Microsoft

MCTS EXAM

70-667

Configuring Microsoft® SharePoint® 2010

Dan Holme
Alistair Matthews

SELF-PACED

Training Kit

How to access your CD files

The print edition of this book includes a CD. To access the CD files, go to http://aka.ms/638853/files, and look for the Downloads tab.

Note: Use a desktop web browser, as files may not be accessible from all ereader devices.

Questions? Please contact: mspinput@microsoft.com

Microsoft Press

Exam 70-667: Pro: Configuring Microsoft SharePoint 2010

OBJECTIVE	LOCATION IN BOOK
INSTALLING AND CONFIGURING A SHAREPOINT ENVIRONMENT	
Deploy new installations and upgrades.	Chapter 1, Lessons 1, 2, and 3 Chapter 2, Lesson 2 Chapter 9, Lessons 1 and 2
Configure SharePoint farms.	Chapter 1, Lessons 2 and 3 Chapter 2, Lesson 2 Chapter 3, Lessons 1, 2 and 3 Chapter 8, Lesson 2 Chapter 9, Lessons 1 and 2 Chapter 11, Lesson 1
Configure service applications.	Chapter 5, Lessons 1 and 2 Chapter 6, Lessons 1 and 2 Chapter 8, Lessons 1, 2, 3, 4, 5, and 6 Chapter 9, Lessons 1 and 2
Configure indexing and search.	Chapter 7, Lessons 1, 2 and 3
MANAGING A SHAREPOINT ENVIRONMENT	
Manage operational settings.	Chapter 4, Lesson 1 Chapter 12, Lessons 1, 2, and 3
Manage accounts and user roles.	Chapter 1, Lessons 2 and 3 Chapter 2, Lessons 1 and 2 Chapter 4, Lessons 1 and 2 Chapter 9, Lesson 2
Manage authentication providers.	Chapter 3, Lesson 2
DEPLOYING AND MANAGING APPLICATIONS	
Manage Web Applications.	Chapter 1, Lesson 3 Chapter 2, Lesson 2 Chapter 3, Lesson 1, 2 and 3 Chapter 4, Lessons 1 and 2
Manage site collections.	Chapter 1, Lesson 3 Chapter 2, Lesson 2 Chapter 4, Lessons 1 and 2 Chapter 5, Lessons 1 and 2 Chapter 9, Lessons 1 and 2 Chapter 10, Lessons 1, 2 and 3 Chapter 12, Lesson 4
Deploy and manage SharePoint solutions.	Chapter 10, Lessons 2 and 3
MAINTAINING A SHAREPOINT ENVIRONMENT	
Back up and restore a SharePoint environment.	Chapter 11, Lesson 2
Monitor and analyze a SharePoint environment.	Chapter 12, Lessons 1, 2 and 3
Optimize the performance of a SharePoint environment.	Chapter 9, Lesson 2 Chapter 12, Lesson 4

Exam Objectives The exam objectives listed here are current as of this book's publication date. Exam objectives are subject to change at any time without prior notice and at Microsoft's sole discretion. Please visit the Microsoft Learning Web site for the most current listing of exam objectives: *http://www.microsoft.com/learning/en/us/Exam.aspx?ID=70-667#tab2*.

MCTS Self-Paced Training Kit (Exam 70-667): Configuring Microsoft SharePoint 2010

Dan Holme
Alistair Matthews

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Control Number: 2011934666
ISBN: 978-0-7356-3885-3

Printed and bound in the United States of America.

Second Printing

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at mspinput@microsoft.com. Please tell us what you think of this book at http://www.microsoft.com/learning/booksurvey.

Microsoft and the trademarks listed at http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions Editor: Jeff Koch
Developmental Editor: Karen Szall
Project Editor: Karen Szall
Editorial Production: Christian Holdener, S4Carlisle Publishing Services
Technical Reviewer: Bob Hogan; Technical Review services provided by Content Master, a member of CM Group, Ltd.
Copyeditor: Becka McKay
Indexer: Maureen Johnson
Cover: Twist Creative • Seattle

[2012-04-20]

Contents at a Glance

Contents

Chapter 7 Administering SharePoint Search 393

Introduction

This training kit is designed for information technology (IT) professionals who support or plan to support SharePoint Server 2010 and who also plan to take the Microsoft Certified Technology Specialist (MCTS) exam 70-667, *TS: Microsoft SharePoint 2010, Configuring*.

The material covered in this training kit and on exam 70-667 relates to SharePoint products and technologies, which enable business collaboration in an enterprise and on the web. It is assumed that before you begin using this training kit, you have a solid, foundation-level understanding of Microsoft Windows client and server operating systems and common Internet technologies. The MCTS exam and this book assume that you have at least one year of experience configuring SharePoint and related technologies, including Internet Information Services (IIS), Windows Server 2008, Active Directory, DNS, SQL Server, and networking infrastructure services.

The topics in this training kit cover what you need to know for the exam, as described on the Skills Measured tab for the exam, which is available at *http://www.microsoft.com/learning/en/us/exam.aspx?ID=70-667&locale=en-us#tab2*.

By using this training kit, you will learn how to do the following:

- Deploy SharePoint Server 2010 farms.
- Create a logical architecture of web applications, content databases, site collections, and sites.
- Manage security of SharePoint content by configuring authentication and access controls.
- Configure SharePoint services including search, user profiles, and the managed metadata service.
- Optimize, monitor, and troubleshoot performance of SharePoint servers and services.
- Ensure that data is protected and highly available.
- Deploy and manage customized SharePoint functionality and solutions.

Refer to the Objective map in the front of this book to see where in the book each exam objective is covered.

System Requirements

Practice exercises are a valuable component of this training kit. They allow you to experience important skills directly, reinforce material discussed in lessons, and even introduce new concepts.

Each lesson and practice describes the requirements for exercises. Many lessons require only two computers, one configured as a domain controller for a sample domain named contoso.com and the second configured as a SharePoint server running Microsoft SQL Server 2008 R2 and SharePoint Server 2010. However, some lessons require additional computers acting as a second server in the SharePoint farm.

The companion media includes the “Lab Environment Build Guide” document, which contains detailed setup instructions for the computers used throughout this training kit. Lessons that require additional computers provide guidance regarding the configuration of those computers.

Hardware Requirements

You can perform exercises on physical computers. Each computer must meet the minimum requirements for RAM, free hard disk space, and processor cores shown here:

- **Domain Controller** 1.5 GB RAM, 40 GB free disk space, and at least 1 processor core.
- **SharePoint server** 6 GB RAM, 128 GB free disk space, and at least 2 processor cores.
- **Additional SharePoint server** 4 GB RAM, 128 GB free disk space, and at least 2 processor cores.

To minimize the time and expense of configuring the computers required for this training kit, it’s recommended that you perform the practices in this training kit on virtual machines. The training kit assumes you will use virtualization software that supports snapshots, so that you can roll back to a previous state after performing an exercise.

You can create virtual machines by using Hyper-V—a feature of Windows Server 2008 and Windows Server 2008 R2—or other virtualization software, such as VMware Workstation. The Lab Environment Build Guide details the configuration of the virtual machines required for this training kit. Refer to the documentation of your selected virtualization platform for hardware and software requirements, for instructions regarding host setup and configuration.

If you choose to use virtualization software, you can run more than one virtual machine on a host computer. The host computer must have sufficient RAM for each virtual machine that you will run simultaneously on the host, plus sufficient RAM to meet the RAM requirements of the host operating system.

If you plan to run all virtual machines on a single host, the host must have at least 12 GB of RAM. For example, one of the most complex configurations you will need is one domain controller using 512 MB of RAM, and two SharePoint servers using 6 GB and 4 GB of RAM. On a host computer with 12 GB of RAM, this would leave just over 1 GB for the host.

The host computer must have sufficient disk space for each virtual machine plus snapshots. We recommend that you have at least 512 GB of free disk space if you want to run all virtual machines on a single host computer. Note that you never use more than three virtual machines together at the same time.

If you encounter performance bottlenecks while running multiple virtual machines on a single physical host, consider running virtual machines on more than one physical host.

Ensure that all machines—virtual or physical—that you use for exercises can network with each other. It is highly recommended that the environment be totally disconnected from your production environment. Refer to the documentation of your virtualization platform for network configuration procedures.

We recommend that you preserve each of the virtual machines you create until you have completed the training kit. After each chapter, create a snapshot of the virtual machines used in that chapter so that you can reuse them, as required in later exercises.

Finally, you must have a physical computer with a CD-ROM drive with which to read the companion media. (If you have the eBook, you can retrieve the companion media from the book's web page.) You must also have Internet connectivity so that you can download the evaluation versions of software, as specified in the "Lab Environment Build Guide."

Software Requirements

The following software is required to complete the practice exercises:

- Windows Server 2008 R2
- SQL Server 2008 R2 (64-bit)
- SharePoint Server 2010 (Enterprise Client Access License features)
- SharePoint Designer 2010
- Office Professional Plus 2010
- Silverlight

You can download evaluation versions of the products from the TechNet Evaluation Center at *http://technet.microsoft.com/en-us/evalcenter*. If you use evaluation versions of the software, pay attention to the expiration date of the product. The evaluation version of Windows Server 2008 R2, for example, can be used for up to 60 days.

If you have a TechNet or MSDN subscription, you can download the products from the subscriber downloads center. These versions do not expire. If you are not a TechNet or MSDN subscriber, it is recommended that you subscribe so that you can access benefits such as product downloads.

To configure the computers and to access files on the companion media, the following software is required:

- If you are not using virtualization software, you need software that allows you to handle .iso files. This software needs to perform either of the following functions:
  - Burn .iso files to CDs or DVDs. (This solution also requires CD/DVD recording hardware.)
  - Mount .iso files as virtual CD or DVD drives on your computer.
- A web browser such as Internet Explorer version 8 or later.
- An application that can display PDF files, such as Adobe Acrobat Reader, which can be downloaded at *http://www.adobe.com/reader*.

IMPORTANT **LAB ENVIRONMENT BUILD GUIDE**

Be sure to read the "Lab Environment Build Guide" on the companion media for detailed instructions regarding the setup of computers for this training kit.

Using the Companion CD

A companion CD is included with this training kit. The companion CD contains the following:

- **Practice tests** You can reinforce your understanding of the topics covered in this training kit by using electronic practice tests that you customize to meet your needs. You can run a practice test that is generated from the pool of Lesson Review questions in this book. Alternatively, you can practice for the 70-667 certification exam by using tests created from a pool of more than 200 practice exam questions, which give you many practice exams to ensure that you are prepared.
- **Practice files** Some practices in this training kit refer to files in the Practice Files folder on the companion media. When you prepare for practices by following the instructions in the Lab Environment Build Guide, these files are copied to the C:\70667TK folder on the disk drive of the SharePoint server, so that during the practices you can access the files without the companion media.
- **An eBook** An electronic version of this book is included for when you do not want to carry the printed book with you.
- **Practice answers** At the end of each lesson, one or more hands-on practice exercises challenge you to apply the concepts and skills discussed in the lesson to real-world scenarios. Each exercise presents high-level instructions, similar to what you might receive from a manager, colleague, or end user in an enterprise environment. We recommend that you try to complete the exercise by recalling and reviewing what

you've learned in the lesson. If you cannot complete a step or exercise, you can use the practice answers on the companion CD, which include detailed, step-by-step instructions for each exercise.

NOTE **COMPANION CONTENT FOR DIGITAL BOOK READERS**

If you bought a digital-only edition of this book, you can enjoy select content from the print edition's companion CD. Visit http://www.microsoftpressstore.com/title/9780735638853 to get your downloadable content.

How to Install the Practice Tests

To install the practice test software from the companion CD to your hard disk, perform the following steps:

1. Insert the companion CD into your CD drive and accept the license agreement. A CD menu appears.

   NOTE **IF THE CD MENU DOES NOT APPEAR**

   If the CD menu or the license agreement does not appear, AutoRun might be disabled on your computer. Refer to the Readme.txt file on the CD for alternate installation instructions.

2. Click Practice Tests and follow the instructions on the screen.

How to Use the Practice Tests

To start the practice test software, follow these steps:

1. Click Start, All Programs, and then select Microsoft Press Training Kit Exam Prep.

   A window appears that shows all the Microsoft Press training kit exam prep suites installed on your computer.

2. Double-click the lesson review or practice test you want to use.

   NOTE **LESSON REVIEWS VS. PRACTICE TESTS**

   Select the (70-667) *TS: Microsoft SharePoint 2010, Configuring* lesson review to use the questions from the "Lesson Review" sections of this book. Select the (70-667) *TS: Microsoft SharePoint 2010, Configuring* practice test to use a pool of 200 questions similar to those that appear on the 70-667 certification exam.

Lesson Review Options

When you start a lesson review, the Custom Mode dialog box appears so that you can configure your test. You can click OK to accept the defaults, or you can customize the number of questions you want, how the practice test software works, which exam objectives you want the questions to relate to, and whether you want your lesson review to be timed. If you are retaking a test, you can select whether you want to see all the questions again or only the questions you missed or did not answer.

After you click OK, your lesson review starts.

- To take the test, answer the questions and use the Next and Previous buttons to move from question to question.
- After you answer an individual question, if you want to see which answers are correct—along with an explanation of each correct answer—click Explanation.
- If you prefer to wait until the end of the test to see how you did, answer all the questions and then click Score Test. You will see a summary of the exam objectives you chose and the percentage of questions you got right overall and per objective. You can print a copy of your test, review your answers, or retake the test.

Practice Test Options

When you start a practice test, you choose whether to take the test in Certification Mode, Study Mode, or Custom Mode:

- **Certification Mode** Closely resembles the experience of taking a certification exam. The test has a set number of questions. It is timed, and you cannot pause and restart the timer.
- **Study Mode** Creates an untimed test during which you can review the correct answers and the explanations after you answer each question.
- **Custom Mode** Gives you full control over the test options so that you can customize them as you like.

In all modes, the user interface when you are taking the test is basically the same but with different options enabled or disabled depending on the mode. The main options are discussed in the previous section, "Lesson Review Options."

When you review your answer to an individual practice test question, a "References" section is provided that lists where in the training kit you can find the information that relates to that question and provides links to other sources of information. After you click Test Results to score your entire practice test, you can click the Learning Plan tab to see a list of references for every objective.

How to Uninstall the Practice Tests

To uninstall the practice test software for a training kit, use the Program And Features option in Windows Control Panel.

Acknowledgments

Although the authors' names appear on the cover of this book, we are but one part of the incredible team that has brought this—the first training kit for SharePoint administration published by Microsoft Press—to fruition. Our technical reviewer is Bob Hogan, and the copy editor is Becka McKay. Both of them went well beyond the call of duty, and their attention to detail and to accuracy added tremendous value to this work. Christian Holdener is our project manager. He coordinated the many reviews and, more important, kept the production schedule moving despite the challenges we threw at him. Most important is the astounding Karen Szall, our editor *extraordinaire*, with whom I've worked on many Microsoft Press titles. She has earned herself a place in editorial heaven with this one. We the authors are deeply grateful for the efforts of this talented group of colleagues. Dan also extends a big *mahalo* to Wyatt, Keith, Maddie, Jack, and the team at AvePoint for their support and soul-nourishment over the course of this project.

Support & Feedback

The following sections provide information on errata, book support, feedback, and contact information.

Errata

We've made every effort to ensure the accuracy of this book and its companion content. Any errors that have been reported since this book was published are listed on our Microsoft Press site at:

http://www.microsoftpressstore.com/title/9780735638853

If you find an error that is not already listed, you can report it to us through 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 is not offered through the addresses above.

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:

http://www.microsoft.com/learning/booksurvey

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in Touch

Let us keep the conversation going! We are on Twitter: *http://twitter.com/MicrosoftPress*.

Preparing for the Exam

Microsoft certification exams are a great way to build your resume and let the world know about your level of expertise. Certification exams validate your on-the-job experience and product knowledge. While there is no substitution for on-the-job experience, preparation through study and hands-on practice can help you prepare for the exam. We recommend that you round out your exam preparation plan by using a combination of available study materials and courses. For example, you might use the Training Kit and another study guide for your "at home" preparation, and take a Microsoft Official Curriculum course for the classroom experience. Choose the combination that you think works best for you.

Microsoft®
CERTIFIED
Technology
Specialist

CHAPTER 3

Managing Web Applications

Web applications are the top component of the logical hierarchy of SharePoint content within a farm. All user access to SharePoint content is performed within the context of a web application. Although content itself is contained within site collections and stored in content databases, web applications and their associated IIS Web sites manage important functions, including authentication and SSL encryption. Web applications also scope configuration, including important settings that enforce consistent security across all site collections in the web application. In this chapter, you will learn the procedures and settings related to web applications, and you will master important concepts including Claims Based Authentication, access mappings, and zones. In Lesson 1, you will explore, in detail, the numerous settings that you can configure when you create a web application. Lesson 2 is dedicated to managing authentication. In Lesson 3, you will learn how to configure access to web applications in more complex scenarios, in which users access web applications via more than one URL.

Exam objectives in this chapter:

- Configure SharePoint farms.
- Manage web applications.
- Manage authentication providers.

Lessons in this chapter:

Before You Begin

To complete the lessons in this chapter, you must have done the following:

- Performed the practices in Chapter 1.

REAL WORLD

Dan Holme

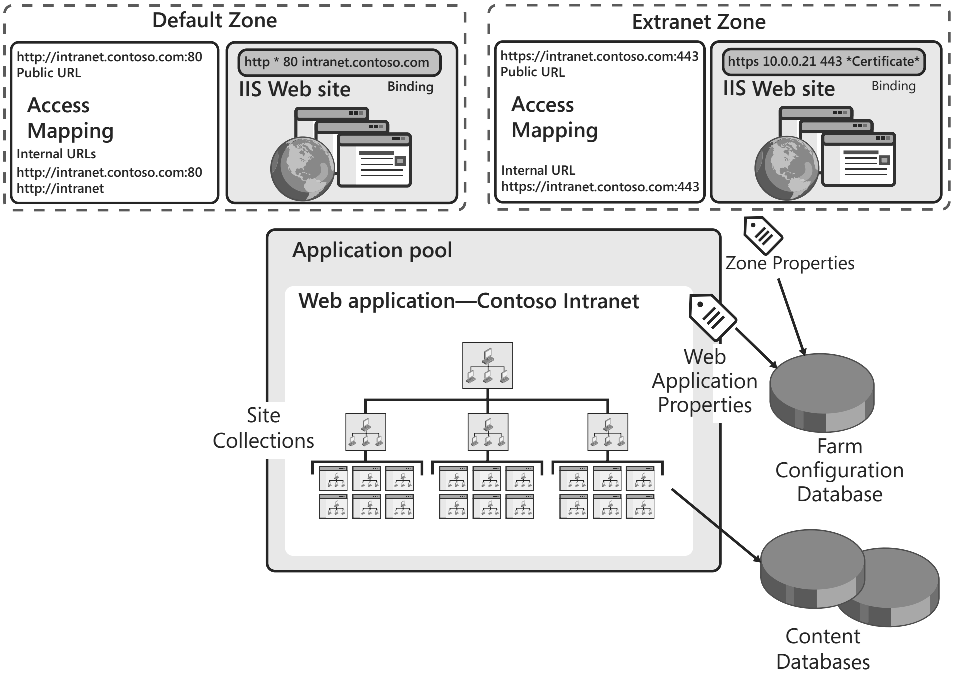
A web application is a SharePoint component that is closely related to and dependent on—but separate from—an IIS Web site. Thrown in the mix are access mappings and zones, each of which relate to the URLs with which a web application is accessed. The tangled relationship between these components is not seamless to manage. Vague terminology and documentation make it easy for even a seasoned SharePoint veteran to make configuration mistakes that prevent access to a website. I've worked hard in this chapter to clarify concepts and procedures that are quite confusing for many SharePoint administrators, and I've centered the discussion of these concepts and procedures around the real-world scenarios you will face as you manage SharePoint web applications in your enterprise.

Lesson 1: Configure Web Applications

Web applications are at the top of the logical hierarchy of a SharePoint farm. A SharePoint farm will typically have at least two web applications: Central Administration and a web application that contains content accessed by users, such as *http://intranet.contoso.com*.

A web application is composed of a collection of settings stored in the farm's configuration database, one or more content databases, one or more site collections containing content, and one to five zones—each supported by Internet Information Services (IIS) Web sites running with a single application pool. All of these are accessible using URLs called *access mappings* that enable a request to reach the correct SharePoint web application and enable content to be rendered properly to a user.

A diagram of the components related to a SharePoint web application is shown in Figure 3-1.



Default Zone
http://intranet.contoso.com:80
Public URL
Access Mapping
Internal URLs
http://intranet.contoso.com:80
http://intranet
http * 80 intranet.contoso.com
IIS Web site
Binding
Extranet Zone
https://intranet.contoso.com:443
Public URL
Access Mapping
Internal URL
https://intranet.contoso.com:443
https 10.0.0.21 443 *Certificate*
IIS Web site
Binding
Application pool
Web application—Contoso Intranet
Site Collections
Zone Properties
Web Application Properties
Farm Configuration Database
Content Databases

FIGURE 3-1 SharePoint web application components

In this lesson, you will learn to create a web application and to configure many web application settings. In Lesson 2, you will learn to configure authentication. And, in Lesson 3, you will learn to manage access mappings and zones.

After this lesson, you will be able to:

- Create a web application.
- Configure web application settings, including Recycle Bin and self-service site collection creation.
- Describe the relationship between a SharePoint web application and an IIS Web site.
- Configure a web application to use SSL.

Estimated lesson time: 60 minutes

Understand Web Applications and IIS Web Sites

When you create a web application, you create a site in Internet Information Services (IIS). An IIS Web site has *bindings*, which can include a unique IP address, a host header, or a port. Bindings enable IIS to determine which site is being requested by an inbound HTTP request.

When a user requests a page with a unique resource indicator (URI), such as *http://intranet.contoso.com/SitePages/Home.aspx*, the client creates an HTTP request packet. The client determines the IP address by querying DNS to resolve the host name—in this example, *intranet.contoso.com*. The client adds the host name to the host header field of the packet. The client then sends the packet to the server's IP address over the default HTTP port (80) unless otherwise specified. If the request is an HTTPS request, the destination port is 443 unless otherwise specified. IIS receives and parses the request and identifies that the request is for the *Contoso Intranet* IIS Web site, which is bound to port 80 with the host header *intranet.contoso.com*.

After IIS has determined the Web site from which content is being requested, it begins a series of processes that return the requested content to the client. IIS first looks into the web application's physical path, also called the *root directory*. When you create a SharePoint web application, you specify the physical path. For example, the intranet web application is stored at C:\inetpub\wwwroot\wss\VirtualDirectories\clients.contoso.com80. In traditional ASP.NET websites, actual content is stored in the root directory. However, as you learned in previous chapters, SharePoint stores content in content databases on a SQL server. IIS has no idea how to locate and access this content. However, the configuration file, Web.config, in the root directory of the IIS Web site defines the site as a SharePoint application and instructs IIS to pass the request to SharePoint.

SharePoint then parses the URI to determine which site collection and thereby which content database is being requested. SharePoint can then access the content from SQL and return the content to IIS, which then delivers the content to the user.

The request is handled by IIS and SharePoint within the context of the application pool of the IIS Web site. The application pool is an isolated memory space that is routed to one or more worker processes (w3wp.exe) that handle requests sent to a server for the sites associated with the application pool. The application pool identity is a domain user account that is registered as a managed account in SharePoint.

When you create a web application, SharePoint creates a content database and assigns the application pool identity the permissions it requires to access content. SharePoint also creates the physical path, the Web.config file, the IIS Web site, and several virtual directories.

As you will learn later in this lesson, SharePoint does not manage the configuration of a security certificate. Although SharePoint specifies that a web application uses SSL, you must install a certificate and bind the certificate to the site in IIS Manager on each server in the farm. Additionally, SharePoint does not give you the option of binding a web application to a specific IP address. If you want to bind a web application to one or more specific IP addresses, you must do so manually in IIS Manager on each server in the farm.

> ***TIP*** **MANAGE WEB APPLICATIONS BY USING CENTRAL ADMINISTRATION OR WINDOWS POWERSHELL**
>
> **Do not create or change settings directly on sites by using IIS Manager, except when assigning an IP address binding to a site or binding a certificate to a site that uses SSL. When you manage configuration with SharePoint, the settings are stored in the configuration database, and are applied to each new server that you add to the farm. If you change settings in IIS Manager, you must make the same changes on each server in the farm.**

Design Considerations: One or More Web Applications

More web applications are necessary if your governance requirements must be implemented using configuration that is scoped to web applications. For example, if your governance plan requires isolation of internal and external content from cross-site scripting attacks, you can divide content into separate web applications, each of which will have a distinct IIS Web site with a unique domain name, such as *http://intranet.contoso.com* and *http://clients.contoso.com*. Doing so will also physically isolate content in separate content databases that can optionally be hosted on separate SQL servers with distinct SQL authentication configuration.

Later in this chapter, you will explore the numerous settings that are scoped to a web application. Although the importance of each setting will vary between enterprises, among the most important settings scoped to a web application are the following:

- **Service application connections** Each web application is connected to the services it requires, such as Search. Not all scenarios require all services. For example, a small team site might not require the PerformancePoint Service Application. For performance and governance reasons, you should connect web applications only to the service applications they require.
- **Recycle Bin settings** You can configure whether deleted items are moved to SharePoint's two-stage Recycle Bins, how long items are retained before being permanently deleted, and what storage limits will be.

- **Self-service site collection creation** You can allow users to create site collections, which reduces administrative overhead.
- **Blocked file types** You can configure the file types that are allowed to be uploaded within a web application.

If every scenario that you will support with SharePoint can be implemented with a single set of configuration—for example, if every scenario requires the same services, the same Recycle Bin settings, and the same settings for self-service site collection creation and blocked file types—you need only one web application. However, it is likely that some scenarios will require different sets of configuration. For example, all scenarios might require the same Recycle Bin settings as dictated by your governance plan, but you might want to block audio and video file types on the intranet web application while allowing them within a team site. To support two different collections of settings that are scoped to web applications, you must create two web applications.

In this chapter, and throughout this Training Kit, you will learn about settings that apply to a web application, to a site collection, to a content database, to zones (which you will learn about in Lesson 3), or to other components of the SharePoint logical structure. Pay attention to the scope of settings, because you must consider them when you translate your information management requirements into a SharePoint logical architecture for your enterprise.

Design Considerations: One or More Application Pools and Identities

Because the application pool is a process (w3wp.exe) and an isolated memory space, two web applications running in two separate application pools are isolated from each other. If one web application contains poorly written code that causes the site to crash or consume too many resources, the other application pool and web application will continue to function. And if security is compromised in one application pool, other application pools are not exposed.

You might think that you should create a new application pool for each new web application. In fact, the default settings on the Create New Web Application page encourage you to do just that. However, application pools are a limited resource—each running application pool requires a memory footprint and places performance overhead on the server. Microsoft supports up to 10 application pools per web server; however, the limit depends largely upon the amount of RAM allocated to front-end servers and the workload that the farm is serving: the user base and its usage characteristics. It is therefore recommended that you create web applications in a shared application pool unless there is a significant requirement for process isolation.

Each application pool has an identity—a domain user account that is used by the application pool process. When you create a new application pool, you can use an identity that is shared with other application pools, or you can assign an identity unique to the application pool. The decision of whether to use a shared or unique application pool identity depends on the level of security your governance plan requires. The two primary considerations are access and auditing.

If an application pool identity is breached, it is possible that other application pools running with the same identity might also be exposed. Therefore, if the content accessed by web applications in two application pools shares similar security profiles, you might choose to use a single identity for both application pools. If the security profiles are different, you might choose to use a separate identity for each. For example, Central Administration should run in a separate application pool because the identity used by Central Administration is the SharePoint farm service account—for example, SP_Farm—which is highly privileged. That level of privilege is not appropriate for user-facing web applications. The application pool that is used for user-facing web applications should be a different identity.

If you use unique identities for each application pool, it can be easier to review event log or audit entries and, by examining the identity reported in an event, to distinguish the specific application pool that generated an event.

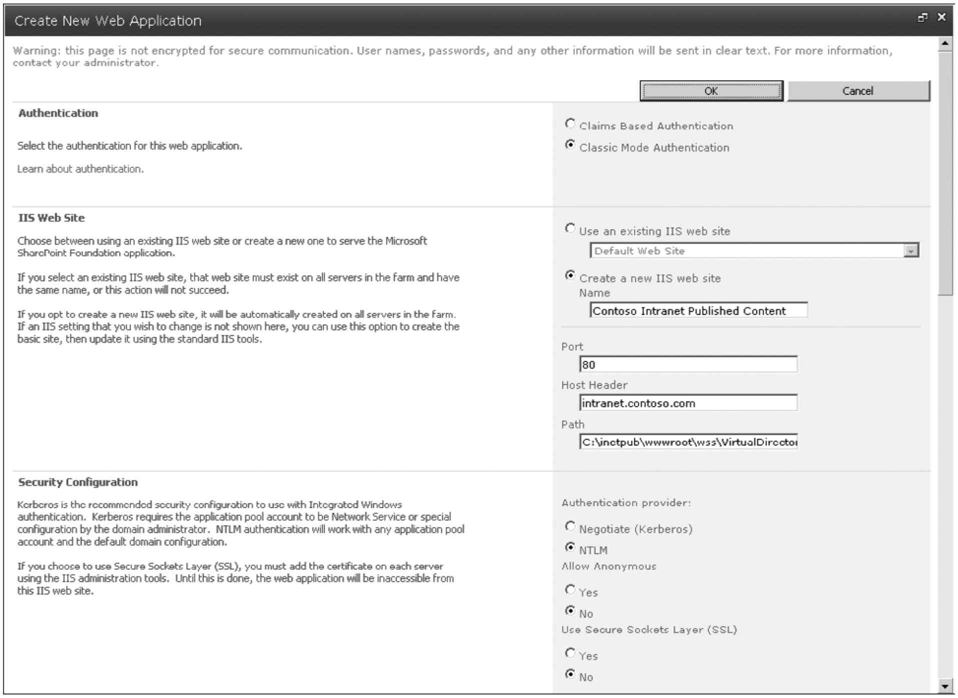
As you learned in Chapter 1, "Creating a SharePoint 2010 Intranet," SharePoint 2010's managed accounts feature makes it much easier to manage password changes for accounts used by services and application pools. Therefore, it is suggested that when you create a new application pool, you consider using a unique identity; however, this is not required.
In the Practice for this lesson, you will create an extranet web application for Contoso, Ltd. Because the extranet website is exposed to the Internet, you will create the web application in a separate application pool. Because the security profile of the extranet application is similar to other user-facing web applications, but is quite different than Central Administration, you will assign the new application pool the same identity as other user-facing application pools: CONTOSO\SP_WebApps. In Chapter 9, "Deploying and Upgrading to SharePoint 2010," you will learn how to configure managed accounts and how to assign and change application pool identities.

