Global Information Assurance Certification Paper

Authentication:
Windows 2000 and MIT Kerberos Interoperability

Prepared for:
SANS GIAC GCNT Certification
Version 2.1b

Frank A. Nevers
July 2001

This page intentionally left blank.

Table of Contents

Deleted: ¶

TABLE OF FIGURES

ACRONYMS AND ABBREVIATIONS

AD	Active Directory
AS	Authentication Service
DC	Domain Controller
ERD	Emergency Repair Disk
GPO	Group Policy Object
GUI	Graphical User Interface
KDC	Key Distribution Center
LSA	Local Security Authority
MIT	Massachusetts Institute of Technology
MMC	Microsoft Management Console
OS	Operating System
OU	Organizational Unit
RFC	Review For Comment
RPC	Remote Procedure Call
SANS	System Administration, Networking, and Security Institute
SAT	Security Access Token
SID	Security Identifier
SSPI	Security Support Provider Interface
TCP	Transmission Control Protocol
TGT	Ticket Granting Ticket
UDP	User Data Protocol

INTRODUCTION

From a security perspective, the key to securing a network operating system is largely dependent on the authentication model. Regardless of what other features are bundled with the software, authorization is the key to the kingdom. It is important to clarify that authentication is the process of verifying who is on the network using a particular account; while authorization determines what resources a user has access to. This paper will examine Microsoft's implementation of the Kerberos protocol, specifically how Windows 2000 Active Directory (AD) and an external Massachusetts Institute of Technology (MIT) Kerberos realm can co-exist.

First, we will examine the basics of Kerberos authentication then discuss the interoperability between an external MIT Kerberos realm and a Windows 2000 AD network. A step-by-step guide on configuring the various components of a Windows 2000 network to use an external Key Distribution Center (KDC) is presented. Examples and third party tools that will assist the security administrator in creating some harmony between these two environments are also documented. It is important to note that this paper identifies only the modifications necessary to make an external MIT KDC work with AD. There are many good articles on Microsoft's implementation of Kerberos within Windows 2000 including two SANS GIAC-NT certified practicums:

- Robert Ashworth: "Kerberos – Windows 2000 Authentication"[1]
- Michael Murphy: "Authentication in Windows NT and Windows 2000"[2]

Many organizations have a large and mature Kerberos infrastructure and for numerous reasons want to maintain that structure. The strategic step from a business point of view is how to incorporate Windows 2000 and Active Directory into this infrastructure. Microsoft claims that Windows 2000 is RFC compliant with Kerberos. While this may be technically true, implementation inconsistencies can cause compatibility issues. Windows 2000 tickets can have extensions that Kerberos does not support. Macintosh clients attempting to use Windows 2000 to authenticate to an external MIT realm cannot authenticate. Older Windows 9x and NT clients also present inconsistencies. These older clients can authenticate but not be authorized to gain access to resources.

Kerberos, developed by MIT in 1988, has been an Internet Engineering Task Force (IETF) protocol since 1993. The word "Kerberos" comes from the mythological Greek creature with three heads. The Latin spelling of "Kerberos" is "Cerberus", which coincidently is the same spelling Microsoft used when users perform a spell check in Office 2000. MIT chose to name their project beginning with a "K," "Kerberos," to represent three features:

1. Client/application - which is the principal.
2. Network resource - which is any resource the principal is attempting to access.
3. Key distribution Center (KDC) – which is the authoritative controller.

System administrators worldwide have used Kerberos as their authentication scheme. This protocol provides a symmetric-key, client server authentication mechanism for large-scale networks.

KERBEROS AUTHENTICATION

Kerberos is based on the idea that no password should travel over the network wire to authenticate a user. Only the end user and the Kerberos server or KDC should know the end user's password. This is the concept of "shared secrets."

For example: If I tell Sam a secret only Sam and I know the secret. If our secret is: "Meet at 10 pm on the Circle" with a keyword of Que. To send a message to Sam through Kim I tell Kim the keyword, which we assume she relays to Sam. Sam takes this keyword and decrypts it to mean "Meet at 10 pm on the Circle". Since Kim gave Sam the message in its entirety then SAM knows the message came from me. Kim never knows what the secret is. Only Sam and I know the meaning. Thus, I could send Sam the keyword and any message through any third party and as long as neither of us revels the secret we can use the keyword indefinitely. Sam would always know the message originated from me.

This is Kerberos in its simplest form. Two parties, the user and the KDC, share a secret password. No one else knows this password as long as the password is kept private at all times, meaning the user does not mention it to anyone and assuming no hacker breaks into the KDC and steals the password database then this asymmetric idea of shared secrets remains intact. This is the underlying concept of the MIT Kerberos. Once you start implementing this concept in a wired network things get a bit trickier. Then throw a corporation like Microsoft into the mix and look out. Actually, it is not all that bad but things can get extremely complicated in a hurry.

The KDC is a single process that performs two logical functions; both services operate within the same machine (see figure 1 below).

- Authentication Service (AS). This service provides session tickets, and approves access credentials.

- Ticket-Granting Service (TGS). This is the service that issues Ticket Granting Tickets (TGT). These tickets allow users in remote domains to access resources in the local domain.

The KDC performs these same functions if you are using Microsoft's KDC service within Active Directory or using an external MIT Kerberos realm. Figure 1 below describes how these services can co-exist on the network. A realm is a boundary between services; in Microsoft terminology a realm is a domain.

Inside the Kerberos Protocol

Expanding on the example of Sam and Kim, we can look at how this concept works in a wired world. By taking a look under the hood we can begin to understand how a user can login without sending a password over the wire. There are six transactions that the client machine must successfully make with the KDC prior to completing the authentication process.

