Global Information Assurance Certification Paper

Kerberos Authentication in Windows 2000

- Vishwas Gadgil.
- Assignment Version – 1.2e

Microsoft's newest operating system Windows 2000 uses a publicly available protocol definition for authentication. The protocol used is Kerberos. This paper attempts to explain the Kerberos based authentication in Windows 2000 and also tries to explain the new and sometimes confusing terminology in a layman's words.

It first discusses the limitations of traditional Windows NT style authentication. Then it shows how Kerberos implementation in Windows 2000 attempts to overcome most of those limitations. This paper also presents a much-detailed analysis of the exact steps that occur when a Kerberos based authentication occurs in Windows 2000.

Limitations of Traditional NT Authentication

PDC and BDC form a single point of failure - the PDC and BDC(s) host The Security Accounts Manager database for the entire domain. The copy of SAM database on a BDC is a read only copy; this means any changes to the security database can only be done using the PDC. This poses a big threat in terms of security. The availability and security of the PDC becomes a critical issue. (Although, Microsoft claims that a BDC could be easily promoted to be a PDC when a real PDC becomes unavailable, it still takes a lot of time (and pains, frustrations) which may not be acceptable in many mission-critical implementations)

Different security databases in a single domain - The Security Accounts Manager database is different on a domain controller and on a regular server. A regular server can NOT be upgraded or promoted to be a domain controller unless the operating system is installed from scratch. The reverse is also true; a domain controller cannot be demoted to be a regular server unless reinstalled completely. The reason of this limitation is the difference in the SAM database.

Size of the SAM database - The total number of computer accounts, user accounts and group accounts in an NT domain is limited by the size of the Registry. The NT registry constitutes the SAM database that authenticates all of these entities. According to Microsoft's TechNet, a typical SAM has enough room for approximately 40,000 Accounts. (1 KB / User Account; 0.5 KB / Computer Account; 4KB / Group Account)

But in practical terms, with the present hardware and infrastructure limitations, if the number of users grows more than approximately 10,000, the performance in terms of logon times degrades and replication delays of SAM database between domain controllers could create inconsistencies, which could create a major security threat.

Trust relationships between domains - In Windows NT, the trust relationship between any pair of domains is strictly non-transitive. This means, if domain A trusts domain B, domain B trusts domain C, then to have a trust between domain A and C, you still need to establish a trust relationship between A and C explicitly. Although, this is not a security concerns so to speak, while managing a large enterprise with more than 6/7 domains could pose a security threat in terms of a human error; an inexperienced or hasty Administrator could inadvertently create or delete a trust relationship between domains. At my facility, where we manage 16 domains, one of the charts that every Administrator has is a graphical representation of the current trust relationships between domains. Managing these large numbers of domains could cause significant administrative overheads and frustration!

Multiple Logons with same user name and password - If a user has both a domain account and a local account on a server with exactly same user name and a password, NT could be cheated to believe that a user logged onto a particular server is either a local or a domain account. This is especially true for Admin level accounts.

Locally cached profiles - By default, NT caches a user profile whenever S/he logs on to a NT machine. This allows access to local machine even after a user account has been deleted from domain. You could simulate this by creating a dummy account in NT, logging onto a machine, and then deleting the account from domain. Next time you try to login to that machine, all you have to do is to disconnect the network cable before logging on. NT will allow you to logon with a message that says " No Domain controller found to authenticate you, still logging you on with the cached profile". This could create a significant security threat.

Performance Issues - Although not directly related to security, having a limited number of domain controllers (Typically, One PDC and a couple of other BDCs) poses poor performance, especially over slow WAN links. If the domain size is significant (~ 5000 users / computers..) launching a GUI such as a Server Manager to manage a server or a domain controller across a slow WAN link is really frustrating.

SAM Database Replication issues - The SAM database gets replicated between the domain controllers in Hub-and-spoke or STAR topology. What this means is every BDC connects to a PDC to get its copy of the SAM database. By default, NT replicates the SAM database whenever there are 200 updates accumulated or every 7 minutes or at a random interval between 1 and 7 minutes. If you have a large domain and a fairly large number of BDCs, the replication of a flat and large SAM database may not get over in 7 minutes and this cold lead to some BDCs not getting their turn. This could lead to inconsistencies in the SAM database available to a portion of the domain.

Windows 2000 Kerberos Approach

In the light of above discussion, it was necessary on Microsoft's part that they improve on the security aspects of Windows 2000. Instead of patching up the existing schema for the security, Microsoft chose to select a completely new protocol that was publicly available for years and has been proven to be much faster and solid. The History and basics of the Kerberos can be found in these links. [5]

In short, Kerberos uses 3 party mechanism for authentication. The entity that needs an authentication, The entity that has the resources, and an entity that holds credentials for both. One assumption in the transactions is that the communication between these entities takes place over a network that is full of adversaries.

I will present a very simple analogy to more understand the 3 party mechanism.
In the United States, to get the right to drive a car requires that you get through the driving license exam and then a road test. Once you clear both the exams, the Department of Transportation issues you a Drivers License that has information about you such as your photo, date of birth etc. You keep your driver's license always with you since in the USA, this is a quick means of identifying yourself. (You 'cache' your driver's license)

Now let us say that you are planning to fly to some other city to spend your summer vacation. You first go the Airport and buy a ticket. The person at the counter asks for your driver's license to make sure you are the person that you say you are. She then issues you a ticket that allows you to get to the appropriate airplane.