By default, service applications such as Search and Managed Metadata share a single application pool. Although this application pool is separate from the application pool shared by user-facing applications, it is recommended that all service applications share an application pool unless, again, there is a significant driver for process isolation.

Configure a New Web Application

In Chapter 1, you learned to create a web application by using Central Administration. In Chapter 2, you learned to use the *New-SPWebApplication* cmdlet to create a web application. In Chapter 9, you will learn to use the *New-SPManagedAccount* cmdlet to create a managed account, which you can then use as the application pool identity for the *–ApplicationPoolAccount* parameter of the *New-SPWebApplication* cmdlet.

You have learned that a web application is a combination of an IIS Web site and a content database. The configuration for each is stored in the configuration database of the farm. In this section, you will explore the most important settings that can be configured when you create a web application using the Create New Web Application page in Central Administration, shown in Figure 3-2.



Create New Web Application

Warning: this page is not encrypted for secure communication. User names, passwords, and any other information will be sent in clear text. For more information, contact your administrator.

Authentication

Select the authentication for this web application.

Learn about authentication.

○ Claims Based Authentication
● Classic Mode Authentication

IIS Web Site

Choose between using an existing IIS web site or create a new one to serve the Microsoft SharePoint Foundation application.

If you select an existing IIS web site, that web site must exist on all servers in the farm and have the same name, or this action will not succeed.

If you opt to create a new IIS web site, it will be automatically created on all servers in the farm. If an IIS setting that you wish to change is not shown here, you can use this option to create the basic site, then update it using the standard IIS tools.

○ Use an existing IIS web site
Default Web Site

● Create a new IIS web site
Name
Contoso Intranet Published Content

Port
80

Host Header
intranet.contoso.com

Path
C:\inetpub\wwwroot\wss\VirtualDirecto

Security Configuration

Kerberos is the recommended security configuration to use with Integrated Windows authentication. Kerberos requires the application pool account to be Network Service or special configuration by the domain administrator. NTLM authentication will work with any application pool account and the default domain configuration.

If you choose to use Secure Sockets Layer (SSL), you must add the certificate on each server using the IIS administration tools. Until this is done, the web application will be inaccessible from this IIS web site.

Authentication provider:
○ Negotiate (Kerberos)
● NTLM

Allow Anonymous
○ Yes
● No

Use Secure Sockets Layer (SSL)
○ Yes
● No

FIGURE 3-2 The Create New Web Application page

Authentication

The first setting that appears on the Create New Web Application page is Authentication. SharePoint Server 2010 offers two types of authentication:

- **Classic Mode Authentication** Classic Mode Authentication is the same type of authentication that was used in Microsoft Office SharePoint Server 2007. Classic Mode Authentication relies on Active Directory to authenticate users.
- **Claims Based Authentication** Claims Based Authentication is a new feature in SharePoint 2010. With Claims Based Authentication you can use Windows authentication (Active Directory); Forms Based Authentication (FBA) against an authentication provider such as Active Directory Lightweight Directory Services (AD LDS), a SQL database of users, or an ASP.NET membership provider; or Security Assertion Markup Language (SAML) tokens generated by trusted authorities such as Windows Live ID or Active Directory Federated Services 2.0 (ADFS 2.0).

If you are new to SharePoint, you should select Claims Based Authentication only if you need to implement Forms Based Authentication or SAML token-based authentication, or if the web application will use code that uses claims. Otherwise, you should select Classic Mode Authentication. Some SharePoint features, including audiences, become more difficult to implement in a web application that uses Claims Based Authentication.

You will learn about each provider and method in Lesson 2. When you understand the various authentication providers, and the nuances of configuring and managing authentication, you can make a more informed decision about whether to select Classic Mode Authentication or Claims Based Authentication when you require only Windows authentication. As you will learn in Lesson 2, you can change the authentication mode from Classic Mode Authentication to Claims Based Authentication by using Windows PowerShell.

IIS Web Site

As you've learned, when you create a SharePoint web application, you also create a corresponding site in IIS. In the Name box, type a name for the web application. This name will appear in Central Administration as the name of the web application and in IIS as the site name. Follow your organization's naming standards, which should be designed to ensure that an administrator can easily identify the purpose of a SharePoint web application or IIS Web site.

Next, in the Port box, configure the port number to which the site will be bound. By default, the Create New Web Application page specifies a random port number, but the port is usually 80 for HTTP or 443 for HTTPS.

When more than one web application is bound to a single IP address and port, a host header is required to allow IIS to route an inbound request to the correct site. Earlier in this lesson you learned that a client embeds the host name portion of the URL in the host header field of the HTTP request packet.

In the Host Header box, type the host header for the web application, which should be the fully qualified domain name (FQDN) of the web application.

Finally, you can configure the root directory by changing the default value in the Path box. The content stored in the root directory of a SharePoint site is minimal, because most content is stored in the content database(s) of the web application. Therefore, you have little reason to change the root directory, unless your governance policies require you to do so. If you do change the path, ensure that the drive letter exists on every SharePoint server in the farm. Also, verify that NTFS permissions allow the root directory to be created successfully on each server in the farm.

The settings discussed in this section are required when you create a new IIS Web site. However, you can also select the Use An Existing IIS Web Site option, and the web application will read the site configuration from IIS on the server running Central Administration. This option is rarely used. Its primary purpose is to fix a broken web application by re-creating the web application and connecting it to the previously created IIS Web site.

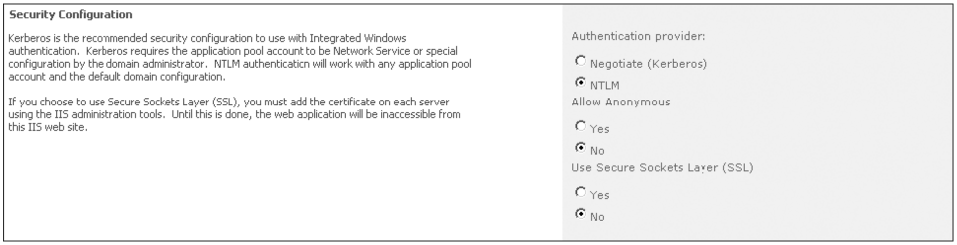
Note the following guidelines related to the creation of the IIS Web site for a SharePoint web application:

- It is not recommended to use flat host names without a domain name component as the host header. In other words, do not configure a host header of *http://intranet*. Instead, use *http://intranet.contoso.com.*

- When you create a new SharePoint web application, you cannot specify a unique IP address for the new IIS Web site within SharePoint. After creating the IIS Web site, you must use IIS Manager to modify the bindings of the IIS Web site so that the site is bound to an IP address. You must repeat this process on each server in the farm. This configuration is not recorded by SharePoint, and therefore it is not backed up by SharePoint. If you restore a web application by using SharePoint, you must manually reconfigure the IP address binding. Bindings are backed up when you back up IIS configuration.
- Enter the host header correctly entered when you create a web application. The host header is recorded in the configuration database and cannot be changed after the web application is created. You can change host header bindings directly in IIS, but you must remember to do so each time you add a new server to the farm, and to update existing servers if you restart the Microsoft SharePoint Foundation Web Application service. If you need to change the host header of the web application, it is recommended that you delete and re-create the web application.
- Only one host header can be defined during the creation of a web application creation. If users will access the web application with more than one host name, such as *http://intranet.contoso.com* and *http://portal.contoso.com*, you must extend the web application to create additional zones. You will learn more about zones in Lesson 3.

Security Configuration

The options that appear in the Security Configuration section depend on whether you selected Claims Based Authentication or Classic Mode Authentication in the Authentication section at the top of the Create New Web Application page. The settings that appear if you selected Classic Mode Authentication are shown in Figure 3-3.



Security Configuration

Kerberos is the recommended security configuration to use with Integrated Windows authentication. Kerberos requires the application pool account to be Network Service or special configuration by the domain administrator. NTLM authentication will work with any application pool account and the default domain configuration.

If you choose to use Secure Sockets Layer (SSL), you must add the certificate on each server using the IIS administration tools. Until this is done, the web application will be inaccessible from this IIS web site.

Authentication provider:

○ Negotiate (Kerberos)

● NTLM

Allow Anonymous

○ Yes

● No

Use Secure Sockets Layer (SSL)

○ Yes

● No

FIGURE 3-3 Security Configuration settings

If you selected Classic Mode Authentication, you must designate the authentication provider for the web application. You can select NTLM or Negotiate (Kerberos) as the authentication provider. Classic Mode Authentication essentially uses IIS to authenticate users with built-in Windows authentication providers, including NTLM, Kerberos, and Basic authentication. However, you cannot select Basic authentication when you create a web application—you must configure Basic authentication after the web application has been created. You will learn more about Windows authentication providers later in this lesson.

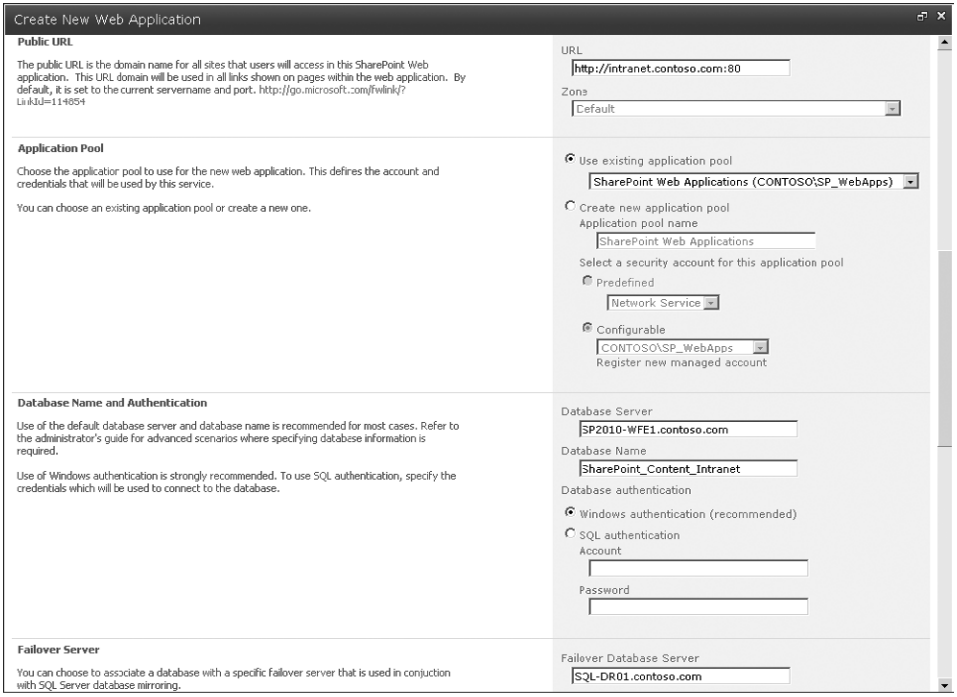
If you selected Claims Based Authentication, the authentication provider is configured in the Claims Authentication Types section of the Create New Web Application page.

For both authentication types, you must specify whether anonymous authentication is allowed and whether SSL is enabled in the Security Configuration section. By default, anonymous access and SSL are disabled. Later in this lesson, you will learn about the additional steps required to implement anonymous access and SSL.

Claims Authentication Types and Sign In Page URL

The Claims Authentication Types and Sign In Page URL sections are visible only if you selected Claims Based Authentication. One or more authentication providers, and a sign-in page are required for Claims Based Authentication. You will learn more about these settings later in this chapter.

After you have configured security and authentication settings, you must configure additional settings for a web application, shown in Figure 3-4.



Create New Web Application

Public URL

The public URL is the domain name for all sites that users will access in this SharePoint Web application. This URL domain will be used in all links shown on pages within the web application. By default, it is set to the current servername and port. http://go.microsoft.com/fwlink/?LinkId=114854

URL: http://intranet.contoso.com:80

Zone: Default

Application Pool

Choose the applicatior pool to use for the new web application. This defires the account and credentials that will be used by this service.

You can choose an existing application pool or create a new one.

● Use existing application pool

SharePoint Web Applications (CONTOSO\SP_WebApps)

○ Create new application pool

Application pool name: SharePoint Web Applications

Select a security account for this application pool

○ Predefined: Network Service

● Configurable: CONTOSO\SP_WebApps

Register new managed account

Database Name and Authentication

Use of the default database server and database name is recommended for most cases. Refer to the administrator's guide for advanced scenarios where specifying database information is required.

Use of Windows authentication is strongly recommended. To use SQL authentication, specify the credentials which will be used to connect to the database.

Database Server: SP2010-WFE1.contoso.com

Database Name: SharePoint_Content_Intranet

Database authentication

● Windows authentication (recommended)

○ SQL authentication

Account:

Password:

Failover Server

You can choose to associate a database with a specific failover server that is used in conjuction with SQL Server database mirroring.

Failover Database Server: SQL-DR01.contoso.com

FIGURE 3-4 Web Application settings

Public URL

The Public URL represents the user-accessible URL of the web application. In the URL box, type the protocol, the fully qualified domain name (FQDN) of the web application, and the port that will be used in URIs of requests to the site, such as *http://intranet.contoso.com:80*.

You will notice that the Public URL is associated with the zone named Default, and you cannot change the zone when creating a new web application. A zone is a path through which content in a web application is actually accessed. When SharePoint receives a URI—for example, *http://intranet.contoso.com/SitePages/Home.aspx*—SharePoint examines the protocol, FQDN, and port of the URI and uses those three elements to identify both the SharePoint web application that is being requested and the zone through which the request is received—in this example, the default zone of the Contoso Intranet Web application.

> ***NOTE*** **SHAREPOINT 2007**
>
> **In SharePoint 2007, this setting was called the *Load Balanced URL*.**

Application Pool

Use the controls in the Application Pool section to specify whether the web application will be hosted within an existing application pool, running in the context of the identity that has been already assigned to the application pool, or within a new application pool running in the context of a managed account that you select in the Configurable list. Earlier in this lesson, you learned that it is a best practice to use a shared application pool for web applications unless there is a significant driver for process isolation, because application pools incur memory and performance overhead and are therefore a limited resource of IIS.

Database Name and Authentication

The Database Server box is prepopulated with the name of the server that hosts the farm's configuration database. If you want to host the web application's content database on another server, replace the value using the <*SERVERNAME\instance*> format, where *SERVERNAME* is the FQDN of the database server and *instance* is the Microsoft SQL Server instance you want to use, if more than one instance is running on the server.

The Database Name box is prepopulated with a sample name that includes a globally unique identifier (GUID). Most database administrators (DBAs) prefer to follow a naming standard that uniquely identifies the database with a descriptive name that does not include a GUID. Replace the default name with a name that follows your naming standards. A guideline is to use a name that follows this example: *SharePoint_Content_Intranet*, where the first two elements of the name identify the database as a SharePoint content database, and the remaining elements of the name correlate to the web application and site collections contained in the database.

In the Database Authentication section, select the method used to connect to the content database. The default and recommended method is Windows authentication, which uses the credentials of the application pool identity to connect to SQL Server. Windows authentication automatically encrypts the password.

If you have configured the SQL Server for mixed mode authentication, you can select SQL authentication. You must specify the credentials that the web application will use to connect to the database. Type the user name in the Account box and the password in the Password box. The user account with which you are logged on to Central Administration must have permission to create and secure databases on the server.

BEST PRACTICES **ENCRYPT SQL AUTHENTICATION**

SQL authentication sends the SQL authentication password to the SQL Server unencrypted. We recommend that you only use SQL authentication if you force protocol encryption to the SQL Server or encrypt your network traffic by using IPsec.

Failover Server

SharePoint 2010 supports failover to a second instance of the database. If you have configured database mirroring in SQL Server, SharePoint can failover to another server in the event that the current database server becomes unresponsive. In the Failover Database Server box, type the name of a specific failover database server for the content database. This setting does not configure SQL database mirroring—it only instructs SharePoint to failover to an already-configured backup instance of the database. You will learn about failover in Chapter 11, "Implementing Business Continuity."

Search Server

The Search Server setting is automatically configured, and cannot be changed, if SharePoint Server 2010 is installed. On a SharePoint Foundation 2010 farm, you associate a search server running SharePoint Foundation 2010 Search service with the content database for the new web application.

Service Application Connections

A web application connects to service applications for shared services such as search. Service application connections are grouped into application connection groups, also called *proxy groups*. In the Service Application Connections section, select either an existing application connection group, or select Custom from the drop-down list and then select the specific service applications you want the web application to use. You will learn more about service application management in Chapter 5.

Customer Experience Improvement Program

Click Yes or No to opt in or out, respectively, of the Customer Experience Improvement Program. If you choose Yes, certain information will be sent to Microsoft that will help Microsoft understand performance and usage patterns of SharePoint implementations in the real world.

Delete a Web Application

You can delete a web application by using Central Administration. As with other changes that involve components of both SharePoint and IIS, you should not use IIS Manager to delete an IIS Web site that services a SharePoint web application. Exercise care when deleting a web application. Before doing so, verify that you have a backup of the web application and of the farm's configuration.

DELETE A WEB APPLICATION USING CENTRAL ADMINISTRATION

1. In the Central Administration Quick Launch, click Application Management.
2. In the Web Applications section, click Manage Web Applications.
3. Click the Web application you want to delete.
4. On the ribbon, click Delete.

   The Delete Web Application page opens.
5. If you want to delete the content databases, click Yes in the Delete Content Databases section.
6. If you want to delete the IIS Web sites associated with the web application, click Yes in the Delete IIS Web Sites section.

> ***NOTE*** **UNDERSTAND YOUR OPTIONS**
>
> **It is possible to delete the definition of the web application in the farm configuration database while leaving both the content databases and the IIS Web sites in place. Although rarely used, this option can be helpful if the configuration of a web application has been corrupted.**

7. Click Delete.

DELETE A WEB APPLICATION USING WINDOWS POWERSHELL

The following example shows the use of the *Remove-SPWebApplication* cmdlet to delete a site:

```
Remove-SPWebApplication <URL> -DeleteIISSite -RemoveContentDatabase -Confirm:$false
```

Where:

- *<URL>* is the URL to the web application that you want to delete.
- The *-DeleteIISSite* switch parameter, if present, instructs SharePoint to delete the IIS Web site associated with the web application.
- The *-RemoveContentDatabase* switch parameter, if present, instructs SharePoint to delete the content databases associated with the web application.
- The *-Confirm:$false* parameter suppresses confirmation prompts.

Secure Communication with a Web Application Using SSL

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols used to encrypt the contents of communications over a network, at the application layer. In the case of SharePoint, the communication is between a client and an IIS Web site, or communications between SharePoint web applications and service applications.

SSL is particularly important if sensitive information will be transmitted to or from a website—without SSL, the information is transmitted in clear text and could be intercepted by a packet sniffer. SSL becomes more important when transmission of information is over untrusted networks, such as the Internet.

Secure communication is made possible with certificates and keys. When a client initiates contact with a secured website, the website provides the client a certificate. Through the series of ensuing processes, the two endpoints agree on a secret—a key—that is used to encrypt and decrypt communications.

The client can also use the server's certificate to verify the identity of the server, by validating the digital signature of the certificate against a trusted certificate authority. In this way, SSL can be used to authenticate a server prior to sending sensitive information to the server. For example, if Internet Explorer cannot verify that a server's certificate is valid, it warns the user and the user can then decide whether to accept the inherent risks and to continue communicating with the server.

To secure communications with a SharePoint web application, you must perform the following steps:

1. Configure the SharePoint web application to use SSL.
2. Create a certificate.
3. Bind the certificate to the IIS Web site of the SharePoint web application.

When you enable SSL for the SharePoint web application, you change the web application scheme to SSL in the configuration database, and you enable SharePoint to recognize the HTTPS protocol in the URL. But SharePoint Server 2010 does not itself provide SSL services and does not store the certificate used to authenticate the web application. These roles are performed by IIS.

Configure a SharePoint Web Application to Use SSL

When you create a new web application, you can enable SSL by clicking Yes for the Use Secure Sockets Layer (SSL) setting of the Create New Web Application page. The procedure to create a web application was introduced in Chapter 1, and the settings for a new web application were detailed earlier in this lesson.

To change an existing HTTP web application to use HTTPS, you must modify access mappings and zones. These procedures will be detailed in Lesson 3.

After the SharePoint web application has been configured to use SSL, you must manage certificates and bindings on each web server in the farm.

Create a Certificate

SSL relies on a certificate provided by the web server to the client. If you want the client to be able to verify the server's identity, you must create a certificate request and send that request to a known certificate authority (CA), such as VeriSign or GeoTrust, or obtain a certificate from an online CA in your domain—for example, from Active Directory Certificate Services.

In a test environment, you can create a self-signed certificate on the web server. The certificate can be used to test SSL configuration and communication with an IIS Web site, but clients will be unable to verify the identity of the server.

CREATE A SELF-SIGNED CERTIFICATE

1. In IIS Manager, in the console tree, click the node representing the server, for example SP2010-WFE1.

> *TIP* **SELECT THE SERVER**
>
> **Be sure that you select the server node in the IIS console tree, not an IIS Web site. Certificates are stored in the Windows Server certificate store, not in IIS itself.**

2. In the IIS section, double-click Server Certificates.
3. In the Actions panel, click Create Self-Signed Certificate.

   The Create Self-Signed Certificate dialog box opens.
4. In the Name box, type a friendly name for the certificate, such as **Test Certificate**.
5. Click OK.

Bind an SSL Certificate to an IIS Web Site

After you have added a certificate to IIS, you can bind the certificate to an IIS site.

CREATE AN SSL BINDING FOR AN IIS WEB SITE

1. In IIS Manager, in the console tree, click the node representing the IIS Web site for which you want to create an SSL binding.
2. In the Actions panel, click Bindings.

   The Site Bindings dialog box opens.
3. Click Add.

   The Add Site Binding dialog box opens.
4. In the Type list, select HTTPS.
5. Optionally, in the IP Address list, select a specific IP address. Otherwise, accept the default value, All Unassigned.

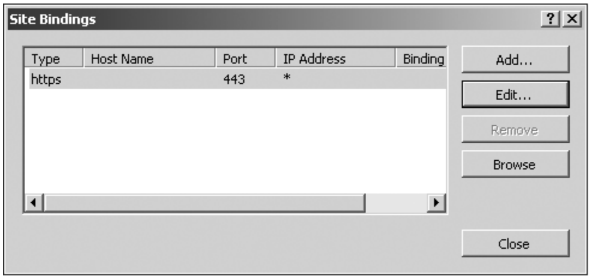
   If you are hosting more than one SSL-enabled web application on a server, you might want to bind each to a specific IP address to avoid using a wildcard SSL certificate. IP address bindings also make it easier to configure network load balancing for high availability and performance.

   Wildcard SSL certificates allow you to secure multiple subdomains under a single parent domain. For example, you could obtain a wildcard SSL Certificate for **.contoso.com*. Wildcard certificates do not allow clients to verify the identity of a specific web application, and can make it more difficult to trace network traffic.
6. Optionally, in the Port box, type a port number or accept the default value, 443.

The default port for the HTTPS protocol is 443.

7. In the SSL Certificate list, select the certificate—for example, Test Certificate.

   The resulting configuration is shown in Figure 3-5.



Site Bindings

Type	Host Name	Port	IP Address	Binding
https		443	*	

FIGURE 3-5 An SSL binding for an IIS Web site

8. Click OK and then click Close.

After you have added a new binding for SSL, you can remove any other bindings that are no longer needed.

MORE INFO **CONFIGURING SSL**

The following article provides additional details regarding the configuration of SSL: "How to Set Up SSL on IIS 7" at *http://go.microsoft.com/fwlink/?LinkId=187887*.

IMPORTANT **RESTORE IIS CONFIGURATION WHEN YOU RESTORE SHAREPOINT**

SSL certificates, SSL bindings, and IP address bindings are not stored in the farm configuration database. If you must restore a web application or web server, you will need to reconfigure IIS Web sites or restore IIS configuration.

✓ **Quick Check**

- **When you configure SSL for a SharePoint web application, what must you do on each server in the SharePoint farm?**

Quick Check Answer

- **Bind the SSL certificate to the IIS Web site**

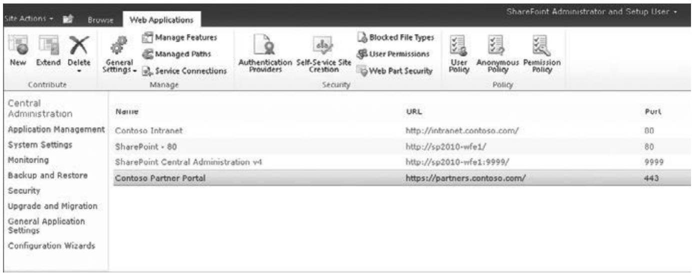
Configure Web Applications

After creating a web application, you can specify additional configuration for the web application. In Lesson 2, you will learn to configure authentication. In Lesson 3, you will learn to configure authentication zones and alternate access mappings. You must also create one or more site collections and, if needed, additional content databases. You will learn about site collections and content databases in Chapter 4, “Administering and Securing SharePoint Content.”

IMPORTANT **NEXT, CREATE A SITE COLLECTION**

Until you create a site collection, a new web application contains no content. Users navigating to the web application will be presented with an error page when a site has no content.

On the Web Applications Management page of Central Administration, you can select a web application and then configure settings for the web application by clicking buttons on the ribbon, shown in Figure 3-6. The remainder of this lesson will cover a variety of web application settings, and will point you to other locations in this Training Kit that provide additional detail.



Name	URL	Port
Contoso Intranet	http://intranet.contoso.com/	80
SharePoint - 80	http://sp2010-wfe1/	80
SharePoint Central Administration v4	http://sp2010-wfe1:9999/	9999
Contoso Partner Portal	https://partners.contoso.com/	443

FIGURE 3-6 The Web Applications Management page and ribbon

Master Page Setting For Application _Layouts Pages

Administrative pages for SharePoint sites are common across all sites in the farm. For example, a site’s Site Settings page, Settings.aspx, is the same ASPX page used by all sites in the farm. This is possible because each SharePoint site is created with a virtual directory called _layouts that points to a common location. Of course, the content that is displayed by the page might be different for each site, based on the site’s features and configuration, but the page itself is common. Because administrative pages are located in the _layouts virtual directory, administrative pages are often called _layouts pages.

Like other ASPX pages, an administrative page refers to a master page, which determines the look and feel and functionality of all pages that refer to it.

CONFIGURE APPLICATION _LAYOUTS MASTER PAGES

1. On the Web Applications Management page of Central Administration, select the web application that you want to configure.
2. On the ribbon, click the General Settings drop-down arrow, and then click General Settings.

   The Web Application General Settings page opens.

3. In the Master Page Setting For Application _Layouts Pages section, for the Application _Layouts Pages Reference Site Master Pages setting, click Yes or No.
4. Click OK.

If the Application _Layouts Pages Reference Site Master Pages setting is disabled (set to No) the administrative pages will use the application.master page in the SharePoint Root directory as their master page. This master page presents the default look and feel of SharePoint Server 2010.

If you have customized the master pages of a site to incorporate custom functionality or branding, you probably want those changes to be visible both on standard content pages and administrative pages. It would not be acceptable for administrative pages to lose the customizations you have made to the site. Therefore, by default, this setting is enabled (set to Yes). This instructs SharePoint to use the site's master pages—rather than the standard, shared application.master page—when you access an administrative page in a site.

However, this presents a risk that if a site's master pages become corrupt or inaccessible, an administrative page will not be able to load, and you could be locked out of the ability to manage a site. Therefore, even if this setting is enabled, if SharePoint cannot render a vital page, such as the Settings.aspx page, because of problems with the site's master pages, the page will be rendered with the default SharePoint master pages, so that the page can be returned to the user successfully.

Recycle Bin Configuration

SharePoint sites support a two-stage Recycle Bin by default. When a user deletes content, the content is moved to the first-stage Recycle Bin, from which the user can restore the content or empty the Recycle Bin. When the Recycle Bin is emptied, content is moved to the second-stage Recycle Bin. Once the content is in the second-stage Recycle Bin, only a site collection administrator can restore it. When the second-stage Recycle Bin is emptied by a site collection administrator, or after another configurable time frame, the content is permanently deleted from the content database. By default, the Recycle Bin is enabled for a new web application, and default configuration is applied.

You should modify Recycle Bin settings in accordance with your specifications.

CONFIGURE RECYCLE BIN SETTINGS FOR A WEB APPLICATION

1. On the Web Applications Management page of Central Administration, select the web application that you want to configure.
2. On the ribbon, click the General Settings drop-down arrow, and then click General Settings.

   The Web Application General Settings page opens.
3. In the Recycle Bin section, in the Recycle Bin Status section, click On to enable the Recycle Bin or click Off to disable the Recycle Bin.

   If you disable the Recycle Bin, all content will be expunged from both first and second stages on Recycle Bins of all sites and site collections in the web application.

4. If you enable the Recycle Bin, you can configure the Delete Items In The Recycle Bin setting and the Second Stage Recycle Bin setting, as discussed later in this section.
5. Click OK.

If the Recycle Bin is enabled, as it is by default, you can configure the time-based expiration of content. Content in a Recycle Bin will be permanently deleted after the number of days that you specify. This time limit does not apply to the first- or second-stage Recycle Bins individually, nor is the time reset when an item is moved to the second-stage Recycle Bin. Instead, the time limit is measured from the time at which the content was deleted.

You can disable the second-stage Recycle Bin by clicking Off in the Second Stage Recycle Bin Setting group. By default, the second-stage Recycle Bin is on, and SharePoint limits the second-stage Recycle Bin size to 50 percent of the storage limit quota for the site collection. However, by default, new site collections have no quota applied, which effectively means that the second-stage Recycle Bin size is also unlimited.

