

Publishing PerformancePoint Server in Extranet Scenarios

Prepared for:
Microsoft

July 10, 2009

Prepared by:
Nelson Puello
Eric Melcher
Dean Mosley
Leo Furlong



Two Concourse Pkwy
Atlanta, GA 30328
404.442.8000
fax 404.442.8001
www.intellinet.com

Contents

INTRODUCTION	1
BACKGROUND	1
BUSINESS SCENARIOS FOR EXTRANETS	1
CHALLENGES SUPPORTING EXTRANETS	2
COMMON EXTRANET DEPLOYMENTS	2
VPN	2
Web Publishing	3
Reverse Proxy	4
TECHNOLOGY OVERVIEW	6
ACTIVE DIRECTORY AND KERBEROS	7
Service Principal Names	8
NTLM vs. Kerberos	11
Kerberos Constrained Delegation	11
WINDOWS SHAREPOINT SERVICES 3.0/OFFICE SHAREPOINT SERVER 2007	14
Web Application Authentication Providers	14
Application Pool Identity	15
Web.config file	15
SharePoint Site Collection Security	16
Alternate Access Mappings	16
SQL SERVER AND SQL ANALYSIS SERVICES	17
Analysis Services Service	18
SSAS Service Principal Name	18
PERFORMANCEPOINT SERVER (PPS) 2007	19
Service Account (Default)	19
Connection per User	19
Custom Data	22
Connection Roles	23
PPS Connection Architecture	25
Impact of Connection Settings	26
ISA SERVER	26
Publishing Rules	27
ENVIRONMENT CONFIGURATION STEPS	30
LAB OVERVIEW	30
ACTIVE DIRECTORY, KERBEROS, AND CONSTRAINED DELEGATION	33
MICROSOFT OFFICE SHAREPOINT SERVER 2007	35
Web Application Configuration	35
Site Collection Configuration	38
Alternate Access Mappings	38
SQL SERVER 2008 AND ANALYSIS SERVICES	41
Create SSAS SPN	41
PERFORMANCEPOINT SERVER 2007	42
Office SharePoint Server Connection per User	42
Dashboard Designer Connection Per User	44
Preview Site Connection Per User	45
ISA SERVER 2006	46
TROUBLESHOOTING	57
COMMON KERBEROS ISSUES	59
KERBEROS TROUBLESHOOTING STRATEGIES	60
SUPPORTED CONFIGURATIONS	61
SERVER OPERATING SYSTEM VERSIONS (NON DOMAIN CONTROLLERS)	61
SERVER OPERATING SYSTEM VERSIONS (DOMAIN CONTROLLERS)	61

WINDOWS SHAREPOINT SERVICES VERSIONS	61
SQL SERVER ANALYSIS SERVICES VERSIONS (CUBES)	61
CLIENT WEB BROWSERS	62
REVERSE PROXY	62
CONCLUSION	63
REFERENCES	64
GLOSSARY.....	64

Tables

TABLE 1 - PERFORMANCEPOINT SERVER TECHNOLOGY REQUIREMENTS	6
TABLE 2 - SETSPN USAGE WINDOWS SERVER 2003	9
TABLE 3 - SETSPN USAGE WINDOWS SERVER 2008	9
TABLE 4 - SPNS FOR SHAREPOINT AND PERFORMANCEPOINT	10
TABLE 5 - KCD CONFIGURATION FOR PERFORMANCEPOINT	13
TABLE 6 - AUTHENTICATION AND VALIDATION COMBINATIONS FOR KCD	28
TABLE 7 - LAB CONFIGURATION	31
TABLE 8 - LAB SERVICE ACCOUNTS	33
TABLE 9 - LAB REQUIRED SPNS	33
TABLE 10 - TROUBLESHOOTING LOGS AND TOOLS	57

Figures

FIGURE 1 - VPN EXTRANET	3
FIGURE 2 - WEB PUBLISHING EXTRANET	4
FIGURE 3 - WEB PUBLISHING WITH REVERSE PROXY	5
FIGURE 4 - PERFORMANCEPOINT SERVER EXTRANET TECHNOLOGY OVERVIEW	7
FIGURE 10 - ALTERNATE ACCESS MAPPING COLLECTION	16
FIGURE 11 - PUBLIC URLS	17
FIGURE 12 - SSAS SERVICE ACCOUNT	18
FIGURE 13 - SSAS SPN EXAMPLE	19
FIGURE 5 - PPS SERVERCONNECTIONPERUSER WEB.CONFIG SAMPLE	20
FIGURE 6 - USEASCUSTOMDATA WEB.CONFIG	23
FIGURE 7 - ROLES WITHIN A PPS DATA SOURCE.....	24
FIGURE 8 - PPS CONNECTIVITY ARCHITECTURE	25
FIGURE 9 - PPS DATA SOURCE CACHE SETTING	26
FIGURE 14 - DELEGATION OF CREDENTIALS IN A PUBLISHING RULE	28
FIGURE 15 - LAB ENVIRONMENT.....	30

Introduction

Background

With the release of PerformancePoint Server 2007 Monitoring and Analytics, Microsoft's Business Intelligence (BI) platform enables enterprises to develop and deploy web based dashboards, scorecards and analytics through SharePoint Products and Technologies. (In this article, "SharePoint Products and Technologies" refer collectively to Windows SharePoint Services 3.0 and Office SharePoint Server 2007.) As Microsoft continues to add functionality and depth to its platform, companies are looking at more advanced ways to use Business Intelligence within their organizations. Historically, most Business Intelligence solutions are developed for use within the organization, and are consumed over the intranet by company employees. There are, however, many situations where a Business Intelligence solution can provide significant value by allowing users external access, over the Internet.

Business Scenarios for Extranets

A common scenario requiring extranet access for Business Intelligence is for remote employees who typically don't connect directly to the corporate network, but still need easy access to certain corporate resources. Similar to many web-based e-mail applications, users in the field need to be able to access a Business Intelligence portal securely over the Internet to obtain information needed to perform their jobs or make decisions. This is common for outside sales representatives who are in the field and need information about their individual performance or the performance of their customers.

Another typical scenario is for organizations that need to disseminate information to their customers or other individuals outside their organization. For some companies this may be a revenue generating opportunity where consumer, statistical, or market data is provided to customers as a paid service. Other companies may provide services to their customers, and as part of those services, collect data on behalf of their customers. An example of this would be a company that provides call center outsourcing. This company would use their own internal call center systems to manage the call center, and collect call-based data. That information would then be stored in their internal systems, but need to be available for their customers to access and analyze.

In both scenarios, Business Intelligence solutions can significantly streamline processes and improve information access for users not connected to the corporate network. Manual processes for report generation and distribution via e-mail and file shares can be eliminated, and external users can be given a self-service portal for finding and consuming the information they need.

In this white paper, we will illustrate Microsoft's supported architecture and configuration for extranet Business Intelligence solutions using Microsoft Office PerformancePoint Server 2007 Monitoring and Analytics. This will include an overview of Microsoft's supported architecture, essential technologies, and the steps for configuring a solution. This is a technical white paper intended for network and application engineers and administrators.

Challenges Supporting Extranets

Microsoft has always emphasized integration between products, allowing enterprises to gain additional value by re-leveraging existing technologies for new solutions. Instead of creating a new means of authentication for its Business Intelligence platform, Microsoft relies upon the Active Directory directory service to provide authentication and security. This gives enterprises access to a robust set of tools for managing users and authentication with little impact to existing architectures and processes. This integration leads to a seamless experience for internal users who already have an Active Directory account, and who access the network on a daily basis to perform their job.

However, providing access to a Business Intelligence solution externally over the Internet can be a bit more complicated. One challenge is that users coming from outside the internal network are not authenticated. This means that the identity of the user has not been validated, which prevents the user from being able to access secured resources. Additionally, users that are truly external to an organization, i.e., a customer's employees, are unlikely to have an account on the internal network. What most organizations require in an extranet scenario is to be able to control the data available to each individual user. This requires that each user is uniquely authenticated, and that user credentials are passed through to Analysis Services where data security is applied.

Fortunately, these challenges around extranet solutions can be addressed in a number of ways using Microsoft technologies. Some of the more common extranet architectures are outlined below including Microsoft's supported extranet architecture for Business Intelligence solutions. This supported architecture will be described in more detail throughout the course of this whitepaper.

Common Extranet Deployments

An extranet can be defined as a company's privately held network, similar to an Intranet, which allows access to certain others such as customers or business partners. Extranets are unique in that they are private in nature, yet they are typically accessible over the public Internet. Since intranets are also usually associated with websites, extranets can be viewed as publicly reachable websites which are made accessible only to a select group of individuals.

Extranets can be implemented in a variety of ways and using a myriad of technologies. The most common ways of deploying extranets are presented below.

VPN

Virtual Private Networks are comprised of protocols that encrypt and encapsulate TCP/IP packets as they are transmitted over the public Internet. This "tunnel" created over public networks enables only the sender and the intended receiver to interpret the information that is being transmitted. VPNs are a very common way for remote users to access their Local Area Network resources, and for providing access to corporate resources to business partners and customers.

Advantages

- Widely used technology
- Most operating systems offer VPN clients out of the box
- Published resources can remain in the LAN where there are fewer restrictions for data access and management

Disadvantages

- Requires client installation and/or configuration. Even with most web browser based SSL VPNs, an ActiveX control or Java component must be installed
- Compatibility problems are common

Figure 1 below illustrates a typical VPN deployment.

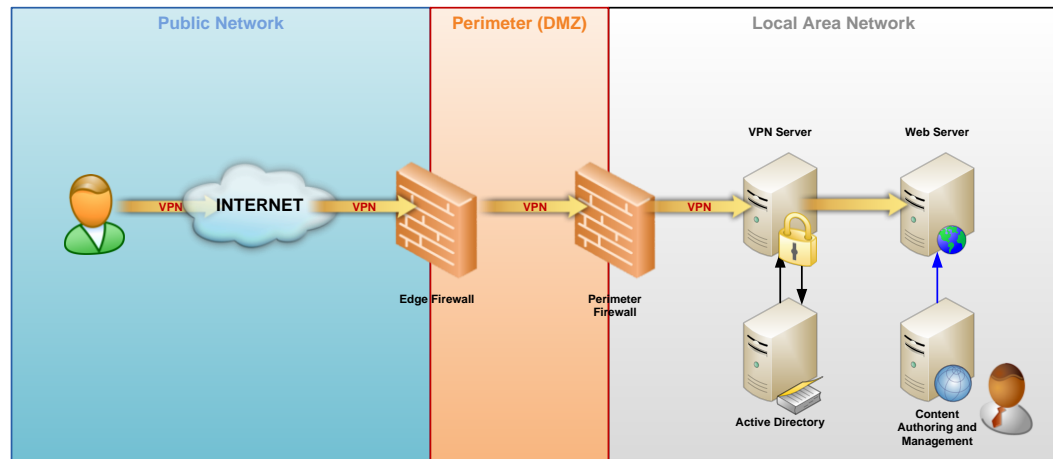


Figure 1 - VPN Extranet

Web Publishing

Another common method of publishing websites to an extranet is to place web servers in the perimeter network or Demilitarized Zone (DMZ). In this way web servers can be directly accessible from public networks without the need of establishing a VPN first. Access to the web servers is controlled with access accounts, the source of which range from account databases local to the web servers to Active Directory and other LDAP-compliant sources. The exchange of information between the clients in the public network and the web servers is encrypted using Secure Sockets Layer (SSL).

Advantages

- Web servers can be accessed securely without the need for VPN.
- Placing the web servers in the perimeter network minimizes the risk of direct access to the corporate LAN.
- Simple solution to deploy and maintain.

Disadvantages

- In many situations it may be necessary to open several ports in the perimeter firewalls to enable access to data sources or authentication services such as Active Directory, thus increasing the risk of security breaches.
- Management of the web servers can be cumbersome because of the security restrictions placed in the perimeter network.
- Authentication sources based on local security accounts or other providers such as SQL Server databases do not provide Kerberos authentication and delegation.
- Implementing dedicated perimeter security providers such as a dedicated Active Directory forest adds complexity and management overhead.

Figure 2 illustrates a typical web publishing scenario.

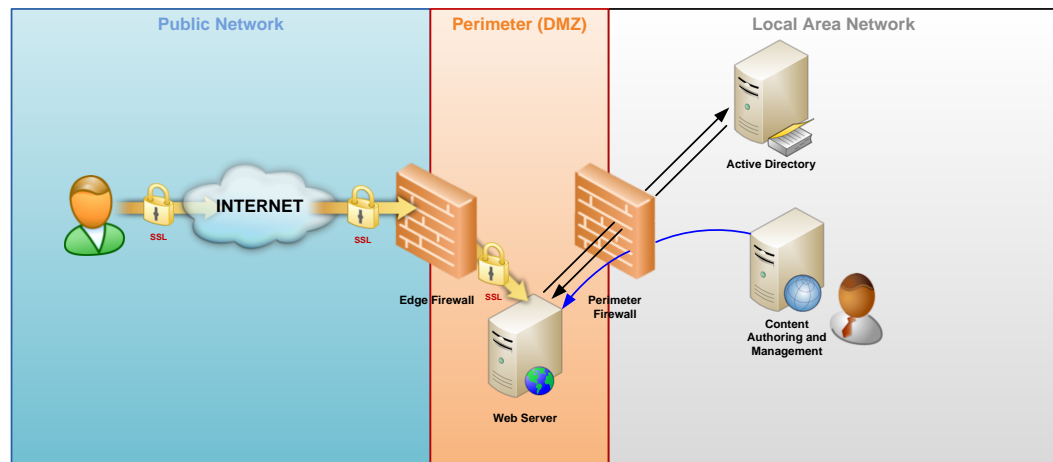


Figure 2 - Web Publishing Extranet

Reverse Proxy

The concept of a reverse proxy is that of a device or service which stands between the users in the public network and the LAN resources, and which securely accesses the resources in the LAN on behalf of the user. The connection through the proxy is seamless from the perspective of the user. There are significant advantages to using a reverse proxy such as Microsoft ISA Server as compared to other forms of web publishing.

Advantages

- Pre-authentication of user accounts prior to granting access
- Support for multiple authentication protocols and form factors such as Forms Based authentication, RSA SecurID, RADIUS, and Client Certificates.
- Application-layer filtering
- Integration with Active Directory and support for Kerberos authentication and Constrained Delegation
- Provides a seamless experience to the external users
- Web servers can remain in the LAN where they have direct access to data and authentication sources, and where they can be directly managed

Disadvantages

- The tradeoff for seamless access is some complexity in the configuration of authentication

Figure 3 illustrates the publishing of a web server through ISA Server.

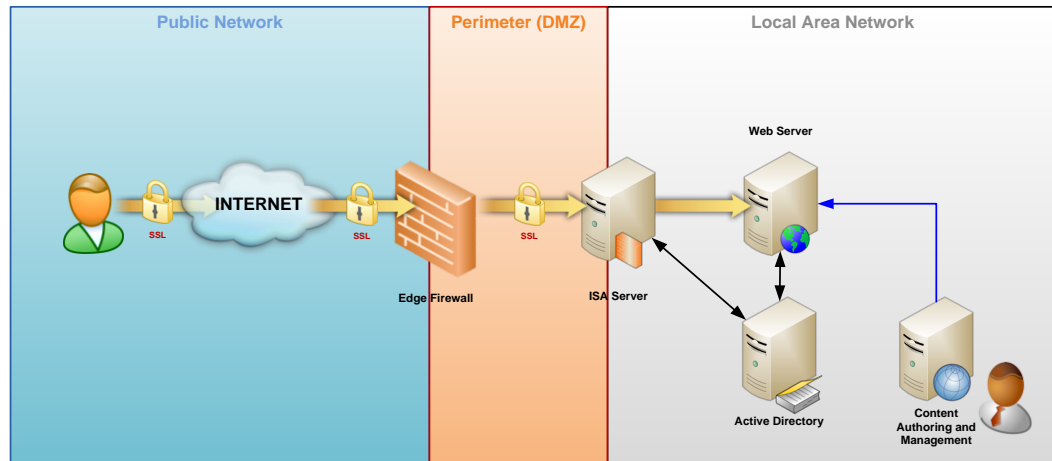


Figure 3 - Web Publishing with Reverse Proxy

Due to the security requirements of PerformancePoint Server 2007, which will be covered in a later section, the focus of this whitepaper is on the reverse proxy extranet deployment scenario.

Technology Overview

Microsoft's supported extranet architecture for Business Intelligence requires the integration of a number of technologies. Before diving into the step-by-step procedures for configuring each component of the overall solution, it is best to understand the role of each technology, the configurations and changes that need to be made to support authentication in an extranet scenario, and their implications. This will provide the context for the step-by-step procedures that will follow in the "Environment Configuration Steps" section.

Deploying PerformancePoint Server 2007 dashboards in an extranet environment requires the deployment and configuration of several components based on Microsoft Server technologies. These technologies and their roles are presented in Table 1.

Table 1 - PerformancePoint Server Technology Requirements

Technology	Role
Windows Server 2003, or Windows Server 2008	The core Operating System for all technologies involved.
Windows SharePoint Services 3.0, or Microsoft Office SharePoint Server 2007	The presentation platform for PerformancePoint Server 2007. Dashboards are web parts embedded in SharePoint Products and Technologies pages.
Microsoft Office PerformancePoint Server 2007	The presentation engine providing web based analytics. It includes a content authoring/publishing interface called Dashboard Designer, as well as a set of web parts for displaying content through SharePoint Products and Technologies.
Microsoft SQL Server 2005/2008	The database system which provides the content for SharePoint sites.
Microsoft SQL Server 2005/2008 Analysis Services	The primary engine that supports advanced analytics within Microsoft's BI platform. Analysis Services provides multi-dimensional cubes enabling advanced calculations, drill-downs, and navigation of data.
Microsoft Internet Security and Acceleration Server 2006	The reverse proxy server which publishes the SharePoint sites to the public Internet.
Active Directory	The core Directory Services of Windows Server. Provides the

Technology	Role
	security subsystem for the entire solution.
Kerberos	The authentication mechanism built into Windows Server Active Directory.

Figure 4 illustrates a technological overview of a PerformancePoint Server 2007 solution.