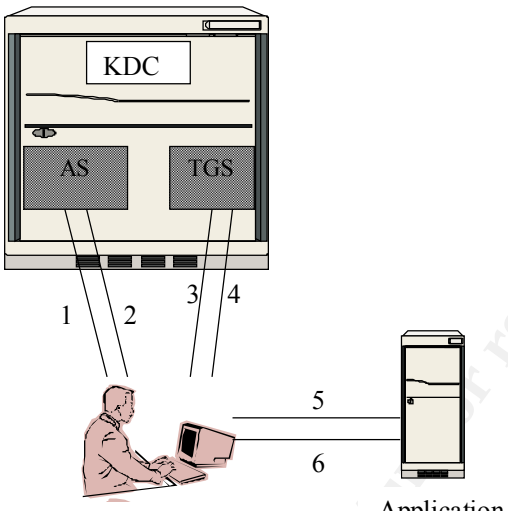

Figure 1. Kerberos Protocol

1. The user initiates contact with the KDC requesting to talk by creating a pre-authentication request. This is done when a user initiates a request to type in their logon credentials (username and password). With Windows 2000 this occurs when a user presses Ctrl + Alt + Delete.

2. The Authentication Service (AS) within the KDC receives the request and creates a key called a session key to send back to the client. The key contains a notification that it will talk, its name (the realm), a randomly generated hash number, and the original request of the client.

At this point the user has not yet been authenticated. In reality the user is still typing in the logon credentials (username and password).

3. When the client receives the session key back the client extracts the session request created in step one. The KDC session key becomes the key for all future communication. The client then appends a timestamp to the end of the session key and sends this back to the KDC with a hash of the user credentials.

4. The KDC opens this new key containing its data sent in step two along with the new client data. Based on this information, the Authentication Services makes a determination if the data received from the client matches what it knows about the client in its database. If it matches, authorization to use the network is granted, and the AS then passes the remaining processes off to the TGS. At this point the TGS creates and sends a Ticket Granting Ticket (TGT) back to the client. After this point the KDC is out of the loop and all future access is handled by the network operating system.

5. The client takes this TGT and presents it to the application service, which then appends authorization (access) data to the ticket.

6. When the application server receives the TGT it says, "Ok, I can grant resources to this client since I trust the KDC." The server then sends the client a service ticket allowing the user to access its resources without having to request user authentication credentials.

The TGT then remains valid for a pre-determined amount of time (usually 10 hours) and no further requests for authentication data is made during the session. All a KDC does is review authorization requests to use the network and provide a ticket. What has made this protocol so widely used is that it performs only one function and does that extremely well. This is one of the reasons Microsoft made Kerberos the default authentication mechanism with Windows 2000 AD networks.

Now we will turn our attention to looking at what configuration changes are necessary to incorporate this external KDC into a Microsoft AD network. Our discussion assumes have the following network environment:

- Windows 2000 Server installed on domain controller hardware
  - Active Directory in a single domain environment.
  - Native or mixed mode.

- Windows 2000 Professional workstations

- MIT Kerberos version 5 realm

EMERGENCY REPAIR DISK

Prior to discussing how to implement an external Kerberos realm to authenticate AD users, it is a good idea to review the procedure for creating an Emergency Repair Disk (ERD) in Windows 2000. There are a number of unexpected issues that can cause implementations to go wrong some for no apparent reason. Therefore, it's highly recommended that you have a current ERD prior to beginning the configuration changes necessary to use an external KDC to authenticate users. The ERD should be created at the time a new server is brought on-line and updated any time you add or remove system components. The process for creating an ERD in Windows 2000 has changed from NT 4.0.

Steps:

To create an ERD you'll need a blank formatted floppy disk. Begin the ERD creation procedure by:

1. Selecting Start
2. Run
3. Type in ntbackup
4. Select Emergency Repair Disk
5. Follow the prompts for creating the ERD.

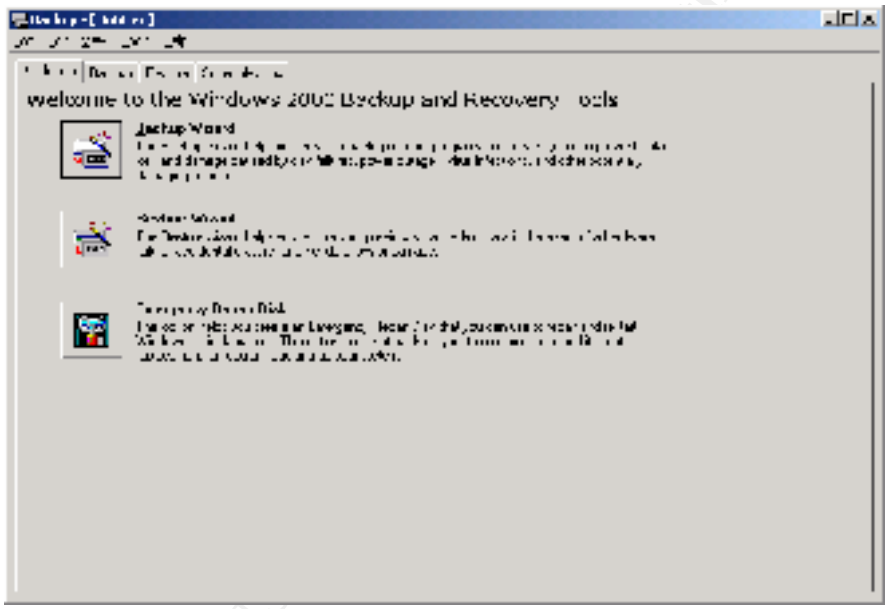

Figure 2. Ntbackup

Once you have completed the ERD creation process store the disk in a secure location. Remember to update the ERD as needed as the system changes. It's a good idea to make a second ERD to use as a backup once the Kerberos connections have been completed.