At the airplane's door, the Flight attendant again checks the ticket that you possess and then guides you to your seat, based on your privileges such as

Business Class or coach class etc.
The link here is that the Flight Attendant knows that the ticket you posses is actually issued by someone who S/he knows (the Airline counter).
If s/he finds that the ticket information does not match with the one that s/he knows about, s/he will not allow you access to the airplane.

This is a very simple analogy just to understand the 3 party nature of Kerberos authentication. The actual protocol transactions are much more complex (involve encryption) and the 2 different entities issuing tickets (DOT and Airline counter) in the example are same in Windows 2000 (The Domain Controller).

Protocol Vocabulary
The new protocol comes with new vocabulary. The definitions could be found in SANS reading room article. [1] But for the sake of this paper a very few definitions are in line.

Security Principal - Any entity that need an authentication. E.g. Users, Computers etc.

Realm - The database that holds the credentials for all the security principals. In windows 2000, a Realm and Domain are synonymous.

Ticket - Basic unit for all transactions in Kerberos. A Ticket contains encrypted information used by the 3 party mechanism of authentication. Details of the contents of the ticket could be found at this link. [2]

Key Distribution Center - In Kerberos terminology, a Key is another name for the Ticket. So the Key Distribution Center or KDC is the entity that issues the tickets. KDC has 2 functional parts.
- Authentication Service.
- Ticket Granting Service.
  Note: Please do not get confused wit the word SERVICE. These 2 services are not same as classic NT Services. They are just a part of the KDC.

KDC\Authentication_Service - Before a Security Principal gets a Ticket to validate itself; its credentials must be verified so as to make sure that the Ticket requesting entity is a valid Security Principal in the Realm. (In layman's terms - There should be a way to make sure that the ticket is issued to a valid

and already known entity. This is similar to the process where the department of transportation checks your Birth Certificate and Social Security Number card before they issue you your driver's license)

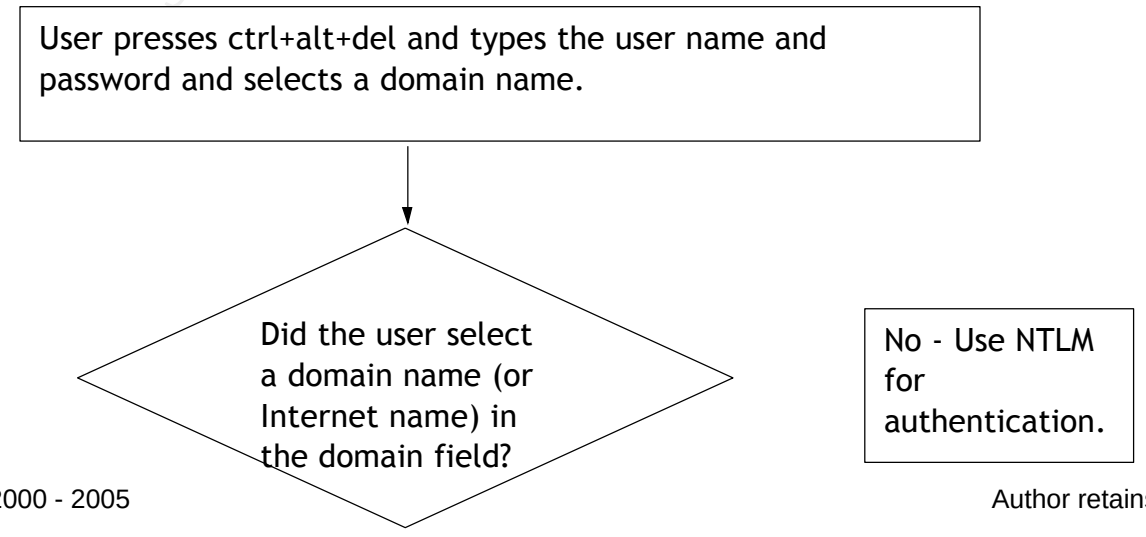
KDC\Ticket_Granting_Service - Once the KDC determines that the ticket requesting entity is a valid (already known) entity in terms of the Realm, the KDC issues the ticket that is used by the security principal in future transactions with KDC (It's similar to getting the actual driver's license). The KDC/Ticket Granting Service also functions to issue the 'Service Tickets' to mediate a transaction between a Principal and a Validation Server. (Similar to the Airline Counter who issues you a ticket for flying after they check your driver's license)

