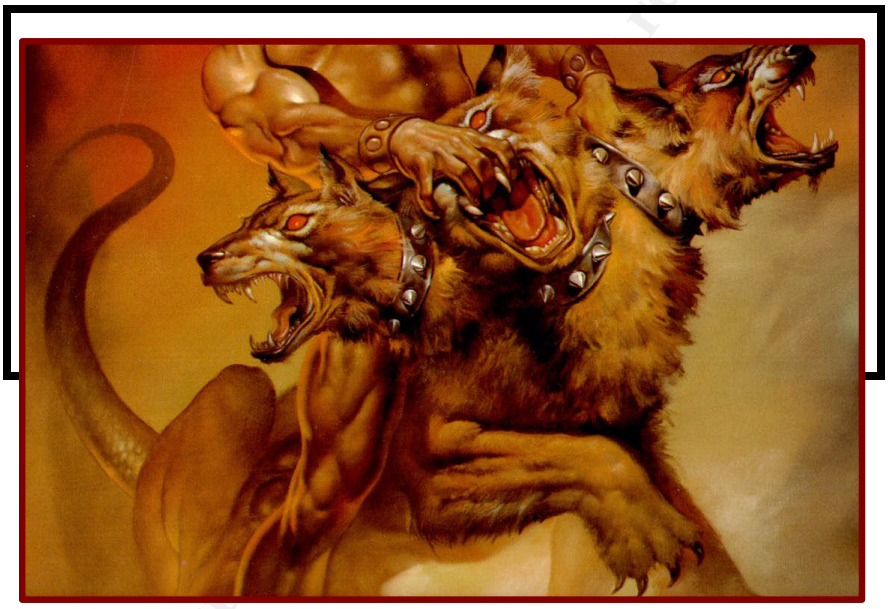
Global Information Assurance Certification Paper

SANS
Global Information Assurance Certification (GIAC) Program

Securing Windows
GCNT Practical Assignment
Version 2.1b



This Page Intentionally Left Blank

This Page Intentionally Left Blank

TABLE OF CONTENTS

LIST OF FIGURES

Understanding Kerberos Authentication in Windows 2000

I. INTRODUCTION

The purpose of this paper is to fulfill requirements for the Global Information Assurance Certification for Windows NT (GIAC-NT). This paper examines the Kerberos authentication protocol and how it is implemented in Windows 2000 Security Services.

The scope of this paper is to provide the reader with a greater understanding of Kerberos and authentication methods used in a Windows 2000 native environment. While many details provided pertain to non-Windows 2000 systems, the content is strictly aimed at Windows 2000 Kerberos realms and Windows 2000 systems.

Microsoft began developing Windows 2000 in the late 1990's as a follow-on to Microsoft Windows NT 4.0. With the introduction of Windows 2000, Microsoft delivered, for the most part, a secure "out of the box" network operating system with the capability to manage and configure network security best suited to the customer's requirements.

Windows 2000 supports several core authentication protocols: Windows NT LAN Manager (NTLM), Secure Sockets Layer (SSL), Distributed Password Authentication (DPA) and Kerberos version 5. Kerberos was chosen by Microsoft as the core security protocol for Windows 2000. This paper focuses on Microsoft's implementation of Kerberos and details the basic concepts, components, and configuration of Kerberos in a Windows 2000 network. The benefits of using tools and utilities for Kerberos implementation are also discussed.

II. BACKGROUND

Authentication is the process of verifying the identity of a user that is logging onto a computer system or verifying the integrity of a transmitted message or object. The goal of authentication is to verify that the user is actually who they claim to be. Windows 2000 authentication is based on *transitive trusts*. Transitive trust refers to authentication across a chain of trust relationships. In Windows 2000, trust relationships support authentication across domains by using the Kerberos v5 protocol and NTLM authentication. [Ref. 1]

While Windows 2000 supports several core authentication protocols, the two most prevalent methods of authentication utilized are NTLM and Kerberos. This section briefly introduces Windows 2000 security components that are essential to the authentication process and provides a basic understanding of NTLM and Kerberos.

A. Windows 2000 Security Components

The following is a brief description of the security components essential to secure authentication in Windows 2000:

- **Security Reference Monitor (SRM)** – The SRM performs security access checks, adjusts privileges, and generates audit messages when necessary.

- **Local Security Authority (LSA)** – The LSA ensures that the user has permission to access the system and enforces the local security policy. It generates access tokens, manages the local security policy, and provides interactive user authentication services. The LSA also processes logons and audits and works with the SRM.

- **Logon Process (WINLOGON.EXE)** – A security service running in a process it shares with the LSA and acts as a guard in the authentication process. Winlogon displays a dialog box that asks for the account information (*username and password*) and security authority (*the domain*) that issued it, and passes the data to the LSA for verification.

- **Graphical Identification and Authentication (GINA)** – Applies the user interface portion of providing the logon credentials.

- **Network Logon Service** – A user-mode process that responds to network logon requests.

- **Security Packages** – Provide security services to the system. Packages include Kerberos, MSVL_0 (used by NTLM), and Schannel (used with SSL and PKI).

- **Security Support Provider (SSP)** – A dynamic-link library supplied with Windows 2000.

- **Security Support Provider Interface (SSPI)** – A Win32 Application Program Interface (API) that provides the interface between applications and the security packages.

- **Security Account Manager (SAM)** – The database where local account and policy information is stored. [Ref. 2]

B. NT LAN Manager (NTLM)

NTLM is used by Windows NT 3.x-4.0 (referred to as *downlevel* domains) and remains

largely proprietary. Since only Windows 2000 and UNIX clients can utilize Kerberos authentication in a Windows 2000 domain, Microsoft continues to support NTLM to provide access to Windows NT 4.0 and Windows 9x clients. The latest version is NTLM version 2 and is implemented in Windows 2000. NTLMv2 was originally developed for Windows NT 4.0 Service Pack 4, and is enabled by installing the Directory Service Client (***dsclient.exe***, included on the Windows 2000 Server CD) on downlevel clients. NTLMv2 is also used to authenticate logons to Windows 2000 computers that are not participating in a domain. It offers 128-bit security that reduces the possibility of unauthorized access. NTLM is implemented in Windows 2000 as a Security Support Provider (SSP). [Ref. 3]

C. Kerberos

Kerberos replaces NTLM as the primary security protocol for access to resources within or across Windows 2000 domains. Kerberos was developed at The Massachusetts Institute of Technology (MIT) in the 1980's and was named after the three-headed mythological Greek three-headed dog ***Cerberus*** whom guarded the gates of Hades (not implied to be the network you administer). It offers a three-sided (hence the name *Cerberus!*) authentication process with shared-secret keys that enable users to prove their identity.

D. Benefits of Kerberos

Kerberos is more flexible, efficient, and more secure than NTLM. The Kerberos authentication protocol augments the built-in security features of Windows 2000 with the following enhancements:

- ***Faster server authentication performance*** during initial connection establishment. The application server does not have to connect to the domain controller to authenticate the client, allowing application servers to efficiently manage large numbers of client connection requests.

- ***Mutual Authentication.*** Think of this as a two-way authentication. The client authenticates to the requested server, and the server authenticates to the client. Both the client and the server know that the party at the other end of the network is who they claim to be.

- ***Delegation of authentication*** for multitier client/server application architectures. When a client connects to a server, the server impersonates the client on that system. But if the server needs to make a network connection to complete the client transaction, Kerberos allows delegation of authentication for the first server to connect on the client's behalf to another server, and allows the second server to also impersonate the client.

- ***Transitive trust relationships*** for interdomain authentication. Users can

authenticate to domains anywhere in the domain tree because the KDCs in each domain trusts tickets issued by other KDCs in the tree. Transitive trust simplifies domain management for large networks.

- ***Single Sign-On (SSO)*.** Kerberos provide the SSO capability that allows network clients to seamlessly access all authorized network resources on the basis of a single authentication process. [Ref. 5]

- ***Interoperability*.** Microsoft's implementation of Kerberos in Windows 2000 closely follows the MIT release of Kerberos v5, which is on a standards track with the Internet Engineering Task Force (IETF), and is compliant with Request for Comment (RFC) 1510 and RFC 1964. [Ref. 2, 6]

III. AUTHENTICATION USING THE KERBEROS PROTOCOL

Kerberos is an Internet standard for authentication and defines the interactions between a client and a network authentication service. In Windows 2000, Kerberos plays an integral part in user authentication and relies heavily on the concept of shared secrets. These techniques will be discussed in more detail later in this paper.

A. Basic Concepts of Kerberos

The Windows 2000 implementation of a network authentication service is known as a ***Key Distribution Center (KDC)***. The Kerberos protocol is based upon the KDC and *tickets.* Tickets are encrypted data packets issued by the KDC. A ticket vouches for a user's identity as well as carrying other information. A KDC provides tickets for all users within its area of authority (normally a domain, or what Kerberos calls a ***Realm***). [Ref. 4]

There are a few things to remember about Windows 2000 Kerberos Realms. All Windows 2000 domains are also Kerberos Realms. And since Windows 2000 domain names are also DNS domain names, the Kerberos Realm has the same name. To differentiate between the two, Kerberos Realms are always in uppercase. However, this only affects interoperability with other non-Windows Kerberos-based environments. Another important item to remember is that every domain controller in a Windows 2000 domain is a KDC. Each domain controller is therefore enabled to distribute session keys to authenticated users.

The Kerberos protocol defines a series of exchanges between clients, the KDC, and servers to obtain and use Kerberos tickets. Kerberos security is based on two fundamental concepts:

- **Shared Secrets (Keys) –** The user and the KDC share the same secret key (normally the user's password).