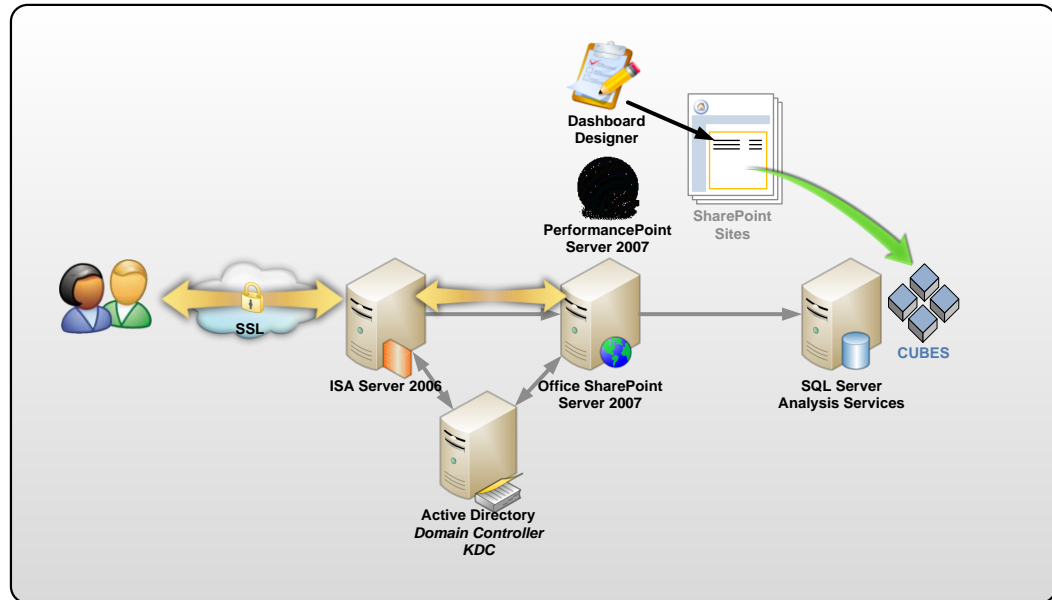


Figure 4 - PerformancePoint Server Extranet Technology Overview

Active Directory and Kerberos

Kerberos is the preferred mechanism for authentication in Active Directory. The name Kerberos (or Cerberus) comes from the three-headed dog in Greek Mythology that guarded the gates of Hades. Kerberos is a mutual authentication mechanism that builds on symmetric key cryptography and which requires that two entities both trust a third party authority in order to authenticate with each other, hence the concept of three heads. Although the purpose of this white paper is not to provide an in-depth technical explanation of how Kerberos works, an overview of how it is implemented in Active Directory is important here. For a detailed explanation of Kerberos in Active Directory see <http://technet.microsoft.com/en-us/library/cc739058.aspx>.

Note: The proper configuration of Kerberos in Active Directory must be completed prior to the configuration of PerformancePoint Server 2007 for role-based access to SQL Server Analysis Services cubes. Implementing the changes to PerformancePoint Server 2007 presented in this document without properly configuring Kerberos first will cause access to the SQL Server Analysis Services cubes to fail. The configuration of PerformancePoint Server 2007 is covered in a later section in this document.

As it is implemented in Windows 2000 and greater versions of Active Directory, Kerberos authentication relies on service tickets which are passed between Active Directory accounts to mutually authenticate and to establish secure communication sessions between clients and servers. In Active Directory, the three “heads” are comprised of a client, a server or service, and the Key Distribution Center (KDC) which runs on Domain Controllers.

The KDC for its part is comprised of an Authentication Service (AS) and a Ticket Granting Service (TGS). The AS is responsible for verifying the client’s identity in Active Directory, and the TGS is responsible for provisioning the tickets. In order for a client to request tickets from the KDC, it first must establish a secure session with the KDC by passing its domain credentials to the AS and requesting a Ticket Granting Ticket (TGT). A TGT is a special ticket that authorizes the client to request tickets from the TGS for other servers or services. At a high-level, the interaction between a client, the KDC, and a server takes place as follows:

1. The client passes its domain credentials to the KDC and requests a TGT.
2. The KDC’s Authentication Service validates the credentials against Active Directory, and if the credentials are valid it gives the client a TGT.
3. The client uses its TGT to request a service ticket from the Ticket Granting Service for the server with which it intends to communicate.
4. The TGS gives the client a service ticket for the server it requested.
5. The client presents the service ticket to the server and requests authentication.
6. The server authenticates the user and a secure session is established between the client and the server.

The process of Kerberos authentication is actually much more complex than the above depiction —there is a great deal of encryption and decryption that takes place in the midst of it all, and there is also a time-stamp component that is critical for Kerberos to work. None of those details are covered in this white paper. Nevertheless, one key item to note here as it pertains to configuring Kerberos for PerformancePoint 2007 is that tickets are intended for Security Principals (user and computer accounts). That is, when a client requests a ticket from the TGS, it does so as an Active Directory user or computer, and the intended recipient of the ticket is also an Active Directory user or computer. The ticket granted by the TGS is intended for establishing secure communications only between the requesting client and its intended recipient. A client must request a ticket for each Security Principal with which it intends to communicate, therefore a client can have multiple tickets simultaneously.

Service Principal Names

When a client requests a ticket from the TGS, it passes the name of the intended recipient in the form of a Service Principal Name (SPN). An SPN specifies a service type being hosted by the server or service, and a host name in the form <SERVICETYPE>/<HOSTNAME>. For example, if the intended recipient of a service ticket is a web server named WEB1, the client will request a ticket using the SPN HTTP/WEB1. If a client makes a request to access a SQL server named sql1.contoso.com, the SPN would be MSSQLSVC/sql1.contoso.com. MSSQLSVC is the service type, and sql1.contoso.com is the name of the server hosting the service.

When the TGS receives the request from the client, it searches Active Directory for a user or computer object with a ServicePrincipalName attribute matching the SPN passed by the client, and then uses the user or computer object name to form the service ticket. For this reason SPNs must be unique, meaning, any given SPN should be registered to only one Active Directory user or computer object. The ServicePrincipalName attribute is blank by default for Active Directory user objects. Computer objects, however, do get an SPN of the service type HOST when the computer is joined to an Active Directory domain. This enables Kerberos to work

seamlessly in the domain environment as long as the computer name is used when requesting a service. When the name used to request a service is different than the host name, there must be a matching SPN registered to the computer or user account responsible for the service. What this means is that when a service runs under the context of a user account instead of the Network Service or Local Service accounts, a SPN with the service type and corresponding host name must be registered for the user account.

Adding a SPN to a user or computer account in Active Directory can be achieved in a number of ways. ServicePrincipalName is a multi-value string attribute of user and computer objects, and it can be modified via Active Directory Services Interface (ADSI) programming through scripting. It can also be modified by directly editing Active Directory using the ADSIEDIT management console snap-in for Windows Server. The preferred way of modifying this attribute is to use the SETSPN.EXE utility which is included as part of the Windows Server 2008 Operating System, and the Windows Server 2003 Support Tools. SETSPN is a command line tool with built-in error checking for improperly configured SPNs. Although the Windows Server 2008 version of the tool includes the ability to search for duplicate SPNs, the Windows Server 2003 Support Tools version does not. The following tables show the usage of the SETSPN tool.

Table 2 - SETSPN Usage Windows Server 2003

Usage	<i>setspn [switches data] computername or username</i> The computername or username name can be in the form DOMAIN\name
Switches	-R Reset HOST ServicePrincipalName Usage: <i>setspn -R computername</i> -A Add arbitrary SPN Usage: <i>setspn -A SPN computername</i> -D Delete arbitrary SPN Usage: <i>setspn -D SPN computername</i> -L List registered SPNs Usage: <i>setspn -L computername</i>
Examples	<i>setspn -A HTTP/web1.contoso.com CONTOSO\WEB1</i> Registers SPN "HTTP/web1.contoso.com" for computer "WEB1" in the CONTOSO domain. <i>Setspn -D MSSQL/sql1.contoso.com CONTOSO\SQLSvc</i> Deletes SPN "MSSQL/sql1.contoso.com" for service account "SQLSvc" in the CONTOSO domain.

Table 3 - SETSPN Usage Windows Server 2008

Usage	<i>setspn [modifiers switches data] computername or username</i> The computername or username can be in the form DOMAIN\name
Modifiers	-F Perform the duplicate checking on forest wide level -P Do not show progress (useful for redirecting output to a file)
Switches	-R Reset HOST ServicePrincipalName Usage: <i>setspn -R computername</i>

	<p>-A Add arbitrary SPN Usage: setspn -A SPN computername</p> <p>-S Add arbitrary SPN after verifying no duplicates exist Usage: setspn -S SPN computername</p> <p>-D Delete arbitrary SPN Usage: setspn -D SPN computername</p>
	<p>-L List registered SPNs Usage: setspn -L computername</p> <p>-Q Query for existence of SPN Usage: setspn -Q SPN</p> <p>-X Search for duplicate SPN Usage: setspn -X</p>
Examples	<p>setspn -A HTTP/web1.contoso.com CONTOSO\WEB1 Registers SPN "HTTP/web1.contoso.com" for computer "WEB1" in the CONTOSO domain.</p> <p>setspn -D MSSQL/sql1.contoso.com CONTOSO\SQLSvc Deletes SPN "MSSQL/sql1.contoso.com" for service account "SQLSvc" in the CONTOSO domain.</p> <p>setspn -F -S HTTP/web2.contoso.com CONTOSO\WEB2 Registers SPN "HTTP/web2.contoso.com" for computer "WEB2" if the SPN does not exist in the forest.</p>

The following table shows the accounts that would be needed in a deployment of PerformancePoint 2007 and the corresponding SPNs that must be registered.

Table 4 - SPNs for SharePoint and PerformancePoint

Active Directory Account	Purpose	Service Principal Names	Examples
SharePoint Web Application Pool Account	Serves as the Identity for the application pool which hosts the SharePoint Products and Technologies web application where PerformancePoint dashboards are deployed. The worker process (w3wp.exe) corresponding to this application pool runs under the context of this account.	<ul style="list-style-type: none"> • HTTP/<fully qualified domain name of web application URL> • HTTP/<host only part of URL> 	HTTP/portal.contoso.com HTTP/PORTAL

Active Directory Account	Purpose	Service Principal Names	Examples
Monitoring Server Application Pool Account ¹	Serves as the Identity for the PPS Monitoring Web Service and PPS Monitoring Preview application pools.	<ul style="list-style-type: none"> • HTTP/<fully qualified domain name of PPS Monitoring Server:Port_Number> • HTTP/<host name:Port_Number> 	HTTP/pps.contoso.com:4000 HTTP/PPS:4000
SQL Server Analysis Services Account	The SQL Server Analysis Services service runs under the context of this account	<ul style="list-style-type: none"> • MSOLAPSvc.3/<fully qualified domain name of SQL Server Analysis Server> 	MSOLAPSvc.3/sql1.contoso.com

¹ The configuration of PerformancePoint Monitoring Web Services for Kerberos authentication and delegation is not covered in this document. The Monitoring Web Services are required for the Dashboard Designer to work. The Dashboard Designer is not supported for publishing to an extranet. For more information on configuring Kerberos for the Monitoring Web Services see <http://technet.microsoft.com/en-us/library/bb838742.aspx>.

By enabling Kerberos authentication at the web application level and registering a SPN to the application pool identity account with the service type HTTP and the URL of the web application as the host, clients can use Kerberos to authenticate against SharePoint Products and Technologies. If the SPN is either configured incorrectly or missing, or if the web application authentication provider is not set to Kerberos, the authentication between the client and SharePoint Products and Technologies will default to NTLM.

NTLM vs. Kerberos

The NTLM protocol has a limitation in that it was not designed to transition authentication beyond two parties. In other words, if a server authenticates a client and they both establish a secure session between each other, the server cannot in turn pass the client credentials to another server to establish a secure session on behalf of the user. This is known as an authentication double-hop, and it is a very common scenario in web applications where web servers need to access back end database systems. The combination of SharePoint Products and Technologies, PerformancePoint, and SQL Server Analysis Services is such an example. There are ways to make up for this protocol shortcoming, such as implementing a programming technique called Impersonation. However this technique does not scale well. Kerberos, on the contrary, does support the passing of user credentials through multiple hops thanks to the Constrained Delegation protocol extension of Active Directory.

Kerberos Constrained Delegation

The implementation of Kerberos in Active Directory for Windows Server 2000 includes a feature which enables servers or services to request service tickets on behalf of other users or services, and to present those tickets to any additional servers or services. This is known as Kerberos Delegation. Windows Server 2003 Active Directory introduced a more secure approach to delegation where security principals are trusted to delegate to specific services instead of to any service. This is known as Kerberos Constrained Delegation (KCD). When KCD is configured for a computer or user account, Active Directory trusts that user or computer to delegate to specific user or computer accounts and to specific SPNs. KCD can be configured to use any

authentication protocol (Protocol Transition is allowed), or to use Kerberos only for delegation (Protocol Transition is disallowed). When Protocol Transition is allowed for a service trusted for delegation, the service can obtain a Kerberos ticket on behalf of the client without the client having to present credentials to the KDC. Protocol Transition is vital in extranet scenarios where alternate non-Kerberos forms of authentication are used such as client certificates and forms-based authentication.

Properly configuring KCD is vital to meet the PerformancePoint requirements for role-based access. The identity of the client (the user account in this case) must be successfully passed between the client and the server running SharePoint Products and Technologies and subsequently from that server to the SQL Server Analysis Services databases. Knowing the points of authentication in a Windows Server infrastructure helps determine which user or computer accounts must be configured as Trusted for Delegation in Active Directory. For example, when connecting to a web server, the point of authentication is the web application, which is executed under the identity of an application pool. If the web application needs to make a connection to a SQL server database on behalf of the user, then the account (identity) for the application pool in Active Directory must be configured as Trusted for Delegation to the SQL service account. If the identity is the Network Service account, then the computer account in Active Directory is the object that needs to be configured for Kerberos Constrained Delegation to the SQL Service account. In a PerformancePoint extranet deployment, a typical KCD configuration would be as follows:

Table 5 - KCD Configuration for PerformancePoint

Server or Service (point of authentication)	Active Directory Account	Trusted for Delegation to ...	Type of KCD
ISA Server	ISA Server Computer account	The ISA Server computer account does not need to be trusted for delegation. The only exception is when Protocol Transition is required because alternative non-Kerberos authentication mechanisms are used such as client certificates. In such cases, the ISA server computer should be trusted for delegation to: SharePoint Web Application App Pool Identity Account – SPN matching HTTP service type and web application URL.	To any service
SharePoint Products and Technologies Web Application	App Pool Identity account (if it is Network Service, then it is the computer account)	SQL Server Analysis Services service account – SPN matching MSOLAPSVC.3 service type and SQL server FQDN	Kerberos only or To any service (Kerberos only is preferred)
SQL Server Analysis Services	Service account (user)	Does not need to be trusted for delegation.	N/A

It is important to note that Kerberos constrained delegation requires that all servers in a delegation chain as well as the back-end belong to the same Active Directory domain, and that once constrained delegation has been configured for a service all subsequent “hops” in the delegation chain must also be configured for constrained delegation.

Windows SharePoint Services 3.0/Office SharePoint Server 2007

Windows SharePoint Services 3.0 (or Office SharePoint Server 2007) provides the user interface through which PerformancePoint 2007 content is delivered. In essence, PerformancePoint 2007 dashboards are SharePoint Products and Technologies pages comprised of PerformancePoint web parts. These pages are deployed to a page library using PerformancePoint Dashboard Designer 2007. These pages are subject to the security of the site to which they are deployed and the permissions of each PerformancePoint item on a dashboard page. The data rendered within a PerformancePoint page is controlled by the credentials under which queries to Analysis Services are executed, and roles that are defined within the Analysis Services cubes.

Since Windows SharePoint Services or Office SharePoint Server is the host of PerformancePoint content, it must be properly configured to support the delegation of credentials from an end-user to Analysis Services. Currently, SQL Server Analysis Services only supports Windows Authentication for assigning membership to roles within an Analysis Services database. In order for the Active Directory credentials of each user to be passed through to Analysis Services, the authentication provider for SharePoint Products and Technologies must also be set to Windows Authentication instead of custom providers such as .NET authentication. SharePoint Products and Technologies must also be configured to support Kerberos authentication so that Analysis Services can pass the identity of each individual user, and associate that user with the appropriate roles within an Analysis Services database.

Aligning security for SharePoint Products and Technologies to work with PerformancePoint 2007 involves the configuration of the following components:

1. The web application Authentication Provider
2. The Application Pool Identity which hosts the web application
3. The web.config file for the web application
4. SharePoint Products and Technologies security at the Site Collection level

Web Application Authentication Providers

Windows SharePoint Services 3.0 and Microsoft Office SharePoint Server 2007 web applications support handing off authentication activities to other services such as Active Directory and custom membership providers such as .NET and SQL Server. These services are known in SharePoint Products and Technologies as Authentication Providers. Since they also provide the source for accounts which are then granted or denied access to SharePoint sites, these services are also known as Membership Providers. There are three different types of Authentication Providers in SharePoint Products and Technologies: Windows, Forms, and Web Single Sign On (Web SSO). Given the requirements for PerformancePoint 2007 and Windows Authentication, this whitepaper will only focus on the Windows Authentication Provider.

Configuring a web application in SharePoint Products and Technologies for Windows Authentication is quite simple. In fact, when creating a new Web Application in SharePoint Central Administration, "Windows" is the available default Authentication Provider, and enabling the Web Application for Kerberos authentication is as simple as choosing the appropriate radio button. The Authentication Providers of an already created web application can be configured by using the Application Management page of SharePoint Central Administration and then clicking the Authentication Providers hyperlink. There are two options when selecting Integrated Windows Authentication in the Authentication Providers configuration page: Negotiate (Kerberos), and NTLM. As its name implies, setting the authentication to NTLM uses the NTLM protocol strictly.

Setting the authentication to Negotiate (Kerberos) will cause the web application to attempt Kerberos authentication first, and if Kerberos authentication fails, it defaults back to NTLM.

Enabling a web application for Kerberos authentication through SharePoint Central Administration sets the NTAuthenticationProviders value in the IIS metabase to "Negotiate,NTLM" for the IIS virtual server corresponding to the web application. This is equivalent to using the ADSUTIL.VBS VBScript found in %SYSTEMDRIVE%\InetPub\AdminScripts to set the value of w3svc/<virtual_server_number>/root/NTAuthenticationProviders to "Negotiate,NTLM." There is a difference, however, in that using the script to manipulate the IIS metabase instead of using SharePoint Central Administration changes the configuration in the IIS metabase but does not change the configuration inside SharePoint Products and Technologies. This results in a mismatch in the settings between SharePoint Products and Technologies and IIS. Modifying the authentication mechanism of a web application should always be done from within SharePoint Central Administration. This will guarantee that both SharePoint Products and Technologies and IIS are in sync in terms of authentication settings.

Application Pool Identity

In Internet Information Services (IIS) 6 and 7, application pools provide a way to isolate web applications from one another. Each application pool is associated with a worker process (w3wp.exe), and like any service or process in the Windows Server platform, the worker process must run within the context of a user or computer account. Application Pool Identity refers to the account under which a worker process runs.

When application pools are created, their default identity account is Network Service. The Network Service account is a security principal that is local to a server, that is, it is not an Active Directory domain account. When the Network Service account accesses resources outside of the server, the account is then translated to the server's domain account. For example, if a process on a server called WEB1 runs under the Network Service account and the process connects to a file share on a server called WEB2, the request for the file share is seen by WEB2 as coming from WEB1 and not from Network Service. As a security best practice, SharePoint Products and Technologies application pools should run under the context of dedicated domain accounts instead of Network Service.

