Kerberos interoperability issues

Paul B. Hill
Massachusetts Institute of Technology

Abstract

MIT's computing environment is a heterogeneous environment that has used Kerberos as a primary authentication method for over a decade. Instead of migrating our existing KDCs to Windows 2000 we have chosen to use cross realm trust to support our Windows 2000 computing environment. During our deployment project we have encountered some interoperability problems and have worked with Microsoft to resolve these. We have also encountered protocol extensions that have been used by Microsoft and we have been working with Microsoft under the umbrella of the IETF to have these documented. Some of the problems were only identified and resolved by analyzing network traffic.

1. An Introduction to Kerberos

Kerberos is a secure authentication protocol for use in distributed computing environments. When used ubiquitously within a computing environment it can also provide single sign-on capabilities. Kerberos was first introduced to the USENIX community in 1988 in the *Proceedings of the Winter 1988 Usenix Conference* and *Proceedings of the Usenix Workshop on Workstation Security*. Articles on Microsoft's implementation of Kerberos have also appeared in the November 1997 and May 1998 issues of *;login.*

The protocol was initially developed at MIT as part of Project Athena. The work was funded by IBM and Digital. The source code was made available to others and the copyright allows the development of derivative and commercial work with few restrictions.

MIT remains very active in the development of Kerberos. Change control of the protocol is now the responsibility of the IETF. Version 5 of the protocol is currently defined by RFC 1510. Revisions to the protocol are in progress and there are related drafts closely associated with Kerberos being developed as well.

MIT had several goals when developing Kerberos. These included:

- raising the awareness of security issues in a distributed computing environment, especially those that rely on packet technology
- providing a solution to security problems that nobody else was attempting to address at the time
- encouraging vendors to adopt Kerberos so the we could purchase secure systems for our own needs

Although this paper does not explore many details of the protocol, some basic definitions will be helpful for understanding.

User - A human being who wishes to a use a computer system.

Service - An abstract specification of some actions to be performed. The actions are performed by a program or set of programs running on a computer which is accessible over the network.

Principal - An entity which can both prove its identity and verify the identities of other principals who wish to communicate with it; each **user** and each **service** registered with Kerberos is thus a principal.

Ticket - A block of data which, when given to a user, enables him to prove his identity to a service.

Realm - an authentication domain or authentication namespace.

TGT - A ticket granting ticket. A special Kerberos ticket which enable a user to get other service specific tickets.

KDC - Key Distribution Center. The Kerberos server that provides tickets to users.

1.1. MIT's use of Kerberos v4 and v5

MIT uses Kerberos throughout its academic and administrative computing environments. Our computing environment is heterogeneous and includes many versions of UNIX, UNIX variants, IBM's VM,

Apple Macintosh OS, and most versions of Microsoft operating systems.

Although Kerberos v4 and v5 do not interoperate both version 4 and version 5 of the Kerberos protocol are used at MIT at this time. We expect this situation to exist for many years. Support of both protocols may be achieved by one of two mechanisms. The MIT Kerberos 5 release can speak the Kerberos 4 protocol, assuming it was built with the `"--with-krb4"` option (which is the default). Or separate KDCs can be maintained for each protocol; the database information may be propagated between the different KDCs to simplify administration.

The applications used consist of a mixture of internally developed applications and commercially available packages. Examples of the commercially available packages include Transarc's AFS, which uses Kerberos version 4, and SAP R3, which uses Kerberos version 5.

Despite supporting both version 4 and version 5 of the protocol the MIT Kerberos environment is remarkably simple. MIT does not use DCE anywhere in its infrastructure. At this time only a single Kerberos realm is used on campus for all supported services. This is the ATHENA.MIT.EDU realm. Although many other realms exist on campus they have no operational impact beyond a very small set of users. They only exist for testing, research, or as educational exercises. Cross realm trust relationships are not established with departmental Kerberos realms.

The Athena realm is maintained by Information Systems and of course runs the MIT implementation of Kerberos.

