Troubleshooting Kerberos Errors

Microsoft Corporation

Published: March 2004

Abstract

This white paper can help you troubleshoot Kerberos authentication problems that might occur in a Microsoft® Windows Server™ 2003 operating system environment. It outlines some simple troubleshooting basics and explains the causes of common Kerberos errors. It also summarizes common tools used to troubleshoot problems with Kerberos authentication.

To troubleshoot Kerberos authentication, you need to understand how Kerberos authentication interacts with its supporting technologies (such as Active Directory® directory service and time servers) as well as how the Kerberos authentication process works. With that understanding, you can use specific diagnostic tools to find answers to specific questions, and to identify and resolve problems.

This white paper does not provide detailed information about Kerberos authentication or its supporting technologies, but does provide references to that information.

Contents

Introduction

The Kerberos V5 protocol assumes that transactions between clients and servers take place on an open network, in which packets transmitted along the network can be monitored and modified at will. The assumed environment, in other words, is very much like today's Internet, where an attacker can easily pose as either a client or a server, and can readily eavesdrop on or tamper with communications between legitimate clients and servers.

Microsoft's implementation of the Kerberos V5 protocol is the default authentication package for Windows Server 2003. The Kerberos V5 protocol became the default authentication package with Windows 2000. Windows Server 2003 still supports NTLM for non-Kerberos clients such as the Windows NT® Server 4.0 operating system.

Infrastructure Requirements

Problems with Kerberos authentication often involve technologies on which the Kerberos SSP depends, or stem from easy-to-correct oversights in the configuration of Kerberos settings. This section reviews these dependencies and summarizes how they relate to troubleshooting Kerberos authentication.

Active Directory Domain

Kerberos authentication is not supported in earlier operating systems such as Windows NT. For more information about the Active Directory® directory service, see "Active Directory Collection" on Microsoft TechNet at http://go.microsoft.com/fwlink/?LinkId=25389.

TCP/IP Network Connectivity

For Kerberos authentication to occur, there must be TCP/IP network connectivity between the client and the domain controller(s), and the client and the target server. Issues that can affect network connectivity include:

- **Firewalls.** If you use a firewall, be sure that the Kerberos ticket-granting service ports (TCP port 88, UDP port 88) are enabled on the network.

  TCP and UDP Ports Required for Correct Operation of the Kerberos Protocol

Port	Service	Description
53/TCP 53/UDP	DNS service	The internal DNS server needs to be accessible to all clients for the location of KDC computers. The Active Directory domain controllers need to be able to access external DNS servers for resolving external domain name requests.

88/TCP 88/UDP	Kerberos ticket-granting service	All clients need to be able to connect to this port on the KDC servers.
123/TCP 123/UDP	Time service	All clients need to be able to connect to this port for time synchronization, either to an internal time server or to an external time source. The internal time server will need to connect to an external time source to synchronize.
464/TCP	Microsoft Windows 2000 Kerberos change password protocol	This port is also used by the kpasswd protocol. This port should only be open if clients use the kpasswd protocol.

For more information about ports domain controllers use, see "A List of the Windows Server Domain Controller Default Ports" in the Microsoft Knowledge Base at http://go.microsoft.com/fwlink/?LinkId=22894. For more information about TCP/IP, see "TCP/IP Technical Reference" on Microsoft TechNet at http://go.microsoft.com/fwlink/?LinkId=25392.

- **Cached credentials.** A user who can log on with cached credentials might not be aware of a connectivity issue.

Domain Name System

The client uses the fully qualified domain name (FQDN) to access the domain controller. In order for the client to obtain the FQDN, DNS must be functioning. For best results, do not use host files with DNS. For more information about DNS, see "Deploying DNS" in the Microsoft Windows Server 2003 Deployment Kit at http://go.microsoft.com/fwlink/?LinkId=23041.

Time Service

For Kerberos authentication to function properly, it is vital that the time on all of the computers on a network be synchronized — that is, that all of the domains and forests in a network are using the same time source. An Active Directory domain controller will act as an authoritative source of time for its domain, which guarantees that an entire domain will have the same time. For more information, see "Windows Time Service Technical Reference" on Microsoft TechNet at http://go.microsoft.com/fwlink/?LinkId=25393.

Operating System

Kerberos authentication relies on client functionality that is built in to Windows 2000, Windows Server 2003, and the Microsoft Windows® XP operating system. If a client, domain controller, or target server is running an earlier operating system, it cannot use Kerberos authentication natively.

Troubleshooting Strategies

As with most technologies, the better you understand how Kerberos authentication should work and how to confirm whether it is being used, the better you can isolate a problem and determine a solution.

Kerberos Authentication Basics

Kerberos authentication provides a mechanism for mutual authentication between a client and a server on an open network, and in which packets transmitted along the network can be monitored and modified at will. In order to provide secure authentication, Kerberos authentication uses symmetric keys, encrypted objects, and Kerberos services.

Keys

Kerberos authentication relies on different types of keys:

- **User, service, and system keys.** Long-term symmetric keys generated from passwords.
- **Public keys.** Long-term asymmetric keys used with smart cards.
- **Session keys.** Short-term symmetric keys created by domain controllers.

Tickets

The main component in Kerberos authentication is the ticket. Essentially, the goal of Kerberos messages is to request and deliver tickets. There are two types of tickets used in Kerberos authentication, ticket-granting tickets (TGTs) and session tickets:

- **TGT.** The KDC responds to a client's authentication service request by returning a session ticket for itself. This special session ticket is called a ticket-granting ticket (TGT). A TGT enables the authentication service to safely transport the requestor's credentials to the ticket-granting service.
- **Session ticket.** A session ticket allows the ticket-granting service (TGS) to safely transport the requestor's credentials to the target server or service.

Key Distribution Center

To solve the problem of key distribution, the Kerberos protocol, similar to its namesake in Greek mythology, uses three "heads" — a client, a server, and a trusted third party that mediates between the other two.

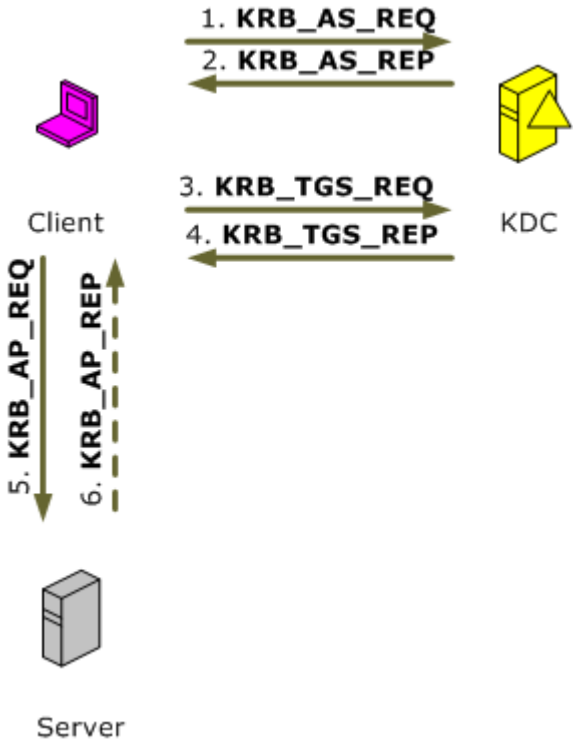
The trusted intermediary in the protocol is the Key Distribution Center (KDC).The KDC is a service that runs on a physically secure server. The KDC maintains a database with account information for all security principals in its realm (the Kerberos equivalent of a Windows Server 2003 domain). Along with other information about each security principal, the KDC stores a cryptographic key known only to the security principal and the KDC. This key is used in exchanges between the security principal and the KDC.

As in other implementations of the Kerberos protocol, Microsoft implements the KDC as a single process that provides two services:

- **Authentication service (AS).** The AS issues TGTs good for admission to the ticket-granting service in its domain. Before network clients can get tickets for services, each client must get an initial TGT from the AS in the user's account domain.
- **Ticket-granting service (TGS).** The TGS issues tickets good for admission to other services in the TGS's domain or to the ticket-granting service of a trusted domain. When a client wants access to a service, it must contact the ticket-granting service in the service's account domain, present a TGT, and ask for a ticket. If the client does not have a TGT valid for admission to that TGS, it must get one through a referral process that begins at the TGS in the user account's domain and ends at the TGS in the service account's domain.

Windows Server 2003 implements the KDC as a domain service. It uses the domain's Active Directory as its account database and gets some information about users from the global catalog.

Kerberos Exchange and Message Summary



Authentication service exchange

Kerberos authentication service request (KRB_AS_REQ)

The client contacts the KDC's authentication service for a short-lived ticket (a message containing the client's identity and — for Windows clients — SIDs) called a ticket-granting ticket (TGT). This happens at logon.

Kerberos authentication service response (KRB_AS_REP) The AS constructs the TGT and creates a session key the client can use to encrypt communication with the ticket-granting service.

The TGT has a limited lifetime, which is 10 hours by default. At the point that the client has received the TGT, the client has not been granted access to any resources, even to resources on the local computer.

Ticket-granting service exchange

Kerberos ticket-granting service request (KRB_TGS_REQ) The client wants access to local and network resources. To gain access, the client sends a request to the TGS for a ticket for the local computer or some network server or service. This ticket is referred to as the session ticket. To get the ticket, the client presents the TGT, an authenticator, and the Service Principal Name (SPN) of the target server.

Kerberos ticket-granting service response (KRB_TGS_REP) The TGS examines the TGT and the authenticator. If these are acceptable, the TGS creates a service ticket. The client's identity is taken from the TGT and copied to the session ticket. Then the ticket is sent to the client.

> **Important**
> The TGS cannot determine if the user will be able to get access to the target server. It simply returns a valid ticket. Authentication does not imply authorization.

Client/server exchange

Kerberos application server request (KRB_AP_REQ) After the client has the session ticket, the client sends the ticket and a new authenticator to the target server, requesting access. The server will decrypt the ticket, validate the authenticator, and for Windows services, create an access token for the user based on the SIDs in the ticket.

Kerberos application server response (KRB_AP_REP) Optionally, the client might request mutual authentication — that is, that the target server verify its own identity. If mutual authentication is requested, the target server will take the client computer's timestamp from the authenticator, encrypt it with the session key the TGS provided for client-target server messages, and send it to the client.

For more detailed information about how Kerberos authentication works, see:

- "Windows 2000 Kerberos Authentication White Paper" on the Microsoft website at http://go.microsoft.com/fwlink/?LinkId=23128.
- Windows Server 2003 Technical Reference on Microsoft TechNet at http://go.microsoft.com/fwlink/?LinkId=21711.

Isolating the Problem

Knowing where a problem exists and whether that problem might indicate a problem with Kerberos authentication is crucial to success.

Any issue related to authentication such as console logon, network logon, access to network resources, or remote access might possibly indicate some sort of Kerberos error because Kerberos is the default authentication protocol.

How do you determine that a problem is occurring with Kerberos authentication? If the system event log shows errors from any services that provide authentication such as Kerberos, KDC, LsaSrv, or Netlogon, there might be Kerberos errors associated, as well. Also failure audits in the security event log might show that the Kerberos protocol was being used when a logon failure occurred.

Where to start

1. Use Kerberos Tray or Kerberos List to confirm that you have a session ticket for the server you are attempting to connect to. If you have a session ticket for the server and you are still getting an error message, consider these two possibilities:
   - You might have an issue with SPNs. For more information about SPN issues, see Need an SPN Set and 0x8 KDC_ERR_PRINCIPAL_NOT_UNIQUE later in this white paper.
   - You might have an authorization issue instead of an authentication issue. If this is the case, most likely Kerberos authentication is not the problem.
2. If you do not have a session ticket, then use Kerberos Tray or Kerberos List to confirm that you have a TGT.
   - If you have a TGT but no session ticket, examine the system event log. Errors logged in the system log will help you determine why you cannot get a ticket to the server.
3. If you are auditing successful logons, then you can check the security event log on the client to see if the system is using NTLM instead of Kerberos authentication. Use of NTLM can occur because:
   - The application uses NTLM. See NTLM Fallback later in this white paper for an example of this condition.
   - Kerberos authentication is failing and Negotiate is using NTLM.
4. If Kerberos authentication is failing, the system event log or captured data in a network trace should contain the Kerberos error code that was returned by the KDC or the Kerberos SSP. You can also debug to get more information.

Common Issues

The following sections detail the most common problems encountered by users in Kerberos authentication environments, explain the possible causes of those problems, and suggest how to resolve those problems.

Time Synchronization (Clock Skew)

One type of attack that Kerberos authentication was designed to prevent is known as a “replay” attack. In a replay attack, a malicious user captures the network traffic and replays it to fool the

authenticating server into accepting the attacker as a legitimate user who is providing credentials. Kerberos authentication prevents a replay attack with two mechanisms:

- The Kerberos client on the local computer encrypts a timestamp inside the authenticator and then sends it to the KDC. If the KDC verifies that the time it decrypts from the authenticator is within a specified amount of the local time on the KDC (the default is 5 minutes), the system can assume that the credentials presented are genuine.
- All tickets issued by the KDC have an expiration time. Thus, if a ticket is compromised, it cannot be used outside of a specified time range — usually short enough to make the risk of a replay attack minimal.

Because of these mechanisms, Kerberos authentication relies on the date and time that are set on the KDC and the client. If there is too great a time difference between the KDC and a client requesting tickets, the KDC cannot determine whether the request is legitimate or a replay. Moreover, if the time difference is so great that the client is far into the future, the client might attempt to compensate for the clock skew, but will receive tickets that have already expired and are useless. If the client requests new tickets, that will not solve the problem because the KDC uses its own clock as a reference instead of the time on the client computer.