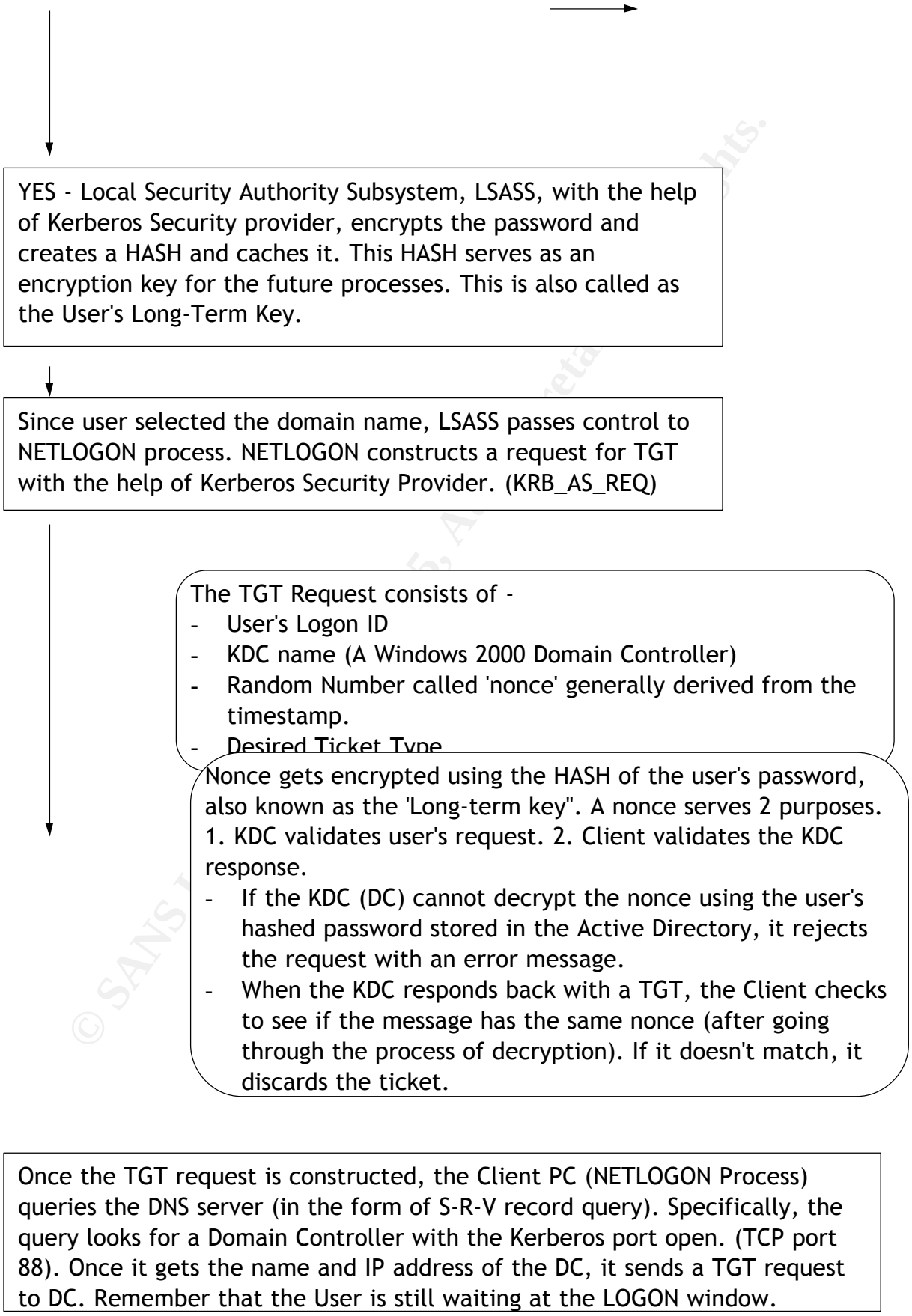
Validating Server - Validating Server is the server who has what the security principal needs. (Such as files, databases etc.) In Windows 2000, a Validating server is a member server. This server must be a part of the same realm as that of the KDC. (Same Domain) (This is similar to the fact that the Flight Attendant at the doors of the airplane, The Airline Counter and The Flight itself belong to the same Airline Company)

