Kerberos Authentication and delegation

Applies To: Windows 8, Windows Server 2012

Kerberos is an authentication protocol that is used to verify the identity of a user or host. This topic contains information about Kerberos authentication in Windows Server 2012 and Windows 8.

Feature description

The Windows Server operating systems implement the Kerberos version 5 authentication protocol and extensions for public key authentication, transporting authorization data, and delegation. The Kerberos authentication client is implemented as a security support provider (SSP), and it can be accessed through the Security Support Provider Interface (SSPI). Initial user authentication is integrated with the Winlogon single sign-on architecture.

The Kerberos Key Distribution Center (KDC) is integrated with other Windows Server security services that run on the domain controller. The KDC uses the domain's Active Directory Domain Services database as its security account database. Active Directory Domain Services is required for default Kerberos implementations within the domain or forest.

Practical applications

The benefits gained by using Kerberos for domain-based authentication are:

- **Delegated authentication.**

  Services that run on Windows operating systems can impersonate a client computer when accessing resources on the client's behalf. In many cases, a service can complete its work for the client by accessing resources on the local computer. When a client computer authenticates to the service, NTLM and Kerberos protocol provide the authorization information that a service needs to impersonate the client computer locally. However, some distributed applications are designed so that a front-end service must use the client computer's identity when it connects to back-end services on other computers. Kerberos authentication supports a delegation mechanism that enables a service to act on behalf of its client when connecting to other services.

- **Single sign on.**

Using Kerberos authentication within a domain or in a forest allows the user or service access to resources permitted by administrators without multiple requests for credentials. After initial domain sign on through Winlogon, Kerberos manages the credentials throughout the forest whenever access to resources is attempted.

- **Interoperability.**

The implementation of the Kerberos V5 protocol by Microsoft is based on standards-track specifications that are recommended to the Internet Engineering Task Force (IETF). As a result, in Windows operating systems, the Kerberos protocol lays a foundation for interoperability with other networks in which the Kerberos protocol is used for authentication. In addition, Microsoft publishes Windows Protocols documentation for implementing the Kerberos protocol. The documentation contains the technical requirements, limitations, dependencies, and Windows-specific protocol behavior for Microsoft's implementation of the Kerberos protocol.

- **More efficient authentication to servers.**

Before Kerberos, NTLM authentication could be used, which requires an application server to connect to a domain controller to authenticate every client computer or service. With the Kerberos protocol, renewable session tickets replace pass-through authentication. The server is not required to go to a domain controller (unless it needs to validate a Privilege Attribute Certificate (PAC)). Instead, the server can authenticate the client computer by examining credentials presented by the client. Client computers can obtain credentials for a particular server once and then reuse those credentials throughout a network logon session.

-

- **Mutual authentication.**

By using the Kerberos protocol, a party at either end of a network connection can verify that the party on the other end is the entity it claims to be. NTLM does not enable clients to verify a server's identity or enable one server to verify the identity of another. NTLM authentication was designed for a network environment in which servers were assumed to be genuine. The Kerberos protocol makes no such assumption.

How to Setup Kerberos Constrained Delegation to use Single Sign On (Password Manager) and Smartcard Authentication from Clients not Joined to the Domain