APPLICATION OF SERVICE PACKS

Microsoft has released two Service Packs for Windows 2000 and numerous hotfixes since the initial release of the operating system (OS). It is important from a security and performance standpoint to stay current with these fixes to the OS. As of this writing Service Pack 2 (SP2) for Windows 2000 has just been released. This service pack fixes an important bug in the round robin allocation of KDC's. To determine which service pack is installed on your machine, right mouse click on " My Computer" and select properties.

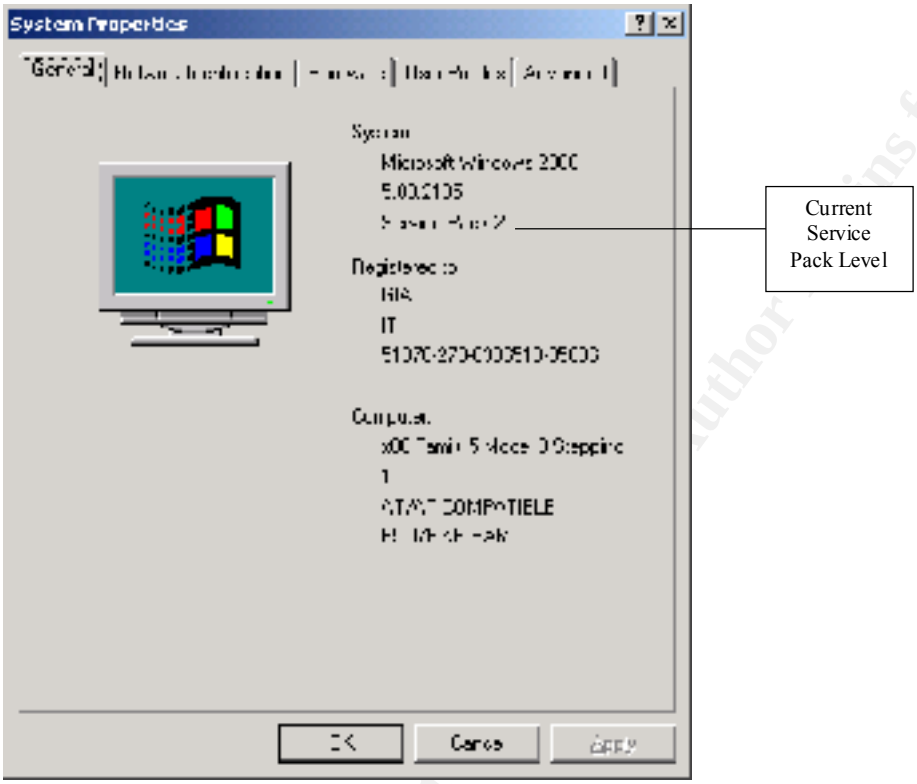

Figure 3. System Property Sheet

The AD domain controller and all client machines should be running SP2. This service pack corrects a problem where Windows 2000 Professional clients are only able to authenticate to the primary or first KDC even if others exist. Applying the service pack allows users to authenticate to secondary KDC's if the first one fails. Microsoft provided a Post SP1 hotfix for this problem, refer to Q article: Q288337.

KDC CONFIGURATION

On the KDC side, the Kerberos administrator will need to create a KDC password for AD service account (kerbtgt) to use. This should be completed at approximately the same time the steps for configuring the AD side of the trust are made.

By completing these commands on the KDC side your providing a password for AD domains to trust your KDC for authentication services. As we learned above, the KDC consists of two logical services running on the same machine, The Authentication Service that approves users logon credentials and the Ticket Granting Service that grants the TGT. No changes are necessary to these services. One of the benefits of an external MIT KDC providing authorization to the network is that the KDC trusts no one. Other services like AD are required to trust the KDC. This allows the KDC to be well guarded and highly secured.

Steps:

1. To create the trust password for the cross realm host principal use the following commands on the Unix KDC:

% Kadmin –q "ank –pw password kerbtgt/COMPANY.COM@MYKDC.COMPANY.COM"

% Kadmin –q "ank –pw password kerbtgt/MYKDC.COMPANY.COM@COMPANY.COM"

Where COMPANY.COM is the name of the Active Directory domain and MYKDC is the name of the external MIT Kerberos realm. Kerbtgt is the name of the AD service account.

Note: The syntax for the commands is case sensitive. You can also use the Netdom utility to establish a trust between the KDC and AD. The utility is located in the \support\reskit\netmgmt folder on the Windows 2000 Server CD.

ACTIVE DIRECTORY CONFIGURATION

Once the domain controller hardware and the Windows 2000 Server Active Directory software have been installed, we can begin to make the necessary modifications on the Windows 2000 server to tell Active Directory to use an external MIT Kerberos realm for authenticating domain users.

How is Kerberos is implemented within the Security model? Windows 2000 Server uses the Security Support Provider Interface (SSPI) between transport level applications and network security service providers. In an article in Microsoft System Journal, Mr. David Chappell describes the Windows 2000 security design:

> In Windows 2000, Kerberos is implemented as a Security Service Provider (SSP) that is accessible via the Security Support Provider Interface (SSPI). Applications can directly access Kerberos services through SSPI. Most applications won't use SSPI directly, however. For example, as shown in Figure 4, a Distributed COM (DCOM) or COM+ application uses the security interfaces DCOM provides. (Although it's not shown in the diagram, DCOM actually relies on the authenticated RPC interfaces, which are the direct users of SSPI in this case.) Other application protocols work in much the same way to shield developers from the details of SSPI. The key point to remember is that you don't need to worry about the details of how distributed security is provided—whatever SSP is used handles them more or less transparently.