Therefore, it is vital that the time on all of the computers on a network be synchronized in order for Kerberos authentication to function properly. This means that all of the domains and forest in a network must use the same time source. An Active Directory domain controller will act as an authoritative time server for its domain, which guarantees that an entire domain will have the same time. However, multiple domains might not have their times synchronized. It is recommended that you use either an external time source or a single time source within the network to synchronize all computers.

Problem

The difference between client timestamp in the authenticator or KRB_AS_REQ and the server is greater than the **Maximum tolerance for computer clock synchronization** setting in the domain policy.

Confirmation

Clock skew can be easily diagnosed by reviewing data in Event Viewer. For more information, see:

- 0x25: KRB_AP_ERR_SKEW: Clock Skew too great later in this white paper.
- Clock Skew network trace in Appendix A.

Resolution

For information about how to use an external time source to synchronize all the computers in a domain, see “How to Configure an Authoritative Time Server in Windows 2000” in the Microsoft Knowledge Base at http://go.microsoft.com/fwlink/?LinkId=23042.

UDP Fragmentation

By default, Kerberos authentication uses User Datagram Protocol (UDP) to transmit its data.

Problem

UDP provides no guarantee that a packet sent along the network will reach its destination intact. Thus, in environments with a high amount of network congestion it is common for packets to get lost or fragmented on the way to their destination.

Confirmation

You can diagnose UDP fragmentation by reviewing Network Monitor captured data. For more information, see the UDP Fragmentation network trace in Appendix A.

Resolution

Because the only way to decrease the likelihood of UDP fragmentation occurring is to reduce network traffic — a usually impractical solution — it is almost always better to configure the Kerberos authentication service to use TCP instead of UDP. TCP provides a guarantee that a packet that is sent will reach its destination intact and can therefore be used in any network environment. In order to force Kerberos authentication to use TCP, see "How to Force Kerberos to Use TCP Instead of UDP" in the Microsoft Knowledge Base at http://go.microsoft.com/fwlink/?LinkId=23043.

Group Membership Overloads PAC

In order to provide the proper authorization information to a user after the user logs on with Kerberos authentication, the KDC will transmit Privilege Attribute Certificate (PAC) data in the TGT. The PAC contains various types of authorization data including groups that the user is a member of, rights the user has, and what policies apply to the user. When the client receives a ticket, the information contained in the PAC is used to generate the user's access token.

Problem

In order to optimize performance, the buffer size for the PAC is pre-allocated. The pre-allocated buffer size is usually adequate to hold all the required authorization data. However, if a user has a very high group membership — from over 70 to over 120, depending on what groups — the size of the PAC might exceed the pre-allocated buffer size. In such a case, the system will generate a memory allocation error, PAC creation will fail, and the Kerberos ticket-granting service will either fail to generate a valid ticket or will generate a ticket with an empty PAC. This sort of error usually manifests itself in the form of a memory allocation error, which gets reported as 0x3C - KRB_ERR_GENERIC: Generic error. This also can result in the failure of clients to apply Group Policy settings.

Confirmation

The Kerberos Token Size tool, described in the Diagnostic Tools section of this white paper, is specifically designed to check for this problem.

Resolution

You can solve this problem in two ways:

- Reduce the number of groups that the user is a member of. Because nested groups are expanded out within the PAC, the actual number of groups that the user is a member of might be greater than you suspect.

- Alternatively, install a hotfix that will enable you to set the maximum size of a Kerberos token via the registry. See “New Resolution for Problems That Occur When Users Belong to Many Groups” in the Microsoft Knowledge Base at http://go.microsoft.com/fwlink/?LinkId=23044.

Need an SPN Set

Service Principal Names (SPNs) are unique identifiers for services running on servers. Each service that will use Kerberos authentication needs to have an SPN set for it so that clients can identify the service on the network. It is registered in Active Directory under a user account as an attribute called **Service-Principal-Name**. The SPN is assigned to the account under which the service the SPN identifies is running. Any service can look up the SPN for another service. When a service wants to authenticate to another service, it uses that service’s SPN to differentiate it from other services running on that computer.

SPNs are crucial to constrained delegation. When you set up a computer for delegation, one of the steps of the process is to list the services on other computers that the computer is allowed to delegate to. This list forms a type of ACL. The services running on the other computers are identified by the SPNs that are issued to those services.

In general, SPNs should be set when you create an account, because setting an SPN on an account is virtually the security equivalent of creating that account. Special accounts that were created for services are useless until an SPN is set on those accounts for whatever service will be running under them.

Problem

If an SPN is not set for a service, then clients will have no way of locating that service. Thus, common results of not setting an SPN are KDC_ERR_C_PRINCIPAL_UNKNOWN or KDC_ERR_S_PRINCIPAL_UNKNOWN errors. These two errors usually indicate that an SPN has not been set correctly. Furthermore, there are many other errors for which the cause might be a missing or incorrectly set SPN. Kerberos authentication is not possible without properly set SPNs.

Resolution

Because multiple services can run simultaneously under the same account, setting an SPN requires four pieces of information that will make the SPN unique:

- The service class. This allows you to differentiate between multiple services running under the same account.
- The account under which the service is running.
- The computer on which the service is running, including any aliases that point to that computer.
- The port on which the service is running.

These four pieces of information uniquely identify any service running on a network and can be used to mutually authenticate to any service.

An SPN itself consists of *ServiceClass*/*Host*:*Port,* where:

- *ServiceClass* is the service class of the SPN.

- *Host* is the name of the computer to which the SPN belongs.
- *Port* is the port on which the service that the SPN is registered to run.

For more information about how to set SPNs using setspn.exe, see Setspn in the Diagnostic Tools section later in this white paper.

Kerberos Logons Failing in a Mixed Windows and UNIX Environment with Windows NT 4.0 Computers

In an environment where there exists a trust between a Kerberos realm and an Active Directory domain, authentication data can come from one of two sources, either a UNIX KDC or the Active Directory domain controller. If the authentication data is coming from a UNIX KDC, then Windows users have account mappings set up for them to map their UNIX user account to a Windows user account. Normally, the password on the Windows account does not matter, because all authentications are done by the UNIX KDC.

Problem

The Windows NT 4.0 operating system does not support Kerberos authentication. Therefore, if there are Windows NT computers on the network running services, any authentications involving these computers will occur using NTLM and these authentications will be conducted by the domain controller. In this scenario, the password on the domain controller must match the password stored on the UNIX KDC. The passwords must match because the fallback to NTLM occurs transparently. If the passwords do not match, the domain controller will return an Access denied error because the user has provided a password that does not match the one stored on their Active Directory account.

Resolution

Reset the password on the account that the user's UNIX principal is mapped to in order to match the password stored on the UNIX KDC.

NTLM Fallback

You might find that the security log recorded an event in which logon occurred using NTLM when it should have occurred using Kerberos authentication.

Problem

There are two situations in which this might happen:

- The first situation is where the system attempts authentication using the Kerberos protocol but it fails. As a result, the system attempts to authenticate using NTLM. Windows Server 2003, Windows XP, and Windows 2000 use an algorithm called Negotiate (SPNEGO) to negotiate which authentication protocol is used. Although the Kerberos protocol is the default, if the default fails, Negotiate will try NTLM.
- The second situation is one in which a call to Negotiate returns NTLM as the only protocol available.

Confirmation

The first situation will result in a failed Kerberos authentication that you can investigate by examining errors in the event log or data packets captured by Network Monitor. Both investigation methods are discussed later in this document.

The second situation is much more difficult to diagnose. There are two common causes of the second situation — when Internet Explorer is being used and the Kerberos protocol is not being attempted:

- **Enable Integrated Windows Authentication (requires restart)** setting is not selected in Internet Explorer 6
- Internet Explorer is accessing a site in the Internet zone instead of the intranet zone.

Resolution

Internet Explorer 6 will, by default, not attempt to use the Kerberos protocol to authenticate to any site. To change this, you must select the **Enable Integrated Windows Authentication (requires restart)** setting. For more information, see "Unable to Negotiate Kerberos Authentication After Upgrading to Internet Explorer 6" in the Microsoft Knowledge Base at http://go.microsoft.com/fwlink/?LinkId=23045.

The second common cause is that Internet Explorer 6 is attempting to access a site located in the Internet zone. Internet zone sites are prevented from using Integrated Windows authentication because these protocols will not typically work through Web proxies, among other reasons. If a site is located in the Internet zone, Internet Explorer 6 will not attempt to use Kerberos authentication, and will automatically try NTLM. In all versions of Internet Explorer, when accessing a website to which you want to use Kerberos authentication, you must verify that the website appears as being in the local intranet zone. An icon in the lower right-hand corner of the Internet Explorer window will indicate what zone a website is in. It will display "Internet" for the Internet zone and "Local Intranet" for the intranet zone. If the website appears as being in the Internet zone, you must manually add the site to the local intranet sites list.

To add an Internet site to the local intranet sites list

1. Click **Tools**, and then click **Internet Options**.
2. Click the **Security** tab, then click **Local Intranet**, then click **Sites**, and then click **Advanced**.
3. In the box under **Add this Web site to the zone:** type the name of the website which you want to authenticate with Kerberos authentication and then click **Add**.
4. Click **OK**.

After you perform the above procedure, if you find that NTLM authentication is still being used, or that Kerberos is not even being attempted in a situation where Kerberos authentication should be used, contact Product Support Services for help in diagnosing the problem.

Diagnostic Tools

Some tools — for example, Event Viewer and Network Monitor — that you use to diagnose Kerberos errors are the same you would use for other network-related or authentication issues. More specific tools — such as Kerberos List, Kerberos Tray, and Kerberos Token Size — can be used for detailed Kerberos-specific information. To get more even more detailed information, you can enable debug output. Information about troubleshooting tools is provided in this section.

Event Viewer

Event Viewer is included in Windows Server 2003, Windows XP, and Windows 2000. The system and security logs will contain Kerberos error codes and other events related to authentication. For more information about using Event Viewer, see "HOW TO: Diagnose System Problems with Event Viewer in Windows Server 2003" in the Microsoft Knowledge Base at http://go.microsoft.com/fwlink/?LinkId=23046.

System event log The first place to look if you are experiencing a problem with Kerberos authentication is the system event log.

Any critical errors that Kerberos authentication encounters will show a Source value of Kerberos, KDC, or LsaSrv. The event itself will contain a Kerberos error code and might contain information about how to fix the problem.

If there are no errors listed in the system log or if the errors that appear are not detailed enough to pinpoint the problem, you can configure more detailed Kerberos event logging.

Windows Server 2003 and Windows 2000 can log detailed Kerberos events in the event logs. You can use the resulting information when you troubleshoot Kerberos authentication errors.

How to enable Kerberos event logging on a specific computer

1. Start Registry Editor.

> **Caution** Incorrectly editing the registry might severely damage your system. Before making changes to the registry, you should back up any valued data on the computer.

2. Add the following registry value:

   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters

   Registry Value: LogLevel

   Value Type: REG_DWORD

   Value Data: 0x1

3. If the Parameters subkey does not exist, create it.

> **Note**
> Remove this registry value when it is no longer needed so that performance is not degraded on the computer. Also, you can remove this registry value to disable Kerberos event logging on a specific computer.

4. Quit Registry Editor, and then restart the computer.

After you have configured the system for detailed Kerberos event logging, reproduce the problem, and then check for any Kerberos-related events in the system event log. A new set of Kerberos error messages should now appear in the system event log with event ID 4. These

errors might give more specific information about the problem. However, there might be errors that are simply recording diagnostic information. You should verify that the error reported within the event pertains to the affected systems, and that it occurred in response to the specific authentication request that was performed.

Security event log

The security event log contains information that can explain whether Kerberos authentication was at fault or if perhaps some other authentication protocol was responsible. The details of a specific logon/logoff event for a user will show what authentication protocol was used.

Although Kerberos authentication is preferred, the system might revert to NTLM if errors or failures occur. This reversion can cause further problems, because the user will not have obtained any Kerberos tickets and might not be able to access Kerberos-aware services or might not have the functionality of single sign-on across the entire network.

How do you know if logging on with NTLM or Kerberos protocol?

All account logons that occur from computers running Windows Server 2003 or Windows 2000 should occur using the Kerberos protocol (or Negotiate, which could imply that Kerberos protocol was used). To catch these events you need to enable auditing of successful Account Logon events for user authentication and Logon events for computer authentication. If the security log shows that NTLM was used and there are authentication-related issues present, you will need to investigate by using some of the tools outlined in this section. The following table lists event IDs and information potentially associated with Kerberos authentication. Only relevant event information will be present in the event log. For example, only failure audits will have Kerberos error codes; smartcard logons will have certificate information.

Security Log Events That Might Contain Kerberos Error Codes

Event ID	Account Logon Event Type	Event Information Potentially Associated with Kerberos Authentication
672	• Success audit (Windows 2000 and Windows Server 2003) • Failure audit (Windows Server 2003)	• Authentication Ticket Request: • User Name • Supplied Realm Name • User ID • Service Name • Service ID • Ticket Options • Result Code: *Kerberos error code* • Ticket Encryption Type • Pre-Authentication Type • Client Address • Certificate Issuer Name • Certificate Serial Number • Certificate Thumbprint

673	• Success audit (Windows 2000 and Windows Server 2003) • Failure audit (Windows Server 2003)	• Service Ticket Request: • User Name • User Domain • Service Name • Service ID • Ticket Options • Ticket Encryption Type • Client Address • Failure Code: *Kerberos Error Code* • Logon GUID • Transited Services
675	• Failure audit	• Pre-authentication Failed: • User Name • User ID • Service Name • Pre-authentication Type • Failure Code: *Kerberos error code* • Client Address
676	• Failure audit (Obsolete in Windows Server 2003; both success and failure audits use	• Authentication Ticket Request Failed: • User Name • Supplied Realm Name • Service Name • Ticket Options • Failure Code: *Kerberos error code* • Client Address

	even t ID 672.)	