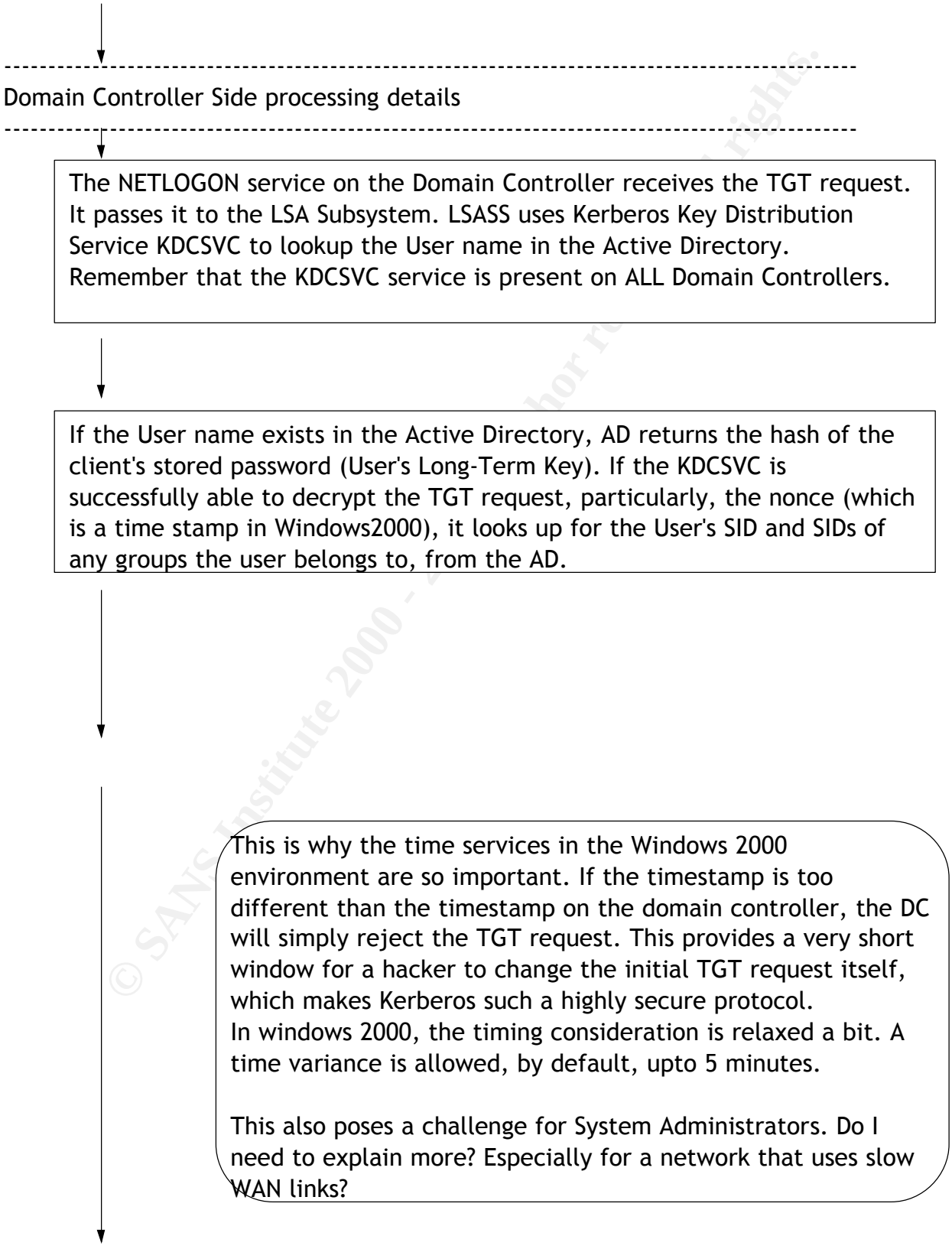
Windows 2000 Kerberos Transactions
Windows 2000 does not use Kerberos if the user is trying to logon to a standalone Windows 2000 computer. It uses NTLM. Windows 2000 also uses NTLM to authenticate classic NT and Win 9.x machines that do not have Active Directory Snap-in.

But when a Windows 2000 computer is a part of Windows 2000 domain, it uses Kerberos to authenticate the user to the domain. It also uses Kerberos when the user wants to access other resources in a Windows 2000 domain. Here is what happens.

User Logon Details (Initial Logon)







The KDCSVC\Authentication_Service on the DC now has all the information that is needed to build a TGT for the client. A typical TGT contains

- Timestamp and TTL - Time to Live or the interval for which this TGT is valid.
- Session Key (A random number) to uniquely identify this ticket.
- Authorization Data - User's Credentials and a Nonce.

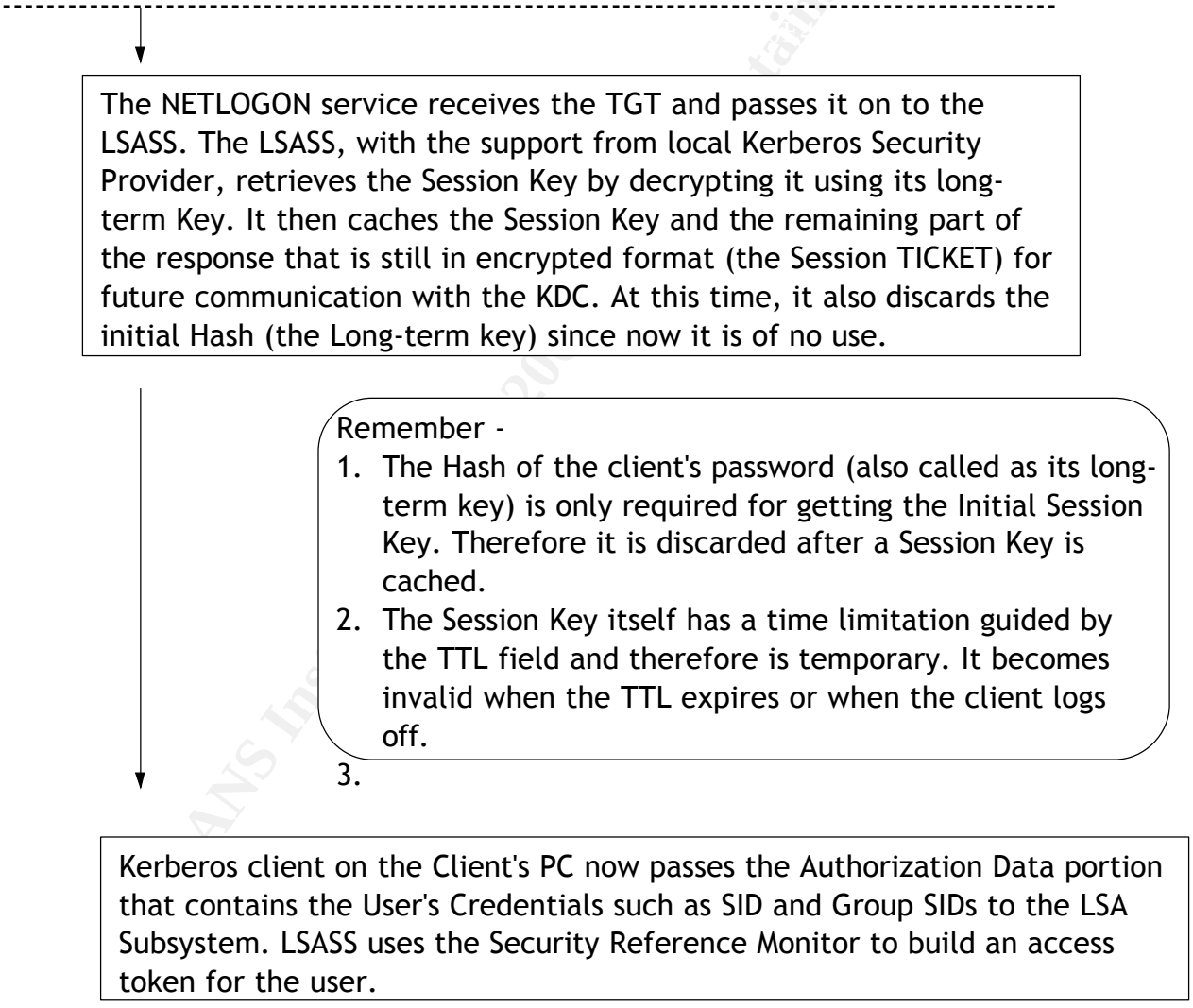
The rituals that the KDCSVC\Authentication_Service performs next are very important to understand (Also read, very confusing to understand).

