Global Information Assurance Certification Paper

Sans Spring Break 2001 Windows NT/2000 Security Version 2.1a

Windows 2000 Implementation of Kerberos: An Overview

Submitted by Sidney Faber

Introduction

This paper examines the Windows 2000 implementation of Kerberos. Despite the small number of configurable options to support Kerberos, this change in authentication scheme has a significant security impact. The apparently seamless migration from NTLM to Kerberos hides the complexity of this new environment. The Kerberos authentication scheme is elegant in its simplicity and robust in its security, but a misunderstanding of some basic concepts could still lead to mismanaged authentication.

Due to the complexity of a full Kerberos implementation, this paper presents many unanswered questions. Any implementation of this protocol is difficult to examine since much of the data seen on the wire is encrypted. However, the goal of this paper is to provide a technical background of how Microsoft has apparently implemented the specifications of RFC 1510 based solely on authoritative documentation, and to outline areas of interest that deserve additional focused research.

Kerberos Overview

The following sections outline the three basic message exchanges in Kerberos. The Authentication exchange between a client and the Kerberos server verifies the client by proving the client knows a secret. This is the only exchange that requires knowledge of a password. The ticket-granting server exchange allows an authenticated client to request a "ticket" from the Kerberos server to access a service. The application authentication exchange allows a client and server to authenticate each other without any interaction with the Kerberos server.

This paper assumes a basic familiarity with the Kerberos protocol, for an excellent background, take a few minutes to read the paper by Bill Bryant, Designing an Authentication System: a Dialogue in Four Scenes (Reference 3).

Authentication and Authorization

When discussing Kerberos, it is important to understand the distinction between authentication and authorization. Authentication is the process of establishing identity (I have an ATM card and a PIN, so I can use an automatic teller). Authorization is the process of limiting the actions by a managed object (I can only withdraw money from the ATM only if I have sufficient funds). Many documents blur the distinctions between these two services.

Windows 2000 manages authorization with a Security Identifier (SID). The SID is a unique "name" assigned to a consumer (anything that might need to use a resource). SIDs are assigned to accounts and are guaranteed to never repeat. An access control list (ACL) is assigned to each resource, and defines which consumers (SIDs) can use the resource. To ease management, SIDs can be assigned to groups, and the ACL can grant or deny access based on the group SID. Windows 2000 also supports a "SID history": an additional SID can be assigned to an account to support migration.

SIDs in Windows 2000 are managed using the Active Directory in the same manner as Kerberos keys. SIDs are passed between resources using Kerberos data structures. This

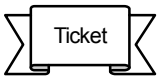
is functionality built in to the Kerberos protocol and necessary for a solid security design, but authorization is not governed by the Kerberos standard.

Symbols

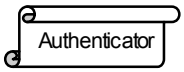
The following symbols are used throughout this document to describe the Kerberos authentication process:

An encryption key

Data encrypted using the encryption key above



A Kerberos ticket



A Kerberos authenticator

Acronyms

The following reference list of acronyms may be helpful in understanding this document.

ACL	Access Control List. Governs which identities have access to the resource being controlled.
AS	Authentication Service. The part of the Kerberos service that verifies a client by using a shared secret.
DNS	Domain Name System. A means for computers on the internet to resolve a unique human-readable name to an internet address.
KDC	Key Distribution Center. A process that provides both the authentication service and the ticket-granting service.
Krbtgt	The account and service principal name used by the key distribution center.
LSA*	Local Security Authority. The process running on every Windows NT and 2000 machine which handles authentication and authorization.
NTFS*	Windows NT file system. The file system defines access control lists, which require authentication to grant access.
NTLM*	Windows NT LAN Man authentication protocol. The standard

authentication scheme for NT 4.0

RFC — Request for Comments. An internet standard.

SID* — The Security Identifier. A unique name issued to each identity in Windows that may request access to resources. Access control lists specify SIDs that are allowed to use the resource.

SPN — Service Principal Name. A unique identifier assigned to each object that can be managed within the Kerberos realm.

SSP* — Security Support Provider. A library that can provide the basic authentication functions required by Windows 2000. Examples include NTLM and Kerberos.

SSPI* — Security Support Provider Interface. The set of common functions used by client-server application functions to access security objects.

TGS — Ticket Granting Service. A process that issues service tickets when a valid ticket-granting ticket is presented.

TGT — Ticket Granting Ticket. A Kerberos ticket presented to obtain tickets to services.