1.2. Microsoft's support of v5

The default authentication protocol used by Microsoft's Windows 2000 operating system is Kerberos version 5. The Kerberos protocol is just one of the security protocols supported by the operating system. Others include NTLM for backwards compatibility, SSL / TLS for public-key authentication, SPNEGO for security protocol negotiation, and IP Security (IPSec) for network layer security.

Windows 2000 only supports Kerberos version 5. There is no support for Kerberos version 4, nor is DCE style cross-realm trust supported. If an organization requires Kerberos version 4, or DCE security, support the organization must examine its interoperability options and develop a strategy.

Microsoft provides and uses a Security Support Provider Interface (SSPI) to the Kerberos protocol. The Kerberos security provider may be used by any application designed to use SSPI for network security. Microsoft is using this support to secure extensive portions of any Windows 2000 infrastructure. A number of services support Kerberos authentication in Windows 2000. Here is a partial list:

- authentication to the Active Directory using LDAP for queries or directory management
- CIFS/SMB remote file access protocol
- distributed filesystem management and referrals
- secure DNS address update
- print spooler services
- optional IPSec host-to-host authentication in ISAKMP/Oakley
- reservation requests for network Quality of Service
- intranet authentication to Internet Information Server
- authentication of public-key certificate requests to the Microsoft Certificate Server for domain users and computers
- remote server or workstation management using authenticated RPC and DCOM

The support of Kerberos over such a wide range of systems does imply that Microsoft has a high level of commitment to the SSPI interface and that this is a flexible interface. The SSPI interface is very similar to the IETF's GSS API, RFC 2743. Microsoft has stated that in order to have all of these services support Kerberos the most complex change was to the SMB server, which did not use SSPI prior to Windows 2000.

1.3 Microsoft's Windows 2000 Kerberos implementation

There are several more important aspects to Microsoft's Kerberos implementation that system architects should keep in mind.

Every Windows 2000 Domain Controller is a KDC. The KDC is a logical process that is part of the LSA process. It cannot be removed.

Active Directory, via LDAP, is the administrative interface to the KDC. Please note that the administrative interface to the KDC has never been standardized through the IETF.

Although not required by the Kerberos standard, Microsoft's implementation requires that the DNS domain and Kerberos realm names be identical. Per the

current standards DNS names are case insensitive and the Kerberos realm name will always be upper case. In the context of Windows 2000, a Domain encompasses both the DNS domain and the Kerberos realm. This overloading of terms can become very confusing when talking about configuration and support issues.

The Microsoft libraries locate the KDC using DNS service location records instead of relying on local configuration files. There is an IETF draft in progress to standardize this behavior. When using non-Microsoft realms for authentication local client configuration information is also supported.

To foster interoperability Microsoft implements DES-CBC-CRC and DES-CBC-MD5 encryption types. These are 56 bit symmetric key algorithms that are used by other Kerberos implementations. The implementation does not support the MD4 checksum type. Microsoft's preferred encryption type is RC4-HMAC. This is currently documented in an informational IETF draft.

Microsoft's use of a new encryption type had two motivations. Most importantly upon upgrading from a Windows NT 4.0 environment to Windows 2000, accounts will not have the appropriate DES keying material to do the standard DES encryption. If a new encryption type was not supported an organization would have to reissue passwords for all user accounts. In large environments this would be unacceptable. The new encryption type also helped Microsoft resolve some of the possible barriers to export of the software imposed by US regulations. Early in the development process, 56 bit DES encryption could not be exported.

Microsoft uses structured service naming conventions. This does raises some issues for developers wishing to use the GSS API libraries to support multiple operating systems.

Microsoft's implementation of the Kerberos protocol supports and assumes the use of authorization data in tickets. This is compliant with the current proposed draft revisions to Kerberos that the IETF is working on. Other implementations that include authorization data within the Kerberos tickets are DCE and Sesame.

The Windows 2000 authorization data is ignored by current UNIX implementations. Although Microsoft has released information about their use of the authorization field it appears that the Kerberos community is precluded from writing any code that can use this information in any way.

The Windows 2000 KDC supplies the authorization data that is placed into the tickets. Depending on the type of ticket the authorization data included may consist of user SIDs, global or universal group SIDs, or domain local group SIDs.