Figure 4. Security Support Provider Interface[3]

Microsoft's implementation of Kerberos is contained within Active Directory and uses this SSPI interface to provide an easy mechanism to allow Active Directory to point to an external KDC for authenticating domain users. A non-transitive or one-way trust will be created between a Windows 2000 primary domain controller and the MIT Kerberos realm, to provide the access.

Trusts are implemented with Net Logon Remote Procedure Call (RPC) channels. Figure 5 below shows the trust relationships, between an External Kerberos realm and Microsoft's Active Directory.

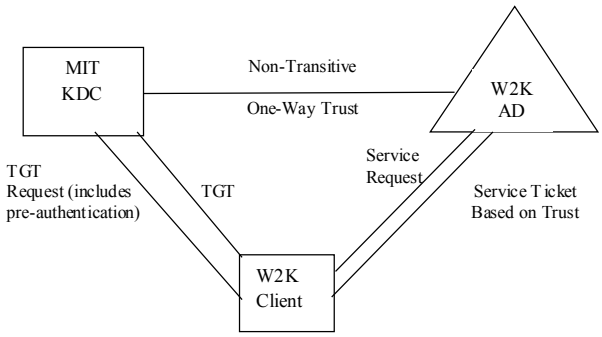

Figure 5. AD and Kerberos Trust Relationships

In order to setup this one way trust model the AD administrator must complete two tasks to tell AD how to allow authenticated users from an external KDC to access its resources. The fist step is to create a service account and then create a trust. Once created, this trust instructs AD to accept data (TGT's) from the trusted KDC.

Steps:

1. Create a service instance account in AD.
   a. Using AD Users and Computers create a user account. For this example call it kerbtgt.
   b. Create an account mapping for this user using ktpass from a command window or using the GUI in AD Users and Computers.
      Ktpass – princ service – instance@REALM -mapuser kerbtgt –pass password -out kdc keytab.

2. Modify the domain controller's registry to point to a Kerberos realm for authentication.
   a. Run ksetup.exe from a command window (a detailed explanation of ksetup can be found in the Workstation configuration section below).
   b. Ksetup /addkdc MYKDC.COMPANY.CO mykdc.realmname.company.com

3. To configure the trust complete the following:
    a. Programs➔Administrative Tools➔AD Domains and Trusts
    b. Right click on your domain select Properties➔Trust➔Add
    c. The passwords required are created on the external KDC see KDC Configuration steps (above).
    d. After entering the KDC password select OK.

4. After you have completed steps 1-3 above reboot, the domain controller.

Caution: It is possible to lock yourself out of AD even as administrator if you switch the name of the AD and the external KDC in step 2 above. This creates a loop where you attempt to authenticate to the KDC but the pointer is pointing at AD for authentication. In some instances the delete KDC command does not work. Thus, forcing you to rebuild AD using the ERD. Be careful using Ksetup.

Once the one-way trust is set up between AD and the KDC, we can look at how AD handles a Kerberos authenticated client. The client can present a Ticket Granting Service request to the Kerberos realm to obtain a TGT (see figure 5 above). SANS instructor Bill Boswell states, this initial request "Incorporates a pre-authentication mechanism to speed up initial TGT requests and to prevent denial of service attacks."[4] By default Windows 2000 requires all accounts to use pre-authentication. After the Authentication Service authenticates the user it provides a Ticket Granting Ticket back to the client to requests services from an Active Directory Domain Controller. The TGT contains a timestamp valid for 10 hours. The advantage is the user will only need to present his/her logon credentials to the KDC once during the session (average work day). It's important to keep in mind that all this occurs in the background, all the user sees on the display is the Windows logon splash screen.

Once the client receives the TGT, it sends this service ticket to AD, which in turn determines the groups a user belongs to and populates a Security Access Token (SAT). This token also contains a Security ID (SID) number for that user and a list of rights the user has on the network. The SAT is sent back to the client over the network via packets. As long as the token remains under two packets in size it is sent via UDP, otherwise it is sent over a TCP connection. Tokens seldom grow over two packets in size unless the user belongs to more than one hundred groups. The client than presents this access token to the resource servers it wants to gain access to. Kerberos uses port 464 for passwords and TCP or UDP port 88 for secure authentication between the KDC & AD.

Since the user has already been authorized to use the network, no other logon credentials need to be transmitted to other resource servers. Any resource server that trusts AD will be presented with a SAT in this manner. As long as the client was a member of the group when it obtained the SAT the user can request access to that service. All tickets and tokens are stored in the credentials cache in volatile memory protected by the Local Security Authority (LSA). According to Microsoft documentation: "The credentials are never paged to disk. All objects stored there are destroyed when the security principal logs off or the system is shut down."[5]

WORKSTATION CONFIGURATION

Once the KDC and AD modifications have been made the client machines that will be using the Kerberos KDC for authentication need to be modified. The registry value named KdcNames needs to be added to each registry on every workstation.

Hive: HKEY_LOCAL_MACHINE
Key: System\CurrentControlset\Control\Lsa\Kerberos\Domains
Value Name: KdcNames
Value Type: REG_MULTI_SZ

Microsoft has provided a utility on the Windows 2000 Media CD in the /Support/Tools/Support directory to assist administrators in adding this registry entry. This command line utility is called ksetup.exe. The table below outlines the syntax of this command.