- **Three-sided (Multiple) Authentication –** The authentication process involves

three components:

- The ***client***, which represents the user, or application.
- The ***network resource*** that wants to ensure the client is legitimate, often a server.
- A ***KDC***, which serves as a central repository for client information and issues tickets. [Ref. 2]

Shared Secrets. Kerberos is a shared-secret authentication protocol because both the user (client) and the KDC know the user's secret. Kerberos uses the shared-key concept by implementing secret key cryptography. Instead of sharing a password, the client and KDC share a cryptographic key, and use their mutual knowledge of the secret key to verify each other's identity. Kerberos keys are symmetric – that is, each shared secret key is capable of both encryption and decryption. One party proves knowledge of the key by encrypting part of the secret and the other party decrypts it. Many articles on Kerberos referenced the well-known (and well-used) computer users Bob and Alice, which this paper illustrates in Figure 1.

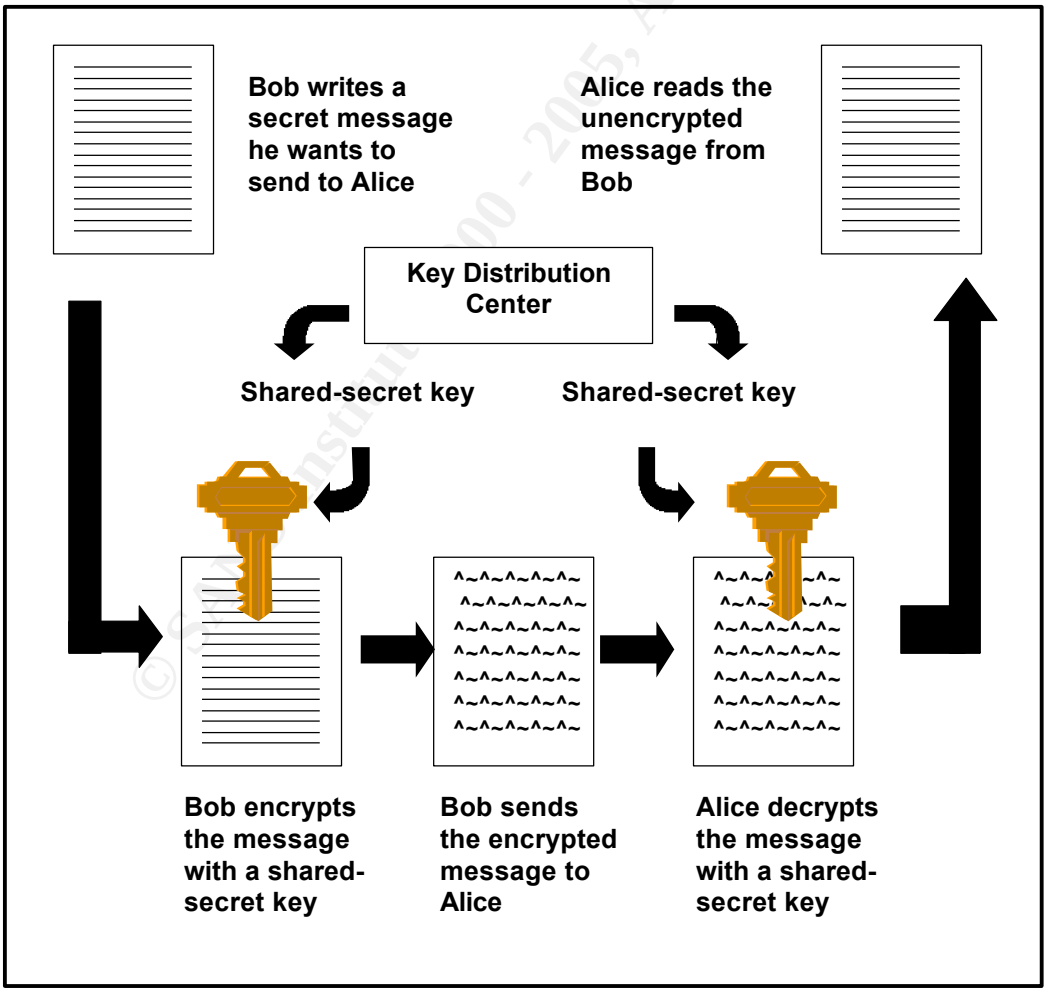


Figure 1. Shared Secret Keys.

Suppose Bob wants to send Alice a secret message. Bob uses a shared-secret key to encrypt it, and then sends it to her. Once Alice receives the encrypted data, she uses a copy of the same shared-secret key to decrypt it. The concept is that shared-secret keys can be distributed by a trusted source (the KDC) so others can use it to encrypt data to send to you that you can decrypt with your copy of the shared-secret key.

This simple explanation does not detail how the secret key was distributed; rather it is only used to explain the concept of shared secret keys. The following sections discuss the mechanics of how keys are stored and distributed in a Windows 2000 environment.

Multiple Authentication. Kerberos uses three exchanges of information to securely authenticate when a client initially accesses a network resource: the Authentication Service, the Ticket-Granting Service, and the Client/Server exchanges. Each of these services is addressed in detail in the next section.

B. Kerberos Components

Now that we have a basic understanding of Kerberos authentication, let's look at the key components of Kerberos authentication. These components include the KDC, account database, Kerberos SSP, and DNS name resolution.

1. Key Distribution Center (KDC)

The KDC is a trusted authority that issues tickets. Tickets are encrypted data packets (the secret key mentioned above) used as a voucher for a user's identity. The KDC provides tickets for all users in its realm (remember, a Kerberos realm corresponds to a domain). In Windows 2000, every domain controller is a KDC. The KDC is the main component of Kerberos authentication as it links the other components together into a secure authentication process. It utilizes Active Directory as the account database (and also gets some user information from the Global Catalog).

Windows 2000 implements the KDC as a domain service. The KDC service is started automatically and cannot be stopped. Two important services running under the single KDC process are:

- **Authentication Service (AS)** – The AS provides the initial authentication of the user on the network and issues the Ticket Granting Tickets (TGTs). A TGT is used to request a ticket that will establish an encrypted session between the client and a server. Before users can receive tickets for network services, they must first get an initial TGT from the AS which is then presented to the KDC to validate their authentication.

- **Ticket-Granting Service (TGS)** – The TGS issues session tickets good for

access to domain resources based on the users rights and permissions, or to the TGS of other trusted domains. When a user (client) requests a domain service, such as a printer, they must contact the TGS in the user's domain, present a TGT (issued by the AS), and request a ticket. [Ref. 2,6]

Both services are started automatically by the Local Security Authority (LSA) on each domain controller and run in the process space of the LSA. The security principal name (userid which uniquely identifies users) used by the KDC is ***krbtgt***. The krbtgt (which stands for *Kerberos Ticket Granting Ticket*) is unique among all of the built-in user accounts. It is always disabled and cannot be enabled (See Figure 2).

The krbtgt account is created automatically with the creation of a new domain. A password is automatically assigned, changed on a regular basis, and is used to create the secret key for encrypting and decrypting the TGTs issued by the AS. When the client is given a TGT, the TGT is encrypted with the KDC's secret key. The krbtgt is the secret key by which the domain TGTs are encrypted and decrypted. Remember, the krbtgt is a domain account and therefore is present on every domain controller in the domain. [Ref. 6, 7]

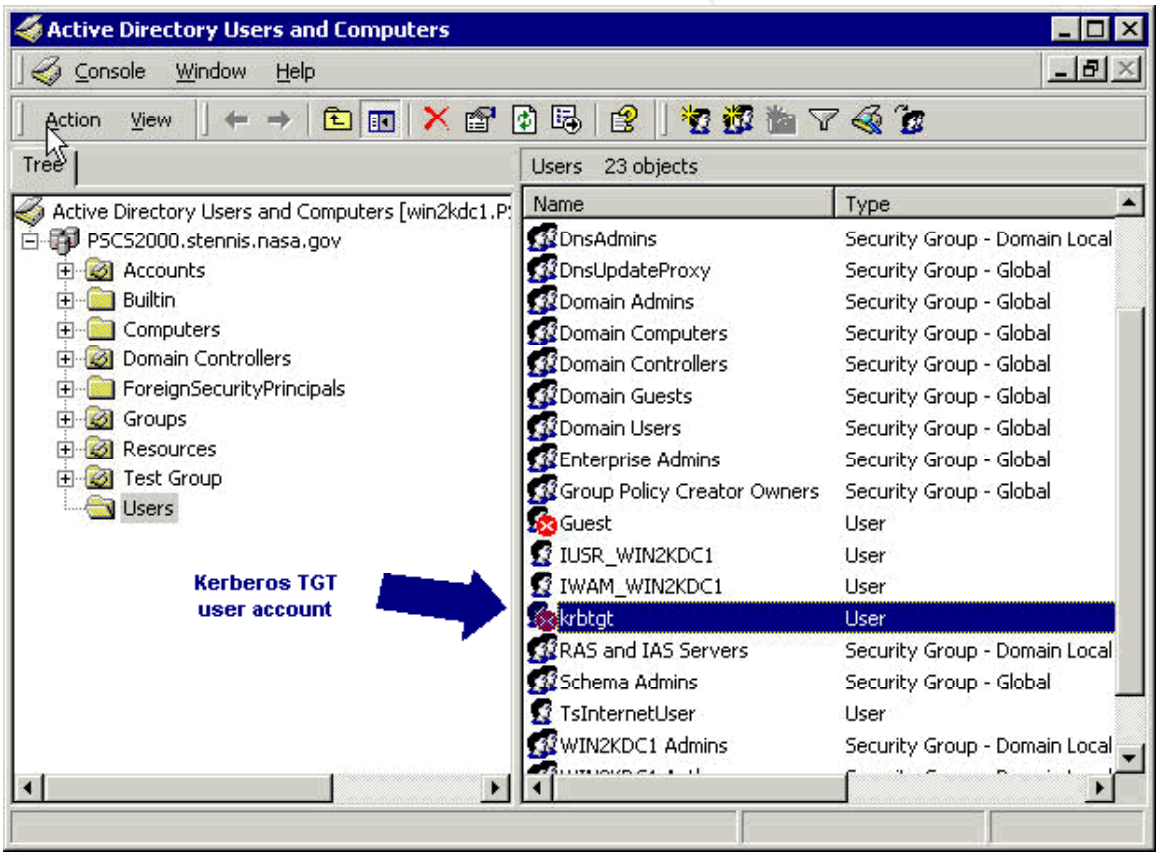


Figure 2. Krbtgt user account in Active Directory.

