How to use SPNs when you configure Web applications that are hosted on Internet Information Services

Summary

This article describes service principal names (SPNs). This article also describes how to use SPNs when you configure Web applications that are hosted on Microsoft Internet Information Services (IIS). This article also describes the Negotiate process in Windows Integrated authentication. The Negotiate security header lets clients select between Kerberos authentication and NTLM authentication.

Additionally, this article describes common scenarios that require an SPN to enable Kerberos authentication. The last section describes situations in which Kerberos authentication may fail. The last section also provides troubleshooting steps.

INTRODUCTION

Important The topics that are discussed in this article apply to Internet Information Services (IIS) 6.0. The topics also apply to IIS 7.0 and 7.5 if Kernel-Mode Authentication is disabled by setting the configuration setting **useKernelMode** to *false*.

This step-by-step article describes how to use service principal names (SPNs) when you configure Web applications that are hosted on IIS.IIS passes the Negotiate security header when Windows Integrated authentication is used to authenticate client requests. The Negotiate security header lets clients select between Kerberos authentication and NTLM authentication. The Negotiate process selects Kerberos authentication unless one of the following conditions is true:

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

To enable the Negotiate process to select the Kerberos protocol for network authentication, the client application must provide an SPN, a user principal name (UPN), or a NetBIOS account name as the target name. If the client application does not provide a target name, the Negotiate process cannot use the Kerberos protocol. If the Negotiate process cannot use the Kerberos protocol, the Negotiate process selects the NTLM protocol.

More Information

Concepts

Definition of an SPN

An SPN is the name by which a client uniquely identifies an instance of a service. If you install multiple instances of a service on computers throughout a forest, each service instance must have its own SPN. A particular service instance can have multiple SPNs if there are multiple names that clients might use for authentication. For example, an SPN always includes the name of the host computer on which the service instance is running. Therefore, a service instance might register an SPN for each name or alias of its host.

The HTTP service class

The HTTP service class differs from the HTTP protocol. Both the HTTP protocol and the HTTPS protocol use the HTTP service class. The service class is the string that identifies the general class of service. Well-known service class names include "www" for a Web service and "ldap" for a directory service. Generally, the service class name can be any string that is unique to the service class. Be aware that the SPN syntax uses a forward slash character (/) to separate elements. Therefore, the forward slash character (/) cannot appear in a service class name.

The HOST service and the HTTP service class

The HOST service represents the host computer. The Kerberos protocol uses the HOST SPN to access the host computer. The Kerberos protocol uses the long-term key on the host computer to create a service ticket.

The HTTP service class is one of the built-in services that act as an alias to the HOST SPN. The HOST SPN is mapped to the host computer account. Therefore, when you use the default HTTP service class, the Kerberos protocol uses the computer account as the service account to request a service ticket.

Common scenarios

This section describes scenarios that may require an SPN. Additionally, this section demonstrates how to determine which SPN to set for each scenario. The following terms are used in these scenarios:

IIS6server1	The host name of the computer that is running IIS

mydomain	The domain to which the IIS6server1 computer is joined
appPool1	The user account in the mydomain domain that is used for the application pool identity
appPool2	The user account in the mydomain domain that is used for the second application pool identity
www.test.com	The first host header for a Web site
www.test2.com	The second host header for a Web site
www.test3.com	The third host header for a Web site
www.myIIScluster.com	The fully qualified domain name of a cluster of computers that are running IIS
www.myEXCHcluster.com	The fully qualified domain name of a cluster of computers that are running Microsoft Exchange on IIS

The Setspn.exe tool

The Setspn.exe tool enables you to read, modify and delete the SPN directory property for an Active Directory service account. SPNs are used to locate a target principal name for running a service. The SetSpn.exe tool also enables you to view the current SPNs, reset the account's default SPNs, and add or delete supplemental SPNs.

To obtain the Setspn.exe tool for Microsoft Windows Server 2003, click the following article number to view the article in the Microsoft Knowledge Base:
970536 Setspn.exe support tool update for Windows Server 2003

Scenario 1: Access an IIS application when the application pool identity has been modified

When an IIS application runs under a domain user account instead of under the default network service account, you must set the SPN for the HTTP service under the domain account. In this scenario, you access the IIS application by using either the NetBIOS name of the server that is running IIS or the FQDN of the server that is running IIS.