Document ID: **CTX134070** / Created On: Oct 10, 2012 / Updated On: Oct 10, 2012

Average Rating: not yet rated

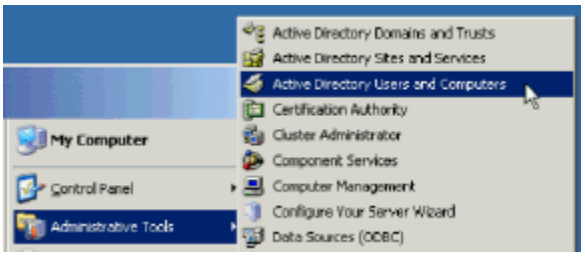
View products this document applies to

Summary

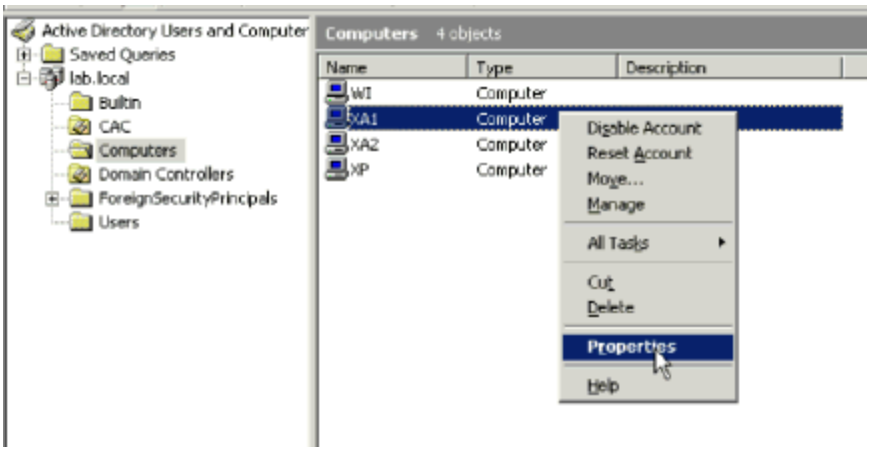
When utilizing constrained delegation in combination with Smartcard authentication on client machines that are not in the same domain as the Kerberos environment, we need to setup Protocol Transition. This is for the transition between Smartcard and Kerberos for the ICA session. If this is not set, Single Sign On (CPM) loses its authentication to the domain after the Kerberos ticket has expired within the ICA session.

Procedure

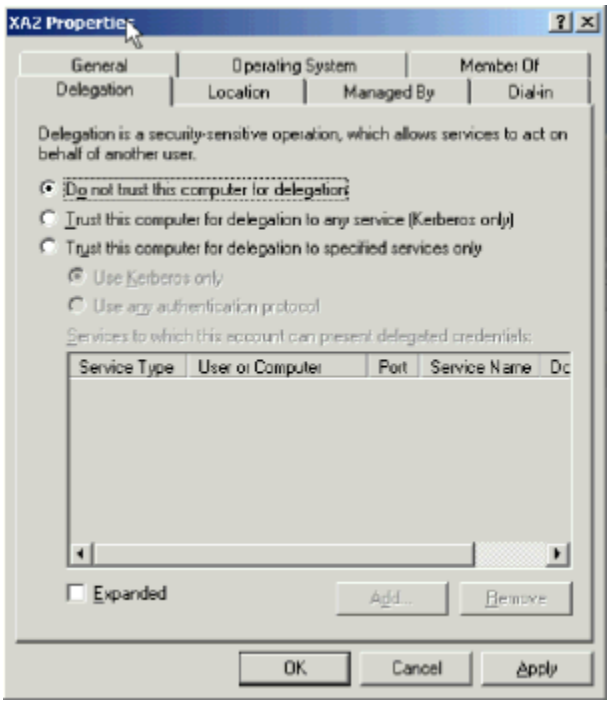
1. Open **Active Directory Users and Computers** console.



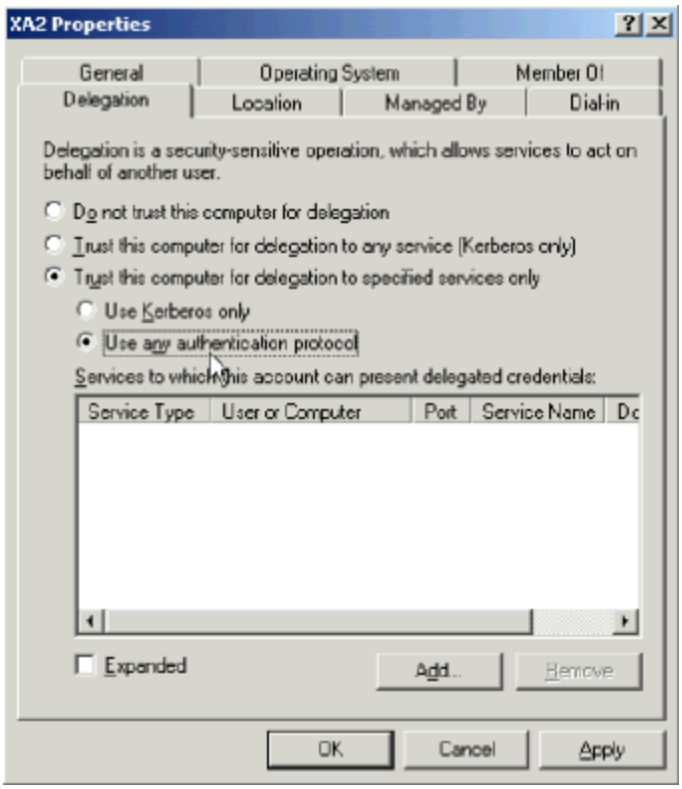
2. Locate the XenApp servers which use the Kerberos delegations. Select the server, **Right-Click** and select **Properties**.