677	• Failu re audit (Obs olete in Wind ows Serv er 20 03; both succ ess and failur e audit s use even t ID 673.)	• Service Ticket Request Failed:
		  • User Name
		  • User Domain
		  • Service Name
		  • Ticket Options
		• Failure Code: *Kerberos error code*
		• Client Address

Network Monitor

If the errors in the event logs do not help you solve the problem, or if you need more detailed information, use Network Monitor to capture a network trace for inspection of the actual Kerberos packets being sent across the network.

> **Note**
> For more information about Network Monitor, see “Network Monitor” on Microsoft TechNet at http://go.microsoft.com/fwlink/?LinkId=23049. For best practices and procedures associated with Network Monitor, see “Checklist: Monitoring network traffic on your local computer” on Microsoft TechNet at http://go.microsoft.com/fwlink/?linkid=23047.

The full version of Network Monitor is included with Microsoft Systems Management Server (SMS). A limited version of the tool is included with Windows 2000, Windows XP, and the Windows Server 2003 family. It is also available from Microsoft Product Support Services.

How to install Network Monitor on Windows Server 2003

1. Open the Windows Components Wizard.

2. In the Windows Components Wizard, click **Management and Monitoring Tools**, and then click **Details**.
3. In **Subcomponents of Management and Monitoring Tools**, select the **Network Monitor Tools** check box, and then click **OK**.
4. If you are prompted for additional files, insert the installation CD for your operating system, or type a path to the location of the files on the network.

Note

- To perform this procedure, you must be a member of the Administrators group on the local computer, or you must have been delegated the appropriate authority. If the computer is joined to a domain, members of the Domain Admins group might be able to perform this procedure.
- To open the Windows Components Wizard, click **Start**, click **Control Panel**, click **Add or Remove Programs**, and then click **Add/Remove Windows Components**.
- Certain Windows components require configuration before they can be used. If you installed one or more of these components but did not configure them, when you click **Add/Remove Windows Components**, a list of components that need to be configured is displayed. To start the Windows Components Wizard, click **Components**.
- This procedure automatically installs the Network Monitor driver.

How to install Network Monitor on Windows XP

Network Monitor is included with the Windows XP Support Tools.

1. Insert the Windows XP CD-ROM in the drive.
2. Double-click **My Computer**, right-click the CD-ROM drive, and then click **Explore**.
3. Go to **Support\Tools**, and then double-click **Setup.exe**.
4. When the Windows Support Wizard starts, click **Next**.
5. Click **I agree** on the End User License Agreement.
6. Type your name and organization and then click **Next**.
7. Click either the **Typical** or **Complete** installation type, and then click **Next**.
8. Verify the installation location, and then click **Install**.

How to install Network Monitor on Windows 2000

1. Click **Start**, point to **Settings**, and then click **Control Panel**.

2. Double-click **Add/Remove Programs**.
3. Click **Add/Remove Windows Components**.
4. Click **Management and Monitoring Tools**, and then click **Details**.
5. Click to select the Network Monitor Tools check box, and then click **OK**.
6. Click **Next**.

> **Important**
> Network monitoring on Windows XP is done with the Netcap.exe tool. This tool only allows the capture of network traffic. The capture cannot be viewed with the same tool. You must use the full version of Network Monitor on Windows 2000 or the Windows Server 2003 family to view the captured data.

How to capture network traffic with Windows XP

If you are using the version of Netcap.exe provided in the Windows XP Support Tools, use the following procedure:

1. Click **Start**, click **All Programs**, click **Accessories**, and then click **Command Prompt**. This will open a command prompt window.
2. Type **Netcap.exe /c:***path* and press ENTER, where *path* is the full path to the directory and file name where you want to store this network trace.
3. After the error has been reproduced, type **Netcap.exe /remove** and press ENTER. This will stop the network capture.

The procedure for capturing network traffic with Windows 2000 and Windows Server 2003 is different than the Windows XP procedure. Use the following procedure on these operating systems:

How to capture network traffic with Windows 2000 and the Windows Server 2003 family

1. Click **Start**, click **Control Panel**, click **Performance and Maintenance**, click **Administrative Tools**, and then double-click **Network Monitor**.
2. Click the **Start** button to begin capturing network traffic.
3. Reproduce the error.
4. Click the **Stop** button to stop capturing network traffic.
5. In the capture statistics information on the right-hand side, verify that no packets were lost because of the buffer overflowing. If any packets were lost, increase the buffer size in the **Buffer Settings** dialog box on the **Capture** menu and perform the capture again.

For more information, see “To capture network frames” on Microsoft TechNet at http://go.microsoft.com/fwlink/?LinkId=23052.

How to filter Kerberos-specific network traffic

You can filter out packets from all protocols except the Kerberos protocol. To apply a filter to only show Kerberos protocol-related network traffic, perform the following steps in Network Monitor:

1. Click **Capture**, and then click **Display captured data**.
2. Click the Funnel button and then double-click **Protocol == ANY**.
3. Click **Disable all**.
4. Select **Kerberos** from the list and click **Enable**. Click **OK**, and then click **OK** again.

After you perform this procedure, the only packets that should appear are Kerberos packets.

If no packets appear after applying the filter

Three possible causes are:

- The Kerberos tickets have either already been issued or have been cached. You can use Kerberos List to show all the tickets currently issued on the computer. You can also use Kerberos List to purge all tickets. Kerberos List is described in further detail later in this section.
- The Kerberos authentication protocol is not even being attempted. The associated logon event in the security event log should say which protocol was used to authenticate the user. If NTLM is listed, then Kerberos was not even attempted. For more information about NTLM fallback, see NTLM Fallback earlier in this white paper. If NTLM fallback is occurring at logon or when requesting a network resource, the event logs (described in the previous section) might contain useful information.
- The Network Monitor buffer overflowed. On high-traffic networks, this can be easily happen if the tool is configured with the default buffer size.

Analyzing the captured Kerberos traffic

After you have captured some Kerberos packets, the problem can be diagnosed by determining how the captured data differs from a successful authentication. In most cases, the diagnosis will involve following the packet exchange and looking for a KRB_ERROR packet somewhere in the captured data. However, in some cases, especially if everything appears normal, more in-depth analysis is required. Several examples of captured network data — demonstrating a successful logon and showing common failures — are provided in Appendix A: Network Monitor Sample Traces. The captured data is annotated to help explain each frame's overall impact on the success or failure of the authentication attempt.

For more information, see:

- "How to View HTTP Data Frames Using Network Monitor" in the Microsoft Knowledge Base at http://go.microsoft.com/fwlink/?LinkId=23055.
- "Frequently Asked Questions About Network Monitor" in the Microsoft Knowledge Base at http://go.microsoft.com/fwlink/?LinkId=23056.

Klist.exe: Kerberos List

Kerberos List is a command-line tool that is used to view and delete Kerberos tickets granted to the current logon session. To use Kerberos List to view tickets, you must run the tool on a computer that is a member of a Kerberos realm.

When Kerberos List is run from a client, it shows the:

- Ticket-granting ticket (TGT) to a Kerberos Key Distribution Center (KDC) in Windows.
- Ticket-granting ticket (TGT) to Ksserver on UNIX.

How to install Kerberos List

Kerberos List is supported for Windows Server 2003, Windows XP, and Windows 2000.

You can download Klist.exe from "Windows Server 2003 Resource Kit Tools" on the Microsoft Download Center at http://go.microsoft.com/fwlink/?LinkID=16544.

How to use Kerberos List

Kerberos List is a command-line tool that uses the following syntax:

klist [tickets | tgt | purge] [-?]

1. Click **Start**, click **All Programs**, click **Accessories**, and then click **Command Prompt**. This will open a command prompt window.
2. Type **klist.exe** *parameter* (where *parameter* is one of four parameters: **tickets**, **tgt**, **purge**, or **-?**) and then press ENTER. Each of these parameters is discussed in the following section.

Kerberos List parameters

Tickets Lists the current cached tickets of services to which you have authenticated since logging on. **Tickets** can be used to verify that a Kerberos ticket has been issued to the user. After an authentication request, several tickets should appear. This command will also show detailed information about the tickets obtained including the servers for which they were issued, the validity period, and ticket options.

Tickets displays the following attributes of all cached tickets:

Option	Description
End Time	Time when the ticket becomes invalid. After a ticket is past this time, it cannot be used to authenticate to a service.
KerbTicket Encryption Type	Encryption type used to encrypt the Kerberos ticket.
Renew Time	Maximum lifetime of a renewable ticket (see TicketFlags in the following table). To continue using this ticket, you must renew it before reaching the established End Time and before the expiration date established in RenewUntil.

Server	Server and domain for the ticket.

tgt Lists the initial Kerberos ticket-granting ticket (TGT). **Tgt** displays the following attributes of the currently cached ticket:

Option	Description
AltTargetDomainName	Name supplied to InitializeSecurityContext that generated this ticket, typically an SPN.
DomainName	Domain name of the service.
EndTime	Time when the ticket becomes invalid. When a ticket is past this time, it cannot be used to authenticate to a service.
FullServiceName	Canonical name of the account principal for the service.
KeyExpirationTime	Expiration time from the KDC reply.
RenewUntil	Maximum lifetime of a renewable ticket (see TicketFlags). To continue using a ticket, you must renew it. Tickets must be renewed before the expiration time set in End Time and in RenewUntil.
ServiceName	A TGT is a ticket for the Key Distribution Center (KDC) service. The service name for a TGT is krbtgt.
StartTime	Time when the ticket becomes valid.
TargetDomainName	For a cross-realm ticket, this is the realm, rather than the issuing realm, in which the ticket is good.
TargetName	Service name for which the ticket was requested. This is the name of a servicePrincipalName property on an account in the directory.
TicketFlags	Kerberos ticket flags set on the current ticket in hexadecimal. The Kerberos Tray tool displays these flags on the Flags tab.
Time Skew	The reported time difference between the client computer and the server computer for a ticket.

purge Deletes all Kerberos tickets held by the user. **Purge** destroys all tickets that you have cached, so use this with caution. It might stop you from being able to authenticate to resources. If this happens you must log off, and then log on again.

-? Displays command-line help

Kerbtray.exe: Kerberos Tray

Kerberos Tray is a graphical user interface tool that displays ticket information for a computer running Microsoft's implementation of the Kerberos version 5 authentication protocol. Kerberos Tray is supported for Windows Server 2003, Windows XP, and Windows 2000.

You can view and purge the ticket cache by using the Kerberos Tray tool icon located in the notification area of the desktop. By positioning the cursor over the icon, you can view the time left until the initial TGT expires. The icon also changes in the hour before the Local Security Authority (LSA) renews the ticket.

How to install Kerberos Tray

Kerberos Tray is included in the Windows Server 2003 Resource Kit and the Windows 2000 Resource Kit.

You can download Kerbtray.exe from "Windows Server 2003 Resource Kit Tools" on the Microsoft Download Center at http://go.microsoft.com/fwlink/?LinkID=16544.

How to use Kerberos Tray

1. To run Kerberos Tray, double-click the **Kerbtray** file. The Kerbtray icon appears in the notification area.
2. To open the main Kerbtray window, double-click the Kerbtray icon in the notification area. Information about all tickets for the current user will be displayed. Ticket options for each ticket are displayed on the **Flags** tab.
3. To purge tickets, right-click the Kerbtray icon in the notification area, and then click **Purge Tickets**.

Tokensz.exe: Kerberos Token Size

You can use Kerberos Token Size to verify whether the source of the Kerberos errors stems from a maximum token size issue. The tool will simulate an authentication request and report the size of the resulting Kerberos token. The tool will also report the maximum supported size for the token. If the size of the token exceeds this maximum supported value, then Kerberos authentication will exhibit the maximum token size behavior seen when group membership overloads PAC. This issue is described in more detail in Group Membership Overloads PAC earlier in this white paper.

Kerberos Token Size is a command-line tool that you can use to view the maximum Kerberos token size for a given account to a given service. To view maximum Kerberos token size, you must run the tool on a computer that is a member of an Active Directory domain.

When Kerberos Token Size is run from a client, it shows:

- The maximum Kerberos token size for the authentication package requested.
- The maximum Kerberos token size required to authenticate to the service.

How to install Kerberos Token Size

You can download Tokensz.exe from the Microsoft Download Center at http://go.microsoft.com/fwlink/?LinkId=25830.

Using Kerberos Token Size

- Example command:

  tokensz /compute_tokensize /package:negotiate /use_delegation /target_server:host/*ServerName*

- Example output:

```
Name: Negotiate Comment: Microsoft Package Negotiator

Current PackageInfo->MaxToken: 12128

MaxTokenSize (incomplete context): 2181
```

In this example:

- **Name** shows the name of package chosen, in this case Negotiate, which is the Microsoft Package Negotiator.
- **Current PackageInfo** shows the current MaxTokenSize value in registry when package was initialized.
- **MaxTokenSize** is the maximum token size required to authenticate to *ServerName.*
- **Incomplete context**. The tool will return **(incomplete context)** when it does not have the key for all the tickets.

> Note
> **Incomplete context versus complete context.** Two situations might cause the tool to return **(complete context)**:
>
> - Using the **/system** switch. The tool will run the test in the system context and thus have the key needed to open the service ticket.
> - Using the **/serveruser** switch and specifying the password. The service ticket can be opened.

Syntax and parameters

Kerberos Token Size supports two optional sets of command-line parameters, **compute_tokensize** or **calc_groups**.