As you can see, you should carefully plan and configure Recycle Bin settings.

General Settings

The Web Application General Settings page exposes many common web application settings, including the Recycle Bin and Master Page Setting For Application _Layouts Pages configuration discussed earlier.

MODIFY WEB APPLICATION GENERAL SETTINGS

1. On the Web Applications Management page of Central Administration, select the web application that you want to configure.
2. On the ribbon, click the General Settings drop-down arrow, and then click General Settings.

   The Web Application General Settings page opens.
3. Modify the settings, as described below, and then click OK.

The following additional settings can be configured on the Web Application General Settings page.

- **Default Time Zone** By default, each web application uses the time zone of the host operating system, and each site uses the time zone of the parent web application. It is recommended that you manually configure the time zone to prevent potential inconsistencies, particularly across servers of a multi-server farm.
- **Default Quota Template** When you create a site collection, you can configure the quota for the site collection, which establishes storage limits and warning levels at which administrators can be notified by email that the size of a site collection is approaching its storage limit. This setting, at the web application level, determines the default quota template for new site collections. You must have previously created a quota template before you can configure the Default Quota Template for a web application. See Chapter 4 for more information about quotas.

NOTE CHANGES AFFECT ONLY NEW SITE COLLECTIONS

If you change the Default Quota Template setting, no change is made to existing site collections. The setting only affects the default quota template assigned to new site collections.

- **Person Name Actions And Presence Settings** This setting determines whether online status of users will be displayed within the web application. Online status can be queried from Microsoft Office Communicator Server (OCS), and can be displayed next to a user's name wherever the user's display name appears. Additionally, if you right-click a user name, additional commands will appear that allow you to communicate directly with the user. By default, this setting is enabled for a new web application.

NOTE NO NEED TO DISABLE SETTINGS IF YOU DO NOT HAVE OCS

If you do not have OCS in your environment, there is no reason to disable Person Name Actions And Presence Settings. The setting will simply be ignored by SharePoint.

- **Alerts** Alerts are email notifications regarding changes to content in a list, library, folder, page, item, or document. By default, users are allowed to create alerts—up to 500 alerts across all sites in the web application. It is recommended that you do not configure the limit too high, or choose Unlimited, because it opens the possibility that a user might create sufficient alerts to degrade the performance of SharePoint or Exchange Server.
- **RSS Settings** Really Simple Syndication (RSS) feeds allow users and applications to monitor content in lists and libraries. For example, a user can subscribe to alerts in a list or library using an RSS reader such as Internet Explorer or Microsoft Outlook 2010. By default, RSS feeds are enabled for a web application, and the RSS feed of each list and library is enabled. You can disable RSS at the web application level.

  If RSS is enabled for the web application, you can enable, disable, and customize the RSS feed of a specific list or library. Open the Settings page for the list or library, and then click RSS Settings. The Modify RSS Settings page opens, with which you can configure RSS settings.
- **Blog API Settings** The MetaWeblog API is a standard API used by many blog applications to accept blog posts published directly from blogging applications, including Microsoft Office Word 2010. By default, the blog API is enabled. You can also configure whether the web application's authentication will be used to authenticate the user, or whether the API should accept the user's user name and password.
- **Browser File Handling** By default, SharePoint protects users by preventing certain types of files, such as HTML files, from being executed locally when a user clicks the file on the SharePoint site. SharePoint adds headers to these sensitive file types that cause

the browser to prompt the user to download the file, rather than allowing the browser to open the file immediately. This default setting, called Strict, should not be changed unless you have specific reasons to do so and you are in a controlled environment. The Strict setting also prevents attacks such as cross-site scripting from compromising the integrity of your server farm by forcing code in such files to be executed on the client browser, instead of on the SharePoint server.

- **Web Page Security Validation** When enabled, as it is by default, this setting prevents a client session from being used indefinitely. By default, 30 minutes after authentication of a request for a page, the client's security validation expires. Therefore, after 30 minutes of inactivity, the user must refresh the page or otherwise reestablish the connection, at which point authentication will be performed.

  The setting does not produce a visible effect for web applications that use Classic Mode authentication, which uses Windows authentication mechanisms. Internet Explorer will transparently re-authenticate the user, as long as the web application's URL is in Internet Explorer's Trusted Sites or Local Intranet security zone. If the web application uses other authentication providers, a sign-in page will be presented and the effect of re-authentication will be more noticeable.
- **Send User Name And Password In E-mail** SharePoint 2010 can be installed in Active Directory account creation mode. This mode is included for Internet Service Providers and is being deprecated. The mode is not enabled by default, is being deprecated from SharePoint, and is included primarily to support upgrades of legacy environments for SharePoint hosting services. It is not recommended that you configure an organizational SharePoint farm in Active Directory account creation mode. SharePoint also ignores this setting when it is not installed in Active Directory account creation mode. Therefore, you can ignore this setting.
- **Maximum Upload Size** By default, a user cannot upload a single file, group of files, or other piece of content greater than 50 MB in size. You can modify this limit, but you must be cognizant of timeouts when transferring large files using HTTP, which is the transfer mechanism used by SharePoint, particularly over slow or high-latency networks such as the Internet or a WAN link.

NOTE **2 GB MAXIMUM FILE SIZE**

There is a fixed limit of 2 GB for any file stored in SharePoint. It is not possible to store files larger than 2 GB in a SQL record. Some third-party solutions might address scenarios that require SharePoint-based interaction with files larger than 2 GB.

IMPORTANT **MAXIMUM UPLOAD SIZE AFFECTS UPLOAD MULTIPLE FILES**

The Maximum Upload Size setting limits the aggregate size of a single upload action, which includes uploads using the Upload Multiple Files command. For example, by default, you cannot upload 10 files of 10 MB each with the Upload Multiple Files command because the total size of 100 MB exceeds the default limit of 50 MB.

- **Customer Experience Improvement Program** See the description of this setting in the "Configure a New Web Application" section.

Workflow Settings

From the General Settings menu, you can configure workflow settings for a web application. The Workflow Settings page exposes the following configuration:

- **Enable User-Defined Workflows** By default, this option is set to Yes and workflows are enabled for a new web application. Users can create declarative workflows—workflows that are based on building blocks that are available by default, such as SharePoint Designer workflow activities, or code that has been deployed to the server by an administrator. Users cannot add new compiled code workflows to the server. Users must have at least the Design permission level for a site to create a workflow in that site.
- **Alert Internal Users Who Do Not Have Access** Enabled—set to Yes—by default, this option will send an email notification to a user who has been assigned a task in a workflow. The email will include a hyperlink that will generate an access request for the site, at which point an administrator can grant the user the permissions necessary to perform the workflow task. If this option is disabled—set to No—a user who does not have access to the target item of the workflow task will not be notified of the task.
- **Allow External Users To Participate In Workflow** When this option is enabled, SharePoint will email a copy of a document to a user who has been assigned a workflow task related to the document. For security reasons, and to reduce the proliferation of independent copies of documents, this option is disabled by default.

Outgoing Email Settings

Outgoing email settings are required for alerts to function. You learned in Chapter 1 that after creating a SharePoint farm you should configure the outgoing email settings for the farm. By default, a web application will use those farm-level outgoing email settings. However, you can override the outgoing email settings for a specific web application. You must define the SMTP Relay Server, From Address, and Reply To Address.

CONFIGURE OUTGOING E-MAIL SETTINGS FOR A WEB APPLICATION

1. On the Web Applications Management page of Central Administration, select the web application that you want to configure.
2. On the ribbon, click the General Settings drop-down arrow, and then click Outgoing E-mail.

   The Web Application Outgoing E-Mail Settings page opens.
3. In the Outbound SMTP Server box, type the FQDN of an SMTP-compliant server to which SharePoint can connect to using TCP port 25.

> ***IMPORTANT*** **The SMTP server must be accessible over TCP port 25, and must permit relay from servers based on IP address. SharePoint products do not support SMTP authentication mechanisms.**

4. In the From Address box, type the email address that will be used as the sender's address of outgoing email messages.
5. In the Reply-To Address box, type the email address to which replies should be sent.
6. In the Character Set list, select the character set for email messages. The default is 65001 (Unicode UTF-8), which is the character set most commonly used for email, and supports characters in all languages supported by Unicode.
7. Click OK.

IMPORTANT **SharePoint allows you to configure only one SMTP server address. If you want to ensure availability in the event of a failure of an SMTP server, you must configure redundancy outside of SharePoint's configuration.**

Text Message Service Settings

If users do not have smart phones with which to monitor SharePoint email alerts, you can send alerts via text message, which allows alerts to be sent to almost any cellular telephone.

You must first subscribe to a third-party SMS service provider. The SMS provider relays alerts, based on the email address of the user in the alert, to the user's mobile phone.

MORE INFO **SMS PROVIDERS**

You can find an up-to-date list of SharePoint 2010 compatible SMS providers at *http://messaging.office.microsoft.com/HostingProviders.aspx?src=O14&lc=1033*.

Your costs will vary based on factors including your geographic location, volume of SMS alerts, and fees imposed by your cellular telephone provider.

IMPORTANT **THROTTLING MOBILE ALERTS**

There is no way to throttle alerts sent by SharePoint. If users create many alerts, and those alerts are sent via SMS, your costs might skyrocket. Consider throttling capabilities of your SMS provider as a way to limit out-of-control costs of mobile alerts.

CONFIGURE SMS-BASED ALERTS

1. On the Web Applications Management page of Central Administration, select the web application that you want to configure.
2. On the ribbon, click the General Settings drop-down arrow, and then click Mobile Account.

   The Web Application Text Message (SMS) Service Settings page opens.
3. In The URL Of Text Message (SMS) Service box, type the URL provided by your SMS provider.

4. In the User Name box, type the user name provided by your SMS provider.
5. In the Password box, type the password provided by your SMS provider.
6. Click Test Service to test the configuration of the service.
7. Click OK.

Self-Service Site Creation

By default, you must be a member of the Farm Administrators group to create a site collection in a web application. However, in certain scenarios—a team or project collaboration web application, for example—you might want users to be able to create site collections without administrator intervention. To support these scenarios you can enable self-service site collection creation, which is disabled by default, by using one of two interfaces in Central Administration.

ENABLE SELF-SERVICE SITE CREATION FROM THE WEB APPLICATIONS MANAGEMENT PAGE

1. Select the web application for which you want to enable self-service site creation.
2. In the Web Applications ribbon, click Self-Service Site Creation.

   The Self-Service Site Collection Management page opens.
3. In the Enable Self-Service Site Creation section, click On.

   Optionally, select the Require Secondary Contact check box. If this check box is selected, a user will be required to provide a secondary contact when the user creates a site collection. The secondary contact becomes the secondary site collection administrator. A primary site collection administrator is always required.
4. Click OK.

ENABLE SELF-SERVICE SITE CREATION FROM THE APPLICATION MANAGEMENT PAGE

1. In the Central Administration Quick Launch, click Application Management.
2. In the Site Collections section, click Configure Self-Service Site Creation.

   The Self-Service Site Collection Management page opens.
3. Click the Web Application picker, and then click Change Web Application.

   The Select Web Application page opens.
4. Click the name of the web application for which you want to enable self-service site creation.
5. In the Enable Self-Service Site Creation section, click On.
6. Optionally, select the Require Secondary Contact check box.
7. Click OK.

In Chapter 4, you will learn how to create a site collection when self-service site creation is enabled for the web application.

NOTE **CREATING A SELF-SERVICE SITE COLLECTION**

Even though the feature is called *Self-Service Site Creation*, it is actually self-service site *collection* creation that is managed by this setting. If this feature is enabled, users can create a new site collection, for which they can assign themselves as site-collection administrators. As for creating a subsite within a site collection, users can do so if they have the Create Subsites permission on the parent site.

Blocked File Types

SharePoint Server 2010 allows you to prevent certain types of files from being uploaded to a web application, based on file extension.

CONFIGURE BLOCKED FILE TYPES

1. On the Web Applications Management page of Central Administration, select the web application that you want to configure.
2. On the ribbon, click Blocked File Types to open the Blocked File Types page.

   You can open the Blocked File Types page with an alternate method. In the Central Administration Quick Launch, click Security. Then, in the General Security section, click Define Blocked File Types. The Blocked File Types page opens. Click the Web Application picker, and then click Change Web Application. The Select Web Application page opens. Click the name of the web application for which you want to define blocked file types.
3. Add or remove extensions, each on a separate line of the list, and then click OK.

MORE INFO **WHEN TO BLOCK FILE TYPES**

For guidance regarding which file types should and should not be blocked, see the *Microsoft SharePoint 2010 Administrator's Companion* by Bill English, Brian Alderman, and Mark Ferraz (Microsoft Press, 2010).

The extension-based protection provided by the Blocked File Types list is rudimentary. Users can change the extension of a file and then upload it for storage. For example, you can rename a blocked .exe file with a .txt extension, and then upload the file to a document library. SharePoint looks only at the extension. However, you cannot upload a file and then change the extension.

IMPORTANT **SECURITY TIP**

Microsoft Forefront for SharePoint Server can manage blocked file types based on more sophisticated rules such as the binary file type.

Other Settings in the Web Applications Ribbon

The Web Applications ribbon also exposes a number of additional settings that are scoped to a web application. The following settings are discussed in Chapter 10, "Administering SharePoint Customization":

- **SharePoint Designer governance** From the General Settings menu, you can manage what users are able to do with SharePoint Designer within the web application.
- **Manage Features** Features are bundles of functionality that can be enabled or disabled for scopes of the SharePoint logical architecture, including web applications, site collections, and sites.
- **Web Part Security** You can define the availability, behavior, and security of web parts in the web application.

Chapter 4 details the following settings:

- **User Permissions** You can configure the granular permissions available to be used in permission levels defined for sites in the web application.
- **User Policy** You can define access policies at the web application that override any permissions, or lack thereof, for content within the web application. For example, the SharePoint search crawling account is assigned a Full Read permission policy for each new web application. This enables the search crawling account to index all SharePoint content, without the need to explicitly assign permissions to content in the web application. User policy is actually scoped to zones, not to web applications. You will learn more about zones in Lesson 5.
- **Permissions Policy** Permission policies are collections of permissions that can be assigned to a user or group as a user policy, as described earlier. SharePoint 2007 had a fixed number of permission policies: Full Control, Full Read, Deny Write, and Deny All. In SharePoint 2010, you can define custom permission policies.

The remaining settings are discussed later in this Training Kit:

- **Resource Throttling** SharePoint 2010 introduces resource throttling, which is designed to protect a server and the users of a server from the negative impact of large queries and other performance-degrading activities. Resource throttling allows you to control resource utilization and optimize server performance. Resource Throttling settings are exposed by the General Settings menu. You will learn about resource throttling in Chapter 12, "Monitoring and Optimizing SharePoint Performance."
- **Managed Paths** Managed paths specify the URLs in a web application at which site collections can be created. While managed paths are a property of a web application, they are conceptually more related to site collections, and are therefore detailed in Chapter 4.

- **Service Connections** You can specify the application connection group with which a web application is associated, or you can specify individual service applications to which a web application connects. Service applications, application connection groups, and web application associations are discussed in Chapter 5.
- **Authentication Providers** Lesson 2 of this chapter details the configuration of web application authentication. Authentication providers are scoped to the zone, not to the entire web application. You will learn more about zones in Lesson 3 of this chapter.
- **Anonymous Policy** Anonymous access restrictions are described in Lesson 2.

PRACTICE Configure Web Applications

Practices are designed to guide you through important procedures. The instructions in the Training Kit are high-level instructions that will challenge you to think carefully and to apply the procedures that are covered in this lesson and elsewhere in the Training Kit. If you need assistance, consult the detailed, step-by-step instructions in the Practice Answers on the companion media.

In this practice, you will create a web application to support collaboration with Contoso partners. The web application will be accessible from the Internet, so you want to ensure that communication between clients and the web application is secure. Therefore, you will configure the web application to use SSL. Finally, you will make configuration changes to support both the business and governance requirements of the partner collaboration website.

Prepare for the Practice

Before you perform this practice, you must ensure that your lab environment has been built according to the instructions found in the Introduction to this Training Kit.

1. Apply the snapshot CHAPTER01 to CONTOSO-DC.
2. Apply the snapshot CHAPTER01 to SP2010-WFE1.
3. Start CONTOSO-DC.

   Wait for the virtual machine to complete startup, at which time the Press Ctrl+Alt+Del prompt appears.
4. Start SP2010-WFE1.

EXERCISE 1 Add DNS Host Records for New Web Applications

In this exercise, you add DNS host records for web applications you will create in subsequent exercises.

1. Log on to SP2010-WFE1 as **CONTOSO\SP_Admin** with the password **Pa$$w0rd**.
2. Start Command Prompt.
3. Use Dnscmd.exe to create a new host (A) records on the DNS server (*contoso-dc.contoso.com)* for *partners.contoso.com* that resolve to the IP address *10.0.0.21*.

4. Use Dnscmd.exe to create a new host (A) records on the DNS server (*contoso-dc .contoso.com)* for *extranet.contoso.com* that resolve to the IP address, *10.0.0.21*. Then close Command Prompt.

EXERCISE 2 Create a Web Application Using Central Administration

In this exercise, you create a web application for collaboration with partners of Contoso.

1. Use Central Administration to create a web application collaboration with partners. Use the following specifications and guidance:
   - Authentication: Classic Mode Authentication
   - Name: Contoso Partner Portal
   - Port: 443
   - Host header: partners.contoso.com
   - Authentication provider: NTLM
   - Anonymous authentication: No
   - Secure Sockets Layer (SSL): Yes
   - URL: *https://partnerss.contoso.com:443*
   - Application pool: SharePoint Extranet Applications
   - Application identity: CONTOSO\SP_WebApps
   - Content database name: SharePoint_Content_Partners

EXERCISE 3 Create a Site Collection Using Central Administration

In this exercise, you use Central Administration to create a site collection at the root of the new web application.

1. Use Central Administration to create a site collection. Use the following specifications and guidance:
   - Web application: *https://partners.contoso.com*
   - Title: Contoso Partner Portal
   - Description: Sites for collaboration with partners
   - URL: *https://partners.contoso.com/*
   - Template: Team Site
   - Primary site collection administrator: CONTOSO\SP_Admin
2. Open a new tab in Internet Explorer and browse to ***https://partners.contoso.com***.

   An Internet Explorer Cannot Display The Webpage error page opens. The site cannot be accessed using HTTPS because SSL has not been configured for the IIS Web site associated with the application.

EXERCISE 4 Create a Self-Signed Certificate

In this exercise, you create a self-signed certificate that, in the next exercise, you will bind to the site to enable SSL.

- In IIS Manager, create a self-signed certificate named **Test Certificate** in the certificate store of SP2010-WFE1.

EXERCISE 5 Create an SSL Binding for an IIS Web Site

In this exercise, you bind the certificate you created in the previous exercise to the Contoso Partner Portal IIS Web site.

1. Bind the certificate named *Test Certificate* to the Contoso Partner Portal IIS Web site by modifying the site's existing incomplete binding.
2. In Internet Explorer, browse to ***https://partners.contoso.com***.

   An error page opens: *There is a problem with this website's security certificate.*

   Question: Why does this error appear?
3. Click Continue To This Website (Not Recommended).

   The site is loaded, compiled, and cached for first-time access, and then authentication proceeds. The Windows Security dialog box opens.

   Question: Why does this dialog box appear?
4. Authenticate as **CONTOSO\SP_Admin** with the password **Pa$$w0rd**.

   The site is loaded, compiled, and cached for first-time access, and then the site opens.

   If an error appears, refresh the page. It is possible that the client timed out while the site was being loaded by IIS.

EXERCISE 6 Configure Web Application Settings

In this exercise, you enable self-service site creation, configure the Recycle Bin to retain items for 60 days, and prevent users from uploading MP3 files.

1. Switch to the Internet Explorer tab that displays Central Administration. Navigate to the Web Applications Management page, and then make the following changes to the configuration of the Contoso Partner Portal Web Application:
   - Enable Self-Service Site Creation. Require that users add a secondary site collection administrator.
   - Configure the Recycle Bin to retain items for 60 days.
   - Block the upload of MP3 files. For additional manageability and elegance, add the MP3 file extension to the list of blocked file types in alphabetical order.
2. Use Notepad to create a file named **TEST.MP3**.

   Ensure that the file extension is MP3, and that a TXT extension is not added.

3. Attempt to upload the MP3 file to the Contoso Partner Portal's Shared Documents document library.

   An error message appears. It indicates that the file has been blocked by an administrator.

Lesson Summary

- When you create a SharePoint web application, you create an IIS site including a folder, a Web.config file, bindings, and a virtual directory. You should manage all configuration by using Central Administration or Windows PowerShell, except for binding a security certificate to a site, which must be performed in IIS Manager on each server in the farm.
- It is recommended that you create all web applications within a single application pool, unless you have a significant requirement for process-level isolation. If you create more than one application pool, your requirements for auditing and access are likely to drive you to create a unique managed account as the identity for each application pool.
- You can encrypt communication between clients and a SharePoint web application by configuring the web application to use the HTTPS protocol, which relies on SSL. You must also add the certificate to the server, and add an SSL binding to the IIS Web site.
- Numerous settings are scoped to a web application, including self-service site collection creation, service application connections, and Recycle Bin settings. If your requirements call for more than one collection of these settings, you will need more than one web application to support those requirements.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 1, "Configure Web Applications." The questions are also available on the companion media in a practice test if you prefer to review them in electronic form.

> ***NOTE*** **ANSWERS**
>
> **Answers to these questions and explanations of why each answer choice is right or wrong are located in the "Answers" section at the end of the book.**

1. You want to enable SSL encryption for a new SharePoint web application. The server farm currently has no IIS Web sites that use SSL. What do you need to do? (Choose all that apply. Each correct answer is a part of the complete solution.)

   A. Add a binding to the IIS site.

   B. Configure the new SharePoint web application to use SSL.

   C. Add a certificate to the SharePoint configuration database.

   D. Add a certificate to the server.

2. You want to distribute email alerts to different SMTP servers based on the site from which the alert originates. Where can you do this?
   - **A.** Site Collection Administration settings on the Site Settings page of each site.
   - **B.** The Configure Outgoing E-Mail Server command on the General Settings page of Central Administration.
   - **C.** The SMTP Relay settings of the SMTP server.
   - **D.** The Outgoing E-Mail Server setting on the Web Application General Settings page.
3. Your information security and compliance requirements state that if a user accidentally deletes an item, the user must be able retrieve the item for 75 days. How can you configure SharePoint to support this requirement?
   - **A.** Configure Recycle Bin Settings for the web application.
   - **B.** Configure permissions so that the user cannot delete items.
   - **C.** Configure information management policy.
   - **D.** Configure a User Policy for the web application.

Lesson 2: Configure Authentication

SharePoint Server 2010 is a distributed application that is logically divided into three tiers: the front-end web server tier, the application server tier, and the back-end database tier. SharePoint can also interact with external systems—for example, by presenting data stored in an external database in a list. Each tier or system is a trusted subsystem, and authentication is required by default. *Authentication* is the process of verifying the identity of a user making a request to an application. The application must be assured that the user is authentic before the system performs *authorization*, which is the process of verifying that the user has permission to make the request, and *personalization*, which determines how the application interacts with the user.

SharePoint 2010 supports numerous methods by which users can be authenticated, including Windows authentication methods such as NTLM or Kerberos, forms-based authentication with methods that use LDAP directories or SQL databases as sources of user credentials and groups, and claims authentication using Security Assertion Markup Language (SAML) tokens. In this lesson, you will master the concepts and procedures related to authentication in SharePoint.

After this lesson, you will be able to:

- Describe classic-mode authentication and identify the authentication provider and methods it supports.
- Configure classic-mode authentication.
- Describe integrated Windows authentication.
- Configure Kerberos authentication.
- Describe additional Windows authentication methods.
- Describe claims-based authentication and identify the authentication providers and methods it supports.
- Configure claims-based authentication using Windows authentication methods.
- Configure forms-based authentication.
- Configure SAML token authentication.
- Convert an upgraded or other web application using Classic Mode Authentication to Claims Based Authentication.

Estimated lesson time: 120 minutes

Configure Anonymous Access

Let’s start our exploration of authentication by detailing the processes by which you can configure anonymous access, so that users can access SharePoint content without validation of the users’ identities. Anonymous access is disabled by default, which provides an additional layer of security because IIS rejects anonymous access requests before they can ever be processed by SharePoint. To configure the level of access that anonymous users have to content, you must manage three settings:

- Anonymous authentication for the web application
- Permissions assigned to anonymous users for sites, lists, and libraries
- Anonymous access restriction policies for the web application's zones

Enable Anonymous Authentication

You can enable anonymous authentication when you create a web application or after creating a web application. To enable anonymous authentication while creating a web application, simply click Yes for the Allow Anonymous setting on the Create New Web Application page, or, in Windows PowerShell, use the *-AllowAnonymous* switch parameter of the *New-SPWebApplication* cmdlet.

ENABLE OR DISABLE ANONYMOUS ACCESS ON AN EXISTING WEB APPLICATION

1. In the Central Administration Quick Launch, click Application Management.
2. In the Web Applications section, click Manage Web Applications.
3. On the Web Applications Management page, click the name of the web application for which you want to enable or disable anonymous access.
4. On the ribbon, click Authentication Providers.
5. On the Authentication Providers page, click the name of the zone for which you want to enable or disable anonymous access. For example, click Default.

   The Edit Authentication page opens.
6. On the Edit Authentication page, select or clear the Enable Anonymous Access check box, and then click Save.

When you enable anonymous access, SharePoint enables anonymous authentication for the IIS Web site.

> ***IMPORTANT*** **USE SHAREPOINT TO MAKE THIS CHANGE**
>
> **As with other IIS Web site settings, you should not make the change directly in IIS Manager. When you make the change by using SharePoint, the web application properties are modified in the configuration database. Therefore, when you add a new server to the farm or restore a web application, the setting is applied correctly to the new IIS Web site.**

When you install the Web Server IIS role, IIS creates the IUSR_*computername* account to authenticate anonymous users in response to a request for web content. The IUSR_*computername* account, where *computername* is the name of the server that is running IIS, gives the user access to resources anonymously under the context of the IUSR account.

NOTE **HOW TO USE ANONYMOUS ACCOUNTS**

You can reset anonymous user access to use any valid Windows account. You can also set up different anonymous accounts for different Web sites, virtual or physical directories, and files.

Grant Permissions to Anonymous Users

Enabling anonymous access for a web application allows anonymous authentication, but it does not authorize anonymous users, in the context of the IUSR account, to access any content. Therefore, it is not enough simply to enable anonymous access for a web application—you must also grant permissions to anonymous users at the site level.

IMPORTANT **CONSIDER THE SECURITY IMPLICATIONS**

Although enabling anonymous authentication at the web application level does not actually allow anonymous users to access content in the web application, it does allow site collection administrators to grant permissions to anonymous users to access content. Therefore, it is recommended that you do not enable anonymous access for a web application until you have considered the security implications of enabling anonymous access.

CONFIGURE ANONYMOUS ACCESS FOR A SITE

1. Click Site Actions, and then click Site Permissions.

   The Permissions page opens.

2. On the ribbon, click Anonymous Access.

   If you are not in the top-level site of a site collection, but rather are in a subsite, and if the top-level site does not allow anonymous access, you will not see the Anonymous Access button on the ribbon. This is because the subsite inherits the permissions from its parent site. Click Stop Inheriting Permissions to block inheritance, and then you can configure anonymous access permissions for the subsite.

   The Anonymous Access page opens.

3. In the Anonymous Users Can Access group, choose one of the following options:

   - **Entire Web Site** Anonymous users can view content on the entire Web site.
   - **Lists And Libraries** Anonymous users can view content in certain lists or libraries.
   - **Nothing** Anonymous users have no access to the site.

4. Click OK.

If you select the Lists And Libraries option, all lists and libraries do not allow anonymous access by default. You must therefore assign anonymous access permissions to specific lists and libraries.

CONFIGURE ANONYMOUS ACCESS TO A LIST OR LIBRARY

1. Navigate to a list or library for which you want to configure anonymous access.
2. On the ribbon, click the List or Library tab.
3. Click the List Permissions or Library Permissions button.

   The Permissions page opens.
4. Click Stop Inheriting Permissions.
5. Click Anonymous Access.

   The Anonymous Access page opens.
6. In the Anonymous Users Can list, select the check boxes for the permissions you want to assign to anonymous users.

   In a document library, anonymous users can, at most, view items. Anonymous users cannot be granted add, edit, or delete item permission.
7. Click OK.

As you've learned, anonymous access involves configuration at both the web application and site levels. You must enable anonymous authentication for the web application, which in turn enables anonymous authentication for the IIS Web site, and then you must specify what content anonymous users can access. You will learn more about configuring permissions and security on sites, lists, and libraries, including the concept of inheritance, in Chapter 4.

> ***NOTE*** **USE SITE DESIGN TO MANAGE ANONYMOUS USERS' ACCESS**
>
> **You should not configure anonymous access at the per-list and library level. It is difficult to manage access at that level, and it is very difficult to provide anonymous users access to those lists and libraries because the home page is not accessible to anonymous users. Therefore, you should strive to design your site structure so that content that should be accessed by anonymous users is in separate sites from content that requires authenticated access, so that you can manage anonymous access at the site level.**

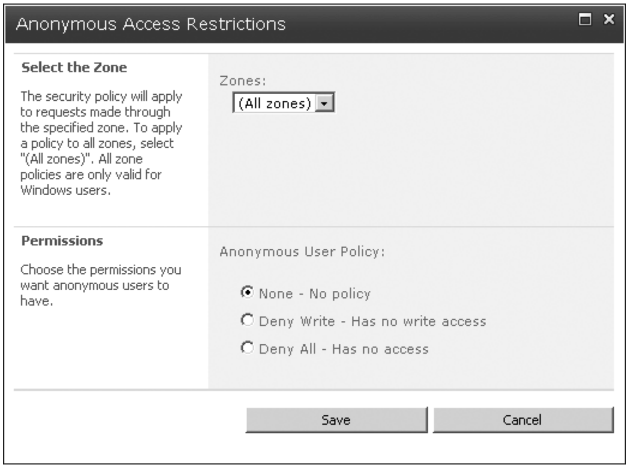
Anonymous Access Restrictions

Farm administrators can enforce permissions related to anonymous access across all sites in a web application by using anonymous access restrictions.