* These terms are specific to Microsoft platforms

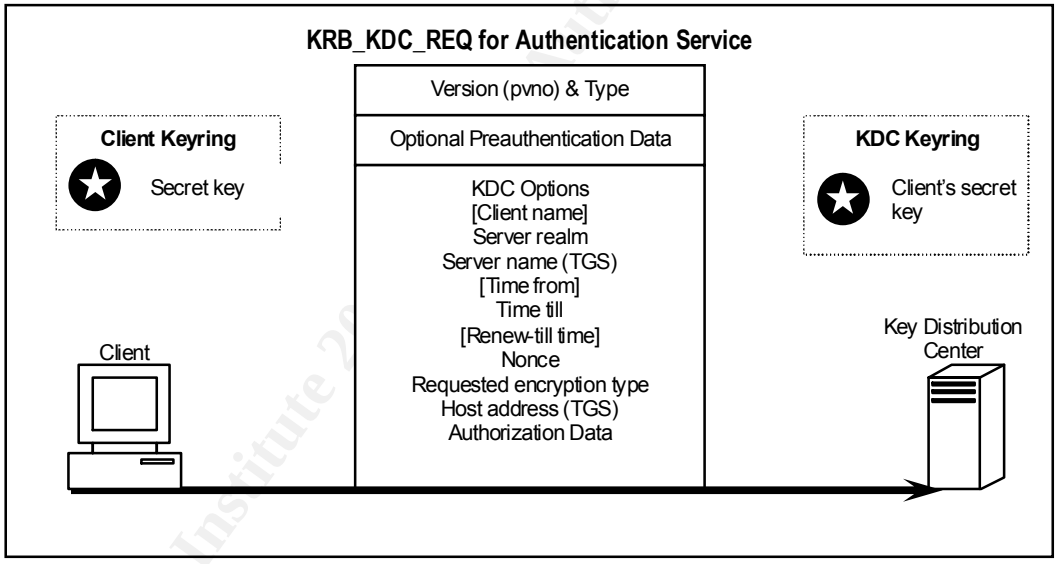
The Authentication Service Exchange

The Authentication Service exchange is performed when a client needs authentication credentials for a server. The exchange is encrypted using a pre-shared secret (password) known only to the client and the server. This is typically performed during login to obtain the Ticket-Granting Ticket (TGT)—a standard ticket used for the Ticket-Granting Service (a TGS Ticket). The TGT can then be used to obtain additional service tickets without requiring knowledge of the pre-shared secret.

KRB_AS_REQ

RFC 1510

The KRB_AS_REQ uses the KRB_KDC_REQ format. The message is sent from the client to the Authentication Server.



Preauthentication data, if included, holds a timestamp encrypted in the client's secret key. If the server requires preauthentication an error is sent unless the preauthentication data is valid.

The client name is optional for the Authentication Service request. *This field is not necessarily the client to which the ticket will be encrypted.* Rather, the actual client is required in the "Host Addresses" field. The server name and realm is required; the server realm must also be the client realm since a client can only authenticate to a KDC in its own realm.

The nonce field holds a random number. When the client receives the encrypted reply from the authentication server, it checks to make sure the nonce returned is the same as the one presented. Although all other fields for two subsequent requests could possibly

be unchanged, the nonce will always change. This allows the client to detect replay attacks from a spoofed authentication server.

KDC options can be any of the following:

■ Forwardable	■ Allow-Postdate	■ Encrypt Ticket in Session Key *
■ Forwarded	■ Postdated	■ Renew *
■ Proxiable	■ Renewable	■ Validate *
■ Proxy	■ Renewable-OK	* Available for he KRB_TGT_REQ only

The purpose of these options is described below in the "Kerberos Ticket" section.

Implementation of KRB_AS_REQ in Windows 2000

The first chore of the client is to locate the Kerberos server. For non-Windows 2000 Kerberos realms, the DNS name of the server is located in the client's registry. In Windows 2000 domains, the client uses the DNS entry for the active directory server (*_ldap._tcp.dc._msdos.DnsDomainName*) or the DNS entry for Kerberos (*_kerberos._udp.DnsDomainName*).

The RFC recommends a simple preauthorization by encrypting a timestamp with the client's secret key, and putting the value in the preauthorization field. By default, all accounts use preauthentication, and no reply is generated for invalid requests. Preauthentication can be disabled on the "Account" tab of a user object in the active directory, using the account option **Do not require Kerberos preauthentication**.

Windows 2000 implements both the Authentication Service and the Ticket Granting Service in a single "Key Distribution Center" (KDC) process. This is common with most Kerberos implementations. The KDC runs on every domain controller and uses the RFC standard *krbtgt* account. Neither the KDC service nor the account can be removed from a domain controller.

All the private key information needed by the KDC process is stored in active directory.

On interactive logon, the KRB_AS_REQ message is not actually sent until after the user types in the password and dismisses the login dialog with the "OK" button.

Applications can specify a particular Security Support Provider (Kerberos), or they can allow the operating system to choose the most secure security service.

Selecting a Security Support Provider (SSP)

Kerberos is the default SSP in Windows 2000. However, if the logon process fails to find a Kerberos server for authentication, it will use NT 4.0 NTLM authentication. NTLM authentication is necessary to logon locally to a machine, and to access the local SAM database.

Transport

RFC 1510 defines UDP port 88 as the transport for Kerberos authentication. Windows 2000 supports this transport. However, authentication for Windows 2000 clients requires a larger datagram, so Microsoft has implemented a complimentary TCP transport. This is keeping with a proposed revision to RFC 1510.

Key Management

All secret keys are stored as attributes in Active Directory. Since the Active Directory is multi-master, all secret keys are shared with all Active Directory domain controllers. Physical security of all the domain controllers becomes very important. It is common to physically and logically locate a domain controller close to clients, so with NT 4, they tend to exist as multi-purpose servers in remote offices. When migrating to Windows 2000, understand that the domain controller at the remote office now holds the same data and privileges as the controllers in the central data center. It must be bound to the same physical controls as domain controllers in the main corporate data center.

Domain controllers in a domain keep each other up-to-date using the proprietary Active Directory multi-master replication protocol. Since replication includes all the private keys in the domain, Kerberos is only as secure as the replication protocol.

Private keys are encrypted with an additional encryption key prior to being written to disk. The encryption key can be stored on the server or a separate physically secured floppy. This feature prevents private keys from being read off backup tapes. However, this key does not protect replication, since the private encryption key is not shared with the other domain controllers.