2. Account Database

The Active Directory service provides the account database used by Kerberos to obtain information about each security principal. An account object in Active Directory represents each principal, and the encryption key utilized by the user to access other resources is stored as an attribute of the account object. It is important to note that only the account holder has access to the account object's password attribute. [Ref. 6]

3. Kerberos Security Support Provider (SSP)

Kerberos is implemented as an SSP in Windows 2000. The Kerberos SSP is loaded by the LSA at system boot up. The NTLM SSP is also loaded at boot up and can be used for authentication, but the Kerberos SSP is ***always*** the first choice. Which SSP is used depends on the capabilities of the client computer (i.e.; non-Kerberos capable clients would use NTLM).

Systems services and transport-level applications access SSPs via the Microsoft Security Support Provider Interface (SSPI). All distributive services in Windows 2000 use SSPI to access Kerberos SSP. [Ref. 2,6]

4. DNS Name Resolution

Microsoft's implementation of Kerberos in Windows 2000 is compliant with Request for Comment (RFC) 1510, which specifies that IP transport should be used for all client/KDC messaging. The Kerberos SSP on a client computer utilizes DNS to resolve the server name (the domain controller and the KDC service) to an IP address. Suppose a Windows 2000 computer is operating in a Kerberos realm that is not part of a Windows 2000 domain. In this case, the KDC will not be a service on a domain controller, so the servers running the KDC must be stored in the client computer's Registry. [Ref. 6]

5. Time Synchronization

Windows 2000 utilizes time synchronization in Kerberos transactions as a deterrent to replay attacks. The current time of the client is included in all client requests sent to network servers or the KDC. The client computer time is compared to the target server's current time. By default, if the difference between the two times is greater than five minutes the connection is not validated. Windows 2000 utilizes the Windows Time Synchronization Service (W32Time.exe) to ensure all computers in the domain (and the forest) have synchronized clocks. The Time Synchronization Service uses the Simple Network Time Protocol (SNTP).

The Primary Domain Controller (PDC) emulator in the forest root domain is the "keeper

of the clock," and is considered the authoritative time source for the entire forest. Each PDC emulator in a domain contacts the forest root domain PDC emulator for clock synchronization. Then all domain controllers synchronize their clocks with the PDC emulator of their domain. Finally, all client computers in the domain synchronize their clocks with the authenticating domain controller. Client clocks are reset to match the domain controller's time if the difference is more than two minutes. [Ref. 3]

C. How Kerberos Works

We have discussed the basic concepts and components of the Kerberos protocol. This section provides details on how Kerberos authentication works. As stated earlier, the KDC performs two service functions: The Authentication Service (AS) and the Ticket-Granting Service (TGT). Three exchanges are involved when a client initially authenticates with the network and a network resource for the first time:

- AS Exchange
- TGS Exchange
- Client/Server (CS) Exchange [Ref. 8]

Figure 3 (below) illustrates the message exchanges (sometimes referred to as *sub protocols*) that take place during the initial authentication of a user and request for a network resource. Each step is critical in establishing a secure network authentication and is outlined below:

- **Step 1**. The user presses Ctrl-Alt-Del to display the login screen, and provides a username, password, and domain so the information can be verified by the AS portion of a KDC in the domain. The client sends a Kerberos Authentication Service Request (**KRB_AS_REQ**) to the KDC. The KRB_AS_REQ contains the user's account information and the client clock time and is encrypted with the user's long-term key (**Note:** Remember the long-term key is a shared-secret key between the user and the KDC. See part III A. for details). The KDC accesses Active Directory to get user account information. If the user information is verified in Active Directory, the user is authenticated.

- **Step 2**. Once the KDC approves the user's request for a TGT, the AS generates a TGT and sends the TGT to the user in a Kerberos Authentication Service Response (**KRB_AS_REP**) message. The KRB_AS_REP includes two sections:
    - The **TGT** encrypted with a key only the TGS service of the KDC process can decrypt.
    - The **session key** encrypted with the user's long-term key to manage future communications with the KDC. Each user shares a long-term key with the KDC. [Ref. 3,8]

At this point of the authentication process the user is authenticated on the local

domain. The user's TGT is cached locally on the client machine and will be used to request network resources. The TGT includes time to live (TTL) parameters, authorization data, a session key, and the user's name. The user needs to acquire a service ticket (ST) to access network resources. The following steps occur when a user requests access to a server, or network resource, such as a printer, application, file share on a server, etc., noted as a ***target server***. **Note:** Each time a user makes a request for a new network resource or service on a remote server, he will perform steps 3-6 for that new request.

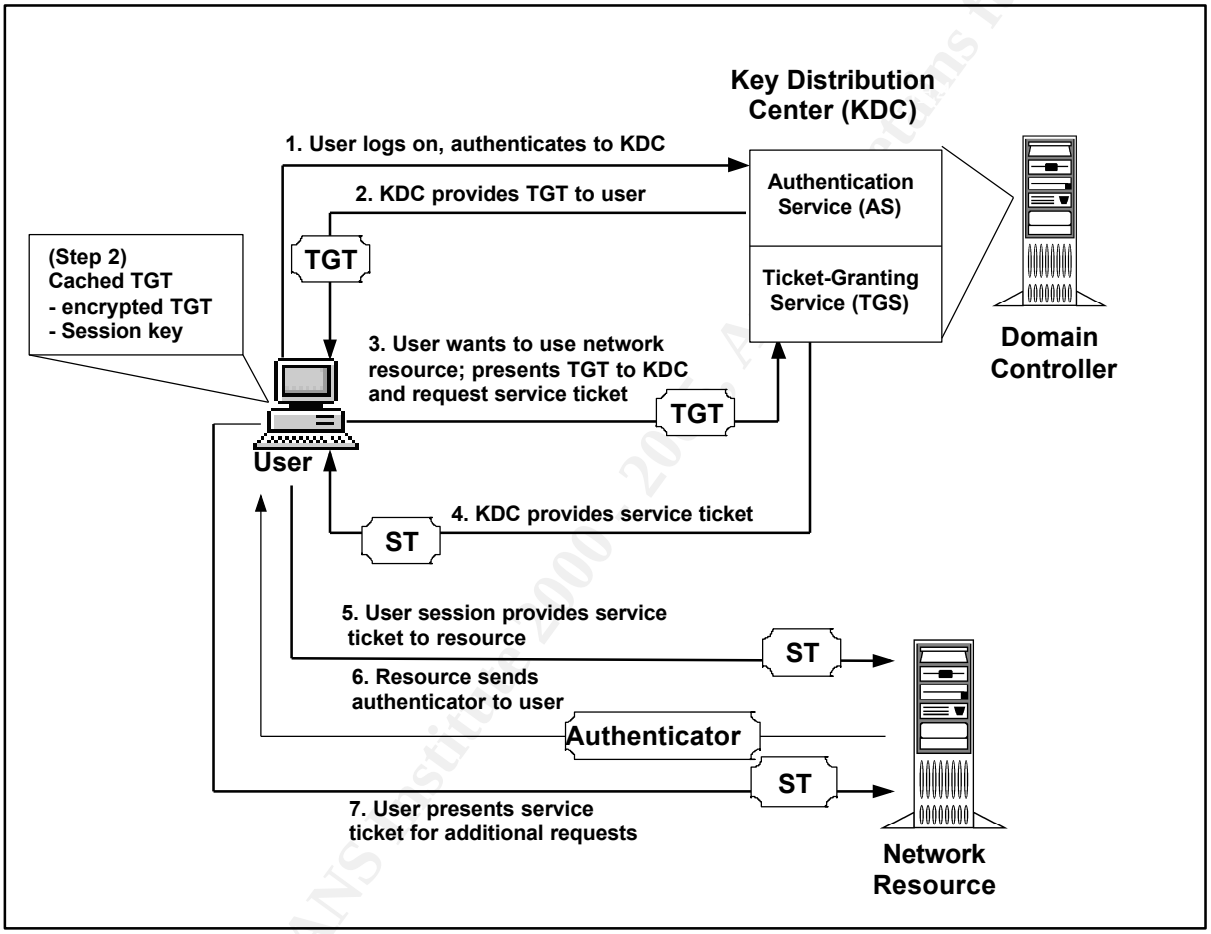


Figure 3. Authentication using the Kerberos Protocol. After Ref. [5]

- **Step 3**. The user sends a Ticket Granting Service Exchange Request (**KRB_TGS_REQ**) to the TGS to request an ST for access to the target server. The KRB_TGS_REQ includes the TGT and an authenticator. The client system cannot read the contents of the TGT so it must unquestioningly present the ticket to the TGS to receive STs.

- **Step 4**. The TGS authenticates the user's TGT using its own key and generates a new ST. The ST is sent back to the user in the Kerberos Ticket Granting

Service Response (**KRB_TGS_REP**). The ST is encrypted using the long-term key between the KDC and the target server. The ST is generated based on the user's right and permissions to the target server. The client reads the user session key using the TGS session key retrieved from **step 2**.