CONFIGURE ANONYMOUS ACCESS RESTRICTIONS

1. On the Web Applications Management page of Central Administration, select the web application that you want to configure.
2. On the ribbon, click Anonymous Policy.

   The Anonymous Access Restrictions page opens, as shown in Figure 3-7.



Anonymous Access Restrictions

Select the Zone

The security policy will apply to requests made through the specified zone. To apply a policy to all zones, select "(All zones)". All zone policies are only valid for Windows users.

Zones: (All zones)

Permissions

Choose the permissions you want anonymous users to have.

Anonymous User Policy:

● None - No policy

○ Deny Write - Has no write access

○ Deny All - Has no access

FIGURE 3-7 Anonymous Access Restrictions

3. In the Zones list, select the zone to which the policy will apply.

   To apply to the policy to all access to the web application, select All Zones.

4. In the Permissions section, click one of the following options:

   - **None** No policy is defined. Anonymous access will be determined by permissions granted to sites, lists, and libraries.
   - **Deny Write** Anonymous users will be unable to modify content. This policy overrides access granted on content within the web application, effectively ensuring that if a site collection administrator has granted any permissions at all to anonymous users, the maximum level of access will be Read.
   - **Deny All** This policy overrides all permissions granted on content within a web application. Anonymous users will not have access.

5. Click Save.

The two policies that override site content permissions are primarily used in the following two scenarios:

- **Temporarily Disable Access** You want to prevent anonymous users temporarily from writing or accessing content, but you do not want to change permissions on content.
- **Restrict Anonymous Access Through A Zone** You want to restrict anonymous access using one zone, or URL, that is otherwise allowed using another zone. For example, if a web application can be accessed using HTTP using the URL *http://partners.contoso.com*, and can also be accessed using HTTPS with the URL *https://partners.contoso.com*, you might want to ensure that anonymous access is only possible using HTTPS. To do this, you must enable anonymous authentication for the web application; then grant permissions to anonymous users on sites, lists, or libraries in the web

application; and then configure anonymous access restrictions to deny all access through the zone associated with the HTTP URL. You will learn more about zones in Lesson 3.

Understand Authentication Types

Authentication—verification of a user's credentials—is performed by a software component called an *authentication provider*. Authentication providers support one or more *authentication methods*. For example, the integrated Windows authentication provider supports both the NTLM and the Negotiate (Kerberos or NTLM) methods. An authentication method defines the protocols and data sources by which the provider performs authentication. In Lesson 1, you learned that there are two types of authentication in SharePoint Server 2010:

- **Classic Mode Authentication** Classic Mode Authentication is the same type of authentication that was used in Microsoft Office SharePoint Server 2007. Classic Mode Authentication uses Windows authentication provider, which relies on Active Directory to authenticate users.
- **Claims Based Authentication** Claims Based Authentication is a new feature in SharePoint 2010. Claims Based Authentication can use the Windows authentication provider—just as can Classic Mode Authentication—as well as Forms Based Authentication (FBA) and SAML token providers. You will learn more about Claims Based Authentication later in this lesson.

Each of the providers supports multiple authentication methods. Table 3-1 summarizes the authentication types, providers, and methods. You will learn about each provider and method later in this lesson.

TABLE 3-1 Authentication Options for SharePoint Web Applications

TYPE	PROVIDER	METHODS
Classic Mode Authentication	Windows	Anonymous, Basic, Digest, NTLM, Negotiate (Kerberos or NTLM)
Claims Based Authentication	Windows	Anonymous, Basic, Digest, NTLM, Negotiate (Kerberos or NTLM)
	FBA	LDAP, SQL database, Other DB, Custom
	SAML	ADFS 2.0, Windows Live ID, Third Party

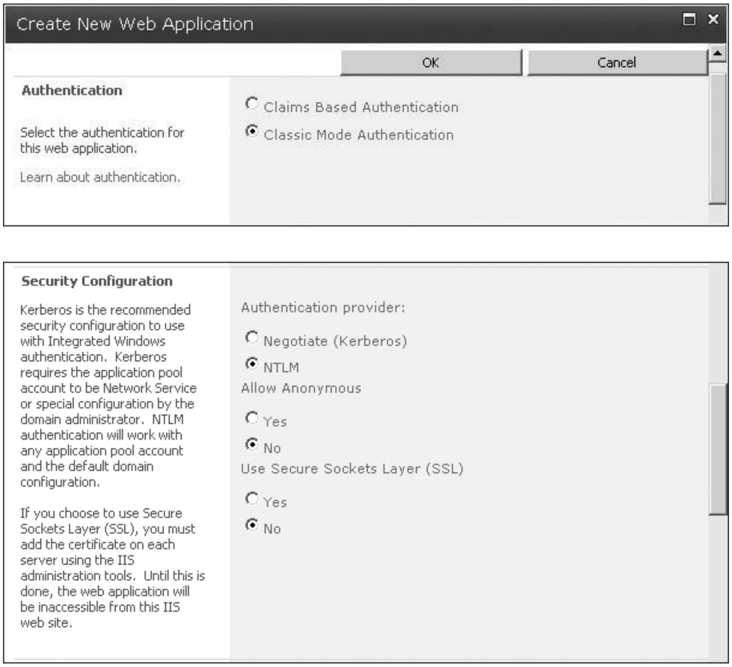
Configure Classic Mode Authentication

Classic Mode Authentication is one of the two types of authentication supported by SharePoint 2010. If you do not require claims, and if you will use only Windows authentication, you can create a web application that uses Classic Mode Authentication.

Classic Mode Authentication supports one authentication provider—Windows—and several methods of Windows authentication: NTLM, Kerberos, Basic, Digest, and Anonymous. You can configure the authentication method when you create the web application or after a web application has been created.

Create a Web Application with Classic Mode Authentication

As you learned in Lesson 1, when you create a web application, you can specify authentication settings on the Create New Web Application page, shown in Figure 3-8.



Authentication

Select the authentication for this web application.

Learn about authentication.

○ Claims Based Authentication
● Classic Mode Authentication

Security Configuration

Kerberos is the recommended security configuration to use with Integrated Windows authentication. Kerberos requires the application pool account to be Network Service or special configuration by the domain administrator. NTLM authentication will work with any application pool account and the default domain configuration.

If you choose to use Secure Sockets Layer (SSL), you must add the certificate on each server using the IIS administration tools. Until this is done, the web application will be inaccessible from this IIS web site.

Authentication provider:
○ Negotiate (Kerberos)
● NTLM

Allow Anonymous
○ Yes
● No

Use Secure Sockets Layer (SSL)
○ Yes
● No

FIGURE 3-8 Classic Mode Authentication settings for a new web application

You can also specify authentication settings when you create a new web application by using Windows PowerShell.

CREATE A NEW WEB APPLICATION WITH CLASSIC MODE AUTHENTICATION

The following example shows the use of the *New-SPWebApplication* cmdlet to create a new web application:

```
New-SPWebApplication -Name <Name> -Port <Port> -HostHeader <HostHeader>
-AuthenticationMethod <AuthenticationMethod> [-AllowAnonymousAccess]
[-SecureSocketsLayer] -URL <URL> -ApplicationPool <ApplicationPool>
-ApplicationPoolAccount <ApplicationPoolAccount> -DatabaseName <DatabaseName>
```

Where:

- *<Name>* is the name of the new web application.
- *<Port>* is the port on which the web application will be created in IIS.
- *<HostHeader>* is the host header, in the format *server.domain.com*.

  Note that the *Get-Help* documentation for the cmdlet states that the format for *<HostHeader>* is *http://server.domain.com*. The documentation is incorrect.
- *<AuthenticationMethod>* is the Windows authentication method, which can be *NTLM* or *Kerberos*.

  If you specify *Kerberos*, it is actually the *Negotiate (Kerberos or NTLM)* method that is used.
- The *-AllowAnonymousAccess* switch parameter, if specified, enables anonymous authentication.
- The *-SecureSocketsLayer* parameter, if specified, enables SSL for the web application.

  As you learned in Lesson 1, you must also use IIS Manager to create the certificate in the server's certificate store and bind the certificate to the IIS Web site.
- *<URL>* is the public URL for the web application's default zone.
- *<ApplicationPool>* is the name of the application pool.
- *<ApplicationPoolAccount>* is the managed account that the application pool will use.

  This parameter is required if you are specifying an *<ApplicationPool>* that does not already exist. Use the *Get-SPManagedAccount* cmdlet as shown in the following example. If the *<ApplicationPool>* already exists, do not include this parameter.
- *<DatabaseName>* is the name for the first content database for the web application.

For example, the following command creates the Contoso partner portal web application with configuration similar to the web application that was created by using Central Administration in Lesson 1:

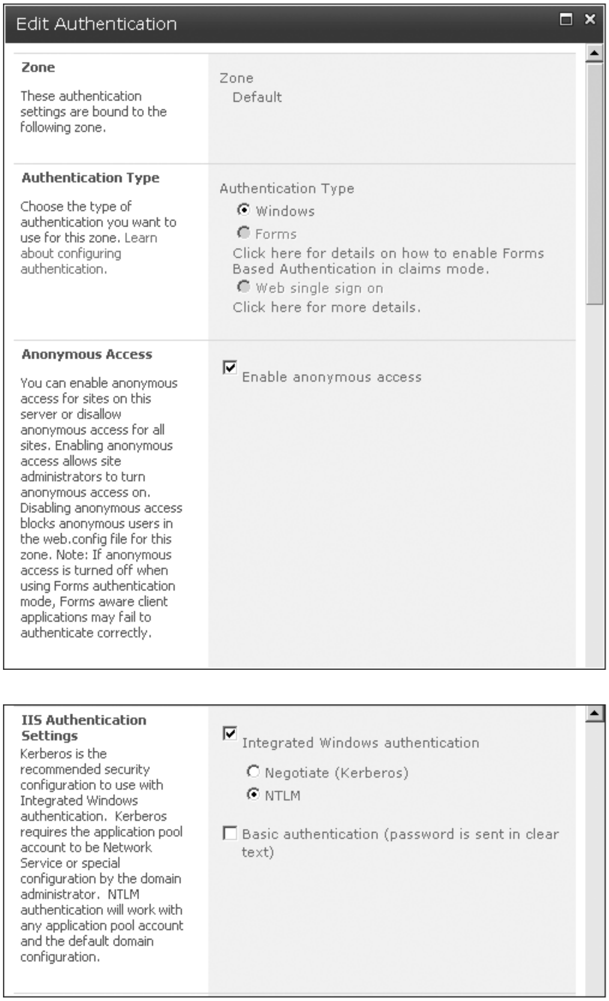
```
New-SPWebApplication -Name "Contoso Partner Portal" -Port 443
-HostHeader "partners.contoso.com" -AuthenticationMethod "NTLM" -SecureSocketsLayer
-URL "https://partners.contoso.com:443" -ApplicationPool "SharePoint Extranet Applications"
–ApplicationPoolAccount (Get-SPManagedAccount "CONTOSO\SP_WebApps")
-DatabaseName "SharePoint_Content_Partners"
```

The command creates a new application pool. If the application pool already exists, you would not include the *-ApplicationPoolAccount* parameter and value.

Configure a Web Application with Classic Mode Authentication

After a web application is created, you can modify authentication settings on the Edit Authentication page, shown in Figure 3-9.

You can access the Edit Authentication page from the Web Applications Management or the Authentication Providers pages of Central Administration.



Edit Authentication

Zone

These authentication settings are bound to the following zone.

Zone
Default

Authentication Type

Choose the type of authentication you want to use for this zone. Learn about configuring authentication.

Authentication Type
● Windows
○ Forms
Click here for details on how to enable Forms Based Authentication in claims mode.
○ Web single sign on
Click here for more details.

Anonymous Access

You can enable anonymous access for sites on this server or disallow anonymous access for all sites. Enabling anonymous access allows site administrators to turn anonymous access on. Disabling anonymous access blocks anonymous users in the web.config file for this zone. Note: If anonymous access is turned off when using Forms authentication mode, Forms aware client applications may fail to authenticate correctly.

☑ Enable anonymous access

IIS Authentication Settings
Kerberos is the recommended security configuration to use with Integrated Windows authentication. Kerberos requires the application pool account to be Network Service or special configuration by the domain administrator. NTLM authentication will work with any application pool account and the default domain configuration.

☑ Integrated Windows authentication
○ Negotiate (Kerberos)
● NTLM

□ Basic authentication (password is sent in clear text)

FIGURE 3-9 Edit Authentication page for Classic Mode Authentication

CONFIGURE AUTHENTICATION SETTINGS FROM THE WEB APPLICATIONS MANAGEMENT PAGE

1. In the Central Administration Quick Launch, click Application Management.
2. In the Web Applications section, click Manage Web Applications.
3. Select the web application that you want to modify.
4. On the ribbon, click Authentication Providers.
5. Click the link to the zone that you want to modify.

   By default, each new web application has a single zone, called Default. You will learn more about zones later in this chapter.

   The Edit Authentication page appears.
6. Make your changes, and then click Save.

CONFIGURE AUTHENTICATION SETTINGS FROM THE AUTHENTICATION PROVIDERS PAGE

1. In the Central Administration Quick Launch, click Security.
2. In the Web Applications section, click Specify Authentication Providers.
3. Click the Web Application picker to select the web application that you want to modify.
4. Click the link to the zone that you want to modify.
5. On the Edit Authentication page, make your changes, and then click Save.

Windows Authentication Methods

Windows authentication is available in both classic-mode and claims-based authentication. However, when a web application is using classic-mode authentication, only the Windows authentication provider is supported.

Windows authentication supports the following authentication methods:

- Integrated Windows authentication, which can use either NT LAN Manager (NTLM) or Negotiate (Kerberos or NTLM) authentication methods.
- Basic.
- Anonymous.
- Digest.
- Client certificates.

NTLM

Introduced more than a decade ago, NTLM is the most established form of authentication in Microsoft products.

When a user logs on to his or her computer, the user is prompted for a user name and password. The user name is sent to the domain controller, but the password is never sent over the network. Instead, there is an encrypted challenge/response protocol through which a hash of the password is passed through a one-way hashing algorithm (the challenge) by both the client and the domain controller. The client sends the result (the response) to the domain controller. If the result matches what the domain controller obtained as a result, the password entered by the user must have been correct, and the user is authenticated.

It gets more complicated when a user connects to a server, such as a SharePoint server. If the SharePoint server is a member server—not a domain controller—it has no way of knowing the user's password. So when the user connects to the server, the server has to pass the authentication request up to a domain controller. If the domain controller responds to the server that the user is valid, the authentication succeeds.

Although NTLM is not the most efficient authentication method, and is slightly less secure than Kerberos, it is often chosen as the authentication method for SharePoint web applications because it is easy to set up—it just works, out of the box.

NOTE **NTLM AND PROXY SERVERS**

NTLM can fail when traversing proxy servers because it is connection-based and proxies don't always sustain connections.

Kerberos

Kerberos is the default authentication method for Windows clients and servers in an Active Directory domain.

Kerberos uses a process that involves encrypted tickets to verify authenticity. When a user logs on and authenticates with the domain, the domain controller's Key Distribution Center (KDC) issues the user a *ticket-granting-ticket* (TGT) that effectively represents that the user has been authenticated. For the lifetime of the TGT (10 hours by default), the user no longer needs to be authenticated.

When the user wants to connect to a service, such as a SharePoint web application that uses Kerberos authentication, the client application returns to a domain controller's KDC, presents the TGT, which confirms that the client has already been authenticated, and requests from a domain controller a service ticket for the specific service to which the client will connect. The client then goes to the service and presents the service ticket.

Because the entire process is encrypted with keys unique to each player (the client, the service, and the domain), the service is able to examine the service ticket and determine that it is being presented by an authenticated client. The service ticket contains the client's identity and roles; the session is established.

This is a very simplistic—but accurate—explanation of Kerberos. If you are interested in more details about Kerberos, see the resources listed in the "Additional Resources About Kerberos Authentication" section.

One of the benefits of Kerberos is that when the client connects to the service, the service does not have to round-trip the authentication to a domain controller, as in NTLM. Instead, the client's ticket for the service ensures the client has been authenticated. This results in improved authentication performance for Kerberos as compared to NTLM.

Another benefit is that Kerberos tickets can be *delegated*—that is, forwarded or *proxied* between tiers. For example, a client connecting to a website provides a Kerberos ticket, and the website can pass the ticket to a back-end data source that can authenticate the user for data access. The web tier does not need to know the user's password to achieve this "double-hop" authentication. The web tier also does not need permissions to the back-end data source—it is all done using the authentication of the client.

To secure this "double-hop" authentication, you can configure Kerberos *constrained delegation*. Constrained delegation restricts which services are allowed to delegate user credentials by specifying, for each application pool or service, the services to which a Kerberos ticket can be forwarded. If you choose to configure constrained delegation, you should configure and test Kerberos with unconstrained delegation and resolve any issues you might encounter prior to configuring constrained delegation.

Kerberos is considered by many organizations to be a preferable authentication mechanism because of the following advantages:

- More secure than NTLM. Kerberos protocols ensure mutual authentication, which prevents what are called "man in the middle" attacks whereby a rogue service could pretend to be a domain controller and intercept authentication requests from clients. Kerberos tickets also contain timestamps that reduce the likelihood of "replay attacks" in which an authentication token can be intercepted and used at a later date for malicious purposes.
- More scalable than NTLM. Kerberos supports authentication across trusted realms and, because it is an industry standard, is supported by platforms other than Windows.
- Supports delegation. Delegation and constrained delegation were explained earlier. Delegation allows a service to impersonate a user without knowing the user's password. Windows Server 2003 and later support constrained delegation as well, which adds a further level of security to the implementation of Kerberos in a Windows enterprise.
- Reduced load on domain controllers. Kerberos requires fewer trips to a domain controller for authentication than NTLM.

The disadvantage of Kerberos is that it requires additional steps to configure. You will learn the fundamental steps to configure Kerberos later in this lesson.

Negotiate (Kerberos or NTLM)

To use Kerberos authentication for a SharePoint web application, select the Negotiate (Kerberos or NTLM) authentication method. The Negotiate authentication method attempts to use Kerberos authentication. But if Kerberos authentication is not supported in the deployed environment, or if the client does not support Kerberos, authentication falls back to NTLM.

NOTE **KERBEROS-ONLY ISN'T AN OPTION**

There is no option to use Kerberos as the only authentication method for Windows authentication.

IIS passes the Negotiate security header when Windows Integrated authentication is used to authenticate client requests. The Negotiate security header lets clients select between Kerberos authentication and NTLM authentication. The Negotiate process selects Kerberos authentication unless one of the following conditions is true:

- One of the systems that is involved in the authentication cannot use Kerberos authentication.
- The calling application does not provide enough information to use Kerberos authentication.

If the Negotiate process cannot use the Kerberos protocol, the Negotiate process selects the NTLM protocol.

Configure Kerberos Authentication

To configure Kerberos authentication, you must use *service principal names* or *SPNs* for your SharePoint services, web applications, and SQL Server. This section will summarize the process. For a detailed walkthrough of creating a SharePoint farm with Kerberos authentication, see the TechNet article "Configure Kerberos authentication (SharePoint Server 2010)" at *http://technet.microsoft.com/en-us/library/ee806870.aspx*.

Earlier in this lesson, you learned that when a client wants to connect to a web application that uses Kerberos authentication, the client requests a service ticket from a domain controller's KDC. The request indicates the service to which the client will connect by specifying the service's *service principal name* or *SPN*.

The SPN is made up of three components. The first is the *service class* for the request, which is always *HTTP*—the *HTTP* service class includes both the HTTP and HTTPS protocols. The second is the host name, and the third is the port (if not port 80) of the web application. Together, these three components comprise the SPN of the web application.

For example, a request to *http://intranet.contoso.com* on port 80 equates to an SPN of *HTTP/intranet.contoso.com*. Note that the SPN syntax uses a single forward slash between the service class and host name portions of the name. A request to *https://partners.contoso.com* on port 443 equates to an SPN of *HTTP/partners.contoso.com:443*. A request to *http://sp2010-wfe1:9999* for Central Administration equates to an SPN of *HTTP/sp2010-wfe1:9999*.

A security principal—a user or computer account in Active Directory—can have one or more associated SPNs. SPNs are an attribute of security principals in Active Directory. That means an account, such as an application pool account, can have multiple SPNs—for example, both *HTTP/intranet.contoso.com* and *HTTP/partners.contoso.com:443*.

When a domain controller's KDC receives the service ticket request from a client, it looks up the requested SPN. The KDC then creates a *session key* for the service and encrypts the session key with the password of the account with which the SPN is associated. The KDC issues a service ticket, containing the session key, to the client. The client presents the service ticket to the service. The service, which knows its own password, decrypts the session key and authentication is complete.

If a client submits a service ticket request for an SPN that does not exist in the identity store, no service ticket can be established and the client will throw an *access denied* error.

For this reason, each component of a SharePoint infrastructure that uses Kerberos authentication requires at least one SPN. For example, the intranet web application app pool account must have an SPN of *HTTP/intranet.contoso.com*.

Configure Service Principal Names for a Service or Application Pool

Note that it is the app pool—not the server—that is associated with the SPN because the app pool is the security context within which the service—the web application in this case—is running. It also makes sense if you consider that each SPN can be associated with only one security principal, and if a web app is load balanced—running on several servers—it is the one app pool account that is constant across all servers and therefore must have the SPN.

For each web application, you should assign two SPNs—one with the fully qualified domain name for the service, and one with the NetBIOS name of the service. Therefore, the intranet web application pool account should also be assigned an SPN of HTTP/intranet.

In many environments, a single application pool can be used by multiple web applications. The app pool account should be given a pair of SPNs for each of its web applications that use Kerberos authentication.

You can use ADSI Edit to add SPNs to an account. To configure an SPN for a service or application pool account, you must have domain administrative permissions or a delegation to modify the *servicePrincipalName* property.

CONFIGURE SPNS USING ADSI EDIT

1. Start ADSI Edit.
2. In the console tree, right-click ADSI Edit, and then click Connect To.
3. In the Connection Settings dialog box, click OK.
4. In the console tree, expand Default Naming Context, then expand the domain, and then expand the nodes representing the OUs in which the account exists. Click the OU in which the account exists.
5. In the Details pane, right-click the service or application pool account, and then click Properties to open the Properties dialog box.
6. In the Attributes list, double-click servicePrincipalName to open the Multi-Valued String Editor dialog box.
7. In the Value To Add field, type the SPN, and then click Add.

   Repeat step 7 for additional SPNs. Remember that an app pool account should have two SPNs, in the form HTTP/site.domain.com and HTTP/site, for each web application that uses Kerberos authentication in the app pool. Remember also to add the port number if the site runs on a port other than port 80—for example, HTTP/site.contoso.com:9999 and HTTP/site:9999.
8. Click OK twice.

You can also use the command-line tool Setspn.exe to add SPNs to an account.

CONFIGURE SPNS USING SETSPN

The following example shows the use of the SetSPN command to add an SPN to an account:

```
setspn <domain\user> -s <SPN>
```

Where:

- *<domain\user>* identifies the security principal to which you want to add an SPN.
- <SPN> is the service principal name that you want to add.

For example, to add SPNs for the intranet web application to the app pool account, you can type the following commands:

```
setspn CONTOSO\SP_WebApps –s HTTP/intranet.contoso.com
setspn CONTOSO\SP_WebApps –s HTTP/intranet
```

The most useful facts to know about SetSPN are the following:

- The *-s* parameter adds an SPN to an account after verifying that a duplicate SPN does not already exist. Duplicate SPNs can cause authentication problems, and it is recommended that you use each SPN only once in a forest. The *-s* parameter is new in Windows Server 2008. Previously, you used the *-a* switch, which adds an SPN but does not check for duplicates. It is recommended that you use *-s* now that it is available, but some documentation might refer to *-a*.
- The *-L* switch lists the SPNs associated with a specific user or computer account.
- The *-Q* switch lists the accounts associated with a specific SPN.

You can type **setspn.exe /?** for more information about SetSPN.

Configure Service Principal Names for SQL Server

To configure Kerberos authentication for SQL Server, you will need to add SPNs to the SQL Server service account—for example, CONTOSO\SVC_SQL. By default, SQL Server communication is over port 1433, so the two SPNs for a SQL Server running on a server named SQLSERVER01 would be the following:

- MSSQLSvc/sqlserver01:1433
- MSSQLSvc/sqlserver01.contoso.com:1433

✓ Quick Check

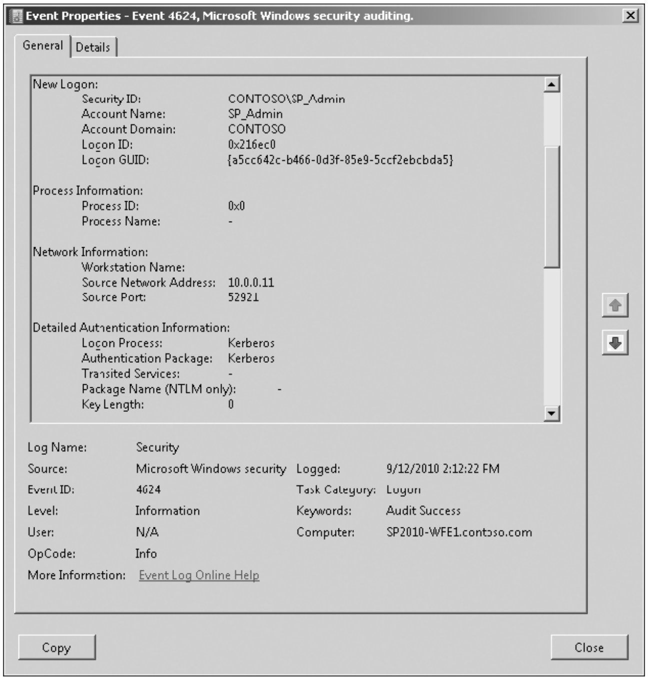
- **When you configure Kerberos authentication for a web application, what change must be made in Active Directory?**

Quick Check Answer

- **An SPN must be added to the user account of the application pool identity.**

Verify Kerberos Authentication to a Web Application

After you have configured your environment to support Kerberos authentication for a web application, you can validate that Kerberos is being used to authenticate a user. By opening the website you will generate an entry in the Windows Security event log. Examine the Security event log on the web server. The audit event generated by the user's logon will show the security ID of the user and the Logon Process, which should be Kerberos, as shown in Figure 3-10.

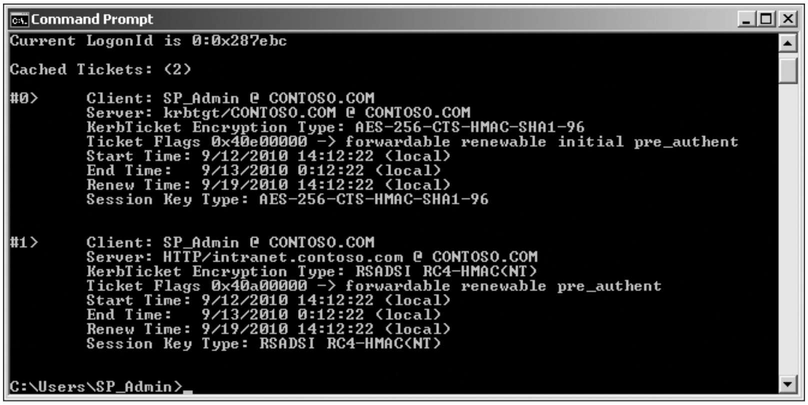

Event Properties - Event 4624, Microsoft Windows security auditing.
General
Details
Copy
Close

```
New Logon:
          Security ID:               CONTOSO\SP_Admin
          Account Name:              SP_Admin
          Account Domain:            CONTOSO
          Logon ID:                  0x216ec0
          Logon GUID:                {a5cc642c-b466-0d3f-85e9-5ccf2ebcbda5}

Process Information:
          Process ID:                0x0
          Process Name:              -

Network Information:
          Workstation Name:
          Source Network Address:    10.0.0.11
          Source Port:               52921

Detailed Authentication Information:
          Logon Process:             Kerberos
          Authentication Package:    Kerberos
          Transited Services:        -
          Package Name (NTLM only):          -
          Key Length:                0
```

```
Log Name:          Security
Source:            Microsoft Windows security   Logged:          9/12/2010 2:12:22 PM
Event ID:          4624                         Task Category:   Logon
Level:             Information                  Keywords:        Audit Success
User:              N/A                          Computer:        SP2010-WFE1.contoso.com
OpCode:            Info
More Information:  Event Log Online Help
```

FIGURE 3-10 A Kerberos logon event

Alternately, you can use Klist.exe. KList is a command-line utility included in the default installation of Windows Server 2008 and Windows Server 2008 R2 which can be used to list and purge Kerberos tickets on a given computer. To verify Kerberos with KList, open the website on a client, and then use KList on the client to enumerate its tickets. You will see the ticket with the SPN of the web application, as shown in Figure 3-11.



```
Current LogonId is 0:0x287ebc

Cached Tickets: (2)

#0>     Client: SP_Admin @ CONTOSO.COM
        Server: krbtgt/CONTOSO.COM @ CONTOSO.COM
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
        Start Time: 9/12/2010 14:12:22 (local)
        End Time:   9/13/2010 0:12:22 (local)
        Renew Time: 9/19/2010 14:12:22 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96


#1>     Client: SP_Admin @ CONTOSO.COM
        Server: HTTP/intranet.contoso.com @ CONTOSO.COM
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0x40a00000 -> forwardable renewable pre_authent
        Start Time: 9/12/2010 14:12:22 (local)
        End Time:   9/13/2010 0:12:22 (local)
        Renew Time: 9/19/2010 14:12:22 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)


C:\Users\SP_Admin>_
```

FIGURE 3-11 KList enumeration of Kerberos tickets

Connect to Back-End Systems

In Chapter 8, "Implementing Enterprise Service Applications," you will learn to connect SharePoint to back-end data systems. For example, you might use Excel Services to present information in a database outside of SharePoint, or you might use Business Data Connectivity Services to create a SharePoint list that displays items stored in another system.