- A copy of session key along with Timestamp, Authorization Data and Time-to-live information is encrypted using Long-Term Key (the hash of the password) of the account that controls the KDCSVC service on the Domain Controller. This is also called as 'Logon Session Ticket'. This account, by default, is called as 'krbtgt' and is similar to the 'system' account in classic Windows NT 4.
- Another copy of Session Key along with the Client Supplied Nonce and a CRC is encrypted by the client's password hash (remember that this hash was retrieved from the AD)
- Both of these, The Logon Session Ticket and the Logon Session key are returned to Client using the LSASS and NETLOGON services running on the DC. (KRB_AS_REP)

The 'krbtgt' account is automatically generated on a Domain Controller. This account cannot be deleted and the name cannot be changed. We can change the password but Microsoft does not recommend it.
Reason? Simple. Now YOU know the password and that might result in a security breach.
An aggravated IT Auditor might make you loose your job for this!!! Be careful..

Authorization Data Field - This field is used in 2 ways. During the Initial Logon, it includes a random number called Nonce, encrypted by KDC's long-term key and User Credentials. Nonce acts as a backup validation when the client uses its session key for further transactions, such as requesting a Service Ticket.
In a Service Ticket, it stores the Client's SID and SIDs of the groups it belongs to and encrypts it using the Validation Server's Hash (Long term key). This way the Validation Server could validate the user and allow appropriate access to the client.

Client PC Side processing details



Hushhhhhhh!!! At this time the User Logon is complete and the User is returned to the Windows 2000 desktop console (Explorer Shell). The processes initiated by the user from this point on are run in the context of the Access Token.

Now that the user is logged on s/he could start using network resources. This is where the actual 3 party mechanism comes into picture. Here are the detailed steps that occur when a Windows 2000 user already logged on a PC tries to access a networked resource.

In short, the Client must get a ticket from the KDC to present to the service provider. This ticket will have client's credentials and Service Provider's encrypted session key.

Network Resource Access Authentication

The Kerberos client on User's PC requests credentials for the service by sending the KDC (DC) a Kerberos Ticket-Granting Service Request (KRB_TGS_REQ). Note that this is a TGS request as opposed to TGT request.

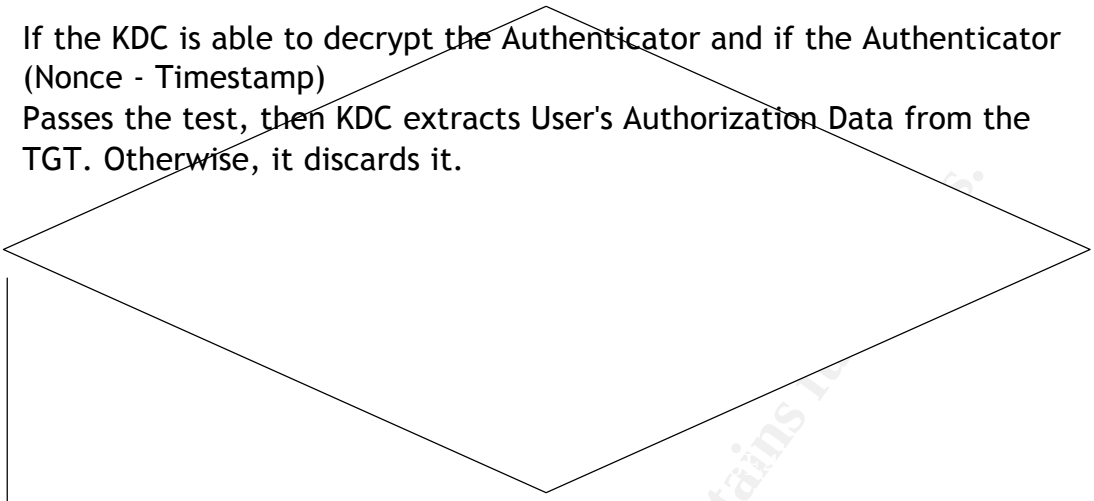
The TGS Request consists of -
- Name of the target service provider -Validation Server.
- User's Logon ID
- The TGT received from KDC during initial logon
- Nonce - A random number derived from the timestamp.