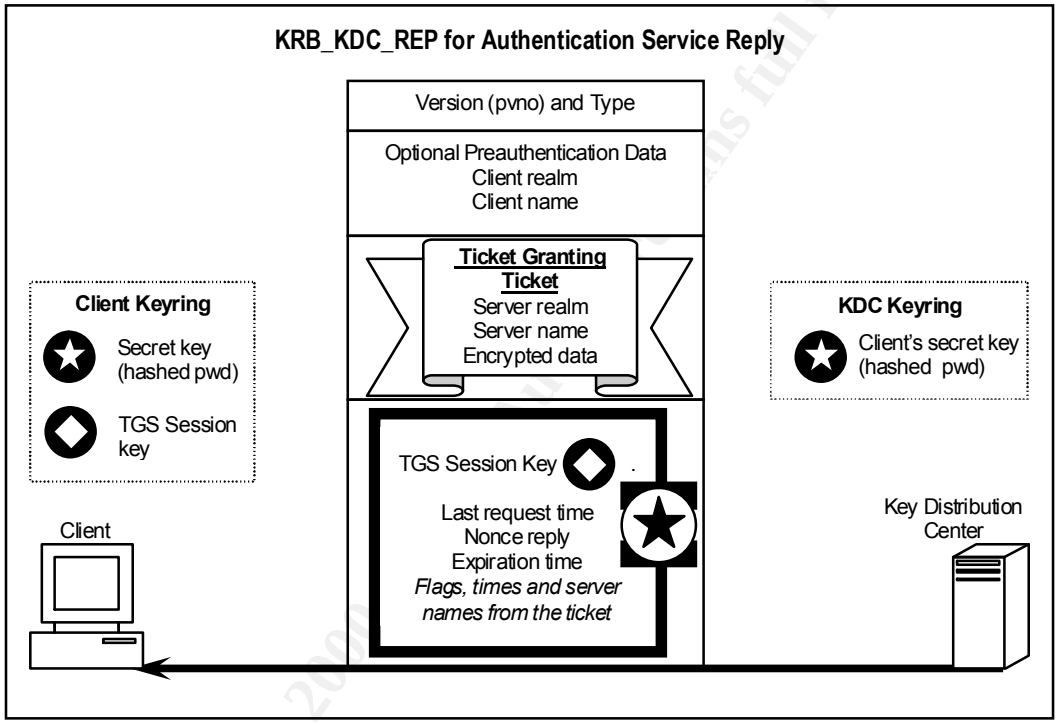
Passwords

The client's secret key is a 64-bit hash of the client's password. Password policy is set for all accounts on the domain controller through the "Domain Controller Security Policy". Standard options exist for password length, complexity, age and history.

KRB_AS_REP

RFC 1510

Similar to the KRB_AS_REQ, the KRB_AS_REP uses the KRB_KDC_REP format. The message is sent from the Authentication Server to the client.



The reply consists of two key components. The ticket contains mostly encrypted data that cannot be read by the client; the client can only retrieve the server name and realm that was requested. However, the client must present this ticket when requesting authorization by that server. For the AS request, this is the Ticket-Granting Ticket (TGT).

The second component is an encrypted block of data. The data is encrypted to the client's secret key, known only by the client and the KDC. This data holds the session key to be used by the client when connecting to the server. It also holds the nonce sent in the initial request which must be verified upon receipt. Additional administrative data (expiration times and flags) from the ticket are presented to the client, since the client cannot actually access data in the encrypted part of the ticket.

Knowledge of the ticket and the session key constitutes authorization between the client and the server. The KDC discards the session key once it has been sent to the client.

The client realm and client name are the same as initially sent in the KRB_AS_REQ message.

Implementation of KRB_AS_REP in Windows 2000

During interactive logon, the TGT is always immediately used to request a ticket for access to the computer.

In Windows 2000, *the Local Security Authority (LSA) keeps a copy of the logged-on user's password in memory* so that a new TGT can be obtained without interrupting the user. This means that an account can still remain logged on to the domain forever. This is the same password and storage mechanism used for NTLM authentication.