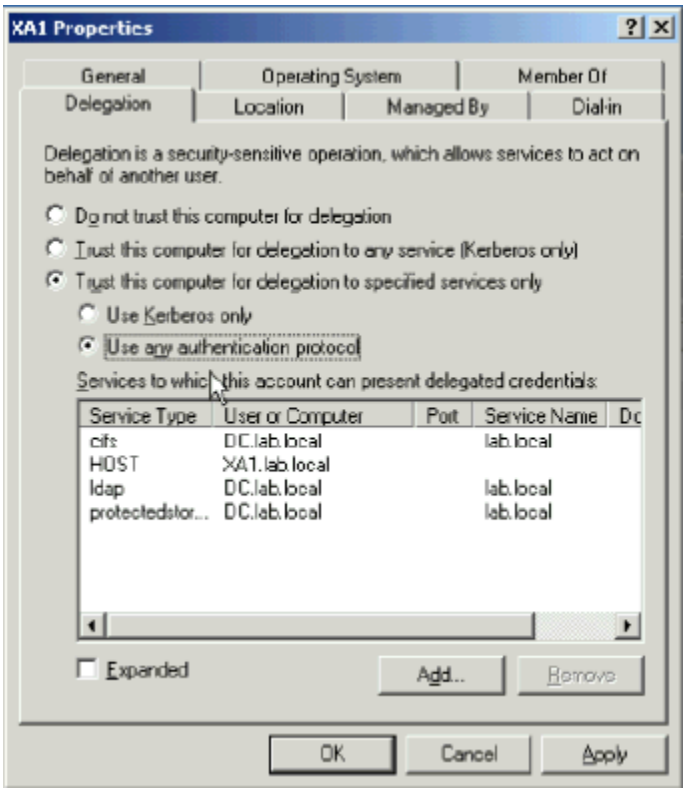
3. In **Properties**, select the **Delegation** tab.



4. In the Delegation tab select the **Trust this computer for delegation to specified services only** option.
   Select **Use any authentication protocol** option.
   Note: This Use any authentication protocol option enables Protocol Transition so it is the most important setting.



5. Add the following services for the Domain Controller and the XenApp servers in the farm and press **OK** to save the settings.



Add each domain controller and select the following services:

- Service: CIFS, LDAP, ProtectedStorage

Add each XenApp server and select the following services:

- Service: HOST

6. Locate the server that must be the Web Interface and open the Delegation Properties same as Step 1 – 3. On the Delegation Properties select first 2 options similar to what was selected for the XenApp services. Then add the service mentioned below, press **OK** to save the settings.

Add each Web Interface server and select the following services:

- Service: HTTP

IMPORTANT! It is important that these settings are setup correctly. If not set correctly, the authentication from non-domain joined machines or machines that are attached to a different domain, will have problems. Also, if the option for Kerberos Only is selected, the data store synchronization may have LDAP errors and problems may occur accessing the data store after the Kerberos ticket expires. Without protocol transition it is not possible to maintain the authentication in the session, when the end user is using a machine not in the same domain where the XenApp farm is located.