The TGT that is included in this request is still in the encrypted format and only the KDC can decrypt it. Remember that the KDC encrypted it using the long-term key of the special account 'krbtgt'.

Nonce - Also called as Authenticator, is encrypted with the user's logon session key.

When the KDC receives KRB_TGS_REQ, it decrypts the TGT with its own secret key, extracting User's logon session key. It uses the logon session key to decrypt the authenticator and evaluates that.

If the KDC is able to decrypt the Authenticator and if the Authenticator (Nonce - Timestamp)
Passes the test, then KDC extracts User's Authorization Data from the TGT. Otherwise, it discards it.



Again, this is why the time services on the Windows 2000 environment are so important. If the timestamp is too different than the timestamp on the domain controller, the DC will simply reject the TGS request. This provides a very short window for a hacker to change the TGS request itself, which makes Kerberos such a highly secure protocol.
In windows 2000, the timing consideration is relaxed a bit. A time variance is allowed, by default, upto 5 minutes.

The KDC now invents a session key for the client, to share with the Service Provider. The KDC encrypts one copy of this session key with User's logon session key.
It embeds another copy of the session key in a ticket, along with Timestamp, TTL and User's authorization data, such as SIDs and encrypts the ticket with the Validation Server's (Service Provider) long-term key. The KDC\Ticket_Granting_Service then sends these credentials back to the client in a Kerberos Ticket-Granting Service Reply (KRB_TGS_REP).

Here is the key that makes Kerberos tick. The ticket that was just generated has a session key encrypted with User's Logon Session key so only user can extract it. It also has a copy of the same session key encrypted with the Service Provider's long-term key so that only that server could decrypt it. This key is retrieved from AD by the KDCSVC.

When the client receives the reply, it uses User's logon session key to decrypt the Service Session key to use with the service provider, and stores the key in its credentials cache. Then it extracts the ticket to the service and stores that in its cache. This ticket is still in encrypted format since the User can NOT decrypt it.

At this point, the client has cached 2 session keys. Please do not get confused with the word "Session" since both these keys server a different purpose and cannot be used interchangeably.

- The logon session key - used to talk to KDC
- The Service Session Key - used to talk to the Validation Server (Service Provider)

CS Exchange - Client is now ready to talk to the service provider

---

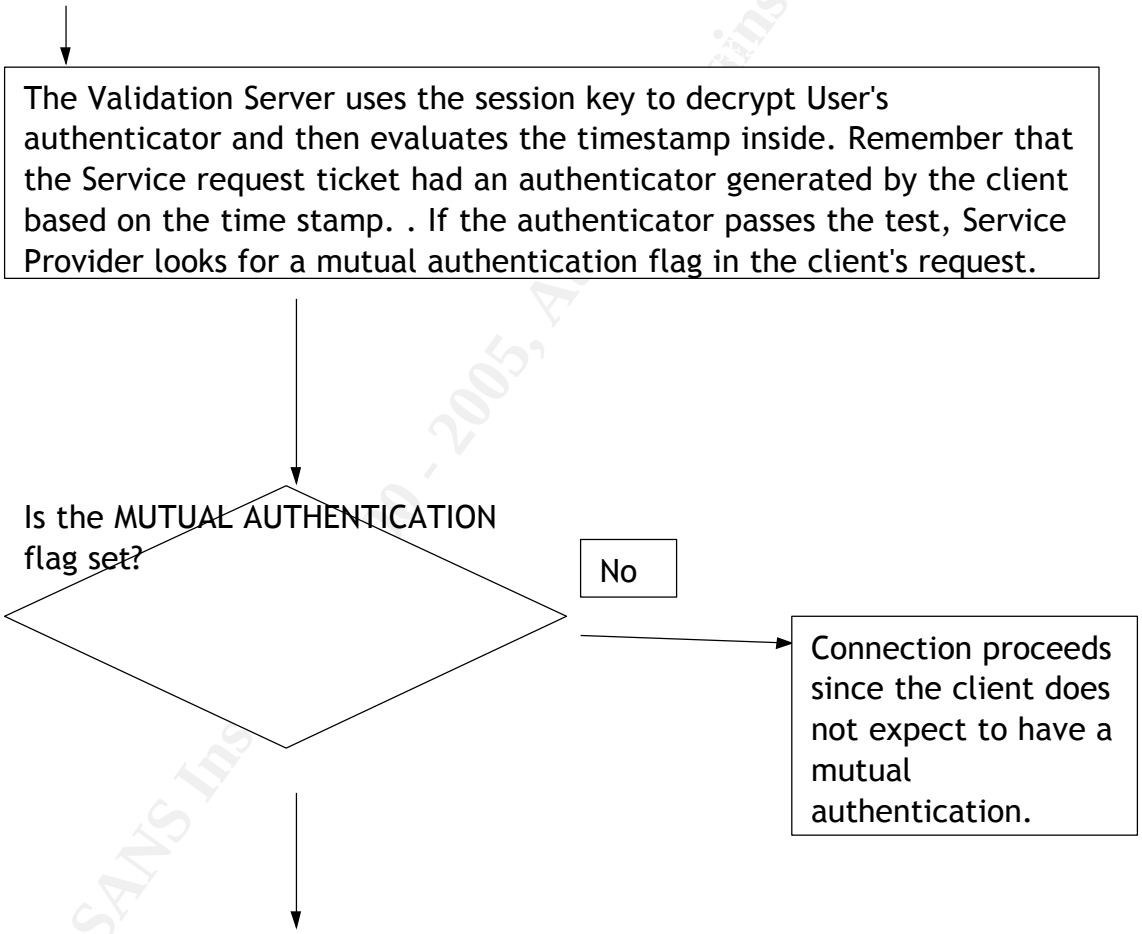
The Kerberos client on User's workstation requests service from Service Provider by sending Service Provider a Kerberos Application Request (KRB_AP_REQ).

