Global Information Assurance Certification Paper

An Overview of the Kerberos Authentication Protocol

Pam Todaro

October 14, 2003

SANS Security Essentials Certification
Practical Assignment version 1.4b option 1

Contents

Introduction

In today's open networks, authentication is a fundamental building block for a secure network environment. By using strong cryptography, the Kerberos authentication protocol allows both client and server to provide identity to each other over an insecure network connection. Communications between client and server, using the Kerberos protocol, is encrypted to ensure privacy and data integrity.

Due to the complexity of the Kerberos authentication protocol, this overview of its utilization characteristics will be very broad. This paper will expound on some of the benefits gained by using the Kerberos authentication protocol rather than the Windows NT LAN Manager protocol. An explanation will also be provided on how the Kerberos protocol works and how the utilization of this protocol can enable strong authentication for client/server applications.

Before starting, I would like to provide the location of a paper that I believe will benefit anyone who takes the time to read it. This material was not used in this paper because it was written to provide readers with a fundamental understanding of the Kerberos V4 protocol. "Designing an Authentication System: A Dialogue in Four Scenes" at http://web.mit.edu/kerberos/www/dialogue.html.

Definitions

Two terms that are used throughout this paper are defined for the reader's understanding:

User - An individual who uses a program or service while accessing a Windows 2000 network.[4]

Client - A client can be either a person or a program. Often a network application consists of two parts: a client side program running on one machine that request a remote service and a server side program that performs the service. The client will frequently contact the server on behalf of a user.[4]

Authentication for Windows 2000

The two choices for network authentication within Windows 2000 domains are: Kerberos authentication protocol and Windows NT LAN Manager.[1] When a Windows Domain is converted from a mixed to native mode, no down-level Windows NT controllers exist Kerberos can be utilized as the default network authentication protocol. To maintain system compatibility, integrity and reliability with down level clients or servers, the Windows NT LAN Manager protocol must be retained. Any Windows 2000 standalone computers will also require the Windows NT LAN Manager protocol to authenticate their logons. Once all

network clients are capable of Kerberos authentication the authentication protocol should be switched from the NT LAN Manager to the more flexible, efficient, and secure Kerberos authentication protocol.

Benefits of Kerberos

Some of the benefits gained by using the Kerberos authentication protocol are as follows:

1. Faster connections:[1] When using the NT LAN Manager protocol an application server is required to contact a domain controller in order to authenticate each client. When the Kerberos authentication protocol is being used, it is not necessary for the server to contact a domain controller each and every time a client needs to be authenticated. The client is responsible for acquiring and storing credentials for each server it needs to access. The credentials can be reused to access the same servers throughout the logon session without the client having to obtain new credentials.

2. Mutual authentication:[1] Unlike Windows NT LAN Manager protocol, Kerberos does not assume that servers and clients are genuine and accordingly allows clients to verify a server's identity, or one server to verify the identity of another server. The Kerberos protocol will not allow a connection to be established until both ends of the network have been appropriately verified.

3. Simplified trust management:[1] In Windows 2000 domains trust between security authorities is by default a two-way and transitive link. Many domains of a large network can be organized in such a manner wherein credentials issued by the security authority for any domain are accepted everywhere in a tree of mutual trust. If more than one tree is present on the network, credentials issued by any domain will be accepted throughout the forest.