How Windows Server 2012 Eases the Pain of Kerberos Constrained Delegation

Let's talk authentication—specifically, Kerberos constrained delegation. But first, if you're new to Kerberos or need a quick refresher, I would suggest that you read the Ask the Directory Services Team blog posts "Kerberos for the Busy Admin" and "Understanding Kerberos Double Hop" to get up to speed on the terminology and concepts contained throughout this article. Part 1 of this article describes the benefits of using Kerberos constrained delegation with Windows Server 2012 over earlier versions of Windows Server. Part 2 will provide a technical walkthrough of how Kerberos constrained delegation works in Server 2012.

What Is Constrained Delegation?

Constrained delegation lets you limit the back-end services for which a front-end service can request tickets on behalf of another user. A common example of constrained delegation is the web-browser-to-IIS-to-SQL-Server scenario. In this scenario, a user navigates to a web-based reports server hosted on Microsoft IIS, which retrieves data using an authenticated connection to a Microsoft SQL Server system. The IIS server needs to authenticate to the SQL Server system on behalf of the user. Through Kerberos delegation, the IIS server (i.e., the front-end service) can request a service ticket for any service (i.e., back-end service)—not just SQL Server—on behalf of the user. This means that the IIS server can essentially authenticate on behalf of the user to SQL Server, a file share, or a web service. Using constrained delegation, you can limit the IIS server (the front end) so that it can authenticate the user only to SQL Server (the back end) and no other service or application.

Kerberos constrained delegation has been a part of the OS since Windows Server 2003. It requires you to configure an allow list of service principal names (SPNs) on user or computer objects in Active Directory (AD). You add the list of SPNs that represent the back-end services to which a front-end service is allowed to request tickets on behalf of the user to the ms-DS-Allowed-To-

Delegate-To attribute of the principal under which the application or service on the front-end server runs. In the previous example, the front-end service is IIS and the back-end service is SQL Server. To constrain the delegation for IIS, you would add SPNs for the SQL Server instances running on the SQL Server system to the ms-DS-Allowed-To-Delegate-To attribute on the IIS computer account in AD—or the user account running the IIS application pool. This model constrains the front-end service to only request service tickets that are listed in the ms-DS-Allowed-To-Delegate-To attribute. The downside of this delegation model is that it relies heavily on SPNs.

Service Principal Names

SPNs are difficult to manage. Although they're simple in concept, SPNs can cause a significant amount of frustration, stemming from unique constraints associated with using them.

Kerberos uses SPNs to identify the security principal responsible for running an application or service. This enables the Key Distribution Center (KDC) to encrypt tickets and keys with the correct hash so that the security principal (running the service or application) can decrypt the service ticket upon receiving the AP_REQ. This design requires that SPNs registered on security principals be unique for the AD forest. An SPN registered on multiple security principals will cause authentication to fail.

Constrained delegation appears to contradict the basic rule of registering a duplicate SPN; however, this is only in appearance. The KDC doesn't look at the ms-DS-Allowed-To-Delegate-To attribute when trying to map an SPN to a security principal. Therefore, the unique SPN requirement is limited only to values in the servicePrincipalName attribute. To constrain the delegation of a service or application, you must list the service's SPNs on the security principal that runs the application on the front-end server.

Managing the number of SPNs, knowing when and where to register them, and avoiding duplicates is cumbersome. This makes constrained delegation difficult to implement, maintain, and troubleshoot.

Point of Delegation

Constrained delegation is a model that controls delegation on the front-end server. Most delegation models manage the point of delegation closest to the resource (i.e., on the back end). Implementing delegation on the front end removes control from the resource administrator and places it on the administrator of the front-end server (and application). This model prevents the resource administrator from managing access to the resource. The current model requires domain administrative privileges to modify the ms-DS-Allowed-To-Delegate-To attribute, thus adding more administrative overhead to the management of constrained delegation.

Scope of Delegation

Scope of delegation refers to the limit to which the delegation extends from the front-end server to the back-end server. The current constrained delegation model scope is limited to the domain, meaning that the security principal under which the application or service runs can forward constrained delegated tickets only to an application or service running under a security principal in the same domain. You can't use constrained delegation across domain or forest trusts.

