**Kerberos Notes**

**Kerberos** is an authentication protocol and a software suite implementing this protocol. Kerberos uses symmetric cryptography to authenticate clients to services and vice versa. For example, Windows servers use Kerberos as the primary authentication mechanism, working in conjunction with Active Directory to maintain centralized user information. Other possible uses of Kerberos include allowing users to log into other machines in a local-area network, authentication for web services, authenticating email client and servers, and authenticating the use of devices such as printers. Kerberos is a protocol for authenticating service requests between trusted hosts across an untrusted network.

Kerberos was created by MIT as a **solution to these network security problems**. The Kerberos protocol uses **strong cryptography** so that a client can prove its identity to a server (and vice versa) across an insecure network connection. After a client and server has used Kerberos to prove their identity, they can also encrypt all of their communications to assure privacy and data integrity as they go about their business.

Kerberos uses the concept of a ticket as a token that proves the identity of a user. Tickets are digital documents that store session keys. They are typically issued during a login session and then can be used instead of passwords for any Kerberized services. During the course of authentication, a client receives two tickets:
- A ticket-granting ticket (TGT), which acts as a global identifier for a user and a session key
- A service ticket, which authenticates a user to a particular service

These tickets include time stamps that indicate an expiration time after which they become invalid. This expiration time can be set by Kerberos administrators depending on the service.
To accomplish secure authentication, Kerberos uses a trusted third party known as a key distribution center (KDC), which is composed of two components, typically integrated into a single server:
– An authentication server (AS), which performs user authentication
– A ticket-granting server (TGS), which grants tickets to users
The authentication server keeps a database storing the secret keys of the users and services. The secret key of a user is typically generated by performing a one-way hash of the user-provided password. Kerberos is designed to be modular, so that it can be used with a number of encryption protocols, with AES being the default cryptosystem. Kerberos aims to centralize authentication for an entire network—rather than storing sensitive authentication information at each user’s machine, this data is only maintained in one presumably secure location.

To start the Kerberos authentication process, the initiating client sends a request to an authentication server for access to a service. The initial request is sent as plaintext because
no sensitive information is included in the request. The authentication server retrieves the initiating client’s private key, assuming the initiating client’s username is in the KDC database. If the initiating client’s username cannot be found in the KDC database, the client cannot be authenticated and the authentication process stops. If the client’s username can be found in the KDC database, the authentication server generates a session key and a ticket granting ticket. The ticket granting ticket is timestamped and encrypted by the authentication server with the initiating client’s password. The initiating client is then prompted for a password; if what is entered matches the password in the KDC database, the encrypted ticket granting ticket sent from the authentication server is decrypted and used to request a credential from the ticket granting server for the desired service. The client sends the ticket granting ticket to the ticket granting server, which may be physically running on the same hardware as the authentication server, but performing a different role.

The ticket granting service carries out an authentication check similar to that performed by the authentication server, but this time sends credentials and a ticket to access the requested service. This transmission is encrypted with a session key specific to the user and service being accessed. This proof of identity can be used to access the requested “kerberized” service, which, once having validated the original request, will confirm its identity to the requesting system. The timestamped ticket sent by the ticket granting service allows the requesting system to access the service using a single ticket for a specific time period without having to be re-authenticated. Making the ticket valid for a limited time period makes it less likely that someone else will be able to use it later; it is also possible to set the maximum lifetime to 0, in which case service tickets will not expire. Microsoft recommends a maximum lifetime of 600 minutes for service tickets; this is the default value in Windows
Server implementations of Kerberos.

Kerberos Advantages

• The Kerberos protocol is designed to be secure even when performed over an insecure network.
• Since each transmission is encrypted using an appropriate secret key, an attacker cannot forge a valid ticket to gain unauthorized access to a service without compromising an encryption key or breaking the underlying encryption algorithm, which is assumed to be secure.
• Kerberos is also designed to protect against replay attacks, where an attacker eavesdrops legitimate Kerberos communications and retransmits messages from an authenticated party to perform unauthorized actions.
  – The inclusion of time stamps in Kerberos messages restricts the window in which an attacker can retransmit messages.
  – Tickets may contain the IP addresses associated with the authenticated party to prevent replaying messages from a different IP address.
  – Kerberized services make use of a “replay cache,” which stores previous authentication tokens and detects their reuse.
• Kerberos makes use of symmetric encryption instead of public-key encryption, which makes Kerberos computationally efficient
• The availability of an open-source implementation has facilitated the adoption of Kerberos.

Kerberos Disadvantages

• Kerberos has a single point of failure: if the Key Distribution Center becomes unavailable, the authentication scheme for an entire network may cease to function. – Larger networks sometimes prevent such a scenario by having multiple KDCs, or having backup KDCs available in case of emergency.
• If an attacker compromises the KDC, the authentication information of every client and server on the network would be revealed.
• Kerberos requires that all participating parties have
synchronized clocks, since time stamps are used.