Syntax	Description
SetRealm DnsDomainName	Set name of RFC1510 Kerberos Realm
/MapUser Principal Account	Map Kerberos Principal to account (* = any/all)
/AddKdc RealmName KdcName	Add additional KDC address for the given realm
/DelKdc RealmName KdcName	Delete instance(s) of KDC address for the realm
AddKpasswd Realmname KpasswdName	Add Kpasswd server address for a realm
/DelKpasswd Realmname KpasswdName	Delete Kpasswd server address for a realm
/Server Servername	Specify name of a Windows 2000 machine to target changes
/SetComputerPassword Password	Set the local machine's password
/Domain DomainName	Use this domain (blank for domain in your logged-on domain
ChangePassword OldPasswd NewPasswd	Change logged-on user's password via Kpassword

Figure 6. Ksetup Syntax

The gray highlighted line in figure 6 shows the primary function of this command. To make the necessary registry modifications open a command prompt window and run the following command:

Ksetup /addkdc MYCOMPANY.COM mykdc.mycompany.com

Where MYCOMPANY.COM is the name of the Active Directory domain and mykdc is the name of the external MIT Kerberos Realm.

Caution: This registry value only needs to be added once. Ksetup does not check for redundancy. Every time ksetup is run on the same machine it'll simple add another value, to the registry.

For a large organization, physically touching every desktop to make the registry change can be a large, daunting task. Fortunately, this is where AD shines over NT 4.0. The recommended method for making the registry change on multiple machines within a domain is to use Group Policy Objects (GPO's) to push the settings out. The optimum location for this GPO is to use a machine startup script that is run on individual workstations within the domain at boot time.

Windows 2000 and Unix machines are the only clients at the time of this writing that can take advantage of Kerberos to authenticate. Windows XP is slated to have this functionality built in upon release. Any pre-Windows 2000 operating system including Win9x, ME, or NT 4.0 and any other OS (i.e. Macintosh) will use NTLM v1 by default to authenticate to AD. Microsoft has created an application called **Dsclient** to allow older clients to take advantage of some features of AD. However, Kerberos authentication is not part of this extension package. It appears that Microsoft has made a business decision to leave these clients behind by not providing Kerberos authentication with these extensions.

The structure of the TGT received from the KDC comes in two parts, the client data and the TGS or KDC information. Figure 7 below outlines the major fields of the TGT that is sent back to the client after the AS approves the user's logon credentials. There are some other fields that are not shown here, the Figure has been simplified for clarity. Readers should refer to RFC 1510 for a more detailed explanation of the "Kerberos Protocol."[6]

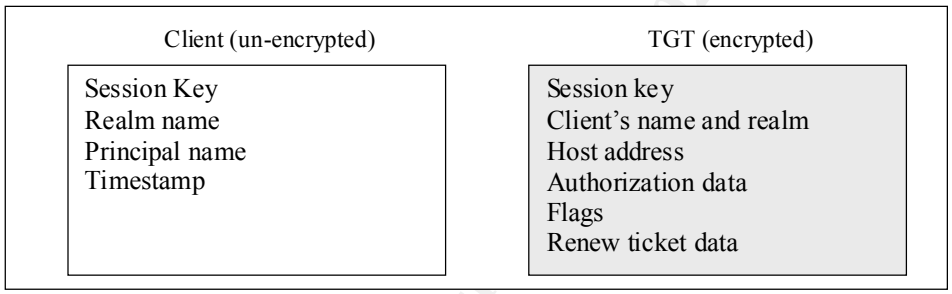

Figure 7. Ticket Granting Ticket Packet

The TGT portion of the ticket shaded in figure 7 is returned encrypted with the Kerbtgt password hash, while the client side is transmitted in clear text.

TOOLS

There are a number of tools available to assist administrators in ticket management or troubleshooting individual problems. A brief description of each tool and screenshots are detailed below:

GUI Tools:

- Leash32 – available from MIT.
- Kerbtray – available on the Windows 2000 Resource Kit.

Command Line tools:

- Klist – available from MIT or the resource kit.
- Kdestroy – available from MIT.
- Kinit – available from MIT.
- Ktpass – available on Windows 2000 media.
- Netdom – available on Windows 2000 media.
- Gpresult – available from Microsoft fee tool web site.

Leash32

A third party utility that allows users to view their available Kerberos tickets is called Leash 32. Leash 32 is available from MIT.

Download and install the software then follow these steps to use Leash32 to view your tickets:

Step1:

After downloading and installing Leash 32 select:

Start➔Programs➔Kerberos Utilities➔Leash32

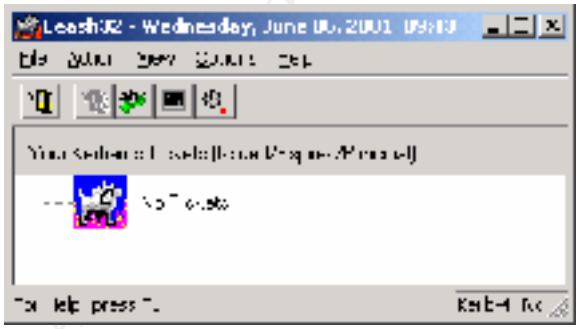

Figure 8. Leash32

Step 2:

Select **A**ction➔**G**et Tickets

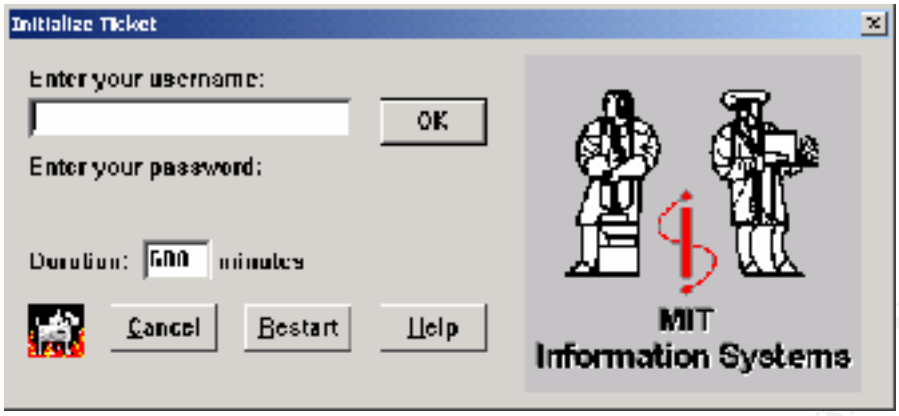

Figure 9. Leash32 Password Screenshot

Step 3:

Enter Your Username Select "ok" then Enter your Password in the next dialog box Select "ok"

Step 4:

Click on the plus sign to expand the ticket

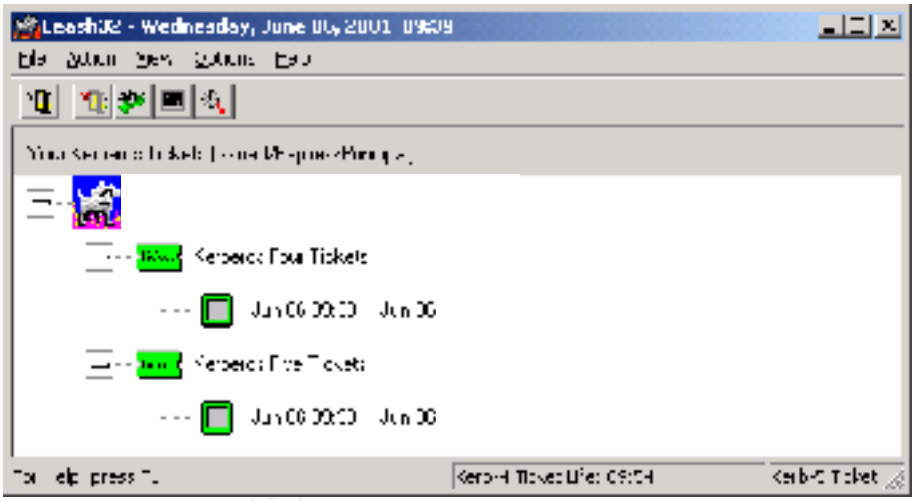

Figure 10. Leash32 Ticket Screenshot

Once you have your tickets you can use the GUI icons to renew or destroy your tickets, or change your password.

In the above example we have two Kerberos tickets one for Kerberos version 4 and one for Kerberos version 5. Obtaining these tickets allows the user to access different resources that are "Kerberized" (for instance e-mail or web based services) without having to re-authenticate to those services. "Kerberized" applications include any application or service that uses the Kerberos protocol to authenticate clients.

Kerbtray

This is a Microsoft tool located in the Windows 2000 Resource CD. Kerbtray provides much of the same functionality as Leash32 allowing the user a graphical interface for viewing Kerberos tickets. After starting kerbtray an icon will be placed on the toolbar. If you right click on the icon you can list ticket information or to purge tickets.

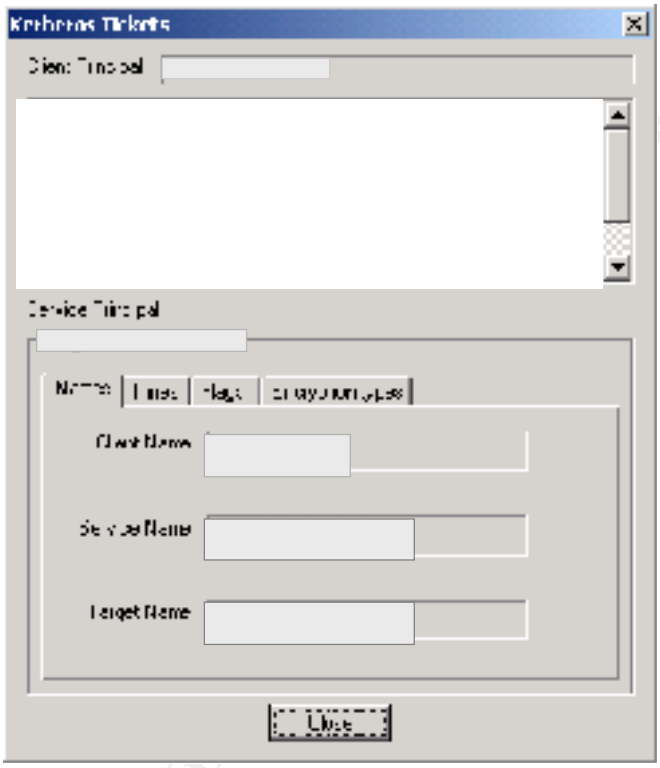

Figure 11. Kerbtray

In addition to the GUI tools described above there are a number of command line utilities that can be used to perform many of these same functions.

Klist

Klist is included on the Resource kit, this utility provides the same functionality that the GUI kerbtray does but in a command line interface. It displays the contents of the default cache. Using the –tickets option, Klist will display the total number of tickets the client machine has available, the server from which the ticket originated (TGT); the ticket encryption type; and the end and renew times of each ticket.

Kdestroy

Kdestroy will allow users to kill active tickets in the cached credentials memory space of Windows 2000. This utility also destroys a principal's login context and credentials.

Kinit

With Kinit a user can change their Kerberos password from a command prompt. This takes effect immediately. The utility can also be used to refresh or renew tickets using the –r option.

Ktpass

A command line utility that sets up account mappings for Kerberos services. Ktpass also is used to configure the one-way trust between AD and the KDC.