Mutual authentication

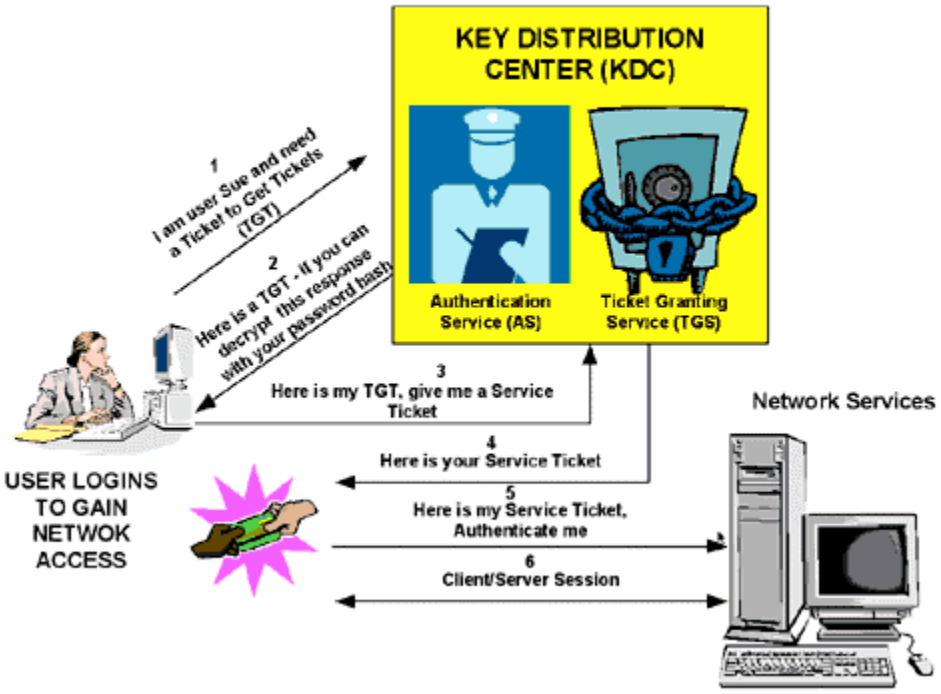
Clients logging on to a network cannot be certain that their initial transactions will not be monitored or modified by an unauthorized person. An attacker can easily pose as a legitimate server or can tamper with communications between an authorized client and a legitimate server. For this reason, the Kerberos authentication protocol always assumes that the network is not secure and provides for mutual authentication between partners before a network connection is even opened between them. The Kerberos protocol uses a cryptographic key that is shared by the partners and is used by them to verify one another's identity. The shared key is symmetric and a single key must be capable of both

encryption and decryption.[1] Each party proves knowledge of this key by either decrypting or encrypting a piece of information.

Kerberos components

The kerberos protocol consists of a client, a server, and a trusted third party to mediate between them. The Key Distribution Center is the trusted intermediary and is implemented as a domain service. The Key Distribution Center runs on every domain controller and uses the Active Directory as its account database. If the network contains more than one domain, it also receives various information regarding users from the Global Catalog. Both the Key Distribution Center and the Active Directory Service are located on physically secure domain controllers and run in the process space of the Local Security Authority.[1] These services cannot be stopped; however, the Kerberos protocol pre-authentication feature may be disabled for specific users in order to support some applications that don't uphold the security feature.[2] Allowing each domain to have several domain controllers that can accept requests to the domain's Key Distribution Center insures availability of the services.

Kerberos Ticket exchange



[2]

Key Distribution Center

The Key Distribution Center performs two service functions:

1. Authentication Service (AS): When a user initially logs on to the network, a login name and password must be presented to the Authentication Service portion of the Key Distribution Center within the user's domain. This portion of the Key Distribution Center is also responsible for issuing Ticket Granting Tickets good for admission to the Ticket-Granting Service in its domain. [3]

2. Ticket-Granting Service: This service is responsible for issuing tickets to other services. The client requesting access to a service must present to the Ticket-Granting Service, in the services account domain, a request for a session ticket and a Ticket Granting Ticket. If the service being requested is not in the Ticket-Granting Service's account domain, the client must use the referral process that starts at the Ticket-Granting Service in the user's account domain and ends at the Ticket-Granting Service in the service's account domain.

Long-term Key

The Key Distribution Center has access to information regarding each security principal (user, computer or service). [1] The Key Distribution Center also has access to a cryptographic key known only to the Key Distribution Center and the security principal. This key is commonly known as a long-term key and is used in exchanges between the Key Distribution Center and the security principal. The client's copy of the long-term key is usually created when a user logs on to a Windows 2000 network and the Kerberos client, on the workstation, converts the user's password to an encryption key by entering the password into a one-way hashing function. The Kerberos client saves the long-term key in its credentials cash located in volatile memory (not on disk). Once the client has a long-term key, it sends the Key Distribution Center's Authentication Service a message consisting of two parts:

1. Part one of the message identifies the user and requests access to the Ticket-Granting Service.

2. Part two usually contains a timestamp encrypted with the user's copy of the long-term key.

The Key Distribution Center uses its copy of the long-term key to decrypt the message and validate the timestamp. The Key Distribution Center obtains its copy of the long-term key from the domain's Active Directory. When the timestamp check occurs, it is critical that times are synchronized in the network. The timestamp check must be processed within a certain time frame or the logon will be rejected. The logon will also be rejected if the time, in the timestamp, is earlier than a time already requested for a session by the same logon ID. The timestamp evaluation is critical in an insecure network environment, to avoid replay.