Aside from being a security best practice, there are other reasons for using domain accounts for the Application Pool Identity that relate to Kerberos configuration. Kerberos requires the registration of Service Principal Names (SPNs). SPNs are registered to the Application Pool Identity accounts and they must be unique. In SharePoint Products and Technologies farms where there is more than one server, application pools running under the context of Network Service would need to have the same SPNs registered for each machine account. This would result in duplicate SPN registrations that would cause Kerberos authentication to fail.

Web.config file

In order to create a unique connection to Analysis Services under the credentials of each user from a SharePoint Products and Technologies web application, a SharePoint Application web.config file must be modified so that this functionality is enabled. Refer to the PerformancePoint 2007 section of this document for details on the configuration change that is required.

SharePoint Site Collection Security

Security for SharePoint sites is defined at the Site Collection level. By default, permissions in SharePoint Products and Technologies defined at the Site Collection level are inherited by all sites and sub sites in the Site Collection. SharePoint Products and Technologies defines groups which are granted different levels of access to the site, and Active Directory users and groups can then be added to these SharePoint groups. Users and groups that will be accessing the PerformancePoint 2007 dashboards require at a minimum the View Only SharePoint level of access to the site.

Alternate Access Mappings

Windows SharePoint Services 3.0 and Microsoft Office SharePoint Server 2007 include a feature that can assist in extranet scenarios where a web application must be accessible using multiple unique names. This feature is called Alternate Access Mappings (AAM), and it is specifically useful in situations where reverse proxy technologies such as ISA Server 2006 are used to publish web applications to the Internet. Basically, AAM enables SharePoint Products and Technologies to map web requests to the appropriate web application, and to direct the user to the proper URL as the user interacts with SharePoint Products and Technologies. In other words, AAM enables SharePoint Products and Technologies to build the links in the pages it generates based on a mapped URL rather than the URL specified in the request it received.

Consider the following scenario. A SharePoint site is locally accessible using an "internal" URL such as `http://intranet.nwtraders.msft`, which is not resolvable from the Internet. A reverse proxy like ISA Server 2006 is used to publish the SharePoint site using a "public" URL such as `http://portal.nwtraders.com`, which is accessible from the Internet. Without AAM, when the reverse proxy makes a request to SharePoint Products and Technologies using the "internal" URL then SharePoint Products and Technologies builds the links in the pages using the "internal" name and gives them to the proxy server which in turn gives the page to the user on the Internet. Because the links were built using the "internal" name eventually the user requests will fail when attempting to access certain pages or features due to the fact that the "internal" URL is not resolvable on the public Internet. Using AAM to create a mapping between the "internal" name and the "public" name enables SharePoint Products and Technologies to build the links using the name that the user originally intended.

AAM works by associating an Internal URL with one of five different Public names, each one corresponding to a Zone. The five available Zones are Default, Intranet, Internet, Custom, and Extranet. Functionally, there are no differences between the zones so they should be viewed more like an administrative feature. The following figures illustrate the AAM collection for a web application and its Public URLs.

Internal URL	Zone	Public URL for Zone
<code>http://intranet.nwtraders.msft</code>	Default	<code>http://intranet.nwtraders.msft</code>
<code>http://portal.nwtraders.com</code>	Internet	<code>http://portal.nwtraders.com</code>
<code>http://extranet.nwtraders.msft:9000</code>	Internet	<code>http://portal.nwtraders.com</code>

Figure 5 - Alternate Access Mapping Collection

Central Administration > Operations > Alternate Access Mappings > Edit Public Zone URLs

Edit Public Zone URLs

Alternate Access Mapping Collection
Select an Alternate Access Mapping Collection.

Alternate Access Mapping Collection: **NorthWinds Traders Portal**

Public URLs
Enter the public URL protocol, host, and port to use for this resource in any or all of the zones listed. The Default Zone URL must be defined. It will be used if needed where the public URL for the zone is blank and for administrative actions such as the URLs in Quota e-mail. <http://go.microsoft.com/fwlink/?LinkId=114854>

Default

Intranet

Internet

Custom

Extranet

Figure 6 - Public URLs

As exemplified above, multiple Internal URLs can be mapped to a zone which associates the Internal URL to the Public URL given to that zone. In this example, the Internet zone has a Public URL of `http://portal.nwtraders.com`, and there are two different Internal URLs mapped to this zone (`http://portal.nwtraders.com` and `http://extranet.nwtraders.msft:9000`). Note that for every Internal URL that is added to an Alternate Access Mapping collection, a corresponding host header value must be added to the site in IIS. Otherwise, IIS will not respond to the request for the URL and will never hand the request off to SharePoint Products and Technologies.

As of this writing, PerformancePoint 2007 web parts do not support Alternate Access Mappings which contain Internal URLs that are different than the mapped Public URL when publishing with SharePoint Products and Technologies through a reverse proxy such as ISA Server 2006. Using the example given above, configuring the reverse proxy to connect to SharePoint Products and Technologies using the URL `http://extranet.nwtraders.msft:9000` which maps to `http://portal.nwtraders.com`, would result in the dashboards losing functionality because the web parts that render the dashboards do not support AAM in this fashion. For the solution to work, the reverse proxy must be configured to connect to an AAM which contains `http://portal.nwtraders.com` for both the Internal and the Public URLs.

SQL Server and SQL Analysis Services

Virtually all Microsoft based Business Intelligence solutions leverage SQL Server's relational (SQL Server RDBMS) and multi-dimensional (SQL Server Analysis Services) database engines. SharePoint Products and Technologies and PerformancePoint both use SQL Server relational databases as a data repository to store configuration and content databases. However, these databases are always accessed under the credentials of service accounts, and not directly by end-users. As a result, no material configuration changes are required for SQL Servers' relational database engine for extranet solutions.

Analysis Services is in many ways the heart of Microsoft's Business Intelligence platform. For PerformancePoint, Analysis Services is the engine that truly enables

advanced analytics through its multi-dimensional databases. Analysis Services has an advanced role based security model that provides granular control over the data a user can access. For extranet scenarios, Analysis Services needs to be passed the credentials of the end-user so that security can be applied for each user. In order for Analysis Services to receive these credentials, configuration changes need to be made. As a result, we will be focusing primarily on Analysis Services in this white paper.

Analysis Services Service

Analysis Services runs as a service under the credentials of a service account. It is a recommended best practice for Analysis Services to run under a dedicated Active Directory service account versus Local System. This is also a prerequisite for configuring Service Principal Names for Analysis Services so that it can interact with the Kerberos delegated authentication. Open the Services console in Windows to see what account Analysis Services is running under.

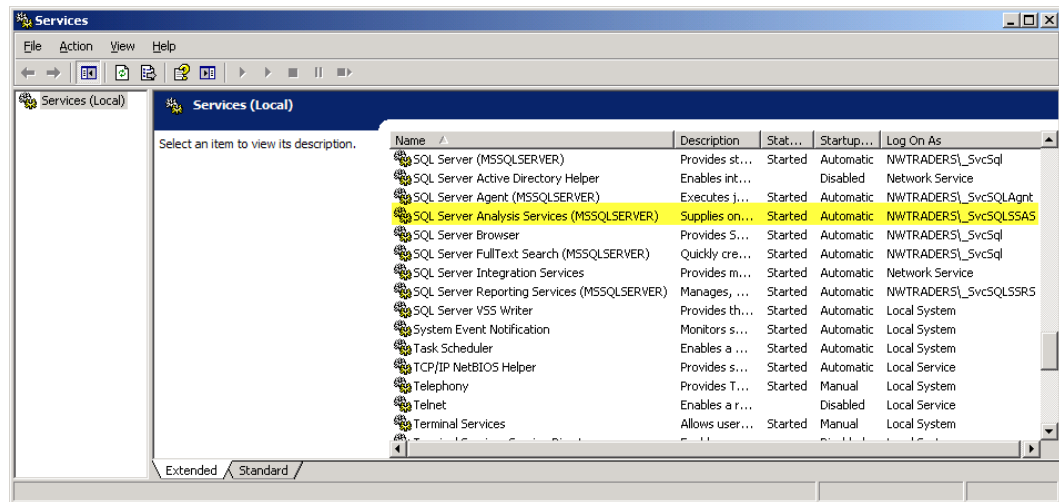
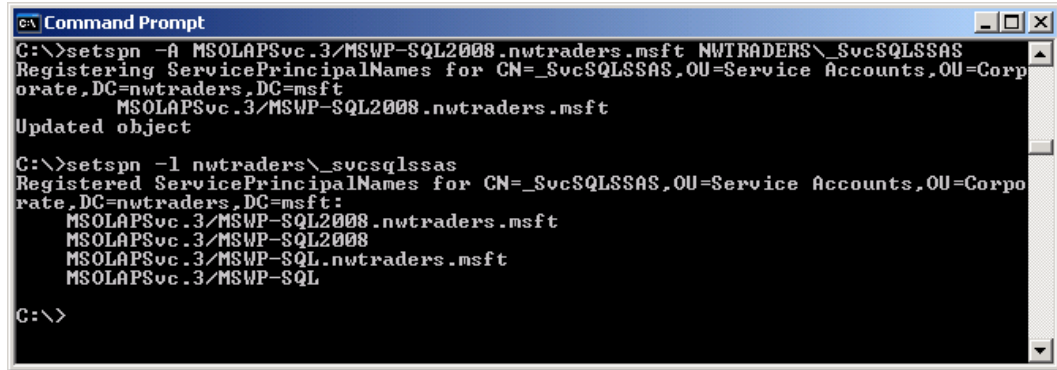


Figure 7 - SSAS Service Account

SSAS Service Principal Name

Kerberos delegated authentication is the mechanism that enables Analysis Services to be connected and authenticated to by service accounts under the credentials of end-users. In order for Analysis Services to participate in Kerberos delegated authentication, a Service Principal Name must be created that tells Active Directory what account the Analysis Services service is authorized to run under. This is done using the setspn.exe utility.

When creating Service Principal Names (SPN's) for Analysis Services, a SPN should be created for both the Fully Qualified Domain Name (FQDN) and NetBIOS name of the host server. This allows authentication to work whether connection to the server is made using the FQDN or NETBIOS names. The following illustrates a command for creating a SPN as well as how to look up the current SPNs for a service account:



```

C:\>setspn -A MSOLAPSvc.3/MSWP-SQL2008.nwtraders.msft NWTRADERS\_SvcSQLSSAS
Registered ServicePrincipalNames for CN=_SvcSQLSSAS,OU=Service Accounts,OU=Corporate,DC=nwtraders,DC=msft:
MSOLAPSvc.3/MSWP-SQL2008.nwtraders.msft
Updated object

C:\>setspn -l nwtraders\_svcsqlssas
Registered ServicePrincipalNames for CN=_SvcSQLSSAS,OU=Service Accounts,OU=Corporate,DC=nwtraders,DC=msft:
MSOLAPSvc.3/MSWP-SQL2008.nwtraders.msft
MSOLAPSvc.3/MSWP-SQL2008
MSOLAPSvc.3/MSWP-SQL.nwtraders.msft
MSOLAPSvc.3/MSWP-SQL
C:\>

```

Figure 8 - SSAS SPN Example

PerformancePoint Server (PPS) 2007

Microsoft Office PerformancePoint Server 2007 Monitoring and Analytics (a.k.a. PerformancePoint or PPS) is used to deliver Business Intelligence content through SharePoint Products and Technologies. It is an advanced data visualization tool, and can also be used to aggregate the display content from Excel Services, ProClarity Analytics Server, Reporting Services, and the web.

While PerformancePoint supports multiple types of data sources, Analysis Services is the native data source that provides the most functionality to the end user. As a result, the focus of this whitepaper will be on enabling extranet connectivity and authentication for native PerformancePoint content (Scorecards, charts, and grids) to Analysis Services.

PerformancePoint has several methods for managing connections to Analysis Services. These are:

- Service Account
- Connection Per User
- Custom Data
- Connection Roles

It is important to understand the implications of each of these options when creating a Business Intelligence solution. The various connectivity options are outlined below.

Service Account (Default)

In the first method, PerformancePoint connects to Analysis Services under the credentials of a service account. This is the default behavior of PerformancePoint, and is appropriate if all users in the organization should have the same permissions to Analysis Services as the service account that runs the SharePoint Application Pool.

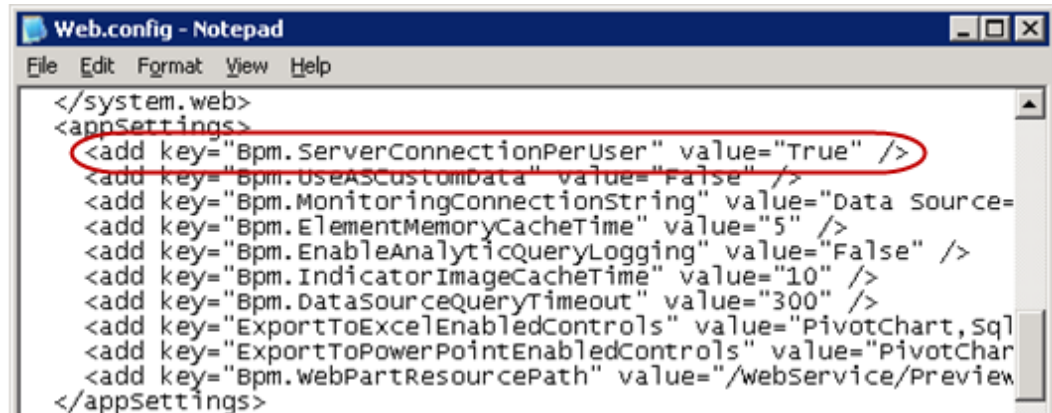
For most implementations, including extranet scenarios, this method rarely meets the requirements for security.

Connection per User

In the second method, PerformancePoint creates connections to Analysis Services under the identity of the end-user. This means that Analysis Services is able to identify

the actual user accessing OLAP data, enabling Analysis Services to enforce its role-based security model for requests submitted by each user.

This behavior is enabled through a configuration named `ServerConnectionPerUser`. This configuration can be set independently for different PerformancePoint clients, which means Dashboard Designer can be configured to behave differently from dashboards published in SharePoint Products and Technologies. The `ServerConnectionPerUser` configuration is maintained in the `Web.config` file for each PerformancePoint client. The following is a sample of one of these `Web.config` files:



```

Web.config - Notepad
File Edit Format View Help
</system.web>
<appSettings>
<add key="Bpm.ServerConnectionPerUser" value="True" />
<add key="Bpm.UseASCUSTOMData" value="False" />
<add key="Bpm.MonitoringConnectionString" value="Data Source=
<add key="Bpm.ElementMemoryCacheTime" value="5" />
<add key="Bpm.EnableAnalyticQueryLogging" value="False" />
<add key="Bpm.IndicatorImageCacheTime" value="10" />
<add key="Bpm.DataSourceQueryTimeout" value="300" />
<add key="ExportToExcelEnabledControls" value="PivotChart,Sql
<add key="ExportToPowerPointEnabledControls" value="PivotChar
<add key="Bpm.WebPartResourcePath" value="/WebService/Preview
</appSettings>
  
```

Figure 9 - PPS `ServerConnectionPerUser` `Web.config` sample

The `ServerConnectionPerUser` setting is a Boolean value (true/false) that specifies how PerformancePoint Server should handle authentication to Analysis Services. The default value is false, which means that all users will share a single connection to Analysis Services using the credentials of a service account. When `ServerConnectionPerUser` is set to true, PerformancePoint will create a unique connection per user, and connect to Analysis Services under the credentials of each user. It should be noted that manipulating the `ServerConnectionPerUser` configuration for PerformancePoint is not unique to extranet solutions, and is often required for intranet solutions as well.

In almost all cases, changing `ServerConnectionPerUser` to true requires additional configuration changes for each of the other technologies that comprise an end-to-end Business Intelligence solution for authentication to work. This includes Active Directory, SharePoint Products and Technologies, Analysis Services, as well as client machines. These changes are discussed in detail throughout this whitepaper.

Choosing the Server Connection per User Setting

In any implementation of PerformancePoint, the `ServerConnectionPerUser` settings need to be determined based on solution requirements. Since it is set uniquely for each PerformancePoint client interface, it takes some planning to ensure the desired functionality is obtained.

There are three PerformancePoint client interfaces, each with a unique `ServerConnectionPerUser` configuration. These are outlined below including the details of how their authentication behavior changes depending on the value of the `ServerConnectionPerUser` configuration.

Dashboard Designer Client Interface

Dashboard Designer is the development environment for creating PerformancePoint content and publishing it to SharePoint Products and Technologies. Dashboard Designer is a developer focused tool that is primarily used by administrators and developers, and will rarely if ever be accessed by an end-user. Dashboard designer connects to Analysis Services through the PPS Web Service, which runs under the PPSMonitoringWebService Application Pool. Therefore, by changing the ServerConnectionPerUser setting for the Dashboard Designer virtual directory within the PPSMonitoring Web Service, one can control the authentication behavior of Dashboard Designer.

When ServerConnectionPerUser is set to false (default), Dashboard Designer will connect to Analysis Services using the identity of the PPSMonitoringWebService Application Pool. Therefore, ensure that the identity of the PPSMonitoringWebService Application Pool is able to access the appropriate Analysis Services databases. This also means that anyone using Dashboard Designer will be able to access Analysis Services databases through the credentials of the PPSMonitoringWebService Application Pool.

When ServerConnectionPerUser is set to true, Dashboard Designer will connect to Analysis Services using the identity of the user. This allows Analysis Services to enforce role-based security for Dashboard Designer users.

The default path for Dashboard Designer's web.config is:
*C:\Program Files\Microsoft Office PerformancePoint
Server\3.0\Monitoring\PPSMonitoring_1\WebService\Web.config*

Preview Site Client Interface

The PerformancePoint Preview web site allows users to preview a dashboard before it is published to a SharePoint site. This is used primarily by developers working within the Dashboard Designer client. The ServerConnectionPerUser setting is stored within the Preview virtual directory within the PPSMonitoring Web Service.

When "Connection per User" is set to false (default), the preview site will connect to Analysis Services using the identity of the PPSMonitoringWebService Application Pool. So once again, ensure that the identity of the PPSMonitoringWebService Application Pool has the appropriate permissions to Analysis Services Databases.

When "Connection per User" is set to true, the preview site will connect to Analysis Services using the identity of the user. This allows Analysis Services to enforce role-based security when accessing the preview site.

The default path to the Preview site's Web.config file is:
*C:\Program Files\Microsoft Office PerformancePoint
Server\3.0\Monitoring\PPSMonitoring_1\Preview\Web.config*

SharePoint Site Client Interface

Unlike the Dashboard Designer and Preview clients that are used almost exclusively by developers, SharePoint Products and Technologies is the client used to deliver content to end users. For most organizations, this content should be secured for each individual user, requiring connections to Analysis Services to be made under the identity of that user. This is again done by manipulating the ServerConnectionPerUser setting in the web.config file for the SharePoint application.