Netdom

Netdom is a command line tool that allows the user to view and manage all Windows 2000 domains and trust relationships. With Netdom you can establish trust relationships between domains (one- or two-way), including trusts for the following domain types:

- Windows NT domains.
- Windows 2000 parent and child domains in a domain tree.
- The Windows 2000 portion of a trust link to a Kerberos realm.

This can be a useful tool in determining which domains trust each other and the Kerberos authentication path (see Multiple Domain Environments below).

Gpresult

This tool that allows the user or an administrator to view what policy settings have affected the local computer and user account. It lists the distinguished name; date and time policy settings were applied. Including where it received, the GPO for the Registry, Scripts, Security, EFS Recovery and Application Management.

OTHER CONSIDERATIONS

Multiple Domain Environments

To this point our discussion has centered on making the configuration changes to use an external MIT KDC to authenticate users in an Active Directory single domain model. Some interesting developments occur when you expand this model to include multiple domains.

Each time you create a new domain tree in a forest, a trust path is automatically created between the forest root and the new domain tree. This trust path allows the authentication scheme to flow to all domains in the forest. When a user attempts to gain access to a resource in another domain in the new forest the user's computer contacts the domain controller where the user account is located and initiates a session ticket. The user's KDC also issues a referral ticket to the new forest domain. The client SAT follows this Kerberos protocol trust path up the tree structure until it locates the domain in which the resource resides. At this point the domain issues a request to the external KDC for authentication.

For example:

A Windows 2000 forest root domain consists of a tree Xcompany.com and a domain called production.xcompany.com. You create a new tree called suppliers.Xcompany.com. Once you create the second tree a two-way transitive trust is automatically created between the primary root domain of Xcompany.com and the new tree called suppliers.xcompany.com.

A user in the suppliers domain needs to gain access to a server in the production domain the following authentication process occurs.

The client asks the local KDC for a session ticket and a referral ticket to the other tree. Both tickets are sent back to the client. The client then presents the referral ticket to the KDC in the forest root domain. The forest root domain supplies a session ticket to the client telling it which domain to look in. The user presents the session ticket to the KDC in the production.xcompany.com domain. At this time the Kerberos authentication is initiated. If the password is correct then access is granted to the server and a TGT is sent back to the client.

Native Mode vs. Mixed Mode

The use of an external KDC for authentication described above will work in either native mode or mixed mode. AD can maintain backwards compatibility to NT 4.0 domains, which is called Mixed mode. Native mode is when all domain controllers, resource servers and client machines have been converted to Windows 2000. To enhance security it is recommended that enterprises move as quickly as feasible to native mode when using an external Kerberos realm for authentication.

Profiles

After a user logs into AD the first time a profile is created and this becomes the default user profile. If the user then wanted to use an external Kerberos realm for authentication the profile will remain the same. All the user would need to do is run ksetup on the workstation and select the external KDC name from the drop down list in the logon screen.

Known Interoperability Limitations

There are some widely known issues related to using an external Kerberos realm to authenticate AD users. Some of these may present concerns in your organization and cause you to lift an eyebrow. Others may not. Regardless, it is important to be aware of them and then decide how it might impact your network before implementing the use of an external KDC. Limitations identified by Microsoft are outlined in the "Step by Step Guide to Kerberos 5 Interoperability" include:

- Only DES-CBC-MD5 and DES-CBC-CRC encryption types are available for MIT interoperability.
- Hierarchical realm support for cross-platform trust between the Windows 2000 and MIT Kerberos realms is not included; however, transitive trust is supported between Windows 2000 domains in the domain tree.
- The KDC does not support post-dated tickets.
- Any upgraded user accounts and the administrator account in a new domain must have the password changed before non-Windows Kerberos clients can use them. [7]

Other limitations include:

- Extensions to the Kerberos 5 protocol that includes support for public key certificates, which can provide support for smart card authentication. This is not part of the IETF Kerberos Protocol.
- Backward compatibility with pre-windows 2000 clients. While users can authenticate against the external realm, AD has no method of populating the SAT for these clients, leaving them clueless as to what they can access.
- There is no client support for Macintoshes.
- If laptop users authenticate to the KDC then remove the laptop from they will not be able to log in locally to the desktop since Kerberos does not cache logon credentials.

Account Mappings:

It is possible to use an external MIT KDC to authenticate Windows 2000 clients and not have an Active Directory server in operation. This is useful when you have a large infrastructure of Unix applications that are "Kerberized" and want to incorporate Windows 2000 on some clients, but do not want to install a Windows 2000 domain infrastructure.

For instance if you want to deploy Windows 2000 notebooks to senior management but do not want to install and maintain a Windows 2000 Active Directory domain. Notebook users can still authenticate to the existing Kerberos KDC and continue to access the "Kerberized" applications by establishing mappings between the user account in the Kerberos realm and a local user on the Windows computer. To set up account mappings on a Windows 2000 machine complete the following:

Steps:

Go to Users and computers (make sure Advance Features are turned on)

Right click to select the user➔Select name mappings➔Kerberos names
Select Add➔Enter the users Kerberos Principle name i.e. User@KDC name➔ok➔ok