Requesting a Ticket Granting Ticket

When the Key Distribution Center's Authentication Service receives a request for a Ticket Granting Ticket it checks the Active Directory in the user's account domain. The user's account record contains attributes for the user's security Identifier as well as security identifiers for any domain security groups to which the user belongs. If the network contains more than one domain, the Global Catalog is also queried for any universal groups that include the user or one of the user's domain security groups.

Key Distribution Center's reply

After the Key Distribution Center's Authentication Service verifies the client, it replies to the client in two ways:

1. First, the Key Distribution Center develops a logon session key and encrypts a copy of it with the client's long-term key.

2. Second the Key Distribution Center creates a Ticket Granting Ticket and places the list of security identifiers returned by its query into the Ticket Granting Ticket's authorization data field. The Ticket Granting Ticket will also contain another copy of the logon session key. Once the ticket has been completed, it is encrypted with the Key Distribution Center's long-term key. [1].

When the client receives a copy of the encrypted logon session key and the Ticket Granting Ticket, it uses its cached copy of the user's long-term key to decrypt its copy of the logon session key. The logon session key is temporary and is only valid until the Ticket Granting Ticket expires or the user logs off. Once the client has the logon session key, it can discard the long-term key because it will use the logon session key for any future communications.

Session tickets and keys

When a client needs access to a service, it first checks its credentials cache for a session ticket to that service. If the client does not have a session ticket, it checks the cache again for a Ticket Granting Ticket. If the client has a Ticket Granting Ticket in its credentials cache, it retrieves the corresponding logon session key from the cache. The client uses the logon session key to prepare an authenticator. The authenticator and the Ticket Granting Ticket are sent to the Key Distribution Center, along with a request for a session ticket to the service. If the Ticket-Granting Service approves the client's request, it generates a

temporary private key called a session key for both the client and server. The Ticket-Granting Service also creates a session ticket that contains the server's copy of the session key as well as information about the client. The Key Distribution Center uses the key it shares with the server to encrypt the session ticket, and then sends a reply back to the client that includes the session ticket and the client's copy of the session key. When the client receives the Key Distribution Center's reply, it extracts the session ticket and the client's copy of the session key and stores both in its credentials cache. It is the responsibility of the client to present the session ticket to the server each time it wants access to the server. The session ticket can be used any number of times when accessing the same server. Both the Key Distribution Center and the target server benefit from this because the client does not have to request a session ticket from the Key Distribution Center each time it needs access to the same server and, likewise, the target server does not need to keep the client's session ticket after a connection has been terminated.

When requesting admission to a server, the client must send the server a message consisting of both the session ticket and an authenticator. The server decrypts the session ticket with its secret key and extracts the session key. It then uses the session key to decrypt the client's authenticator. If the server is able to decrypt the client's authenticator, it knows that the Key Distribution Center issued the client's credentials. The client can request that the server authenticate itself to the client. If the client has asked for mutual authentication, the server uses its copy of the session key to encrypt the timestamp from the client's authenticator and returns the result to the client as its authenticator. The client decrypts the server's authenticator and compares the timestamp with that of its original authenticator. If the timestamp checks out, the client knows that the server is legitimate. Both the server and the client have been authenticated to each other.

The Key Distribution Center does not need to keep track of its messages or verify that data reaches the intended address. Only someone who knows the client's secret key can decrypt the client's copy of the session key; likewise, only someone having knowledge of the server's secret key can read what is inside the ticket

All tickets have a start time and an expiration time. A client can use a session ticket any time after the start time but before the expiration time to gain access to a service. A valid session ticket can be used any number of times to gain access to the same service. When the user logs off, the credentials cache is flushed and all session tickets, as well as all keys, are destroyed.

Defending against attacks

To minimize the risk that a session ticket or the corresponding session key may be compromised, an administrator can set the maximum lifetime for tickets using the time element in the Kerberos policy.

Two ways an administrator can set Kerberos policy to defend against attacks on session keys are:

1. Have tickets renewed at relatively short intervals. When tickets are renewed, a new session key is issued, minimizing the value of a compromised key.