When ServerConnectionPerUser is set to false, SharePoint Products and Technologies will use the credentials of its Application Pool to connect to Analysis Services when users are consuming PerformancePoint content. This again means that the Application Pool account for SharePoint Products and Technologies will need access to the cubes.

This is not a common configuration unless the information being made available is not sensitive, and can be viewed by anyone.

When `ServerConnectionPerUser` is set to true, SharePoint Products and Technologies uses the credentials of the user accessing the site.

The path to this `Web.config` file is dependent upon the configuration of the SharePoint Products and Technologies environment. The following is a sample path to this `Web.config` file:

```
C:\Inetpub\wwwroot\wss\VirtualDirectories\80\Web.config
```

Custom Data

Another method for managing authentication is to pass Analysis Services a text value containing the name of the currently authenticated user. This is done by utilizing the Custom Data property of an Analysis Services connection string. This connection string property enables client applications to pass Analysis Services a text value when creating a new connection. PerformancePoint uses this to pass the name of the authenticated user to Analysis Services where it can be leveraged to apply security within the Analysis Services cubes.

The usage of the Custom Data connection string property is managed through the `UseASCustomData` configuration for PerformancePoint. Just like the Connection Per user configuration, the Use Custom Data configuration is managed within `Web.config` files, and is set to a Boolean (true/false) value.

Typically the user name that is passed to Analysis Services is utilized within the dimension data tab of an Analysis Services Role. The `CUSTOMDATA()` MDX function returns the value passed to Analysis Services through the Custom Data connection string property. The `CUSTOMDATA()` function can be used to construct an MDX set expression that defines a sub-set of dimension members that a particular user should be able to access.

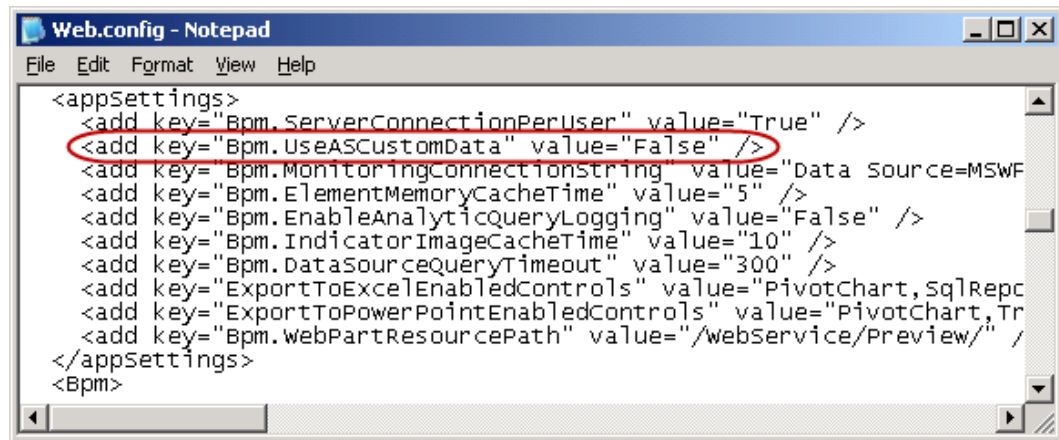
Note that the primary advantage of the Use Custom Data configuration is that in some three-tier intranet architectures, Kerberos and delegated authentication may not have to be configured, which significantly reduces the complexity of the environment. This is because a connection to Analysis Services can be made with the service account, and the name of the current user can still pass to Analysis Services through the Custom Data connection string property. This eliminates the double-hop issue that typically drives the need for Kerberos and delegated authentication.

The benefit of this approach is the obvious reduction in complexity and configurations required to stand up a new environment. The drawback is that MDX set expressions must be utilized within the Dimension Data tab of Analysis Services to secure the cubes, and Role membership cannot be used to enforce security between individuals since connections to the cubes are made under the credentials of a service account.

Using SSAS Custom Data

The configuration of an environment to use the `UseASCustomData` configuration is relatively straightforward in comparison to `ServerConnectionPerUser`. When `UseASCustomData` is set to false, no information is passed to Analysis Services through the Custom Data connection string property. When it is set to true, the current user's name is passed to Analysis Services via the Custom Data connection string property.

Just like the `ServerConnectionPerUser` configuration, the `UseASCustomData` configuration is set for Dashboard Designer, the Preview Site, and the SharePoint site, through the same respective `web.config` files. An example of the `UseASCustomData` configuration within one of these `web.config` files is below:



```
<appSettings>
  <add key="Bpm.ServerConnectionPerUser" value="True" />
  <add key="Bpm.UseASCUSTOMData" value="False" />
  <add key="Bpm.MonitoringConnectionString" value="Data Source=MSWF
  <add key="Bpm.ElementMemoryCacheTime" value="5" />
  <add key="Bpm.EnableAnalyticQueryLogging" value="False" />
  <add key="Bpm.IndicatorImageCacheTime" value="10" />
  <add key="Bpm.DataSourceQueryTimeout" value="300" />
  <add key="ExportToExcelEnabledControls" value="PivotChart,SqlRepc
  <add key="ExportToPowerPointEnabledControls" value="PivotChart,Tr
  <add key="Bpm.WebPartResourcePath" value="/WebService/Preview/" /
</appSettings>
</Bpm>
```

Figure 10 - UseASCUSTOMData web.config

For locations of each of the three web.config files, refer to the **Error! Reference source not found.** section above.

Connection Roles

Another less commonly used option for managing security is the Roles setting within a PerformancePoint Data Source to Analysis Services. This enables a Data Source to connect to Analysis Services with the permissions of the specified roles. In the figure below, the Roles setting is highlighted, and is configured to use the combined permissions of roles RoleX, RoleY, and RoleZ.

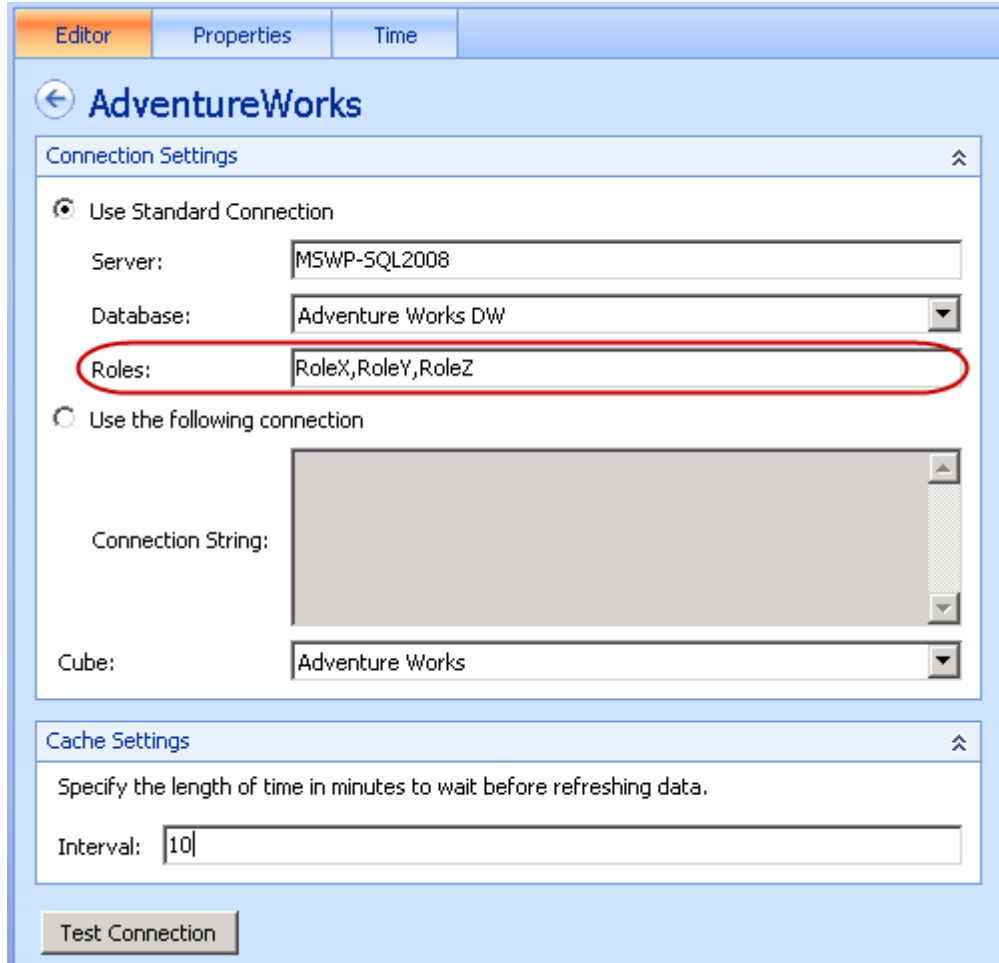


Figure 11 - Roles within a PPS Data Source

The drawback of this approach is that all users will have the same rights to Analysis Services data through the data source(s) used within a dashboard. This decentralizes the administration of security outside of Analysis Services, and may not be the most desired solution. For the purposes of this whitepaper and its lab environment, the Roles setting within a Data Source is not used.

**PPS
Connection
Architecture**

The following architecture diagram illustrates the various connection points in the architecture of PerformancePoint. Note that the connections that are configurable are reflected in the diagram.

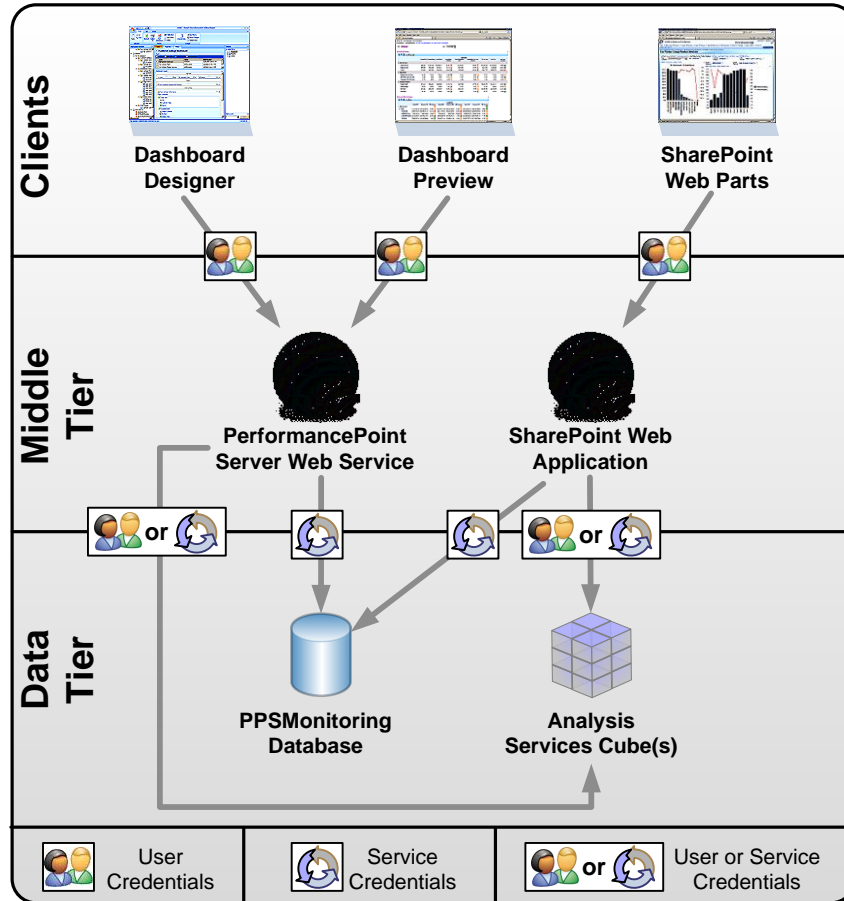


Figure 12 - PPS Connectivity Architecture

Note that for the purposes of this whitepaper a lab environment was created, and it has been configured to have ServerConnectionPerUser set to true, and Custom Data set to false. This is the most common configuration for extranet scenarios, and is supported by Microsoft.

Impact of Connection Settings

There can be a significant impact to performance when changing PerformancePoint connection configurations. PerformancePoint Server has functionality for caching the results of queries for a period of time, allowing subsequent requests to be answered directly without another query being issued to Analysis Services. This setting is maintained for each data connection in Dashboard Designer, highlighted in the following figure:

The screenshot shows the 'Connection Settings' dialog box with the 'Cache Settings' section highlighted in yellow. The 'Cache Settings' section contains the text 'Specify the length of time in minutes to wait before refreshing data.' and an 'Interval' field set to '10'. The 'Connection Settings' section has two radio buttons: 'Use Standard Connection' (selected) and 'Use the following connection'. Under 'Use Standard Connection', there are fields for 'Server' (MSWP-SQL2008), 'Database' (Adventure Works DW), and 'Roles'. Under 'Use the following connection', there is a 'Connection String' field and a 'Cube' dropdown menu (Adventure Works). A 'Test Connection' button is at the bottom.

Figure 13 - PPS Data Source Cache Setting

When the `ServerConnectionPerUser` or `UseASCUSTOMData` configuration for PerformancePoint are both set to false, PerformancePoint will create a single connection to Analysis Services under the credentials of a Service Account, and all users will see the same results when viewing a PerformancePoint dashboard. Because of this, the cache can be shared between all users, which can lead to significant performance improvements for frequently viewed dashboards.

However, when either the `ServerConnectionPerUser` or `UseASCUSTOMData` configuration for PerformancePoint is set to true, a unique connection is created to Analysis Services for each user accessing PerformancePoint. This means that a cache is created for each individual user, and can't be shared between users. As a result, the caching benefits are significantly reduced, and the solution will generally not be able to support the same number of users with equal performance.

For more information regarding PerformancePoint Connectivity, refer to this whitepaper:

http://ppsblog.members.winisp.net/Img/MonitoringServerConnectivityDocument_A409/PPSMSConnnectivity02092008.pdf

ISA Server

Microsoft Internet Security and Acceleration Server (ISA) 2006 is an application-layer filtering capable firewall that among other things provides network edge protection, secure application publishing, web caching, VPN, and proxy services. As a proxy, ISA

Server can function in both forward and reverse capacities. When ISA Server publishes a server or an application to the public Internet, it functions as a reverse proxy, forwarding client requests to the published resources on behalf of the client while keeping the process completely seamless. Client requests go to the ISA Server computer where they are filtered and analyzed, and then the ISA Server computer forms a request to the published resource which in turn responds back to the ISA Server computer which then responds to the client. From the perspective of the client, the published resource seems to be directly accessible from the Internet.

One of the key benefits of implementing ISA Server 2006 is that it provides pre-authentication of clients prior to granting access to the published resources. It is considered pre-authentication because ISA Server 2006 does not merely pass the authentication request to the published resource, but instead it authenticates the client against a number of security providers such as Active Directory, RADIUS, and RSA Security among others prior to sending the request to the published resource. If the client is successfully authenticated, then ISA server can seamlessly pass (or delegate) the client credentials to the published resource in whatever form the published resource requires. This enables organizations to mix certain forms of authentication and delegation to meet specific security requirements. For example, ISA Server may enforce RADIUS pre-authentication for clients and pass the credentials in Basic form to the published resource. For a list of valid pre-authentication and delegation combinations please see the following link:
<http://technet.microsoft.com/en-us/library/bb794722.aspx>

As a Windows Server family product, ISA Server 2006 can be joined to an Active Directory domain. One key benefit of ISA Server 2006 being a member of a domain is that it can delegate the client credentials using Kerberos Delegation (both constrained and unconstrained). That is, ISA Server can obtain a service ticket on behalf of the client whether the client computer is a member of the domain or not. Because the ServerConnectionPerUser configuration for PerformancePoint Server 2007 requires Kerberos authentication, ISA Server 2006 becomes a critical component of any extranet solution.

Publishing Rules

ISA Server 2006 publishing rules provide the means through which internal services and applications are made available to clients in the public Internet. Publishing rules are comprised of a set of conditions that when met, certain actions are then taken. Publishing rules are associated with a Listener which provides the authentication mechanism. A single Listener can be shared across multiple publishing rules. The following bulleted list details what a Listener controls versus what the Publishing rules control:

- Listener
 - ▶ Network and IP addresses for incoming requests
 - ▶ Connection ports
 - ▶ Concurrent client connections
 - ▶ SSL Certificate Assignments
 - ▶ Authentication and Validation methods
 - ▶ Authentication forms customization and password management
 - ▶ Single Sign On (SSO)

- Publishing Rule
 - ▶ Action (Allow/Deny)
 - ▶ Traffic Sources
 - ▶ Destination (published resource)
 - ▶ Allowed Protocols
 - ▶ Assigned Listener
 - ▶ URLs

- ▶ Published Paths
- ▶ Authentication Delegation
- ▶ Protocol Bridging
- ▶ Users and Groups
- ▶ Rule Schedule
- ▶ Link Translation

ISA Server 2006 includes easy to use wizards for publishing SharePoint sites. Properly configuring a publishing rule for SharePoint sites that have PerformancePoint dashboards requires that the delegation of credentials in the publishing rule be set to Negotiate (Kerberos/NTLM) or Kerberos Constrained Delegation. Kerberos Constrained Delegation can only be used when the ISA Server computer account in Active Directory is trusted for constrained delegation to any authentication protocol. When either form of credential delegation is used, the SPN used by ISA Server 2006 to obtain a Kerberos ticket on behalf of the user must also be specified. This SPN must match the SPN that was registered to the Application Pool Identity account for the SharePoint Products and Technologies Web Application hosting the PerformancePoint dashboards. Figure 14 - Delegation of credentials in a publishing rule illustrates the Authentication Delegation page of an ISA Server publishing rule.

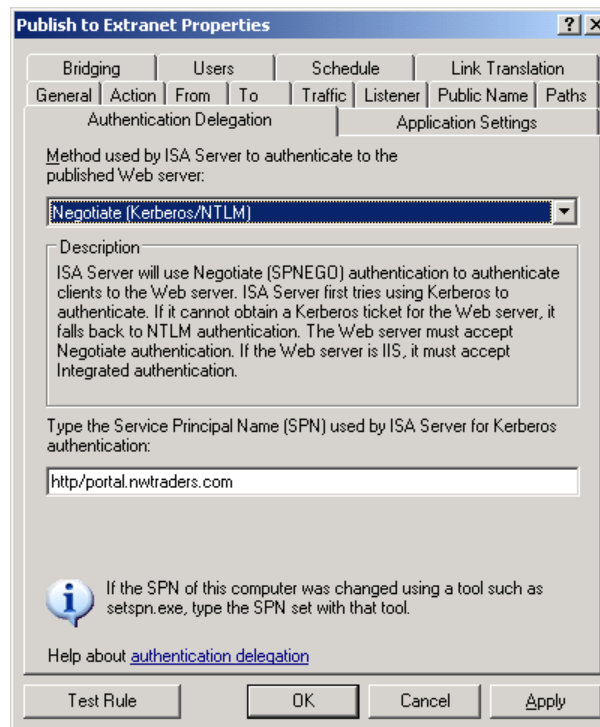


Figure 14 - Delegation of credentials in a publishing rule

Kerberos Constrained Delegation is only supported when the Listener is configured with the following combinations of authentication and validation methods:

Table 6 - Authentication and Validation Combinations for KCD

Authentication	Validation
Forms-based authentication	Active Directory (Windows) Active Directory (LDAP) RADIUS

Authentication	Validation
Basic	Active Directory (Windows) Active Directory (LDAP) RADIUS
Digest	Active Directory (Windows)
Integrated	Active Directory (Windows)
Forms-based authentication with passcode	SecurID RADIUS one-time password
Client certificate	Active Directory (Windows)

Environment Configuration Steps

This section covers the lab which served as the proof of concept for this White Paper. This section also covers how the lab was configured should readers wish to try the concepts presented herein themselves. Also provided are the step-by-step instructions on how each component of the solution was configured (SQL Server, SSAS, Office SharePoint Server, PerformancePoint Server, ISA, Active Directory, Kerberos, Client).