- **Step 5**. The client initiates a client/server session by blindly passing the server portion of the ST to the target server using a Kerberos Application Request (**KRB_AP_REQ**).
- **Step 6**. The target server can decrypt the ST using its long-term key (which it shares with the KDC), then authenticate the user and establish the client/server session. The target server sends an authenticator back to the user in a Kerberos Application Response (**KRB_AP_REP**) that provides mutual authentication of the user and the target server. The ST must be renewed once its lifetime is exceeded.

- **Step 7**. As long as the ST is valid the user can present the ST for additional access to the resource without contacting the KDC. The server never has to establish a session with the KDC, but instead receives the authentication information indirectly from the client via the server session key generated in **step 4**. [Ref. 3,8]

D. Single Sign-On

While this Kerberos process may seem cumbersome and full of overhead, it actually improves network performance. In NTLM, a domain controller has to vouch a client's identity every time he requests a network resource. This means that each time a user needs to print to a domain printer, the domain controller is involved. Kerberos removes this bottleneck by issuing tickets, introducing **Single Sign-On** (SSO) as a major feature of Kerberos. Once the user is authenticated and receives an ST, the domain controller is no longer involved (see Figure 4). The user just presents its ticket to the resource. Remember, though, if the user makes a *new* request for another resource, another ST has to be issued by the TGS. [Ref. 5]

The greatest benefits of SSO are obtained from implementing Windows 2000 homogenous domains, but significant benefits can be found even when deploying Windows 2000 in heterogeneous networks. One such benefit is that since SSO for Windows 2000 interoperates with numerous operating systems, it is an ideal choice to serve as an SSO hub in heterogeneous networks. Other benefits include:

- **Simpler administration.** SSO for Windows 2000 uses the same administration tools for SSO-specific tasks an administrator uses for other administrative tasks.

- **Better administrative control.** Network management information, including SSO-specific information, is stored in the Active Directory, producing a single authoritative list listing of each user's rights and privileges. Any changes made to a user's account will propagate through the entire network.

- **Improved user productivity.** Clients no longer have to use multiple logons or multiple passwords in order to access network resources.

- **Better network security.** All SSO methods provide secure authentication and provide a foundation for encrypting the client's session with the target resource. Network security is enhanced because of the consolidation of network management information in the Active Directory. When an administrator disables a user's account, the administrator knows with certainty that the account is fully disabled.

- **Consolidation with heterogeneous networks.** Joining other dissimilar networks, allowing a central administrative focal point for security policies to be enforced can consolidate administrative efforts. [Ref. 5]

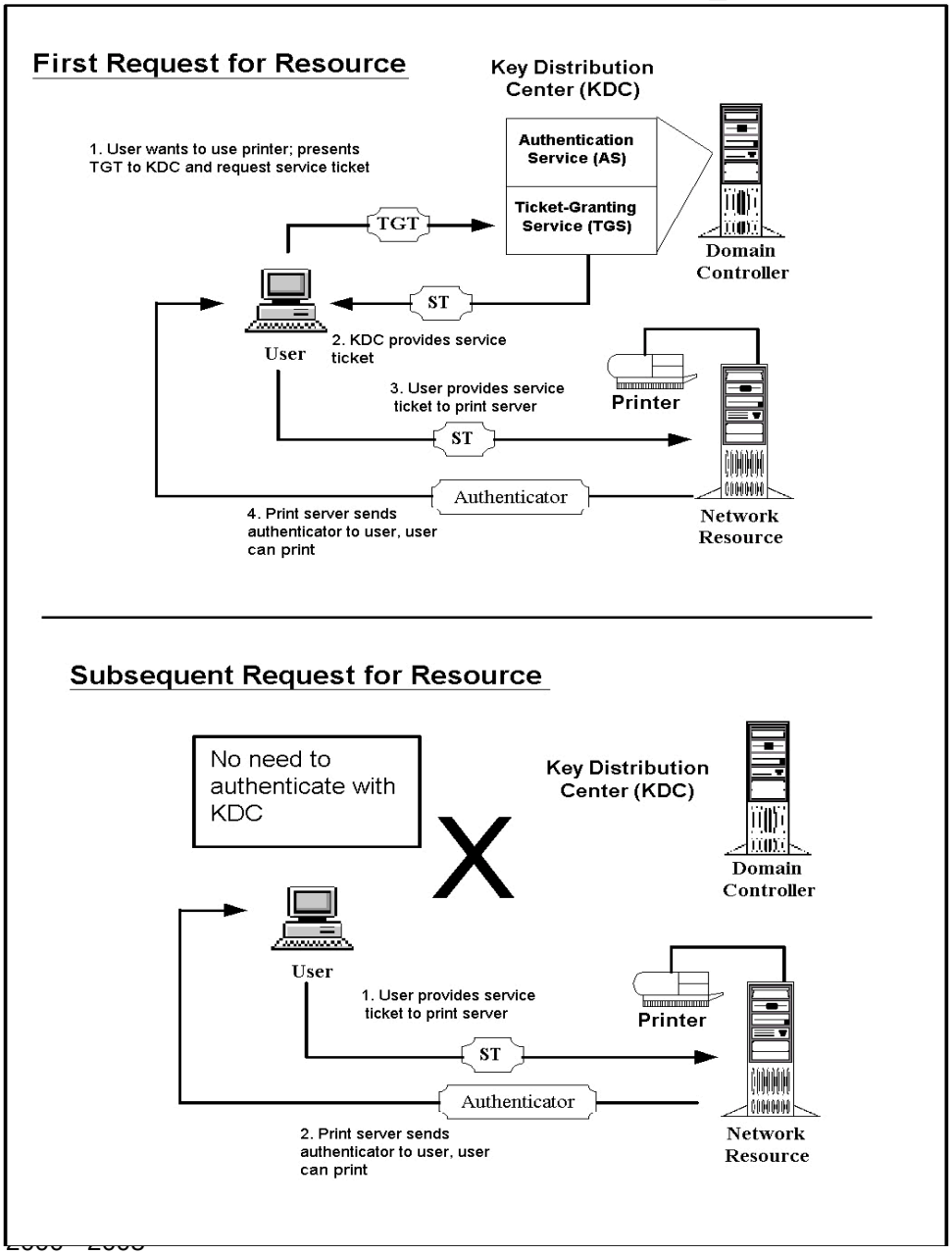


Figure 4. Initial VS. Subsequent Requests for Services.

E. Cross-Realm Authentication

Suppose a user needs to access a resource in another Windows 2000 domain. Kerberos is responsible for establishing two-way, transitive trust relationships between domains (or Kerberos Realms!). A trust relationship between the domains is required, but as long as the domains are members of the same *domain tree* a trust relationship is established by default.

In Windows NT 4.0, trusts could be one-way or two-way. In Windows 2000 and Active Directory, all trusts are two-way, or bi-directional. More importantly, all trusts are transitive, which means that a trust exists between two domains if both have trust relationship with a common intermediate domain. For example, if Domain A trusts Domain B, and Domain B trust Domain C, then there is an automatic implicit trust between Domain A and Domain C (see Figure 5). Kerberos takes advantage of this implementation and alleviates the cumbersome task NT 4.0 administrators deal with: establishing and maintaining trust relationships.

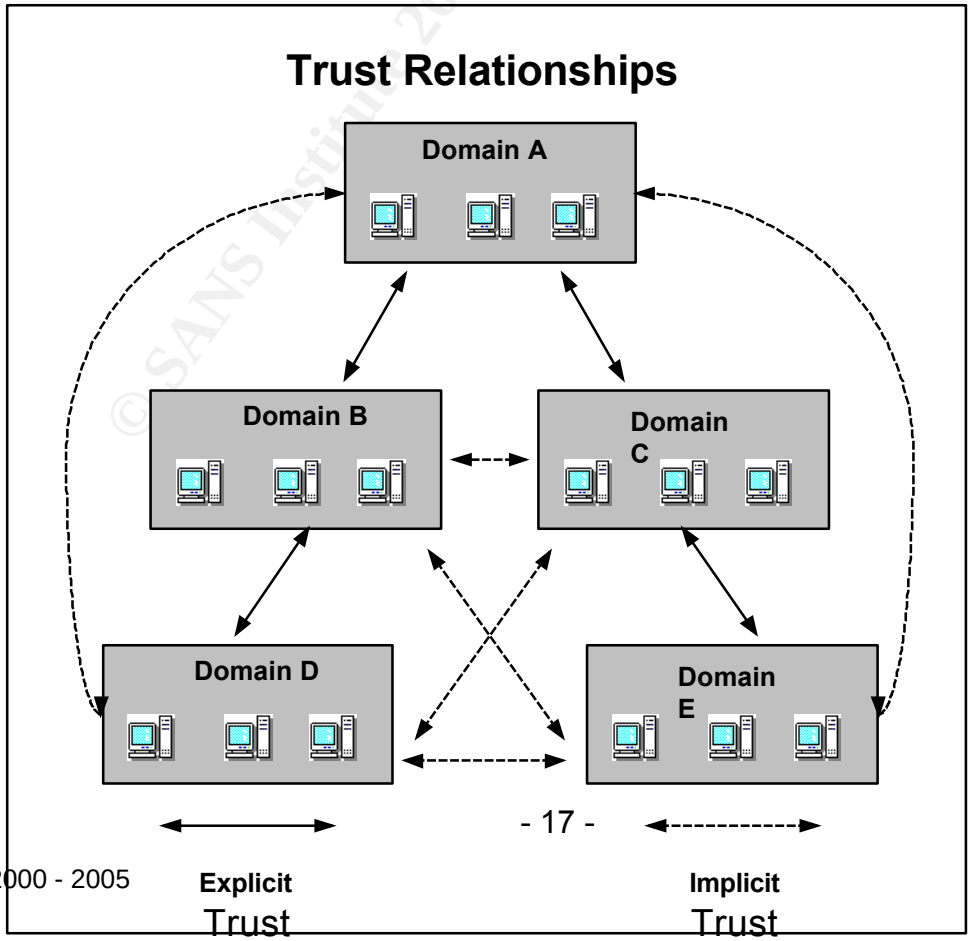


Figure 5. Trust Relationships with Windows 2000. After Ref. [9]