2. The Kerberos policy can be set to allow renewable tickets. The session keys, in a renewable ticket, can be refreshed periodically without issuing a completely new ticket. Renewable tickets have two expiration times set in each ticket.

    a. The first expiration time limits the life of the current instance of ticket usage and is held in the end-time field. With both non-renewable and renewable tickets, end-time equals the value of the start-time field plus the maximum ticket life specified in the Kerberos policy. Before the end-time is reached, a client must send the renewable ticket and a fresh authenticator to the Key Distribution Center.

    b. The second expiration time sets a limit on the cumulative lifetime of all instances of the ticket and is held in the renew-till field. The value in this field equals the ticket's start-time plus the maximum cumulative ticket life specified by the Kerberos policy. When the Key Distribution Center receives a ticket for renewal, it checks to see that the renew-till time has not elapsed. If the time has not elapsed, the Key Distribution Center issues a new instance of the ticket with a later end-time and a new session key. Upon expiration of the renew-till time, the ticket expires and is no longer valid for renewal.

Ticket times

A client requesting a ticket to a service from the Key Distribution Center may request a specific start time. If the requested time is in the past or is missing from the request, the Key Distribution Center will set the value of the start-time field to the current time.

The client must include a desired expiration time when submitting a request for a ticket to a service. The maximum ticket life, fixed by the Kerberos policy, is

added to the value of the ticket's start-time field, in order to determine the value of the ticket's end-time field. The Key Distribution Center compares the result with the requested expiration time; whichever comes first will determine the ticket's end-time.

Expired or outdated tickets

Clients are not notified when session tickets or Ticket Granting Tickets are about to expire. The Key Distribution Center does not keep track of any transactions with clients beyond short-term records. The only reason the Key Distribution Center keeps short-term records is to prevent replay attacks.

If a client presents an expired session ticket while trying to authenticate a new connection to a server, the server returns an error message. The connection is not interrupted, however, if the session ticket, used to authenticate, expires during the connection.

When the client presents an outdated Ticket Granting Ticket while requesting a session ticket from the Key Distribution Center, the Key Distribution Center responds with an error message. The client must request a new Ticket Granting Ticket and requires the user's long-term key to perform this function. If the client discarded the long-term key, when receiving the logon session key, the client must ask the user for his or her logon password and create a new long-term key.

Special situations

An application that allows a client to connect to a server which must itself connect to a second server presents a special situation for the Kerberos protocol. To deal with this situation, the Kerberos protocol uses delegation of authentication. The client can delegate authentication to a server by communication to the Key Distribution Center, stating that the server is authorized to represent the client. Delegation of authority can be done in one of two ways:

1. The client can obtain a proxy ticket from the Ticket-Granting Service by presenting a Ticket Granting Ticket along with a request for a ticket to the back-end server. The request must communicate that the client requires a proxy ticket and contain the name of the server representing the client. When the Key Distribution Center receives the client's request, it reviews the Kerberos policy to ensure that proxy tickets are allowed. If proxy tickets are allowed, the Key Distribution Center creates a ticket for the back-end server and sends the ticket to the client. It is the responsibility of the client to forward the ticket to the front-end server. The front-end server uses the ticket to access the back-end server.

2. Clients can obtain forwarded tickets by indicating to the Key Distribution Center the name of the front-end server that is to act on its behalf. This type of ticket delegates the task of obtaining tickets for the back-end server to the front-end server. When the Key Distribution Center receives a request for a forwarded ticket, it once again reviews the Kerberos policy. If the Kerberos policy permits forwarded tickets, the Key Distribution Center creates a Ticket Granting Ticket for the front-end server to use in the client's name and sends the ticket to the client. It is the responsibility of the client to forward the Ticket Granting Ticket to the front-end server. When the front-end server requests a ticket to the back-end server, it must submit the Ticket Granting Ticket to the Key Distribution Center. The Key Distribution Center returns the ticket to the front-end server, not to the client.

Authentication across domain boundaries

As discussed previously, the function of the Key Distribution Center is divided into two distinct services: an Authentication Service whose job is to issue Ticket Granting Tickets, and a Ticket-Granting Service whose job is to issue session tickets. This division of labor allows the Kerberos protocol to operate across domains and utilizes the Authentication Service portion of the Key Distribution Center in one domain to retrieve session tickets from the Ticket-Granting Service portion of the Key Distribution Center in another domain. [2] This authentication across domain boundaries is possible because the Key Distribution Centers in each domain share an inter-domain key. An inter-domain key is automatically created when two domains establish a trust relationship in the Windows 2000 environment. Once a trust relationship has been established between domains, the Ticket-Granting Service of each domain is registered as a security principal with the other domain's Key Distribution Center.[1]