Lab Overview

This section covers the details of the lab including the platform, OS, Networking configuration, VM specs, Directory Services configuration, etc.

To demonstrate the extranet environment depicted throughout this document, a lab environment was built as a proof of concept. This lab was built using Microsoft virtualization technology based on Windows Server 2008 and Hyper-V. The virtual environment simulates the infrastructure depicted in the following diagram.

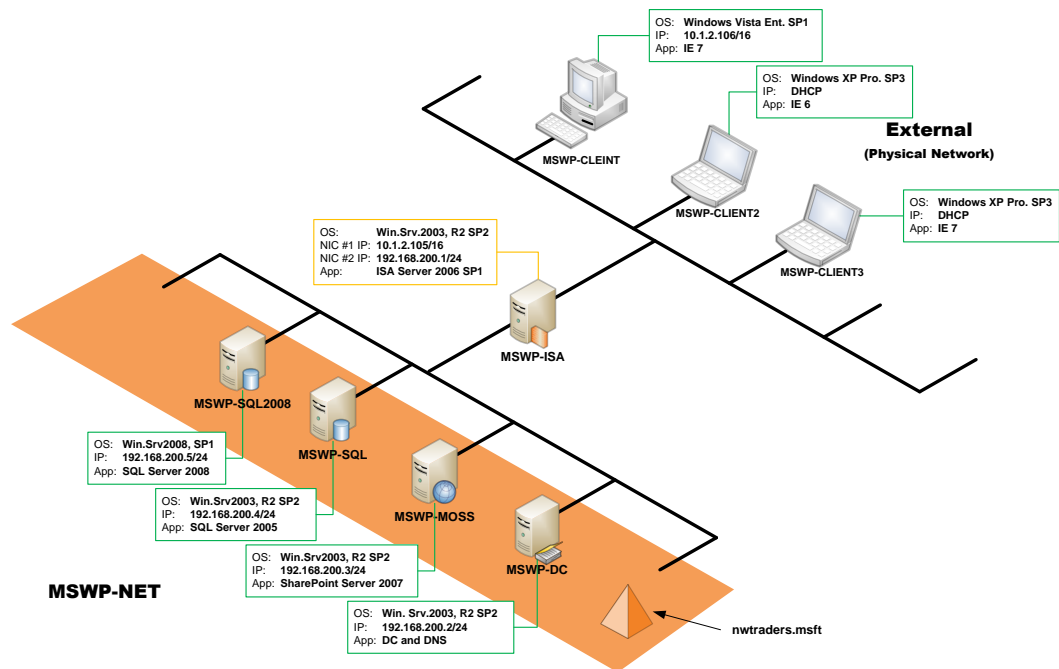


Figure 15 - Lab Environment

The following tables detail the virtual environment.

Table 7 - Lab Configuration

Virtual Networks		
Name	Purpose	Connection Type
MSWP-NET	Simulate LAN. Servers such as Active Directory Domain Controllers, SQL Server, and Office SharePoint Server connect to this network.	Private virtual machine network.
External	Simulate WAN. Clients external to the Domain/LAN connect to this network. This network is bound to a physical network adapter on the Hyper-V host.	External

Virtual Machines		
Name	Purpose	Configuration
MSWP-DC	Active Directory Domain Controller and DNS.	RAM = 512MB CPU = 1x DISK1 = Differencing Disk (10GB Max) CD-ROM = 1x NIC Connection = MSWP-NET NIC IP = 192.168.200.2/24 OS = Windows Server 2003 R2 SP2 (x86)
MSWP-SQL	SQL Server 2005. Back-End for Microsoft Office SharePoint Server 2007.	RAM = 1024MB CPU = 1x DISK1 = Differencing Disk (10GB Max) CD-ROM = 1x NIC Connection = MSWP-NET NIC IP = 192.168.200.4/24 OS = Windows Server 2003 R2 SP2 (x86)
MSWP-SQL2008	SQL Server 2008. Analysis Services databases for PerformancePoint 2007.	RAM = 2048MB CPU = 1x DISK1 = Dynamic VHD (25GB Max) DISK2 = Dynamic VHD (30GB Max) CD-ROM = 1x NIC Connection = MSWP-NET NIC IP = 192.168.200.5/24 OS = Windows Server 2008 SP1 (x64)
MSWP-MOSS	Microsoft Office SharePoint Server 2007.	RAM = 1024MB CPU = 1x DISK1 = Differencing Disk (10GB Max) CD-ROM = 1x NIC Connection = MSWP-NET NIC IP = 192.168.200.3/24 OS = Windows Server 2003 R2 SP2 (x86)
MSWP-ISA	Microsoft Internet Security	RAM = 512MB

Virtual Machines		
Name	Purpose	Configuration
	and Acceleration Server 2006. Firewall and Reverse Proxy.	CPU = 1x DISK1 = Differencing Disk (10GB Max) CD-ROM = 1x NIC #1 Connection = External NIC #1 IP = 10.1.2.105/16 NIC #2 Connection = MSWP-NET NIC #2 IP = 192.168.200.1/24 OS = Windows Server 2003 R2 SP2 (X86)
MSWP-CLIENT	Windows Vista + Internet Explorer 7 external client.	RAM = 512MB CPU = 1x DISK1 = Dynamic VHD (15GB Max) CD-ROM = 1x NIC Connection = External NIC IP = 10.1.2.106/16 OS = Windows Vista Enterprise SP1 (x86)
MSWP-CLIENT2	Windows XP + Internet Explorer 6 external client.	RAM = 512MB CPU = 1x DISK1 = Dynamic VHD (15GB Max) CD-ROM = 1x NIC Connection = External NIC IP = (DHCP) OS = Windows XP Professional SP3 (x86)
MSWP-CLIENT3	Windows XP + Internet Explorer 7 external client.	RAM = 512MB CPU = 1x DISK1 = Dynamic VHD (15GB Max) CD-ROM = 1x NIC Connection = External NIC IP = (DHCP) OS = Windows XP Professional SP3 (x86)
Active Directory		
Forest Name:	nwtraders.local	
Forest Functional Level:	Windows Server 2003	
Domain Controllers:	MSWP-DC (SM, DNM, IM, RIDM, PDCE, GC)	

Active Directory, Kerberos, and Constrained Delegation

The following steps walk through the configuration of Active Directory accounts to enable Kerberos authentication and constrained delegation. In the lab, the following service accounts require configuration for Kerberos and KCD:

Table 8 - Lab Service Accounts

Account Name	Purpose	Addition of SPN Required	KCD Required
_SvcMossAppPoolID	General purpose application pool account. This is the Identity account for the application pool hosting the web app for the site where PerformancePoint dashboards are deployed	YES	YES
_SvcSQLSSAS	Service account for SQL Server Analysis Services service	YES	YES

Table 9 - Lab required SPNs

SPN	Registered To	Reason
HTTP/intranet.nwtraders.msft	_SvcMossAppPoolID	The web app intranet.nwtraders.msft runs in an app pool with the identity _SvcMossAppPoolID
HTTP/portal.nwtraders.com	_SvcMossAppPoolID	The web app has a host header and AAM for the Public URL http://portal.nwtraders.com
MSOLAPSvc.3/MSWP-SQL2008.nwtraders.msft	_SvcSQLSSAS	The SQL Server Analysis Services service runs on MSWP-SQL2008 under the context of _SvcSQLSSAS

1. Register SPN for App Pool Account
 - a. On a server running the Windows Server 2003 Support Tools or running Windows Server 2008, open a command prompt.
 - b. At the command prompt, issue the following commands to add a SPN for the web app Internal URL to the app pool Identity account:

```
SETSPN -A HTTP/[web_app_internal_url] [DOMAINNAME]\[username]
```

and

```
SETSPN -A HTTP/[web_app_url_host] [DOMAINNAME]\[username]
```

In our lab, these commands look as follows:

```
SETSPN -A HTTP/intranet.nwtraders.msft NWTRADERS\_SvcMossAppPoolID
SETSPN -A HTTP/INTRANET NWTRADERS\_SvcMossAppPoolID
```

- c. At the command prompt, issue the following commands to add a SPN for the web app Public URL to the app pool identity account:

```
SETSPN -A HTTP/[web_app_public_url] [DOMAINNAME]\[username]
```

and

```
SETSPN -A HTTP/[web_app_url_host] [DOMAINNAME]\[username]
```

In our lab, these commands look as follows:

```
SETSPN -A HTTP/portal.nwtraders.com NWTRADERS\_SvcMossAppPoolID
SETSPN -A HTTP/PORTAL NWTRADERS\_SvcMossAppPoolID
```

2. Register SPN for SSAS Service Account

- a. On a server running the Windows Server 2003 Support Tools or running Windows Server 2008, open a command prompt.
- b. At the command prompt, issue the following commands to add a SPN for the SQL Server Analysis Services host to the SSAS service account:

```
SETSPN -A MSOLAPSvc.3/[sqlserver_fqdn] [DOMAINNAME]\[username]
```

In our lab, this command looks as follows (all on a single line):

```
SETSPN -A MSOLAPSvc.3/mswp-sql2008.nwtraders.msft
NWTRADERS\_SvcSQLSSAS
```

3. Configure App Pool Account for KCD

- a. On an Active Directory domain controller, open the **Active Directory Users and Computers** management console.
- b. Browse for the application pool identity account. Right-click on the User object and choose **Properties**. In our lab, this account is **_SvcMossAppPoolID**.
- c. On the User properties window, click the **Delegation** tab.

Note: if no SPNs are registered to a user account, the delegation tab in the properties window is not available. Ensure that at least one SPN has been registered to the user object prior to configuring delegation.

- d. Click the **"Trust this user for delegation to specified services only"** radio button, and then click the **"Use Kerberos only"** radio button.

Note: For best security the application pool identity account should always be trusted for constrained delegation with the "Use Kerberos only" option. If the "Use any authentication protocol" option is enabled

instead, Protocol Transition is enabled for the application pool account, and the web application can accept less secure forms of authentication while still obtaining a Kerberos ticket on behalf of the user.

- e. Click the **Add** button to open the Add Services dialog box.
- f. In the Add Services dialog box, click the **Users or Computers button** to search for specific user or computer objects.
- g. In the Select Users or Computers dialog box, enter the name of the **service account for SSAS**. In our lab this is **_SvcSQLSSAS**. Click **OK**.
- h. Back in the Add Services dialog box the SPNs registered to the account will be displayed. **Select the SPN for the SSAS service**. In our lab this SPN is:

MSOLAPSvc.3/mswp-sql2008.nwtraders.msft
- i. Click the **OK** button to close the Add Services dialog box.
- j. Click **Apply**, and then click **OK** to close the User account properties window.

Microsoft Office SharePoint Server 2007

This portion of the document focuses on the steps necessary to configure a Microsoft Office SharePoint Server 2007 farm to support a deployment of PerformancePoint Server 2007. The steps carried out here are also applicable to Windows SharePoint Services 3.0 without modifications. Please note that the steps outlined below assume that an Office SharePoint Server farm has already been deployed and properly configured, therefore the installation and configuration of Microsoft Office SharePoint Server 2007 or Windows SharePoint Services 3.0 is not covered.

Web Application Configuration

The following procedures walk through the process of enabling Kerberos Authentication for a web application. The walkthrough covers the configuration for both new and existing web applications.

New Web Applications

1. Create a web application
 - a. Open the **SharePoint Central Administration** website and navigate to **Application Management → Create or extend Web application**.
 - b. In the Create or Extend Web Application page, click on the **Create a new web application** link to get to the Create New Web Application page which is divided into sections.

IIS web site section

- i. The **Create a new IIS web site** radio button will be selected by default. Enter a descriptive name for the web application. This description is displayed in Internet Information Services Manager as the web site, and it becomes the name of the web application inside Office SharePoint Server. For our lab, the description is **Northwind Traders Portal**.
- ii. Enter a **port** number for the web application. This is the port that IIS uses to listen for HTTP requests. For our lab, the port is set to **80**.
- iii. Enter the **Host Header** which IIS will use to listen for HTTP requests. If this field is left blank, IIS will respond to requests to the host name and the port number set in the previous step. For our lab, the host header is set to **http://intranet.nwtrades.msft**.
- iv. Specify the path for the IIS virtual directory. Make note of this directory as it will contain the web.config file for the web application. For our lab, we accepted the default path **C:\Inetpub\wwwroot\wss\VirtualDirectories\intranet.nwtraders.msft80**.

Security configuration section

- i. Choose the option button for **Negotiate (Kerberos)** as the authentication provider.
- ii. Accept the default of **No** to allow **Anonymous** access.
- iii. If your solution requires **SSL** for the Internal Web site, chose Yes for the use of SSL. In our lab, SSL will not be used.

IMPORTANT: Secure Sockets Layer (SSL) is **required** to securely deploy PerformancePoint Server. The product capabilities expose the server and data sources to potential tampering or disclosure without SSL. This document explains how to configure ISA server and Kerberos constrained delegation with SSL enabled for the external Web address. The internal Web address should also be secured using SSL when dealing with sensitive data. Please refer to Office SharePoint Server, IIS, and ISA server documentation for instructions on configuring SSL for the internal server address.

Load balanced URL section

- i. Enter the Internal URL in the **URL** field. Since this web application is being created, the **Zone** field is set to **Default** and cannot be changed. The load balanced URL for our lab is **http://intranet.nwtraders.msft:80**.

Application pool section

- i. Choose the radio button to **Create** new application pool, and provide a descriptive **Application pool name**. For the security account, select the **Configurable** radio button and

then enter the username and password for the account which will serve as the Identity for the application pool. In our lab, the application pool name is **NorthwindsTradersPortal**, and the Identity account is **_SvcMossAppPoolID**.

Database name and authentication section

- i. Enter the name of the **SQL server** where the database for the web app will be created. Normally, this field is populated with the name of the default SQL server for the farm. In our lab, the SQL server name is **MSWP-SQL**.
 - ii. Enter the **Database Name**. In our lab, the database name is **MOSS_Portal_Content**.
 - iii. If the SQL server in your environment is configured for **Windows authentication** (recommended), then you can accept the default selected option. Otherwise, enter a SQL login and password with sufficient rights to create the content database. In our lab, SQL server is configured for **Windows authentication** only.
2. Create a site collection
 - a. Open the **SharePoint Central Administration** website and navigate to **Application Management → Create Site Collection**
 - b. Choose the web application created in step 1 above. In our lab this web application is **Northwinds Traders Portal**.
 - c. Enter a title and description for the site collection.
 - d. If the site collection is going to be hosted on a path other than the root (/), then enter the path. Otherwise, accept the defaults. In our lab, the site collection is hosted at the root of the URL.
 - e. Select a template for the site
 - f. Specify a primary and secondary Site Collection Administrator
 - g. Specify Site Quota Template
 - h. Click OK to create the site collection.

Existing Web Applications

1. Modify Authentication Providers
 - a. In the **SharePoint Central Administration** website navigate to **Application Management → Authentication Providers**.
 - b. In the Authentication Providers page select the **web application** for which the Membership Provider will be modified.
 - c. Click the **zone** for which the Membership Provider will be modified. In our lab this is the **Default** zone.
 - d. Set the **Authentication Type** to **Windows**, and in the IIS Authentication Settings section check the box for **Integrated**

Windows Authentication. Select the radio button for **Negotiate (Kerberos)**. A warning indicating that manual configuration steps by a domain administrator are required will appear. Click **OK** to close the warning. **Click** Save to store the settings.

Site Collection Configuration

The following steps walk through the configuration of SharePoint site collections that will host PerformancePoint 2007 Dashboard Pages. The first step will be to give access to the appropriate users or groups to the SharePoint site collection. The second step is to add a Page Library to the site where PerformancePoint 2007 dashboards will be deployed.

1. Configure Site Collection Security
 - a. Using Internet Explorer, browse to the SharePoint site that was created during the previous steps. On the landing page, browse to **Site Actions** → **Site Settings**.
 - b. In the **Site Settings** page click the Advanced Permissions link.
 - c. In the permissions page click **New** → **Add User**.
 - d. Enter the Active Directory **Users** and/or **Groups** which will be granted access to the site collection.
 - e. In the **Give Permission** section of the page select a **SharePoint group** to which the users will be added. Alternatively, users can be given permission directly. Dashboards require at least **View Only** rights.
 - f. Click **OK** to accept the settings.
2. Add a Page Library for the Dashboards
 - a. In the SharePoint site that was created during the previous steps, navigate to **Site Actions** → **Create** → **Document Library**.
 - b. Enter a name and description for the document library.
 - c. Specify whether to place a link for the document library in the quick launch left navigation pane.
 - d. Leave version history settings off, as dashboard pages do not require version tracking.
 - e. For **Document Template** select the **Basic page** template.
 - f. Click **Create** for the document library to be created.

Alternate Access Mappings

The following steps walk through the configuration of Office SharePoint Server Alternate Access Mappings for the web application hosting the sites with PerformancePoint 2007 dashboards. Alternate Access Mappings are necessary when the URL to access the web application externally is different than the URL to access the web application internally.

1. Add a public URL for the Internet zone

- a. In SharePoint Central Administration navigate to **Operations** → **Alternate Access Mappings**.
 - b. By default, all AAM collections are shown. Change the AAM Collection by clicking the **Show All** drop-down menu and choosing **Change Alternate Access Mapping Collection**. Select the collection matching the web application that was created for the dashboards. In our lab this is **Northwinds Traders Portal**.
 - c. Click the **Edit Public URLs** link. Add the URL that users will type when browsing for the site on the public Internet to the **Internet** field. In our lab the URL is **https://portal.nwtraders.com**. This will add an Alternate Access Mapping containing "https://portal.nwtraders.com" as both the Internal and Public URLs for the zone.
 - d. Click **Save** to accept the settings.
2. Add host header to web application in IIS

Windows Server 2003

- a. Open **Internet Information Services Manager**. Navigate to **[SERVERNAME] → Web Sites**.
- b. Right-click the web application for the SharePoint site that will host the dashboards and choose **Properties**.
- c. In the **Web site** tab of the properties window click the **Advanced** button.
- d. In the Advanced Web Site Identification dialog box click the **Add** button. This will open the Add/Edit Web Site Identification dialog box.
- e. In the Add/Edit Web Site Identification dialog box enter the **TCP port** that will be used to listen for HTTP requests. In our lab this is set to port **80**. If a port other than 80 is specified, be sure to include the port when setting the connection from a reverse proxy.
- f. In the **Host Header value** field, enter the public name as the host header value. In our lab this is **portal.nwtraders.com**. Click **OK** to close the dialog boxes and the web site properties window.