To access the IIS application by using the NetBIOS name, use the following command,

where **NETBIOS_NAME_OF_IIS_SERVER** is the NetBIOS name of the server that is running IIS:
Setspn –S HTTP/NETBIOS_NAME_OF_IIS_SERVER domain\username
For example, the command may resemble the following command:
Setspn –S HTTP/iis6server1 mydomain\appPool1
To access the IIS application by using the FQDN, use the following command,
where **FQDN_OF_IIS_SERVER** is the FQDN of the server that is running IIS:
Setspn –S HTTP/FQDN_OF_IIS_SERVER domain\username
For example, the command may resemble the following command:
Setspn –S HTTP/iis6server1.mydomain.com mydomain\appPool1

Scenario 2: Access a Web application by using a host header

When you access a Web application by using a host header, you must set an SPN for the HTTP service.

When you run the Web application under a default account such as the network service account, the local service account, or the local system account, you can use the following command:
Setspn –A HTTP/HOST_HEADER NETBIOS_NAME_OF_IIS_SERVER
In this command, **HOST_HEADER** is the host header that you type in a browser window to access the application, and **NETBIOS_NAME_OF_IIS_SERVER** is the NetBIOS name of the server that is running IIS.

For example, the command for the application may resemble the following command:
Setspn –A HTTP/www.test.com iis6server1
If you run the application under a domain account, you can use the following command:
Setspn –A HTTP/HOSTHEADER_OR_DNS_ALIAS domain\username
In this command, **HOSTHEADER_OR_DNS_ALIAS** is the host header or DNS alias that you use to access the Web application.

For example, the command for the application may resemble one of the following commands:

- **Setspn –A HTTP/www.test.com mydomain\appPool1**
- **Setspn –A HTTP/www.test3.com mydomain\appPool2**

Scenario 3: Access an IIS application in a clustered or load-balanced environment

When you run IIS in a clustered environment or in a load-balanced environment, you access applications by using the cluster name instead of by using a node name. This scenario includes network load balancing. In cluster technology, a node refers to one computer that is a member of the cluster. To use Kerberos as the authentication protocol in this scenario, the application pool

identity on each IIS node must be configured to use the same domain user account. To configure each IIS node to use the same domain user account, use the following command:
Setspn –A HTTP/**CLUSTER_NAME** domain\username
For example, the command may resemble one of the following commands:

- **Setspn –A HTTP/www.myIISCluster.com mydomain\appPool1**
- **Setspn –A HTTP/www.myEXCHCluster.com mydomain\appPool2**

Scenario 4: Use SQL Server to access an application

The SPNs on the back-end server may have to be verified if the IIS Web application requires access to a back-end computer that is running SQL Server. The Web application may not work correctly if the required SPNs are configured incorrectly.

If the back-end computer that is running SQL Server is running under the local system account, the MSSQLSvc/**FQDN_OF_SQL_SERVER:port** port number must be available for the computer name. Use the following command to determine the SPNs for the computer that is running SQL Server:
Setspn –L SQL_SERVER_COMPUTER_NAME
Use the following command to set the SPN for the name of the computer that is running SQL Server:
Setspn –A MSSQLSvc/FQDN_OF_SQLSERVER:port SQLSERVER_COMPUTER_NAME
If the back-end computer that is running SQL Server is running under a domain account, the MSSQLSvc/**FQDN_OF_SQL_SERVER:port** port number must be available for the domain account. Use the following command to determine the SPNs for the domain account:
Setspn –L domain\username
Use one of the following commands to set the SPN for the name of the computer that is running SQL Server:

- **Setspn –A MSSQLSvc/FQDN_OF_SQLSERVER:portFQDN_OF_SQLSERVER**
- **Setspn –A MSSQLSvc/FQDN_OF_SQLSERVER:portdomain\username**

The SPN helper script

You can use the following sample script to find the SPNs for an IIS application. You can also use this script to find duplicate SPNs. To use this script, follow these steps:

1. Click **Start**, click **Run**, type Notepad, and then click **OK**.

2. In the Notepad file, paste the following script:

```
Dim argSPN, argUser, argComputer, spnToSearch, objCategory, strFilter, searchCategory, domainInput

Function Help()
Dim strMessage
strMessage = strMessage & "Usage:" & chr(13)
strMessage = strMessage & "For accurate results run this script from the IIS server or a member server in
the same domain as IIS server." & chr(13)
strMessage = strMessage & "Check the article's failure scenarios and make sure no duplicate SPNs exist.
" & chr(13)
strMessage = strMessage & "cscript spnHelper.vbs /f:spn /spn:HTTP/www.test.com /user:mydomain\appp
ool1" & chr(13)
strMessage = strMessage & "cscript spnHelper.vbs /f:spn /spn:HTTP/www.test.com /computer:iis6server1
" & chr(13)
strMessage = strMessage & "cscript spnHelper.vbs /f:user /user:mydomain\apppool1" & chr(13)
strMessage = strMessage & "cscript spnHelper.vbs /f:computer /computer:iis6server1" & chr(13)
strMessage = strMessage & "cscript spnHelper.vbs /f:duplicatespn /spn:HTTP/www.test.com" & chr(13)
strMessage = strMessage & "cscript spnHelper.vbs /f:requiredspn" & chr(13)
MsgBox strMessage,,"SPN Helper"
WScript.Quit
End Function

Function setArguments()
argSPN = lcase(WScript.Arguments.Named("spn"))
argUser =  lcase(WScript.Arguments.Named("user"))
argComputer = lcase(WScript.Arguments.Named("computer"))
searchCategory = lcase(WScript.Arguments.Named("f"))
if instr(argUser,"\")>0 then
domainInput = ",DC=" & split(argUser,"\")(0)
argUser = split(argUser,"\")(1)
end if
End Function

Function resetValues()
spnToSearch = ""
objCategory = ""
strFilter = ""
End Function

Function getGCPath()
Dim tempGCPath, objGC, tempGC, tempStr
Set objGC = GetObject("GC:")
for each tempGC in objGC
tempGCPath = tempGC.ADsPath
next
if tempGCPath <> "" then
getGCPath = tempGCPath
```

```
else
WScript.Echo "Unable to find active directory"
WScript.Quit
end if
For tempCounter=0 to UBound(split(lcase(split(getGCPath,"//")(1)),"."))
If tempCounter = UBound(split(lcase(split(getGCPath,"//")(1)),".")) Then tempSeperator="" else tempSepe
rator = ","
tempStr = tempStr & "DC=" & split(lcase(split(getGCPath,"//")(1)),".")(tempCounter) & tempSeperator
Next
getGCPath = tempGCPath & "/" & tempStr
End Function

Function getSPNClass()
Dim tempSPNClass
If trim(argSPN)="" Then getSPNClass = "*": Exit Function
If instr(argSPN,"/")=0 Then getSPNClass = "*": Exit Function
If instr(split(argSPN,"/")(0),"*")>0 Then getSPNClass = "*": Exit Function
getSPNClass = split(argSPN,"/")(0)
End Function

Function isSPNInputValid(spnIN)
isSPNInputValid = ""
If instr(spnIN,"/")=0 Then Exit Function
If instr(spnIN,"*")>0 Then Exit Function
isSPNInputValid = spnIN
End Function

Function Main()
Dim paramSPN
paramSPN = ""
call  resetValues()
call setArguments()
Select Case searchCategory
Case "spn"
if (argUser = "" and argComputer = "") or (argUser <> "" and argComputer <> "") then WScript.Echo "You
must use /spn along with /computer or /user": WScript.Quit
if argSPN = "" then argSPN = "*"
spnToSearch = "(servicePrincipalName=" & argSPN & ")"
if argUser <> "" then objCategory = "(objectCategory=person)(sAMAccountName=" & argUser & ")"
if argComputer <> "" then
objCategory = "(objectCategory=computer)(cn=" & argComputer & ")"
End If
strFilter = "(&" & spnToSearch & objCategory & ")"
Case "duplicatespn"
If isSPNInputValid(argSPN)="" Then WScript.Echo "Invalid SPN input. Please verify and try again.": WScr
ipt.Quit
spnToSearch = "(servicePrincipalName=" & argSPN & ")"
strFilter = spnToSearch
```

```
paramSPN = argSPN
Case "requiredspn"
call showRequiredSPNs("IIS")
WScript.Quit
Case "computer"
objCategory = "(&(objectCategory=computer)(cn=" & argComputer & "))"
strFilter = objCategory
Case "user"
objCategory = "(&(objectCategory=person)(sAMAccountName=" & argUser & "))"
strFilter = objCategory
Case else
call Help()
WScript.Quit
End Select
call getSPNs(paramSPN)
End Function

Function getPingResult(hostName,errorMessage)
'On Error Resume Next
getPingResult = ""
If instr(hostName,".")=0 Then
Dim tempGCPath, objGC, tempGC
Set objGC = GetObject("GC:")
for each tempGC in objGC
tempGCPath = tempGC.ADsPath
next
if tempGCPath <> "" then
gcPath = tempGCPath
else
WScript.Echo "Unable to find active directory"
WScript.Quit
end if
Set adConn = CreateObject("ADODB.Connection")
Set adCmd = CReateObject("ADODB.Command")