Windows 2000 enables authentication across domains by sharing an inter-domain key. The inter-domain key is shared automatically when two domains establish a trust relationship. Kerberos simplifies cross-domain authentication by registering the TGS of each domain as a security principal with the other domain KDCs. The TGS in each domain can then regard the TGS in the other domains as just another service. This scenario is made possible because the two functions of the KDC, the AS and the TGS, are two distinct services. A user can receive a TGT from its home domain and present it to the TGS of another domain to get an ST. Figure 6 illustrates cross-real authentication. **Note that for this illustration, the user has already been authenticated to *domain1* and has already received a TGT.**

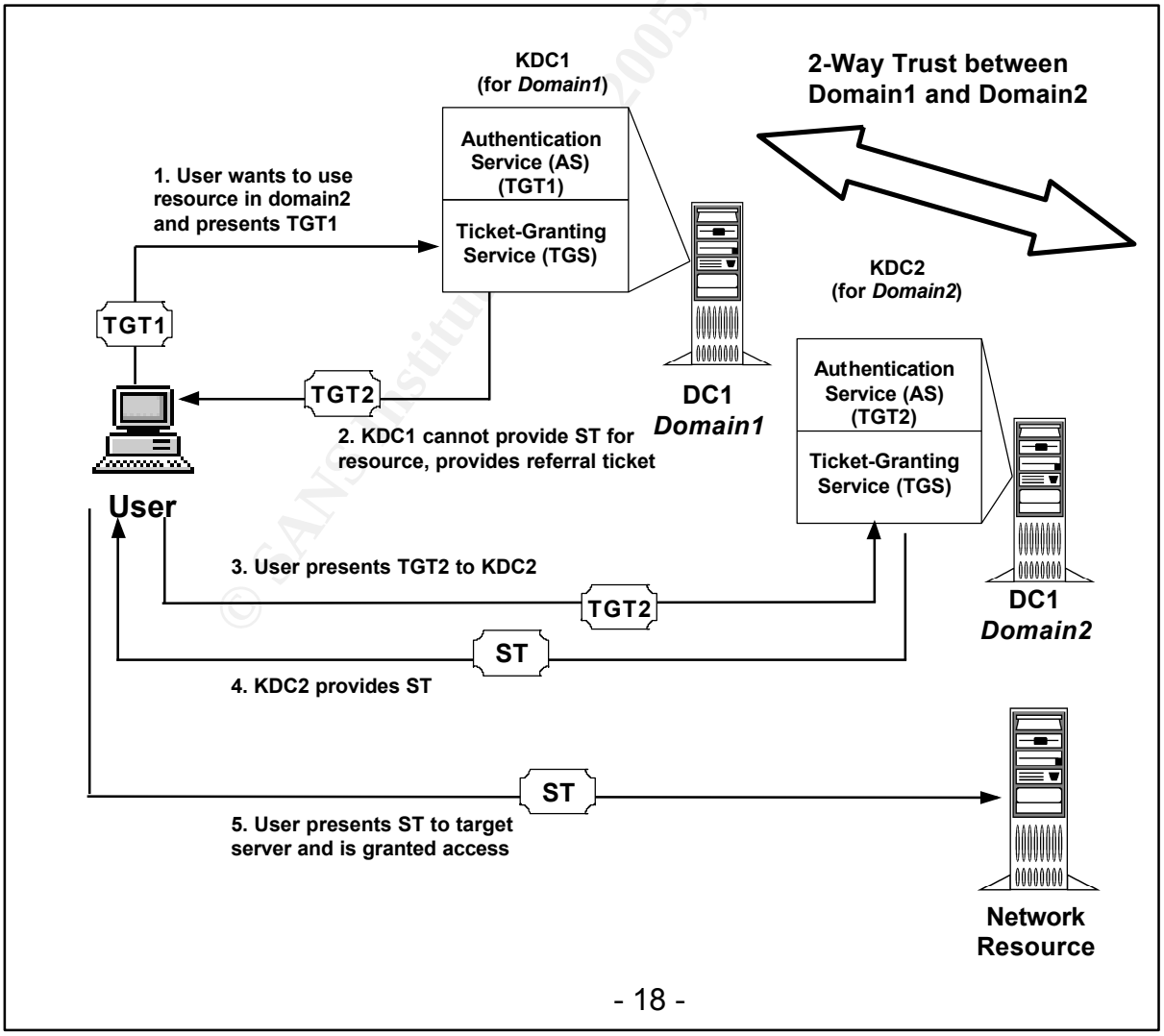


Figure 6. Cross Realm Authentication. After Ref. [5].

In this example, the user presents his TGT to the Domain1 TGS (KDC1). The TGS on KDC1 recognizes that the target server is not a security principal in Domain1 (or in the Kerberos Realm1) and issues a ***referral ticket***. The referral ticket is a TGT encrypted with the inter-domain key shared by the KDCs in both domains. The user presents the referral ticket (TGT2) to the TGS in Domain2, receives a ST, and presents the ST to the target server. Notice that after step 3 in figure 6, authentication is exactly the same as the steps when requesting a ST in a user's local domain (Figure 3). [Ref. 5, 6]

For **multiple-domain** authentication, Kerberos enables clients to travel a referral path to access resources in other domains. A user can present a TGT to a KDC in another domain, which refers the user to another intermediate domain (remember transitive trusts from Figure 5), which issues a referral ticket to another domain, until the domain with the target server is reached and a ST is issued. [Ref. 5]

F. Kerberos Weaknesses

To some degree, all networked systems are susceptible to some type of computer attack, unauthorized access, or computer fraud. While Kerberos is the most optimal solution to securing distributed networks, it does have weaknesses.
The most obvious risk to Kerberos is from a dictionary attack on passwords. A dictionary attack is one where commonly used passwords are compared against a password or password file to gain unauthorized access. While a strict and effective security policy, such as enforcing the Account Lockout threshold, will offset this weakness, Kerberos still remains susceptible.

Another weakness, though correctable with resource planning, is that the KDC must be physically secured. An unauthorized user who gains access to the KDC could potentially gain unrestricted access to the entire network.

Possibly the most difficult risk to offset is the human factor. It is virtually impossible to stop trusted staff members or administrators from accessing applications or files in which they have no need to know, but because of the nature of their job, they must have access to properly administer the system. In a case such as this, no security system is able to provide the needed protection.
[Ref. 9]

IV. KERBEROS ADMINISTRATION

Kerberos does not require extensive configuration or administration; it is installed and automatically enabled when Active Directory or the ***dcpromo*** utility is installed. This section discusses Kerberos Policy and Kerberos administration utilities.

A. Kerberos Policy

Kerberos policy is defined at the *domain level* and implemented by the KDC on each domain controller. Kerberos policy cannot be defined at the organizational unit (OU) level. Since all domain controllers enforce the account policies that are defined in the Default Domain Policy, domain controllers ignore Kerberos policies that are defined at the OU level. Kerberos policies are always defined on the first domain controller installed in a Windows 2000 domain. This practice allows for Windows 2000 installations, such as upgrades from Windows NT 4.0 that did not support Kerberos, and guarantees that Kerberos is defined and enabled on the domain. Also, since the Kerberos GPO is a domain-wide account policy setting and is enforced by all domain controllers, all domain controllers always retrieve the values of these account policy settings from the Default Domain Policy GPO. [Ref. 10]

Kerberos Policy can be accessed using several methods. The first method to administer Kerberos policy is through the **Domain Security Policy** tool under **Administrative Tools**. Once Domain Security Tools is opened, select Security Settings ➔ Account Policies ➔ Kerberos Policy. Alternatively, you can administer Kerberos Policy through the **Group Policy Object** (GPO) snap-in. It is important to select the ***Default Domain Policy*** option when installing the GPO snap-in. Once the GPO snap-in is enabled, select Default Domain Policy ➔ Computer Configuration ➔ Windows Settings ➔ Security Settings ➔ Kerberos Policy. Using the GPO snap-in method ensures that the Kerberos policy is defined in the Default Domain GPO. Figure 7 illustrates administering Kerberos Policy using the Group Policy snap-in.

As mentioned earlier, Kerberos does not require a system administrator to enable, or "turn on" the protocol: when a Windows 2000 server is promoted to a domain controller, Kerberos is installed and started. Looking at Figure 7, you would think otherwise, as the four Kerberos option settings are listed as "Not defined." This is the default setting for these options and only means that an administrator has not changed the configuration. By default, you must be a member of the Domain Admins Security group to modify the Kerberos policy settings. Table 1 list the Kerberos Policy options and default values.

Kerberos Policy settings are interdependent on one another. For example, look at Figure 8. When the ***Maximum lifetime for service ticket*** was defined for a maximum lifetime of 600 minutes (default), the ***Maximum lifetime for user ticket*** and ***Maximum lifetime for user ticket renewal*** options are also changed to suggested values. Note that the ***Maximum lifetime for service ticket*** value must be greater than 10 minutes and less than the ***Maximum lifetime for user ticket renewal*** value. If you try to set values out of accepted ranges a Suggested Value Changes

screen will appear with suggested values.