- **Compute_tokensize** syntax is:

  tokensz /**compute_tokensize [/package:***PackageName*] [/**target_server:***TargetName*] [/**user:***UserName*] [/**domain:***UserDomain*] [/**password:***ClientPassword*] [/**serveruser:***UserName*] [/**serverdomain:***ServerDomain*] [/**serverpassword:***ServerPassword*] [/**dump_groups**] [/**system**] [/**nopac**][/**use_delegation**] [/**purge_tickets:**[*SPN* | NULL]]

  /compute_tokensize

  This switch will use the SSPI APIs InitializeSecurityContext() / AcceptSecurityContext to compute the maximum token size required to authenticate using the Kerberos protocol.

  /**package:***PackageName*.

The default package, if none is specified, is the Kerberos package. For many applications — for example, Group Policy — the Negotiate package is used, which increases the maximum token size by about 200 bytes. To simulate the Negotiate package, use **/package:negotiate**.

/**target_server:***SPN/ServerName*.

The proper usage of this parameter is to include an SPN — for example, host/dc.tailspintoys.com — which directs Kerberos Token Size to obtain the token size for a ticket to this specific server. If this parameter is omitted, the Kerberos package will get a service ticket to the user account running the tool, in many cases using the User-to-User extension to the Kerberos protocol. For more information about SPNs, see Need an SPN Set earlier in this white paper.

> **Note**
> If you provide a **target_server** parameter, you must specify the [**/serveruser:***UserName*] [**/serverdomain:***ServerDomain*] [**/serverpassword:***ServerPassword*] parameters corresponding to the account on which the SPN is registered. Alternatively, you can run the tool under the service or computer account on which the SPN is registered. If you choose to run the tool under the account corresponding to the SPN and you want to get the MaxTokenSize value for a user, you must specify the user's credentials using the [**/user:***UserName*] [**/domain:***UserDomain*] [**/password:** *ClientPassword*] parameters.

In nearly all cases, the maximum token size is determined by the first call to InitializeSecurityContext(). Thus, if you have connectivity issues to a target server you are troubleshooting, you can select another server in the same domain as a target server for this tool and be reasonably assured that accurate results will be reported. That is, if the target server is in the same domain as the user, the credentials will contain global group, domain local groups, and universal group membership. On the other hand, if you were to choose a target server in a different domain, the credentials will include the domain local groups for that domain.

/**user:***UserName*

/**domain:***UserDomain*

/**password:***ClientPassword*

These switches enable you to specify client credentials. For example, if you have a user experiencing maximum token size issues, you can run this tool with that user's credentials to get an accurate idea of the MaxTokenSize values. Alternatively, the user can run the tool.

/**serveruser:***UserName*

/**serverdomain:***ServerDomain*

/**serverpassword:***ServerPassword*

These switches enable you to specify server credentials. They must match the credentials of the account specified in the [**/target_server:***TargetName*] parameter, or you will only get information for the incomplete context. In many cases — for example, computer accounts — it is not possible to know the service password, so these switches are useful when the target is a service account.

/**dump_groups**

This switch extracts and displays all of the user's token groups, relative to the server. The output of this switch is dependent on which server is being used. For example, domain local group membership is relative to which domain the service resides in. The choice of target domain will affect the number of groups reported and the maximum token size.

/**system**

You must have administrator rights to use this switch, which runs the test under the Local System context of the computer that the test is being run on. In these cases, you must use the [**/user:***UserName*] [**/domain:***UserDomain*] [**/password:***ClientPassword*] switches, because the tool will be running under the Local System context. This switch is useful to see what groups are present in the Local System logon context.

/**nopac**

This switch uses the SEC_WINNT_AUTH_IDENTITY_ONLY flag for AcquireCredentialsHandle(), which informs the KDC not to include a PAC in the TGT or the service ticket. Because a service ticket's size is primarily influenced by the PAC, you can use this option to determine the size of the Kerberos ticket without including the PAC.

/**use_delegation**

This switch passes the ISC_REQ_DELEGATE flag into InitializeSecurityContext(). If you are testing delegation scenarios, use this flag in conjunction with a server in the **/target_server** parameter that has the **Trusted for Delegation** option selected. When delegation is used, the system passes both a service ticket and a TGT to the remote server. This approximately doubles the required maximum token size, and is responsible for several Group Policy issues that might be encountered.

/**purge_tickets:**[*SPN* | NULL]

You can use this switch to purge all of the tickets for the user before starting the token size calculation. This is important if you want to judge the effects of adding groups to a user's maximum token size. This parameter provides the same functionality as using Kerberos List to purge a user's ticket.

- **Calc_groups** syntax is:

  tokensz /calc_groups *ClientName* [**/user:***UserName*] [**/domain:***UserDomain*] [**/password:***ClientPassword*] [/system]

  /**calc_groups**

  This option is only available if you are using Windows Server 2003 KDCs and running the tool on a computer running Windows Server 2003. The parameter passed to this option is a user and (optionally) a domain that this user is a member of. Kerberos Token Size will list

all the groups that this user is a member of. If the user is a member of a large number of groups — generally more than 70-120 — this user might exhibit maximum token size issues.

/**user:***UserName*

/**domain:***UserDomain*

/**password:***ClientPassword*

These switches enable you to specify client credentials.

/**system**

You must have administrator rights to use this switch, which runs the test under the Local System context of the computer that the test is being run on. In these cases, you must use the [/**user:***UserName*] [/**domain:***UserDomain*] [/**password:***ClientPassword*] switches, because the tool will be running under the Local System context. This switch is useful to see what groups are present in the Local System / logon context.

Examples of Kerberos Token Size in Use

Example 1: Incomplete context

To determine the maximum Kerberos token size using incomplete context:

- Type the following at the command line:

  tokensz /compute_tokensize /package:negotiate /use_delegation /target_server:host/server1

- When you press ENTER, the following output is displayed:

```
Name: Negotiate Comment: Microsoft Package Negotiator

Current PackageInfo->MaxToken: 12128


MaxTokenSize (incomplete context): 2181
```

In this example:

MaxTokenSize (incomplete context) indicates that the protocol could not perform all legs of authentication. In this case, **(incomplete context)** was returned because the server was specified as server 1, but the test was run under the user account. However, this is still a reasonable estimation of the maximum token size required for this user to authenticate to server 1.

Example 2: Administrator account to server host with delegation requested

To determine the maximum Kerberos token size for administrator to the host server 1:

- Type the following at the command line:

  tokensz /compute_tokensize /package:negotiate /target_server:host/server1 /user:administrator /password:*ClientPassword* **/domain:***UserDomain* **/use_delegation**

- When you press ENTER**,** the following output is displayed:

```
Name: Negotiate Comment: Microsoft Package Negotiator

Current PackageInfo->MaxToken: 12128
```

```
Asked for delegate, but didn't get it


Check if server is trusted for delegation.
QueryKeyInfo:
Signature algorithm =
Encrypt algorithm = RSADSI RC4-HMAC
KeySize = 128
Flags = 2081e
Signature Algorithm = -138
Encrypt Algorithm = 23
 Start:4/2/2003 5:54:19
 Expiry:4/2/2003 6:54:19
Current Time: 4/2/2003 5:54:19
MaxToken (complete context)  1375
```

In this example:

- **Asked for delegate, but didn't get it** indicates that delegation was not used. This happens if the target server is not trusted for delegation, or if the user account has the **Account is sensitive and cannot be delegated** option selected.
- **MaxToken (complete context)** indicates that all authentication legs have been completed, and that this is a reliable value for maximum token size for server 1.

Example 3: Using /calc_groups

To calculate group membership for user 1:

- Type the following at the command line:

  tokensz /calc_groups user1

When you press ENTER, the tool returns a list of Kerberos token contents. In this example, the following output is displayed:

```
Username = user1
TS Session ID: 0
User
    S-1-5-21-148402017-3776891892-3157626230-1945
Groups:
 00   S-1-5-21-148402017-3776891892-3157626230-513      Attributes -
Mandatory Default Enabled
```

```
 01   S-1-1-0   Attributes - Mandatory Default Enabled

 02   S-1-5-32-545      Attributes - Mandatory Default Enabled

 03   S-1-5-32-554      Attributes - Mandatory Default Enabled

 04   S-1-5-2   Attributes - Mandatory Default Enabled

 05   S-1-5-11  Attributes - Mandatory Default Enabled

 06   S-1-5-15  Attributes - Mandatory Default Enabled

 07   S-1-5-5-0-17077419        Attributes - Mandatory Default Enabled
LogonId

Primary Group:

    S-1-5-21-148402017-3776891892-3157626230-513

Privs

 00 0x000000017 SeChangeNotifyPrivilege           Attributes - Enabled
Default

 01 0x000000006 SeUnsolicitedInputPrivilege       Attributes - Enabled
Default


Auth ID  0:10494b4

Impersonation Level: Identification

TokenType  Impersonation
```

Setspn.exe: Manipulate Service Principal Names for Accounts

The Setspn utility sets SPNs. Because SPNs are security-sensitive, you can only set SPNs for user objects if you have domain administrator privileges. Setspn.exe is included in the Windows Server 2003 Support Tools.

How to use Setspn

- To add an SPN, you can type the following at a command prompt:

  setspn –A *ServiceClass*/*Host*:*Port AccountName*

- To delete an SPN, you can type the following at a command prompt:

  setspn –D *ServiceClass*/*Host*:*Port AccountName*

- To view the SPNs that are registered for an account, you can type the following at a command prompt:

  setspn –L *AccountName*

- To reset the default SPN registrations for the host names for an account, you can type the following at a command prompt:

  setspn –R *AccountName*

The following section discusses the parameters listed above.

- *ServiceClass*. There are many different types of SPNs, and each service that is running on a computer should have the appropriate SPN service class assigned to it. If an application is written to take advantage of Kerberos authentication and delegation, it has the specific type of SPN that it needs to access predetermined. For example, when Internet Explorer 5.5 and later uses the Kerberos protocol to authenticate to a Web server, it looks for the **http/** SPN, whereas a SQL Server client looks for the **MSSQLSvc/** SPN. If the wrong service class is used on an SPN, then the SPN will not be located when a service searches for it.
- *Host*. The computer to which the SPN belongs is all the names by which a computer on which the service is running can be referenced. This usually includes a NetBIOS name, the FQDN, and any aliases that might have been assigned to this computer A separate SPN will need to be set for each name by which the computer can be referenced, with the *Host* parameter changing respectively.
- *Port.* The port that the service is running on. If this is a default port for that service (such as 80 for HTTP), then it can be omitted. However, it is recommended the port be included regardless of what service is running.
- *AccountName*. The name of the domain account under which the service runs. If the service runs as Local System or Network Service, you usually do not need to set an SPN explicitly for the service because most common SPN service classes will automatically be mapped to the host/ SPN which is in turn automatically generated for each computer account.

Debug Output

You can use debug output associated with Kerberos authentication to obtain information if other troubleshooting tools fail to produce useful information. Debug output is not meant to be used in day-to-day troubleshooting. It should only be used if there is absolutely no other means available to get information about the error. Debug output sometimes has extremely detailed error messages that might help you find the source of the problem. However, debug output also can contain messages that appear to indicate an error, but are actually normal messages that result from routine operations of the Kerberos protocol.

The directions in the following sections apply only to Windows Server 2003. In order to view the debug output on Windows 2000 Server, you must obtain instructions and a checked build of the Kerberos dynamic-link libraries (DLLs) from Microsoft Product Support Services.

How to turn on debug output

There are a number of ways to view the debug output from Kerberos. The easiest way is by logging the debug output to a file and then opening this file in Notepad.

1. Click **Start**, click **Run**, type **regedit.exe,** and then press ENTER.
2. Open the following registry key:

> **Caution**
>
> Incorrectly editing the registry might severely damage your system. Before making changes to the registry, you should back up any valued data on the computer.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters\

3. Create the following entry:

   Value: KerbDebugLevel

   Type: DWORD

   Data: c0000043 (this value will print the most standard set of debug messages. Try it first. If you still want to see more output, set it to ffffffff).

4. Create the following entry in the same registry location:

   Value: LogToFile

   Type: DWORD

   Data: 1

5. Reproduce the error

6. Open the file lsass.log, located in the System32 directory of your Windows folder. You can find the debug output inside this file.

> **Note**
> After you have obtained the necessary output, delete the two registry keys that you added in order to return the system to its full performance.

You might want to view and print the debug output in real time, as the errors actually happen. To do this, you can use a debugging tool called Ntsd. Ntsd is included in both Windows 2000 and Windows Server 2003.

How to use Ntsd to view real-time debug output from Kerberos authentication

> **Note**
> Ntsd is included as a courtesy to software developers. Only system developers should use this command. For more information, see the Help file included with Ntsd.

1. Click **Start**, click **Run**, and then type **regedit.exe**.

> **Caution**
> Incorrectly editing the registry might severely damage your system. Before making changes to the registry, you should back up any valued data on the computer.

2. Open the following registry key:
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters\
3. Create the following entry:

   Value: KerbDebugLevel

   Type: DWORD

   Data: c0000043 (this value will print the standard set of debug message. If you still want to see more output, set it to ffffffff).
4. Determine the Process Identifier (PID) for the lsass.exe process from the Task Manager.
   a. Click the **Processes** tab.
   b. Select the **View** menu and choose **Select Columns**.
   c. Click the **PID (Process Identifier)** check box and then click **OK**.
5. Click **Start**, **Run**, and then type **ntsd -p** *PID of lsass.exe*.

   This will start the debugger and attach it to lsass. While the debugger loads, you might need to wait a few minutes before the system presents a prompt.
6. At the prompt, type **g**. The debugger will now print out any errors that Kerberos authentication encounters.
7. Try to authenticate using the Kerberos protocol, and then check the debug output for any error messages that might further elaborate upon the Kerberos errors seen in the event log.

The following section about Kerberos errors lists many debug messages that are associated with Kerberos errors.