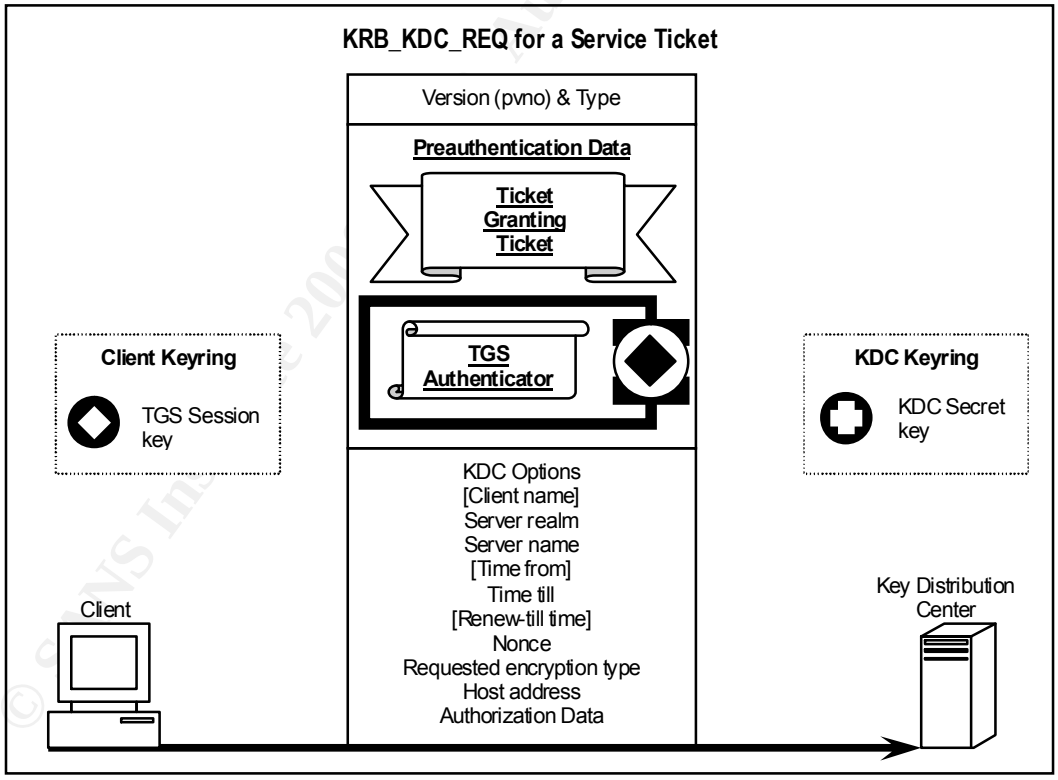
The Ticket Granting Server Exchange

After the client has received a Ticket-Granting Ticket, it will request access a server. Kerberos will allow the server and client to both trust each other's identity. The first part is for the client to request credentials from the Ticket Granting Service. This is done with the Ticket-Granting Server exchange. Data received by the client in the TGS exchange can be presented to the server to verify identities.

KRB_TGS_REQ

RFC 1510

The KRB_TGS_REQ takes the format of the KRB_KDC_REQ, similar to the KRB_AS_REQ message.



The Preauthentication Data field is used to transmit the Ticket Granting Ticket and an authenticator for the Ticket Granting Service. If the client is renewing or validating an existing ticket, then the ticket for renewal is also transmitted in the preauthentication data.

Other options are the same as for an authentication request.

Implementation of KRB_TGS_REQ in Windows 2000

The client has to find the service name in order to request a ticket to use the service. This is done using the Service Principal Name (SPN). The SPN is assigned by a service when it is initially installed. SPNs must include the DNS name of the service's host server. Windows 2000 then registers the SPN to a particular account, and Kerberos policy is then managed for that account. The SPN uses the format *ServiceClass/Host:Port/ServiceName*.

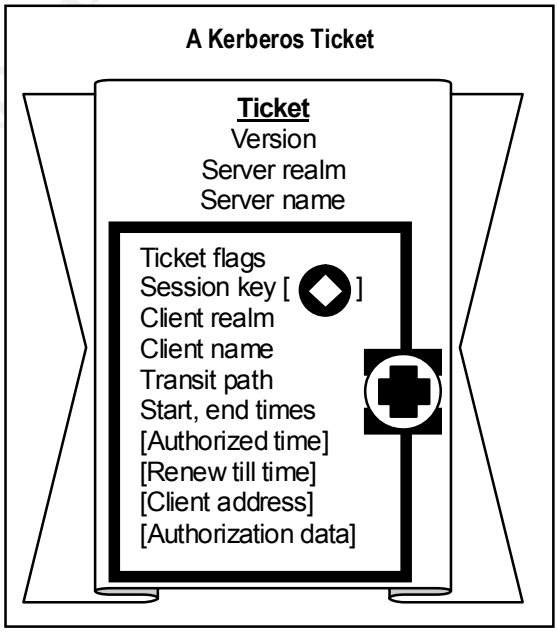
The Kerberos Ticket

RFC 1510

A ticket is used to exchange the session key. Once the client receives a ticket for a service, it can initiate secure and authenticated communications with the service until the ticket expires.

The server realm, client realm and transited fields all contain Kerberos "realms". The realm is the family of accounts managed by a single web of trust, and is analogous to a Windows 2000 domain.

Data in the authorization data field is used to pass specific authorization data for the targeted service. The authorization data field allows a principal to forward or proxy a ticket for a specific service only.



RFC 1510 supports the following ticket flags:

- Forwardable
- Forwarded
- Proxiable
- Proxy
- May-Postdate
- Postdated
- Invalid
- Renewable
- Initial
- Pre-Authent
- HW-Authent

Note that these flags are set inside the encrypted portion of the ticket, and so cannot be modified by the client or the server. These options are negotiated when the client requests the ticket from the KDC without any interaction from the server.

Postdating Tickets

A ticket may be issued that does not become valid until a time in the future. A postdated ticket will have its "invalid" flag set and a "Time from" in the future. Once the ticket reaches its start time, it can be submitted to the KDC to have the "invalid" flag reset. This is useful for issuing forwarded or proxied tickets now for an action to occur in the future.

Renewing Tickets

The client requests a specific ticket lifetime by using the "till" time. If policy prohibits issuing a single ticket of the desired lifetime, a ticket with a shorter lifetime and a "renewable" flag set.

When a renewable ticket expires, it must be resubmitted to the KDC to have the times updated. When the times are updated, the KDC also checks to make sure the ticket is still valid. For example, a ticket may be renewable for 7 days; however, if it only has an 8-hour lifetime, then the maximum window for a compromised ticket would be 8 hours.

Forwarding and Proxying Tickets

One great benefit of Kerberos is that it supports authentication forwarding and proxying. For instance, suppose a client is logged in to a web application. The web service can then connect to a database service under the client's security context, and the database can manage access based on the user, not the service. The database can also then log changes and updates based on the actual network account, and not the web application account.

If the AS request sets the "Proxiable" flag, the TGT issued can be used to issue proxy service tickets. Once a client has a proxiable TGT, it can request a service ticket that allows a specific service (e.g., web server) to use another service (e.g., database service) on its behalf. This service ticket request would have the "Proxy" flag set.

In a similar manner, the AS request can set the "Forwardable" flag to issue a TGT with the authority to forward credentials. Forwarding is different from proxying, however, because it releases complete use of the client's identity to the server. The client actually requests a TGT to be used by the server. The server can then use this TGT to request services using the client's authentication credentials but without actually knowing the client's secret key. All service tickets issued based on a forwarded TGT will have the "Forwarded" flag set.