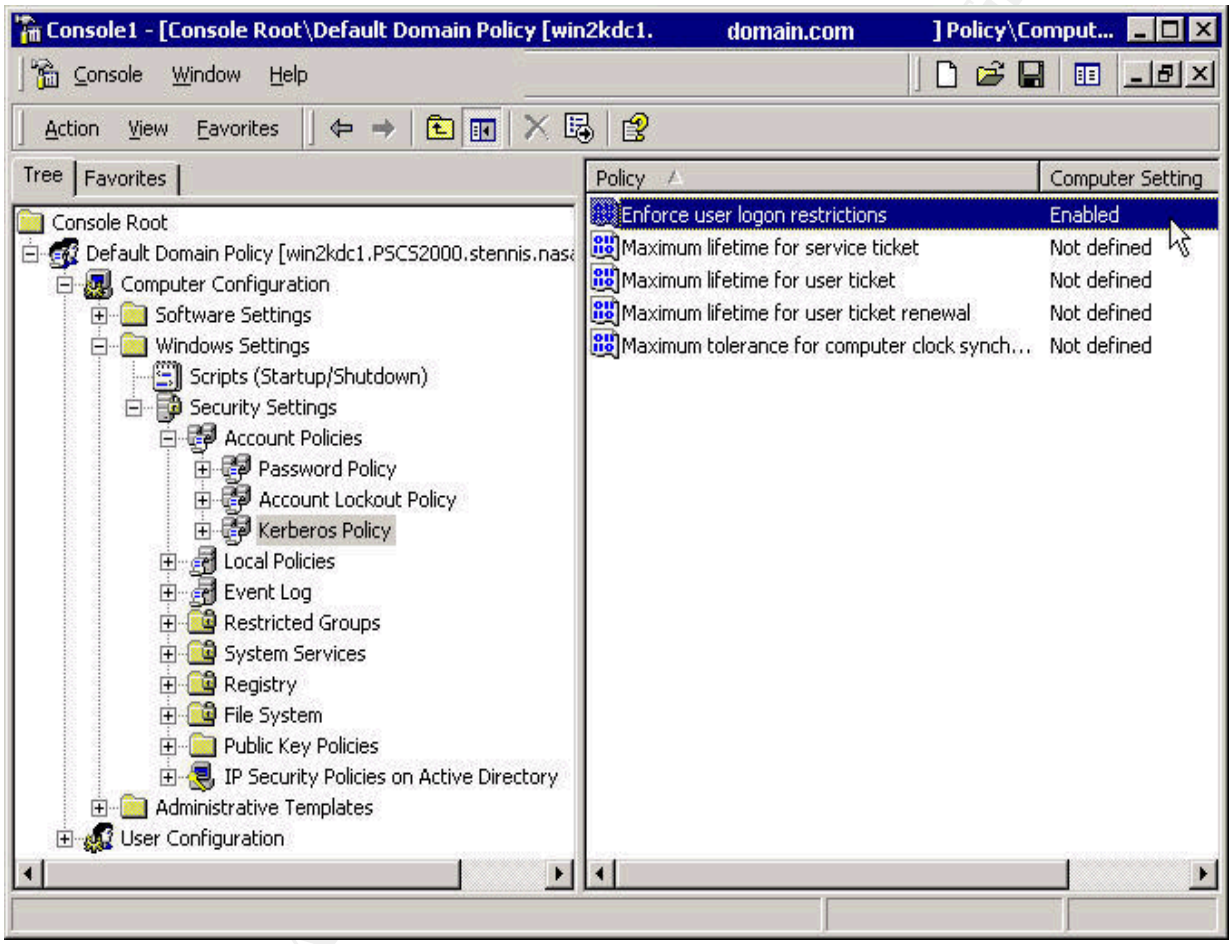


Figure 7. Default Domain Policy – Kerberos Policy Settings.

Kerberos Policy Settings			
Policy	***Default Setting***	***Recommended Setting***	***Function***
Enforce user logon restrictions	Enabled	ALL – Enabled	Determines whether the KDC validates every request for a session ticket. Any restrictions placed on a user account are enforced.

Maximum lifetime for service ticket	600 minutes	ALL – 600 minutes	Maximum duration for which a service ticket is valid for access to a service.
Maximum lifetime for user ticket	10 hours	ALL – 10 hours	Maximum duration for which a user's TGT is valid.
Maximum lifetime for user ticket renewal	7 days	ALL –10 days	Maximum length of time that a TGT can be used if repeatedly renewed.
Maximum tolerance for computer clock synchronization	5 minutes	All – 5 minutes	Maximum time within which computers in the domain must be synchronized. If they fall outside of this time, authentication fails

Table 1. Kerberos Policy Settings.

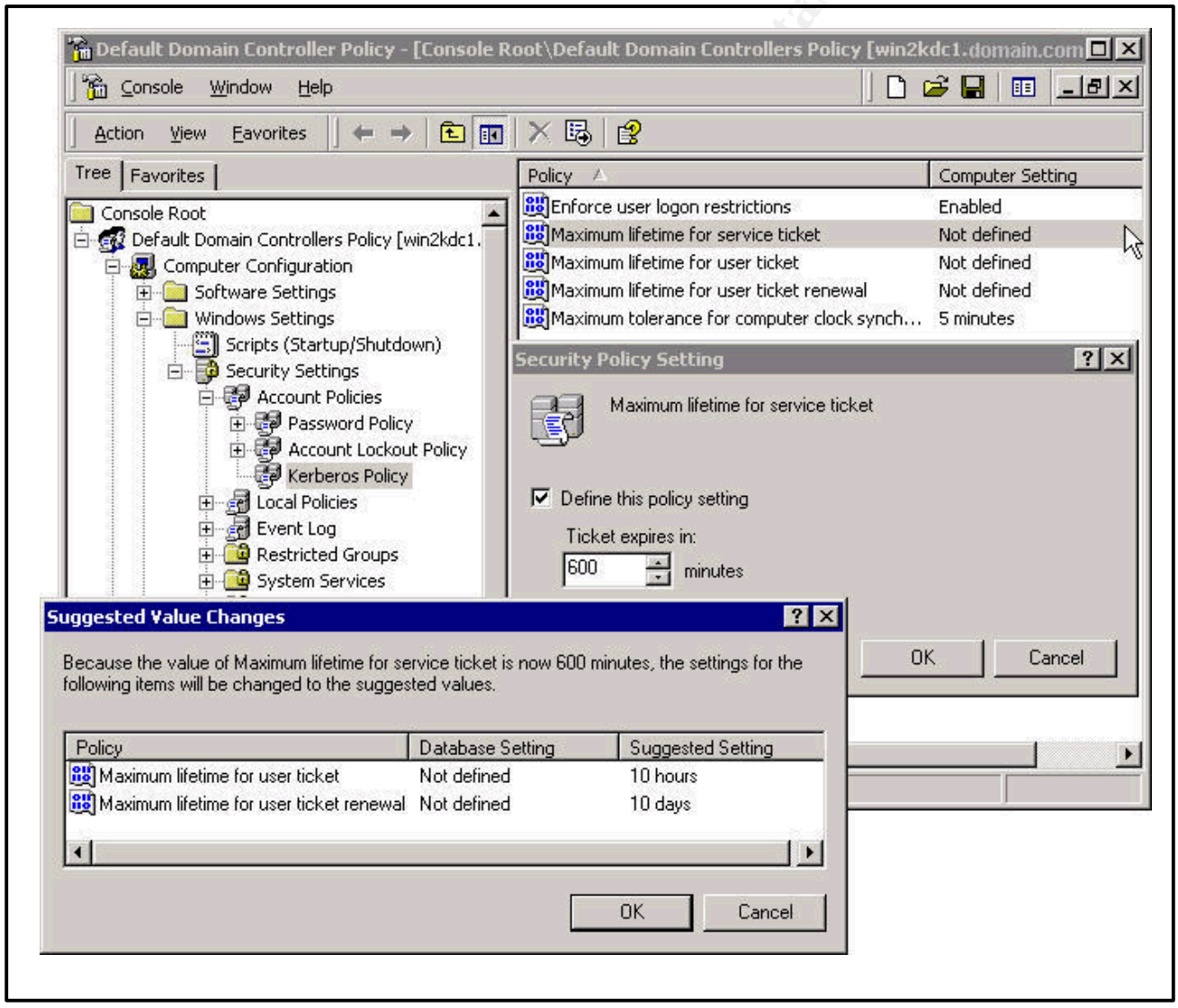


Figure 8. Kerberos Policy Interdependencies.

A final note on Kerberos Policy options: While these settings can be changed to strengthen network security, keep in mind that when one setting is changed, it affects other settings. Many times security and performance are a tradeoff. If security is more important than network performance, a shorter ticket lifetime for tickets is the recommended approach. Likewise, a longer ticket lifetime improves network performance but could adversely affect security.

B. Kerberos Utilities

There are several utilities included in Windows 2000 (or in the Windows 2000 Server Resource Kit) that are helpful in Kerberos administration. Some are command line utilities while others are GUI-based tools. This section discusses some of the most useful tools, including KerbTray, Netdom, and klist.

1. KerbTray

KerbTray is a useful utility found on the Windows 2000 Server Resource Kit (in the Network Management Tools folder). KerbTray allows a user to view his ticket information. The kerbtray.exe tool must be resident on the client machine or server in order to be executed. KerbTray can be installed from the Windows 2000 Server Resource Kit on a Windows 2000 Professional client or a Windows 2000 server. (Of course, The Kerberos protocol must be running on the computer in order for KerbTray to work!) On Windows 2000 computers, tickets and session keys are stored in a credentials cache. This cache is volatile memory, not stored on disk and is protected by the LSA. When a user logs off or shuts down the computer, all tickets and keys are destroyed. [Ref. 6]

The KerbTray tool can be run from the command line or the Start ➔ Run menu. Once executed, a KerbTray icon is placed in the System Tray. Right-click on the icon to see the menu option (**List Tickets** and **Purge Tickets**, see Figure 9). To see a list of the tickets right-click on the icon and select **List Tickets**. You can also double-click the icon to bring up the same screen as **List Tickets**.