Windows Server 2008

- a. Open **Internet Information Services Manager**. Expand **Server** → **Sites**.
 - b. Right-click the Web application for the SharePoint site that will host the dashboards and choose **Edit Bindings**.
 - c. Click the **Add** button. Enter the public name in the **Host Name** field. In our lab this is **portal.nwtraders.com**. If a port other than 80 is specified, be sure to include the port when setting the connection from a reverse proxy.
3. Add SSL Certificate to the web application in IIS

Windows Server 2003

- a. Obtain a server certificate from a publicly trusted certification authority that matches the published name of your site. For our lab, the certificate is for **portal.nwtraders.com**. For details on how to generate a certificate request from IIS see [this article on TechNet](#).
- b. Once the certificate has been installed, assign the certificate to the web application by following these steps:
 - i. **Right-click** the web application in IIS and choose **Properties**.
 - ii. Go to the **Directory Security** tab, and then click the **Server Certificate** button. This will start the Web Server Certificate wizard. Click **Next** to begin.
 - iii. Choose the **Assign an existing certificate** radio button and click **Next** to continue.
 - iv. In the list of available certificates choose the appropriate certificate which was installed on the server. In our lab, the certificate name is **portal.nwtraders.com**. Click **Next** to continue.
 - v. Accept the default port number of **443** for the SSL port, and click **Next** to continue.
 - vi. Review the changes that are about to be made and click **Next**. Click **Finish** to complete the wizard.

Windows Server 2008

- a. Obtain a server certificate from a publicly trusted certification authority that matches the published name of your site. For our lab, the certificate is for **portal.nwtraders.com**. For details on how to generate a certificate request from IIS see [this article on TechNet](#).
- c. Once the certificate has been installed, assign the certificate to the web application by following these steps:
 - i. **Right-click** the web application in IIS and choose **Edit Bindings**.
 - ii. Click the **Add** button to add a new binding.
 - iii. Click the drop-down box for the binding **Type** and choose **HTTPS**.
 - iv. Click the drop-down box for the SSL certificate, and choose the certificate which was installed on the server. In our lab, the certificate name is **portal.nwtraders.com**.
 - v. Click the **OK** button to close the add site binding dialog box.
 - vi. Click the **Close** button to close the site bindings window.

Important Notes about the Above Procedures

In a nutshell, following the steps thus far has effectively created a site which is accessible using two distinct URLs: `http://intranet.nwtraders.msft` and `https://portal.nwtraders.com`. This has been done to illustrate the use of Alternate Access Mappings, and how valuable this feature is in situations where the same site needs to be accessed using different URLs. In many real world deployments extranets are published using a single dedicated site with a single URL. In those cases configuring multiple Alternate Access Mappings is not necessary.

Since our site is being hosted on a single web application, turning on the option in IIS to require SSL would render the URL `http://intranet.nwtraders.msft` inaccessible. This does not mean that SSL encryption cannot be used, but instead that IIS will allow the site to be accessed both with and without encryption using both URLs. In cases where ISA Server is used to publish a website, not enabling this option in IIS is typically not a security concern because ISA Server while using unencrypted traffic between itself and the published site it can encrypt the traffic between itself and the clients on the Internet. It would have been possible then to not use SSL for the URL which is intended for the extranet (`portal.nwtraders.com`), however this would have caused the dashboards to work incorrectly (see the Alternate Access Mappings section of this document).

Another approach to this solution would be to extend the web application to an additional IIS site using the public URL for both the host header and the Load Balanced URL. That is, the same web application would be hosted in two distinct IIS virtual directories/websites each one with its own URL. Following this approach requires that the PerformancePoint web parts be manually deployed to the new IIS virtual directory, and that the same procedure for modifying the `web.config` file that was described earlier in this document be followed.

SQL Server 2008 and Analysis Services

Create SSAS SPN

The following steps guide the process of creating a Service Principal Name (SPN) for Analysis Services. This assumes that the `setspn.exe` utility is installed. For details Service Principal Names, or on installing the `setspn.exe` utility, refer to the Active Directory Technology Overview section. For details of Analysis Services' usage of Service Principal Names, please see the SQL Server and SQL Analysis Services Technology Overview section.

Before continuing, you should understand the following abbreviations:

- **[NETBIOS NAME]**: The NetBIOS name of the host server for Analysis Services.
 - o In our lab environment, this is MSWP-SQL2008.
- **[FQDN]**: The fully qualified domain name of the host server for Analysis Services.
 - o In our lab environment, this is MSWP-SQL2008.NWTRADERS.MSFT
- **[DOMAIN]**: The domain for your environment.
 - o In our lab environment, this is NWTRADERS
- **[SERVICE ACCOUNT]**: The service account that Analysis Services is running under.
 - o In our lab environment, this is `_SvcSQLSSAS`.

To configure the Service Principal Names for Analysis Services:

1. Open a command prompt
 - a. Click Start → Run

- b. Type cmd and press Enter
2. Create a SPN for Analysis Services for the FQDN name
 - a. Type the following command at the command prompt

```
setspn -A MSOLAPSvc.3/[NETBIOS NAME] [DOMAIN]\[SERVICE ACCOUNT]
```

3. Create a SPN for Analysis Services for the NETBIOS name
 - a. Type the following command at the command prompt

```
setspn -A MSOLAPSvc.3/[FQDN] [DOMAIN]\[SERVICE ACCOUNT]
```

The following commands were used for configuring our lab environment: `setspn -A MSOLAPSvc.3/MSWP-SQL2008 NWTRADERS_SvcSQLSSAS`
`setspn -A MSOLAPSvc.3/MSWP-SQL2008.nwtraders.msft NWTRADERS_SvcSQLSSA`

PerformancePoint Server 2007

Office SharePoint Server Connection per User

Refer to the Technology Overview for PerformancePoint Server to determine the appropriate Connection Per User setting for SharePoint Products and Technologies.

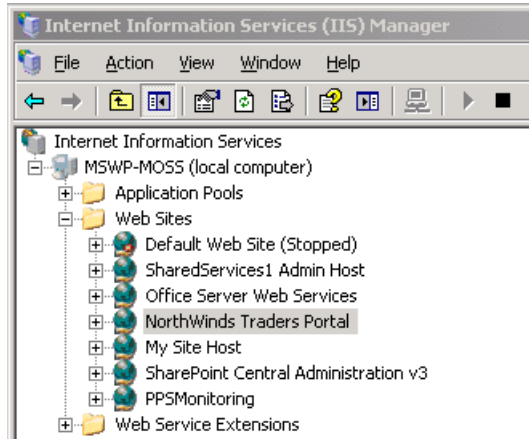
1. Determine the Web Application you are using for Office SharePoint Server.
 - a. Go to **Start** → **All Programs** → **Microsoft Office Server** → **SharePoint Central Administration**.
 - b. Click on the **Application Management** tab.
 - c. Select **Web Application List** under the SharePoint Web Application Management Section.
 - d. Find the **URL** for the Web Application you wish to modify, and note its related **Web Application Name**. For our lab environment, this is **Northwinds Traders Portal**.

Central Administration > Application Management > Web Application List

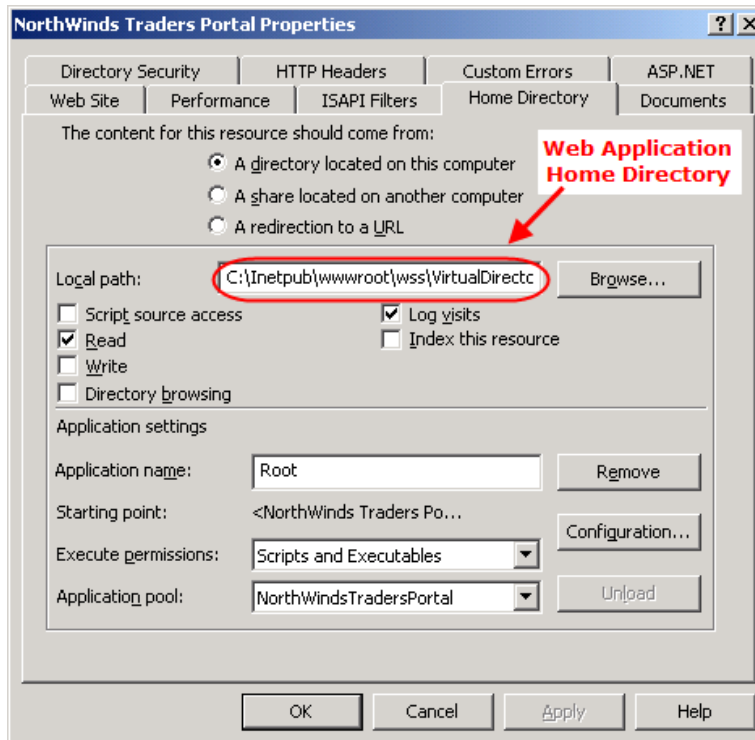
Name	URL
My Site Host	http://my.nwtraders.msft/
Northwinds Traders Portal	http://intranet.nwtraders.msft/
SharedServices1 Admin Host	http://ssp1.nwtraders.msft:7001/
SharePoint Central Administration v3	http://mswp-moss:7000/

2. Find the location of the **web.config** for the Office SharePoint Server Web Application.

- a. Open up **Internet Information Services (IIS) Manager** by navigating to **Start → Administrative Tools → Internet Information Services (IIS) Manager**.
- b. Within IIS Manager, expand **[Computer Name] (local computer) → Websites**.
- c. Right click on the Web Application and select **Properties**.

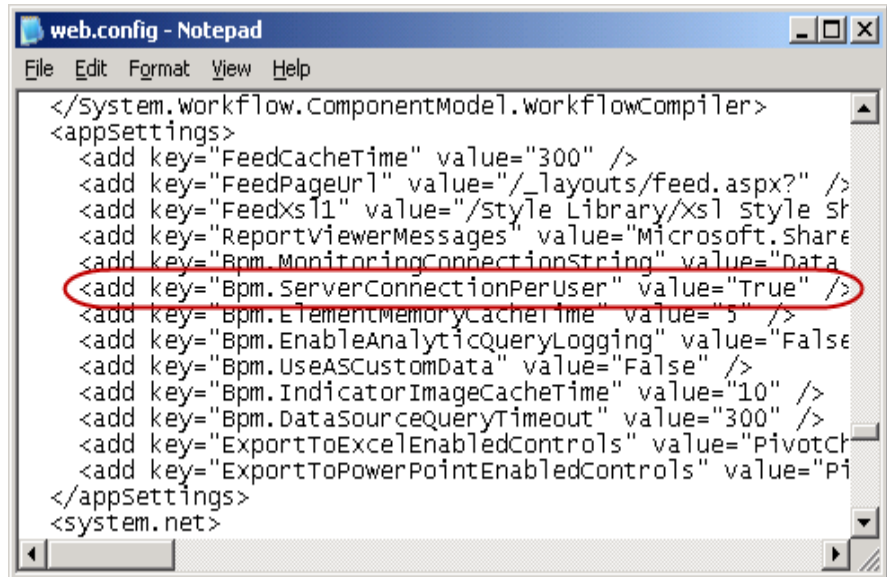


- d. Within the Properties window, select the **Home Directory** tab.
- e. Note the home directory **path** for this Web Application. This is where our web.config file is located.



3. Change the Connection Per User application setting in the Office SharePoint Server Web Application's web.config.

- a. Navigate to the Home Directory for your Office SharePoint Server Web Application.
- b. Locate the **web.config** file and open it using **Notepad** or another text editing application.
- c. Locate the **Bpm.ServerConnectionPerUser** configuration under the **appSettings** section.
- d. Change its value to **True**.



```

</System.Workflow.ComponentModel.WorkflowCompiler>
<appSettings>
  <add key="FeedCacheTime" value="300" />
  <add key="FeedPageUrl" value="/_layouts/feed.aspx?" />
  <add key="FeedXsl1" value="/Style Library/Xsl style sh
  <add key="ReportViewerMessages" value="Microsoft.Share
  <add key="Bpm.MonitoringConnectionString" value="Data
  <add key="Bpm.ServerConnectionPerUser" value="True" />
  <add key="Bpm.ElementMemoryCacheTime" value="5" />
  <add key="Bpm.EnableAnalyticQueryLogging" value="False
  <add key="Bpm.UseASCUSTOMData" value="False" />
  <add key="Bpm.IndicatorImageCacheTime" value="10" />
  <add key="Bpm.DataSourceQueryTimeout" value="300" />
  <add key="ExportToExcelEnabledControls" value="PivotCh
  <add key="ExportToPowerPointEnabledControls" value="Pi
</appSettings>
<system.net>

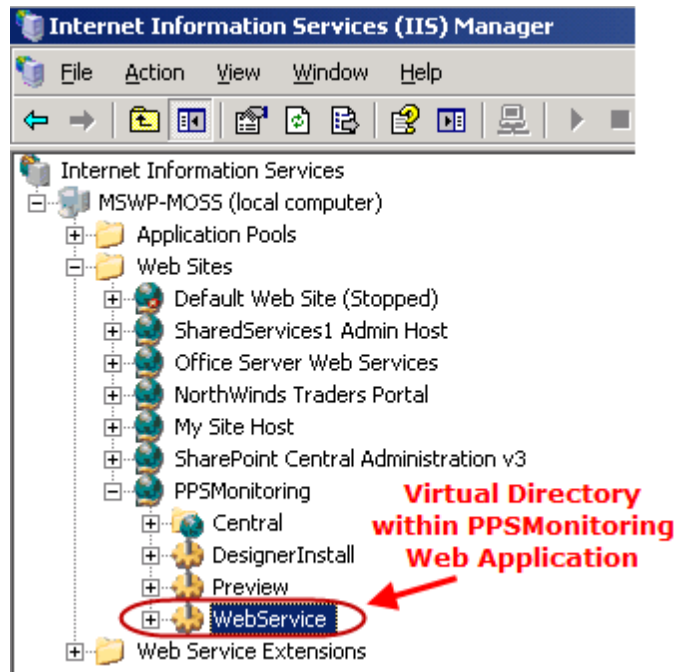
```

- e. **Save** and close the web.config file.
- f. To ensure the change is reflected by IIS, run an **iisreset** command from a command prompt.

Dashboard Designer Connection Per User

Refer to the Technology Overview for PerformancePoint Server to determine the appropriate Connection Per User setting for Dashboard Designer.

1. Determine the path to the web.config for the "WebService" virtual directory under the PPSMonitoring Web Application.
 - a. Open up **Internet Information Services (IIS) Manager** by navigating to **Start → Administrative Tools → Internet Information Services (IIS) Manager**.
 - b. Within IIS Manager, expand **[Computer Name] (local computer) → Websites → PPSMonitoring**.
 - c. Right click on the Virtual Directory named **WebService** and select **Properties**.



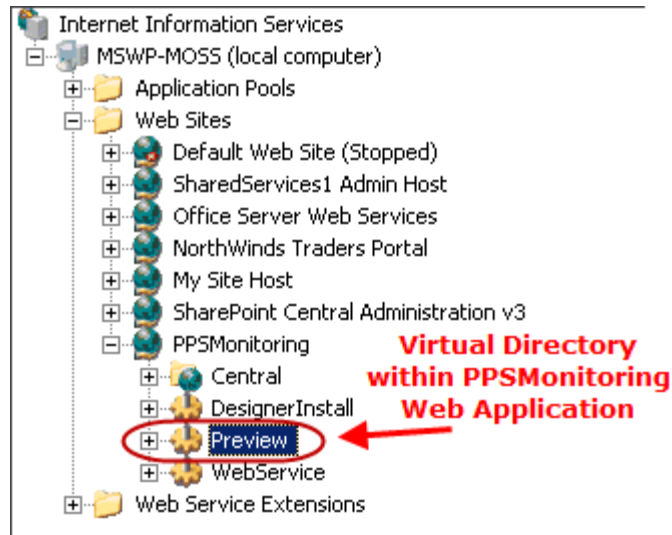
- d. Within the Properties window, select the **Home Directory** tab.
 - e. Note the Home Directory path for this virtual directory. This is where our web.config file is located.
2. Change the Connection Per User application setting in the PPSMonitoring Web Service's web.config
 - a. Navigate to the Home Directory for your PPSMonitoring WebService
 - b. Locate the **web.config** file and open it using **Notepad** or another text editing application
 - c. Locate the **Bpm.ServerConnectionPerUser** configuration under the **appSettings** section.
 - d. Change its value to **True**.
 - e. **Save** and close the web.config file.
 - f. To ensure the change is reflected by IIS, run an **iisreset** command from a command prompt.

Preview Site Connection Per User

Refer to the Technology Overview for PerformancePoint Server to determine the appropriate Connection Per User setting for the PerformancePoint Preview Site.

1. Determine the path to the web.config for the "Preview" virtual directory under the PPSMonitoring Web Application.
 - a. Open up **Internet Information Services (IIS) Manager** by navigating to **Start → Administrative Tools → Internet Information Services (IIS) Manager**

- b. Within IIS Manager, expand [**Computer Name**] (**local computer**) → **Websites** → **PPSMonitoring**
- c. Right click on the Virtual Directory named **Preview** and select **Properties**.

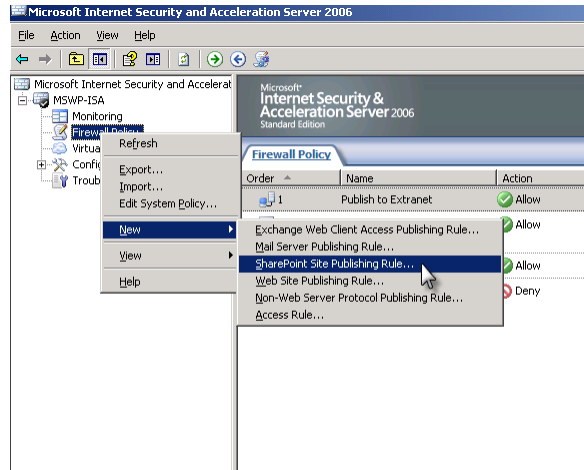


- d. Within the Properties window, select the **Home Directory** tab.
 - e. Note the home directory path for this directory. This is where our web.config file is located.
2. Change the Connection Per User application setting within the web.config for the Preview virtual directory under the PPSMonitoring Web Application.
 - a. Navigate to the Home Directory for the Preview virtual directory.
 - b. Locate the **web.config** file and open it using **Notepad** or another text editing application
 - c. Locate the **Bpm.ServerConnectionPerUser** configuration under the **appSettings** section.
 - d. Change its value to **True**.
 - e. **Save** and close the web.config file.
 - f. To ensure the change is reflected by IIS, run an **iisreset** command from a command prompt.

