename that is created automatically when you choose this option.

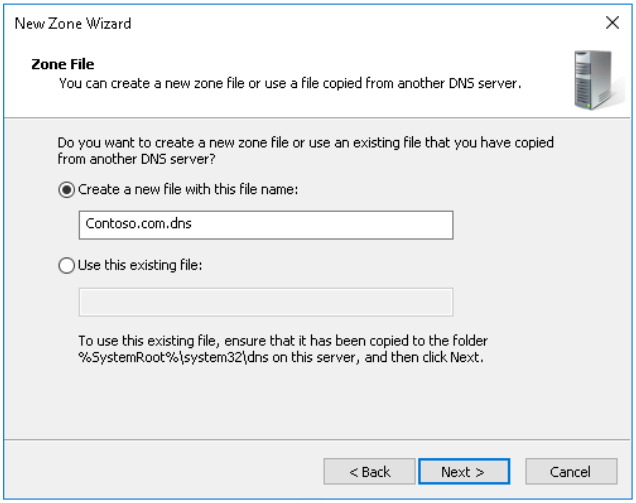

FIGURE 1-18 Defining the zone file

6. On the Dynamic Update page, shown in Figure 1-19, choose one of the following, and then click Next:
   - **Allow Only Secure Dynamic Updates (Recommended For Active Directory)** This option enables clients that support dynamic DNS to update their records in the DNS zone, such as when a client computer obtains a different IPv4 address from a Dynamic Host Configuration Protocol (DHCP) server. This option requires that each DNS record has an owner—the entity that registered the original record. Only the owner can update the record, which helps you secure your DNS records. This option is only available if you are creating an AD DS–integrated zone.
   - **Allow Both Nonsecure And Secure Dynamic Updates** This option also enables clients that support dynamic DNS to update their records in the DNS zone. It also supports nonsecure dynamic updates.
   - **Do Not Allow Dynamic Updates** Choose this option if you want to manually maintain all DNS records.

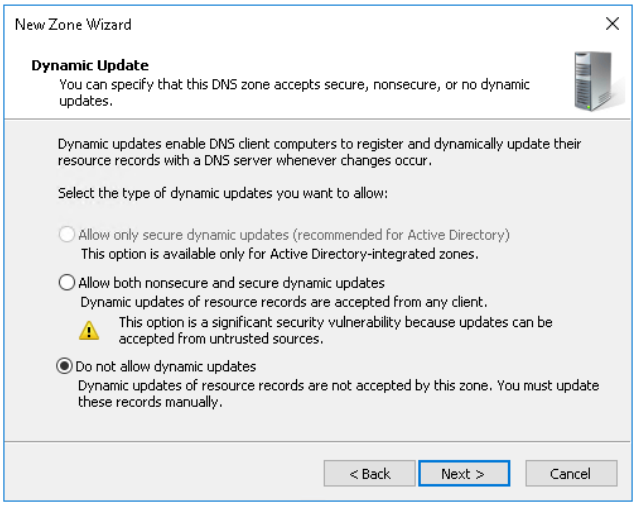

FIGURE 1-19 Choosing dynamic updates

7. On the Completing The New Zone Wizard page, click Finish.

After you have created your primary zone, you can view the initial contents of the zone by using the DNS Manager console, as shown in Figure 1-20. It contains the Start of Authority (SOA) record and a Name Server (NS) record. These two records define which computer(s) are responsible, or authoritative, for the zone.

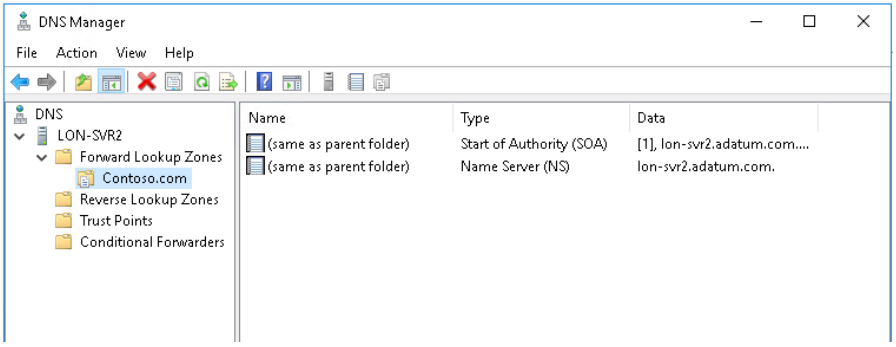

FIGURE 1-20 Viewing the completed Contoso.com zone

You can also add a primary zone by using the Add-DnsServerPrimaryZone Windows PowerShell cmdlet. For example, to complete the same process as in the preceding example by using Windows PowerShell, run the following command:

```
Add-DnsServerPrimaryZone -Name "Contoso.com" -ZoneFile "Contoso.com.dns"
-DynamicUpdate None
```

After you have created the primary zone, you can reconfigure it from the DNS Manager console by right-clicking the zone in the navigation pane and clicking Properties. You can then configure the following properties on each of the following tabs:

- **General** You can change the zone type, zone file name, the dynamic updates setting, and configure aging and scavenging.
- **Start of Authority (SOA)** Shown in Figure 1-21, you can reconfigure the SOA record. This includes the Primary server's Fully Qualified Domain Name (FQDN), the responsible person's contact details, and the Refresh, Retry, and Expire intervals. These intervals determine:
  - **Refresh interval** The frequency with which other DNS servers that host the zone must refresh the zone data.
  - **Retry interval** The interval at which other DNS servers retry a refresh operation.
  - **Expires after** The length of time after failure to refresh zone data other DNS servers assume that the zone data has expired.

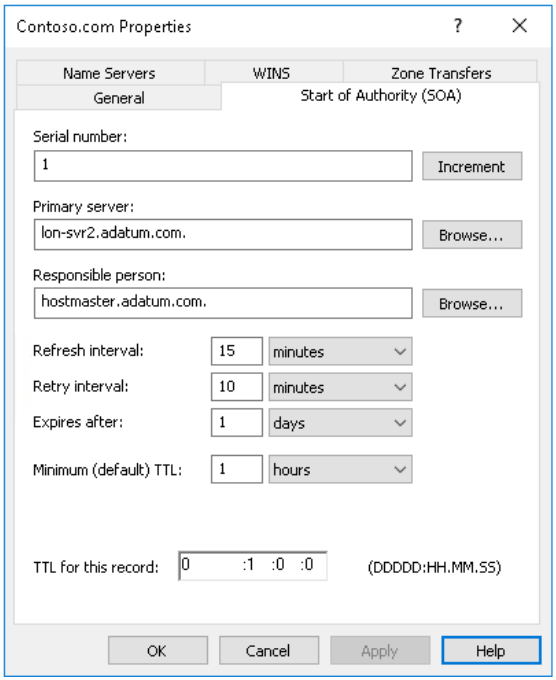

FIGURE 1-21 Editing the Contoso.com DNS zone properties

The Start of Authority (SOA) tab also contains the Minimum (Default) TTL value. This is the value that determines how long records in this zone can be cached by other recursive DNS servers.

- **Name Servers** Use this tab to add, remove, or edit the name and IP addresses of other DNS servers that host this zone.
- **Zone Transfers** Use this tab to configure how the zone data is transferred to other name servers hosting copies of the zone.
- **WINS** Use this tab to configure Windows Internet Name Service (WINS) and DNS integration. WINS supports the resolution of NetBIOS names. Less relevant today, NetBIOS names use a nonhierarchical structure based on a 16-character name. Enabling the Use WINS Forward Lookup option enables the DNS server to respond to requests for NetBIOS names without the client computer having to petition a WINS server directly.