adConn.Provider = "ADsDSOObject"
adConn.Open "ADs Provider"
Set adCmd.ActiveConnection = adConn
adQuery = "<" + gcPath + ">;" & "(&(objectCategory=computer)(cn=" & hostName & "))" & ";dnsHostNam
e;subtree"
'WScript.Echo adQuery
'WScript.Quit
adCmd.CommandText = adQuery
Set adRecordSet = adCmd.Execute
if adRecordSet.RecordCount>0 Then
If IsNull(adRecordSet.Fields("dnsHostName"))=0 Then
getPingResult = adRecordSet.Fields("dnsHostName")
hostName = getPingResult
Else
```

```
getPingResult = hostName
End If
else
errorMessage = "Could not find " & hostname & " in the active directory"
end if

Exit Function
End If
getPingResult = hostName
Exit Function
'If Err Then getPingResult = hostName
End Function

Function getSPNs(spn)
Dim spnClass, duplicateSPNArray
spnClass = getSPNClass()
duplicateSPNArray = ""
gcPath = getGCPath()
Set adConn = CreateObject("ADODB.Connection")
Set adCmd = CreateObject("ADODB.Command")
adConn.Provider = "ADsDSOObject"
adConn.Open "ADs Provider"
Set adCmd.ActiveConnection = adConn
adQuery = "<" + gcPath + domainInput + ">;" & strFilter & ";distinguishedName,objectCategory,dnsHostN
ame,servicePrincipalName,sAMAccountName;subtree"
'WScript.Echo adQuery
'WScript.Quit
adCmd.CommandText = adQuery
Set adRecordSet = adCmd.Execute
if adRecordSet.EOF and adRecordSet.Bof Then
WScript.echo "No " & searchCategory & " found with the given criteria."
else
If adRecordSet.RecordCount>10 Then
If msgbox(adRecordSet.RecordCount & " Records are returned with the given criteria. Printing all of them
might take a long time" & chr(13) & " Do you want to print all of them?",vbYesNo,"Kerberos")=vbNo Then
Exit Function
End If
Do While not adRecordset.Eof
If Err Then Exit Do
WScript.echo "Class: " & split(split(adRecordSet.Fields("objectCategory"),",")(0),"=")(1)
WScript.Echo adRecordSet.Fields("distinguishedName")
if UCase(adRecordSet.Fields("objectCategory")) = "COMPUTER" Then
WScript.echo "Computer Name" & adRecordSet.Fields("dnsHostName")
else
WScript.echo "User Name: " & adRecordSet.Fields("samAccountName")
end if
if instr(searchCategory,"spn")>0 Then
spnCollection = adRecordSet.Fields("servicePrincipalName")
```

```
for each individualSPN in spnCollection
if spnClass="*" Then
WScript.Echo Chr(9) + individualSPN
else
Select Case searchCategory
Case "spn"
if Lcase(split(individualSPN,"/")(0)) = lcase(spnClass) Then
WScript.Echo Chr(9) + individualSPN
end if
Case "duplicatespn"
if Lcase(individualSPN) = lcase(spn) Then
duplicateSPNArray = duplicateSPNArray & Lcase(individualSPN) & " for " & split(split(adRecordSet.Fields
("objectCategory"),",")(0),"=")(1) & ":" & adRecordSet.Fields("samAccountName") & Chr(29)
end if
Case "requiredspn"
End Select
End if
next
end if
WScript.Echo
adRecordSet.MoveNext
Loop
If searchCategory = "duplicatespn" Then
If UBound(Split(duplicateSPNArray,Chr(29)))>1 Then
WScript.Echo "Duplicate SPNs found"
For tempDuplicateCount=0 to UBound(Split(duplicateSPNArray,Chr(29)))-1
WScript.Echo Split(duplicateSPNArray,Chr(29))(tempDuplicateCount)
Next
End If
End If
WScript.Echo ""
If adRecordset.RecordCount>1 Then WScript.Echo "Found " & adRecordset.RecordCount & " accounts"
Else WScript.Echo "Found " & adRecordset.RecordCount & " account"
end if
adRecordset.Close
adConn.Close
If Err Then MsgBox Err.Message
End Function

Function getCategoryCount(myFilterValue, myFilterCategory)
'This function accepts 2 parameters. First paramenter is the filter value and second param is filter categor
y.
'If you want to pass in your own filter string with various categories, you can pass "" as the second param.
gcPath = getGCPath()
searchCategory = myFilterCategory
Select Case lcase(searchCategory)
Case "spn"
tempFilter = "(servicePrincipalName=" & myFilterValue & ")"
```

```
Case "user"
tempFilter = "(&(objectCategory=person)(sAMAccountName=" & myFilterValue & "))"
Case "computer"
tempFilter = "(&(objectCategory=computer)(cn=" & myFilterValue & "))"
Case else
tempFilter = myFilterValue
End Select
Dim tempCategoryCount
tempCategoryCount = 0
Set adConn = CreateObject("ADODB.Connection")
Set adCmd = CReateObject("ADODB.Command")
adConn.Provider = "ADsDSOObject"
adConn.Open "ADs Provider"
Set adCmd.ActiveConnection = adConn
adQuery = "<" + gcPath + domainInput + ">;" & tempFilter & ";objectCategory,dnsHostName,servicePrinci
palName,sAMAccountName;subtree"
'WScript.Echo adQuery
'WScript.Quit
adCmd.CommandText = adQuery
Set adRecordSet = adCmd.Execute