Application services that receive a ticket are able to extract the list of SIDs and use this information to determine what the client is allowed to do based on the Windows 2000 group membership information.

2.0 Interoperability scenarios

When we talk about interoperability we will use Domain to mean a Microsoft Windows 2000 Domain which by definition include a Kerberos realm. We will use the term realm to mean a Kerberos realm that is not a Microsoft Domain.

There are several interoperability scenarios that could be considered, not all are listed here.

- Windows 2000 domain without a Microsoft KDC
- Kerberos clients in a Windows 2000 domain
- Kerberos application servers in a Windows 2000 domain
- Standalone Windows 2000 systems in a Kerberos realm
- Using a Kerberos realm as a resource realm
- Using a Kerberos realm as an account domain.

The first scenario is not supported by Microsoft and not available to anybody at this time. Providing this option would require a 3rd party Domain Controller replacement or functionally equivalent clone. It appears likely that Microsoft will try to prevent any third party from implementing a solution that enables a customer to choose this option.

The Windows 2000 domain security model depends on the authorization information being present in the ticket. Microsoft is asserting intellectual property issues on their use of the authentication field and apparently preventing others from developing compatible implementations. Furthermore, the Microsoft KDC is tightly integrated into the Active Directory and LSA process.

2.1 Kerberos clients in a Windows 2000 Domain

Some organizations may find this option attractive. Suppose for example that an organization has a heterogeneous computing environment but does not use Kerberos today. If Windows 2000 is used for account management and authentication its use can be leveraged to improve the security of the other computing platforms as well.

Kerberos version 5 client libraries and applications are available for most versions of UNIX, Linux, and Mac OS. The versions of kinit, klist, kdestroy as well as other applications from the MIT distribution have been tested against the Microsoft KDC. No code changes were required in order to make the applications work.

The MIT Kerberos libraries will ignore the contents of the Microsoft authorization field, per the specification. The other operating systems would not be able to use the Microsoft SIDs to determine the intended authorization access. The authorizations methods supported by most UNIX-based services are application specific today.

Applications that use the GSS API and the Kerberos v5 mechanism, will also continue to function is this type of deployment.

Note that this scenario will not be suitable for organizations that intend to use DCE security for application services. The DCE libraries will be expecting different authorization data within the field and will not be able to use the data supplied by the Microsoft KDC. Nor is this scenario appropriate for organizations that need to support version 4 of the Kerberos protocol.

2.2 Kerberos application servers in Windows 2000 Domain

Everything that was stated in the preceding section applies to this scenario as well. There are also other situations where this configuration becomes attractive.

Suppose that you have a UNIX database server that supports Kerberos authentication and you would like to provide users within your Domain access to the database via a Web interface. Since Internet Explorer and IIS support authentication using SSPI it is possible to create a multi-tier application that uses Kerberos authentication.

You have to create and manage service accounts for the UNIX servers. In this case the computer accounts are the same as Windows 2000 user accounts. You may find it useful to create a separate organizational unit (OU) within the AD for these accounts.

You also have to create and install a keytab file on the application server. Microsoft provides the Ktpass.exe program as part of the Windows 2000 Resource Kit. This program can be used to generate the keytab file. You will have to copy this onto the correct host and merge it into the UNIX keytab file. Be sure to copy the file from the location where it was created to its destination in a secure manner.

Microsoft has published a Kerberos interoperability paper that describes the creation of the computer accounts and use of the keytab program quite thoroughly.

Note that in the case of a multi-tier application using IIS, the IIS might be trusted for delegation. This means that you could create a system that would use Kerberos authentication across all of the tiers.

2.3 Standalone Windows 2000 systems in Kerberos realm

This scenario is similar to that used by most UNIX centric Kerberos deployments today. If you have an existing Kerberos realm that provides application services and all network resources it may be attractive to you. It does assume that you plan to offer no Microsoft application services or network resources that support Kerberos authentication.

The Windows 2000 computers will not be members of a Microsoft domain. Local accounts will be used on each machine to establish an account mapping but authentication will be performed using a KDC that is not implemented by Microsoft.