You can configure the zone properties by using the Set-DnsServerPrimaryZone Windows PowerShell cmdlet. For example, to change the Contoso.com Primary Zone Dynamic Update settings with Windows PowerShell, run the following command:

```
Set-DnsServerPrimaryZone -Name "Contoso.com" -DynamicUpdate "NonsecureAndSecure"
```

NEED MORE REVIEW? **SET-DNSSERVERPRIMARYZONE**

To review further details about modifying primary zone properties with Windows PowerShell, refer to the Microsoft TechNet website at *https://technet.microsoft.com/en-us/library/jj649865.aspx*.

Create and configure secondary zones

Creating a secondary zone is a different process from a primary zone. This is because a secondary zone hosts a read-only copy of a zone, which it obtains from another DNS server.

To create a secondary zone, you must know the name of the zone, and have the name and IP address of a DNS server that hosts a copy of the zone.

EXAM TIP

The name server you specify as a source for a secondary zone does not have to be hosting a primary copy of the zone. You can point one secondary zone server to another secondary zone server. However, somewhere a primary copy of the zone must exist.

You can use the DNS Manager console to create a secondary zone. To do this, use the following procedure:

1. Right-click the Forward Lookup Zones node, and then click New Zone.
2. In the New Zone Wizard, on the Welcome To The New Zone Wizard page, click Next.
3. On the Zone Type page, select Secondary Zone, and then click Next.
4. On the Zone Name page, in the Zone Name box, type the zone name, and click Next.

5. On the Master DNS Servers page, in the Master Servers list, type the FQDN or IP address of the server that hosts a copy of the zone, press Enter, and then click Next, as shown in Figure 1-22.

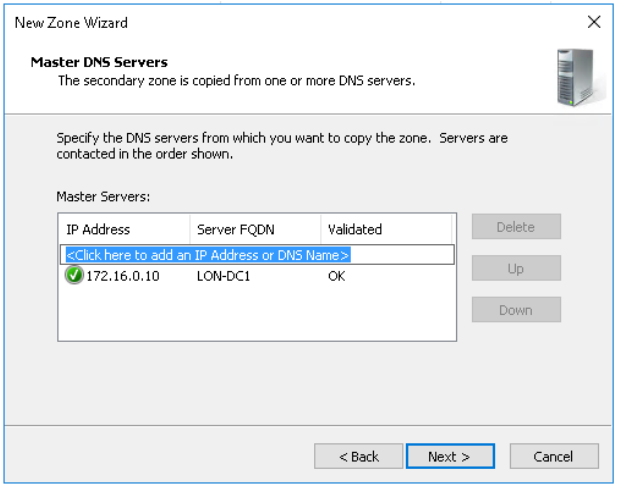

FIGURE 1-22 Defining the master server for a secondary zone

6. On the Completing The New Zone Wizard page, click Finish.

After you have added the secondary zone, it is necessary to configure the master DNS server that you specified. This is to enable zone transfers to your secondary server. To perform this step, switch to the DNS Manager console on the master server and perform the following procedure:

1. Right-click the appropriate zone, and then click Properties.
2. On the Name Servers tab, in the Name servers list, click Add to specify the FQDN and IP address of the DNS server hosting the secondary copy of the zone, as shown in Figure 1-23. Click OK.

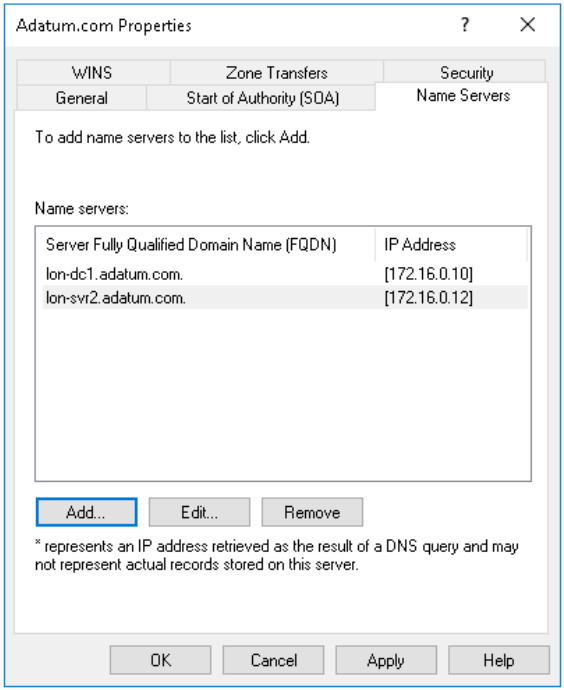

FIGURE 1-23 Configuring the Name Servers list

3. Click the Zone Transfers tab.
4. Select the Allow Zone Transfers check box. Then, as shown in Figure 1-24, choose one of the following:
   - To Any Server.
   - Only To Servers Listed On The Name Servers Tab.
   - Only To The Following Servers (If you choose this option, you must click Edit to specify the list of name servers that you want to allow).

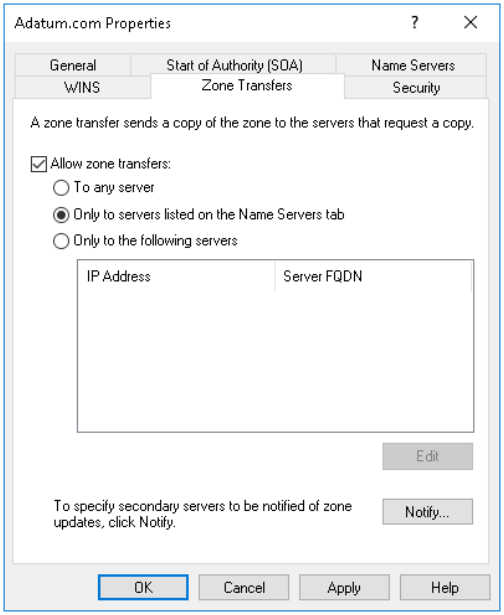

FIGURE 1-24 Configuring zone transfers

5. Click Notify.
6. In the Notify dialog box, either select Servers Listed On The Name Servers Tab, or else click The Following Servers, and then type the IP addresses of the secondary name servers you want to notify.
7. Click OK twice to complete configuration. Next, switch back to the DNS Manager console on the server hosting the secondary zone. You should see the DNS records populate into the secondary zone. If this does not happen immediately, right-click the secondary zone, and then click Transfer From Master.

You can use the Add-DnsServerSecondaryZone Windows PowerShell cmdlet to create a secondary zone. For example, the following command creates a secondary zone for the Adatum.com zone:

```
Add-DnsServerSecondaryZone -Name "Adatum.com" -ZoneFile "Adatum.com.dns"
-MasterServers 172.16.0.10
```

Configure delegation

DNS delegation is when a DNS server delegates authority over a part of its namespace to one or more other DNS servers. For example, Adatum.com and sales.adatum.com could be hosted in the same zone, Adatum.com, with the sales.adatum.com merely being a subdomain record. In this case, the authoritative DNS servers for Adatum.com and sales.adatum.com are the same. There is no need for the DNS servers in Adatum.com to refer recursive DNS servers to another domain.

Alternatively, you could create a separate zone for both Adatum.com and sales.adatum.com, each with their own DNS servers. Because one domain, sales.adatum.com, is a child domain of another domain, Adatum.com, there must exist a method to enable the authoritative name servers for the subdomain to be located. This method is called *delegation*, and is essentially a pointer to the authoritative name servers for a subdomain.

In Figure 1-25, you can see two DNS zones: Adatum.com, which contains a subdomain, marketing.adatum.com, and a second zone, sales.adatum.com, which contains a single domain, sales.adatum.com.