if adRecordSet.EOF and adRecordSet.Bof Then
else
Do While not adRecordset.Eof
If Err Then Exit Do
if searchCategory = "spn" Then
spnCollection = adRecordSet.Fields("servicePrincipalName")
for each individualSPN in spnCollection
If lcase(individualSPN) = lcase(myFilterValue) Then
tempCategoryCount  = tempCategoryCount  + 1
End If
next
else
tempCategoryCount = tempCategoryCount + 1
end if
adRecordSet.MoveNext
Loop
end if
getCategoryCount = tempCategoryCount
adRecordset.Close
adConn.Close
End Function

Function showRequiredSPNs(Product)
Select Case Product
Case "IIS"
If MsgBox("Is IIS running in a Cluster or NLB",vbYesNo)=vbYes Then 'Running in Cluster or NLB is true
strClusterName = InputBox("Enter the Cluster Name")
If strClusterName = "" Then WScript.Quit
```

```
If getPingResult(strClusterName,errorMessage)="" Then
If MsgBox(errorMessage & ". Do you want to continue?",vbYesNo)<>vbYes Then WScript.Quit
End If
strDomainAccount = InputBox("Enter the Domain Account that the application pool is running under")
If strDomainAccount = "" Then WScript.Quit
strRequiredSPN = "HTTP/" & strClusterName
If instr(strDomainAccount,"\") > 0 then
If getCategoryCount(split(strDomainAccount,"\")(1), "user")=0 Then
WScript.Echo "Domain account " & strDomainAccount & " does not exist"
WScript.Quit
End If
Else
If getCategoryCount(strDomainAccount, "user")=0 Then
WScript.Echo "Domain account " & strDomainAccount & " does not exist"
WScript.Quit
End If
End If
If getCategoryCount(strRequiredSPN, "spn")>0 Then
WScript.Echo "SPN " & " is already set. Use search option for finding the account that it is set for"
WScript.Quit
End If
WScript.Echo "You need to set the SPN " & strRequiredSPN & " for domain account " & strDomainAccou
nt
Else
If MsgBox("Is IIS application pool running under domain account",vbYesNo)=vbYes Then
strHostName = InputBox("Enter the hostname or host header or FQDN that you use to access the applica
tion")
If strHostName = "" Then WScript.Quit
If getPingResult(strHostName,errorMessage)="" Then
If MsgBox(errorMessage & ". Do you want to continue?",vbYesNo)<>vbYes Then WScript.Quit
End If
strDomainAccount = InputBox("Enter the Domain Account that the application pool is running under")
If strDomainAccount = "" Then WScript.Quit
If instr(strDomainAccount,"\") > 0 then
If getCategoryCount(split(strDomainAccount,"\")(1), "user")=0 Then
WScript.Echo "Domain account " & strDomainAccount & " does not exist"
WScript.Quit
End If
Else
If getCategoryCount(strDomainAccount, "user")=0 Then
WScript.Echo "Domain account " & strDomainAccount & " does not exist"
WScript.Quit
End If
End If
strRequiredSPN = "HTTP/" & strHostName
If getCategoryCount(strRequiredSPN, "spn")>0 Then
WScript.Echo "SPN " & strSPNRequired & " is already set. Use search option for finding the account that i
t is set for"
```

```
WScript.Quit
Else
WScript.Echo "You need to set SPN " & strRequiredSPN & " for domain account " & strDomainAccount
WScript.Quit
End If
Else
strHostName = InputBox("Enter the host header or FQDN that you use to access the application")
If strHostName = "" Then WScript.Quit
If getPingResult(strHostName,errorMessage)="" Then
If MsgBox(errorMessage & ". Do you want to continue?",vbYesNo)<>vbYes Then WScript.Quit
End If
If MsgBox("Are you accessing the application with netbios name or FQDN or CNAME alias of IIS server?"
,vbYesNo)=vbYes Then
strRequiredSPN = "host/" & strHostName
If getCategoryCount(strRequiredSPN, "spn")>0 Then
WScript.Echo "Required SPN " & strRequiredSPN & " is already set. Use search option for finding the acc
ount that it is set for"
WScript.Quit
Else
WScript.Echo "You need to set SPN " & strRequiredSPN & " for IIS server's netbios name"
WScript.Quit
End If
End If
strHostHeader = InputBox("Enter the host header that you use to access the application")
If strHostHeader = "" Then WScript.Quit
strRequiredSPN = "http/" & strHostHeader
If getCategoryCount(strSPNRequired, "spn")>0 Then
WScript.Echo "A required SPN " & strSPNRequired & " is already set. Use search option to find the accou
nt the SPN is set to. If the required SPN is found under a different account, remove and add it to the IIS s
erver's machine account."
WScript.Quit
Else
WScript.Echo "You need to set SPN " & strRequiredSPN & " for IIS server's netbios name"
WScript.Quit
End If
End If

End If
Case Else
call Help()
End Select
End Function

call Main()
'
```