The Authenticator

RFC 1510

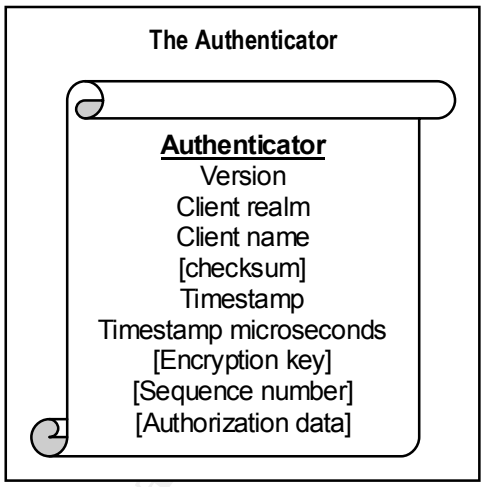
The authenticator is sent with the ticket to certify that the client received the session key from the KDC, and helps the server detect replays. The entire authenticator is encrypted in the session key shared between the client and the server.

The server keeps authenticators for a time to prevent replay. If a second authenticator is presented that is the same as an earlier one, it is rejected.

Kerberos standard time fields limit granularity to the second; an additional microsecond field is added in the authenticator to guarantee that it will be unique.

The checksum field can be used to verify the *application* data in the *ticket*.

The encryption key allows the *client* to specify an encryption key to protect this specific application session.



The authorization data is used just like the field of the same name in the ticket. However, since the client builds the authenticator, it is authorization data provided by the client. The authorization data field in the ticket is generated by the KDC.

Implementation of Tickets and Authenticators in Windows 2000

All tickets created by the Windows 2000 KDC are "forwardable". Documents conflict on how forwarding is actually performed, and more research is necessary to fully understand forwarding in the Windows environment. In particular, page 35 of reference 10 in talking about accessing resources on another computer states "..the client application on the local computer sends you logon credentials to the server application on the remote computer…" Other documentation leads me to believe this is incorrect and should say something like "…the client application on the local computer forwards your [ticket-granting] ticket to the server application on the remote computer…"

Microsoft adds a Privilege Attribute Certificate (PAC) data structure in the authorization data field for every native TGT issued. Interestingly, reference 7 defines this structure as requiring NetBIOS names (I was under the impression that NetBIOS was deprecated with Windows 2000).

When Ticket-Granting Tickets are created, the KDC fills the authorization data with all the SIDs that can be used. This would normally contain one SID for the account requesting the ticket, and one SID for each group within the realm to which the requestor belongs.

For service tickets, the SIDs from the TGT are copied to the service ticket's authorization data. This allows the service to create an impersonation of the requestor's security context.

In short, all the SIDs for any given account are included in every ticket. The SID data structure is small compared to the overall transmission size; however, if network bandwidth is a concern, group structure could have an impact. For instance, the use of overlapping "Everyone" groups should be discouraged.

Authorization data in the ticket is signed by the KDC to prevent manipulation. Since the service knows its secret key, it has access to any tickets presented by clients requesting

services, and could manipulate the authentication data in the ticket. However, any manipulations would void the signature.

Some additional options are defined by the default domain group policy, and managed by domain administrators.

Forwarding and Proxying

All tickets issued by the Windows 2000 KDC are marked as forwardable. However, default options prevent tickets from being forwarded. Options for controlling forwarding of user accounts are under the properties of an Active Directory user object, in the "Account" tab.

When **Account is trusted for delegation** is selected a service running under this account may forward tickets. By contrast, the option **Account is sensitive and cannot be delegated** is used to prevent an account from being forwarded. It is important to understand the difference for these two options: accounts with the first option set are allowed to use another's credentials as their own; accounts with the second option set can not be forwarded by an account with the first option set. For the situation where a user logs into a web service that connects to the database, the following options would be set:

- The user account *must not* have the "Account is sensitive and cannot be delegated" option selected
- The Web service account *must* have the "Account is trusted for delegation" option set
- Neither option need be set for the database service account.

The **Use DES encryption for this account** changes default encryption for the account from the standard 128-bit RC4 algorithm to 56-bit DES. This should only be used when necessary for interoperability with other systems.

The **Do not require Kerberos pre-authentication** option increases the difficulty for an attacker to obtain valid tickets. A ticket can only be requested if the attacker knows the account password. If this option is selected, password-guessing attacks become somewhat easier.

The **Trust computer for delegation** option on the general tab of the computer properties dialog can be set to allow a computer to forward credentials. All services running as the local administrator will use this computer account for authentication, so this option controls ticket forwarding for these services.

No Windows 2000 documentation was found to support ticket proxying. Although proxy flags may be exposed through the programming interface for Kerberos system interoperability, no tools exist to manage proxy functionality.

Enforce User Logon Restrictions

When this option is enabled, the KDC checks ticket requests against user rights for the target computer. This is done every time a TGT is submitted to the KDC. If the user has either 'Log on Locally' or 'Access this computer from the network' rights, the ticket request is allowed.

This option seems to be for backwards compatibility only. It has no parallel in the Kerberos RFC; in fact, it causes some confusion since it directly ties authentication with authorization. It seems this option would be better performed as part of the authentication process of the target service. On the other hand, the Kerberos specification does allow the KDC to perform authorization tasks, so it is not a violation of the RFC.

Maximum Lifetime for a Service Ticket

This option sets the End-time field in a Kerberos ticket. The default setting of 10 hours corresponds to the RFC recommendation.