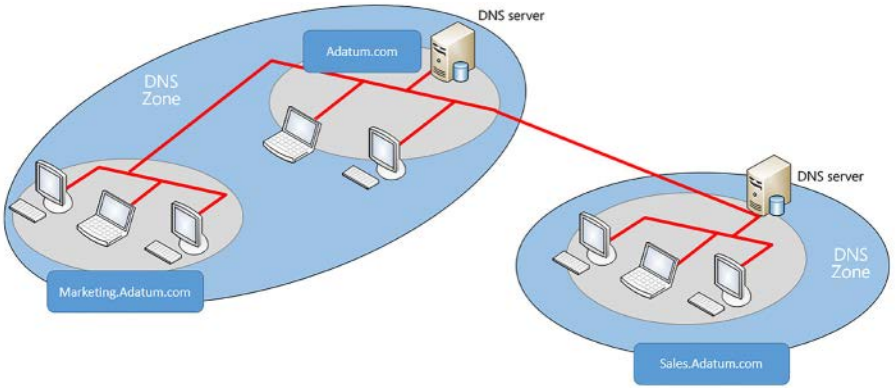

FIGURE 1-25 The Adatum.com DNS namespace separated into two zones

When determining whether to delegate a subdomain, consider the following:

- Your DNS zones are large, and delegation enables you to distribute the zone into smaller pieces across your organization.
- Organizational changes, such as mergers and acquisitions, mean that you have additional subdomains to manage.
- You have a distributed management structure, and want different departments or locations to be responsible for managing their own DNS namespaces.

To create a DNS delegation, in the DNS Manager console, perform the following procedure:

1. Right-click the parent zone. For example, right-click Adatum.com, and then click New Delegation. The New Delegation Wizard launches.
2. In the New Delegation Wizard, on the Welcome page, click Next.
3. On the Delegated Domain Name page, as shown in Figure 1-26, in the Delegated domain box, type the subdomain name. For example, type Sales. The suffix is added automatically. Click Next.

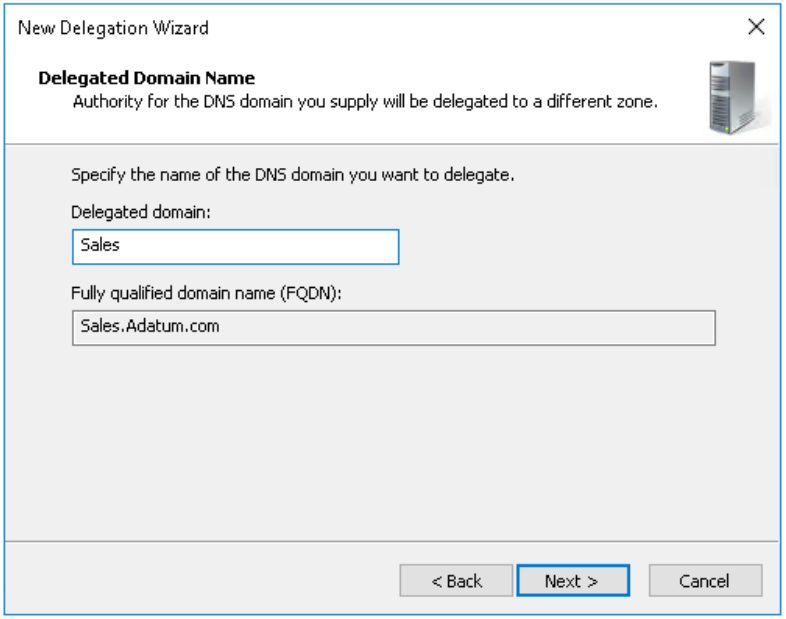

FIGURE 1-26 Delegating the sales.Adatum.com zone

4. On the Name Servers page, click Add.
5. In the New Name Server Record dialog box, on the Server Fully Qualified Domain name (FQDN) box, type the name of the DNS server that hosts the new delegated zone, click Resolve, and then click OK.
6. On the Name Servers page, click Next, and then click Finish.

You can use the Add-DnsServerZoneDelegation Windows PowerShell cmdlet to create a delegated zone in an existing zone. For example, the following command creates the sales.adatum.com delegated zone in the existing Adatum.com zone:

```
Add-DnsServerZoneDelegation -Name "Adatum.com" -ChildZoneName "Sales" -NameServer
"ns1.Sales.Adatum.com" -IPAddress 172.16.0.136
```

After you have completed the delegation, if necessary, you should install DNS on the name server that you specified in the wizard, and create the delegated zone, in this case sales.adatum.com.

NEED MORE REVIEW? **UNDERSTANDING ZONE DELEGATION**

To review further details about delegating DNS zones, refer to the Microsoft TechNet website at *https://technet.microsoft.com/library/cc771640(v=ws.11).aspx*.

Configure Active Directory integration of primary zones

Traditional DNS zones are file-based and are stored in the local file system of the DNS server. DNS servers that host the primary copy of a zone have a writable version of the DNS zone file. Secondary servers have read-only copies of the zone file; they periodically obtain updates by using a zone transfer from their configured master, as you saw in Create and configure secondary zones.

In an AD DS environment, you have the option to create AD DS-integrated zones. In this situation, all copies of the zone data are writable. In addition, the zone data is stored securely in Active Directory and is replicated securely as part of the AD DS database.

The benefits of using AD DS-integrated zones are:

- **Multimaster updates** AD DS-integrated DNS zones are multimaster, and updates can be made to any copy of the zone data. This provides for redundancy in your DNS infrastructure. If your organization implements dynamic updates to the DNS zone, then geographically remote DNS clients can update their records by connecting to the nearest DNS server.
- **Replicated using AD DS replication** AD DS replication is based at the attribute-level. This means that only changed attributes, rather than entire records, are replicated. This means that the volume of zone transfer traffic can be reduced.
- **Secure dynamic updates** You can implement secure dynamic updates in an AD DS–integrated zone. This is discussed in the next section.
- **Improved security** You can delegate administration of AD DS-integrated zone, domains, and resource records with the AD DS object-level Access Control List (ACL) for the zone.

EXAM TIP

When you promote a new domain controller in your AD DS forest, the DNS server role deploys automatically. This is configurable on the Domain Controller Options page of the Active Directory Domain Services Configuration Wizard.

When you create zones on a DNS server that is also a domain controller, you have the option to install an AD DS-integrated zone. To create an AD DS-integrated DNS zone, use the following procedure:

1. On your domain controller, open DNS Manager.
2. Right-click the Forward Lookup Zones node, and then click New Zone.
3. In the New Zone Wizard, on the Welcome To The New Zone Wizard page, click Next.
4. On the Zone Type page, select Primary Zone, as shown in Figure 1-27, select the Store The Zone In Active Directory (Available Only If The DNS Server Is A Writable Domain Controller) check box, and then click Next.

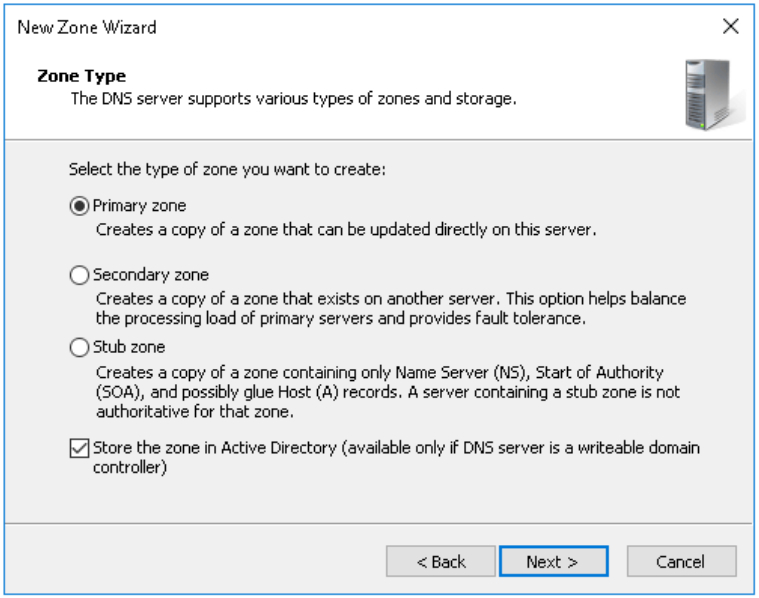

FIGURE 1-27 Selecting the zone type

5. On the Active Directory Zone Replication Scope page, as shown in Figure 1-28, select the appropriate zone replication option from the following:
   - **To All DNS Servers Running On Domain Controllers In This Forest** This option causes the zone data to replicate to all domain controllers running the DNS server role in the forest.
   - **To All DNS Servers Running On Domain Controllers In This Domain** This option (the default) causes the zone data to replicate to all domain controllers running the DNS server role in the current AD DS domain.
   - **To All Domain Controllers In This Domain (For Windows 2000 Compatibility)** This option provides backward compatibility with earlier versions of Windows Server. You would not normally select this option.
   - **To All Domain Controllers Specified In The Scope Of This Directory Partition** Directory partitions enable you to create an AD DS replication boundary that is not restricted to all domain controllers in the forest or local domain. The option is only available if you have created a directory partition before you configure the DNS zone.