What Server 2012 Brings

Server 2012 introduces a new kind of Kerberos constrained delegation that addresses many of the shortcomings that exist with the previous constrained delegation model. The new implementation of constrained delegation removes the dependencies on SPNs for delegation configuration, removes the need for domain administrative privileges, enables the resource administrator to own the delegation experience, and increases the scope of delegation.

Constrained delegation in Server 2012 introduces the concept of controlling delegation of service tickets using a security descriptor rather than an allow list of SPNs. This change simplifies delegation by enabling the resource to determine which security principals are allowed to request tickets on behalf of another user.

Figure 1 shows a sample scenario. A server in Domain A runs an IIS application. The security principal under which the IIS application's AppPool runs has the SPNs registered for the front-end service or application (HTTP/app1.contoso.com). These SPNs allow the user to authenticate to the front-end server using normal Kerberos authentication.

Figure 1: Constrained Delegation in Server 2012

The application retrieves data from a back-end server in Domain B running SQL Server. The security principal running the SQL Server service has the SPNs registered for SQL Server and SQL Server instances. Again, this configuration is normal to enable Kerberos authentication. The KDC in the domain hosting the security principal running SQL Server receives a Service-for-User-to-Proxy (S4U2Proxy) Ticket Granting Service (TGS) request from the IIS server on behalf of another user. The KDC reads the security descriptor stored in the msDS-AllowedToActOnBehalfOfOtherIdentity attribute on the security principal running the SQL Server service and performs an access check using the identity under which the IIS Application Pool runs. A successful access check allows the authentication process to continue, whereas an unsuccessful access check fails the authentication attempt.

Resource-based constrained delegation functions correctly regardless of domain functional level and number of domain controllers (DCs) running a version of Windows Server prior to Server 2012, provided you have at least one Server 2012 DC in the same domain as the front-end server and one Server 2012 DC in the domain hosting the back-end server. When the domain is a hybrid domain (both Server 2012 DCs and DCs running an earlier version of Windows Server), then Windows 8 and Windows 2012 computers ensure they use a Server 2012 DC to use resource-based constrained delegation by deliberately locating a Server 2012 DC.

Requirements

The new implementation of Kerberos constrained delegation has the following requirements:

- Server 2012 KDCs must reside in the front-end account domain
- Server 2012 KDCs must reside in the back-end account domain
- The front-end server must run Server 2012

Server 2012 KDCs are required for this feature because these are the only KDCs that know how to return referred S4U2Proxy requests and use the new msDS-AllowedToActOnBehalfOfOtherIdentity attribute on the service account. The front-end server requires Server 2012 because the version of Kerberos on these servers understands that it must chase S4U2Proxy referrals to trusted domains and forests.

Management

You manage Server 2012 Kerberos constrained delegation using Windows PowerShell. Use the following Windows PowerShell cmdlets to manage constrained delegation. Typically, you'll want to use the Get-ADUser, Get-ADComputer, or Get-ADServiceAccount of the principal running the front-end service and pass that principal object as the argument value to the -PrincipalsAllowedToDelegateToAccount argument.

```
Set-ADComputer computerName -PrincipalsAllowedToDelegateToAccount
principal1, principal2, ...
Set-ADUser userName -PrincipalsAllowedToDelegateToAccount principal1,
principal2, ...
Set-ADServiceAccount serviceAccountName -
PrincipalsAllowedToDelegateToAccount principal1, principal2, ...
Get-ADComputer computerName -Property PrincipalsAllowedToDelegateToAccount
Get-ADUser userName -Property PrincipalsAllowedToDelegateToAccount
Get-ADServiceAccount serviceName -Property
PrincipalsAllowedToDelegateToAccount
```

Stay Tuned for More

This article identifies the challenges IT pros face when implementing Kerberos constrained delegation. Server 2012 provides a compelling answer to these problems by introducing resource-based Kerberos constrained delegation. In Part 2 of this series, I'll provide a more in-depth analysis of how resource-based constrained delegation works and look at the flow of authentication that's exchanged between computers running Server 2012.