There are several ways to configure connection to a back-end or external system, including the Secure Store Service, the Claims To Windows Token Service, and Kerberos delegation. This chapter has examined authentication to a web application. If you need to support authentication to back end systems, be sure to read the details in Chapter 8.

Additional Resources about Kerberos Authentication

If you want to configure your entire SharePoint environment to support Kerberos, there are numerous additional steps to perform. You can learn more about configuring Kerberos by reading the following articles:

- "Configure Kerberos authentication (SharePoint Server 2010)," *http://technet.microsoft.com/en-us/library/ee806870.aspx*.
- "Kerberos (Windows Server 2008 and Windows Server 2008 R2 Technical Library)," *http://technet.microsoft.com/en-us/library/cc753173(WS.10).aspx*.
- "Kerberos Authentication Technical Reference (Windows Security Collection)," *http://technet.microsoft.com/en-us/library/cc739058(WS.10).aspx*.
- "Windows Authentication," *http://technet.microsoft.com/en-us/library/cc755284(WS.10).aspx*.
- "Kerberos Explained," *http://technet.microsoft.com/en-us/library/bb742516.aspx*.
- "How to use SPNs when you configure Web applications that are hosted on Internet Information Services," *http://support.microsoft.com/kb/929650*.
- "Setspn," *http://technet.microsoft.com/en-us/library/cc731241(WS.10).aspx*.
- "Microsoft Kerberos (Windows)," *http://msdn.microsoft.com/library/aa378747*.
- "Ask the Directory Services Team: Kerberos for the Busy Admin," *http://blogs.technet.com/b/askds/archive/2008/03/06/kerberos-for-the-busy-admin.aspx*.
- "Configure Kerberos authentication for the claims to Windows token service (SharePoint Server 2010)," *http://technet.microsoft.com/en-us/library/ee806887.aspx*.

You can also download the white paper "Configuring Kerberos authentication for SharePoint 2010 Products" at *http://technet.microsoft.com/en-us/library/ff829837.aspx*.

Additional Windows Authentication Methods

Although NTLM or Negotiate (Kerberos or NTLM) are the most commonly used authentication methods, Classic Mode Windows authentication also supports anonymous, basic, and digest authentication methods.

Anonymous

Anonymous authentication enables users to connect to a web application without providing credentials. You can enable anonymous authentication on either the Create New Web Application or Edit Authentication pages.

Anonymous authentication was detailed earlier in this lesson. You learned that anonymous authentication does not provide anonymous users with permission to content within a web application. Anonymous access must be granted at the securable object. You can grant anonymous users permission to an entire site or to specific lists and libraries. You can then restrict access at the web application by applying anonymous access restriction policies, which override permissions.

Basic

Like Integrated Windows authentication, Basic authentication relies on a set of credentials for the user in Active Directory. However, Basic authentication enables a web browser to submit credentials when making an HTTP request, and the credentials are sent as Base64 clear text, unencrypted, to the server. Credentials used in Basic authentication are easily compromised. If you choose to use Basic authentication, you should always enable Secure Sockets Layer (SSL) encryption.

> ***NOTE*** **WHEN NTLM AND KERBEROS AREN'T SUPPORTED**
>
> **Certain browsers and connection scenarios, such as users behind some proxy servers, will not support NTLM and Kerberos. In these cases, you might need to resort to Basic authentication.**

You cannot select Basic authentication when you create a SharePoint web application. Instead, you must do so after creating the web application.

ENABLE BASIC AUTHENTICATION

1. In the Central Administration Quick Launch, click Application Management.
2. In the Web Applications section, click Manage Web Applications.

   The Web Applications Management page opens.
3. Click the name of the web application for which you want to enable or disable anonymous access.
4. On the ribbon, click Authentication Providers.

   The Authentication Providers page opens.
5. Click the name of the zone for which you want to enable or disable anonymous access. For example, click Default.

   The Edit Authentication page opens.
6. In the IIS Authentication Settings section, shown earlier in Figure 3-9, select the Basic Authentication check box.
7. Click Save.

8. Close the Authentication Providers page.
9. Start Command Prompt using the Run As Administrator option, and then type **IISRESET**.

If you select Negotiate (Kerberos) and Basic Authentication, clients should attempt authentication in the following order: Kerberos, NTLM, Basic authentication.

Digest

Digest authentication provides the same functionality as Basic authentication, but with increased security. User credentials are encrypted instead of being sent over the network in plaintext. User credentials are sent as an MD5 message digest in which the original user name and password cannot be deciphered. Digest authentication uses a challenge/response protocol that requires the authentication requestor to present valid credentials in response to a challenge from the server. To authenticate against the server, the client has to supply an MD5 message digest in a response that contains a shared secret password string.

Digest authentication for SharePoint is not particularly common. To implement digest authentication, you will have to select Windows authentication in Central Administration, then configure the IIS Web site for Digest authentication.

Understand Claims Based Authentication

Consider the following summary of Claims Based Authentication:

> *Claims Based Authentication is a flexible framework based on Security Assertion Markup Language (SAML) tokens, and built on the Windows Identity Foundation (WIF). Tokens contain assertions about a user's identity that are generated by trusted authentication providers, which include Windows authentication—just as in Classic Mode Authentication—as well as Forms Based Authentication (FBA) and standard SAML tokens issued by trusted authorities such as Windows Live ID or Active Directory Federated Services 2.0 (ADFS 2.0). By extending the reach of trusted authentication providers, Claims Based Authentication enables authentication across Windows-based systems and systems that are not Windows based.*
> *Claims Based Authentication becomes particularly powerful when tokens contain other attributes of a user, such as demographic or organizational information. These attributes can originate within the user's organization, other organizations, or the Internet.*

Doesn't that sound really complex? Don't give up; read on.

Review Authentication in a Windows Domain

If you are not already familiar with Claims Based Authentication, the preceding description of Claims Based Authentication may sound complex. But the concepts related to Claims Based Authentication can be pretty straightforward if you start from the perspective of an authentication scheme that you already understand: authentication within a Windows domain. Let's review the basics of Windows authentication as a basis from which to understand Claims Based Authentication.

When you require access to a system, such as a file server, the system must know who you are before you can be granted access to resources. It would not be manageable to maintain a list of user names and passwords on each system. Therefore, you create a Windows domain by implementing Active Directory Domain Services (AD DS). Within a domain, all systems trust the authentication mechanism of the domain—Kerberos—to validate the identity of a user. So, when you access a file server, the file server does not have to authenticate you. Instead, you bring to the server a Kerberos service ticket that identifies you. The ticket has been created using processes that include encryption using keys known only by the server and the domain. So the server knows that the service ticket is valid. It looks at the ticket to know who you are. The server accepts the ticket's assertion as to your identity because the server trusts the source of the ticket—the AD DS domain's Kerberos KDC. The server does not have to perform authentication—it trusts an external authentication provider.

The Kerberos service ticket does not just identify you. It also contains a list of your domain security group memberships. Again, because the ticket comes from a trusted authority, the server uses that list of groups. The server builds a token that contains your identity—your user account's security identifier, or *SID*—and the SIDs of the groups to which you belong. The token is then used by the local security subsystem to determine whether you have access to a file by comparing the SIDs on the file's access control list to the SIDs in your token. This security token represents you to the local server.

In the past, when a developer wanted to create a secure website, the developer had to build an authentication component. With SharePoint, in Classic Mode Authentication, your Windows security token is translated into an object that represents you within SharePoint—an object called an *SPUser* object. You can think of the *SPUser* object in a SharePoint web application as the conceptual equivalent of your Windows security token—it represents you during your interactions with the web application.

Claims Authentication to a SharePoint Web Application

A *claim* is a set of *assertions*—information about a user. At the most basic conceptual level, a Kerberos service ticket is a claim that, among other things, asserts the identity and group memberships of a user. When you access a SharePoint web application that uses Claims Based Authentication, the web application accepts a claim and translates that claim into the *SPUser* object which, as you know, represents you during your interactions with the web application.

This is the first difference between Classic Mode Authentication and Claims Based Authentication. In Classic Mode Authentication, the web application relies on IIS to pass your Windows security token to the web application. In Claims Based Authentication, the web application relies on the farm's Security Token Service (STS) to deliver a token that contains claims, including claims about your identity.

In Classic Mode Authentication, IIS relies on Active Directory to actually perform authentication. IIS can receive credentials using several methods, including NTLM, Kerberos, Basic, and Digest. In the case of NTLM, Basic, and Digest authentication, IIS authenticates the credentials against Active Directory. In the case of Kerberos authentication, the service ticket contains credentials that have already been authenticated.

In Claims Based Authentication, the STS also does not actually perform authentication. Instead, it relies on a trusted authority to do so. The authority can be Active Directory, or it can be one of a number of other authentication providers. If the Claims Based Application uses the Windows authentication provider, the STS performs essentially the same function as IIS does in Classic Mode Authentication. If Kerberos is available, the service ticket is processed and turned into a set of claims about the user's identity and group memberships. If NTLM, Basic, or Digest authentication are used, the STS authenticates the credentials against Active Directory and then the NT token is translated into a set of claims about the user's identity and group memberships.

The resulting claims are provided to the web application as a token which, as you know, is translated into an *SPUser* object within the web application.

By this point in the discussion, you should understand that a component called an *STS* is doing the work of building tokens that contain claims. You should also have an understanding that if only Windows authentication is used, there is conceptually little difference between Classic Mode Authentication and Claims Based Authentication. But the story is just beginning.

What if you want to make a web application available to partners, but you do not want to add accounts for partner users to your AD DS domain? In the past, a web developer would have to write a custom component to authenticate users and to administer user identities. Now, however, you can use the Forms Based Authentication provider to authenticate users against credentials stored in AD DS; in Active Directory Lightweight Directory Services (AD LDS); in a database such as a SQL Server database; or in an LDAP data store such as Novell eDirectory, Novell Directory Services (NDS), or Sun ONE. Or you can use SAML to authenticate users against credentials stored in Active Directory Federated Services 2.0 (ADFS 2.0), by Windows Live ID, or by a custom trusted source.

Claims Based Authentication thus allows SharePoint web applications to be extended to more diverse sets of users, across domains, forests, and non-Windows environments. You can change the authentication provider or the methods of authentication without having to change the web application itself.

Trust

How are claims actually built? When you attempt to access a web application that uses Claims Based Authentication, you are transparently redirected to a sign-in page for the STS, at which you are authenticated. In some cases, such as Windows authentication, you might never even see this transaction if your browser's security settings are configured to authenticate you silently to trusted sites, and if the website is in a trusted zone. The STS authenticates you and provides a token to your browser. Your browser then returns to the original website, submits the token, and the web application then knows who you are.

But if the browser is submitting a token with assertions about your identity, how does the web application know that those assertions come from a trusted source, and that you have not fabricated a false token containing erroneous statements about who you are?

The process uses a series of standards called *WS-** standards that effectively ensure that the token can be used by the web application. To make a long, complicated story very short, the web application has been configured to trust the STS. The trust involves the exchange of certificates that are used to encrypt the token. If the web application is able to decrypt the token with the shared secret, it knows that the token must have been generated by the trusted STS.

Trust is at the heart of any security system. In an AD DS domain, each component of Windows trusts the local security subsystem, which in turn trusts the domain, which in turn trusts other domains in the forest, and that trust can then be extended to other domains or forests. In SharePoint, all web applications and services in a farm trust the Security Token Service of the farm.

Trust and Claims Based Authentication in Action

When you sign in to a Microsoft website such as Microsoft TechNet or MSDN with your Windows Live ID, you are authenticated using Claims Based Authentication. These websites—which do not run on SharePoint—trust Windows Live ID to verify your identity. They redirect you to a Windows Live ID sign-in control for authentication. Windows Live ID issues your browser an encrypted token that contains assertions as to your identity and other attributes. Your browser passes this token to the website, which can decrypt the token.

Claims

When a claim is presented to a web application, the claim contains assertions about the user's identity. It also can contain claims about the user's group memberships. Each of the authentication methods available in Claims Based Authentication can provide the STS with an enumeration of the user's group memberships, which are added to the claim.

But a claim can provide more than just user and group information, and this is where claims become particularly valuable. Let's assume that you want to be able to send email messages to users from a website. How do you determine a user's email address? You can build and maintain a local database of user email addresses, but in an AD DS domain that information is stored in Active Directory, and so a local database would have to be kept in synch with changes made in Active Directory. Or you can add code to query Active Directory each time an email address is needed. Both approaches require additional work by the website developer.

A claim can include a user's email address or any other attribute of the user, such as the user's manager or the manager's email address, department, job title, age, or gender. Because the claims are presented by the user to the web application, the web application does not need to maintain local copies of the attributes, nor does it need to go look up the attributes in an external source. Instead, the STS is configured to collect the attributes and to create claims.

Claims Based Authentication thus reduces the burden on applications themselves to maintain or look up information about users. Attributes in claims can be used for a variety of purposes. You can assign permissions to content that are based on a claim. For example, you can specify that users must have a job title of Vice President or higher to access content. You can also use claims to look up users. For example, if you want to assign a task to a user, but you can only remember the user's manager, the picker control can expose the manager attribute of users who belong to the site. Developers are particularly excited about the possibilities that are presented now that SharePoint 2010 supports claims.

Federation

Let's now assume that certain content in a web application can only be accessed by users who are employees of your company, Contoso, or of a partner company, Litware. How do you make this work? It would be a burden to have duplicate copies of all Litware user accounts in your AD DS domain or in a separate database, and to keep changes in synch. It would be much easier to simply rely on the administrators at Litware to maintain their user accounts, and to trust the authentication performed by Litware.

With Windows domains, you could configure a trust whereby the Contoso domain trusts the Litware domain. However, firewalls can often prevent trusts from being correctly established and maintained, and many organizations have policies that forbid Windows trusts to external organizations.

Claims Based Authentication supports *federation*, which extends the concepts of trust and claims to third parties. For example, you can configure ADFS 2.0 to authenticate users against both domains, without requiring a trust. You then configure SharePoint's STS to trust the STS exposed by ADFS 2.0. From a terminology perspective, SharePoint's STS becomes the *relying party STS* (RP-STS) and the STS of ADFS 2.0 becomes the *identity provider STS* (IP-STS).

When a user attempts to access a website, the user is redirected to the IP STS for authentication. The token issued by the IP STS (ADFS 2.0 in this example) is then presented to the RP STS (SharePoint's STS in this example), which can augment the token with additional claims before giving the client the token that is then submitted to the web application.

Another example of federated identity is Windows Live ID authentication. You can configure SharePoint's STS to trust tokens issued by Windows Live ID, just as some Microsoft sites do.

Claims Authentication

Claims authentication is built on the Windows Identity Foundation (WIF). WIF is a set of .NET Framework classes that are used to implement claims-based identity. Claims authentication relies on standards such as WS-Federation, WS-Trust, and protocols such as SAML. Claims Based Authentication thus enables you to extend both authentication (identification) and the collection of informational attributes about a user to sources beyond your domain.

It's not important that you, as an IT Pro, master all of the concepts, standards, and protocols, and the tools used to create code used to leverage claims. However, you must be able to configure SharePoint to support claims authentication. In the next section, you will learn how to configure SharePoint for Windows-Claims, Forms-Claims, and SAML-Claims authentication.

Now, test yourself: Return to the beginning of this section, "Understand Claims Based Authentication," and read the summary once again. Does it make sense now?

MORE INFO **CLAIMS AUTHENTICATION AND THE WIF**

The following article provide additional detail regarding claims authentication and the WIF: "Claims-based Identity for Windows: An Introduction to Active Directory Federation Services 2.0, Windows CardSpace 2.0, and Windows Identity Foundation (white paper)," at *http://go.microsoft.com/fwlink/?LinkId=198942*. You can also visit the Windows Identity Foundation home page at *http://go.microsoft.com/fwlink/?LinkId=198943*.

Configure Windows-Claims Authentication

Now that you understand Claims Based Authentication, we can turn our attention to the procedures required to create and configure web applications that use Claims Based Authentication. First, we will explore creating a web application that uses the Windows authentication provider for Claims Based Authentication—*Windows-Claims* authentication. You can create a web application that uses Claims Based Authentication by using Central Administration or Windows PowerShell.

CREATE A WEB APPLICATION WITH WINDOWS-CLAIMS AUTHENTICATION USING CENTRAL ADMINISTRATION

1. In the Central Administration Quick Launch, click Application Management.
2. In the Web Applications section, click Manage Web Applications.
3. On the ribbon, click New to open the Create New Web Application page.
4. In the Authentication section, click Claims Based Authentication.
5. In the Claims Authentication Types section, select the Enable Windows Authentication check box.
6. If you want to use NTLM or Kerberos as the authentication method, select the Integrated Windows Authentication check box. Then, in the drop-down menu, select Negotiate (Kerberos) or NTLM.
7. If you want users' credentials to be sent over a network in a nonencrypted form, select the Basic Authentication (Password Is Sent In Clear Text) check box. If you use basic authentication, ensure that SSL is enabled; otherwise, the credentials can be intercepted by a malicious user.
8. Configure other settings for the new web application. See Lesson 1 for more information about the settings you can configure when creating a web application.

NOTE WHEN ONLY WINDOWS AUTHENTICATION IS USED

The Sign In Page URL setting is not used if Claims Based Authentication uses only Windows authentication.

To create a web application by using Windows PowerShell, you must first create an object that represents the authentication provider by using the *New-SPAuthenticationProvider* cmdlet.

CREATE AN AUTHENTICATION PROVIDER USING WINDOWS POWERSHELL

The following example shows the use of the *New-SPAuthenticationProvider* cmdlet to create a new Windows authentication provider.

```
$ap = New-SPAuthenticationProvider [-UseWindowsIntegratedAuthentication]
[-DisableKerberos | DisableKerberos:$false]
[-UseBasicAuthentication] [-AllowAnonymous]
```

Where:

- The *-UseWindowsIntegratedAuthentication* switch parameter specifies that the authentication provider will be Windows.
- The *-DisableKerberos* switch parameter, if specified, disables Kerberos authentication. The authentication provider uses NTLM only.

  The *-DisableKerberos:$false* syntax enables authentication.

NOTE AN RTM BUG

In the RTM version of SharePoint, there is a bug in the class that initializes *DisableKerberos* to *true*. Therefore, the switch parameter does not work as documented. You must use *-DisableKerberos:$false* to enable Kerberos. It is likely that in a future release, Microsoft will either correct the behavior so that Kerberos is enabled by default and the documentation will be correct, or add a switch parameter that enables Kerberos.

- The *-UseBasicAuthentication* switch parameter, if specified, enables Basic authentication.

After you create the object representing the authentication provider, you pass the object as the *-AuthenticationProvider* parameter to the *New-SPWebApplication* cmdlet.

CREATE A WEB APPLICATION WITH CLAIMS BASED AUTHENTICATION USING WINDOWS POWERSHELL

The following example shows the use of the *New-SPWebApplication* cmdlet to create a new web application:

```
New-SPWebApplication -Name <Name> -Port <Port> -HostHeader <HostHeader>
-AuthenticationProvider <AuthenticationProvider> [-AllowAnonymousAccess]
[-SecureSocketsLayer] -URL <URL> -ApplicationPool <ApplicationPool>
-ApplicationPoolAccount <ApplicationPoolAccount> -DatabaseName <DatabaseName>
```

Where:

- *<Name>* is the name of the new web application.
- *<Port>* is the port on which the web application will be created in IIS.
- *<HostHeader>* is the host header, in the format *server.domain.com*.

  Note that the *Get-Help* documentation for the cmdlet states that the format for *<HostHeader>* is *http://server.domain.com*. The documentation is incorrect.
- *<AuthenticationProvider>* is an object representing an authentication provider.

  Use the *New-SPAuthenticationProvider* cmdlet to create an object representing an authentication provider, as described earlier.
- The *-AllowAnonymousAccess* switch parameter, if specified, enables anonymous authentication.
- The *-SecureSocketsLayer* parameter, if specified, enables SSL for the web application.

  As you learned in Lesson 1, you must also use IIS Manager to create the certificate in the server's certificate store and bind the certificate to the IIS Web site.
- *<URL>* is the public URL for the web application's default zone.
- *<ApplicationPool>* is the name of the application pool.
- *<ApplicationPoolAccount>* is the managed account that the application pool will use.

  This parameter is required if you are specifying an *<ApplicationPool>* that does not already exist. Use the *Get-SPManagedAccount* cmdlet as shown in the following example. If the *<ApplicationPool>* already exists, do not include this parameter.
- *<DatabaseName>* is the name for the first content database for the web application.

For example, the following command creates the partner portal web application with configuration similar to the web application that was created by using Central Administration in Lesson 1, but with Claims Based Authentication. A Windows authentication provider is constructed that uses only NTLM—Kerberos is disabled—and passed as the authentication provider for the new web application.

```
$ap = New-SPAuthenticationProvider –UseWindowsIntegratedAuthentication
   –DisableKerberos
New-SPWebApplication -Name "Contoso Partner Portal" -Port 443
   -HostHeader "partners.contoso.com" –AuthenticationProvider $ap –SecureSocketsLayer
   -URL "https://partners.contoso.com:443"
   -ApplicationPool "SharePoint Extranet Applications"
   –ApplicationPoolAccount (Get-SPManagedAccount "CONTOSO\SP_WebApps")
   -DatabaseName "SharePoint_Content_Partners"
```

After you have created the web application, create a site collection. When you create a site collection, you must specify the primary site collection administrator. You can use Central Administration or the *New-SPSite* cmdlet, as described in Chapter 2.

Configure Forms Based Authentication

Forms Based Authentication (FBA) is an identity management system that is based on ASP.NET membership and role provider authentication.

If an unauthenticated user attempts to access a web application using FBA, the user is redirected to a logon form, with which the user submits credentials. The credentials are authenticated against an identity store, which can be AD DS; a database such as a SQL Server database; or an LDAP data store such as Active Directory Lightweight Directory Services (AD LDS), Novell eDirectory, Novell Directory Services (NDS), or Sun ONE.

SharePoint Server 2010 uses the standard ASP.NET membership provider interface to authenticate the user, and the standard ASP.NET role manager interface to gather group information about the user. Each ASP.NET role is treated as a domain group by the authorization process in SharePoint Server 2010. The resulting information about the user is converted into claims by the STS, thus FBA is also called *Forms-Claims* authentication.

To configure FBA, you must manage the following settings, each of which is detailed later in this section:

- **The web application's authentication mode** The web application must use Claims Based Authentication. In SharePoint Server 2010, Forms-Based Authentication is available only when you use Claims Based Authentication.

> ***NOTE*** **FBA AND CLASSIC MODE AUTHENTICATION**
>
> **If you upgrade a SharePoint 2007 web application that uses FBA, the upgraded web application is configured to use Classic Mode Authentication, and FBA will not function. You must convert the web application to Claims Based Authentication, as described later in this lesson.**

- **The config file of the Security Token Service (STS) Application** As you have learned, the STS generates and manages claims tokens. The STS uses the FBA authentication provider to authenticate the user on behalf of the relying party—either the web application or Central Administration. It is the STS that actually performs the authentication, so it must know which provider and data source to use.
- **The Web.config file of the web application's IIS site** You must register the membership provider and role manager in the Web.config file. Although the web application does not perform authentication, it does perform other tasks against the users and roles that are provided. For example, when you assign a task or grant permissions to a user or group, the People Picker control must know the sources from which it can find users.

NOTE **REGISTERING THE ROLE MANAGER IS NOW REQUIRED**

In SharePoint 2007, you were required to register only the membership provider; registering the role manager was optional In SharePoint 2010, you are required to register both.

- **The Web.config file of the Central Administration IIS site** If you want to manage membership users or roles from the SharePoint Central Administration web site, you must register the membership provider and the role manager in the Web.config file of the Central Administration website as well. For example, you might want assign a user as the primary site collection administrator. If Central Administration does not know how to locate and interact with the FBA provider, it will be unable to locate the user and add the user as the site collection owner.
- **Access to the database against which users are authenticated** The user database must allow SharePoint to authenticate and look up users.

Create a Web Application with Forms-Claims Authentication

Forms Based Authentication is available only to an application that uses Claims Based Authentication. You can create the web application by using Central Administration or Windows PowerShell.

CREATE A WEB APPLICATION WITH FORMS-CLAIMS AUTHENTICATION USING CENTRAL ADMINISTRATION

1. In the Central Administration Quick Launch, click Application Management.
2. In the Web Applications section, click Manage Web Applications.
3. On the ribbon, click New to open the Create New Web Application page.
4. In the Authentication section, click Claims Based Authentication.
5. In the Claims Authentication Types section, select the Enable Forms Based Authentication (FBA) check box.

NOTE **IF YOU DON'T ENABLE FBA**

If you clear the Enable Windows Authentication check box, a warning message appears: "If Windows authentication is not selected on any Zone of this Web application, crawling for this Web application will be disabled." The crawler requires a zone that uses Windows authentication to index SharePoint content. You will learn more about zones in Lesson 3, and about crawling in Chapter 7, "Administering SharePoint Search."

6. Enter the membership provider name and the role manager name in the boxes. You will learn more about these settings later in this lesson.
7. In the Sign In Page URL section, do one of the following:

    - Click Default Sign In Page if you want users to be redirected to a SharePoint's default sign-in page for claims-based authentication.
    - Click Custom Sign In Page, and then type the URL of the customized sign-in page to which you want users redirected for Claims Based Authentication for the web application.

8. Configure other settings for the new web application. See Lesson 1 for more information about the settings you can configure when creating a web application.

CREATE A WEB APPLICATION WITH FORMS-CLAIMS AUTHENTICATION USING WINDOWS POWERSHELL

The following example shows the use of the *New-SPAuthenticationProvider* cmdlet to create a new Forms Based Authentication authentication provider.

```
$ap = New-SPAuthenticationProvider –ASPNETMembershipProvider <MembershipProviderName>
-ASPNETRoleProviderName <RoleProviderName>
```

Where:

- *<MembershipProviderName>* specifies the name of the membership provider. The name must be the valid name of an ASP.NET provider defined in the Web.config file of the application.
- *<RoleProviderName>* specifies the name of the membership provider. The name must be the valid name of an ASP.NET membership provider defined in the Web.config file of the application.

After you create the object representing the authentication provider, you pass the object as the *-AuthenticationProvider* parameter to the *New-SPWebApplication* cmdlet, as described earlier in the procedure, "Create a Web Application with Claims Based Authentication Using Windows PowerShell."

For example, the following command creates the partner portal web application with configuration similar to the web application that was created by using Central Administration in Lesson 1, but with Forms-Claims authentication.

```
$ap = New-SPAuthenticationProvider -ASPNETMembershipProvider "MyMembershipProvider"
   -ASPNETRoleProviderName "MyRoleManager"
New-SPWebApplication -Name "Contoso Partner Portal"
   -Port 443 -HostHeader "partners.contoso.com" –AuthenticationProvider $ap
   –SecureSocketsLayer
   -URL "https://partners.contoso.com:443"
   -ApplicationPool "SharePoint Extranet Applications"
   -ApplicationPoolAccount (Get-SPManagedAccount "CONTOSO\SP_WebApps")
   -DatabaseName "SharePoint_Content_Partners"
```

Configure Web.config Files

After you have successfully created a web application that uses Claims Based Authentication, you must manually configure the specifics of the authentication provider by modifying the configuration file of the IIS site, Web.config.

This section details the configuration of Web.config. Do not be concerned if it sounds confusing. It is! In the practice for this lesson, you will configure FBA. The practice will thus give you hands-on experience modifying Web.config files, and a chance to review and reinforce the details presented here.

The following sample illustrates the structure of the Web.config file, focused on the elements that are important to configure for FBA.

```
<configuration>
...
  <SharePoint>
    <PeoplePickerWildcards>
      <clear />
      <add key="AspNetSqlMembershipProvider" value="%" />
      <add key="MyMembershipProvider" value="*"/>
      <add key="MyRoleManager" value="*"/>
    </PeoplePickerWildcards>
  </SharePoint>
  <connectionStrings>
    <add name="MyConnectionString" [define the connection] />
  </connectionStrings>
  <system.web>
  ...
    <membership>
      <providers>
       <add name="MyMembershipProvider" [define the membership provider] />
      </providers>
    </membership>
    <roleManager>
      <providers>
        <add name="MyRoleManager" [define the role manager] />
      </providers>
    </roleManager>
    ...
  </system.web>
  ...
</configuration>
```

As you work with the Web.config files, keep the following tips in mind:

- An element can have values—for example, you might see <roleManager enabled="true">, where enabled="true" is a value of the element roleManager. Do not change the existing values within a tag unless you have been instructed to do so.
- If an element does not exist, you can create it. For example, the Web.config file of the STS site does not have a *<system.web>* element. You can create it in the relative position shown in the preceding example.
- Names are used to link configuration elements. The name of the membership provider and role manager you configure for the SharePoint web application must have a matching entry in the *<membership><providers>* and the *<roleManager><providers>* elements, respectively. If the providers connect to a data source, the connection string must be registered in the *<connectionStrings>* element with a name that matches the

name that is used in the definition of the providers in the *<membership><providers>* and the *<roleManager><providers>* elements. Keep close tabs on the names that you use to ensure all of the configuration elements are properly associated.

The *<connectionStrings>* element shown in the preceding example defines a connection to the identity store, typically a SQL database or LDAP directory. The following example registers a connection string named *MySQLDatabase* that connects to the database named *aspnetdb* on the server named *SP2010-WFE1.contoso.com* using integrated authentication:

```
<connectionStrings>
    <add name="MySQLDatabase"
         connectionString="server=SP2010-WFE1.contoso.com;
         database=aspnetdb;
         Integrated Security=SSPI" />
</connectionStrings>
```

You define a membership provider to connect the web application to a provider—a software component that performs the authentication. The most popular out-of-box FBA membership provider is the ASP.NET SQLMembershipProvider provider, which uses a membership database on a SQL server that contains information about users, groups (roles), and profile attributes. When you define the provider, you specify values of the provider that determine its exact behavior. For example, you pass a connection string to the provider so that the provider knows which data source to work with.