3. Save the file as Spnhelper.vbs.

4. At a command prompt, run the Spnhelper.vbs file by using the appropriate command-line option.

   Note To view the command-line options for the Spnhelper.vbs file, type Spnhelper.vbs/help at a command prompt, and then press ENTER.

   Additional considerations

   Kerberos authentication may fail when the required SPNs are set for the computer accounts or for the domain accounts. If Kerberos authentication fails unexpectedly, do the following:

- Verify that no duplicate SPNs exist in the global catalog for an SPN.

  For example, if the HTTP/www.test.com SPN is set for the myDomain\appPool1 account and for the myDomain\appPool2 account, a duplicate SPN exists even though the SPNs are set for different accounts. Additionally, if the HTTP/iis6server1.mydomain.com SPN is set for the myDomain\appPool1 account, and the HTTP/iis6server1.mydomain.com SPN is also associated with the computer account for the server, you also have a duplicate SPN. You can have one HOST SPN and one HTTP SPN. The explicit HTTP SPN will override the implicit one that is covered under the HOST entry. However, if the URL that the user types is associated with more than one user account or with more than one computer account, you have a duplicate SPN.

- Verify that the SPNs have replicated to other domain controllers.

  Replication issues between the domain controllers can prevent the SPNs from replicating to the other domain controllers. When the SPNs do not replicate to the other domain controllers, the application may not work from some client computers. For example, if the HTTP/www.test.com SPN is set for the myDomain\appPool1 account on a domain controller, the HTTP/www.test.com SPN may not be found for the myDomain\appPool1 account on a second domain controller if the SPN has not been replicated to the second domain controller.

  Note By default, replication takes 15 minutes.
- Verify that the Web server is configured to support Kerberos authentication. To do this, verify that the setting for the NTAuthenticationProviders key in the IIS metabase has not been changed to **NTLM**. The default setting is **Negotiate,NTLM**.

- Verify that the server or service that is delegating the credentials is trusted for delegation.

- Verify that the account that the service is running under is trusted for delegation.

- In Active Directory, verify that the **Account is sensitive and cannot be delegated** check box is cleared for users who access the application.