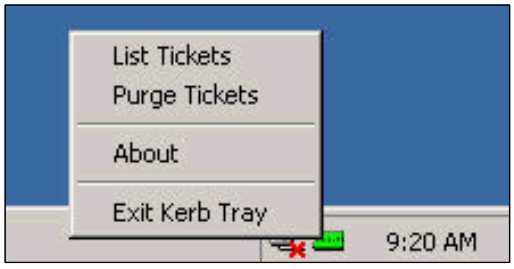


Figure 9. The KerbTray icon in the System Tray.

The KerbTray tool allows users to view important information about tickets issued to them. Four information tabs reveal Kerberos ticket information:

- **Names –** Lists the requestor of the ticket, the service name and target name (the host computer).
- **Times –** Reveals when the ticket became valid, when the validity ends, and renewable information.
- **Flags –** Information concerning configured settings for tickets, such as Forwardable, Preauthenticated, and Renewable settings.
- **Encryption Types –** Encryption type used to encrypt the Kerberos ticket and the session key.

Figures 10-13 are screen shots from a Windows 2000 testbed that illustrate the KerbTray utility. **Note:** the information provided is a depiction of the actual testbed data to maintain privacy.

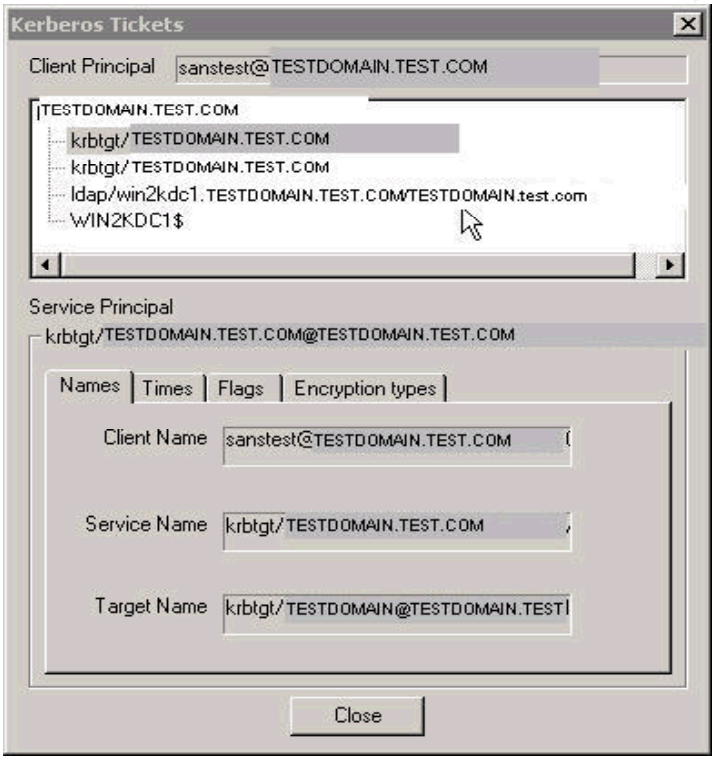


Figure 10. The KerbTray Clients Name Tab.

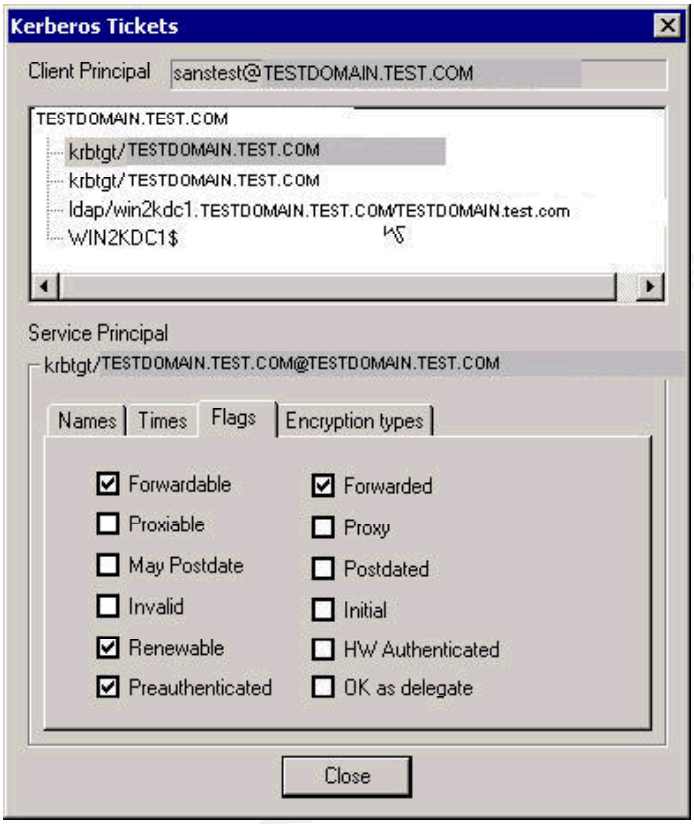


Figure 11. The KerbTray Times Tab.

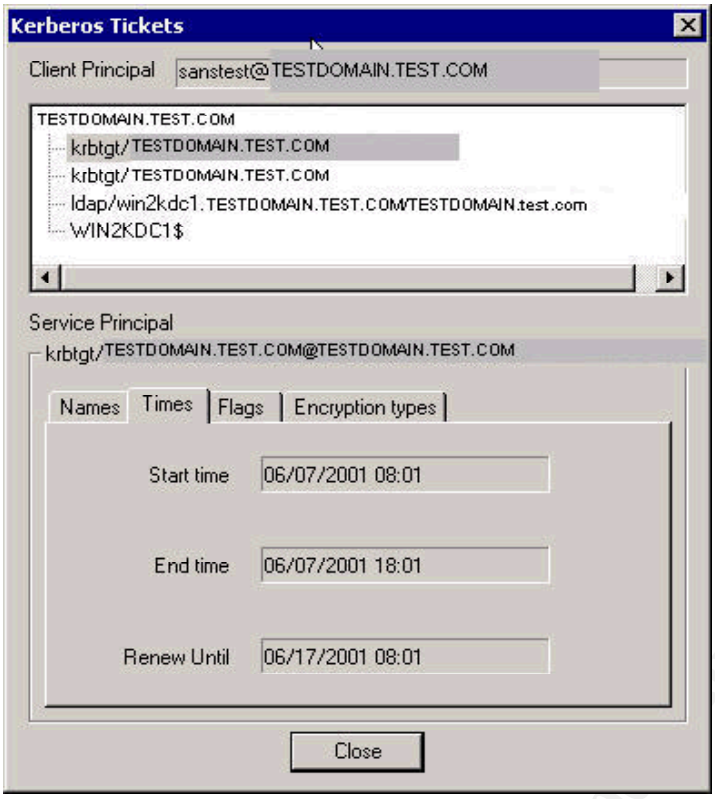


Figure 12. The KerbTray Flags Tab.

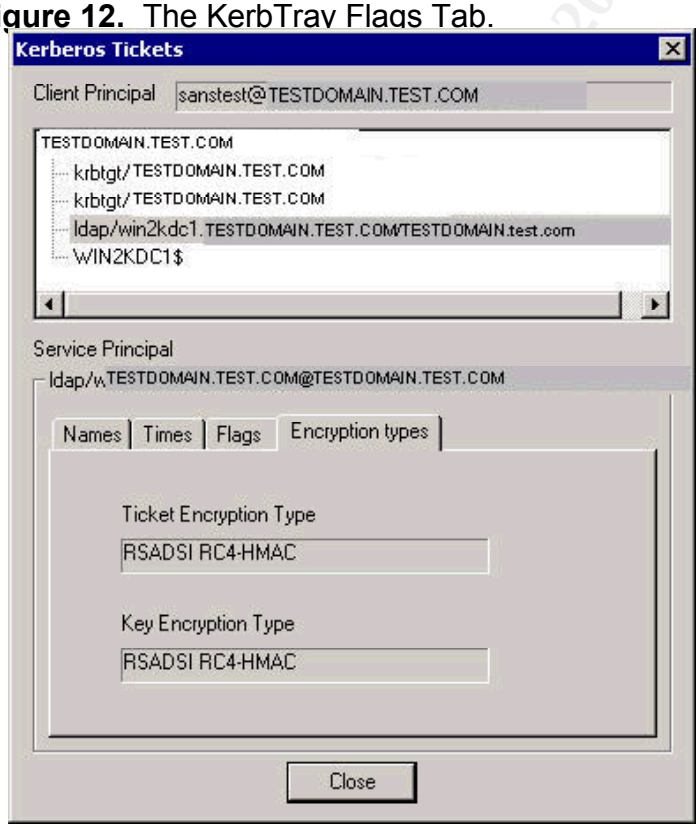


Figure 13. The KerbTray Encryption Tab.

The other KerbTray option is **Purge Tickets** (see Figure 9). Be sure you really want to purge the ticket cache, because once selected, all cached tickets are deleted, and the user will have to log off and login again to authenticate.

2. NetDom

NetDom (*Network Domains*) is a command-line utility used to manage Windows NT/2000 domains and trusts, and is found on the Windows 2000 Server Resource Kit. Netdom allows administrators to verify and reset domain trusts, add a Windows 2000 computer to a Windows NT/2000 domain, and manage computer accounts for domain member clients and servers.