If the maximum lifetime is set to zero, ***the ticket never expires***. This apparently violates the general intent of the RFC regarding tickets. Some provisions are made for services to override the general realm expiration policy, but never for the purpose of extending it indefinitely. More specifically RFC 1510, section 3.13 states:

```
The expiration time of the ticket will be set to the minimum of the
following:

+The expiration time (endtime) requested in the KRB_AS_REQ
 message.

+The ticket's start time plus the maximum allowable lifetime
 associated with the client principal (the authentication
 server's database includes a maximum ticket lifetime field
 in each principal's record; see section 4).

+The ticket's start time plus the maximum allowable lifetime
 associated with the server principal.

+The ticket's start time plus the maximum lifetime set by
 the policy of the local realm.
```

Further investigation is necessary to see how this setting is implemented. Perhaps it simply extends the ticket lifetime to the TGT maximum allowed time, or the documentation may be in error. Or it is possible that this option was included for services that cannot properly manage renewal of an expired, renewable ticket.

In any event, system administrators are strongly cautioned against setting an expiration time of zero.

Maximum Lifetime of a User Ticket

This option also corresponds to the End-time field of a Kerberos ticket. The default value of 10 hours follows RFC recommendations. Documentation did not mention the validity of using a lifetime of zero.

Maximum Lifetime for User Ticket Renewal

This option corresponds to the optional Renew-till field in the ticket. Again, the default value of seven days follows RFC recommendations.

Maximum Tolerance for Computer Clock Synchronization

The default value of five minutes follows the RFC general recommendation for clock synchronization. This does not correspond to a specific setting from the RFC, and the actual implementation of this setting is unclear. All Microsoft documentation indicates that this limits the tolerance between the KDC and the client, and nothing refers to the tolerance between the client and the server.

If this option specifies a tolerance between the KDC and the client, it can only be implemented for the AS and TGS exchanges. The AS exchange optionally uses a timestamp in the header for preauthentication; if preauthentication is not required, this option would not apply to the AS exchange. The TGS exchange includes an authenticator that contains a timestamp; this is obviously governed by this setting.

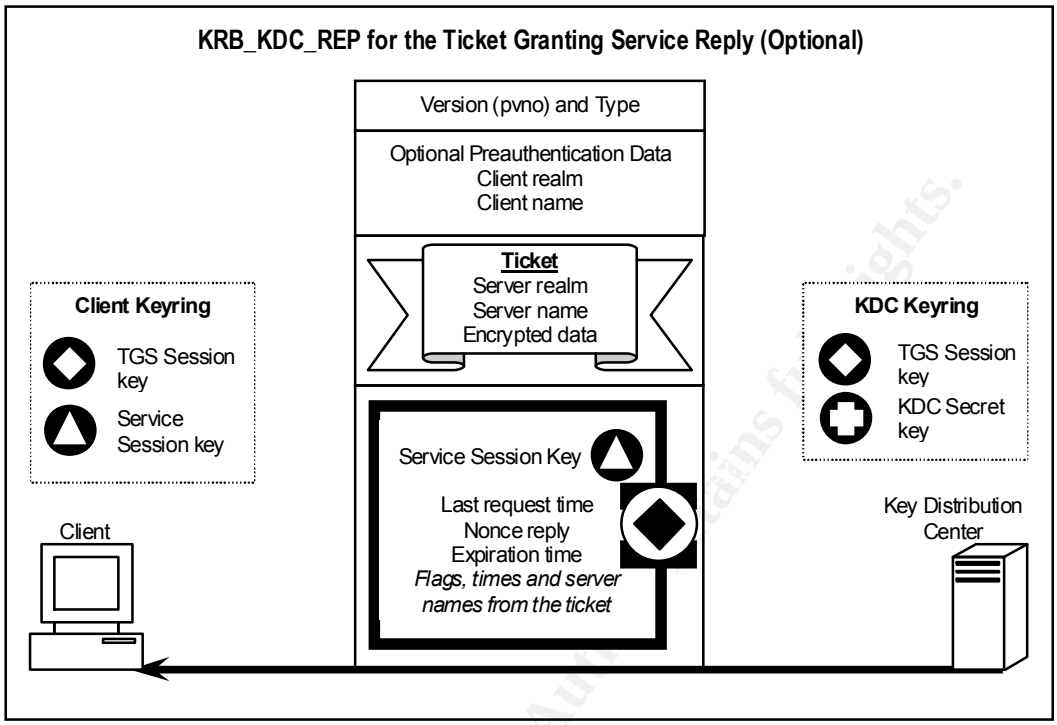
However, according to the RFC, an AP exchange also includes an authenticator with a timestamp. This authenticator must be verified by the server, and returned to the client if mutual authentication is required. If all authenticators are governed by this setting, then all computers within the realm must be synchronized with each other, not just with the KDC.

In any case, the system administrator will rarely see access denied due to clock synchronization. When authentication fails due to synchronization, an error is returned to the client. The Windows 2000 Kerberos client reads the timestamp in the error message and creates another request and authenticator using this new timestamp. This should successfully complete the exchange without user interaction.

KRB_TGS_REP

RFC 1510

In the TGS reply message, the TGS replies with a session key to be used in authentication with the target service.



Fields in this message are used for the same purpose as those in the Authentication Service reply

Implementation of KRB_TGS_REP in Windows 2000

Microsoft's implementation of the KRB_TGS_REP message follows the RFC standards. All the options used in the reply are as discussed elsewhere in this document.

