Kerberos

- **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.
- Services using Kerberos authentication are commonly referred to as “Kerberized”.
Kerberos Tickets

• 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.

Kerberos Servers

• 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.
Kerberos Authentication

- The client and authentication server authenticate themselves to each other.
- The client and ticket-granting server authenticate themselves to each other.
- The client and requested service authenticate themselves to each other, at which point the service will be provided to the client.

Authentication Details

1. The user provides a username and password on the client machine, which is cryptographically hashed to form the secret key for the client.
2. The client contacts the AS, which replies with the following items:
   - The *client-TGS session key*, $K_{CT}$, encrypted using the client’s secret key, $K_C$ (which the AS has stored in its database).
   - The *ticket-granting ticket (TGT)*, encrypted with the secret key of the TGS, $K_T$ (also stored in the AS database). The TGT includes key $K_{CT}$ and a validity period.
3. The client decrypts the TGS session key $K_{CT}$ using $K_C$. To request a service, the client sends the following two messages to the TGS:
   - The TGT (still encrypted using the TGS’s secret key, $K_T$) and the name, $S$, of the service being requested.
   - An authentication token consisting of the client ID and time stamp, encrypted using the client-TGS session key $K_{CT}$.

4. The TGS decrypts the TGT using $K_T$, thus retrieving the client-TGS session key $K_{CT}$ and the validity period of the TGT. If the current time is within the validity period, the TGS decrypts the authentication token with key $K_{CT}$ and sends two messages to the client:
   - A new client-server session key, $K_{CS}$, encrypted with $K_{CT}$.
   - A client-to-server ticket, encrypted using the specific service’s secret key, $K_S$, which is known to the TGS. This ticket contains the client ID, network address, validity period, and key $K_{CS}$.

5. After decrypting the client-server session key $K_{CS}$, the client authenticates itself to service $S$ by sending the following two messages:
   - The client-to-server ticket, sent by the TGS in the previous step.
   - The client ID and time stamp, encrypted with $K_{CS}$.

6. The service decrypts the client-to-server ticket using its secret key $K_S$ and obtains the client-server session key $K_{CS}$. Using $K_{CS}$, it decrypts the client ID and time stamp. Finally, to prove its identity to the client, it increments the time stamp by 1 and sends it back to the client reencrypted with $K_{CS}$.

7. The client decrypts and verifies this response using $K_{CS}$. If the verification succeeds, the client-server session can begin.
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.