Microsoft provides a Ksetup.exe utility as part of the Windows 2000 Resource Kit. This utility can be used to configure the realm information on each computer. The same program is also used to establish the local account mapping.

The local account mapping can be done on an individual basis where each account in the realm is mapped to a corresponding local account on the machine. This does not scale well.

An alternative is to map multiple individual accounts in the realm to a single account on the local machine. This may not be suitable for many environments.

2.4 Using a Kerberos realm as resource domain

Many sites have multiple user namespaces today. By this I mean that they manage user accounts for their UNIX operating systems independently from their user account management on NT, or other operating systems. If this works well for an organization the practice may continue after Windows 2000 is deployed. Despite this bifurcation there may still be a desire to provide services across the environments.

Suppose that one user population primarily uses Windows 2000 file and print services but many of the users also need to access some Kerberized services located on UNIX servers in a Kerberos realm, for example an IMAP service that supports v5.

By establishing a one way trust relationship between the Windows 2000 Domain and the Kerberos realm, such that the Kerberos realm trusts the Windows Domain, an organization can provide their Windows 2000 users access to the UNIX hosted IMAP service.

Users will initially authenticate to the Windows 2000 Domain Controller. When the SSPI enabled application needs to authenticate to the IMAP server the SSPI libraries will transparently perform the cross realm authentication and present the correct ticket to the IMAP server.

The application services in the UNIX realm will have to determine the access rights of the user. This will not use the Windows 2000 authorization data, instead the application server will resort to the methods that it normally uses.

If the organization wished to migrate to a uniform name space this strategy would still be useful. Over time the UNIX user accounts could all be migrated to the Windows 2000 Domain and the UNIX realm would only contain the service principal names for the UNIX based application services.

2.5 Using a Kerberos realm as an account domain

Now I'll focus on the deployment scenario that has occupied a great deal of time, and the time of John Brezak, Program Manager of Kerberos at Microsoft. I'll describe more of the details than have been covered in the other scenarios.

In this case the Kerberos realm will be used as an account domain and the Windows 2000 Domain will be used primarily to provide authorization data. This means that all of our users will initially authenticate to our existing Kerberos realm but we will still have an operational Windows Domain and all the services that it can provide.

Earlier I stated that MIT has only a single Kerberos realm that is used on campus for all supported services, and that cross realm trust relationships are not established with departmental Kerberos realms. Our Windows 2000 deployment is compelling us to change that slightly. We are creating a second Kerberos realm, in this case a Domain, and establishing a trust relationship with it. We are not going to make this a common practice since we wish to avoid a proliferation of realms that add no value to the community.

Our existing supported realm is ATHENA.MIT.EDU. It is used for authentication within the MIT.EDU DNS domain and the few DNS subdomains that we have.

We have created a WINDOWS.MIT.EDU Domain, which means that we now have a WINDOWS.MIT.EDU Kerberos realm and a WINDOWS.MIT.EDU DNS subdomain. We will not be providing dynamic DNS for our domain, and we are not placing all Windows 2000 machines into the windows subdomain. The only machines that will actually appear in our windows subdomain are the Domain Controllers, our RIS servers, and a few other miscellaneous servers. All of the DNS information for the Windows 2000 workstations will reflect that they are in the top level MIT.EDU DNS domain.