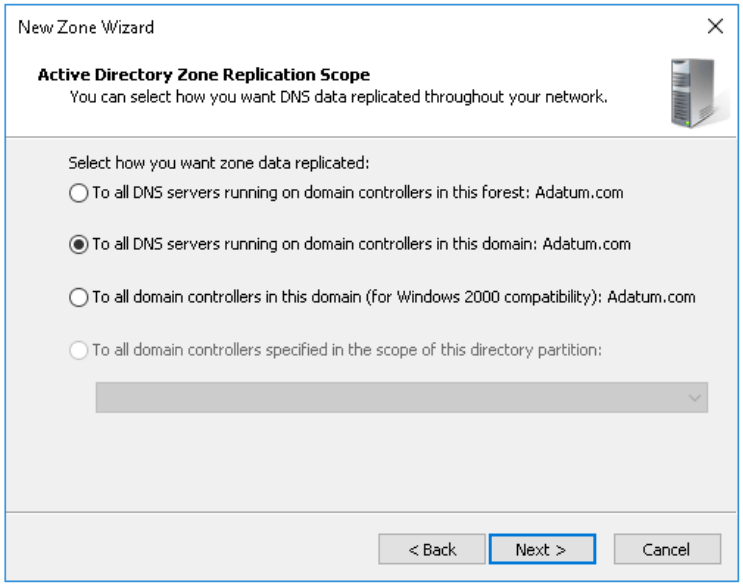

FIGURE 1-28 Specifying the preferred zone replication scope

6. Click Next.
7. On the Zone Name page, in the Zone name box, type the zone name, for example, type Contoso.com. Click Next.
8. On the Dynamic Update page, choose one of the following, and then click Next.
   - Allow Only Secure Dynamic Updates (Recommended For Active Directory)
   - Allow Both Non-Secure And Secure Dynamic Updates
   - Do Not Allow Dynamic Updates
9. On the Completing The New Zone Wizard page, click Finish.

You can also create an AD DS-integrated primary zone by using the Add-DnsServerPrimaryZone Windows PowerShell cmdlet. For example, to complete the same process as in the preceding example by using Windows PowerShell, run the following command:

```
Add-DnsServerPrimaryZone -Name "Contoso.com" -ReplicationScope "Domain"
```

On domain controllers, existing standard primary zones can be converted to AD DS–integrated zones. In DNS Manager, right-click the zone, and then click Properties. On the General page, click Change, and then select the Store The Zone In Active Directory (Available Only If The DNS Server Is A Writable Domain Controller) check box. Click OK twice.

Configure secure dynamic updates

If you have implemented an AD DS-integrated primary zone, you have the option of enabling secure dynamic updates. Dynamic updates is a feature in which DNS clients can update their own DNS records on their configured DNS server. This is particularly convenient when an organization assigns IP configuration to networked clients by using DHCP. If a client obtains a different IP address from a DHCP scope, they can register this change automatically on DNS.

With secure dynamic updates, the DNS server assigns ownership to the registered DNS records, and only the owner—the original DNS client—can update the records. To enable secure dynamic updates, you can choose one of the following options:

- Select the Allow Only Secure Dynamic Updates (Recommended For Active Directory) option on the Dynamic Updates page of the New Zone Wizard when you create an AD DS-integrated primary zone.
- After creating the AD DS-integrated primary zone, in DNS Manager, right-click the DNS zone, and then click Properties. On the General page, in the Dynamic Updates list, click Secure Only.
- After creating the AD DS-integrated primary zone in Windows PowerShell, use the Set-DnsServerPrimaryZone command. For example:

```
Set-DnsServerPrimaryZone -Name "Contoso.com" -DynamicUpdate "Secure"
```

Create and configure stub zones

You can use conditional forwarding as a means to redirect query traffic to a designated DNS server. With conditional forwarding, if a DNS query contains a specific domain name, for example Contoso.com, it is forwarded to a specific DNS server. To learn more about Conditional Forwarding, see "Configure forwarders, root hints, recursion, and delegation."

Alternatively, you can also use a stub zones to achieve a similar result. A stub zone is used by a DNS server to help resolve names between two separate DNS namespaces, such as following a merger or acquisition. A stub zone differs from conditional forwarding in that the stub zone contains the complete list of DNS servers for the other domain.

COMPARING CONDITIONAL FORWARDING WITH STUB ZONES

Imagine that two DNS namespaces, Adatum.com and Contoso.com, are now owned by the A. Datum Corporation following an acquisition. For DNS clients in the Adatum.com domain to locate resources in the Contoso.com domain it requires the use of root hints by Adatum.com DNS servers.

To avoid this, in the Adatum.com domain, you could configure DNS conditional forwarding for the Contoso.com domain. With conditional forwarding, you configure to which DNS server(s) in the Contoso.com domain to forward DNS queries.

You can also use a stub zone for Contoso.com in the Adatum.com domain. This stub zone contains the complete list of DNS server that are authoritative for the foreign domain. This list of servers is updated automatically.

When considering whether to use conditional forwarding or stub zones, remember:

- You must manually maintain conditional forwarding records, while stub zones are maintained automatically.
- With conditional forwarding, you can designate the specific foreign DNS server to forward queries to, but with a stub zone, you cannot.

CREATING A STUB ZONE

You can use the following procedure to create a stub zone. Open DNS Manager, and then:

1. Right-click the Forward Lookup Zones node, and then click New Zone.
2. On the New Zone Wizard, on the Welcome to the New Zone Wizard page, click Next.
3. On the Zone Type page, select Stub Zone, and then click Next.
4. On the Zone Name page, in the Zone name box, type the DNS domain name for the foreign domain, and then click Next.
5. On the Zone File page, if you have a DNS zone file that you use to populate your zone (for example, from another DNS server), click Use This Existing File, specify the path to the file on the Zone File page, and then click Next.
6. On the Master DNS Servers page, in the Master Servers list, type the IP address or FQDN of the DNS server in the foreign domain from which the DNS server obtains zone updates, and then click Next.
7. Click Finish to create the stub zone.

You must now populate the stub zone with the required records that includes the Start of Authority (SOA) record, and the Host (A) and NS records that pertain to the foreign DNS servers, and are retrieved from the specific master server(s). To manually perform this task, in the DNS Manager, right-click the stub zone and then click Transfer from Master.

You can use the Windows PowerShell Add-DnsServerStubZone cmdlet to create a stub zone. For example, to create a stub zone for Contoso.com, use the following command:

```
Add-DnsServerStubZone -Name "Contoso.com" -MasterServers "172.16.0.66" -ZoneFile
"Contoso.dns"
```

EXAM TIP

You can create AD DS-integrated stub zones, either in the DNS Manager console, or by using Windows PowerShell. To use Windows PowerShell, replace the zonefile parameter with ReplicationScope.

Configure a GlobalNames zone

Some older networked apps rely on a non-hierarchical naming standard known as NetBIOS. In the past, network clients that accessed these apps needed to be able to resolve these single-label NetBIOS names. You can use the WINS server feature to provide for NetBIOS name registration, resolution, and release.