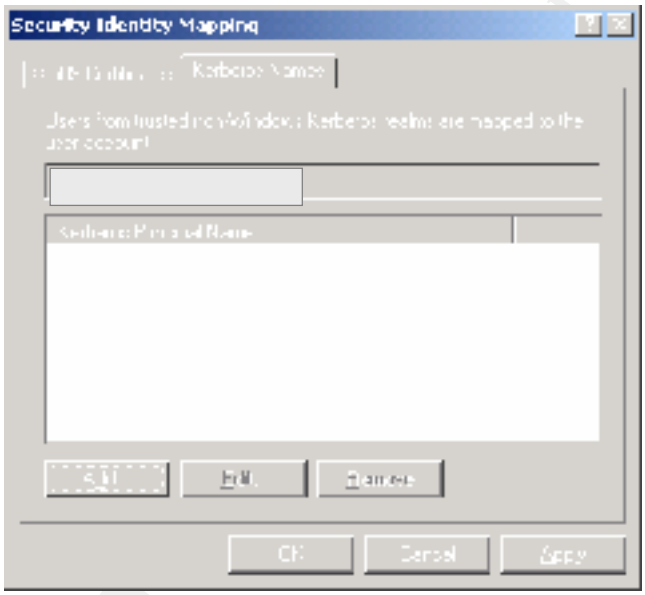

Figure 12. Kerberos Name Mappings

This mapping will then allow Windows 2000 client notebooks to use the existing Kerberos KDC for authentication. Unix resources will still be available to the notebook clients. This mapping will have to be done manually on each notebook.

CONCLUSION/SUMMARY

As this document has outlined, it is possible to incorporate Windows 2000 and a MIT Kerberos infrastructure. Organizations may want to utilize an external realm for authentication to achieve a number of different goals. Benefits from using this model for authentication include (in no particular order):

- Not becoming solely dependent on one vendor (Microsoft) for both your network operating system and your authorization infrastructure.
- Not having your authentication data maintained by one group within your organization. With an external realm duties can be separated allowing the infrastructure group to manage AD, while the security team maintains the Kerberos infrastructure.
- Enhancing security by dramatically reducing the chance of "man in the middle" attacks that NTLM authentication is susceptible to.
- Tickets expire automatically, which can reduce the amount of time an attacker has credentials on the network.
- Using an external KDC, all authentication data can be kept to a limited number of machines.
- Depending on how AD is implemented within an organization, a single MIT Kerberos realm can provide authentication services to multiple forests.

Of course security is a balancing act of trade offs. There are also some disadvantages to implementing authentication using this model, again in no particular order:

- AD administrators are relying on a virtual "trust" connection to maintain logon credentials.
- With an external KDC you need to maintain additional hardware and software that can be a point of failure.
- If one group is maintaining the Kerberos infrastructure and another AD, political and individual personalities can come into play. AD administrators may feel they are giving up too much control.
- If the link between the AD domain controller and the KDC is interrupted users will be able to authenticate but not access resources.
- Registry modifications on all Windows 2000 client machines need to be made. Ksetup must be run locally or via GPO and SP 2 or later should also be installed.
- Only Windows 2000 clients can authenticate to a Kerberos realm; all other clients will use NTLM.

Given the benefits and drawbacks outlined above, there are no clear-cut answers as to how authentication in an organization should be handled. Every organization will have too carefully consider the options if they want to maintain an external MIT Kerberos realm to authenticate users with AD. Organizations will have different security requirements that determine how the keys to the kingdom are issued and maintained.

ENDNOTES

[1] Ashworth Robert. “Kerberos – Windows 2000 Authentication”. SANS. 10 May 2001. URL: http://www.sans.org/y2k/practical/Robert_Ashworth_GCNT.doc. (9 June 2001).

[2] Murphy Michael. “Authentication in Windows NT and Windows 2000”. SANS. 26 May 2001. URL: http://www.sans.org/y2k/practical/Michael_Murphy_GCNT.doc. (9 June 2001).

[3] Cappell, David. "Exploring Kerberos, the Protocol for Distributed Security in Windows 2000". Microsoft System Journal. August, 1999. URL: http://www.microsoft.com/msj/defaulttop.asp?page=/msj/0899/kerberos/kerberostop.htm. (12 June 2001).

[4] Boswell, Bill. "Windows 2000 How it Works". Baltimore, SANS. May 2001. 112.

[5] “Microsoft White Paper Windows 2000 Kerberos Authentication". Microsoft. May 1999. URL: http://www.microsoft.com/windows2000/docs/kerberos.doc. (15 June 2001).

[6] Kohl, J. T. and Neuman, B.C. "The Kerberos network authentication service. Internet RFC 1510". Isi.edu. September 1993. URL:ftp://ftp.isi.edu/in-notes/rfc1510.txt. (12 June 2001).

[7] “Microsoft White Paper Step-by-Step Guide to Kerberos 5 (krb5 1.0) Interoperability”. Microsoft. 10 January 2000. URL: http://www.microsoft.com/windows2000/techinfo/planning/security/kerbsteps.asp. (20 June 2001).

REFERENCES

"Implementing and Installing Microsoft Windows 2000 Directory Services". Redmond: Microsoft Training, June 2000.

"IT Toolbox Security". Ittoolbox URL: http://security.ittoolbox.com/nav/t.asp?t=388&p=388&h1=388 (18 June 2001).

"Kerberos: The Network Authentication Protocol". MIT. 9 December 2000. URL: http://web.mit.edu/kerberos/www/ (9 June 2001).

"Microsoft White Paper Windows 2000 Kerberos Authentication". Microsoft. 9 July 1999 URL: http://www.microsoft.com/windows2000/library/howitworks/security/kerberos.asp. (18 June 2001).

"Q article Q288337; Only First Key Distribution Center in a Configured Kerberos Realm Is Used". Microsoft. 27 May 2001. URL: http://support.microsoft.com/support/kb/articles/Q288/3/37.asp. (18 June 2001)

"The Kerberos Network Authentication Service". Isi.edu. URL: http://www.isi.edu/gost/gost-group/products/kerberos/. (9 June 2001).

Tung, Brian; "The Moron's Guide to Kerberos". Isi.edu. 19 December 1996. URL: www.isi.edu/~brian/security/kerberos.html. (15 June 2001).

This page intentionally left blank.