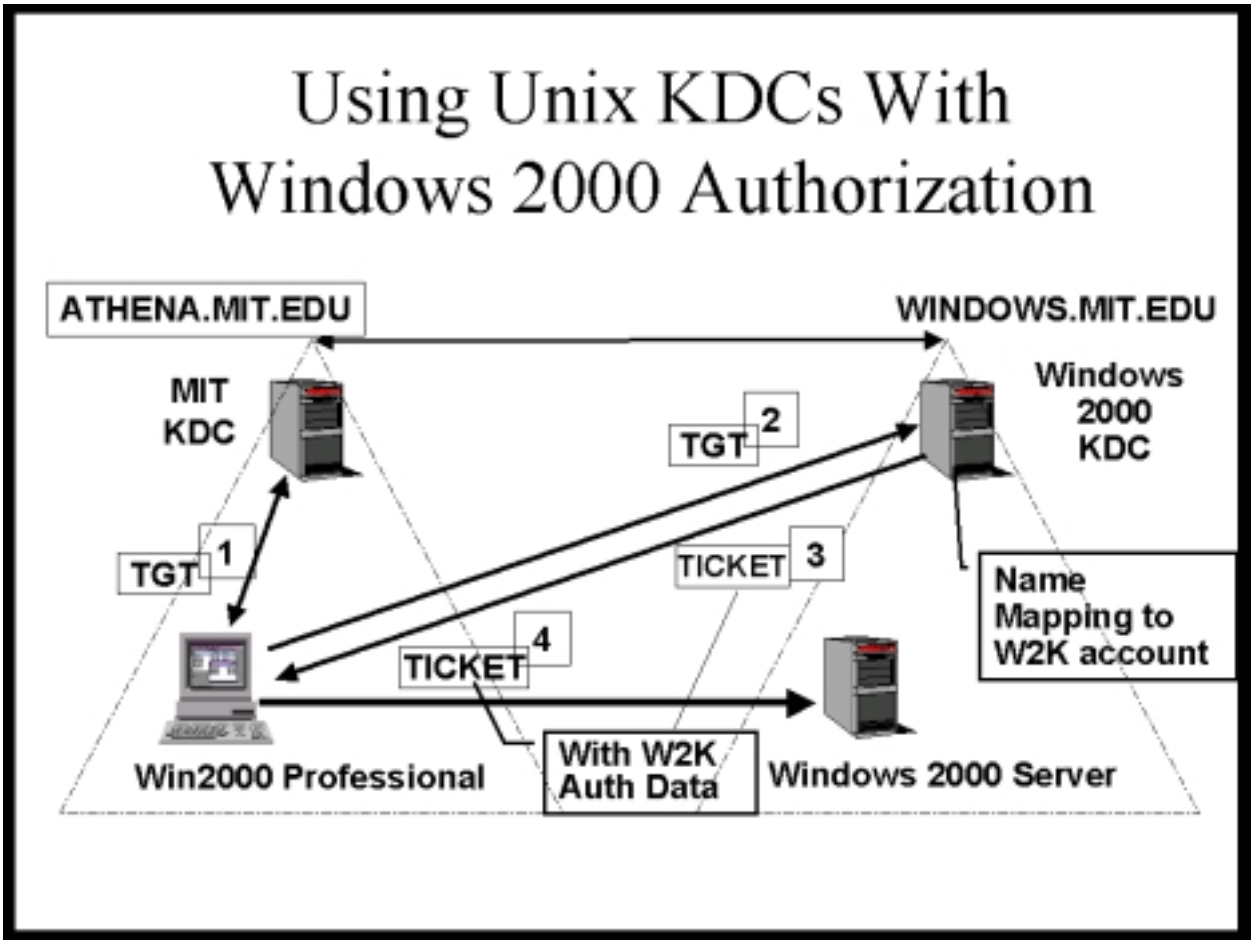
We then created a trust relationship between the realms so that the Windows Domain trusts the Athena realm. In order to make the trust relationship useful, each Kerberos principal in the Athena realm has a corresponding account in the Windows Domain. This is done using the altSecurityIdentities attribute within AD. Each Win2K account has this attribute populated with the corresponding Athena principal information. For example, the Win2K account pbh@WINDOWS.MIT.EDU has an altSecurityIdentities entry that contains Kerberos:pbh@ATHENA.MIT.EDU.

This configuration enables me to log into a computer that is a member of the Windows Domain as pbh@ATHENA.MIT.EDU. I will obtain my Athena TGT so that I can subsequently access resources in the Athena realm. I will also automatically obtain as needed a TGT and service tickets within the Windows Domain. The Windows tickets will include my

authorization data which domain applications can evaluate to determine my access rights to resources.

With the configuration and application requirements that have described so far there is no need to synchronize passwords between the Domain and the realm. A compromise to the security of the Win2K domain will not compromise any of our existing resources in the Athena realm.