The disadvantages of using WINS are:

- Organizations must maintain two name services, with the resultant administrative overhead.
- Network clients potentially use both DNS and WINS to resolve names, resulting in possible name resolution delay.

As an alternative to WINS, you can use the DNS GlobalNames zone in Windows Server 2016. When clients resolve single-label names, such as LON-SVR2, these names are resolved by reference to the GlobalNames zone. An organization has only a single GlobalNames zone, which you must create manually. Also, you must populate the zone with the required CNAME resource records that point to your organization's server and app resources.

To create the GlobalNames zone, use the following procedure:

1. Open Windows PowerShell.
2. Run the Set-DnsServerGlobalNameZone –AlwaysQueryServer $true command to enable GlobalNames zone support.
3. Run the Add-DnsServerPrimaryZone –Name GlobalNames –ReplicationScope Forest command to create the GlobalNames zone.
4. Open DNS Manager and locate the GlobalNames zone node.
5. Create the required CNAME records for server resources that still use single-label names.

> ***NEED MORE REVIEW?*** **DEPLOYING A GLOBALNAMES ZONE**
>
> **To review further details about the GlobalNames zone, refer to the Microsoft TechNet website at *https://technet.microsoft.com/library/cc731744(v=ws.11).aspx*.**

Configure DNS records

Zones contain DNS records that point either to name servers, hosts, services, or to other zones. After you have created your zones, you must populate them with records appropriate to your organization's network. You must also be prepared to maintain these records to ensure the accuracy of zone data.

Create and configure DNS resource records

A DNS server exists to provide name resolution for DNS clients. In order for this to be possible, you must populate the DNZ zones that you create with appropriate DNS resource records. These resource records include:

- **Host** A host record—commonly given the abbreviation A—holds the IPv4 address for the specified hostname. AAAA records hold the IPv6 address for the specified hostname. These are probably the most common resource records and are found in forward lookup zones.
- **Pointer** Also known as PTR records, these enable a DNS client to resolve an IPv4 or IPv6 address into a hostname. These records exist in reverse lookup zones.
- **Start of Authority** Created when you create a primary zone, the SOA record contains information about the authoritative server for the zone, contact information for the zone, and other information, including TTL values for resource records in the zone.
- **Name Server** Name server (NS) records identify the authoritative name servers in the zone, including both primary and secondary servers. They also identify name servers for any delegated zones.
- **Service Location** Known as SRV records, these enable you to specify by service, protocol, and DNS domain name, which servers are hosting particular apps or services. If a DNS client is looking for a web server, for example, you can configure SRV records for the http service enabling clients to find all servers providing that service. Clients then use corresponding A or AAAA records to resolve the server names into IP addresses. An SRV record is more complex than many others, and contains the following fields: _Service.Proto.Name TTL Class SRV Priority Weight Port Target. For example: http._tcp.Contoso.com. IN SRV 0 0 80 www.Contoso.com. AD DS services, such as the Kerberos authentication service, use SRV records to advertise themselves to clients in an AD DS network.
- **Alias** CNAME records enable you to create an alias for a host. For example, the server lon-svr2.adatum.com might host a website. You can create an alias for this host with the name www by adding a CNAME record that points to the FQDN of the lon-svr2 server.
- **Mail Exchanger** MX records are used by Simple Mail Transfer Protocol (SMTP) hosts for transferring email around the Internet. In order for an originating SMTP host to route mail to a recipient with the email address Dave@Contoso.com, it is necessary for the originating host to know which hosts can receive the email at Contoso.com. You create MX records in the Contoso.com namespace to advertise which hosts provide this service. To help to ensure a reliable inbound email flow, you can advertise several hosts by using multiple MX records. Each can be assigned the same or different mail server priorities, or preference values. If you implement MX records with the same priority, email is routed to them randomly, distributing the load. If you use different values, the lower value is used first by the originating server, thereby enabling you to specify a preferred inbound server.

NOTE **ENABLING DYNAMIC UPDATES**

Remember that you can enable dynamic updates for your DNS zone in Microsoft DNS. This enables hosts to register and update their own resource records. This is particularly relevant for A, AAAA, PTR, and SRV records that might commonly be expected to change.

NEED MORE REVIEW? **RESOURCE RECORD TYPES**

To find out more about common DNS resource record types, refer to the Microsoft TechNet website at *https://technet.microsoft.com/library/cc958958.aspx*.

You can create these records manually from the DNS Manager console. Right-click the appropriate forward lookup zone (reverse lookup zone for PTR records), and then click the appropriate option for the record type you want to create, as shown in Figure 1-29.

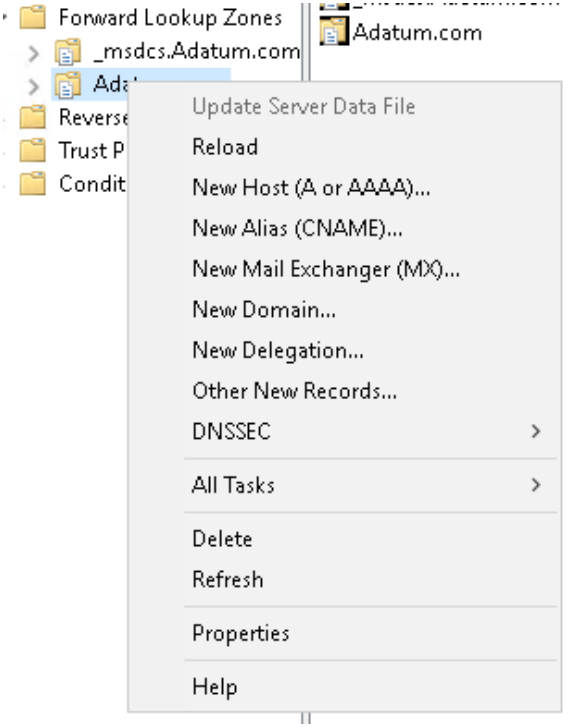

FIGURE 1-29 Creating resource records

Creating the resource records are straightforward, but vary slightly for each record type because you must specify different information for each separately. However, this is a very intuitive process. For example, for a new host record, you must specify the host's name and its IP address. You can also select the option to create a PTR record for the host automatically. For an MX record, you must specify the FQDN of the host that provides SMTP email support, and a Mail server priority value.

If the record type you want is not listed on the context menu, (for example, SRV), select Other New Record from the context menu. Then select the record type you want in the Resource Record Type dialog box, and then click Create Record, shown in Figure 1-30. This list contains all record types, including those used less frequently.

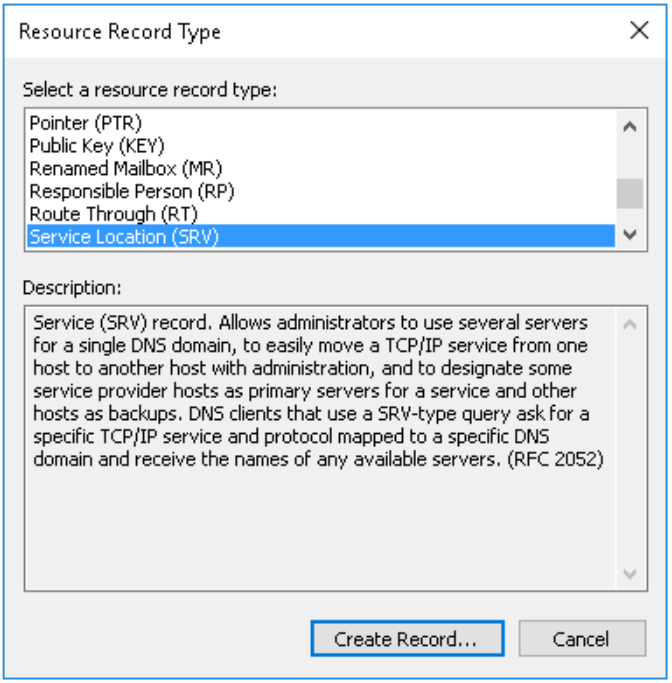

FIGURE 1-30 Selecting resource record types

You can use the Add-DnsServerResourceRecord Windows PowerShell cmdlet to create resource records. For example, the following command creates a host called lon-svr2 in the Contoso.com domain:

```
Add-DnsServerResourceRecord -ZoneName "Contoso.com" -A -Name "lon-svr2"
-AllowUpdateAny -IPv4Address "172.16.0.27" -TimeToLive 01:00:00 -AgeRecord
```

NEED MORE REVIEW? **ADD-DNSSERVERRESOURCERECORD**

To find out more about adding DNS resource records with Windows PowerShell, refer to the Microsoft TechNet website at *https://technet.microsoft.com/library/jj649925.aspx*.