The following example configures a SQLMembershipProvider named *MyMembershipProvider*, and instructs the provider to access the data source referred to by the connection string named *MySQLDatabase*:

```
<add name="MyMembershipProvider"
     connectionStringName="MySQLDatabase"
     applicationName="/"
     type="System.Web.Security.SqlMembershipProvider, System.Web, Version=2.0.3600.0,
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
     description="Stores and retrieves roles from SQL Server"
     passwordAttemptWindow="5"
     enablePasswordRetrieval="false"
     enablePasswordReset="false"
     requiresQuestionAndAnswer="true"
     requiresUniqueEmail="true"
     passwordFormat="Hashed"/>
```

NOTE **HOW TO USE THE TYPE TAG**

The *type* tag must be on one line. In this example, it is shown breaking across lines for formatting purposes only.

Next, you define a role manager. The role manager, also called a role provider, is the software component responsible for identifying the roles, or groups, to which a user belongs. The most popular out-of-box role manager is the ASP.NET SQLRoleProvider, which works against the same membership database as the SQLMembershipProvider, but is responsible for determining the user's group memberships.

The following example configures a SQLRoleProvider named *MyRoleManager* that uses the same data source referred to by the connection string name, *MySQLDatabase*:

```
<add name="MyRoleManager"
     connectionStringName="MySQLDatabase"
     applicationName="/"
     type="System.Web.Security.SqlRoleProvider, System.Web, Version=2.0.3600.0,
     Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
     description="Stores and retrieves roles from SQL Server"/>
```

The definition of the membership provider and role manager determines how authentication is performed for a web application. The following examples define the membership provider named *MyMembershipProvider* and the role manager named *MyRoleManager* to use the *LDAPMembershipProvider* and *LDAPRoleProvider* providers, respectively, with the *contoso.com* domain as the data source. If these providers were registered in the *Web.config* files instead of the identically named providers shown previously, the web application would authenticate against the domain instead of against a SQL membership database.

```
<configuration>
...
  <system.web>
  ...
    <membership>
      <providers>
     <add name="MyMembershipProvider"
         type="Microsoft.Office.Server.Security.LdapMembershipProvider, Microsoft
         .Office.Server, Version=14.0.0.0, Culture=neutral,
         PublicKeyToken=71e9bce111e9429c"
         server="contoso.com"
         port="389"
         useSSL="false"
         userDNAttribute="distinguishedName"
         userNameAttribute="sAMAccountName"
         userContainer="DC=contoso,DC=com"
         userObjectClass="person"
         userFilter="(ObjectClass=person)"
         scope="Subtree"
         otherRequiredUserAttributes="sn,givenname,cn" />
      </providers>
    </membership>
    <roleManager>
      <providers>
      <add name="MyRoleManager"
         type="Microsoft.Office.Server.Security.LdapRoleProvider, Microsoft.Office
         .Server, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c"
         server="contoso.com"
         port="389"
         useSSL="false"
         groupContainer="DC=contoso,DC=com"
         groupNameAttribute="cn"
         groupNameAlternateSearchAttribute="samAccountName"
         groupMemberAttribute="member"
         userNameAttribute="sAMAccountName"
         dnAttribute="distinguishedName"
         groupFilter="(ObjectClass=group)"
         userFilter="(ObjectClass=person)"
         scope="Subtree" />
      </providers>
    </roleManager>
    ...
  </system.web>
  ...
</configuration>
```

In the case of the LDAP providers, you do not use or need a *<connectionStrings>* element, because all of the configuration of the connection is defined in the provider itself. Each provider, data source, and scenario requires slightly different configuration.

Hopefully you can see how the architecture of Claims Based Authentication allows the configuration of authentication to be separated fully from the web application itself. All you have to do is change the configuration of the IIS Web site's Web.config file and you can completely change the source of authentication.

One additional change to Web.config is often overlooked: the *<PeoplePickerWildards>* element. If you don't configure this element, the People Picker control will only accept exact matches. The People Picker is used in many places in SharePoint, including assigning tasks and assigning security permissions. There is already a *<PeoplePickerWildcards>* element in the Web.config file—you must simply define the wildcard that is used for your membership and role providers, as shown in the following example:

```
<PeoplePickerWildcards>
    <clear />
    <add key="AspNetSqlMembershipProvider" value="%" />
    <add key="MyMembershipProvider" value="*"/>
    <add key="MyRoleManager" value="*"/>
</PeoplePickerWildcards>
```

Two wildcard definitions are added to the default list—one for the membership provider and one for the role manager. The *key* tag must match the name of the provider that you configured by using Central Administration or the *New-SPAuthenticationProvider* cmdlet. For LDAP data sources, the *value* should be an asterisk (*). For SQL data sources, the *value* should be a percent symbol (%).

To properly configure SharePoint for Forms Based Authentication for a web application, you must configure three Web.config files:

- The Web.config file of the forms-based authentication claims-based web application
- The Web.config file of the Central Administration Web application
- The Web.config file of the Security Token Service

EXAM TIP

Remember the three Web.config files that must be changed to successfully configure Forms Based Authentication for a web application.

Assign Permissions to the User Database

SharePoint will authenticate users against the associated directory, which can be a database such as a SQL database, an LDAP directory such as an AD DS domain or an instance of AD LDS, or a custom provider. SharePoint will also use the directory to look up users, such as when you use the People Picker control to grant users or groups permissions, or to assign a task. For these reasons, the database must allow access by the application pool identities used by Central Administration, by the web application, by the Security Token Service Application, and by services.

The exact permissions required will vary based on your environment and the provider and database that you use for FBA. For example, in the environment that is built by this training kit, if you use a SQL database with the ASP.NET SQLMembershipProvider and SQLRoleProvider providers, the SQL database must allow access by the application pool identity of the web application (*SP_WebApps*, for example), the SharePoint farm service account (*SP_Farm*), which is used by Central Administration, and by the SharePoint service applications app pool identity (*SP_ServiceApps*), which is used by the Security Token Service Application and other services.

Validate Configuration

You can test and verify that your configuration is successful by performing the following actions:

- In Central Administration, create a test site collection in the site. When you configure the Site Collection Administrator, click the Browse button. Search for a user in the Select People And Groups dialog box. Search both with an exact match of the user name, which tests the *membership* provider configuration and by typing only the first few characters, which tests the *PeoplePickerWildcards* configuration.
- Sign in to the web application by using credentials in the FBA provider. Be certain to close any existing connections to the web application before doing so, so that any cached connections are purged. After the website has rendered, test *PeoplePickerWildcards* by adding a user from the membership provider and a group from the role manager to the site's Visitors group.

Create a Site Collection for a Claims Based Authentication Web App

As mentioned earlier, you can use Central Administration to create a site collection. You can also use the *New-SPSite* cmdlet, as described in Chapter 2, but an additional step is required to specify a site collection administrator in the -*OwnerAlias* attribute. If a web application uses Windows authentication, you can simply specify the user name—for example, CONTOSO\SP_Admin. With FBA and SAML authentication providers, you must pass the user as an *SPClaimsPrincipal* object. To do this, you must first create the *SPClaimsPrincipal* object, as shown in the following example:

```
$cp = New-SPClaimsPrincipal -Identity "<MembershipProvider>:<SiteOwner>"
-IdentityType FormsUser
```

Where:

- *<MembershipProvider>* is the name of your membership provider
- *<SiteOwner>* is the user name of the user that you want to assign as the site collection owner.

You then pass the object to the *New-SPSite* cmdlet, as in the following example:

```
$cp = New-SPClaimsPrincipal -Identity "MyMembershipProvider:SiteAdministrator"
   -IdentityType FormsUser
New-SPSite -Url "https://partners.contoso.com" -Name "Contoso Partner Portal"
   -OwnerAlias $cp -Template "STS#0"
```

Configure SAML Token Authentication

SAML token-based authentication allows SharePoint web applications to accept claims of identity that are authenticated from an STS other than SharePoint's STS. For example, you might configure a SharePoint web application to use Active Directory Federated Service 2.0 (AD FS 2.0) for authentication.

SAML token-based administration is the most generic and standards-based implementation of a claims-based environment. Earlier in this lesson, you learned how such an environment works: SharePoint is the relying party STS (RP-STS) and the external STS is the identity provider STS (IP-STS). The IP-STS authenticates the user against the user directory associated with the IP-STS, and then issues a token with claims about the user. The IP-STS is the conceptual equivalent of a domain controller in a claims-based environment.

Let's follow an example. A common scenario is that two organizations want to collaborate together on a project, but each organization wants to be in full control of the user accounts in its Windows domain. AD FS is a federated authentication service, so it can be configured to use multiple mechanisms of authentication. Instead of creating a trust between each organization's Windows domain, AD FS is configured to authenticate users against each domain, and to generate a security token for the user that can be used by the collaborative environment's web application. The domains are the authentication provider, but AD FS is the IP-STS. Users sign in to AD FS and AD FS issues a signed SAML token with claims about the user's identity. The RP-STS trusts AD-FS.

Tokens can include any number of claims about a user, such as a user name and groups the user belongs to, as well as descriptive attributes. The party application receives the SAML token and uses the claims inside to decide whether to grant the client access to the requested resource. Therefore, one of the claims in the token must uniquely identify the user: this is called the *identity claim*. The IP-STS does not have to create the identity claim with the user name that is submitted when the user logs on to the IP-STS. For example, AD FS does not have to create the identity claim with a user's domain user name. The IP-STS can instead create the identity claim using another unique identifier. Many implementations of claims use the email address attribute as the identity claim. The RP-STS must know which claim is guaranteed to be unique for tokens created by the IP-STS.

For this reason, configuration of a claims environment using SAML token-based authentication requires cooperation between the administrators of the RP-STS and IP-STS. The following elements must be coordinated:

- In SharePoint 2010 products, each web application that is configured to use a SAML provider is added to the IP-STS server as a separate RP-STS entry. This task is performed by the owner of the IP-STS. Each web application is identified as a *realm*, which is simply the URL namespace associated with the relying party web application, such as *https://portal.contoso.com*.
- Only the owner of the IP-STS knows which value in the token will always be unique per user and therefore can be relied upon as the identity claim. That information must be communicated to the owner of the IP-STS.
- Tokens will be signed using a certificate generated by the IP-STS. That certificate must be transferred from the IP-STS to the RP-STS.

Implementing SAML token-based authentication with SharePoint 2010 products involves the following processes:

1. Export the token-signing certificate from the IP-STS. This certificate is known as the ImportTrustCertificate.
2. Copy the certificate to a server computer in the SharePoint Server 2010 farm.
3. Define the claim that will be used as the unique identifier of the user. Identifying the unique identifier for the user is part of the claims-mapping process. Claims mapping is performed by using Windows PowerShell.
4. Define additional claims mappings. Define other values in the token that will be used by the RP-STS. For example, many tokens include a value that specifies user roles that can be used to permission resources in the SharePoint Server 2010 farm. All claims from an incoming token that do not have a mapping will be discarded.
5. Create a new authentication provider by using Windows PowerShell. This process creates the SPTrustedIdentityTokenIssuer.

   During this process, you submit the ImportTrustCertificate, the identity claim mapping, and additional claim mappings. You must also create and specify a realm—the URL namespace that is associated with the first SharePoint web applications that you are configuring for SAML token-based authentication.

   After the SPTrustedIdentityTokenIssuer is created, you can create and add more realms for additional SharePoint web applications. This is how you configure multiple web applications to use the same SPTrustedIdentityTokenIssuer.
6. For each realm that is added to the SPTrustedIdentityTokenIssuer, you must create an RP-STS entry on the IP-STS.
7. Create a new SharePoint web application and configure it to use the newly created authentication provider. The authentication provider will appear as an option in Central Administration when claims mode is selected for the web application.

You can configure multiple SAML token-based authentication providers. However, you can only use a token-signing certificate once in a farm. All providers that are configured will appear as options in Central Administration. Claims from different trusted STS environments will not conflict.

If you are implementing SAML token-based authentication with a partner company and your own environment includes an IP-STS, we recommend that you work with the administrator of your internal claims environment to establish a trust relationship from your internal IP-STS to the partner STS. The result is a type of chain of trust and authentication. This approach does not require adding an additional authentication provider to your SharePoint Server 2010 farm. It also allows your claims administrators to manage the whole claims environment.

NOTE **SECURITY AND PERFORMANCE ISSUES**

If you use SAML token-based authentication with AD FS on a SharePoint Server 2010 farm that has multiple web servers in a load-balanced configuration, the performance and functionality of client web-page views can be affected. When AD FS provides the authentication token to the client, that token is submitted to SharePoint Server 2010 for each permission-restricted page element. If the load-balanced solution is not using affinity, each secured element is authenticated to more than one SharePoint Server 2010 server, which might result in rejection of the token. After the token is rejected, SharePoint Server 2010 redirects the client to reauthenticate back to the AD FS server. After this occurs, an AD FS server might reject multiple requests that are made in a short time period. This behavior is by design, to protect against a denial of service attack. If performance is adversely affected or pages do not load completely, consider setting network load balancing to single affinity. This isolates the requests for SAML tokens to a single web server.

Multiple Authentication Providers

In SharePoint 2007, if you wanted users to authenticate to a web application using both Windows authentication and Forms Based Authentication, you were required to extend the web application to a second zone. A zone is a URL namespace through which a web application can be accessed. You could then configure one zone to use Windows authentication, such as *http://extranet.contoso.com*, and another zone to use Forms Based Authentication, such as *https://partners.contoso.com*.

In SharePoint 2010, this is no longer necessary. If a web application is configured for Claims Based Authentication, you can use multiple authentication providers in a single zone. You will learn more about zones in Lesson 3.

Choose an Authentication Type

As you've learned, the default authentication type is Classic Mode Authentication, which supports only the Windows authentication provider and its methods, NTLM and Kerberos, as well as the less regularly used Basic and Digest authentication provided by IIS. When you upgrade a web application, it is upgraded to Classic Mode Authentication.

When you create a new web application, the default is Classic Mode Authentication. If you will use FBA or SAML token-based authentication, you must choose Claims Based Authentication. If you will use only Windows authentication, you can choose either Classic Mode Authentication or Claims Based Authentication.

Although many resources recommend that you use Claims Based Authentication by default for all new web applications, it is important that you test the functionality of the web application in a lab environment before deploying it in production. Depending on the scenario, Claims Based Authentication might not be the best choice. Claims Based Authentication is a new feature in SharePoint Server 2010, and the community is only now learning the nuances of its implementation. Search the Internet for known issues related to Claims Based Authentication, such as the following:

- Custom code might need to be updated. Web Parts or other custom code that relies on or uses Windows identities will have to be updated. If the custom code uses Windows identities, use Classic Mode Authentication until the code is updated.
- Search alerts are currently not supported with claims-based authentication.
- There are problems using the audiences feature with some authentication providers.
- LDAP environments can be implemented by using either forms-based authentication or SAML token-based authentication. We recommend that you use forms-based authentication because it is less complex. However, if the environment supports WS-Federation 1.1 and SAML Token 1.1, SAML is recommended. Profile synchronization is not supported with LDAP providers that are not associated with ADFS 2.0.

Convert Web Applications to Claims Authentication

If you create a web application with Classic Mode Authentication, you can convert the web application to Claims Based Authentication. This is also important if you upgrade a SharePoint 2007 web application that uses Forms Based Authentication to SharePoint 2010. By default, an upgraded application is configured for Classic Mode Authentication. Classic Mode Authentication does not support FBA, so the application will not be accessible by FBA users until it is converted to Claims Based Authentication.

Before you convert to Claims Based Authentication, you should be aware of considerations related to Claims Based Authentication, as discussed in the previous section.

CONVERT A WEB APPLICATION TO CLAIMS BASED AUTHENTICATION

In SharePoint 2010 Management Shell, type the following:

```
$w = Get-SPWebApplication "http://<WebApplicationURL>/"
$w.UseClaimsAuthentication = 1
$w.Update()
$w.ProvisionGlobally()
```

Where:

- *<WebApplicationURL>* is the URL of the web application that you want to convert to Claims Based Authentication.

After converting the web application, you must migrate users and permissions to account for the new authentication scheme.

MIGRATE USERS AND PERMISSIONS

In SharePoint 2010 Management Shell, type the following:

```
$w = Get-SPWebApplication "http://<WebApplicationURL>/"
$w.MigrateUsers(True)
```

Where:

- *<WebApplicationURL>* is the URL of the web application for which you want to migrate users and permissions.

This process can take quite some time to complete. Be sure to test it in a lab environment so that you can budget appropriate service windows within which to perform the migration in the production environment.

> ***BEST PRACTICE*** **TESTING A CONVERTED WEB APPLICATION**
>
> **You cannot convert a web application from Claims Based Authentication to Classic Mode Authentication. Therefore, you must be certain to test the full functionality of a converted web application in a lab environment before converting the production web application. Validate the functionality of both user and administrative tasks. Also, back up the web application prior to converting to Claims Based Authentication.**

> ***MORE INFO*** **AUTHENTICATION METHODS**
>
> **The following article provides additional details regarding authentication methods: "Plan authentication methods (SharePoint Server 2010)" at *http://technet.microsoft.com/en-us/library/cc262350.aspx*.**

PRACTICE Configure Authentication

Practices are designed to guide you through important procedures. The instructions in the Training Kit are high-level instructions that will challenge you to think carefully and to apply the procedures that are covered in this lesson, and elsewhere in the Training Kit. If you need assistance, consult the detailed, step-by-step instructions in the Practice Answers on the companion media.

In this practice, you will configure authentication for the Contoso Partner Portal web application. First, you will enable anonymous access. Then, you will re-create the application configured to use Claims Based Authentication. You will create a database of users as an identity store for the SQLMembershipProvider, and you will configure Forms Based Authentication for the web application.

Prepare for the Practice

Before you perform this practice, you must ensure that your lab environment has been built according to the instructions found in the Introduction to this Training Kit. You must also have performed the practice in Lesson 1 of this chapter. If you are currently logged on to SP2010-WFE1, log off before beginning the exercises.

EXERCISE 1 Configure Anonymous Access

In this exercise, you configure anonymous access to the Contoso Partner Portal that you created in the practice of Lesson 1.

1. Log on to SP2010-WFE1 as **CONTOSO\SP_Admin** with the password **Pa$$w0rd**.
2. In Central Administration, enable anonymous authentication for the Contoso Partner Portal web application.
3. Open a new tab of Internet Explorer, and then browse to ***https://partners.contoso.com***.

   An error page opens: *There is a problem with this website's security certificate*. Continue to the website.

   The site is loaded, compiled, and cached for first-time access, and then the site opens.

   If an error appears, refresh the page. It is possible that the client timed out while the site was being loaded by IIS.
4. Enable anonymous access to the entire website.
5. Close the tab of Internet Explorer that displays the Partners site, start a new instance of Internet Explorer, and then browse to ***https://partners.contoso.com***.

   You must use a new instance to clear the cache of the authenticated sign-in.
6. Observe that the Welcome control in the upper-right corner of the page reads, "Sign In."

   You are not yet authenticated to the site.
7. Click Site Actions, and then observe that you do not have access to administrative pages as an anonymous user.
8. Close the instance of Internet Explorer that displays the Partners site.

EXERCISE 2 Delete a Web Application

In this exercise, you delete the Contoso Partner Portal site. In the following exercises, you will re-create the application so that it uses Claims Based Authentication.

- Delete the Contoso Partner Portal site. Be sure to delete the content databases and the IIS sites.

EXERCISE 3 Create a Web Application with Claims Based Authentication

In this exercise, you create a web application for collaboration with partners of Contoso. The web application will use Claims Based Authentication with the Windows authentication provider.

- Use Central Administration to create a web application collaboration with partners. Use the following specifications and guidance:
  - Authentication: Claims Based Authentication
  - Name: Contoso Partner Portal
  - Port: 443
  - Host header: partners.contoso.com
  - Authentication provider: Negotiate (Kerberos)
  - Anonymous authentication: No
  - Secure Sockets Layer (SSL): Yes
  - URL: *https://partners.contoso.com:443*
  - Application pool: SharePoint Extranet Applications
  - Application identity: CONTOSO\SP_WebApps
  - Content database name: SharePoint_Content_Partners

EXERCISE 4 Create a Site Collection Using Central Administration

In this exercise, you use Central Administration to create a site collection at the root of the new web application.

1. Use Central Administration to create a site collection. Use the following specifications and guidance:
   - Web application: *https://partners.contoso.com*
   - Title: Contoso Partner Portal
   - Description: Sites for collaboration with partners
   - URL: *https://partners.contoso.com/*
   - Template: Team Site
   - Primary site collection administrator: CONTOSO\SP_Admin
2. Open a new tab of Internet Explorer, and then browse to ***https://partners.contoso.com***.

   An error page opens: *There is a problem with this website's security certificate*. Continue to the website.

   The site is loaded, compiled, and cached for first-time access, and then the site opens.

   If an error appears, refresh the page. It is possible that the client timed out while the site was being loaded by IIS.

NOTE **FOR THE REMAINING EXERCISES…**

Because the configuration of Forms Based Authentication is complex and highly dependent on your environment and the membership and role providers you are using, the remaining exercises in this practice give you the full steps to configure FBA for the Contoso Partner Portal web application using a SQL database of users and the ASP.NET SQLMembershipProvider and SQLRoleProvider providers. Unlike other exercises in this lesson, the steps are identical to the answers in the Practice Answers on the Companion Media.

EXERCISE 5 Configure Forms Based Authentication

In this exercise, you configure the Contoso Partner Portal to use Forms Based Authentication. You then examine the Web.config files for the web application, Central Administration, and STS, in which you will identify the locations that you must modify to configure the authentication provider.

You will not complete the configuration of FBA because that requires establishing an external database of users. The Suggested Practice at the end of this chapter gives you the opportunity to complete the process.

1. In the Central Administration, configure the Contoso Partner Portal web application using the following specifications and guidance:
   - Authentication provider #1: Integrated Windows authentication with the Negotiate (NTLM or Kerberos) method
   - Authentication provider #2: Forms Based Authentication
   - ASP.NET membership provider: MyMembershipProvider
   - ASP.NET role manager: MyRoleManager

EXERCISE 6 Configure Web.config Files

In this exercise, you examine the Web.config files for the web application, Central Administration, and STS, in which you will identify the locations that you must modify to configure the authentication provider for FBA. You will not actually complete the configuration of FBA because FBA requires an external database of users. The Suggested Practice, at the end of this chapter, gives you the opportunity to complete the process.

1. Open the Web.config file of the Contoso Partner Portal IIS Web site.
2. Search for the <*connectionStrings*> element.

   This file does not have an existing <*connectionStrings*> element. In a production environment, a Web.config file might already have a <*connectionStrings*> element, in which case you would simply register the new connection string by inserting an <*add*> element.

   The <*connectionStrings*> section must be a child element of <*configuration*>, which is the root element of Web.config. In other words, <*connectionStrings*> must be a first-level element. It is common practice to place it immediately before the *<system .web>* element begins.

3. Search for the beginning of the *<system.web>* element.

   Be certain that the *<system.web>* element that you find is a first-level element—a child of *<configuration>*. Some *<system.web>* elements are lower-level children of other elements.

4. Inside the *<system.web>* element, find the *<membership>* element. Inside the *<membership>* element, find the *<providers>* element.

   The *<providers>* element contains child *<add>* elements that define each membership provider. You can register a new provider in this element.

   The *name* attribute of the *<add>* element must match the name that you configured as the ASP.NET Membership Provider in the web application. If the provider uses a connection string, the *connectionStringName* attribute must match the *name* of the connection string that you added to the *<connectionStrings>* element.

5. Inside the *<system.web>* element, find the *<roleManager>* element. Inside the *<roleManager>* element, find the *<providers>* element.

   The *<providers>* element contains child *<add>* elements that define each role provider. You can register a new provider in this element.

   The *name* attribute of the *<add>* element must match the name that you configured as the ASP.NET Membership Provider in the web application. If the provider uses a connection string, the *connectionStringName* attribute must match the *name* of the connection string that you added to the *<connectionStrings>* element.

6. Find the *<PeoplePickerWildcards>* element.

   The *<PeoplePickerWildcards>* element defines, for each custom authentication provider, the wildcard that can be used when searching for a user in the People Picker. Without a wildcard definition, the People Picker will locate only the user that is an exact match to the search criteria. With a wildcard defined, you can enter the first characters of the user's name and the search will locate all matching users.

   Each wildcard is defined by an *<add>* element in the *<PeoplePickerWildcards>* element. You can register the wildcards for your membership provider and role provider in this element.

   The *keys* must match the name of the membership and role providers that have been configured for the web application. For a SQL database, the wildcard value is *%*. For an LDAP directory, the wildcard value is *.

Lesson Summary

- If you want to enable anonymous users to access content in a SharePoint website, you must first enable anonymous authentication for the web application zone. Then a site collection administrator can enable anonymous access to an entire site—and to the objects in the site that inherit permissions from the site—or to individual lists and libraries. Finally, a farm administrator can enforce anonymous access restrictions, which are policies applied to a web application zone that prevent anonymous users from changing or even accessing content.

- SharePoint supports two authentication types: Classic Mode Authentication and Claims Based Authentication.
- In Classic Mode Authentication, users are authenticated by Windows—NTLM, Kerberos, Basic, or Digest authentication—and the resulting security token is passed by IIS to SharePoint. SharePoint translates the token to an *SPUser* object, which represents the user and his or her groups to the web application.
- In Claims Based Authentication, an unauthenticated user is redirected to the Security Token Service, which authenticates the user then provides the user with a token that contains claims about the user's identity, and can contain claims about the user's roles and other attributes. The web application trusts the token that was generated and signed by the STS, and translates the claims to an *SPUser* object.
- Windows Authentication and its methods are supported by both Classic Mode Authentication and Claims Based Authentication.
- Forms Based Authentication is an authentication provider supported by Claims Based Authentication. A user is redirected to a form and submits credentials, typically a user name and password. An authentication method validates the credentials. The authentication method can be the ASP.NET SQLMembershipProvider, which uses a SQL database of users as a directory; an LDAPMembershipProvider, which uses an LDAP directory such as an AD DS domain or instance of AD LDS as a directory; or a custom membership provider. FBA can also use a role manager to provide a list of the groups to which a user belongs. The user identity and roles are converted by the STS to a token.
- SAML token-based authentication uses Security Assertion Markup Language (SAML) tokens issued by an IP-STS external to SharePoint, such as AD FS 2.0. The trusted identity provider is registered with the SharePoint farm by importing the trust certificate of the IP-STS. Web applications can then be configured to use the trusted identity provider.
- When you upgrade a SharePoint 2007 web application that uses FBA, the application is configured to use Classic Mode Authentication, which does not use FBA, so the application will be inaccessible to FBA users. You must convert the application and then migrate users and permissions—tasks that you perform by using Windows PowerShell.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 2, "Configure Authentication." The questions are also available on the companion media in a practice test if you prefer to review them in electronic form.

NOTE **ANSWERS**

Answers to these questions and explanations of why each answer choice is right or wrong are located in the "Answers" section at the end of the book.

1. You have just configured a web application to use the Negotiate (NTLM or Kerberos) authentication method of the Windows authentication provider. What else must you do to configure Kerberos authentication?
   - **A.** Use Setspn.exe.
   - **B.** Add an SSL binding to the IIS Web site.
   - **C.** Register a trusted identity provider.
   - **D.** Configure the Web.config file of the web application.
2. You have just configured a web application to use the Negotiate (NTLM or Kerberos) authentication method of the Windows authentication provider. How can you verify that the Kerberos protocol is being used? (Choose all that apply.)
   - **A.** Browse to the website. If the website opens, Kerberos authentication is working properly.
   - **B.** Browse to the website, then examine the Security event log of the client.
   - **C.** Browse to the website, then examine the Security event log of the server.
   - **D.** Use Klist.exe on the server.
   - **E.** Use Klist.exe on the client.
   - **F.** Use Setspn.exe.
3. WebApp1 contains a single site collection. You want to allow site collection administrators to be able to grant anonymous users read-only access to content in WebApp1, based on business requirements known to the administrators. What do you do? (Choose all that apply. Each correct answer is a part of the solution.)
   - **A.** In the Site Permissions page of the top-level website in the site collection, enable anonymous access with the Entire Site option.
   - **B.** Configure anonymous access restrictions on all zones with the Deny Write option.
   - **C.** Enable anonymous access on the web application.
   - **D.** In the Site Permissions page of the top-level website in the site collection, enable anonymous access with the Lists and Libraries option.
4. You have created a web application that uses Classic Mode Authentication and the NTLM authentication method. You want to provide access to users at a partner organization. Their accounts will be kept in a SQL database. What must you do to provide this access? (Choose all that apply. Each correct answer is a part of the solution.)
   - **A.** Modify firewall settings to open port 389 to inbound TCP traffic.
   - **B.** Create SQL logins for each partner user.

C. Modify Web.config files.

D. Convert the web application to Claims Based Authentication.

E. Use Central Administration to configure the membership provider and the role provider of the web application.

5. You have just configured WebApplication1 to use Forms Based Authentication by modifying the authentication provider. When you attempt to authenticate to the website, an error appears. What else do you have to do? (Choose all that apply. Each correct answer is a part of the solution.)

   A. Modify the Web.config file of Central Administration.

   B. Modify the Web.config file of the Secure Store Service.

   C. Modify the Web.config file of the Security Token Service Application.

   D. Modify the Web.config file of the web application.

   E. Assign permissions to application pool identities.

Lesson 3: Configure Authentication Zones and Alternate Access Mappings

In Lesson 1, you learned to create and configure a web application, including the steps used to configure secure communication over SSL. In Lesson 2, you explored the management and configuration of authentication methods, including anonymous authentication. In some environments, users will access a SharePoint web application with a single protocol and URL—for example, *https://partners.contoso.com* will authenticate with a single provider, such as Windows authentication, and will be subject to a single set of policies.

But what if you want external users to access a web application using one URL and protocol, such as *https://partners.contoso.com*, and to be authenticated with Forms Based Authentication, but you want internal users to access the web application with a different URL and protocol, such as *http://extranet.contoso.com*? What if you want to enhance security and performance by implementing off-box SSL termination or a reverse proxy?

In these scenarios, you need zones and alternate access mappings. These concepts can be challenging to understand, and even more challenging to implement, because of the loose association between web applications, IIS sites, zones, intermediate devices, and alternate access mappings. In this lesson, you will learn to manage these components of a SharePoint implementation.