Important

After you have viewed the debug output, exit Ntsd correctly. If you do not exit Ntsd correctly, you can terminate the lsass process, which will force a system restart. To exit Ntsd, press CTRL+C. At the prompt, type **qd** to quit the debugger. To return the system to its full performance level, remove the KerbDebugLevel registry entry.

Kerberos Errors: Codes, Possible Causes, Resolutions

This section explains the causes and resolutions for the various Kerberos errors seen in the event logs or that Network Monitor has traced. These error codes are taken from the Kerberos RFC 1510 and draft extensions; all of the numbers and names are those that are used in the RFC. The first section explains the format of an error message and what each element means.

RFC Hex Error Value - Error Code: Description

Associated internal Windows error codes

These status codes are the internal error messages returned by Microsoft's implementation of the Kerberos protocol. Because these error codes are proprietary, they must be converted into the standard RFC error codes in order to ensure RFC compliance. Because there are many more potential internal errors than there are RFC errors, RFC errors map to more than one internal error. This section shows which internal error codes map to a specific Kerberos error.

Corresponding debug output messages

When debug output is enabled, this is what will be printed if the corresponding Kerberos error is encountered. Debug is often more detailed than a general Kerberos error and might help to pinpoint the source of the problem.

There are two types of debug output messages:

- **DebugLog.** Will be printed if debug output is enabled.
- **D_DebugLog.** Only printed if a checked build of the Kerberos DLL has been installed on the computer. (Checked builds can only be obtained from Microsoft Product Support Services.)

Many of the debug output messages will only be generated in Kerberos client application development environments. An example of this would be if a Kerberos application requested an authentication service request with invalid options.

Possible Causes and Resolutions

This section explains the possible causes of the error. Not all errors mean that something is wrong; some are returned during normal operation. If the latter is the case, it is explained as such.

Resolution This section explains how to resolve a particular error, if applicable.

0x6 - KDC_ERR_C_PRINCIPAL_UNKNOWN: Client not found in Kerberos database

Associated internal Windows error code

STATUS_NO_SUCH_USER

Corresponding debug output messages

- D_DebugLog("KLIN(%x) No principal name supplied to AS request - not allowed\n")
- DebugLog("KdcGetS4UTicketInfo normalize returned referral for S4U client\n")
- DebugLog("Failed Authz check\n")

Possible Causes and Resolutions

This error can occur if the domain controller cannot find the account name in Active Directory. This can occur in three scenarios:

- The actual account does not exist.

Resolution Verify that the name is in the Active Directory. If the principal name is not in the local Active Directory, but you know the account should exist and the user was recently added to the domain, verify that Active Directory replication is current.

- A new account has been created and has not yet replicated to the KDC that the client is using for authentication.

Resolution It could be that the updates have not yet reached the domain controller that is acting as the KDC for that user. For information about how to manually initiate an update, see "Initiating Replication between Active Directory Direct Replication Partners" in the Microsoft Knowledge BaseMicrosoft Knowledge Base at http://go.microsoft.com/fwlink/?LinkId=23063.

- The user's account has expired and the **Enforce user logon restrictions** Group Policy object (GPO) setting is enabled.

Resolution If the user name is in Active Directory, determine whether the account has expired. **Enforce user logon restrictions** forces the domain controller to check the user's account each time a TGT is presented and account expiration will cause the domain controller to refuse an otherwise valid TGT.

0x7 - KDC_ERR_S_PRINCIPAL_UNKNOWN: Server not found in Kerberos database

Associated internal Windows error codes

- STATUS_NO_TRUST_SAM_ACCOUNT
- STATUS_OBJECT_NAME_NOT_FOUND
- STATUS_KDC_UNABLE_TO_REFER

Corresponding debug output messages

- D_DebugLog("Wrong S4UProxytarget %wZ %wZ\n")
- DebugLog("KdcFindReferralTarget KLIN(%x) Needed exact match and got a transitively-trusted domain.\n")
- D_DebugLog("No referral info for %wZ\n")
- D_DebugLog("Got UPN w/ uknown trust path %x\n")
- DebugLog("No auth info for this trust: %wZ. %ws, line %d\n")

Possible Causes and Resolutions

This error can occur if the domain controller cannot find the server's name in Active Directory. This error is similar to KDC_ERR_C_PRINCIPAL_UNKNOWN except that it occurs when the server name cannot be found.

This might be because:

- The actual name is missing.

Resolution Verify that the service is registered and has an SPN set. For more information about setting SPNs, see Need an SPN Set earlier in this white paper.

- A new computer account has been created and has not yet replicated to the KDC that the client is using for authentication.

Resolution It could be that the updates have not yet reached the domain controller that is acting as the KDC for that client. For information about how to manually initiate an update, see "Initiating Replication between Active Directory Direct Replication Partners" in the Microsoft Knowledge Base at http://go.microsoft.com/fwlink/?LinkId=23063.

- UDP fragmentation is occurring. If the SPN is set, or if the request failed for an initial TGT (requesting a TGT does not require any SPNs to be set manually), then UDP fragmentation might be causing the failure.

Resolution Capture a network trace with Network Monitor and compare it to the sample trace associated with UDP fragmentation in Appendix A. If you determine that the cause is UDP fragmentation, see UDP Fragmentation earlier in this white paper for information about how to resolve the issue.

- A trust path has been incorrectly configured. If the SPN is set correctly and this error is not related to UDP fragmentation, then there might be an error while doing the referral. This can occur if the trust path leading to the server has been incorrectly configured.

Resolution Verify that there is a valid trust path to the server's domain and that this path can be followed. You can do this by attempting to logon as a user in the server's domain in the client domain. If the logon is successful and occurs using the Kerberos protocol (this can be verified in the security log), then the trust path is set up correctly.

0x8 - KDC_ERR_PRINCIPAL_NOT_UNIQUE: Multiple principal entries in database

Associated internal Windows error codes

- STATUS_OBJECT_NAME_COLLISION
- KDCEVENT_NAME_NOT_UNIQUE

Corresponding debug output messages

- None

Possible Cause and Resolution

- This error occurs if duplicate principal names exist. Unique principal names are crucial for ensuring mutual authentication. Thus, duplicate principal names are strictly forbidden, even across multiple realms. Without unique principal names, the client has no way of ensuring that the server it is communicating with is the correct one.

Resolution You must remove the duplicate principal name in order for Kerberos authentication to function. To find the duplicate SPN, you can use the LDP tool, or you can use the Ldifde utility. The two methods are described below.

How to use the LDP tool

> **Note** If you do not have the Windows Server 2003 Support Tools installed, install them from the Windows Server 2003 CD-ROM before proceeding. (The Setup executable file for Support Tools is located on the CD-ROM in the Support\Tools folder. The installation does not require you to restart the computer, but you might have to restart the computer so that the environment variables are updated.

1. Click **Start**, and then click **Run**.
2. In the **Open:** text box, type **LDP**, and then click **OK**.
3. On the **Connection** menu, click **Connect**.
4. If you are on the domain controller, leave the default settings, and then click **OK**. If you are not on the domain controller, type the domain controller name in the **Server** text box and then click **OK**.
5. On the **Connection** menu, click **Bind**.
6. Type *User*, *Password***,** and *Domain* in the corresponding text boxes, and then click **OK**.
7. On the **View** menu, click **Tree**.
8. In the **Tree View** dialog box, type the base distinguished name in the **BaseDN** text box or select it from the pull-down menu.
9. On the **Browse** menu, click **Search**.
10. In the **Search** dialog box, type the base distinguished name in the **BaseDN** text box or select from the pull-down menu.
11. In the **Search** dialog box, type the following in the **Filter** text box:

    serviceprincipalname=*SPN*/*FQDN*
12. For *SPN*, type the Service Principal Name that the error refers to — for example, HOST for computer accounts, HTTP for Web services.
13. Under **Scope**, click the **Subtree** option.
14. Click **Run**.

For more information about using ldp.exe to search Active Directory, see “Using Ldp.exe to Find Data in the Active Directory” in the Microsoft Knowledge Base at http://go.microsoft.com/fwlink/?LinkId=23064.

How to use Ldifde

To use the Ldifde utility to extract the accounts for the domain, or from the suspected container or OU:

1. From the domain controller, open a command prompt, and then:

- For computer accounts, type

ldifde -f *filename* **-d** *BaseDistinguishedName* **-r (objectclass=computer) -p subtree**

— or —

- For user accounts, type

  ldifde -f *filename* **-d** *BaseDistinguishedName* **-r (objectclass=user) -p subtree**

Note

If the accounts that seem to have the duplicate SPNs are located in a certain OU (for example, Florida), you can refine the base distinguished name. For example: **-d** ou=sales,dc=tailspintoys,dc=com .

2. Open the text file in Notepad, and then search for the SPN that is reported in the security event log.
3. Note the accounts under which the SPN is located.

Use Setspn to rename or delete the duplicates. For more information about setting SPNs, see Need an SPN Set earlier in this white paper.

0xA - KDC_ERR_CANNOT_POSTDATE: Ticket not eligible for postdating

Associated internal Windows error codes

- None

Corresponding debug output messages

- DebugLog(“Asked for postdate but start time not present\n”)

Possible Causes and Resolutions

- This error can occur if a client requests postdating of a Kerberos ticket. Postdating is the act of requesting that a ticket’s start time be set into the future.

Resolution Windows KDCs do not support postdating and clients should never request the postdating of a ticket.

- There is a time difference between the client and the KDC.

Resolution Verify whether there is a time difference between the client and the KDC. For information about this error and about how to resolve time differences, see Time Synchronization (Clock Skew) earlier in this white paper.

0xB - KDC_ERR_NEVER_VALID: Requested start time is later than end time

Associated internal Windows error codes

- None

Corresponding debug output messages

- DebugLog("Client asked for endtime before starttime\n")

Possible Cause and Resolution

- There is a time difference between the KDC and the client.

Resolution For Kerberos authentication to work, you must synchronize clocks on the client and on the server. For more information about this error and how to resolve it, see Time Synchronization (Clock Skew) earlier in this white paper.

0xC - KDC_ERR_POLICY: KDC policy rejects request

Associated internal Windows error codes

- STATUS_ACCOUNT_RESTRICTION
- STATUS_AUTHENTICATION_FIREWALL_FAILED
- STATUS_CROSSREALM_DELEGATION_FAILURE
- STATUS_PASSWORD_RESTRICTION
- STATUS_DOMAIN_TRUST_INCONSISTENT

Corresponding debug output messages

- D_DebugLog("KLIN(%x) Additional ticket client realm is wrong: %s instead of %s\n")
- DebugLog("Trying to renew a non-renewable ticket or against policy\n")
- DebugLog("AuthzInitializeContextFromSid() failed in KerbPerformTgsAccessCheck %x\n")
- DebugLog("CrossOrg authz AC failed %x\n")
- DebugLog("AuthzAccessCheck() failed in KerbPerformTgsAccessCheck%x\n")
- DebugLog("Trying to do S4UProxy to another realm %wZ\n")
- DebugLog(Failed to change password for user %wZ: 0x%x\n")
- DebugLog("Missing PAC verifier in S4U Tickets\n")
- DebugLog("S4U Pac verifier missing @ sign\n")
- DebugLog("KdcBuildTicketTimesAndFlags asked for allow postdate but not allowed: ")
- DebugLog("Trying to validate a valid ticket\n")

Possible Causes and Resolutions

- KDC_ERR_POLICY is usually the result of logon restrictions in place on a user's account. This error is usually accompanied by an error packet which might contain additional information that can be viewed with a Network Monitor capture.

Resolution Use Active Directory Users and Computers to verify whether restrictions in place on this account might prevent this user from logging on. To use Active Directory Users and Computers:

1. Click **Start**, click **Run**, and then type:

   dsa.msc

2. Locate the user that is having logon problems, right-click the user's account, and then click **Properties**.
3. Verify settings on the **Account** tab for valid logon hours and computer to which this user is allowed to log on.

- Constrained delegation is being attempted across multiple domains.

Resolution No resolution. Windows Server 2003 does not support constrained delegation across multiple domains.

- The server receives a ticket in which client's realm does not match the local realm.

Resolution Confirm the error with a Network Monitor capture. The only way to eliminate this error is to ensure that the server and client are in the same realm (domain).

0xD - KDC_ERR_BADOPTION: KDC cannot accommodate requested option

Associated internal Windows error codes

- STATUS_NO_MATCH

Corresponding debug output messages

- DebugLog("Asked for forwarded but not allowed\n")
- DebugLog("Asked for proxy but not allowed\n")
- DebugLog("Asked for postdate but not allowed\n")
- D_DebugLog("s4u set, but no ticket\n")
- D_DebugLog("Couldn't decrypt evidence ticket %x\n")
- D_DebugLog("Trying to mix S4U proxy and self requests\n")
- D_DebugLog("KLIN(%x) Client %wZ sent AS request with no server name\n")
- D_DebugLog("KLIN(%x) Attempt made to renew non-renewable ticket\n")
- DebugLog("Client tried to use pkinit w/o client cert\n")
- DebugLog("User supplied bad cert type: %d\n")

Possible Causes and Resolutions:

- Impending expiration of a TGT.

Resolution Confirm the cause by verifying the expiration time on the TGT. To do this, use the Kerberos List parameter **tgt**. If you confirm that this is the cause, you need do

nothing more, because the TGT will be automatically renewed or a new one will be requested if needed. For example, Windows XP and Windows Server 2003 will recover from this automatically.

- The SPN to which the client is attempting to delegate credentials is not in its **Allowed-to-delegate-to** list.