- If you are accessing the application directly from the server, verify that the **Loopback Security Check** check box is cleared.For more information about how to verify that the loopback check is disabled, click the following article number to view the article in the Microsoft Knowledge Base:

  896861 You receive error 401.1 when you browse a Web site that uses Integrated Authentication and is hosted on IIS 5.1 or IIS 6

- Verify that the client is a Kerberos-enabled client. Also, verify that the **Enable Windows Integrated Authentication** setting is enabled in Internet Explorer. To do this, click **Internet Options** on the **Tools** menu, click the **Advanced** tab, and then make sure that the **Enable Windows Integrated Authentication** check box is selected.

- Verify that all computers that are part of the Kerberos process have consistent name resolution and are connected by Kerberos trust. For example, verify that the computers that are involved in the Kerberos process are in the same forest or are part of a cross-forest Kerberos trust.

- Verify that the token size does not exceed the value that is set for the **MaxTokenSize**property. Also, verify that the request size does not exceed the value that is set for the **MaxFieldLength** property. These values may be exceeded if users who are part of the Kerberos process are members of many groups.

  920862 Error message when an Outlook Web Access user tries to access a mailbox in Exchange Server 2003: “HTTP 400 Bad Request (Request header too long)”

- Verify that the required security policies are enabled when you configure domain accounts for application pools. To do this, verify that the domain account is a member of the following local security policies on the computer that is running IIS:

- **Adjust memory quotas for a process**

- **Logon as a service**

- **Replace a process level token**

- Verify that Kerberos authentication is working correctly over UDP. By default, Kerberos authentication uses UDP. However, the loss of UDP packets can cause Kerberos authentication to fail. When this issue occurs, you can force Kerberos authentication to use TCP. For more

information about how to force Kerberos authentication to use TCP, click the following article number to view the article in the Microsoft Knowledge Base:

244474 How to force Kerberos to use TCP instead of UDP in Windows Server 2003, in Windows XP, and in Windows 2000

- Verify that the time stamp on the authenticator does not differ by more than five minutes from the time stamp of the server. For more information about how to resolve timestamp differences, click the following article number to view the article in the Microsoft Knowledge Base:

  232386 Cannot log on if time and date are not synchronized

- Verify that TCP/UDP port 88 is not blocked by a firewall or a router. By default, Kerberos authentication uses TCP/UDP port 88.

- Verify that the domain mode environment is at least a Microsoft Windows 2000 native mode environment. For more information about how to determine the domain mode, visit the following Microsoft Web site:

  http://technet2.microsoft.com/windowsserver/en/library/11b2d3d3-980c-4b64-9ed3-51778f1fe5771033.mspx?mfr=true

- On the client, verify that the URL for the Web application is added to the local intranet sites. For more information about how to add a URL to the local intranet, click the following article number to view the article in the Microsoft Knowledge Base:

  303650 Intranet site is identified as an Internet site when you use an FQDN or an IP address

- Verify that each gigabit ethernet device is using the latest driver version. For more information about issues that can occur when the drivers for gigabit Ethernet devices are outdated, click the following article number to view the article in the Microsoft Knowledge Base:

  326152 Cannot connect to domain controller and cannot apply Group Policy with Gigabit Ethernet devices

For more information about how to use Kerberos authentication with load-balanced Web sites, visit the following Microsoft Web site:
http://technet.microsoft.com/en-us/library/cc757299.aspx
For more information about problems that can occur when you use Kerberos authentication with proxy servers, click the following article number to view the article in the Microsoft Knowledge Base:

321728 Internet Explorer does not support Kerberos authentication with proxy servers

For more information about how to configure IIS to support both the Kerberos protocol and the NTLM protocol for network authentication, click the following article number to view the article in the Microsoft Knowledge Base:

215383 How to configure IIS to support both the Kerberos protocol and the NTLM protocol for network authentication

For more information about how to configure a Windows SharePoint Services virtual server to use Kerberos authentication and how to switch from Kerberos authentication back to NTLM authentication, click the following article number to view the article in the Microsoft Knowledge Base:

832769 How to configure a Windows SharePoint Services virtual server to use Kerberos authentication and how to switch from Kerberos authentication back to NTLM authentication

For more information about problems that can occur when you use Kerberos authentication protocol to connect to a Web server that uses a non-standard port on Windows XP or Windows Server 2003, click the following article number to view the article in the Microsoft Knowledge Base:

908209 Internet Explorer 6 cannot use the Kerberos authentication protocol to connect to a Web server that uses a non-standard port on Windows XP or Windows Server 2003