Configure zone scavenging

Resource records often remain in a DNS zone even though the host that registered the record is no longer active. This is known as a stale record. You can use aging settings to determine when the DNS role can remove a stale record. This removal is known as scavenging.

Two parameters determine scavenging behavior:

- **No-refresh Interval** The no-refresh interval is the period of time that the record is not eligible to be refreshed. By default, this is also seven days.
- **Refresh Interval** The refresh interval is the time that the record is eligible to be refreshed by the client. The default is seven days.

Usually, a client host record cannot be refreshed for seven days after it is first registered (or refreshed). But then it must be refreshed within the following seven days. If it is not, the record becomes eligible to be scavenged.

EXAM TIP

By default, aging and scavenging of resource records is disabled.

To enable scavenging, you must enable it on both the DNS zone(s) and on the server(s) that host those zones.

To configure aging and scavenging on a DNS zone:

1. In DNS Manager, right-click the appropriate zone, and then click Properties.
2. On the General tab, click Aging.
3. In the Zone Aging/Scavenging Properties dialog box, shown in Figure 1-31, select the Scavenge Stale Resource Records check box, and then configure your preferred No-Refresh Interval and Refresh Interval values.
4. Click OK twice.

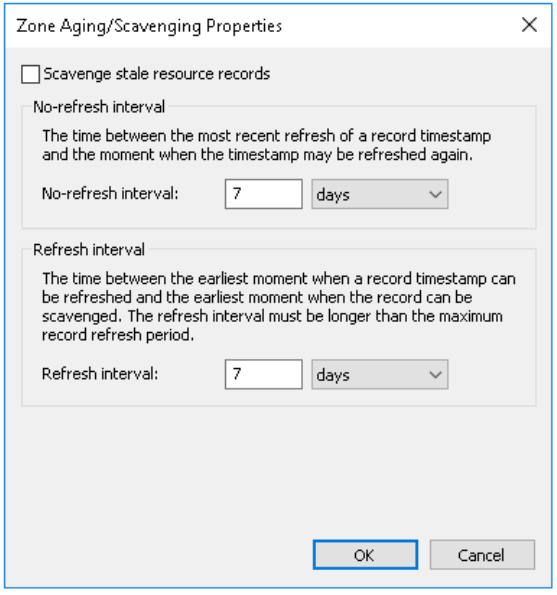

FIGURE 1-31 Enabling and configuring zone aging and scavenging

You can also enable zone aging/scavenging with the Set-DnsServerZoneAging Windows PowerShell cmdlet. For example:

```
Set-DnsServerZoneAging Contoso.com -Aging $True -ScavengeServers 172.16.0.10
```

EXAM TIP

You can configure aging/scavenging for all zones by right-clicking a server in the DNS Manager console and then clicking Set Aging/Scavenging for All Zones.

To configure scavenging on a DNS server, in the DNS Manager console:

1. Right-click the appropriate DNS server and then click Properties.
2. In the Server Properties dialog box, click the Advanced tab.
3. Select the Enable Automatic Scavenging Of Stale Records check box, and then in the Scavenging period box, specify the number of days, and then click OK.

You can enable DNS server aging/scavenging with the Set-DnsServerScavenging Windows PowerShell cmdlet. For example:

```
Set-DnsServerScavenging -RefreshInterval 7.00:00:00
```

EXAM TIP

You can force scavenging to be initiated by right-clicking a server in the DNS Manager console and then clicking Scavenge Stale Resource Records. Alternatively, use the Start-DnsServerScavenging Windows PowerShell command.

NEED MORE REVIEW? **UNDERSTANDING AGING AND SCAVENGING**

To find out more about aging and scavenging, refer to the Microsoft TechNet website at *https://technet.microsoft.com/library/cc771677(v=ws.11).aspx*.

Configure record options

You can configure a number of options for resource records, including preference, weight, priority, and tie to live.

CHANGING PREFERENCE, WEIGHT, AND PRIORITY

Some resource records are assigned preference, weight, and priority values. These are used when there are multiple records that point to the same service or server and you want to control which servers are used first. For example, an AD DS domain controller registers its Kerberos authentication service DNS resource records with a priority value of 0 and a weight of 100, as shown in Figure 1-32.

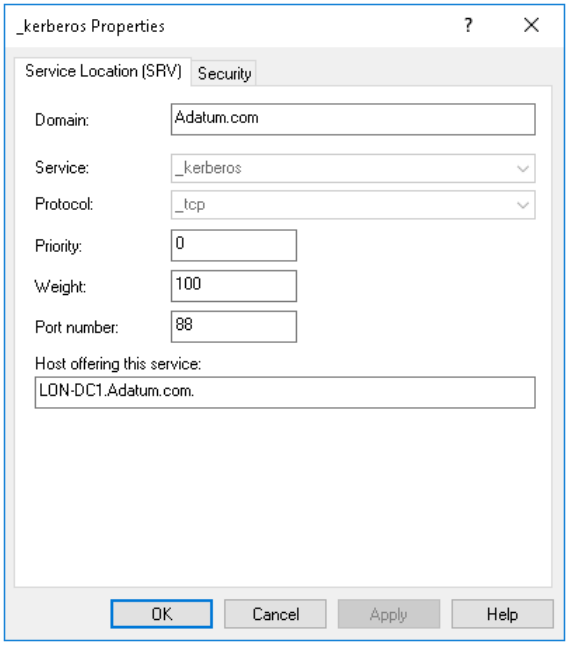

FIGURE 1-32 Viewing resource record Priority and Weight values

You can adjust these initial values to determine which Kerberos authentication server is used by clients. DNS clients attempt to use the server with the lowest priority value. If multiple servers have the same priority value, clients use the server in proportion to their weight values. Similarly, if multiple MX records exist for an email domain name, then the server with the lowest preference value is used. You can adjust these values using the DNS Manager console or the Windows PowerShell Add-DnsServerResourceRecord or Set-DnsServerResourceRecord cmdlets using the Priority, Weight, and Preference parameters.

CHANGING TIME TO LIVE VALUES

All resource records have a time to live (TTL) value. The TTL is used to determine how long a record can reside in the DNS cache of a DNS client or DNS server that has recently resolved the record. These values should be representative of how frequently resource records change in your organization. If record values are fairly static, a higher TTL is acceptable. If they change frequently, then setting a TTL too high results in DNS cache being out of date, with resultant name resolution errors.

EXAM TIP

To change individual TTL values for resource records in a zone by using the DNS Manager console, you must enable the Advanced view. In DNS Manager, click View, and then click Advanced.

To change the TTL of a record in DNS Manager:

1. Right-click a resource record, and then click Properties.
2. In the Record Properties dialog box, in the Time to live (TTL) box, type the preferred value in days, hours, minutes, and seconds and then click OK.

Alternatively, use the Windows PowerShell Add-DnsServerResourceRecord or Set-DnsServerResourceRecord cmdlets with the -TimetoLive parameter.