Resolution

1. Use Network Monitor to determine the SPN to which the client is attempting to delegate credentials. You will need this information in a later step.
2. Click **Start**, click **Run**, and then open Active Directory Users and Computers by typing the following:

   dsa.msc
3. Right-click the user or service account that has problems authenticating, and then click **Properties**.
4. Click the **Delegation** tab.
5. The **Allowed-to-delegate-to** list is the list of servers shown under the heading, **Services to which this account can present delegated credentials**.
6. Add the SPN the client is attempting to delegate to (information from the captured data you obtained in Step 1) to the **Allowed-to-delegate-to** list for that client. This will tell the KDC that this client is indeed allowed to authenticate to this service. The KDC will then grant the client the appropriate ticket.

For information about setting up service accounts for delegation, see "HOW TO: Configure Computer Accounts and User Accounts So That They Are Trusted for Delegation in Windows Server 2003 Enterprise Edition" in the Microsoft Knowledge Base at http://go.microsoft.com/fwlink/?LinkId=23067.

- The server does not support constrained delegation or protocol transition. (Windows 2000 does not support constrained delegation or protocol transition.)

0xE - KDC_ERR_ETYPE_NOTSUPP: KDC has no support for encryption type

Associated internal Windows error codes

- STATUS_KDC_UNKNOWN_ETYPE
- SEC_E_ETYPE_NOT_SUPP

Corresponding debug output messages

- DebugLog("Using a CryptSystem with a BlockSize(%d) > MAX(%d)\n")
- DebugLog("Null password or crypt list passed to KerbFindCommonCryptSystem\n")
- D_DebugLog("Got more than 20 crypto systems in password list\n")

- DebugLog("KdcCheckForEtype no intersection between client and server Etypes!\n")
- DebugLog("KLIN(%x) Failed to find common ETYPE: 0x%x\n")
- DebugLog("KdcUnpackAdditionalTickets no encryption key found in krbtgt's OldPassword\n")

Possible Causes and Resolutions

In general, this error occurs when the KDC or a client receives a packet that it cannot decrypt. Four possible scenarios are:

- UNIX interoperability scenarios in which the UNIX KDC attempts to use 3DES to encrypt its tickets.

  Resolution The UNIX KDC must be configured to use another encryption type, such as DES or RC4. Windows operating systems do not support 3DES.

- Interoperability scenarios in which the target server does support the encryption type used by the KDC.

  Resolution Configure target server to support the RFC standard encryption type RC4 or contact the vendor.

- Recent migration from Windows NT to Windows Server 2003.

  Resolution Change the password of the user who is having difficulty logging on.

- The administrator account's password has not been changed since the domain was created.

  Resolution Change the administrator's password to eliminate the error.

0xF - KDC_ERR_SUMTYPE_NOSUPP: KDC has no support for checksum type

Associated internal Windows error codes

- SEC_E_CHECKSUM_NOT_SUPP

Corresponding debug output messages

- DebugLog("Unsupported signature algorithm (not MD5)\n")

Possible Cause and Resolution

- The KDC, server, or client receives a packet for which it does not it have a key of the appropriate encryption type. The result is that the computer is unable to decrypt the ticket. This error is common in UNIX interoperability scenarios when a new account is created. The new account might have an incompatible key associated with it.

  Resolution Change the password on the account to re-create the key, which should eliminate the error.

0x10 - KDC_ERR_PADATA_TYPE_NOSUPP: KDC has no support for padata type

Associated internal Windows error codes

- STATUS_UNSUPPORTED_PREAUTH
- STATUS_NOT_SUPPORTED

Corresponding debug output messages

- D_DebugLog("KLIN(%x) No pre-auth data in TGS request - not allowed.\n")

Possible Cause and Resolution

- Smart card logon is being attempted and the proper certificate cannot be located. This can happen because the wrong certification authority (CA) is being queried or the proper CA cannot be contacted.

Resolution

1. Verify that there is a functioning CA on the domain.
2. Verify that the client can locate the CA.

0x12 - KDC_ERR_CLIENT_REVOKED: Clients credentials have been revoked

Associated internal Windows error codes

- STATUS_ACCOUNT_DISABLED
- STATUS_ACCOUNT_EXPIRED
- STATUS_ACCOUNT_LOCKED_OUT
- STATUS_ACCOUNT_DISABLED
- STATUS_INVALID_LOGON_HOURS
- STATUS_LOGIN_TIME_RESTRICTION
- STATUS_LOGIN_WKSTA_RESTRICTION
- STATUS_ACCOUNT_RESTRICTION

Corresponding debug output messages

- None

Possible Causes and Resolution

- The account being denied authentication is disabled. This might be because of an explicit disabling or because of other restrictions in place on the account.

Resolution

Use Network Monitor to capture data associated with this error. The KDC_ERR_CLIENT_REVOKED error is usually accompanied by an error packet. The error packet might contain additional information that could help you diagnose the problem.

0x17 - KDC_ERR_KEY_EXPIRED: Password has expired – change password to reset

Associated internal Windows error codes

- STATUS_PASSWORD_EXPIRED
- STATUS_PASSWORD_MUST_CHANGE
- STATUS_ACCOUNT_LOCKED_OUT
- STATUS_NO_LOGON_SERVERS

Corresponding debug output messages

- None

Possible Cause and Resolution

- The user's password has expired.

Resolution The user should change password and logon again to obtain a new key.

0x18 - KDC_ERR_PREAUTH_FAILED: Pre-authentication information was invalid

Associated internal Windows error codes

- STATUS_WRONG_PASSWORD

Corresponding debug output messages

- DebugLog("KLIN(%x) CPAHandlerEncryptedTime::Check: failed to derived compid key 0x%lx.\n")

Possible Cause and Resolution

- The wrong password was provided.

Resolution Try authenticating again. It is possible that the password might have been incorrectly entered.

- Verify that the time on the KDC matches the time on the client. For more information about time differences and how to resolve them, see Time Synchronization (Clock Skew) earlier in this white paper.

0x19 - KDC_ERR_PREAUTH_REQUIRED: Additional pre-authentication required

Associated internal Windows error codes

- STATUS_WRONG_PASSWORD

Corresponding debug output messages

- None

Possible Causes and Resolution

- This error often occurs in UNIX interoperability scenarios. MIT-Kerberos clients do not request pre-authentication when they send a KRB_AS_REQ message. If pre-authentication is required (the default), Windows systems will send this error. Most MIT-Kerberos clients will respond to this error by giving the pre-authentication, in which case the error can be ignored, but some clients might not respond in this way.

Resolution Set the **Do not require Kerberos pre-authentication** flag on the user's account. Alternatively, consider upgrading to the most recent MIT reference distribution of Kerberos authentication.

0x1B - KDC_ERR_MUST_USE_USER2USER: Server principal valid for user2user only

Associated internal Windows error codes

- STATUS_USER2USER_REQUIRED

Corresponding debug output messages

- DebugLog("KLIN(%x) Service principal requires user2user ")
- DbugLog("KdcVerifyKdcRequest must use user2user UserAccountControl %#x, GenericUserName %wZ, PrincipalName: ")

Possible Causes and Resolution

- This error can be remapped from 0x7_KDC_ERR_S_PRINCIPAL_UNKOWN. Causes and resolutions of that error could apply to this error as well.
- The SPN being provided is not registered anywhere.

Resolution Use Network Monitor to verify the SPN being requested. Be sure that the SPN is registered for the appropriate service and under the correct account. For more information about SPNs and how to register them, see Need an SPN Set earlier in this white paper.

0x1C - KDC_ERR_PATH_NOT_ACCEPTED: KDC Policy rejects transited path

Associated internal Windows error codes

- STATUS_TRUST_FAILURE

Corresponding debug output messages

- D_DebugLog("Client from realm %s attempted to access non transitive trust to %wZ : illegal\n")
- DebugLog("TGT S4U Client from realm %s attempted to access non transitive trust to %wZ : illegal\n")
- DebugLog("Missing delegation info while transiting %p\n")

- D_DebugLog("KDC presented w/ a unknown Xrealm TGT (%wZ)\n")

Possible Causes and Resolutions

- A trust is incorrectly set up between two domains.

Resolution Verify that there is a two-way transitive trust set up between the user's domain and the domain on which the user is trying to access resources.

If the domain to which the user is trying to authenticate is in another forest, see "Cannot Use Kerberos Trust Relationships Between Two Forests in Windows 2000" in the Microsoft Knowledge Base at http://go.microsoft.com/fwlink/?LinkId=23069. This article describes why you cannot use internal Kerberos trust relationships between two forests in Windows 2000.

- Constrained delegation is being attempted across multiple domains.

Resolution No resolution. Windows 2000 does not support constrained delegation across multiple domains.

If constrained delegation is being attempted across multiple domains in Windows Server 2003, this error message will read: Constrained delegation is not currently supported across multiple domains.

0x1D - KDC_ERR_SVC_UNAVAILABLE: A service is not available

Associated internal Windows error codes

- STATUS_NO_LOGON_SERVERS

Corresponding debug output messages

- DebugLog("KLIN(%x) Service principal not allowed ")

Possible Cause and Resolution

- A client attempted to query the global catalog, but the global catalog was not available.

Resolution Verify that DNS is correctly set up and that the user's computer can contact the domain controller.

0x1F - KRB_AP_ERR_BAD_INTEGRITY: Integrity check on decrypted field failed

Associated internal Windows error codes

- None

Corresponding debug output messages

- D_DebugLog("Could not decrypt the ticket\n")

Possible Causes and Resolutions

- The authenticator was encrypted with something other than the session key. The result is that the client cannot decrypt the resulting message. The modification of the message could be the result of an attack or it could be because of network noise.

Resolution This error is similar to 0x29_KRB_APP_ERR_MODIFIED. See the resolutions listed under that error later in this white paper.

0x20 - KRB_AP_ERR_TKT_EXPIRED: Ticket expired

Associated internal Windows error codes

- STATUS_TIME_DIFFERENCE_AT_DC

Corresponding debug output messages

- DebugLog("Trying to renew a ticket past its renew time\n")
- DebugLog("Trying to renew an expired ticket\n")

Possible Cause and Resolution

- The smaller the value for the **Maximum lifetime for user ticket** Kerberos policy setting, the more likely it is that this error will occur. Because ticket renewal is automatic, you should not have to do anything if you get this message.

Resolution To change the **Maximum lifetime for user ticket** setting:

1. Click **Start**, click **All Programs**, click **Administrative Tools**, and then click **Domain Security Policy**.
2. Click **Accounts Policies**, and then click **Kerberos Policy**.
3. Increase the value for **Maximum lifetime for user ticket**.
4. Run **gpupdate /force** on any client computer on which you want this policy change to take effect immediately.

0x21 - KRB_AP_ERR_TKT_NYV: Ticket not yet valid

Associated internal Windows error codes

- STATUS_TIME_DIFFERENCE_AT_DC

Corresponding debug output messages

- DebugLog("Trying to validate a ticket before it is valid\n")

Possible Causes and Resolution

- The ticket presented to the server is not yet valid (in relationship to the server time). The most probable cause is that the clocks on the KDC and the client are not synchronized.

- If cross-realm Kerberos authentication is being attempted, then you should verify time synchronization between the KDC in the target realm and the KDC in the client realm, as well.

Resolution For more information about time differences in Kerberos authentication and how to resolve them, see Time Synchronization (Clock Skew) earlier in this white paper.

0x22 - KRB_AP_ERR_REPEAT: Request is a replay

Associated internal Windows error codes

- None

Corresponding debug output messages

- None

Possible Causes and Resolutions

This error indicates that a specific authenticator showed up twice — the KDC has detected that this session ticket duplicates one that it has already received. The cause could be:

- A bad network card.

Resolution Replace the network card in the computer if the cause is failing hardware.

- An attack is in progress.

0x23 - KRB_AP_ERR_NOT_US: The ticket isn't for us

Associated internal Windows error codes

- SEC_E_WRONG_PRINCIPAL

Corresponding debug output messages

- DebugLog("KLIN(%x) Ticket (%s) not for this service (%wZ).\n")
- D_DebugLog("KLIN(%x) Tgt reply is not for our realm: %s instead of %s\n")
- D_DebugLog("KLIN(%x) Verified ticket client realm is wrong: %s instead of %s\n")

Possible Cause and Resolution

- The server has received a ticket that was meant for a different realm.

Resolution Verify that DNS is set up correctly. Verify that packets are correctly routed across the network.

0x24 - KRB_AP_ERR_BADMATCH: Ticket and authenticator don't match

Associated internal Windows error codes

- None

Corresponding debug output messages

- DebugLog("Authenticator principal != ticket principal\n")
- D_DebugLog("Cert name doesn't match user name: %wZ, %wZ\n")
- DebugLog("KLIN(%x) Supplied U2U ticket is not for server: %wZ (%#x) vs. %wZ (%#x)\n")
- DebugLog("S4USelf requestor realm != service realm\n")

Possible Causes and Resolutions

- The KRB_TGS_REQ is being sent to the wrong KDC.
- There is an account mismatch during protocol transition.

Resolution Confirm DNS settings are correct for the domain. Verify that constrained delegation and protocol transition are correctly configured.

0x25 - KRB_AP_ERR_SKEW: Clock skew too great

Associated internal Windows error codes

- STATUS_TIME_DIFFERENCE_AT_DC

Corresponding debug output messages

- DebugLog("Client asked for endtime before starttime\n")

Possible Causes and Resolution

This error is logged if a client computer sends a timestamp whose value differs from that of the server's timestamp by more than the number of minutes found in the **Maximum tolerance for computer clock synchronization** setting in Kerberos policy.

Although this error might show up in the logs, it will not prevent a user from being authenticated. When this error is returned, the domain controller also supplies the correct time on the domain controller. The Kerberos client uses the correct domain controller time to attempt the authentication request a second time. Presuming that the user's credentials are valid, the user will be authenticated on the second try.

- This error can more commonly occur as the number of notebooks — that is, disconnected computers — in your network increases.