After this lesson, you will be able to:

- Describe the purpose of internal and public URLs.
- Describe the relationship between access mappings, zones, and IIS Web sites.
- Extend a web application to a new zone.
- Configure zone properties.
- Configure access to web applications in complex access scenarios.

Estimated lesson time: 90 minutes

Requesting SharePoint Content: Access Mappings, Zones, and URLs

Earlier in this Training Kit, you learned the high-level processes related to requests for content from SharePoint. Let's return to this process by following an example. A user wants to access the home page of the Contoso intranet. The user enters the URL *http://intranet.contoso.com* in a browser. The *public URL* of the Contoso Intranet Web application is *http://intranet.contoso.com*—the URL requested by the user. The client queries DNS to resolve the host name intranet.contoso.com to the IP address, 10.0.0.21, which happens to be an IP address bound to a network interface of a server named *SP2010-WFE1*. The request, *http://intranet.contoso.com,* is sent to 10.0.0.21 over the standard port for HTTP, port 80. The host header of the request contains intranet.contoso.com, the host name and domain name of the user's request.

Now, let's dig deeper into some of the processes that connect the user to the requested content. On SP2010-WFE1, IIS receives the user's request over port 80 and must determine which IIS Web site will service the request. IIS examines the bindings of each site and identifies the Contoso Intranet site as the correct site, because the site is bound to port 80 and the host header, *intranet.contoso.com*. IIS passes the request to SharePoint. The *internal URL* of the Contoso Intranet Web application is the URL of the website as it is received by IIS and passed on to SharePoint. In this simple example, the internal URL is also *http://intranet.contoso.com*—the same URL is passed, unchanged, from the user to IIS to SharePoint. But you will soon learn that it is not always so simple.

Access Mappings

So far, this example illustrates, in more depth, the processes used to fulfill requests for SharePoint content in the most simple environment. Let's now turn our attention to the architectural components and concepts that enable SharePoint to determine which web application should respond to the request.

SharePoint must now determine which SharePoint web application will service the request. SharePoint does not maintain a one-to-one mapping with an IIS Web site. SharePoint web applications and IIS Web sites are two separate entities, although they maintain a close relationship.

Therefore, the fact the Contoso Intranet IIS Web site received the request is not sufficient for SharePoint to know that the Contoso Intranet SharePoint web application must continue processing the request. Instead, SharePoint must examine the URL of the request that has been passed to it by IIS. Again, in this simple example, the URL is *http://intranet.contoso.com*.

SharePoint compares this URL to its access mappings. An *access mapping* associates a URL to a zone of a specific SharePoint web application. SharePoint sees that the URL, *http://intranet.contoso.com,* is associated with the default zone of the Contoso Intranet Web application. SharePoint can then continue processing the request in the context of the Contoso Intranet Web application, and return the requested content to the user.

Zones

In the previous paragraph, a new concept was introduced: the zone. A *zone* is a logical path through which users gain access to a web application. The public face of the zone—the property by which a zone is accessed—is the URL. The zone has other properties that determine how the web application is accessed. For example, the authentication provider is defined for the zone, as are policies including anonymous access restrictions and user policy.

NOTE **WHERE ZONES FIT IN**

You can think of the *zone* conceptually as the entryway to a building. The URL is the address or directions you used to get to the specific entryway—and, of course, a building can have several entryways. The web application is the building you are about to enter. And, at the entryway, your identity is verified (authentication) and policies can be put in place. For example, you might be asked to forfeit your cell phone and camera before entering the building.

When you create a web application, you also create a zone for the web application named *Default*, also referred to as the default zone. When you configure the authentication providers for the new web application, the configuration is actually applied to the default zone, not to the web application as a whole. The *Public URL* setting that you specify when you create the new web application is used to create access mappings that apply the URL as both the public URL and the internal URL associated with the default zone. As you saw earlier, the public and internal URLs are often the same.

A web application can include as many as five zones. The default zone is required, is created automatically, and can be modified. But you cannot delete the default zone. It is deleted automatically when you delete the web application. The four additional zones are optional, can be created, modified, and deleted, and are named *intranet*, *extranet*, *Internet*, and *custom*.

To define an additional zone, you simply create an access mapping that associates a unique URL to one of the four additional zones, and then users can access the web application through the new URL. For example, you could add a URL, *http://company.contoso.com*, to the Contoso Intranet Web application as the URL for the zone named *intranet*. Users could then access the Contoso Intranet Web application using either *http://company.contoso.com* or *http://intranet.contoso.com*. Of course you would need to be certain that there was a DNS host record to resolve the names to the IP address of the server, and you would need to add a host header binding to the IIS Web site so that the site would respond to requests to *company.contoso.com* as well as to *intranet.contoso.com*.

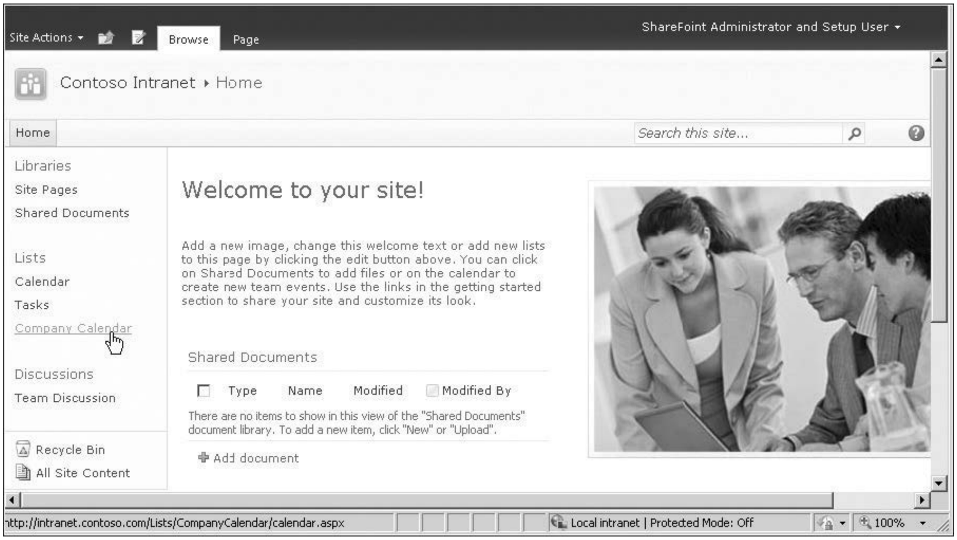
The most important thing to know about a zone at this point in the discussion is that after you have defined an additional zone, the content that users access through the new URL is the same content they access through the default zone. Zones are simply different logical paths to the same web application—an association between the protocol, scheme, hostname, and port of an inbound request from a client, and the web application that will respond to the request.

NOTE **MULTIPLE ZONES**

To continue our conceptual metaphor, you can think of different zones as different entryways into the same building. Each entryway requires a different set of directions to get there, and each zone requires a different URL.

URLs of SharePoint Site Collections and Content

Let’s take a short digression to examine the answer to the question, “What happens next?” Let’s assume one user browses to *http://intranet.contoso.com*, the home page of the Contoso intranet, shown in Figure 3-12, and clicks the link to the Company Calendar. You can see in Figure 3-12 that the URL to the Company Calendar is *http://intranet.contoso.com/Lists/CompanyCalendar/calendar.aspx*.



Welcome to your site!

Add a new image, change this welcome text or add new lists to this page by clicking the edit button above. You can click on Shared Documents to add files or on the calendar to create new team events. Use the links in the getting started section to share your site and customize its look.

There are no items to show in this view of the "Shared Documents" document library. To add a new item, click "New" or "Upload".

http://intranet.contoso.com/Lists/CompanyCalendar/calendar.aspx

FIGURE 3-12 URL of a hyperlink in a web application

What if another user accesses the same Contoso intranet application by entering the URL, *http://company.contoso.com*? When that user clicks the link to the same Company Calendar, the URL will be *http://company.contoso.com/Lists/CompanyCalendar/calendar.aspx*.

How is it that two different users can submit two URLs and access the same content?

Conceptually, SharePoint itself considers the address of the Company Calendar to be *<WebApplicationURL><SiteCollectionURL>Lists/CompanyCalendar/calendar.aspx*, where *<WebApplicationURL>* is the Public URL of the zone, such as *http://intranet.contoso.com:80*, and *<SiteCollectionURL>* is the URL of the site collection relative to the web application. For example, the intranet site collection is at the root of the web application, with the relative URL */*. The URL of content within a site collection is always stored as a value relative to the site collection itself. For example, SharePoint considers the URL of the company calendar to be *Lists/CompanyCalendar/calendar.aspx*.

SharePoint renders the home page of the intranet and generates the hyperlink to the Company Calendar by replacing the variables with their values. For the user that accesses the Contoso intranet through the default zone, the resulting hyperlink target URL is *http://intranet.contoso.com/Lists/CompanyCalendar/calendar.aspx*. SharePoint removes the port, 80, because it is the default port for HTTP. If the Public URL included a non-standard port, it would be included in the link to the Company Calendar. For the user that accesses the Contoso intranet through the intranet zone, the resulting hyperlink target URL is *http://company.contoso.com/Lists/CompanyCalendar/calendar.aspx*.

By abstracting the URL of the web application and the relative URL of the site collection, SharePoint allows a user to access content through more than one URL, or zone. It also enables you to move content easily. If you create a new web application with the public URL *http://portal.contoso.com:80*, and you move the content database from the intranet web application to the portal web application, the URL to the Company Calendar will immediately be generated as *http://portal.contoso.com/Lists/CompanyCalendar/calendar.aspx*. The site collection relative URL, and the relative URL of the Company Calendar did not change—only the web application hosting the site collection changed. Similarly, if you change the external URL of the extranet zone from *http://company.contoso.com* to *http://portal.contoso.com*, the URL of the Company Calendar would be rendered with the new hostname.

Access Mappings

Now that you understand the fundamental concepts of access mappings and zones, let's explore in more detail how you manage each.

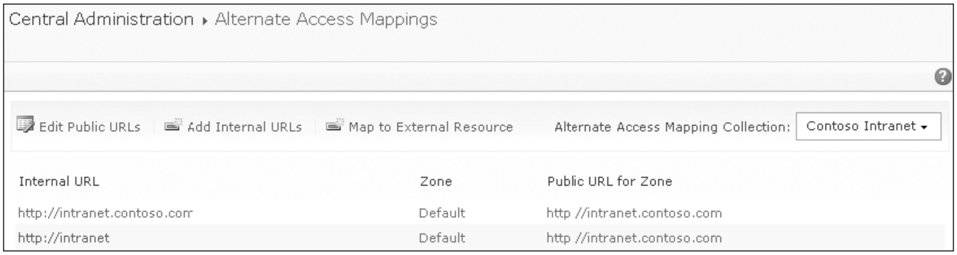
You have learned that an *access mapping*, also called an *alternate access mapping*, associates a URL to a zone of a specific SharePoint web application. Behind the scenes, in the SharePoint object model, an access mapping is called an *alternate URL*—it is an *SPAlternateURL* object in a collection called *SPAlternateURLCollection* that is a member of the web application. Wouldn't it have been easier if they just called these *alternate URLs*?

You can manage URLs from the Alternate Access Mappings page of Central Administration.

MANAGE ALTERNATE ACCESS MAPPINGS

1. In the Central Administration Quick Launch, click Applications Management.
2. In the Web Applications section, click Configure Alternate Access Mappings.

   The Alternate Access Mappings page, shown in Figure 3-13, opens.



Alternate Access Mapping Collection: Contoso Intranet ▾

Internal URL	Zone	Public URL for Zone
http://intranet.contoso.com	Default	http //intranet.contoso.com
http://intranet	Default	http //intranet.contoso.com

FIGURE 3-13 The Alternate Access Mappings page

From here, you can do the following:

- Click the Alternate Access Mappings Collection selector to pick the web application that you want to modify.
- Click Edit Public URLs to add, modify, or delete the public URL of each zone.
- Click Add Internal URLs to add an internal URL to a zone.
- Click Map To External Resource to configure a URL that maps to a resource outside of SharePoint.
- Click an internal URL to edit or delete the internal URL.

It is easy to understand why the public URL and the internal URL are often the same. A user enters the public URL, *http://intranet.contoso.com*, and IIS receives the request with the same URL and passes the request to SharePoint, which inspects the request and extracts the URL as the internal URL.

It gets more interesting when you answer the question, "Why would the internal URL be *different* than the public URL?" We will answer the question with two common scenarios:

- Single-label host names and fully qualified host names
- Off-Box SSL Termination

In the process of exploring these scenarios, you will learn the purpose of the internal URL and of the public URL.

Single-Label Host Names and Fully Qualified Host Names

You have configured the Contoso Intranet Web application with the URL *http://intranet.contoso.com*. This URL is the URL of the web application—a property in and of itself—and it is both the public URL and the internal URL of the web application's default zone. Let's assume you want to allow users to get to the intranet by typing *http://intranet* or with the current URL, *http://intranet.contoso.com*.

This scenario is supported by two access mappings: You add *http://intranet* as an additional internal URL to the default zone. When a request for *http://intranet* is passed to SharePoint by IIS, SharePoint will identify the URL as the internal URL of the default zone of the Contoso Intranet Web application, and will be able to serve the website to the user.

ADD INTERNAL URLS

1. On the Alternate Access Mappings page, click Add Internal URLs.

   The Add Internal URLs page opens.

2. Confirm that the Alternate Access Mapping Collection list displays the web application that you want to modify.

   If it does not, click the list, then click Change Alternate Access Mapping Collection, and then click the name of the web application that you want to modify.

3. In the URL Protocol, Host And Port box, type the internal URL that you want to add.

   The URL should include the protocol and host name, and the port—for example, *http://intranet:80*. Although you can omit the port if it is a standard port, it is recommended that you use the port for clarity and documentation.

4. In the Zone list, select the zone with which to associate the internal URL and click Save.

The internal URL—*http://intranet* in this example—allows SharePoint to determine which SharePoint web application is being accessed, and through which of that web application's zones. However, the request has to be passed by IIS to SharePoint before that can happen. When you add an internal URL, SharePoint does not add a corresponding host header binding to the IIS Web site. You must manually add the host name of the internal URL as a host header binding to the IIS Web site on each server that will respond to the internal URL.

This is not applicable if you are not using host headers—for example, if you have a dedicated IP address bound to the IIS Web site.

ADD A BINDING TO AN IIS WEB SITE

1. In IIS Manager, in the console tree, expand the server, then expand Sites, and then click the site to which you want to add a binding.
2. In the Actions pane, click Bindings to open the Site Bindings dialog box.
3. Click Add to open the Add Site Binding dialog box.
4. In the Type box, select the protocol—http or https.
5. In the Host Name box, type the host name.
6. Click OK and then click Close.

> ***BEST PRACTICE*** **MINIMIZING THE MANAGEMENT BURDEN**
>
> **When you add an internal URL to a zone for a web application that uses host headers, you must add a binding to the IIS Web site. If you add a new server to the farm, SharePoint will create the IIS Web site but will not add the additional bindings. And, if you ever have to restore the web application, SharePoint will not re-create the bindings. In these ways, the management burden is increased. It is therefore recommended that you minimize the number of instances in which a zone has more than one internal URL.**

After an internal URL has been created, you can modify or delete the URL. On the Alternate Access Mappings page, click the URL in the Internal URL column. If you modify or delete an internal URL, be certain to modify bindings on the IIS Web site accordingly.

In our scenario, users can now request *http://intranet*, which has been added as an internal URL to the default zone, as shown in Figure 3-13. The zone now has two internal URLs and one public URL, *http://intranet.contoso.com*.

Let's consider what happens when a user requests *http://intranet*. The request arrives at the web server. The host header binding enables IIS to pass the request to the Contoso Intranet IIS Web site. The website's SharePoint processes receive the request and examine the URL to determine that the request is for the Contoso Intranet Web application. It is now time for SharePoint to render the content of the intranet home page to the user. Remember, from the example presented earlier, that the home page has a link to the company calendar. SharePoint thinks about the link in relative terms. SharePoint considers the URL for the company calendar to be */Lists/CompanyCalendar/calendar.aspx*. When SharePoint renders the link, it adds the public URL of the web application—more specifically, of the zone—to the URL. Therefore, the link that SharePoint renders for the company calendar is *http://intranet.contoso.com/Lists/CompanyCalendar/calendar.aspx*. SharePoint does not render the link as *http://intranet/Lists/CompanyCalendar/calendar.aspx*.

In this scenario, there is no problem. When a user clicks the link to the company calendar, the user can access the calendar with the *http://intranet.contoso.com* URL. In fact, all URLs will be rendered with the fully qualified domain name of the host—with the public URL. The first thing that a user clicks will take the user out of the *http://intranet* URL namespace into the *http://intranet.contoso.com* namespace. Again, no problem is caused—access is still possible. You have simply given users an alternate, shorter URL with which to get to the intranet home page.

Off-Box SSL Termination

The previous scenario was straightforward. Let's explore a slightly more complex scenario. You have configured the Contoso Partner Portal to use SSL with the URL *https://partners.contoso.com*. This is both the public URL and the internal URL of the web application's default zone.

You decide that you want to reduce the performance burden that SSL places on the web server by installing a device that performs off-box SSL termination. This is a device that is placed on the network, logically, between the user and the web server. The device receives the request using SSL over port 443, decrypts the request, and forwards it to the web server, unencrypted, over http with port 80.

In this configuration, users continue to browse to the public URL *https://partners.contoso.com*. DNS resolves the IP address as the network interface of the off-box SSL terminator. The device receives the packet, does its magic, and then forwards the request to IIS as *http://partners.contoso.com*. IIS passes the request to SharePoint, and the internal URL is *http://partners.contoso.com*.

This scenario is addressed with one access mapping. To support this configuration, you must define the zone so that the public URL is *https://partners.contoso.com*, and the internal URL is *http://partners.contoso.com*.

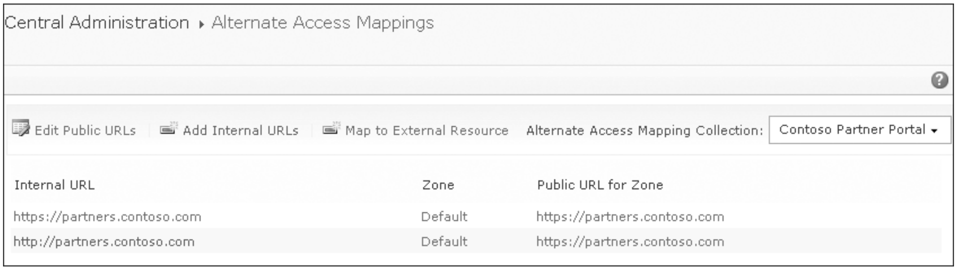
SharePoint must understand that the URL, *http://partners.contoso.com,* is associated with the default zone of the Contoso Partner Portal web application, so that SharePoint can retrieve the requested content. The internal URL is the mapping that is important for SharePoint to process inbound requests. For this scenario to work, the internal URL of the zone must be *http://partners.contoso.com:80*, although you can optionally leave out the port if the URL uses the standard port for the protocol.

It is also important that SharePoint knows that, to the user, the web application is known as *https://partners.contoso.com*, because SharePoint must render hyperlinks and other URLs so that they will be accessible to the user. Consider, again, the link to the company calendar. If SharePoint rendered the link using the internal URL, it would be *http://partners.contoso.com/Lists/CompanyCalendar/calendar.aspx*, and the user would not be able to click the link and connect to the content successfully.

That is why the public URL is important. The public URL is also referred to as the *outgoing URL* or the *response URL*. In this example, the public URL for the default zone is *https://partners.contoso.com:443*. It is the public URL that is used to ensure that URLs are rendered correctly for users. Because SharePoint knows that the request arrived with an internal URL associated with the default zone, and that the public URL for the default zone is *https://partners.contoso.com:443*, SharePoint renders the link to the Company Calendar as *https://portal.contoso.com/Lists/CompanyCalendar/calendar.aspx*. The user is therefore able to click the link and navigate to the company calendar.

> ***IMPORTANT*** **MANAGING URLS IN MULTIPLE ZONES**
>
> **Never configure an internal URL in one zone that is the same as the public URL of a different zone. This can cause SharePoint to render URLs incorrectly to users accessing the site with the URL as its public zone.**

To support the preceding off-box SSL termination scenario, if you have created the web application using SSL as *https://partners.contoso.com*, the public URL is already correct. Add an internal URL, *http://partners.contoso.com*. The resulting Alternate Access Mapping (AAM) collection for the Contoso Partner Portal application is shown in Figure 3-14.



Alternate Access Mapping Collection: Contoso Partner Portal ▾

Internal URL	Zone	Public URL for Zone
https://partners.contoso.com	Default	https://partners.contoso.com
http://partners.contoso.com	Default	https://partners.contoso.com

FIGURE 3-14 Access mappings to support off-box SSL termination for the Contoso Partner Portal

The second access mapping is the one that supports the scenario. In this case, the first access mapping is a special mapping that cannot be deleted. For each web application, one access mapping represents the default zone. You can recognize this mapping because when you click the URL on the Alternate Access Mappings page, the Delete button is disabled—you cannot delete the mapping. The public URL and the internal URL of this mapping are the same. You cannot change one without changing the other. We recommend that you do not change this mapping in any way. Instead, add other internal URLs to the default zone, and extend the web application to create new zones.

Load Balancing with Request Overwrites

A load balancer is a service or device that distributes inbound requests to more than one web server. The web server that services the request returns the content to the user. For example, let's say a user requests *http://intranet.contoso.com/SitePages/Home.aspx*. The public URL of the zone is thus *http://intranet.contoso.com*. The load balancer, which can be the Network Load Balancing (NLB) service provided by Windows Server 2008 or a dedicated hardware device, receives the request. Two web servers, named *SP2010-WFE1* and *SP2010-WFE2*, host the intranet web application. The load balancer uses rules, which can be as simple as a round-robin algorithm, to determine the server that will serve the request. The request is distributed to one of the servers.

Many load balancers use a shared IP address that receives the request, and then the load balancer forwards the request to the IP address of one of the servers. If this mechanism is used, the URL sent to IIS is the same URL received by the NLB service. There is no need to change access mappings or to create additional zones.

Some load balancers overwrite the request received from the client and submit the request to the web server. For example, let's say a load balancer receives a request for *http://intranet.contoso.com/SitePages/Home.aspx*. It changes the URL of the request to address a specific server. The new URL is *http://intranet02.contoso.com/SitePages/Home.aspx*.

The request is received by SP2010-WFE2, which hosts the intranet web application. The IIS Web site must be bound to *intranet02.contoso.com* on SP2010-WFE2. Similarly, the IIS Web site must be bound to *intranet01.contoso.com* on SP2010-WFE1. IIS passes the request to SharePoint as *http://intranet02.contoso.com/SitePages/Home.aspx*. SharePoint must return the content to the user. To achieve this, the zone must be mapped to the public URL *http://intranet.contoso.com* and there must be two internal URLs: *http://intranet01.contoso.com* and *http://intranet02.contoso.com*.

NOTE **FORWARDED REQUESTS FOR AN IP ADDRESS**

Many load balancers forward a request to the specific IP address of a web server without actually changing the URL. This process would not require a different internal URL.

Review Internal and Public URLs

To summarize, each web application zone has one or more internal URLs. When an inbound request is received by IIS and passed to SharePoint, the URL is examined and matched to an internal URL to determine which web application is being accessed, and through which zone. In simple environments, the internal URL of a zone will be the same as the public URL.

The internal URL will be different from the public URL of a zone if a device or service changes the URL that the user requests. For example, an SSL termination service or device changes the protocol of the request from HTTPS to HTTP. A user requests *https://partners.contoso.com* over port 443 and an SSL termination device forwards the request to *http://partners.contoso.com* over port 80. Or, a load balancer uses request overwrites to forward the requests to a web server.

EXAM TIP

The internal and external URLs of an access zone are different when the URL entered by a user is different than the URL received by IIS and SharePoint. This occurs when there is a device, such as an off-box SSL terminator or reverse proxy, between the user and the web server, and when that device overwrites or changes the URL entered by the user.

Public URLs

You can also modify the public URLs of a web application. Remember that the primary purpose of the public URL is to enable SharePoint to render content correctly to users. SharePoint uses the public URL as the URL of the web application as it renders hyperlinks and other URLs.

EDIT THE PUBLIC URLS OF A WEB APPLICATION

1. On the Alternate Access Mappings page, click Edit Public URLs.
2. On the Edit Public Zone URLs page, confirm that the Alternate Access Mapping Collection list displays the web application that you want to modify.

   If it does not, click the list, then click Change Alternate Access Mapping Collection, and then click the name of the web application that you want to modify.
3. Enter, edit, or delete the URLs that users use to access the web application.
4. Click Save.

NOTE **DELETING THE DEFAULT ZONE**

You cannot remove the public URL associated with the default zone, which must always be defined. The default zone is deleted when you delete the web application itself.

Manage Zones

A zone, as you've learned, is a logical path to a web application and, technically, is the result of an access mapping that associates a URL and a web application. Five zones are available for each SharePoint web application: *default*, *intranet*, *Internet*, *extranet*, and *custom*.

BEST PRACTICE **MAKING THE BEST USE OF THE TYPES OF ZONES**

The names of the additional four zones (*intranet*, *extranet*, *Internet*, and *custom*) bear no technical meaning. There is no difference in configuration between the *Intranet* and the *Extranet* zones—both are simply a logical path—a public URL and an "entryway"—to a web application. The names of zones are for guidance only. Therefore, it is a best practice but not a requirement to extend a web application to the *Extranet* zone if the purpose of the zone will be to support access through your extranet. The zones also have no relationship to Internet Explorer security zones. It is a common misconception that the names of the zones connote technical considerations.

The Default zone, however, is—as its name suggests—the default. If SharePoint cannot determine zone-specific policies to apply to an inbound request, it uses the policies associated with the default zone. You will learn more about policies and zones later in this lesson.

You can create and delete a zone in a web application in two ways: You can define an alternate access mapping, or you can extend and unextend the web application. You will learn both methods in this section, but it is highly recommended that when you want to create or delete a zone, you always extend or unextend the web application, respectively.

Define an Alternate Access Mapping

Let's start with the method that is not recommended: defining an Alternate Access Mapping (AAM).

You learned how to add internal and public URLs earlier in the lesson, in the procedures "Add Internal URLs" and "Edit the Public URLs of a Web Application." If you add a URL to a zone that was previously undefined, you create the zone. The URL you specify as the internal or Public URL of an undefined zone is also added as the public or internal URL, respectively, so that the zone has both the required public URL and the required first internal URL.

Conversely, if you clear the public URL of a zone, or if you delete the last internal URL of a zone, you delete the zone.

It is not recommended to create and delete zones in this fashion. The following are best practices that relate to access mappings and zones:

- Do not add an internal URL associated with a zone that does not yet exist. Doing so creates a zone without an associated website. Instead, extend the web application.
- Do not add a public URL to a zone that does not yet exist. Doing so creates a zone without an associated website. Instead, extend the web application.
- Do not delete the last internal URL associated with a zone. Doing so deletes the zone without deleting the associated website. Instead, unextend the web application.
- Do not remove the public URL associated with a zone. Doing so deletes the zone without deleting the associated website. Instead, unextend the web application.

Extend a Web Application

The second, and recommended method to create a new zone in SharePoint 2010 is to *extend* a web application. When you extend a web application, you create a new zone and an associated IIS Web site.

EXTEND A WEB APPLICATION

1. In the Central Administration Quick Launch, click Application Management.
2. In the Web Applications section, click Manage Web Applications.
3. Select the web application to extend.
4. On the ribbon, click Extend.
5. On the Extend Web Application To Another IIS Web Site page, click Create A New IIS Web Site.
6. In the Name box, type a name that is easily recognizable in IIS Manager, such as **Contoso Partners Extranet**.

7. In the Port box, type the port number. If you are using HTTP, this is usually port 80; HTTPS is usually 443.
8. In the Host Header box, type the host header, which is usually the FQDN of the zone, such as **extranet.contoso.com**.

   You should configure the host header even if you plan to bind the IIS Web site to a unique IP addresses. The host header becomes the internal URL of the site.
9. Configure the settings in the Security Configuration section. See the section, "Security Configuration," earlier in this chapter for more information.
10. In the Public URL section, in the URL box, type the external URL of the zone, such as **http://extranet.contoso.com:80**.
11. In the Zone list, select the zone to which you want to extend the web application, and then click OK.

When you extend a web application, an IIS Web site is created. The IIS Web site for the zones will share the application pool of the web application's other zones. Do not change the application pool associated with a zone.

The Case for Extending Web Applications

Why is it not recommended to create a zone by defining an access mapping, and why is it recommended to do so by extending the web application? You will learn several reasons in this lesson. The first is that the access mappings you create manually are not added to the IIS site underlying the web application. You must therefore manually change the bindings of the IIS site on each web server in the farm to add the new URL as a host header binding. If you add a new server to the farm, SharePoint will create the IIS site on the new server, but will not add the URLs that you added as access mappings. Similarly, if you delete the web application and IIS site, and then restore the web application from a SharePoint backup, SharePoint will re-create the IIS site but will not re-create the bindings.

These problems illustrate one of the separations between SharePoint and IIS: Changes made directly to IIS are not stored in the SharePoint configuration database; therefore, you cannot manage the settings by using SharePoint.

More important, when you create a zone by defining an access mapping in SharePoint, the only thing you have accomplished is to provide access to the web application with a different URL. Earlier in this lesson, you learned that a number of settings can be scoped to a zone, including authentication providers, anonymous access, and policy. These settings can only be applied if an IIS Web site is associated with the zone.

When you extend a web application, you create an IIS Web site associated with the zone. SharePoint configures the IIS Web site with bindings—for example, with a host header binding for the URL of the new zone. This allows SharePoint and IIS to stay in synch with each other. When you extend a web application to a zone, the configuration of the extended web application is stored in the SharePoint configuration database. Therefore, if you add a new server to the farm, SharePoint can create the IIS site and configure it automatically. Similarly, if you delete a web application and then restore it from backup, SharePoint can configure the IIS site.

IMPORTANT **EXTEND THE WEB APPLICATION TO CREATE A NEW ZONE**

To create a new zone, it is recommended that you extend the web application. If you create a zone by adding an access mapping, you cannot use the zone to configure authentication, anonymous access, or user policy uniquely for the zone.