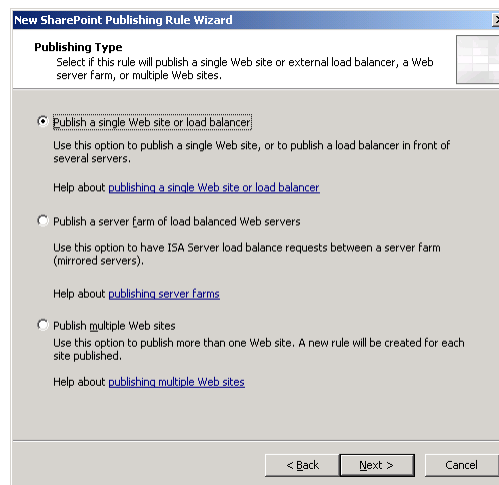
ISA Server 2006

The following steps walk through the configuration of ISA Server 2006 to publish a SharePoint site containing PerformancePoint 2007 dashboards.

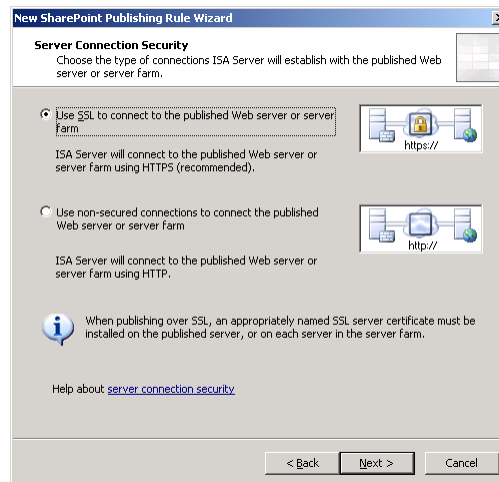
1. Create a Publishing Rule for Office SharePoint Server.
 - a. Open the **ISA Server Management** console.
 - b. Expand ISA Server → Firewall Policy
 - c. Right-click **Firewall Policy** and navigate to **New → SharePoint Site Publishing Rule**. This will start the New SharePoint Publishing Rule Wizard.



- d. On the first screen of the wizard enter a descriptive **name** for the publishing rule. Click **Next** to continue.
- e. In the **Publishing Type** section of the wizard, specify the number of servers or sites to publish by selecting the appropriate radio button. For our lab, we choose to **"Publish a single Web site or load balancer."** Click **Next** to continue.



- f. In the **Server Connection Security** section of the wizard, specify whether the connection to the server will be secured using SSL or not by choosing the appropriate radio button. For our lab, the web app uses SSL and therefore the connection will be encrypted. Click **Next** to continue.

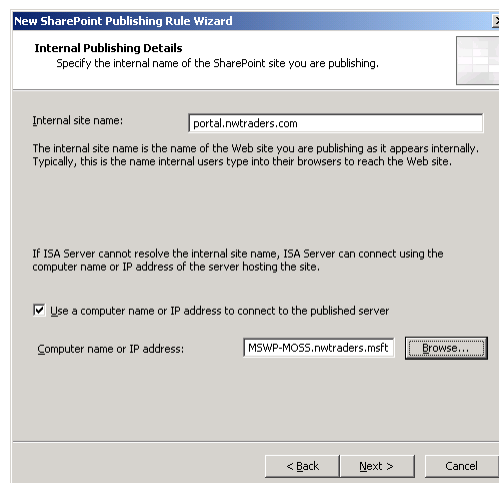


Note: This setting refers to the connection from the ISA sever to the published resource, which in this case is the server running Office SharePoint Server. This is not to be confused with the client connections to the ISA server (the actual publishing) which should always be secured with SSL. The client connection is covered later in the web listener configuration.

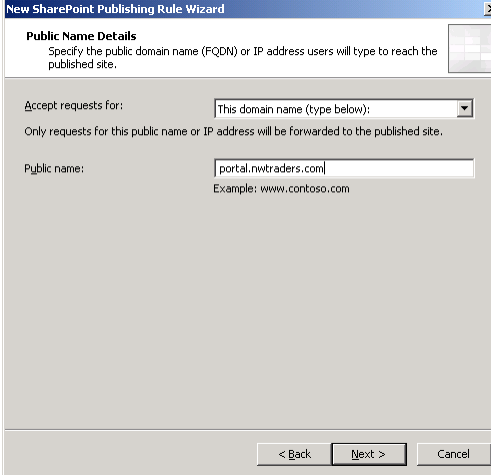
- g. In the **Internal Publishing Details** section of the wizard, enter the **Public name** in the **Internal Site name** field. For our lab, the site published is **portal.nwtraders.com**.

Note: The “public” name of the website is used here because an Alternate Access Mapping has been configured to map the internet URL as both the Internal and Public names. This was done in order for the PerformancePoint dashboards to work properly. Refer to the Alternate Access Mapping section of this document for more information.

Check the box for **“Use a computer name or IP address”** and enter the **MOSS Web Server name or IP address**, or enter a Network Load Balancer IP address. For our lab, the name of the MOSS server entered is **MSWP-MOSS.nwtraders.msft**. Click **Next** to continue.

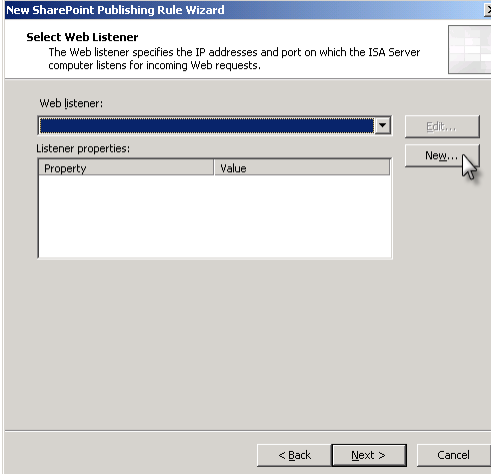


- h. In the **Public Name Details** section of the wizard, enter the Public name as the domain for which requests are accepted. For our lab, the Public name is **portal.nwtraders.com**. Click **Next** to continue.



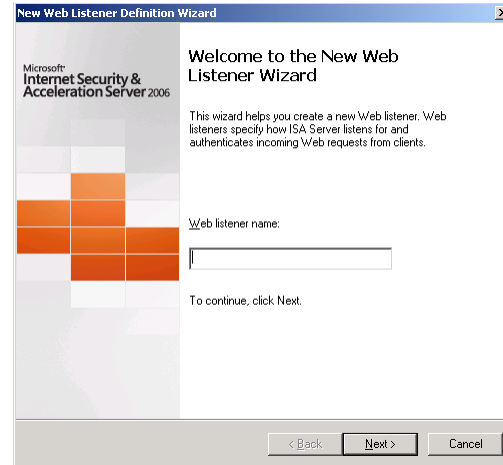
The screenshot shows the 'Public Name Details' section of the 'New SharePoint Publishing Rule Wizard'. The title bar reads 'New SharePoint Publishing Rule Wizard'. The main heading is 'Public Name Details' with a sub-heading 'Specify the public domain name (FQDN) or IP address users will type to reach the published site.' Below this, there is a section 'Accept requests for:' with a dropdown menu set to 'This domain name (type below):'. A note states 'Only requests for this public name or IP address will be forwarded to the published site.' The 'Public name:' field contains 'portal.nwtraders.com' and an example 'www.contoso.com' is shown below it. At the bottom, there are buttons for '< Back', 'Next >', and 'Cancel'.

- i. In the **Select Web Listener** portion of the wizard click the **New** button to create a new listener. This will start the New Web Listener Definition Wizard.

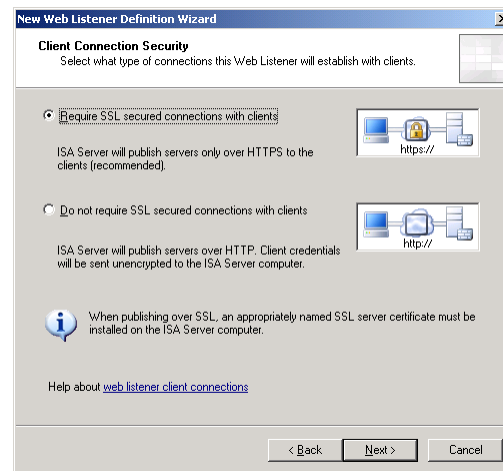


The screenshot shows the 'Select Web Listener' section of the 'New SharePoint Publishing Rule Wizard'. The title bar reads 'New SharePoint Publishing Rule Wizard'. The main heading is 'Select Web Listener' with a sub-heading 'The Web listener specifies the IP addresses and port on which the ISA Server computer listens for incoming Web requests.' Below this, there is a 'Web listener:' dropdown menu. To its right is an 'Edit...' button. Below the dropdown is a 'Listener properties:' table with two columns: 'Property' and 'Value'. To the right of the table is a 'New...' button with a mouse cursor over it. At the bottom, there are buttons for '< Back', 'Next >', and 'Cancel'.

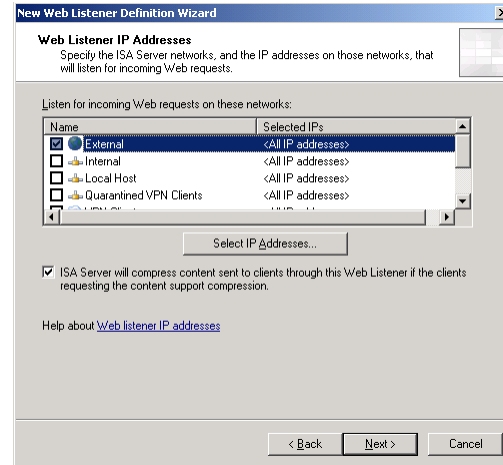
- i. In the first page of the wizard enter a descriptive **name** for the web listener. Click **Next** to continue.



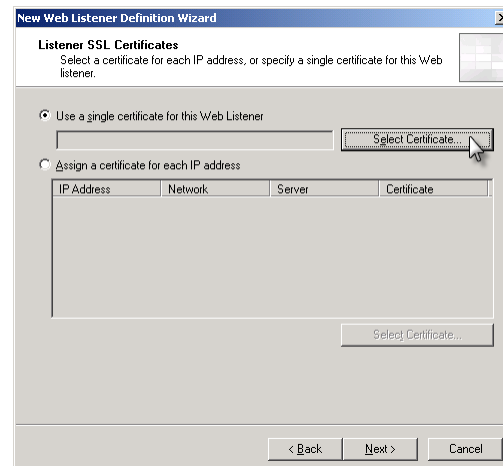
In the **Client Connection Security** section of the wizard select the **“Require SSL secured connections with clients”** radio button to enable SSL.



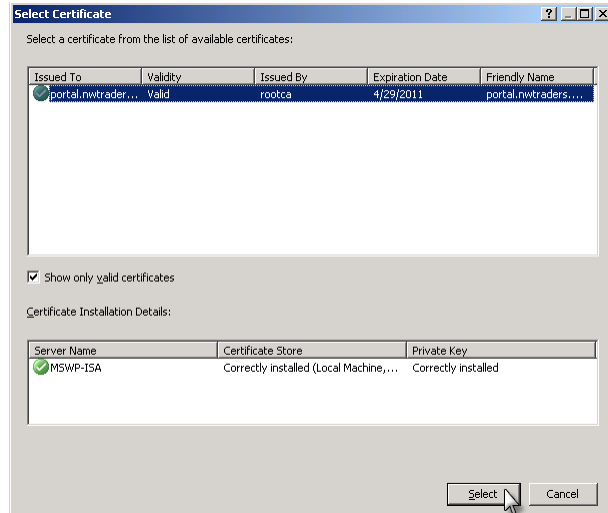
- ii. In the **Web Listener IP Addresses** section of the wizard, check the **box for the appropriate network for incoming requests**. If incoming requests should be accepted only by a specific IP address or set of addresses, click the **Select IP Addresses** button to choose IP addresses associated with the ISA server. In our lab the **External** network and all IP addresses bound to this network are accepted. Click **Next** to continue.



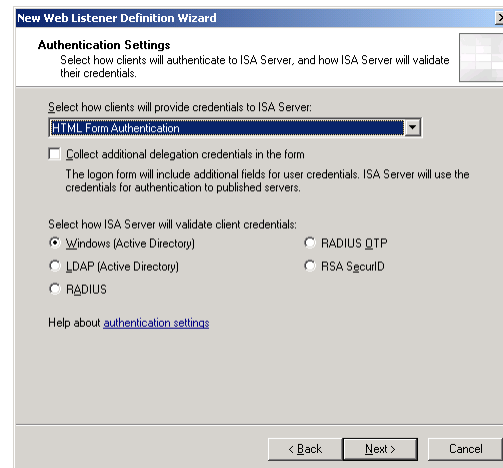
- iii. In the **Listener SSL Certificates** section of the wizard choose the appropriate radio button for your environment, whether a single certificate will be used or if each IP address assigned to the listener will have a different certificate.



- iv. Click the **Select Certificate** button to choose among the different certificates already imported into the ISA server and to assign those to the listener (or IP addresses). Once the certificate has been assigned, click the **Next** button to continue.



- v. In the **Authentication Settings** of the wizard, select the appropriate authentication method for your environment. Please note that the **Windows (Active Directory)** method of validation is required for Kerberos, and that this method is only available when the user credentials are requested via **HTML Form Authentication, HTTP Authentication-Basic, or HTTP Authentication-Integrated**. For our lab, we have chosen to use **HTML Form Authentication and Windows (Active Directory)**. Click **Next** to continue.

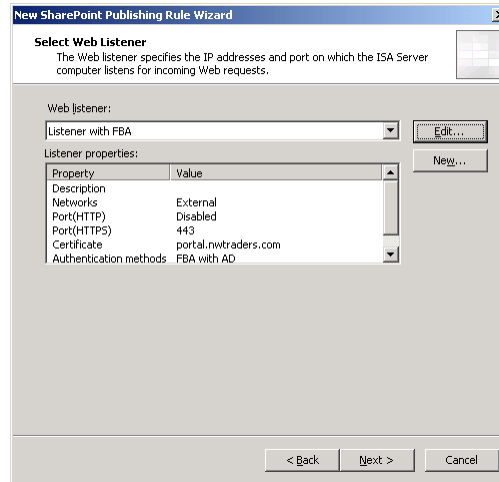


- vi. In the **Single Sign On** section of the wizard, uncheck the box to disable SSO unless your environment requires it. Click **Next** to continue.

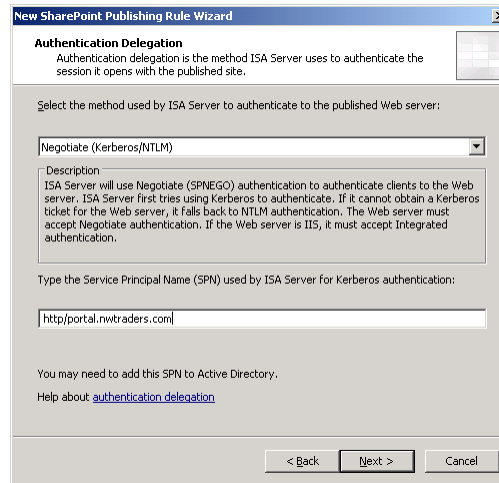
Note: SSO eliminates the multiple authentication prompts users would get when they go from one published website to another within the same web browsing session. This is particularly useful in situations where PerformancePoint has been deployed to more than one web application.

- vii. Click **Finish** to complete the Web Listener Definition Wizard.

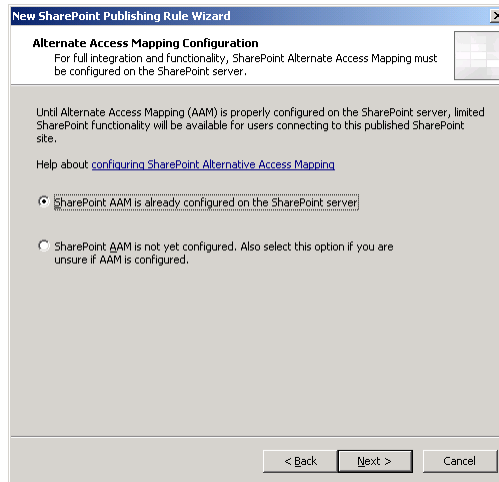
- j. Back in the New SharePoint Publishing Rule wizard, click **Next** to continue.



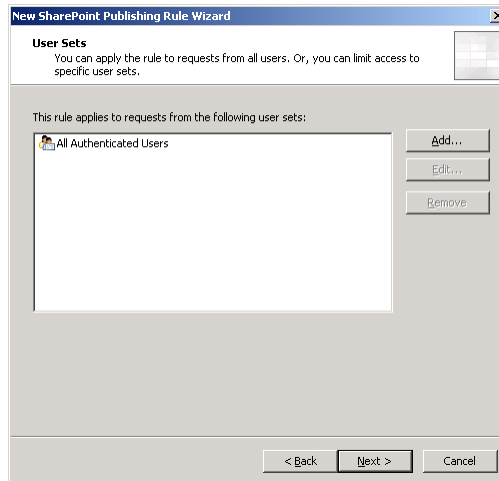
- k. In the **Authentication Delegation** section of the wizard, select **Negotiate (Kerberos/NTLM)** to authenticate against the published web server. Enter the **SPN** registered to the app pool account for the web site URL. For our lab, this SPN is **HTTP/portal.nwtraders.com**. Click **Next** to continue.



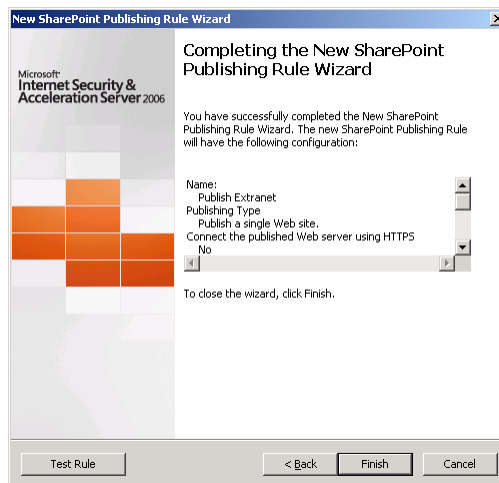
- l. In the **Alternate Access Mapping Configuration** section of the wizard, select the radio button indicating that Office SharePoint Server AAM is already configured. Click **Next** to continue.