Netdom is also a valuable tool when creating non-Windows Kerberos realm trust relationships. The netdom.exe tool can establish two-way, transitive, non-Windows Kerberos realm trust relationships. This procedure may be easier than using the ***Active Directory Domains and Trusts*** Admin Tool, which forms a one-way and nontransitive trust when creating non-Windows Kerberos realms. It can also be used to modify a non-Windows Kerberos realm trust that was created in *Active Directory Domains and Trusts*.

Some of the most useful Netdom commands include (but not limited to):

- **Add** – adds a workstation or server *account* to the domain.
- **Join** – adds a workstation or member server to the domain.
- **Move** – move a workstation or member server to a new domain.
- **Query** – returns domain information.
- **Remove** – removes a workstation or server from the domain.
- **Trust** – Manages or verifies the trust relationship between domains.
- **Time** – Verifies or resets the time synchronization between a workstation and a domain controller.

3. Klist

Klist (*Kerberos List*) is another command-line tool that allows system administrators to delete Kerberos tickets granted to the current logon session. Parameters include:

- **tickets** (Figure 14) – Lists the current tickets of services that the current user has authenticated to in the logon session.
- **tgt** (Figure 15) – Lists the initial Kerberos TGT and TGT data.
- **purge** (Figure 16) – Allows an administrator to delete specific tickets.

```
C:\>klist tickets

Cached Tickets: (4)

   Server: krbtgt/TESTDOMAIN.TEST.COM@TESTDOMAIN.TEST.COM
      KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
      End Time: 6/7/2001 23:21:02
      Renew Time: 6/17/2001 13:21:02

   Server: krbtgt/TESTDOMAIN.TEST.COM @ TESTDOMAIN.TEST.COM
      KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
      End Time: 6/7/2001 23:21:02
      Renew Time: 6/17/2001 13:21:02

   Server: WIN2KDC1$@TESTDOMAIN.TEST.COM
      KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
      End Time: 6/7/2001 23:21:02
      Renew Time: 6/17/2001 13:21:02

   Server: ldap/win2kdc1.TESTDOMAIN.test.com/@TESTDOMAIN.test.com
      KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
      End Time: 6/7/2001 23:21:02
      Renew Time: 6/17/2001 13:21:02

C:\>
```

Figure 14. Klist tickets Information.

```
C:\>klist tgt

Cached TGT:

ServiceName: krbtgt
TargetName: krbtgt
FullServiceName: sanstest
DomainName: TESTDOMAIN.TEST.COM
TargetDomainName: TESTDOMAIN.TEST.COM
AltTargetDomainName: TESTDOMAIN.TEST.COM
TicketFlags: 0x40e00000
KeyExpirationTime: 256/0/29920 0:100:8048
StartTime: 6/7/2001 13:21:02
EndTime: 6/7/2001 23:21:02
RenewUntil: 6/17/2001 13:21:02
TimeSkew: 6/17/2001 13:21:02

C:\>
```

Figure 15. Klist tgt Information.

```
C:\>klist purge

Cached Tickets: (4)

   Server: krbtgt/TESTDOMAIN.TEST.COM@ TESTDOMAIN.TEST.COM
      KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
      End Time: 6/7/2001 23:21:02
      Renew Time: 6/17/2001 13:21:02

Purge? (y/n) : y
        Deleting ticket:
           ServerName = krbtgt/ TESTDOMAIN.TEST.COM (cb=64)
           RealmName  = TESTDOMAIN.TEST.COM (cb=50)
Submit Buffer size = 142
        Ticket purged!

   Server: krbtgt/ TESTDOMAIN.TEST.COM @ TESTDOMAIN.TEST.COM
      KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
      End Time: 6/7/2001 23:21:02
      Renew Time: 6/17/2001 13:21:02

Purge? (y/n) : y
        Deleting ticket:
           ServerName = krbtgt/ TESTDOMAIN.TEST.COM (cb=64)
           RealmName  = TESTDOMAIN.TEST.COM (cb=50)
Submit Buffer size = 142
        Ticket purged!

   Server: WIN2KDC1$@ TESTDOMAIN.TEST.COM
      KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
      End Time: 6/7/2001 23:21:02
      Renew Time: 6/17/2001 13:21:02

Purge? (y/n) : y
        Deleting ticket:
           ServerName = WIN2KDC1$ (cb=18)
           RealmName  = TESTDOMAIN.TEST.COM (cb=50)
Submit Buffer size = 96
        Ticket purged!

   Server:
ldap/win2kdc1.TESTDOMAIN.test.com/TESTDOMAIN.TEST.COM@TESTDOMAIN.TEST.COM
      KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
      End Time: 6/7/2001 23:21:02
      Renew Time: 6/17/2001 13:21:02

Purge? (y/n) : y
        Deleting ticket:
           ServerName = ldap/win2kdc1.TESTDOMAIN.test.com/TESTDOMAIN.test.com
(cb=130)
           RealmName  = TESTDOMAIN.TEST.COM (cb=50)
Submit Buffer size = 208
        Ticket purged!

C:\>
```


Figure 16. Klist purge Information.

Note the difference between the **KerbTray Purge Tickets** option and the **Klist purge** option. The Klist purge option allows you to select which tickets to purge. When you select the KerbTray purge option, all tickets are purged immediately after selecting the option. Be sure that you really want to purge ***all*** the cached tickets using KerbTray, because you do not get a second chance. Once purged, the tickets are gone, and the user may be required to log off and re-authenticate.

C. Event Logging

By default, Kerberos event logging is not enabled on Windows 2000 computers. A system administrator may want to consider enabling Kerberos event logging as it will capture information that could be very useful during Kerberos troubleshooting.

Kerberos logging is enabled in the registry. Microsoft warns that "Using Registry Editor incorrectly can cause serious problems that may require you to reinstall your operating system." If a user is reporting authentication problems you can enable event logging on the client. Use the following steps to enable Kerberos logging:

1. Start the Registry Editor (regedt32.exe).
2. Add the following registry value:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos

Registry Value: LogLevel
Value Type: REG_DWORD
Value Data: 0x1

Once this key is entered, the computer must be restarted for the settings to take place. Enabling logging will allow administrators to view Kerberos specific events in the system log. [Ref. 11]

D. Kerberos and IPSec

IPSec can be used to secure Kerberos traffic between domain controllers. However, Kerberos traffic is not secured between domain controllers by default. Even if the IPSec policy filter is configured to match all IP traffic between the two IP addresses, or even when the IPSec policy specifies that all IP traffic should be secured, Kerberos is still not secured. Installing the latest service pack is a required step in resolution. Get the latest service pack at:

(http://www.microsoft.com/windows2000/downloads/servicepacks/default.asp)

A registry change is also required.

1. Start the Registry Editor (regedt32.exe).
2. Add the following registry value:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IPSEC

On the Edit menu, click Add Value, and then add the following value:

- Value Name: NoDefaultExempt
- Data Type: REG_DWORD
- Data Value: 1

A value of 1 will ensure Kerberos and PSVP are not exempted for the IPSec policy. [Ref. 12]

Be aware that this registry change can cause problems with your current domain trusts. All domain controllers in the forest or tree, including child domains, must have the same settings for Kerberos and IPSec.

V. CONCLUSION

This paper introduced the Kerberos protocol, Kerberos concepts, and the Single Sign-On concept. The reader should have a good understanding of the Kerberos authentication process and the service interdependencies that go along with the logon/logoff process. Kerberos uses mutual authentication to provide clients proof that a server is what it claims to be, and introduces a time limit (by way of a ticket's lifetime) in the client's access to servers and resources across a network.

While Kerberos is currently the best solution for a secure and easily administered security-authentication system for distributed networks, be aware that no system is totally secure. Stay current with security patches, read security notifications, and keep a watchful eye on logging events.

LIST OF REFERENCES

1. Microsoft, *Microsoft Windows 2000 Server Distributed Systems Guide*, Microsoft Press, 2000.

2. Schmidt, Jeff, *Microsoft Windows 2000 Security Handbook*, Que Corporation, 2000.

3. Microsoft, *MCSE Training Kit: Designing Microsoft Windows 2000 Network Security*, Microsoft Press, 2001.

4. Microsoft, "Secure Networking Using Windows 2000 Distributed Security Services," white paper, Internet (http://www.microsoft.com/WINDOWS2000/techinfo/howitworks/security/distsecservices.asp).

5. Microsoft, "Single Sign-On in Windows 2000 Networks," white paper, (http://www.microsoft.com/TechNet/win2000/win2ksrv/prodfact/nt2ksso.asp).

6. Microsoft, “Windows 2000 Kerberos Authentication,” white paper, Internet (http://www.microsoft.com/TechNet/win2000/win2ksrv/technote/kerberos.asp).

7. Spalding, George, *Windows 2000 Administration*, Network Professionals, 2000.

8. Walla, Mark, “Kerberos Explained,” *Windows 2000 Advantage*, May 2000.

9. Kaplan, Ari and Nielson, Morten S., *NT 5: The Next Revolution*, The Coriolis Group, Inc., 1998.

10. Microsoft, “Windows 2000 Distributed Security Features,” white paper, (http://www.microsoft.com/TechNet/win2000/win2ksrv/secover.asp).

11. Microsoft, “How to Enable Kerberos Event Logging [Q262177],” Microsoft Technet article, May 2001 CD.

12. Microsoft, “IPSec Does Not Secure Kerberos Traffic Between DCs [Q254728],” Technet article, May 2001 CD.

Cover page image from http://www.anet.ee/serge/painters/Boris/unknown/17.jpg