Configure unknown record support

Windows Server 2016 adds support in the DNS server role for *Unknown Records*. Unknown Records are resource records with a format that is foreign to the Microsoft DNS server. Support for these unknown records means that you can add the unsupported record types into your zones to support specific apps that require them. The Windows DNS server does not perform record-specific processing for these unknown records, but does respond to DNS client requests for these records.

EXAM TIP

You can use the Add-DnsServerResourceRecord Windows PowerShell cmdlet with the -Unknown parameter to create unknown resource records.

Configure round robin

You can use DNS round robin to help to distribute load across servers providing the same service. For example, if you had three web servers in your network and you wanted to distribute the client load across all three equally, one solution is to configure the same host resource record (www.contoso.com) and point it to three different IP addresses. For example:

www.contoso.com 60 IN A 172.16.0.10

www.contoso.com 60 IN A 172.16.0.12

www.contoso.com 60 IN A 172.16.0.14

Round robin works by responding to each client request for the resolution of www.contoso.com with a differently ordered list of IP addresses. On the first request, the server with the IPv4 address of 172.16.0.10 is returned at the top of the list. Next, 172.16.0.12 is returned first on the list, and so on.

You configure round robin on DNS server on the Advanced server settings dialog box. It is enabled by default. You can also use the Set-DnsServer Windows PowerShell cmdlet.

EXAM TIP

You can also use netmask ordering to achieve a similar result, but in this case, a client receives the result that is most relevant to their location, based on their subnet.

Configure DNS scopes

Windows Server 2016 supports two types of DNS scopes. These are recursion scopes and zone scopes. Recursion scopes are a collection of settings that define recursion behavior in a DNS zone, while zone scopes are collections of resource records. To create, configure, and apply DNS scopes, you must use Windows Server 2016 DNS policies.

> ***NOTE*** **RECURSION SCOPES**
>
> **Implementing recursion scopes is covered in the "Configure recursion" section.**

Configuring zone scopes

In Windows Server 2016, you can configure your DNS zones to have multiple zone scopes. Each zone scope contains its own set of DNS resource records. A resource record can exist in multiple zone scopes, each with different IP addresses depending on the scope. You can also use the zone scope to control zone transfers, enabling resource records from a zone scope in a primary zone to be transferred to the same zone scope in a secondary zone.

Typically, the first step in creating a zone scope is to create and configure client subnets. You do this in reference to your physical network topology. For example, to create subnet for DNS clients in New York and another for clients in London, use the following Windows PowerShell commands:

```
Add-DnsServerClientSubnet -Name "NYCSubnet" -IPv4Subnet "172.16.0.0/24"

Add-DnsServerClientSubnet -Name "LONSubnet" -IPv4Subnet "172.16.1.0/24"
```

Next, you create the DNS zone scopes for New York and London DNS clients by using the following commands:

```
Add-DnsServerZoneScope -ZoneName "Adatum.com" -Name "NYCZoneScope"

Add-DnsServerZoneScope -ZoneName "Adatum.com" -Name "LONZoneScope"
```

Configuring records in zone scopes

After you create the zone scopes, you must populate them with resource records. Again, you must do this with Windows PowerShell. To create a specific IP address record for clients in the New York City zone scope, run the following command:

```
Add-DnsServerResourceRecord -ZoneName "Adatum.com" -A -Name "www" -IPv4Address
"172.16.0.41" -ZoneScope "NYCZoneScope"
```

For clients in the London City zone scope, run the following command:

```
Add-DnsServerResourceRecord -ZoneName "Adatum.com" -A -Name "www" -IPv4Address
"172.16.1.22" -ZoneScope "LONZoneScope"
```

Configuring policies for zones

Finally, you must create the policies that instruct the DNS servers to respond to client queries based upon the previously defined factors. To configure the DNS servers to respond with resource records from the New York zone scope, create a DNS policy:

```
Add-DnsServerQueryResolutionPolicy -Name "NYCPolicy" -Action ALLOW -ClientSubnet
"eq,NYCSubnet" -ZoneScope "NYCZoneScope,1" -ZoneName "Adatum.com"
```

Similarly, for clients based in the London zone scope, you must add another policy:

```
Add-DnsServerQueryResolutionPolicy -Name "LONPolicy" -Action ALLOW -ClientSubnet
"eq,LONSubnet" -ZoneScope "LONZoneScope,1" -ZoneName "Adatum.com"
```

Now, if a client in the New York subnet petitions a DNS server for the IPv4 address of the www.adatum.com host, the DNS server responds with the IP address 172.16.0.41. If a client in London requests the same record, the client receives the IPv4 address 172.16.1.22.

> ***NOTE*** **DNS POLICIES**
>
> Implementing DNS policies is also covered in "Implement DNS policies."

> ***NEED MORE REVIEW?*** **DNS POLICIES OVERVIEW**
>
> For more information about configuring DNS policies, refer to the Microsoft TechNet website at *https://technet.microsoft.com/windows-server-docs/networking/dns/deploy/dns-policies-overview*.

Monitor DNS

Since DNS is such a critical service, providing name resolution and service location for configured clients, it is important that you ensure that DNS is running reliably and is optimized. To help you achieve this, you can use a number of monitoring and auditing tools in Windows Server.

Use DNS audit events and analytical events

Windows Server 2016 can collect a vast amount of logging data. Much of this logging is enabled by default, but features such as Debug logging (discussed in the "Configure DNS logging" section) must first be enabled before you can use them.

You can also use DNS Audit events and DNS Analytic events.

- **DNS Audit Events** These are enabled by default. Use to enable change tracking on your DNS servers. Every time a server, zone, or resource record is edited, an audit event is logged. Event IDs numbered 513 through 582 are logged in this regard, and are explained on the following website.

NEED MORE REVIEW? **AUDIT EVENTS**

For more information about audit events in Microsoft DNS, refer to the Microsoft TechNet website at *https://technet.microsoft.com/library/dn800669(v=ws.11).aspx#audit*.

- **DNS Analytic Events** These are disabled by default. Windows logs an analytic event every time the DNS server sends or receives DNS information. Event IDs numbered 257 through 280 are logged in this regard, and are explained on the website listed below.

NEED MORE REVIEW? **ANALYTIC EVENTS**

For more information about analytic events in Microsoft DNS, refer to the Microsoft TechNet website at *https://technet.microsoft.com/library/dn800669(v=ws.11).aspx#analytic*.

VIEWING AUDIT AND ANALYTICAL EVENTS

To view audit events, use the following procedure:

1. Open Event Viewer.
2. Expand Application And Services Logs, expand Microsoft, expand Windows, and then click DNS-Server.
3. Click the Audit folder, as shown in Figure 1-33. You can review events from here.

To view analytic events, use the following procedure:

1. Open Event Viewer.
2. Expand Application And Services Logs, expand Microsoft, expand Windows, and then click DNS-Server.
3. Right-click DNS-Server, click View, and then click Show Analytic And Debug Logs. The Analytical and Audit log folders display, as shown in Figure 1-33.
4. Right-click Analytical and then click Enable Log.
5. In the Event Viewer pop-up dialog box, click OK.