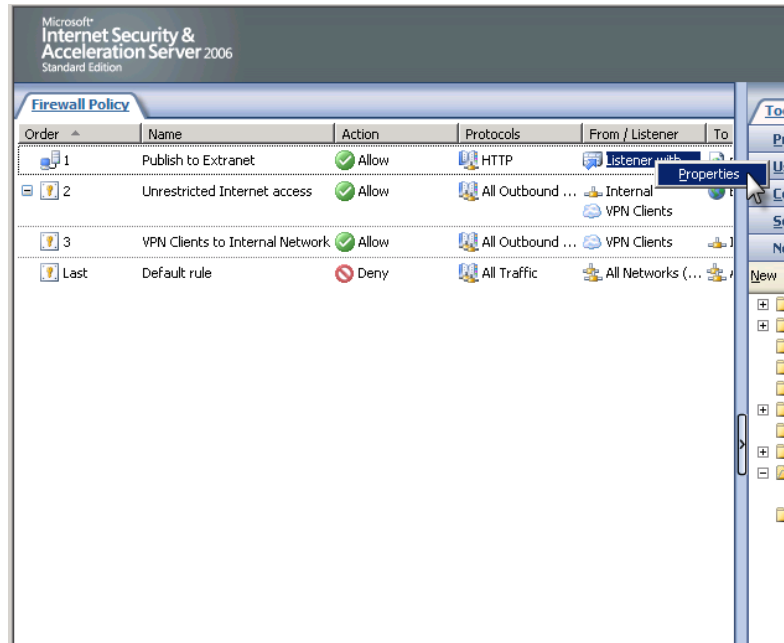
- m. In the User Sets section of the wizard, select the users for which this rule will apply. For our lab, this rule applies to **All Authenticated Users**. Click **Next** to continue.



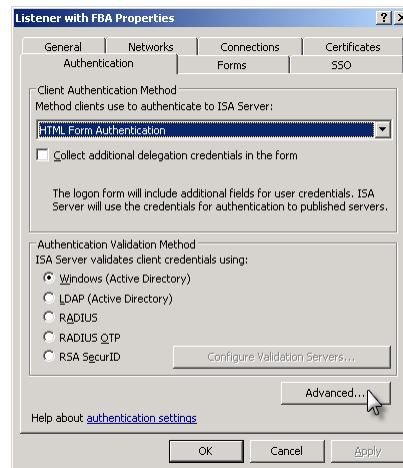
- n. Click **Finish** to complete the New SharePoint Publishing Rule wizard.



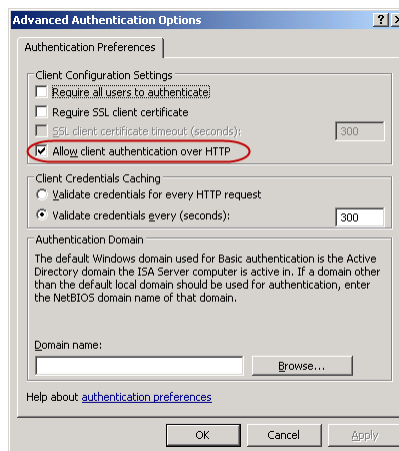
- o. If you have configured the publishing rule to use SSL, then skip to the last step. Otherwise, you must take the following additional steps.
- i. The new publishing rule should now be displayed in the list of firewall policies. **Right-click** the listener in the publishing rule and choose **properties**.



- ii. In the listener properties window, click the **Authentication** tab. Click the **Advanced** button.



- iii. Check the box for **Allow client authentication over HTTP**. Click **OK** to close the Advanced Authentication Options dialog box.



- iv. Click the **OK** button to close the listener properties window.
- p. Click **Apply** for the changes to take effect in the Firewall Policy.

Troubleshooting

Deploying PerformancePoint 2007 to an extranet requires a lot of moving parts. Troubleshooting access to SharePoint Products and Technologies pages and dashboards can be daunting at times, especially as the problems relate to Kerberos. Currently there is not a straight-forward, consolidated and transparent way of troubleshooting Kerberos authentication and delegation problems, and one must rely on a number of different tools and logs to find out where the problems are. The following tables list logs and tools that can be used when troubleshooting, as well as their purpose and where they can be found.

Table 10 - Troubleshooting Logs and Tools

Diagnostic Logs		
NAME	PURPOSE	LOCATION
Windows security event log	Logs authentication successes and failures, as well as authentication protocols used. Useful for determining whether Kerberos or NTLM is being used to authenticate a user.	Start → All Programs → Administrative Tools → Event Viewer. By default, only successes are logged, unless specified in the local security policy of the server, or via Group Policy.
SharePoint Logs	Log SharePoint Products and Technologies activity. The amount of logging can be throttled. Useful for troubleshooting access to SharePoint Products and Technologies pages.	C:\Program Files\Common Files\Microsoft Shared\web server extensions\12\LOGS
Kerberos Logging (Windows event log)	Logs Kerberos activity. This log is turned off by default, and should only be turned on during troubleshooting. While troubleshooting, this log should be enabled on every server involved in Kerberos authentication.	KB262177 describes how to turn on Kerberos logging. The logs can be viewed in the Windows system event log.
IIS Logs	Log IIS activity. Useful for determining whether IIS is responding to client requests for a specific web app, and for determining the username that was passed to IIS from the reverse proxy (ISA Server).	By default the logs are found in C:\WINDOWS\system32\Logfiles

ISA Server logging	Logs ISA server activity. Useful for troubleshooting connectivity problems between the clients and ISA server, and between ISA server and the published resource.	[ISA Server Name] → Monitoring → Logging (tab).
--------------------	---	---

Troubleshooting Tools		
NAME	PURPOSE	WHERE TO FIND
Microsoft Network Monitor (NETMON)	Network capture and analyzing tool. Useful for inspecting the network interactions between clients and servers down to the network packet level.	KB933741 explains Network Monitor 3, and how to obtain it.
Fiddler	HTTP debugging proxy that attaches to Internet Explorer and inspects all HTTP traffic between the client and the server. Useful for diagnosing connectivity and authentication problems.	This tool can be downloaded directly from its developer here http://www.fiddlertool.com
LDP	Tool which can be used to browse Active Directory via the LDAP protocol. Useful in searching for missing or duplicate Service Principal Names (SPNs).	The tool can be found in the Windows Support Tools which is included with Windows Server 2003. The tool is also included in Windows Server 2008. See KB321044 for instructions on how to use LDP to search for duplicate SPNs.
LDIFDE	Tool which can be used to search Active Directory for missing or duplicate SPNs.	This tool is included as part of a standard Windows Server 2003 and Windows Server 2008 installation. See KB321044 for instructions on how to use LDP to search for duplicate SPNs.
QUERYSPN.VBS	VBScript used to search for missing or duplicate SPNs.	The script is available in the Microsoft scripting library here .
DELEGCONFIG	ASP.NET application which diagnoses Kerberos configuration problems in web applications. It can detect whether SPNs have been registered and whether service accounts have been configured for delegation. It can test and diagnose connectivity from clients to web servers to back	The tool can be downloaded from here .

	end databases (double-hop).	
--	-----------------------------	--

Common Kerberos Issues

One of the most difficult components to troubleshoot in a PerformancePoint 2007 extranet deployment is the configuration of Kerberos authentication and delegation. Following are some common issues that result from improper Kerberos configuration along with advice on what to do to troubleshoot them.

SharePoint Products and Technologies pages are accessible but dashboards do not render

When SharePoint Products and Technologies pages are accessible but the dashboards are not, this is typically a sign that authentication has not been successfully passed between SharePoint Products and Technologies and SQL Server Analysis Services. Typically, an error similar to the following is displayed:

The PerformancePoint Server could not connect to the specified data source. Verify that either the current user or application pool user has Read permissions to the data source, depending on your security configuration. Also verify that all required connection information is provided and correct.

Contact the administrator for more detail.

There are a few reasons why authentication would be passed unsuccessfully or incorrectly:

- Missing or mistyped SPN entries for the application pool account would cause Kerberos authentication to SharePoint Products and Technologies to fail, and therefore SharePoint Products and Technologies would fall back to NTLM. SharePoint Products and Technologies would then use the app pool account to connect to SSAS since it is unable to delegate using Kerberos, causing SSAS to deny the request.
- The same SPN registered to more than one account (duplicates) would also cause Kerberos authentication to fail and forcing SharePoint Products and Technologies to fall back to NTLM.
- The application pool account is not trusted for delegation in Active Directory. This would allow Kerberos authentication to work, however SharePoint Products and Technologies would not be able to pass the user credentials to SSAS using Kerberos.
- The ISA Server publishing rule is not configured to delegate credentials using Kerberos. In this scenario ISA server may be configured to pass the credentials to SharePoint Products and Technologies using NTLM in which case authentication to SharePoint Products and Technologies works, but delegation to SSAS fails.
- The ISA Server publishing rule specifies the wrong SPN. Depending on the rule's delegation configuration, this issue may result in dashboards that do not render or in Access Denied errors to the page.

Access to SharePoint Products and Technologies pages is denied or not working

This is usually the result of a misalignment between the authentication method of the ISA Server listener, the delegation method for the rule, and the authentication method set in SharePoint Products and Technologies. A typical working configuration would be Forms Based Authentication with Windows Authentication (ISA Listener) → Kerberos Constrained Delegation (ISA Rule) → Windows Negotiate (SharePoint Authentication Provider).

Kerberos Troubleshooting Strategies

When troubleshooting Kerberos issues, a handful of tools helps to diagnose and resolve those issues. See Table 10 for a list of commonly used tools. Following are strategies on how to find and diagnose Kerberos problems.

1. Check the windows security logs on the server(s) running SharePoint Products and Technologies and the SSAS server for logon/logoff events. If the logs identify the users by name, and both the Logon Process and Authentication package are set to Kerberos, it means that the user has been successfully authenticated using Kerberos. If the event appears on the SSAS server, it means that delegation from the app pool account to SSAS was successful. Conversely, if the Logon Process states NTLM or if the logs in SSAS indicate an ANONYMOUS user, then Kerberos authentication is not working. Time to continue troubleshooting.
2. Turn on failure auditing for logon/logoff events in the local security policy of the servers, or via group policy. This will provide more detail on Kerberos failures and why authentication may be falling back to NTLM.
3. Turn on Kerberos logging ([KB262177](#)). This will enable Kerberos activity to be logged in the System event log. This provides a greater level of detail on Kerberos errors.
4. Search for missing, duplicate, or mistyped SPNs. For each web app that is going to authenticate using Kerberos, there must be a SPN with the URL for the web app registered to the Identity account of the application pool running the web app. Additionally, the service account running the SSAS service must have a SPN registered to it with the name of the SQL server and/or instance. SPNs must be registered to only one account in Active Directory.
5. Verify that delegation is configured properly. The application pool account must be trusted for delegation to the SSAS service account specifically. The DELEGCONFIG tool is very useful in determining whether delegation is configured properly and working.
6. Confirm that the Authentication Provider of the web application hosting the site collection for the dashboard pages is configured as Windows → Negotiate (Kerberos), and that anonymous access is not turned on.

Supported Configurations

PerformancePoint 2007 is only supported on the following platforms.

Server Operating System Versions (Non Domain Controllers)

Supported

Windows Server 2008
Windows Server 2003

Not Supported

Windows 2000 Server

PerformancePoint Server 2007 can run on a Windows Server 2008 as well as 2003. The same is true for Office SharePoint Server 2007. Note that the installation of Office SharePoint Server 2007 on Windows Server 2008 requires Office SharePoint Server 2007 SP1 slipstreamed into the installation source.

Server Operating System Versions (Domain Controllers)

Supported

Windows Server 2003
Windows 2000 Server

Not Supported

Windows Server 2008

The current implementation of Kerberos in Windows Server 2008 prevents PerformancePoint Server 2007 from functioning correctly under certain scenarios and therefore is not currently supported. Kerberos constrained delegation (KCD) is not available in Windows 2000, as well as in Windows Server 2003 domains not running at the Windows Server 2003 functional level.

Windows SharePoint Services Versions

Supported

Windows SharePoint Services 3.0
Microsoft Office SharePoint Server 2007

Not Supported

Windows SharePoint Services 2.0
SharePoint Portal Server 2003

SQL Server Analysis Services Versions (Cubes)

Supported

SQL Server 2008
SQL Server 2005

Not Supported
SQL Server 2000

Client Web Browsers

Supported
Internet Explorer 6
Internet Explorer 7
Internet Explorer 8 (with PerformancePoint Server 2007 SP3)

Not Supported
Mozilla
Firefox
Safari
Opera

Although it is possible to render SharePoint Products and Technologies content on non-IE browsers, the functionality of SharePoint Products and Technologies is greatly reduced on those platforms.

Reverse Proxy

Supported
ISA Server 2006

Not Supported
ISA Server 2004

Conclusion

As you have seen, Microsoft's Business Intelligence platform does support extranet solutions with the proper application of authentication technologies and configurations. Once setup, users are provided a seamless single-sign-on experience, and are shielded from the fact that they are authenticating via Active Directory environment using an Active Directory account. Users are uniquely authenticated, and their credentials are passed between systems allowing granular control of the data, pages and objects they can access. As a conclusion to this white-paper, we share the following thoughts:

1. As with any technology solution, it is critical to capture requirements up-front to ensure that the infrastructure will support the solution, and to provide end-users with the experience they desire.
 2. The lab configuration used in this white-paper should meet the requirements of most extranet Business Intelligence solutions, and is supported by Microsoft.
 3. Over time, extranet solutions will become more widespread as organizations look for new ways to automate manual processes while providing a higher level of service to their external users and customers.
 4. In the future, as Microsoft's Business Intelligence platform becomes more integrated, configuring an extranet Business Intelligence infrastructure will become simpler. Complexities will be mitigated and fewer steps will be required to configure an extranet solution.
-

References

This section provides descriptions and links to any material that was referenced throughout this document.

CUSTOMDATA() MDX Function

<http://msdn.microsoft.com/en-us/library/ms145582.aspx>

The Microsoft PerformancePoint Team Blog

<http://blogs.msdn.com/performancepoint/>

TechNet article on configuring the Monitoring Server for Kerberos Delegation

<http://technet.microsoft.com/en-us/library/bb838742.aspx>

Microsoft's Implementation of Kerberos in Active Directory

<http://technet.microsoft.com/en-us/library/cc739058.aspx>

Glossary

- **Active Directory** – Is Microsoft's Directory Service, providing authentication, system management, user management, and other core functionality for managing security and membership to a network.
- **Alternate Access Mappings (AAM)** - Enables SharePoint Products and Technologies to build the links within the pages it generates with the correct URL's based on where the request came from (intranet, extranet, etc). This allows a single web application to be accessed via different addresses (<http://extranet.com> vs. <http://intranet>) and have all links re-mapped at run-time providing a consistent experience for all users.
- **Application Layer Filtering** - The capability of firewall and reverse proxy devices to inspect the portion of network packets that relate to the application layer protocols of the OSI model. Also known as Layer 7 filtering, this capability provides much more comprehensive protection against network threats.
- **Application Pool** - Component of Internet Information Services (IIS) which isolates web applications into dedicated worker processes (w3wp.exe). Application pools have configurable memory and CPU utilization parameters, and multiple web apps can be hosted by a single application pool.
- **Application Pool Identity** - The account used to run the worker process (w3wp.exe) associated with an application pool.
- **Authentication** - The process of validating a user's given credentials against a membership or authentication provider. User credentials are typically given to an authentication process in the form of a username and a password.
- **Authentication Provider** - The entity in a system that is responsible for enforcing authentication. Also refers to the authentication mechanism preference of a web application in SharePoint Products and Technologies.
- **Demilitarized Zone (DMZ)** – A DMZ is a network placed between a company's private network (intranet) and the general public network (internet). Resources within the DMZ can be accessed over the public internet, and the DMZ can access intranet resources through secure connections to the internal network.
- **Extranet** - An extranet is a privately-held network which certain other external users such as customers or business partners can access. Extranets are unique in that they are private in nature, yet they are typically accessible over the public Internet. Extranets can be viewed as publicly reachable websites that are made accessible only to a select group of individuals.

- **Fully Qualified Domain Name (FQDN)** – Represents the complete path to a location starting with the top level domain (ex: .com) down to the host name (server.domain.com).
- **Impersonation** – When a service uses another account and password to authenticate to another resource.
- **Internet** – The internet is a public network that can be accessed by anyone.
- **Internet Information Services (IIS) Metabase** - The configuration file for Internet Information Services.
- **Internet Security and Acceleration (ISA) Server 2006** - An application-layer filtering capable firewall that among other things provides network edge protection, secure application publishing, web caching, VPN, and proxy services.
- **Intranet** - An intranet is a privately-held network that can only be accessed internally within an organization. Resources on an intranet cannot natively be accessed from outside the internal network, i.e., over the internet.
- **Kerberos** - Is a mutual authentication mechanism that builds on symmetric key cryptography and requires that two entities both trust a third party authority in order to authenticate with each other
- **Kerberos Constrained Delegation** – Is a feature of Active Directory that enables specific servers or services with registered Service Principal Names to request service tickets on behalf of other users or services, and present those tickets to any additional servers or service.
- **Local Area Network (LAN)** -
- **Microsoft Office SharePoint Server 2007** – Runs on top of Windows SharePoint Services 3.0, providing a set of core functionality and features that support the creation of Enterprise collaboration portals.
- **NetBIOS Name** – The name of a host minus its primary DNS suffix. For a system with a FQDN of "server.domain.com", the NetBIOS name would be "server".
- **NTLM** – Is a Microsoft based challenge-response authentication protocol. NTLM has been replaced by Kerberos as Microsoft's preferred and more secure authentication protocol.
- **Pre-authentication** - A process implemented by ISA Server 2006 which validates user credentials before sending them over to the server hosting the resources sought by the user.
- **Reverse Proxy** - A server or service that stands between users on the Internet and corporate resources in a LAN, and that provides access to the resources in the LAN on behalf of the users on the Internet. By virtue of their function reverse proxies typically provide edge firewall services.
- **Secure Sockets Layer (SSL)** – This is an encryption protocol utilizing two keys, a public key and a private key. SSL can be used to create a secure connection between two resources over which information can be transmitted securely.
- **Service Principal Name (SPN)** - the name by which a client uniquely identifies an instance of a service.
- **Site Collection** - A group of sites and sub sites in SharePoint Products and Technologies which share features, security, a common URL namespace, and a common navigational structure.
- **Virtual Private Networks (VPN)** – Are comprised of protocols that encrypt and encapsulate TCP/IP packets as they are transmitted over the public Internet. This "tunnel" created over public networks enables only the sender and the intended receiver to interpret the information that is being transmitted.
- **Web Application** - An application that uses the Hyper-Text Transfer Protocol (HTTP) and HTML-related technologies to render content on a web browser. In SharePoint Products and Technologies a web application is an object containing a root URL and content databases, and that is associated with an IIS virtual directory.
- **Web Publishing** - The process, technology, or method for making resources in a LAN accessible via the public World Wide Web.
- **Windows SharePoint Services 3.0** – Is the foundation for Microsoft's SharePoint products and technologies, and provides the base functionality for the

creation of web portals and collaboration sites. Windows SharePoint Services 3.0 is a free download that can be installed on Windows Servers.