Resolution Beware that the higher you set the value of the **Maximum tolerance for computer clock synchronization** setting, the more susceptible the network becomes to replay attacks.

To set **Maximum tolerance for computer clock synchronization** Kerberos policy:

1. Open the domain security policy by clicking **Start**, **Programs**, **Administrative Tools**, **Local Security Policy**.
2. Click **Account Policies**, and then click **Kerberos Policy**.
3. Increase the value for **Maximum tolerance for computer clock synchronization**.

4. You can either wait for the policy change to propagate or you can run **gpupdate /force** on the client computers to force propagation immediately.

For more information, see Time Synchronization (Clock Skew) earlier in this white paper.

0x28 - KRB_AP_ERR_MSG_TYPE: Invalid msg type

Associated internal Windows error codes

- SEC_E_INVALID_TOKEN

Corresponding debug output messages

- DebugLog("Won't allow user2user with Datagram. %ws, line %d\n")

Possible Causes and Resolutions

- UDP is being attempted with User-to-User protocol. User-to-User is an extension of Kerberos authentication that enables secure servers to be run on personal computers.

Resolution Force Kerberos authentication to use TCP. For information about forcing Kerberos authentication to use TCP see "How to Force Kerberos to Use TCP Instead of UDP" in the Microsoft Knowledge Base at http://go.microsoft.com/fwlink/?LinkId=23043.

0x29 - KRB_AP_ERR_MODIFIED: Message stream modified

Associated internal Windows error codes

- SEC_E_WRONG_PRINCIPAL
- STATUS_WRONG_PASSWORD

Corresponding debug output messages

- DebugLog("Failed to verify message: %x\n",Status)
- DebugLog(""Failed to encrypt message: %x\n",Status)
- DebugLog("Failed to encrypt message (crypto mismatch?): %x\n")
- DebugLog("Checksum on TGS request body did not match\n")
- D_DebugLog("Failed to create S4U checksum\n")
- DebugLog("S4U PA checksum doesn't match!\n")
- DebugLog("Pac was modified - server checksum doesn't match\n")
- D_DebugLog(DEB_TRACE,"Could not decrypt the ticket\n")

Possible Causes and Resolutions

Some encrypted Kerberos authentication data sent by the client did not decrypt properly at the server because:

- A service ticket is issued to the local computer account, for which a host/ SPN is automatically created, instead of to the service account, for which no SPN has been created. The reason for this is that a service does not register an SPN for itself, yet the service belongs to a service class for which the computer will automatically map the SPN to a host/service class. (Examples of this are the HTTP and Common Internet File System (CIFS) service classes.) The result is that the service cannot decrypt the resultant ticket.

Resolution If the root cause appears to be that an SPN has not been set, verify that each service running on the target computer has an SPN set. Those services that do not have SPNs set might have had their SPNs remapped to the computer's host SPN. For more information about SPNs and how to set them, see Need an SPN Set earlier in this white paper.

- The authentication data was encrypted with the wrong key for the intended server.
- The authentication data was modified in transit by a hardware or software error, or by an attacker.
- The client sent the authentication data to the wrong server because incorrect DNS data caused the client to send the request to the wrong server.

Resolution Verify that DNS is functioning properly.

- The client sent the authentication data to the wrong server because DNS data was out-of-date on the client.

Resolution Verify that DNS is functioning properly.

- Two computers in different domains have the same name and the client sent the authentication data to the wrong computer.

Resolution Verify that there are not multiple computers with the same name, including NetBIOS names, anywhere on the network.

0x34 - KRB_ERR_RESPONSE_TOO_BIG: Response too big for UDP, retry with TCP

Associated internal Windows error codes

- STATUS_INVALID_BUFFER_SIZE

Corresponding debug output messages

- D_DebugLog"KLIN(%x) KDC response too big for UDP: %d bytes\n")

Possible Cause and Resolution

- The size of a ticket is too large to be transmitted reliably via UDP. In a Windows environment, this message is purely informational. A computer running a Windows operating system will automatically try TCP if UDP fails.

Resolution If this error occurs in a mixed operating systems environment, upgrade the UNIX KDCs to the latest MIT distribution of the Kerberos protocol, which supports TCP connections if UDP fails.

For information about forcing Kerberos to use TCP, see "How to Force Kerberos to Use TCP Instead of UDP" in the Microsoft Knowledge Base at http://go.microsoft.com/fwlink/?LinkId=23043.

0x3C - KRB_ERR_GENERIC: Generic error

Associated internal Windows error codes

- STATUS_INSUFFICIENT_RESOURCES

Corresponding debug output messages

- DebugLog("SpInitLsaModeContext failed to verify AP reply: 0x%x\n")
- DebugLog("Failed to decrypt AP reply: 0x%x. %ws, line %d\n")
- DebugLog("Failed to encode data: %d\n")
- DebugLog("KerbUnpackData Trying to unpack NULL data\n")
- D_DebugLog("Failed to unmarshal pac\n")
- DebugLog("Failed to get CLIENT Principal : 0x%x\n")
- DebugLog("Failed to get Client principal name: 0x%x\n")
- DebugLog("Failed to acquire KDC certificate private key: 0x%x\n")
- DebugLog("Trying S4UProxy w/ no PAC\n")
- D_DebugLog("KdcUnpackAdditionalTickets KLIN(%x) Trying to unpack null ticket or more than one ticket\n")
- D_DebugLog("The client of kpasswd did not ask for a sub key.\n")
- DebugLog("Failed to create token from ticket: 0x%x\n")
- D_DebugLog("No logon info for PAC - not adding resource groups\n")
- DebugLog("Failed to query domain info for %wZ: 0x%x. %ws, line %d\n")
- DebugLog("Failed to decrypt old password: 0x%x\n")
- DebugLog("KdcGetTicketInfo can't restrict user accounts if USER_EXTENDED_FIELD_SPN is not requested\n")

Possible Causes and Resolutions

- Group membership has overloaded the PAC.

Resolution For information about how to resolve this issue, see Group Membership Overloads PAC earlier in this white paper.

- Multiple recent password changes have not propagated.

Resolution You can wait for the changes to replicate, or you can force replication. To manually initiate replication see “Initiating Replication Between Active Directory Direct Replication Partners” in the Microsoft Knowledge Base at http://go.microsoft.com/fwlink/?LinkId=23063.

- Crypto subsystem error caused by running out of memory.

Resolution Restart system or end processes to free memory.

- SPN too long.

Resolution Use Network Monitor to capture network data. Examine the SPN being requested. Verify that it is a correctly formed SPN and is registered to a service on the network.

- SPN has too many parts.

Resolution Use Network Monitor to capture network data. Examine the SPN being requested. Verify that it is a correctly formed SPN and is registered to a service on the network.

0x44 - KDC_ERR_WRONG_REALM: (user-to-user)

Associated internal Windows error codes

- STATUS_NO_LOGON_SERVERS

Corresponding debug output messages

- DebugLog(“Client tried to logon to account in another realm\n”)
- DebugLog(“KLIN(%x) Failed to locate handle for referral realm”)
- D_DebugLog(“KLIN(%x) Client tried to logon to account in another realm\n”)
- D_DebugLog(“KLIN(%x) Request sent for wrong realm: %wZ\n”)

Possible Causes and Resolution

Although this error rarely occurs, it occurs when a client presents a cross-realm TGT to a realm other than the one specified in the TGT. Typically, this results from incorrectly configured DNS.

Resolution To confirm the cause, capture network data and verify that the realm listed in the TGT is different than the realm of the KDC that the TGT is being presented to. Then investigate why the TGT is routed to something other than the realm it was meant for.

Appendix A: Network Monitor Sample Traces

The following sections detail both normal Kerberos traffic and some situations that involve common Kerberos errors.

> **Note**
> The traces below have been altered to remove irrelevant or unnecessary information.

Kerberos Authentication During Normal Logon

In this example, the Kerberos client obtains a TGT and session ticket from a KDC.

```
  FRAME: Base frame properties
  ETHERNET: EType = Internet IP (IPv4)
  IP: Protocol = UDP - User Datagram; Packet ID = 7027; Total IP Length = 333;
Options = No Options
  UDP: Src Port: Unknown (1676); Dst Port: Kerberos (88); Length = 313 (0x139)
  KERBEROS: KRB_AS_REQ
          KERBEROS: Realm (realm[2]) =multi
          KERBEROS: Server name (sname[3]) =krbtgt/multi
              KERBEROS: Principal name type (name-type[0]) = KRB_NT_SRV_INST
(Service & other unique instance)
              KERBEROS: Principal name value (name-string[1]) =krbtgt/multi
          KERBEROS: Host addresses (addresses[9])
              KERBEROS: Host address =NetBIOS: IIS
******************************************************************************

  FRAME: Base frame properties
  ETHERNET: EType = Internet IP (IPv4)
  IP: Protocol = UDP - User Datagram; Packet ID = 14792; Total IP Length = 1457;
Options = No Options
  UDP: Src Port: Kerberos (88); Dst Port: Unknown (1676); Length = 1437 (0x59D)
  KERBEROS: KRB_AS_REP
      KERBEROS: Client realm (crealm[3]) =MULTI.EXAMPLE.COM
      KERBEROS: Client name (cname[4]) =aaa
          KERBEROS: Principal name type (name-type[0]) = KRB_NT_PRINCIPAL (Name
of Principal)
          KERBEROS: Principal name value (name-string[1]) =aaa
      KERBEROS: Ticket (ticket[5])
          KERBEROS: Realm (realm[1]) =MULTI.EXAMPLE.COM
          KERBEROS: Server name (sname[2]) =krbtgt/MULTI.EXAMPLE.COM
              KERBEROS: Principal name type (name-type[0]) = KRB_NT_SRV_INST
(Service & other unique instance)
              KERBEROS: Principal name value (name-string[1])
=krbtgt/MULTI.EXAMPLE.COM
******************************************************************************

  FRAME: Base frame properties
  ETHERNET: EType = Internet IP (IPv4)
  IP: Protocol = UDP - User Datagram; Packet ID = 7028; Total IP Length = 1436;
Options = No Options
  UDP: Src Port: Unknown (1677); Dst Port: Kerberos (88); Length = 1416 (0x588)
  KERBEROS: KRB_TGS_REQ
      KERBEROS: Pre-authentication Data (padata[3])
```

```
          KERBEROS: Data type = PA-{AP|TGS}-REQ
             KERBEROS: Ticket (ticket[3])
                  KERBEROS: Realm (realm[1]) =MULTI.EXAMPLE.COM
                  KERBEROS: Server name (sname[2]) =krbtgt/MULTI.EXAMPLE.COM
                      KERBEROS: Principal name type (name-type[0]) =
KRB_NT_SRV_INST (Service & other unique instance)
                      KERBEROS: Principal name value (name-string[1])
=krbtgt/MULTI.EXAMPLE.COM
               KERBEROS: Realm (realm[2]) =MULTI.EXAMPLE.COM
          KERBEROS: Server name (sname[3]) =host/iis.multi.example.com
              KERBEROS: Principal name type (name-type[0]) = KRB_NT_SRV_HST (Serv
with host name as instance)
              KERBEROS: Principal name value (name-string[1])
=host/iis.multi.example.com
*******************************************************************************

  FRAME: Base frame properties
  ETHERNET: EType = Internet IP (IPv4)
  IP: Protocol = UDP - User Datagram; Packet ID = 14793; Total IP Length = 1422;
Options = No Options
  UDP: Src Port: Kerberos (88); Dst Port: Unknown (1677); Length = 1402 (0x57A)
  KERBEROS: KRB_TGS_REP
      KERBEROS: Client realm (crealm[3]) =MULTI.EXAMPLE.COM
      KERBEROS: Client name (cname[4]) =aaa
          KERBEROS: Principal name type (name-type[0]) = KRB_NT_PRINCIPAL (Name
of Principal)
          KERBEROS: Principal name value (name-string[1]) =aaa
      KERBEROS: Ticket (ticket[5])
          KERBEROS: Realm (realm[1]) =MULTI.EXAMPLE.COM
          KERBEROS: Server name (sname[2]) =host/iis.multi.example.com
              KERBEROS: Principal name type (name-type[0]) = KRB_NT_SRV_HST (Serv
with host name as instance)
              KERBEROS: Principal name value (name-string[1])
=host/iis.multi.example.com
```

A successful logon will include an initial KRB_AS_REQ and KRB_AS_REP to obtain a TGT. (This only occurs on the first authentication. After the client has a TGT, the protocol will not ask for one again until the TGT expires.) After the AS message exchange, there will be a KRB_TGS_REQ and KRB_TGS_REP for a service ticket to whatever service the client is trying to access. Note that the realm names, the requesting user name, the time, and the SPN can all be viewed in this exchange. This information is often vital in diagnosing problems with Kerberos authentication. The sample packets above have been trimmed to only show the vital data. In a real network capture, there will be much more data displayed, including ticket options and encryption types.

