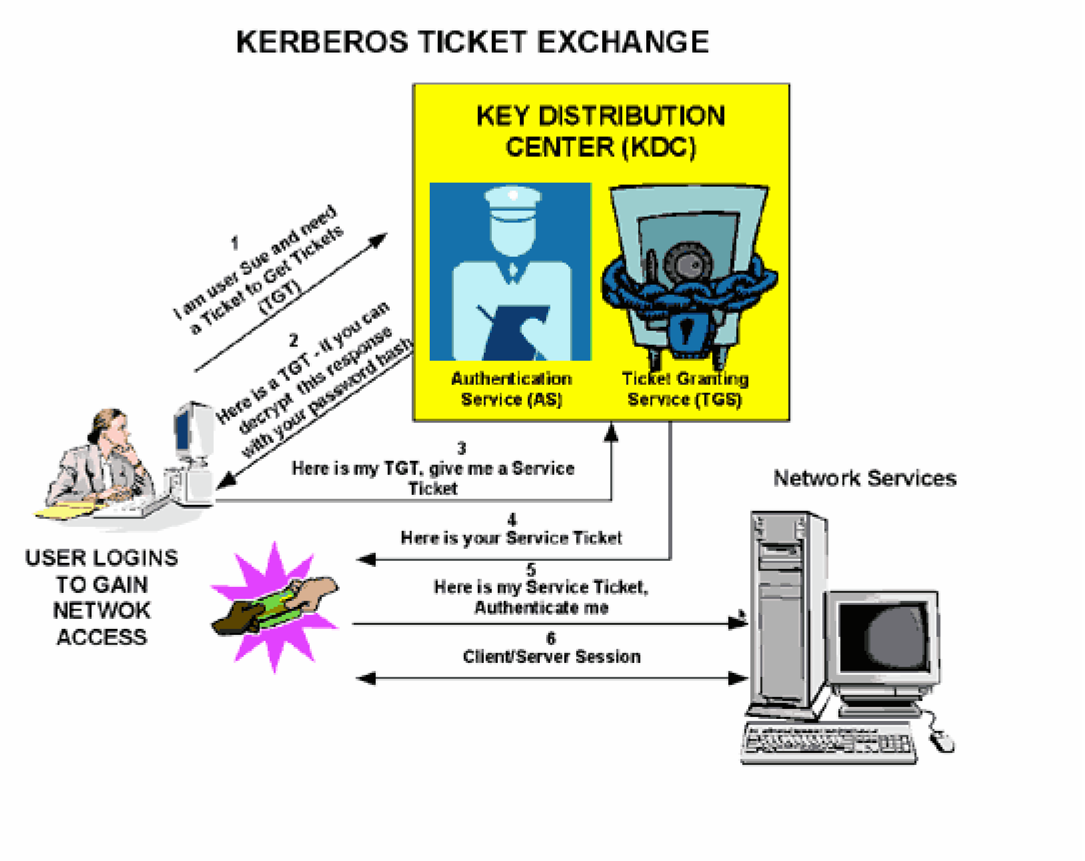

KERBEROS TICKET EXCHANGE
KEY DISTRIBUTION CENTER (KDC)
Authentication Service (AS)
Ticket Granting Service (TGS)
1 I am user Sue and need a Ticket to Get Tickets (TGT)
2 Here is a TGT - if you can decrypt this response with your password hash
3 Here is my TGT, give me a Service Ticket
4 Here is your Service Ticket
5 Here is my Service Ticket, Authenticate me
6 Client/Server Session
Network Services
USER LOGINS TO GAIN NETWOK ACCESS

Understanding Kerberos concepts

Kerberos Version 5 is standard on all versions of Windows 2000 and ensures the highest level of security to network resources. The Kerberos protocol name is based on the three-headed dog figure from Greek mythology known as Kerberos. The three heads of Kerberos comprise the Key Distribution Center (KDC), the client user and the server with the desired service to access. The KDC is installed as part of the domain controller and performs two service functions: the Authentication Service (AS) and the Ticket-Granting Service (TGS). As exemplified in Figure 1, three exchanges are involved when the client initially accesses a server resource:

1.AS Exchange
2.TGS Exchange
3.Client/Server (CS) Exchange

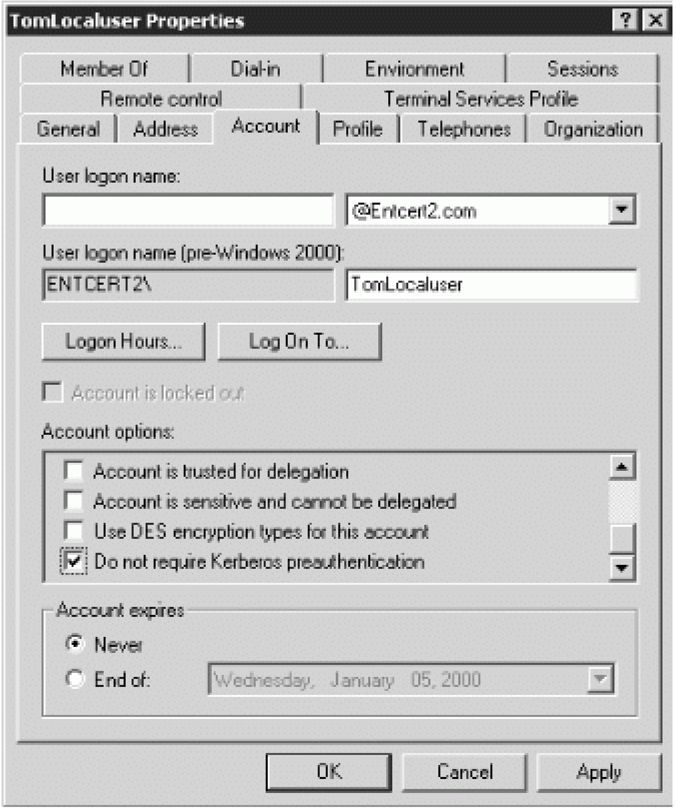
Let's take a closer look at this exchange process and its component parts.

AS Exchange

When initially logging on to a network, users must negotiate access by providing a log-in name and password in order to be verified by the AS portion of a KDC within their domain. The KDC has access to Active Directory user account information. Once successfully authenticated, the user is granted a Ticket to Get Tickets (TGT) that is valid for the local domain. The TGT has a default lifetime of 10 hours and may be renewed throughout the user's log-on session without requiring the user to re-enter his password. The TGT is cached on the local machine in volatile memory space and used to request sessions with services throughout the network. The following is a discussion of the TGT retrieval process.

Example AS Administration

The AS request identifies the client to the KDC in plain text. If preauthentication is enabled, a time stamp will be encrypted using the user's password hash as an encryption key. If the KDC reads a valid time when using the user's password hash (stored in the Active Directory) to decrypt the time stamp, the KDC knows that request isn't a replay of a previous request. The preauthentication feature may be disabled for specific users in order to support some applications that don't support the security feature. Access the user account from the Active Directory users and the computers will snap-in and select the account tab. From the account options: slide window, check mark the "Do not require Kerberos" preauthentication option (Figure 2).



User logon name:

@Entcert2.com

User logon name (pre-Windows 2000):

ENTCERT2\ TomLocaluser

☐ Account is locked out

Account options:

☐ Account is trusted for delegation
☐ Account is sensitive and cannot be delegated
☐ Use DES encryption types for this account
☑ Do not require Kerberos preauthentication

Account expires

● Never
○ End of: Wednesday, January 05, 2000

Figure 2: Disable Kerberos Preauthentication
See full-sized image.

If the KDC approves the client's request for a TGT, the reply (referred to as the AS reply) will include two sections: a TGT encrypted with a key that only the KDC (TGS) can decrypt and a session key encrypted with the user's password hash to handle future communications with the KDC. Because the client system cannot read the TGT contents, it must blindly present the ticket to the TGS for service tickets. The TGT includes time to live parameters, authorization data, a session key to use when communicating with the client and the client's name.

TGS Exchange

The user presents the TGT to the TGS portion of the KDC when desiring access to a server service. The TGS on the KDC authenticates the user's TGT and creates a ticket and session key for both the client and the remote server. This information, known as the service ticket, is then cached locally on the client machine.

The TGS receives the client's TGT and reads it using its own key. If the TGS approves of the client's request, a service ticket is generated for both the client and the target server. The client reads its portion using the TGS session key retrieved earlier from the AS reply. The client presents the server portion of the TGS reply to the target server in the client/server exchange coming next.

Client/Server Exchange

Once the client user has the client/server service ticket, he can establish the session with the server service. The server can decrypt the information coming indirectly from the TGS using its own long-term key with the KDC. The service ticket is then used to authenticate the client user and establish a service session between the server and client. After the ticket's lifetime is exceeded, the service ticket must be renewed to use the service.

Client/Server Exchange Detail

The client blindly passes the server portion of the service ticket to the server in the client/server request to establish a client/server session. If mutual authentication is enabled, the target server returns a time stamp encrypted using the service ticket session key. If the time stamp decrypts correctly, not only has the client authenticated himself to the server, but the server also has authenticated itself to the client. The target server never has to directly communicate with the KDC. This reduces downtime and pressure on the KDC.

Further Clarification of the Log-in Process

A TGT and a service ticket are needed to access services on remote computers, but they are also required to successfully log on to a local system. When the log-on window appears, password encryption using a one-way hash algorithm occurs immediately and negotiations commence with the KDC for a valid TGT and service ticket. The process is the same as accessing a remote service. An access token is created for the user containing all security groups to which they belong. This access token is attached to the user's log-on session and is subsequently inherited by any process or application the user starts.