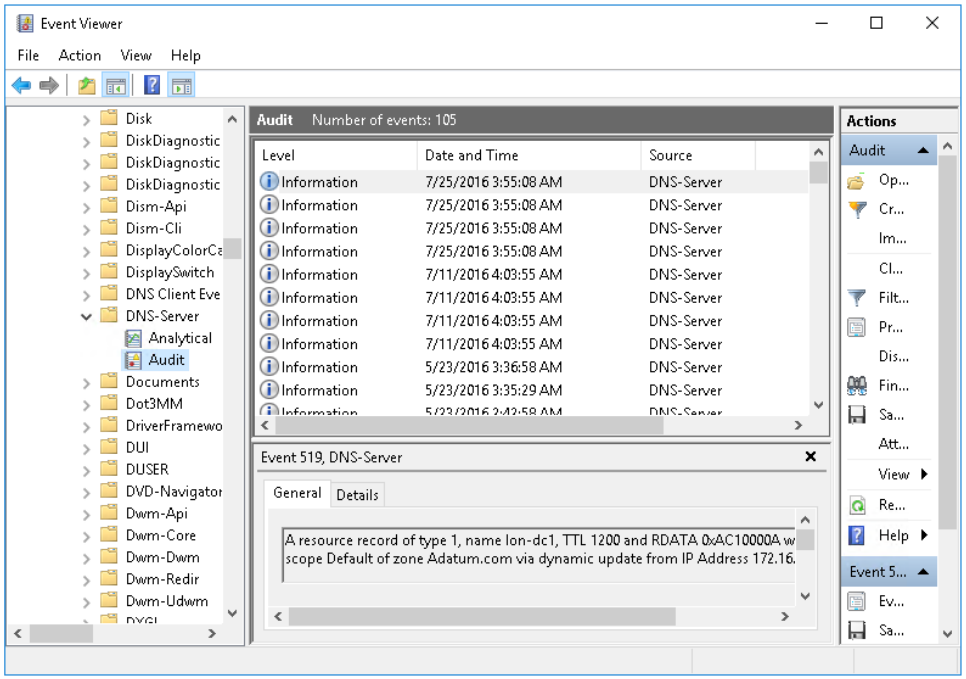

FIGURE 1-33 Viewing Audit events for a DNS server

NEED MORE REVIEW? **DNS LOGGING AND DIAGNOSTICS**

For more information about logging and diagnostics in Microsoft DNS, refer to the Microsoft TechNet website at *https://technet.microsoft.com/library/dn800669(v=ws.11).aspx*.

Analyze zone level statistics

Introduced in Windows Server 2012 R2, and improved in Windows Server 2016, DNS zone level statistics enable you to understand how a DNS server is being used for each authoritative zone on that server.

You can gather and view the following zone level statistics:

- **Zone queries** Provides the number of:
  - Queries received
  - Successful responses
  - Failed query responses
  - Name error responses

- **Zone transfers** Provides the number of zone transfer:
  - Requests received when operating as a primary server for a specific zone
  - Requests sent when operating as a secondary server for a specific zone
  - Requests received when operating as a secondary server for a specific zone
- **Zone transfers statistics** also provide the number of zone transfers
  - Received by the DNS Server service when operating as a secondary server for a specific zone
  - Sent by the DNS Server service when operating as a master server for a specific zone
- **Zone updates** Provides the total number of
  - Dynamic update requests received
  - Dynamic updates rejected

To access these statistics, use the Windows PowerShell Get-DnsServerStatistics cmdlet. For example:

```
Get-DnsServerStatistics -ZoneName "Adatum.com"
```

NEED MORE REVIEW? **DNS ZONE LEVEL STATISTICS**

For more information about analyzing DNS zone level statistics, refer to the Microsoft TechNet website at *https://blogs.technet.microsoft.com/networking/2013/10/04/dns-zone-level-statistics*.

Chapter summary

- You can install the DNS server role on Windows Server 2016 and Nano Server.
- DNS forwarders, recursion, and root hints enable you to control the flow of DNS query traffic throughout your organization's network.
- You can implement DNSSEC, DNS socket pool, cache locking, DANE, and response rate limiting to help to secure your DNS infrastructure from malicious attacks.
- DNS policies in Windows Server 2016 help you configure DNS behavior throughout your organization without needing to manually configure each DNS server.
- DNS logging can help you to pinpoint problems with the DNS servers in your organization before they can affect your users.
- The DNS server role is affected by CPU and memory resources, and proactive monitoring of these resources can be beneficial.
- Although the DNS namespace is based on domains and subdomains, the data that maintains this namespace is stored in DNS zones.
- Secondary zones receive updates via their configured master server.

- DNS delegation enables a part of your DNS namespace, such as a child domain, to be authoritatively maintained in a separate zone.
- AD DS-integrated zones provide for multimaster updates, secure replication, and secure dynamic updates.
- Conditional forwarding provides similar function to stub zones.
- DNS scopes are based on DNS policies.

Thought experiment

In this thought experiment, demonstrate your skills and knowledge of the topics covered in this chapter. You can find answers to this thought experiment in the next section.

You work in support at A. Datum Corporation. As a consultant for A. Datum, answer the following questions about installing and configuring DNS within the A. Datum organization:

1. You have asked a colleague to deploy the DNS server role to a Nano Server installed as a member of the Adatum.com domain. What must your colleague do?
2. At a branch office, you do not want the local DNS server to perform queries for local clients aside from those for which it is authoritative. How could you address this objective?
3. You want only to allow recursion by your DNS servers for queries received on the internal network and not from Internet-based clients. How could you address this requirement?
4. Managers at A. Datum are concerned with security and your boss has asked that you implement DNSSEC to help to secure DNS. You know that DNSSEC relies on distributing the NRPT. How could you configure NRPT distribution easily?
5. You have installed the DNS server role on a computer running Windows Server 2016. You now want to create zones on the server. You want to store the zone data in AD DS, but the option to store the zone in Active Directory is unavailable. Why might this be?
6. You want to be able to deploy an AD DS–integrated primary zone by using Windows PowerShell. What command should you use?
7. A. Datum has just purchased the Contoso Pharmaceuticals company. Your users are frequently accessing server resources in Contoso's network infrastructure. You need to configure DNS to support this change in circumstances. What two options do you have to more efficiently manage name resolution in this situation?
8. Your network consists of many subnets distributed across the globe. You want to make a web server easily accessible from any location by using the same name. However, you want your users to be directed by DNS to a local web server. What feature of Windows Server 2016 would enable this?

Thought experiment answers

This section contains the solutions to the thought experiment. Each answer explains why the answer choice is correct.

1. To install the DNS server role to an existing Nano Server, your colleague should create a remote Windows PowerShell session to the Nano Server and then use the Enable-WindowsOptionalFeature -Online -FeatureName DNS-Server-Full-Role command to add the DNS role.
2. You could configure the branch DNS server to use forwarding. Specify a DNS server elsewhere in the organization to which it forwards all queries it cannot satisfy locally.
3. You could implement DNS policies. Specifically, you could create a recursion scope so that recursion is enabled when requested on a specific DNS server interface, or from a specific internal subnet. The following three Windows PowerShell commands would enable you to achieve your objective:

```
Set-DnsServerRecursionScope -Name . -EnableRecursion $False

Add-DnsServerRecursionScope -Name "InternalAdatumClients" -EnableRecursion $True

Add-DnsServerQueryResolutionPolicy -Name "RecursionControlPolicy" -Action ALLOW
-ApplyOnRecursion -RecursionScope "InternalAdatumClients" -ServerInterfaceIP
"EQ,10.24.60.254"
```

4. The easiest way to distribute NRPT is to use a GPO. Edit the Default Domain GPO and navigate to Computer Configuration / Policies / Windows Settings / Name Resolution Policy. Create a rule containing the domain suffix you want to distribute for, and then enable both Enable DNSSEC in This Rule and Require DNS Clients to Check that the Name and Address Data Has Been Validated By the DNS Server.
5. The option to store the zone in Active Directory is only available on DNS servers that also have the AD DS server role installed and configured.
6. To deploy an AD DS–integrated primary zone on a DNS server, use the Add-DnsServerPrimaryZone cmdlet with the ReplicationScope parameter. For example: Add-DnsServerPrimaryZone -Name “Contoso.com” -ReplicationScope “Domain”
7. Consider implementing conditional forwarding or a stub zone. Both enable clients to more easily access the name servers for a foreign domain.
8. Use DNS policies and DNS zone scopes to configure this behavior. You can create DNS client subnets and assign these subnets into DNS scopes. Next, you create DNS resource records in the zone scopes. Finally, you would use a DNS policy to determine which records are returned to a DNS client, based on the originating subnet.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z