Remove a Zone

To remove a zone properly, you should undo the process you used to create the zone. If you created the zone by defining an alternate access mapping, remove the URLs associated with the zone and the zone will be deleted. If a web application has been extended to the zone, you should delete the extended zone using the procedure that follows.

REMOVE AN EXTENDED ZONE

1. On the Central Administration Quick Launch, click Application Management.
2. In the Web Applications section, click Manage Web Applications.
3. Select the web application for which you want to delete an extended zone.
4. On the ribbon, click the down arrow on the Delete button, and then click Remove SharePoint From IIS Web Site.

IMPORTANT **BE VERY CAREFUL**

Be careful that you choose the correct command. Do not choose the Delete Web Application command or click the Delete button itself, because both actions will result in the deletion of the entire web application.

5. In the Select IIS Web Site And Zone To Remove list, select the zone that you want to remove.

IMPORTANT **DON'T DELETE THE DEFAULT ZONE**

By default, the default zone selected. If you delete the default zone, the web application will be broken. Be certain that you select a zone other than the default zone.

6. If you also want to delete the IIS Web site associated with the zone, click Yes in the Delete IIS Web Sites section.
7. Click OK.

Multiple Zones

Why might you want more than one zone? Zones are also used to scope certain settings for access to a web application, including authentication providers, anonymous access, and policy. If you want to provide access to the content of a web application with more than one variation of these settings, you must use more than one access zone. In this section, you will explore several common scenarios, and you will learn how to address those scenarios by configuring multiple zones.

Problems Resulting from Multiple Zones

Before we proceed, however, let's consider the type of problems that can arise whenever more than two zones are in use. Let's illustrate the problems with examples.

Both John and Jane access the Contoso Partner Portal. John is an internal employee, and accesses the site through the URL *http://extranet.contoso.com*. Jane is a partner, and accesses the site through the URL *https://partners.contoso.com*. Jane wants to send John a link to a the company calendar, which SharePoint renders to her as *http://extranet.contoso.com/Lists/CompanyCalendar/calendar.aspx* because *http://extranet.contoso.com* is the public URL of the default zone. She copies the link to the calendar and sends it to John. John cannot access the site through the URL namespace, *http://extranet.contoso.com*, so the link is inaccessible to John.

SharePoint renders the URLs of content with the public URL of the zone through which the site was accessed. Unfortunately, this is true only of links and URLs that SharePoint generates, such as navigation links in the Quick Launch. However, if a link is hard-coded on a page, it is not altered. For example, if Jane pastes a link to the company calendar into an announcement on the home page, she pastes it as *http://extranet.contoso.com/Lists/CompanyCalendar/calendar.aspx*. When John accesses the site, the URL is not altered to reflect the fact that he is accessing the home page as *https://partners.contoso.com*. The link is effectively broken for John.

NOTE **A CAUTION ABOUT ABSOLUTE URLS**

Absolute URLs—URLs that are hard-coded on a page—cannot be mapped. Be careful about using absolute URLs in the content of a web application that is accessed through more than one URL.

When you are faced with the choice of extending a web application—of creating a new zone—consider the impact on users and applications that will access the web application using a different URL namespace.

With those caveats, let's explore scenarios that can be implemented by creating additional zones. In each of these scenarios, you use the procedure described earlier, "Extend a Web Application," to create an additional zone.

Multiple Authentication Providers

A web application can be configured to only one authentication type—either Classic Mode Authentication or Claims Based Authentication. So if you have two solutions that cannot work in the same mode—for example, custom code that uses Windows identities and therefore cannot work in Claims Based Authentication, and another solution that relies on claims—you must separate the solutions into two different web applications.

Within a web application configured for claims authentication, however, you can have more than one authentication provider. For example, different users can be authenticated by either Windows, forms, or SAML token authentication within a single web application, with a single zone. For example, you can support both Windows and Forms Based Authentication (FBA) on the default zone.

This is a significant improvement over previous versions of SharePoint. In SharePoint 2007, you were required to extend a web application into additional zones to implement different types of authentication for users coming from different networks or authentication providers. This could lead to practical business problems because the different users accessed the content through different URLs.

The fact that Claims Based Authentication can support multiple authentication providers with a single zone reduces a significant design driver for multiple zones.

However, you can still specify different authentication providers on each zone and, in some cases, it will be required. If a zone is configured for forms-based authentication, the zone supports only one provider for FBA. It is not common to require multiple FBA providers, but if some users must be authenticated using the SQLMembershipProvider against a SQL database, and others must be authenticated using the LDAPMembershipProvider against an instance of Active Directory Lightweight Directory Services (AD LDS) or some other LDAP source, those users must access the site through two different zones, and therefore two different URLs. Alternately, you could write a custom FBA provider that abstracts the authentication provided by the two different sources.

Anonymous Access Enabled

Let's examine a scenario in which all content on the Contoso Intranet Web application can be accessed by authenticated users, based on permissions assigned to the content, using the URL *http://intranet.contoso.com*. However, you also want a subset of intranet content to be accessible by anonymous users—such as customers who are visiting Contoso and are connected to the Contoso network—using the URL *http://visitors.contoso.com*.

To support this scenario, you extend the Contoso Intranet Web application to a new zone—perhaps to the zone named *intranet*—with the public URL, *http://visitors.contoso.com*, to the web application. Each zone that has an extended web application supports its own authentication settings. On the zone named *intranet* associated with *http://visitors.contoso.com*, you enable anonymous access. Anonymous access remains disabled on the zone named *default* that is associated with *http://intranet.contoso.com*.

When a user enters a URL that begins with *http://visitors.contoso.com*, the request is received by SharePoint and SharePoint can map the request to the intranet zone of the Contoso Intranet Web application. This zone will accept anonymous connections.

Anonymous Access Restrictions

Anonymous user policy is also set per zone. You might want to allow anonymous users to access both *http://intranet.contoso.com* and *http://visitors.contoso.com*, but enforce read-only access through the zone associated with the URL *http://visitors.contoso.com*.

Reverse Proxy

A reverse proxy is a service that acts as a connector between end users and the web server. A user makes a request, and the reverse proxy receives the request. The reverse proxy filters and translates the request, and then forwards the request to the web server. SharePoint is compatible with many reverse proxy services and devices.

Although each reverse proxy product varies in functionality, they all have the following common characteristics:

- The reverse proxy can authenticate the user and perform inbound filtering based on characteristics of the request packet, and then forward only eligible requests to the web server.
- The reverse proxy can change the URL (host name or port) of the URL requested by the user. For example, a user requests *http://portal.contoso.com* and a reverse proxy receives the request and forwards the request to *http://partners.contoso.com*.
- The reverse proxy can receive requests using one port or protocol, and then forward the requests using another port or protocol. In this way, a reverse proxy can perform off-box SSL termination.
- The reverse proxy can forward the request to a different port than the port on which the request was originally received, and can change the HTTP host header field, thereby masking the internal name of a server or application from external users.

> ***NOTE*** **FOREFRONT AND THE REVERSE PROXY**
>
> **Microsoft Forefront Unified Access Gateway (UAG), formerly Intelligent Application Gateway (IAG), is a very powerful reverse proxy and is the preferred reverse proxy offered by Microsoft. Microsoft Forefront Threat Management Gateway (TMG), formerly Internet Security and Acceleration Server (ISA), can also perform some reverse proxy functionality, but its primary purpose is to serve as an outbound proxy, and to protect users from Internet-based malware.**

In reverse proxy scenarios, the URL of a request is directed to the reverse proxy. The URL sent by the reverse proxy to the web server is typically a different host name or port. For example, a user requests the home page of the Contoso Extranet web application, *https://portal.contoso.com/SitePages/Home.aspx*. The public URL of the Contoso Extranet web application—the URL as known to users—is thus *https://portal.contoso.com*. A reverse proxy that handles SSL encryption receives the request over port 443, and translates the request—for example, to *http://extranet.contoso.com/SitePages/Home.aspx* to the web server over port 80. The internal URL of the Contoso Extranet web application—the URL of the web application as known to SharePoint—is thus *http://extranet.contoso.com*. Typically, this URL is not directly accessible to the user—the user would not be able to resolve or connect to the site using the URL *http://extranet.contoso.com*—port 80 would be blocked by the firewall sitting between the user and the reverse proxy.

NOTE **LINK TRANSLATION AND EXTERNAL URLS**

Some reverse proxies have a feature called *link translation* that can translate outgoing links. Link translation is not sufficient to obviate the need for defining an access zone with the external URL of the web application. Always configure an extended zone to reflect the public URL submitted by the user to the reverse proxy, and the internal URL submitted by the reverse proxy to IIS.

HTTPS and HTTP

Earlier you learned to configure a single zone so that it could be accessed by users over SSL using off-box termination. The intermediate device forwarded requests to HTTP port 80 on the web server, but users did not access the web application directly using HTTP.

What if you want to provide access to a web application to users on your internal network with the URL *http://intranet.contoso.com* but you want access from outside the network to use SSL, and thus the URL *https://intranet.contoso.com*?

In this configuration—when you want to support user access through *both* HTTP and HTTPS protocols, you must have two zones. Extend the web application. One zone—typically the default zone—is configured for SSL and the other zone without SSL. Do not simply add the SSL binding to the IIS Web site that is also bound to HTTP.

Different Policies

You will learn about web policies in Chapter 4. In short, a policy can grant a user or group permissions to content in a web application that override any permissions associated with a specific site, list, library, folder, item, or document. For example, you can specify a policy for the default zone of the Contoso Intranet that grants the Level 3 Help Desk group Full Control permission to content so that they can support users who are attempting to post content to the intranet. This policy applies when the Contoso Intranet Web application is accessed with the URL *http://intranet.contoso.com*, which is the external or public URL associated with the default zone representing the corporate network. You might want to prevent such broad application of Full Control permission when users in the Level 3 Help Desk group access the Contoso Intranet Web application over the Internet. If the Contoso Intranet Web application is extended to the Internet zone as *http://portal.contoso.com*, the web policies associated with the Internet zone are separate, so you can forego granting the Level 3 Help Desk group Full Control policy. Access using the external Internet zone's external URL, *http://portal.contoso.com*, is not subject to the same web policies as the default zone.

NOTE **ZONES AND WEB APPLICATIONS SETTINGS**

Although a zone defines some settings, such as an authentication provider, anonymous access restrictions, and user policy, much belongs to the SharePoint web application regardless of how it is accessed. Most important, the web application contains the content databases, and therein the site collections and content of the website.

The web application also specifies many settings that apply to the web application, rather than the zone through which the web application is accessed, such as the features that are available in the web application.

Guidance and Recommendations for Zones and Access Mappings

You have explored common scenarios that require more than one zone, and you know some of the common problems that might arise when you have multiple zones. You also know that the best practice for creating an additional zone is to extend a web application. You should also be familiar with the following best practices and recommendations:

- Configure the default zone as the most secure zone. Typically, this means that, if you plan to use SSL for any zone, you should use it for the default zone. If you plan to enable anonymous authentication on one but not all zones, do not enable anonymous authentication on the default zone. When SharePoint cannot determine which policies to apply to an inbound request, it applies the policies associated with the default zone.
- System-generated alerts, such as those related to quotas and site collection usage, are sent using the URLs associated with the default zone. Therefore, you should configure a web application's default zone external URL as the most-often used URL. This consideration should be secondary to the primary recommendation in this list.
- You have learned that you can manually modify host header bindings on the IIS Web site from the IIS Manager, but this is not recommended. Any changes you make using the IIS Manager will not be recorded in the configuration database of the farm, and will not be replicated to other servers in the farm.
- Do not modify the host header binding that SharePoint applies to an IIS site. If SharePoint Server 2010 tries to provision an IIS Web site on another computer in the farm for the same web application and zone, the original host header binding is used instead of the modified binding. If you want to modify an existing binding for an IIS Web site, remove the web application from the zone and then re-extend the web application into the zone with the host header you want to use.
- Do not add the public URL of a zone as a binding to the IIS site of another zone. For example, if the public URL of the extranet zone is *http://extranet.contoso.com*, do not add the host header *extranet.contoso.com* to the IIS site of the default zone. Such manual configuration is not replicated to other IIS servers in the farm. It is not recommended to use the same IIS Web site for multiple zones, unless you are specifically told to do so by Microsoft.
- Host-named site collections cannot use alternate access mappings. Host-named site collections are automatically considered to be in the default zone, and the URL of the request must not be modified between the end user and the server. You will learn more about host-named site collections later in this training kit.
- One zone must be configured to use Windows authentication. The crawler uses NTLM to authenticate. If no zone supports NTLM, content will not be indexed, and therefore a search will never produce results from the web application.

- A URL should never be used as the internal URL of two different zones. The public URL of one zone should never be used as the internal URL of a different zone. If you put these rules together, a URL can only be used once within a web application: either as the public URL of a zone or as the internal URL of a zone. A URL can only be used twice when the URL is both the public and internal URL of a single zone.

Lesson Summary

- Each SharePoint web application is created with a zone named *default* that cannot be deleted. Four additional zones can be defined: *intranet*, *Internet*, *extranet*, and *custom*.
- A zone is defined by one public URL and one or more internal URLs. Each URL includes a protocol, scheme, host name, and port. For example, the public URL of the Contoso intranet is *http://intranet.contoso.com:80*, although the port, 80, can be omitted when you use the standard port for the HTTP protocol (80) or HTTPS protocol (443).
- The internal URL of an access mapping associates the internal URL of a request—the URL as it is received by SharePoint—to a web application zone. The internal URL is also called the *incoming URL*.
- The public URL of a web application allows SharePoint to render content to the user with URLs—for example, target URLs of hyperlinks—with a protocol and host name that are valid for the user, which might be different than the internal URLs of the web application itself. The public URL is also called the *external URL*, *load-balanced URL*, or *outgoing URL*.
- Each zone can specify unique configuration, such as anonymous authentication and restrictions, user policy, and SSL.
- If the internal URL of a request SharePoint has received from IIS is *http://intranet.contoso.com*, SharePoint knows that the request is associated with the zone named *Default* for the Contoso Intranet Web application. SharePoint then performs authentication as configured for the zone, and after the user has been authenticated, SharePoint parses the full URL of the request to determine the site collection and content database that must be accessed. Finally, SharePoint authorizes the user's access, based on a combination of the user's permissions to content and any policies that are enforced on the web application or zone, and then fulfills the request, returning the home page of the Contoso intranet to the user.

PRACTICE Configure Access Mappings and Zones

Practices are designed to guide you through important procedures. The instructions in the Training Kit are high-level instructions that will challenge you to think carefully and to apply the procedures that are covered in this lesson, and elsewhere in the Training Kit. If you need assistance, consult the detailed, step-by-step instructions in the Practice Answers on the companion media.

In this practice, you configure common access and authentication scenarios that require the configuration of access mappings and zones.

Prepare for the Practice

Before you perform this practice, you must ensure that your lab environment has been built according to the instructions found in the Introduction to this Training Kit. You must also have performed the practice in Lesson 2 of this chapter. If you are currently logged on to SP2010-WFE1, log off before beginning the exercises.

EXERCISE 1 Modify Access Mappings

In this exercise, you

1. Log on to SP2010-WFE1 as **CONTOSO\SP_Admin** with the password **Pa$$w0rd**.
2. Add **http://intranet** as an internal URL to the default zone of the Contoso Intranet Web application.
3. Add **http://intranet** as a host header binding to the Contoso Intranet IIS Web site.
4. Confirm that you can open the site with the URL ***http://intranet***.

   The first time you open a site, IIS loads, compiles, and caches the site. This can take a while. If the site takes too long to load, an error appears. Refresh the page.
5. Observe that the URL to the home page in the address bar is *http://intranet.contoso.com/SitePages/Home.aspx*.

   A redirector loads the home page of the site. The redirector uses the public URL of the web application zone.

EXERCISE 2 Configure Windows-Claims Authentication

In this exercise, you configure authentication for the Contoso Partners Web application so that Windows Authentication is the only authentication provider. This exercise is intended to ensure that the web application is correctly configured for this practice.

- Verify that the Contoso Partner Portal Web application is configured with the following authentication settings:
  - NTLM authentication: Enabled
  - Forms Based Authentication: Disabled

EXERCISE 3 Extend a Web Application

In this exercise, you enable users to access the Contoso Partners Web application using *http://extranet.contoso.com* from the internal network and *https://partners.contoso.com* from the extranet. To do this, you extend the web application to a new zone for intranet users, with the URL *http://extranet.contoso.com*. Your information security manager has recommended that you use the host name *extranet* for your internal users so that it is clear to them that content in the web application is for external consumption.

1. Extend the Contoso Partner Portal to a new zone. Use the following specifications and guidance:

   - IIS Web site name: Contoso Partners Extranet
   - Port: 80
   - Host header: extranet.contoso.com
   - Authentication: NTLM
   - Public URL: *http://extranet.contoso.com:80*
   - Zone: Intranet

   It might seem counterintuitive to use the *intranet* zone for a zone with the URL *extranet*. Remember that the names of the zones (intranet, extranet, Internet, and custom) have no technical meaning. Furthermore, in this scenario, the zone is for internal users to access the Contoso Partner Portal. Access is from the intranet, using HTTP. The site is an external-facing site on which to collaborate with partners, thus the user-facing name of the zone is *extranet*.

2. Open a new tab in Internet Explorer, and browse to ***https://partners.contoso.com***. Sign in as **CONTOSO\SP_Admin**.

   The first time you open a site, IIS loads, compiles, and caches the site. This can take a while. If the site takes too long to load, an error appears. Refresh the page.

3. Open a new tab in Internet Explorer, and browse to ***http://extranet.contoso.com***.

   The Contoso Partner Portal site opens.

   The first time you open a site, IIS loads, compiles, and caches the site. This can take a while. If the site takes too long to load, an error appears. Refresh the page.

EXERCISE 4 Configure Authentication on a Zone

In this exercise, you enable anonymous users to access the root site collection of the Contoso Partner Portal as a landing page from which you can provide links to other sites that require authentication. So that users on non-Windows systems can authenticate to the portal, you will enable Basic authentication as well.

1. Enable anonymous authentication for the default zone of the Contoso Partner Portal Web application.
2. Enable Basic authentication for the default zone of the Contoso Partner Portal Web application.

EXERCISE 5 Configure Anonymous Access Restrictions

In this exercise, you enforce a security policy of your SharePoint governance plan that requires authenticated access to change any content. You do this by configuring an anonymous access restriction policy on the zone through which anonymous users are allowed to authenticate.

- Apply a Deny Write anonymous access restriction policy to the default zone of the Contoso Partner Portal Web application.

  You could apply the policy to all zones, but in this scenario, anonymous authentication is allowed only for the default zone.

EXERCISE 6 Complete and Validate Anonymous Access

In this exercise, you validate that users must be authenticated to access the Contoso Partner Portal site using the URL *http://extranet.contoso.com*, and that anonymous users can access the site using the URL *https://partners.contoso.com*.

1. Close all instances of Internet Explorer so that cached connections are eliminated.
2. Start Internet Explorer and browse to ***http://extranet.contoso.com***. Sign out of the site.

   A Windows Internet Explorer message opens: *The webpage you are viewing is trying to close the window*.

   Click Yes.

   Question: Why did the window close?
3. Start Internet Explorer and browse to ***https://partners.contoso.com***. When the Windows Security dialog box appears, click Cancel to log on as an anonymous user.

   A 401 Unauthorized error page opens.

   Question: Why can you not access the site as an anonymous user?
4. Refresh the page and sign in as **CONTOSO\SP_Admin** with the password **Pa$$w0rd**. Give anonymous users access to the entire site.
5. Close Internet Explorer. Start Internet Explorer and browse to ***https://partners.contoso.com***.
6. When the Contoso Partner Portal site opens, observe the Sign In control in the upper-right corner. You are connected as an anonymous user.

EXERCISE 7 Create the CHAPTER 03 Snapshot

The CHAPTER 03 snapshot captures the state of the environment at the end of Chapter 03. Perform this procedure for each of the following virtual machines: SP2010-WFE1, CONTOSO-DC.

1. Shut down the virtual machine.
2. Unmount any ISO image currently mounted to the CD/DVD drive. Use the “Unmount an ISO Image” procedure in the Lab Environment Build Guide on the companion media.
3. Create a snapshot named CHAPTER 03. Use the “Create a Snapshot” procedure in the Lab Environment Build Guide on the companion media.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 3, "Configure Authentication Zones and Alternate Access Mappings." The questions are also available on the companion media in a practice test if you prefer to review them in electronic form.

> ***NOTE*** **ANSWERS**
>
> **Answers to these questions and explanations of why each answer choice is right or wrong are located in the "Answers" section at the end of the book.**

1. You have created a SharePoint-based timecard application, *http://timecards.contoso.com*. You want users to be able to browse to the application using either *http://timecards.contoso.com* or *http://timecards*. What steps must you take? (Choose two. Each correct answer is a part of the solution.)

   A. Modify the IIS Web site.

   B. Modify the Web.config file.

   C. Modify the managed paths.

   D. Modify the Alternate Access Mappings.

   E. Extend the web application.

2. You have created a web application with the URL *http://server1*. Users can access the application from systems connected to the corporate network, and can authenticate with Windows authentication. You want to allow users to access the application from external systems with the URL *https://server1.contoso.com*. What do you do? (Choose all that apply. Each correct answer is a part of the solution.)

   A. Extend the web application to a new IIS Web site.

   B. Enable SSL for the web application.

   C. Add a binding to an IIS Web site.

   D. Install a certificate on the web server.

3. You have created a web application with the URL, *http://partners.contoso.com*. You want users to access the web application as *https://partners.contoso.com*, through a device that will offload the processing of SSL. What do you do?

   A. Add a host header, *https://partners.contoso.com*, to the IIS Web site.

   B. Add an SSL binding to the IIS Web site.

   C. Modify Alternate Access Mappings in Central Administration.

   D. Extend the web application to create a new zone.

4. You have created a web application, *http://intranet.contoso.com*. Users access the intranet and make changes to content while connected to the corporate network. You want to provide access to the site from outside the corporate network, but you want to ensure that users accessing the site from outside cannot change content. What do you do? (Choose all that apply. Each correct answer is a part of the solution.)

   A. Extend the web application to the extranet zone.

   B. Configure user policy on the extranet zone.

   C. Set the content database to read-only.

   D. Specify permissions on the top-level site collection that allow only read access.

   E. Configure a Deny Write anonymous access restriction.

5. You have created a web application with the URL *http://server1*. The application uses Windows authentication. You want to allow anonymous connections through the URL *https://server1.contoso.com*. What must you do? (Choose all that apply. Each correct answer is a part of the solution.)

   A. Enable anonymous authentication on the default zone.

   B. Add an SSL binding to the IIS Web site.

   C. Modify Alternate Access Mappings in Central Administration.

   D. Extend the web application to create a new zone.

   E. Modify anonymous access restrictions.

   F. Modify the authentication provider.

Chapter Review

To further practice and reinforce the skills you learned in this chapter, you can perform the following tasks:

- Review the chapter summary.
- Review the list of key terms introduced in this chapter.
- Complete the case scenarios. These scenarios set up real-world situations involving the topics of this chapter and ask you to create a solution.
- Complete the suggested practices.
- Take a practice test.

Chapter Summary

- The logical components that allow SharePoint to receive and process a request for content are the web application itself, its five zones, the alternate access mappings associated with each zone, the IIS Web site associated with each zone, and the bindings on the website.
- A user submits a request using the public URL of the web application zone. The request can be modified by an intermediary device, such as an off-box SSL terminator or a reverse proxy, before being forwarded to the front-end web server. The request received by IIS is matched to an IIS Web site based on the site's bindings, which often are based on a host header or, in the case of SSL, a dedicated IP address. The request is then passed to SharePoint, which examines the request's URL and, by identifying a matching the URL with the internal URLs in the web application's Alternate Access Mappings (AAM) collection. The matching URL identifies the zone with which the request will be processed. The zone determines the authentication and policies applied to the request.
- Authentication is managed by one of three authentication providers: Windows, forms based authentication, and SAML-token based authentication. In Classic Mode Authentication, only Windows is supported, but in Claims Based Authentication, all three providers are supported, and you can use multiple providers in a single zone.
- You can also enable anonymous authentication on a zone. However, a site collection administrator must also enable anonymous access and assign anonymous users permissions to content within a site. You can use enforce restrictions on the maximum access granted to anonymous users on a per-zone basis.
- As you design your environment, you must be aware of which settings are scoped to a web application, to individual zones, and to IIS Web sites. This will help you determine the logical architecture that will meet your requirements.

Key Terms

The following terms were introduced in this chapter. Do you know what they mean?

- bindings
- Web.config

- application pool
- Classic Mode Authentication
- Claims Based Authentication
- Secure Sockets Layer (SSL) and Transport Layer Security (TLS)
- Kerberos delegation
- Service principal name (SPN)
- Claim
- SAML token
- IP-STS
- RP-STS
- Zone
- Alternate access mapping

Case Scenario: Troubleshooting Web Application Configuration

In this case scenario, you apply what you've learned about subjects of this chapter. You can find answers to these questions in the "Answers" section at the end of this book.

You have recently begun working at Contoso, Ltd. The previous SharePoint administrator, who is no longer with the company, created an intranet web application. Users can access the web application by typing *http://intranet.contoso.com* or *http://intranet*. You have been told the intranet was configured so that users could type either URL to make it easier for users.

But users are complaining. They report that if they access the site as *http://intranet*, they have to click the Sign In link in the upper-right corner of the page before they can see certain content. If they access the site as *http://intranet.contoso.com*, they see all of their content immediately.

1. What can cause SharePoint to display a Sign In link? Why are users seeing the Sign In link?
2. What might the previous administrator have done that would cause different behavior for *http://intranet* than *http://intranet.contoso.com*?
3. You want to fix this problem, so that users can access the web application using either *http://intranet* or *http://intranet.contoso.com* and immediately see all of their content. You also want to correct configuration that was made by the previous administrator. Describe the tasks you will perform to resolve the situation.

Suggested Practices

To help you successfully master the exam objectives presented in this chapter, complete the following tasks.

Manage Web Applications

Do all the practices in this section. Be certain that you have created a snapshot of your virtual machines prior to performing these practices. When you have completed the practices, revert to the snapshot.

Practice 1: Configure Network Load Balancing

Network Load Balancing (NLB) allows you to distribute requests across multiple web front-end servers (WFEs). NLB can improve performance, and provides redundancy so that if a web server fails, other WFEs can continue to service requests. Windows Server 2008 R2 supports Network Load Balancing. Add the NLB feature to SP2010-WFE1. Even though you have only one server in the farm, you can configure NLB with the cluster IP address—the address that will be exposed to end users—and the IP address of the server. For example, you can configure the cluster IP address as 10.0.0.20/255.255.255.0. Add the IP address of SP2010-WFE1 (10.0.0.21) as a member of the cluster. Finally, change the DNS records for one or more websites to resolve to the IP address of the cluster. Test the configuration by browsing to a website. If it renders, NLB has been configured successfully. Because Windows Server 2008 R2 NLB does not overwrite the inbound request—instead, it sends the request to the IP address of a member—you do not need to configure any zones or access mappings to support this scenario.

Practice 2: Configure Forms Based Authentication

In the practice of Lesson 2, you began to configure Forms Based Authentication (FBA). You configured the web application for FBA, and you examined the Web.config file of the IIS Web site, but you did not change the three Web.config files that are necessary to configure FBA. You also did not create a SQL database to test the functionality of FBA. FBA is one of the most complex configurations in SharePoint. It is highly recommended that you gain hands-on experience with the process. Use resources on Microsoft TechNet and elsewhere on the Internet to configure Forms Based Authentication. Be certain to create a snapshot before you begin, so that you can revert to a known-good state when you have finished this suggested practice.

Take a Practice Test

The practice tests on this book's companion media offer many options. For example, you can test yourself on just the lesson review content, or you can test yourself on all the 70-667 certification exam objectives. You can set up the test so that it closely simulates the experience of taking a certification exam, or you can set it up in study mode so that you can look at the correct answers and explanations after you answer each question.

MORE INFO **PRACTICE TESTS**

For details about all the practice test options available, see the "How to Use the Practice Tests" section in this book's Introduction.

Index

About the Authors

DAN HOLME As Chief SharePoint Evangelist at AvePoint, Dan Holme utilizes both his expertise in Microsoft technologies and proven experience solving customers' IT business challenges to educate the global SharePoint community, as well as develop solutions that will set the standard for the next generation of collaboration platforms.

A graduate of Yale University and Thunderbird School of Global Management, Dan spent 17 years as a consultant and trainer, delivering solutions to tens of thousands of IT professionals from the most prestigious organizations and corporations around the world.

In addition to earning the prestigious title of Microsoft MVP (Windows Server Directory Services, 2007, and SharePoint Server, 2008-2011), Dan has been recognized as one of the Top 50 Influencers by The SharePoint50 Project and one of the top 10 Microsoft Partner MVPs. Dan is a contributing editor for *Windows IT Pro* and *SharePoint Pro* magazines as well as the community lead of SharePointProMag.com, and has authored several books and courses, including training kits and Microsoft Official Curriculum courses for Active Directory and SharePoint.

Prior to joining AvePoint, Dan founded Aptillon, a SharePoint consulting and development firm, with seven of the world's top SharePoint MVPs. He also played an instrumental role as Microsoft Technologies Consultant for NBC Olympics during the Winter Olympics in Vancouver (2010), Beijing (2008), and Torino (2006)—a role he plans to play again for the broadcast of the 2012 Summer Olympics from London.

ALISTAIR MATTHEWS A consultant with extensive and cutting-edge experience in Microsoft technologies, Alistair has spent the last 10 years developing with, consulting on, and communicating about both the developer and IT professional sides of SharePoint, Visual Studio, Active Directory, Exchange, and Windows. He is currently most interested in SharePoint Web Content Management and likes to impress clients with elegant publishing workflows and custom UI elements. He's also more excited about Office 365 than he cares to admit.

Alistair has a particular passion for writing about technology and has contributed to many Microsoft Learning courses, MSDN and TechNet articles, and white papers. He is the principal consultant at Web Dojo Ltd and lives the telecommuting dream in Cornwall, UK.