Referral tickets

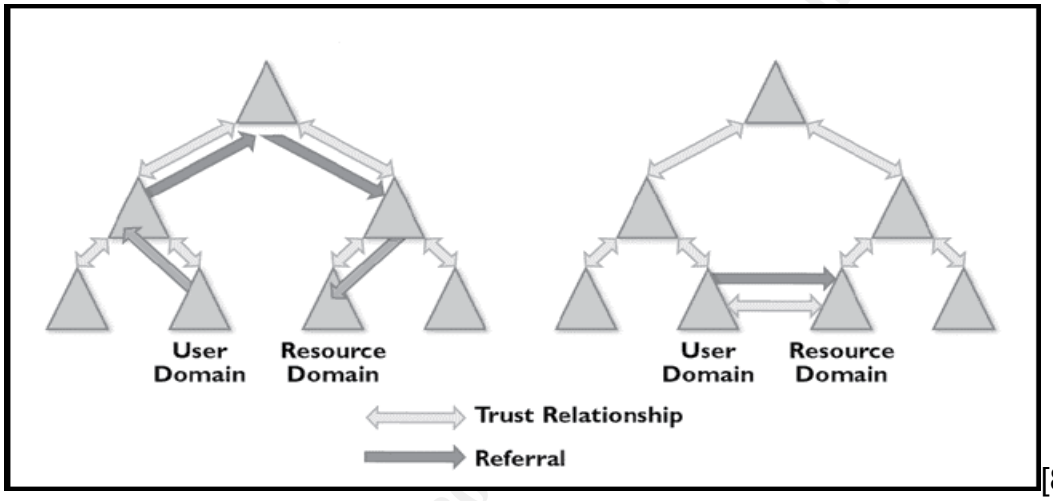
If the user needs access to a server in another domain, the Kerberos client on the user's workstation must still send a request for a session ticket to the Ticket-Granting Service in the user's account domain. The Key Distribution Center will recognize that the desired server is not a security principal in its domain and it will reply by sending the client a referral ticket. This ticket is merely a Ticket Granting Ticket that has been encrypted with the inter-domain key that the Key Distribution Centers in both domains share. By sharing an inter-domain key, the various domains can authenticate across the boundaries within a tree.

Using the referral ticket, the client prepares a second request for a session ticket. The client sends the request to the Ticket-Granting Service in the server's account domain. When the Ticket-Granting Service receives the request, it uses its copy of the inter-domain key to decrypt the referral ticket. If the Ticket-

Granting Service is able to successfully decrypt the referral ticket, it sends the client a session ticket to the desired server in its domain.

The referral process could become very complicated in networks that have more than one domain. In a network consisting of two or more domains, the Key Distribution Center in each domain could establish a direct link to the Key Distribution Center of every other domain, in each case sharing a different inter-domain key. In a very large network, these links could become very complex, and in some cases even unmanageable.

Referral Path



[8]

The Kerberos protocol uses a referral path in large networks with more than two domains in order to resolve complications in the referral process. When a client requests access to a server not in his or her domain, the client will need to travel a referral path, through one or more intermediate domains. The client will need to request a session ticket from the Key Distribution Center in each domain (the user's account domain, all domain/s it passes through, and the server's account domain).

Summary

Kerberos is a powerful authentication protocol that is transparent to the user except when entering the initial password or smart-card. The Kerberos protocol provides authentication and strong cryptography to secure information systems across an entire network or enterprise. The protocol is a highly effective solution to network security problems.

References:

(1). "Windows 2000 Kerberos Authentication"
http://www.microsoft.com/windows2000/docs/kerberos.doc

(2). "Kerberos Explained"
http://www.microsoft.com/TechNet/prodtechnol/windows2000serv/maintain/kerberos.asp

(3). "Kerberos V5 Authentication Protocol"
http://www.microsoft.com/TechNet/prodtechnol/winxppro/reskit/prdp_log_ovqw.asp

(4). "Kereros Overview – An Authentication Service for Open Network Systems"
http://www.cisco.com/warp/public/106/1.html

(5). COMPUTERWORLD "Kerberos"
http://www.computerworld.com/printthis/2000/0,4814,46517,00.html

(6). "Kerberos: The Network Authentication Protocol"
http://web.mit.edu/kerberos/www/

(7). "The Moron's Guide to Kerberos, Version 1.2.2"
http://www.isi.edu/gost/brian/security/kerberos.html

(8). "Windows 2000 Authentication"
http://www.windowsitlibrary.com/Content/617/06/4.html