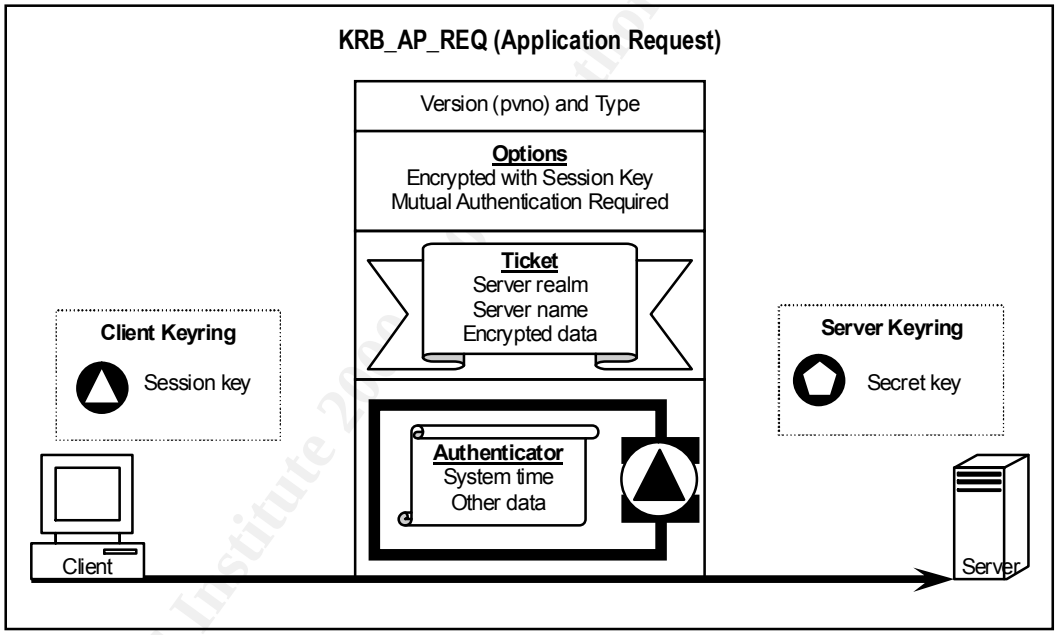
The Client / Server Authentication Exchange

The client-server authentication exchange allows both the client and the server to authenticate each other. This authentication piece is commonly known as "Authentication Header". This flows over the normal service communication channel. In order for this exchange to occur, the client must already have received a ticket for the target service from the Ticket-Granting Service.

KRB_AP_REQ

RFC 1510

The first mandatory part of the exchange consists of an authenticator, a ticket, and some additional header information. This exchange assumes the client has already received a session key and ticket using the KRB_AS exchange.



The server's secret key is an encryption key shared by the server and the Key Distribution Center.

If the Mutual-Authentication-Required flag is set, the server is expected to reply with a KRB_AP_REP message. If the flag is not set, this data packet may also contain the client's initial request.

Authentication of machine accounts presents a topic for further research. The machine must authenticate to the domain automatically to have applicable group policy applied before an interactive user logon. Where is the private key for the machine account stored, and how is it updated? What security risks are presented by this account? If one

significant advantage of Kerberos is mutual authentication, then the server's identity is only as secure as the private key upon which the authentication is based. No documentation was found to discuss machine account authentication.

Programming Kerberos Security into Client Server Applications

In order for a server application to extract values from the ticket in the KRB_AP_REQ message, it must know information about its own secret key. A server receives the necessary information to decrypt the ticket by calling the AcquireCredentialsHandle SSPI function. This is normally done when the service starts.

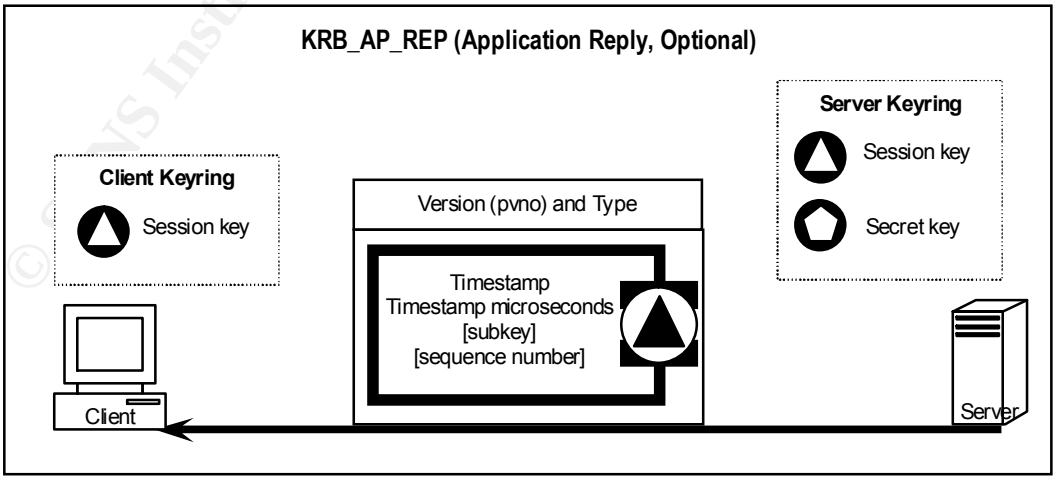
A client calls the InitializeSecurityContext SSPI method to generate the KRB_AP_REQ message. This message is piped to the server through an existing socket.

When the server receives a client request, passes it to the AcceptSecurityContext SSPI function along with its secret key. The Kerberos SSP verifies the information and creates a security token with the client's access privileges. It will also generate the KRB_AP_REP message when required. The server then spawns a thread with the client's security token and this thread (running under the client's privileges) does all additional processing.