Clock Skew

```
+ FRAME: Base frame properties
+ ETHERNET: EType = Internet IP (IPv4)
```

```
+ IP: Protocol = UDP - User Datagram; Packet ID = 6674; Total IP Length = 333;
Options = No Options
+ UDP: Src Port: Unknown (1550); Dst Port: Kerberos (88); Length = 313 (0x139)
+ KERBEROS: KRB_AS_REQ
*******************************************************************************

+ FRAME: Base frame properties
+ ETHERNET: EType = Internet IP (IPv4)
+ IP: Protocol = UDP - User Datagram; Packet ID = 13387; Total IP Length = 126;
Options = No Options
+ UDP: Src Port: Kerberos (88); Dst Port: Unknown (1550); Length = 106 (0x6A)
  KERBEROS: KRB_ERROR
      KERBEROS: Protocol version number (pvno[0]) = 5 (0x5)
      KERBEROS: Message type (msg-type[1]) = KRB_ERROR (0x1E)
      KERBEROS: Server time (stime[4]) = 7/30/2003 11:54:14 PM
      KERBEROS: Microseconds on server (susec[5]) = 401788 (0x6217C)
      KERBEROS: Error code (error-code[6]) = Clock skew too great
      KERBEROS: Correct realm (realm[9]) =multi
      KERBEROS: Correct server name (sname[10]) =krbtgt/multi
          KERBEROS: Principal name type (name-type[0]) = KRB_NT_SRV_INST (Service
& other unique instance)
          KERBEROS: Principal name value (name-string[1]) =krbtgt/multi
*******************************************************************************

+ FRAME: Base frame properties
+ ETHERNET: EType = Internet IP (IPv4)
+ IP: Protocol = UDP - User Datagram; Packet ID = 6675; Total IP Length = 333;
Options = No Options
+ UDP: Src Port: Unknown (1551); Dst Port: Kerberos (88); Length = 313 (0x139)
+ KERBEROS: KRB_AS_REQ
*******************************************************************************

+ FRAME: Base frame properties
+ ETHERNET: EType = Internet IP (IPv4)
+ IP: Protocol = UDP - User Datagram; Packet ID = 13388; Total IP Length = 1457;
Options = No Options
+ UDP: Src Port: Kerberos (88); Dst Port: Unknown (1551); Length = 1437 (0x59D)
+ KERBEROS: KRB_AS_REP
```

An unsuccessful authentication will usually contain a KRB_ERROR packet explaining what error was encountered. In the trivial example of clock skew, there is a KRB_ERROR packet immediately after the KRB_AS_REQ indicating that there was a problem obtaining the initial TGT. In this case, there is another KRB_AS_REQ and KRB_AS_REP because the client will automatically try to compensate for the clock skew in order to obtain a ticket.

Although it might appear that a TGT was successfully granted because there is a KRB_AS_REP, running Kerberos List will show that no tickets have been issued. This is because the compensation actually failed and the ticket had already expired when the client decrypted it. This is common when the client is very far into the future relative to the domain controller. The

domain controller will issue a ticket with the default expiration time of 10 hours. If the client is more than 10 hours into the future, the ticket will have expired when the client receives it. In this case, as an alternative to doing a Network Monitor capture, examine the system event log (with Kerberos event logging turned on) to determine whether it shows a Kerberos error with the code KRB_AP_ERR_SKEW. For information about how to eliminate time synchronization errors, see Time Synchronization (Clock Skew) earlier in this white paper.

UDP to TCP Failover

```
+ FRAME: Base frame properties
+ ETHERNET: EType = Internet IP (IPv4)
+ IP: Protocol = UDP - User Datagram; Packet ID = 893; Total IP Length = 338;
Options = No Options
+ UDP: Src Port: Unknown (1190); Dst Port: Kerberos (88); Length = 318 (0x13E)
+ KERBEROS: KRB_AS_REQ
*******************************************************************************

+ FRAME: Base frame properties
+ ETHERNET: EType = Internet IP (IPv4)
+ IP: Protocol = UDP - User Datagram; Packet ID = 11914; Total IP Length = 126;
Options = No Options
+ UDP: Src Port: Kerberos (88); Dst Port: Unknown (1190); Length = 106 (0x6A)
  KERBEROS: KRB_ERROR
      KERBEROS: Protocol version number (pvno[0]) = 5 (0x5)
      KERBEROS: Message type (msg-type[1]) = KRB_ERROR (0x1E)
      KERBEROS: Server time (stime[4]) = 7/31/2003 7:26:39 PM
      KERBEROS: Microseconds on server (susec[5]) = 469384 (0x72988)
      KERBEROS: Error code (error-code[6]) = Response too big for UDP, retry with
TCP
      KERBEROS: Correct realm (realm[9]) =multi
    + KERBEROS: Correct server name (sname[10]) =krbtgt/multi
*******************************************************************************

+ FRAME: Base frame properties
+ ETHERNET: EType = Internet IP (IPv4)
+ IP: Protocol = TCP - Transmission Control; Packet ID = 896; Total IP Length =
354; Options = No Options
+ TCP: Control Bits: .AP..., len: 314, seq:3601489557-3601489871, ack: 418744507,
win:17520, src: 1191  dst:  88
+ KERBEROS: KRB_AS_REQ
*******************************************************************************

+ FRAME: Base frame properties
+ ETHERNET: EType = Internet IP (IPv4)
+ IP: Protocol = TCP - Transmission Control; Packet ID = 11916; Total IP Length =
2960; Options = No Options
+ TCP: Control Bits: .A...., len: 2920, seq: 418744507-418747427, ack:3601489871,
win:65221, src:  88  dst: 1191
+ KERBEROS: KRB_AS_REP
*******************************************************************************
```

```
+ FRAME: Base frame properties
+ ETHERNET: EType = Internet IP (IPv4)
+ IP: Protocol = TCP - Transmission Control; Packet ID = 11919; Total IP Length =
4420; Options = No Options
+ TCP: Control Bits: .A...., len: 4380, seq: 418747427-418751807, ack:3601489871,
win:65221, src:  88  dst: 1191
  KERBEROS: Kerberos Packet (Cont.) Use the Coalescer to view contents
*******************************************************************************

+ FRAME: Base frame properties
+ ETHERNET: EType = Internet IP (IPv4)
+ IP: Protocol = TCP - Transmission Control; Packet ID = 11923; Total IP Length =
631; Options = No Options
+ TCP: Control Bits: .AP..., len: 591, seq: 418751807-418752398, ack:3601489871,
win:65221, src:  88  dst: 1191
  KERBEROS: Kerberos Packet (Cont.) Use the Coalescer to view contents
*******************************************************************************

+ FRAME: Base frame properties
+ ETHERNET: EType = Internet IP (IPv4)
+ IP: Protocol = TCP - Transmission Control; Packet ID = 903; Total IP Length =
1500; Options = No Options
+ TCP: Control Bits: .A...., len: 1460, seq:1499456721-1499458181,
ack:3422954122, win:17520, src: 1192  dst:  88
+ KERBEROS: KRB_TGS_REQ
*******************************************************************************

+ FRAME: Base frame properties
+ ETHERNET: EType = Internet IP (IPv4)
+ IP: Protocol = TCP - Transmission Control; Packet ID = 904; Total IP Length =
1500; Options = No Options
+ TCP: Control Bits: .A...., len: 1460, seq:1499458181-1499459641,
ack:3422954122, win:17520, src: 1192  dst:  88
  KERBEROS: Kerberos Packet (Cont.) Use the Coalescer to view contents
*******************************************************************************

+ FRAME: Base frame properties
+ ETHERNET: EType = Internet IP (IPv4)
+ IP: Protocol = TCP - Transmission Control; Packet ID = 906; Total IP Length =
1500; Options = No Options
+ TCP: Control Bits: .A...., len: 1460, seq:1499459641-1499461101,
ack:3422954122, win:17520, src: 1192  dst:  88
  KERBEROS: Kerberos Packet (Cont.) Use the Coalescer to view contents
*******************************************************************************

+ FRAME: Base frame properties
+ ETHERNET: EType = Internet IP (IPv4)
+ IP: Protocol = TCP - Transmission Control; Packet ID = 907; Total IP Length =
1500; Options = No Options
```

```
+ TCP: Control Bits: .A...., len: 1460, seq:1499461101-1499462561,
ack:3422954122, win:17520, src: 1192  dst:  88
  KERBEROS: Kerberos Packet (Cont.) Use the Coalescer to view contents
```

This is a common example of the error that occurs when the Kerberos protocol attempts to switch to TCP. The RFC states that the Kerberos protocol should use UDP for transmitting data. However, UDP has a practical limit on how large a packet might be sent across the network. Because Microsoft extensions to the Kerberos protocol place group memberships within a ticket, it is common for a ticket issued to a user that belongs to many groups to be larger than the maximum value that UDP can reliably transmit. If this is what happens, a KRB_ERR_RESPONSE_TOO_BIG is almost always present.

In the system event log (with Kerberos event logging turned on), a Kerberos error will appear, the details of which will state that a KRB_ERR_RESPONSE_TOO_BIG was received. In the example above, there is a KRB_ERROR packet after the initial KRB_AS_REQ with the same failure code in it. Also, note that the initial KRB_AS_REQ and KRB_ERROR occur over UDP, but the rest of the traffic occurs over TCP, indicating that the switchover was successful and the authentication continued normally.

Furthermore, note that the subsequent KRB_AS_REP takes multiple packets. This is because there are many groups to be transmitted and they do not all fit into one packet. Thus, multiple packets are required to transmit the information, something that could not have been accomplished reliably with UDP.

UDP Fragmentation

```
+ FRAME: Base frame properties
+ ETHERNET:  EType = Internet IP (IPv4)
+ IP: Protocol = UDP - User Datagram; Packet ID = 39863; Total IP Length = 1500;
Options = No Options
  UDP: Src Port: Unknown (3010); Dst Port: Kerberos (88); Length = 1798 (0x706)
      UDP: Source Port = 0x0BC2
      UDP: Destination Port = Kerberos
      UDP: Total length = 1798 (0x706)
      UDP: UDP Checksum = Frame was truncated, unable to verify Checksum.
  KERBEROS: KRB_TGS_REQ
      KERBEROS: Protocol version number (pvno[1]) = 5 (0x5)
      KERBEROS: Message type (msg-type[2]) = KRB_TGS_REQ (0x0C)
      KERBEROS: Pre-authentication Data (padata[3])
          KERBEROS: Data type = PA-{AP|TGS}-REQ
              KERBEROS: Protocol version numer (pvno[0]) = 5 (0x5)
              KERBEROS: Message type (msg-type[1]) = 14 (0xE)
            + KERBEROS: AP options (ap-options[2])
              KERBEROS: Ticket (ticket[3])
                  KERBEROS: Ticket version number (tkt-vno[0]) = 5 (0x5)
                  KERBEROS: Realm (realm[1]) =EXAMPLE.COM
                + KERBEROS: Server name (sname[2]) =krbtgt/EXAMPLE.COM
                + KERBEROS: Encrypted part (enc-part[3])
*******************************************************************************
```

```
+ FRAME: Base frame properties
+ ETHERNET:  EType = Internet IP (IPv4)
+ IP: Protocol = UDP - User Datagram; Packet ID = 32022; Total IP Length = 134;
Options = No Options
  UDP: Src Port: Kerberos (88); Dst Port: Unknown (3010); Length = 114 (0x72)
      UDP: Source Port = Kerberos
      UDP: Destination Port = 0x0BC2
      UDP: Total length = 114 (0x72)
      UDP: UDP Checksum = 0x5B13
  KERBEROS: KRB_ERROR
      KERBEROS: Protocol version number (pvno[0]) = 5 (0x5)
      KERBEROS: Message type (msg-type[1]) = KRB_ERROR (0x1E)
      KERBEROS: Server time (stime[4]) = 1/6/2003 6:26:04 PM
      KERBEROS: Microseconds on server (susec[5]) = 665608 (0xA2808)
      KERBEROS: Error code (error-code[6]) = Server not found in Kerberos
database
      KERBEROS: Correct realm (realm[9]) =EXAMPLE.COM
      KERBEROS: Correct server name (sname[10]) =krbtgt/EXAMPLE.COM
          KERBEROS: Principal name type (name-type[0]) = KRB_NT_SRV_INST (Service
& other unique instance)
          KERBEROS: Principal name value (name-string[1]) =krbtgt/EXAMPLE.COM
```

At first glance, the KRB_ERROR packet might seem to indicate the source of the problem. However, upon closer inspection of the UDP header portion of each packet, you can verify that KRB_TGS_REQ has been truncated during transmission. This means that there is no guarantee that the packet received was complete or correct. The KRB_ERROR packet most likely stems from the fact that, because part of the request was lost during transmission, the server could not be located (that is, part of the data required to locate it was missing). In this case, the KRB_ERROR is misleading and the real source of the problem is UDP fragmentation. For more information about how to resolve this type of error, see UDP Fragmentation earlier in this white paper.

Related Information

- "Answers to Frequently Asked Kerberos Questions" in the Microsoft Knowledge Base at http://go.microsoft.com/fwlink/?LinkId=25039
- "Authentication for Administrative Authority" on Microsoft TechNet at http://go.microsoft.com/fwlink/?LinkId=25038
- "Building Security and Directory Solutions for UNIX Using the Windows Server 2003 Active Directory Kerberos and LDAP Services" in the Solution Guide for Windows Security and Directory Services for UNIX on the Microsoft Download Center at http://go.microsoft.com/fwlink/?LinkId=25395
- Windows Server 2003 Technical Reference on Microsoft TechNet at http://go.microsoft.com/fwlink/?LinkId=21711

Acknowledgements

Vincent Abella, Technical Editor, Microsoft Corporation

Leon Arber, University of Illinois at Urbana-Champaign

David Christiansen, Software Design Engineer, Microsoft Corporation

Mike Danseglio, Technical Writer, Microsoft Corporation

Xin Fan, Software Test Engineer, Microsoft Corporation

JK Jaganathan, Program Manager, Microsoft Corporation

Steve Light, Escalation Engineer, Microsoft Corporation

David Longmuir, Technical Editor, Volt

Soumitra Sengupta, Architect, Microsoft Corporation

Michiko Short, Technical Writer, Microsoft Corporation

Tim Springston, Support Professional, Microsoft Corporation

Todd Stecher, Development Lead, Microsoft Corporation

Jonathan Stephens, Escalation Engineer, Microsoft Corporation

Darol Timberlake, Consultant, Microsoft Corporation

Joseph Vasil, Consultant, Microsoft Corporation

Liqiang (Larry) Zhu, Software Design Engineer, Microsoft Corporation