The included figure shows a schematic representation the traffic that occurs when using this configuration.



2.5.1 "unknown passwords" in the Windows 2000 domain

Section 1.2 listed many Windows 2000 services that support Kerberos authentication, however not every Microsoft service or application supports SSPI today. For example, the Macintosh file services and the Macinstosh UAM.

If an organization needs to support the Macintosh file services within a Windows 2000 Domain, NTLM authentication will be used when accessing these services. A similar situation may exist when supporting "downlevel" clients, or application servers that do not use SSPI.

You have a couple of choices at this point. You can maintain separate passwords for your Domain and realm accounts, and hope that your users will not manually synchronize them. When doing this most applications will prompt the user for the Domain password when resorting to the NTLM authentication. Some applications may not support prompting of the user; in this case you will have to synchronize the passwords between your realm and your Domain.

Neither MIT nor Microsoft provide any tools to perform a password synchronization between a realm and a Domain. Reportedly, CyberSafe does have a tool for their Kerberos implementation that provides this feature.

There are two things that you should keep in mind if you are intending to synchronize passwords between a Domain and a realm. A compromised password in either environment can be used to gain access to

resources in the other environment; you will not easily be able to determine where down-level authentication is being used in your environment.

2.5.2 Kerberos referral issues

If you have tried this type configuration yourself, you may have encountered some problems. Just as release to manufacturing occurred I reported to Microsoft that this functionality was not working. It turned out that Microsoft had unknowingly been depending on a bug in the MIT distribution. The bug affected how ticket requests for unknown principals were handled and the implementation of realm referrals. A newer distribution from MIT had fixed the bug and broken the test configuration. The testing had also relied on a hierarchical name relationship between the Domain and realm names, which does not always apply.

Microsoft very quickly proposed a solution that was soon turned into the hotfix described in Microsoft knowledge base article Q253531.

Two configuration options are available with this hotfix.

- If your Kerberos realm does not support name-canonicalization, and the KDC returns a "principal unknown" error in response to a ticket request, the request will be retried to the Domain. This requires the computer issuing the ticket request to belong to a Domain.

- If your realm supports name-canonicalization it must return a referral if the principal is unknown. The client library will then use the referral information for a subsequent ticket request. The machine originating the request must have some registry information present so that the initial request will contain the name-canonicalization bit.

2.5.3 Kerberos name canonicalization issues

The preceding section started off talking about realm referrals but ended up talking about the name-canonicalization. They should be separate topics but their distinction has become somewhat blurred because of the way a bit in the Kerberos options field is being used.

Having the KDC perform name canonicalization is a relatively new concept, introduced by Microsoft. They have submitted a draft to the IETF. The draft has resulted in a lot of discussion, but little closure.

Traditionally Kerberos implementations have performed some name canonicalization on the client. When an unqualified DNS domain name is presented to the client library, the library turns it into a fully qualified domain name. The one problem is that a reverse resolution against DNS is often used and DNS is not secure.

Microsoft has argued that by performing the name canonicalization on the KDC the potential threats posed by DNS attacks are eliminated. This is true. It also true that Microsoft is doing more than their thesis would indicate.

A Microsoft KDC will return some tickets that look very odd to people from the UNIX Kerberos community. For example when requesting service ticket for the machine "foo" Microsoft may respond with a ticket for "FOO$" instead of "FOO.MIT.EDU". This provides Microsoft with some backwards compatibility with protocols that expect NetBIOS style names. It is a neat trick but it does lead to some other problems.

The client libraries are no longer able to perform a simple name matching during their ticket request prior to sending the request. Although a client may already have the needed service ticket, further ticket requests may be issued. Microsoft has acknowledged that this is a minor problem that they discovered late in the development cycle. I have also been told that they expect to address the problem in a future release.

One concern that has been raised by people outside of Microsoft is that using Kerberos as an authentication protocol is well understood, however the implications of using the protocol as a name resolution protocol or directory service are not well understood.