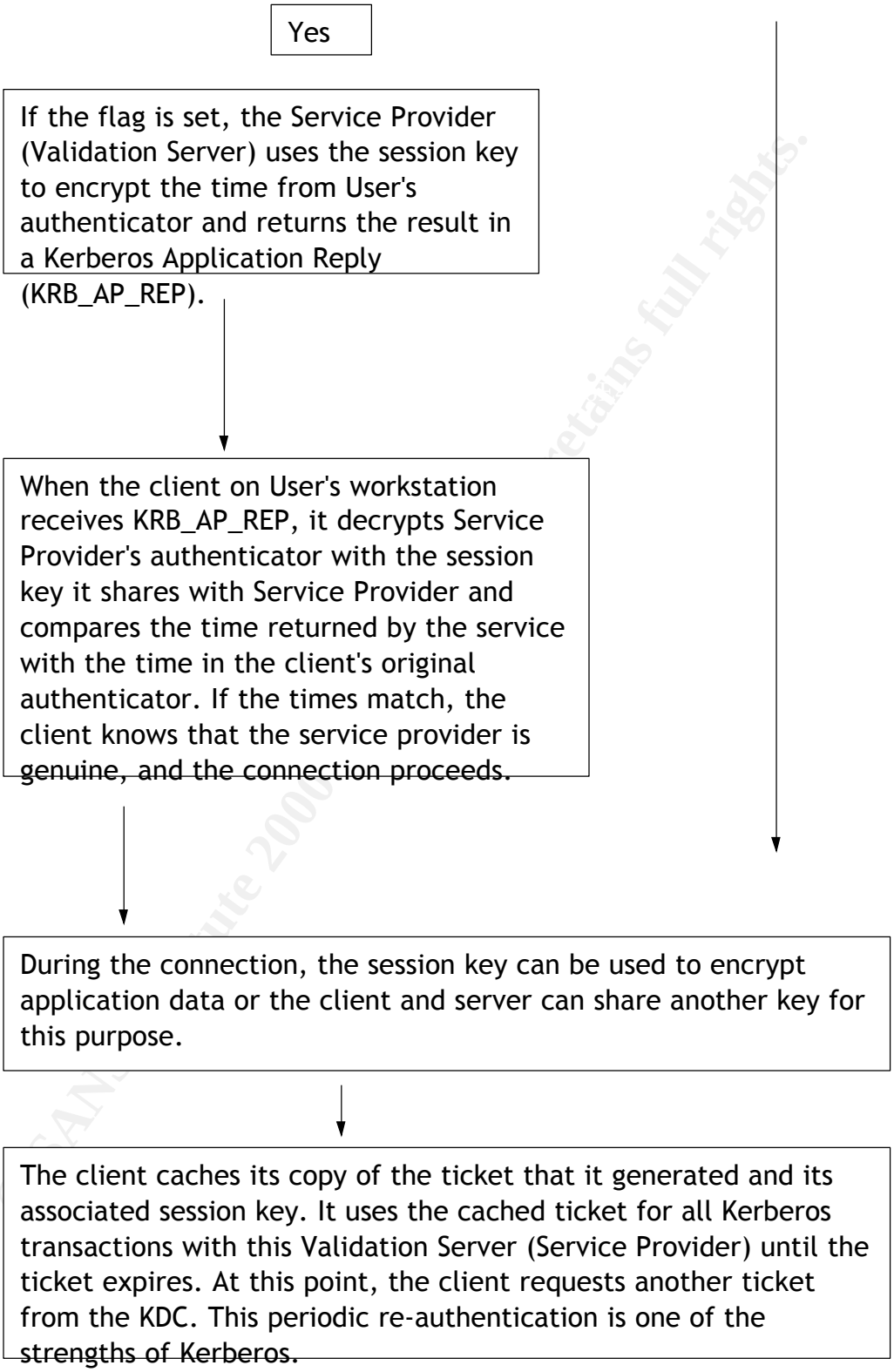
The Service Request Message consists of -

- Authenticator encrypted with the Service Session key.
- The ticket obtained in the TGS Exchange
- A flag indicating whether the client wants mutual authentication. (The setting of this flag is one of the options in configuring Kerberos. The user is never asked.)