KRB_AP_REP

RFC 1510

If the KRB_AP_REQ message has the MUTUAL-REQUIRED flag set, the server must reply with a KRB_AP_REP message to complete the authentication. In its reply, the server proves that it has the secret key necessary to decrypt the ticket presented by the client.



The subkey can be used similar to that of the authenticator above, but this is a subkey generated by the server rather than the client.

Kerberos version 4 required a reply of the timestamp + 1; this requirement was changed in Kerberos version 5 to require a reply with the same timestamp.

Implementation of KRB_AP_REP in Windows 2000

Windows 2000 implements the reply message in accordance with RFC standards. By default, most standard services (including the Server service for NTFS access) perform mutual authentication and send this reply message.

Conclusions

In almost all aspects, Microsoft fully implements the RFC. It is obvious that this is the first implementation of Kerberos integrated into the Windows platform, and future releases should strengthen manageability and tightly control fail over.

One confusing trend in the documentation and the user interface is the continued segregation of user and computer accounts. In the Kerberos standard, a service or computer account is treated identically to a user account. However, Windows presents a completely different interface for Kerberos settings in user and computer accounts. In fact, the computer account interface does not even provide a setting for "Maximum lifetime for service ticket renewal", which should directly correspond to the "Maximum lifetime for user ticket renewal". Since mutual authentication is an integral part of the complete Kerberos implementation, it becomes necessary to manage computer and service accounts with the same integrity as user accounts.

Significant Findings

The Local Security Authority still keeps a cached copy of the client's hashed password. In addition, the LSA may periodically request a new ticket-granting ticket for a logged on user with this password. This exchange will happen without user interaction. This password is also presented whenever necessary for NTLM authentication.

The "Account is trusted for delegation" and "Account is sensitive" properties of user accounts have significantly different meanings but could easily be confused. System administrators should be trained on the specific purpose of these settings.

Documentation indicates that the maximum lifetime of a service ticket can be set to zero, which prevents the ticket from expiring. This issue requires further research; if the ticket truly does not expire, not only would the syntax violate the RFC, but also any clients accepting a ticket without an expiration date would be in violation.

Physical security of every Active Directory domain controller is essential, and should be reviewed when migrating or installing Windows 2000 domain controllers.

Additional Findings

When clocks within a domain are out of synchronization, Kerberos will generate error events, but the user will not notice these events.

Kerberos requires a DNS entry for the active directory or a client-side registry setting to allow a client to locate the Kerberos server. Active Directory also typically requires the same DNS entry.

A service can specify a particular security provider through the SSPI if desired (Kerberos vs. NTLM).

Windows 2000 typically passes Kerberos data on UDP port 88 when authorization data (the PAC) is not included, and uses TCP port 88 for larger exchanges containing authorization data.

All tickets generated by the Windows 2000 Kerberos service are forwardable.

The Windows 2000 interface does not support Kerberos ticket proxying.

Kerberos identifies objects with a Service Principal Name (SPN). The SPN must include the DNS name for the object's host, and be registered to an Active Directory account. Kerberos policy for the SPN is managed by its registered Active Directory account.

Additional research

Some additional topic areas for research include:

- A confirmation of Kerberos authentication is used using network traces, and evidence on how Kerberos fails over to NTLM
- A detailed examination of the tools used to manage Kerberos in Windows 2000 (KerbTray, Netdom, KSetup), and any third party tools available
- Procedures to dynamically expire tickets that are currently in use
- The assignment of principal names, how they relate to DNS and NetBIOS names; the impact of renaming machines and services. Can two separate realms be merged?
- The interaction of Windows 2000 Kerberos with other operating systems. For instance, can a Linux workstation on a Windows network connect to the TGS?
- A detailed case study of how NTFS uses Kerberos
- The KRB_SAFE, KRB_PRIV and KRB_CRED Kerberos exchanges.
- Development of a utility client & server program set which accesses configuration options of the Kerberos SSPI.
- Incident handling procedures for resetting keys following a suspected compromise.

References

1. "Auditing the Windows 2000 Authentication Process." Julio Silveira. 1 April 2001. URL: http://www.sans.org/infosecFAQ/win2000/audit_w2k.htm (13 June 2001).

2. "Authentication for Administrative Authority." Microsoft Corporation. 28 July 2000. URL: http://www.microsoft.com/technet/security/authent.asp (13 June 2001).

3. Bill Bryant. Designing an Authentication System: a Dialogue in Four Scenes. Cambridge: Massachusetts Institute of Technology Project Athena, 1988.

4. Dave Opitz. Guide to Windows 2000 Kerberos Settings (Version 1.0 Draft). Fort Meade: National Security Agency, 2001.

5. J. Kohl, C. Neuman. Request for Comments 1510: The Kerberos Network Authentication Service (V5). Internet Engineering Task Force (IETF) Network Working Group, 1993.

6. Jennifer G. Steiner, et al. Kerberos: An Authentication Service for Open Network Systems. Cambridge: Massachusetts Institute of Technology Project Athena, 1988.

7. Microsoft Corporation. Microsoft Authorization Data Specification v. 1.0 for Microsoft Windows 2000 Operating Systems. Redmond: Microsoft Corporation, 2000.

8. "Mutual Authentication Using Kerberos." Microsoft Corporation. 5 December 2000. URL: http://msdn.microsoft.com/library/psdk/adsi/glmauth_2yb7.htm (30 May 2001).

9. "Overview of the Kerberos Protocol." Microsoft Corporation. 5 December 2000. URL: http://msdn.microsoft.com/library/psdk/secspi/aboutsspi_lnl8.htm (30 May 2001).

10. White Paper, Microsoft Corporation. Windows 2000 Kerberos Authentication. Redmond: Microsoft Corporation, 1999.