Microsoft has responded by saying that they are not using the protocol as a naming service or directory service. This is true. But I'll bet if you take 10 programmers aside and ask each one how they would solve the cache matching problem describe above, at least of few of them would come up with solutions that sound an awful lot like a naming service or a directory service.

The MIT distribution of Kerberos does not currently include any support for name-canonicalization or the generation of referrals. As mentioned, Microsoft has submitted a draft proposing this extension to the protocol to the IETF but there is no clear consensus within the working group on this proposal yet. As the work proceeds I do expect that the MIT distribution will eventually incorporate the functionality. In the meantime a separate patch will be made available for the MIT distribution so that individual sites may add this support as desired.

2.5.4 MIT library modifications to achieve single sign-on

So far most of our discussion in this section has focused on applications that use the SSPI however, other APIs exist for supporting Kerberos. MIT and other vendors provide Kerberos libraries and GSSAPI libraries that are used by various applications. The libraries and applications work well with the configuration described but the 3rd party libraries and Microsoft SSPI do not share a common ticket cache.

This means that subsequent to the initial Windows 2000 login, the MIT libraries do not have access to the TGT that was obtained by the Microsoft libraries.

Once again Microsoft has listened to the 3rd party development community and helped to provide a solution. Normally only the LSA process has access to the ticket Microsoft ticket cache. This means that hostile applications cannot steal the user's credentials and pass them on to something else. Microsoft has provided an API that enables a user process to copy the TGT from the Microsoft cache and store it in another cache.

MIT is currently working on modifying its Windows libraries to support this functionality. The first time the krb_sendauth function is called the library will copy the initial TGT from the Microsoft cache to the MIT ticket cache. The TGT will then be used to obtain the desired service ticket and complete the krb_sendauth call.

This should result in a single sign-on functionality for many applications. This strategy does not rely on a GINA, which in turn means that sites that rely on other GINAs will not encounter any configuration problem.

I expect this functionality will be available from MIT before the Fall of 2000.

3.0 Microsoft Network Monitor and Kerberos support

Debugging the problems encountered in our configuration and learning about the details involved have been greatly facilitated by the staff at Microsoft and the tools that they provided to us.

We used Microsoft's Network Monitor extensively. The Kerberos version 5 protocol uses ANS.1 encoding which normally makes the understanding the network traffic very difficult. Microsoft has developed a Kerberos parser DLL for their Network Monitor. They graciously allowed us early access to this tool. Their goodwill even extended to letting us redistribute it to a limited number of schools that had existing Kerberos realms and were working on a Windows 2000 deployment. If this parser has not already been released, I expect that it will be within the next several months.

4.0 How do you spell interoperability today?

Ultimately interoperability cannot be declared by a vendor or specified by a standard. The true measure of interoperability can only be performed by each customer or user of a protocol. The question to be asked and answered is, "does this meet my needs?"

This paper only presents some of the possible interoperability scenarios that Microsoft and other Kerberos implementations will be faced with.

If your metric is, "can I choose which vendor provides my KDC?" or "can I provide my users with the functionality of a Windows 2000 domain, without running a Microsoft Domain Controller?", then you are likely to be disappointed by the interoperability provided.

On the other hand, by many other metrics Microsoft has provided a highly interoperable Kerberos implementation that will meet many customers needs.

5.0 References

Windows 2000 Kerberos Interoperability, Microsoft whitepaper <http://www.microsoft.com/WINDOWS2000/library/howitworks/security/kerbint.asp >

Kerberos 5 (krb5 1.0) Interoperability, Technical Walkthrough, Microsoft whitepaper <http://www.microsoft.com/technet/win2000/kerbstep.asp >

Kerberos authentication in Windows NT 5.0 domains, Peter Brundrett, ;login:, the magazine of the USENIX Association, May 1998 Special Issue on Security

J. G. Steiner, B. Clifford Neuman, and J.I. Schiller. Kerberos: An Authentication Service for Open Network Systems. In *Proceedings of the Winter 1988 Usenix Conference*. February, 1988.

Microsoft knowledge base article Q253531 <http://support.microsoft.com/support/kb/articles/Q253/5/31.ASP >