Service Provider or the VALIDATION SERVER side processing

The Service Provider (The Validation Server) receives KRB_AP_REQ, decrypts the ticket, and extracts User's authorization data and the session key. The validation server is able to extract the information because the Service Request ticket contains the session key that is encrypted with the Validation Server's Long-term key.
IN our analogy example, this is similar to the Flight Attendant verifying the ticket that you have against her own records.





Now that I have presented the analysis of the transactions, I would like to

conclude the paper with a few points.

- Please do not make an impression that so much of processing is an overhead to the authentication and therefore the Protocol transactions would be very slow. NO. It's not true. Microsoft's implementation of the protocol is very fast and the very fact that the client caches the tickets locally makes further transactions real quick.
- There are very few operational or administrative requirements. Windows 2000 computers ALWAYS use Kerberos when authenticating with other Windows 2000 computers. An administrator may choose to configure certain parameters but it is NOT a mandatory requirement. This could be done using the Configurable POLICIES. From a user perspective, s/he only sees the usual Windows Logon Screen.
- Transitive trusts are enabled by default in a Windows 2000 domain without any additional configuration.
- The Kerberos client is not loaded as a separate service but is loaded as a part of LSASS.
- Unfortunately, as of this date, there are not many tools available from Microsoft for configuring, administering and maintaining Kerberos. A few tools from Microsoft TechNet are listed below.
  - Klist.exe - View and delete Kerberos tickets for current session
  - Kerbtray.exe - Same as above; has a Tray icon.
- The resource kit contains 2 more utilities KSETUP.exe and KTPASS.exe that could be used to configure the Microsoft Domain Controller (KDC) to act as a MIT Distributed version of Kerberos KDC.
- Default parameter values - (Could be changed using the Policy Editor)
  - Enforce User Logon restriction: Enable
  - Maximum lifetime that a TGT could be renewed: 7 days. After 7 days the user will have to re-authenticate
  - Maximum Service Ticket lifetime - 600 Minutes (10 Hrs)
  - Maximum tolerance for clock synchronization - 5 minutes
  - Maximum User (TGT) ticket lifetime - 600 minutes.

I Hope I have provided some insight of the Kerberos transactions. There is a lot of scope to further explore the protocol. A few thoughts or points are

- Ticket Details
- Ticket lifetimes and Expirations and how the protocol handles these; should the default values of these parameters be changed?
- Forwarding / Proxying / Renewing of tickets
- Delegation of Authentication
- Authentication across a domain
- Interoperability with other implementations of Kerberos (Both, Server and

Client)

References

Please note that with the dynamic nature of the Internet, any of the following links may not be valid for a long time. But they are definitely valid as of the date of this writing. (July 27, 2001). This is especially true with the links on the Microsoft and About.com site. If you find that these links are not valid any more, please use a search on the main pages of the respective site with the keyword search 'Kerberos'.
http://www.microsoft.com/technet
http://home.about.com/compute/

1. SANS Reading room reference for Kerberos Terminology
   Author - Richard Tufaro Jr. Dated -August 29, 2000
   http://www.sans.org/infosecFAQ/authentic/kerberos.htm
   There are other articles in the reading room that are a nice read in the related areas, such as PKI in windows 2000. Here is a link to the main page of the SANS reading room. Please search under groups "Windows 2000" "Encryption and VPN" and "Authentication".
   http://www.sans.org/infosecFAQ/index.htm

2. Microsoft's extremely detailed white paper on Kerberos
   http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/tcevents/itevents/kerberos.asp

3. Another paper from Microsoft TechNet - Overview of Kerberos
   http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/tcevents/itevents/kerberos.asp

4. Kerberos Reference pages on ABOUT.com – Excellent Links related to Kerberos.
   http://windows2000.about.com/cs/kerberos/index.htm?rnk=r1&terms=Kerberos

5. The Original MIT Distribution reference page
   http://web.mit.edu/kerberos/www/

6. Definitions, Model, Strengths and Weaknesses of Kerberos - MIT Distribution

This is an old link but I am adding it here so any person interested in knowing the historical growth of Kerberos could get more information. Unfortunately the Author information could not be found.
http://www.oit.duke.edu/~rob/kerberos/kerbauth.html

7. Kerberos Reference Page - Nice links collection.
   Author - Derrick Brashear
   This is an old link (5/25/95) but still has nice links to other Kerberos related material.
   http://www.contrib.andrew.cmu.edu/%7Eshadow/kerberos.html

8. Kerberos FAQ
   http://www.faqs.org/faqs/kerberos-faq/general/

9. Kerberos RFC
   http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1510.html