Kerberos.

Kerberos V5
En standardsikkerhedsprotokol på Internettet til håndtering af godkendelse af en brugers eller et systems identitet. Adgangskoder i Kerberos V5 krypteres, når de sendes over netværkslinjer. De sendes ikke som klartekst.

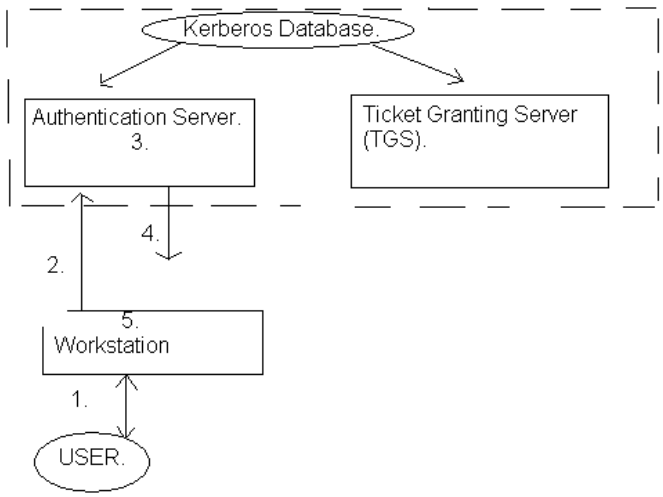
KDC (Key Distribution Center)
En Kerberos V5-tjeneste, der kører på en domænecontroller. Den udsteder billetudstedende billetter (TGT - Ticket-Granting Tickets) og tjenestebilletter til at få netværksgodkendelse i et domæne.

Termen Kerberos server referer generelt til KDC (Key Distribution Center). KDC ´en implementere Authentication Service (AS) og TGS (Ticket Granting Service). KDC´en virker på den måde at den har en kopi af vær enkelt password som er associeret med enhver tjeneste der bruger KDC´en som adgang til et system eller en tjenste.

Hvad er TGS og TGT?

TGT er synonymet for en ”Ticket Granting Ticket.”
TGS er synonymet for en ”Ticket Granting Service.”
Det kan nok se ud som at de to synonymer bliver brugt til det samme, men i virkeligheden referer de til to vidt forskellige ting. TGT er en kerberos ”billet” til TGS og de spiller begge to en speciel rolle i Kerberos systemet.
Når en bruger for første gang bliver verificeret i Kerberos, snakker han sammen med AS (Authentication Service) på KDC´en (Key Distribution Center) for at få en TGT (Ticket Granting Ticket), denne nøgle er krypteret med brugerens password. Når brugeren vil bruge en kerberos service, bruger han TGT (Ticket Granting Ticket) til at snakke med TGS (Ticket Granting Service) som også kører på KDC´en (Key Distribution Center).
Ticket Granting Service verificere brugerens identitet ved hjælp af TGT (Ticket Granting Ticket) som så udsteder en ”billet” til den ønskede service.
Grunden til at TGT (Ticket Granting Ticket) eksistere er den at brugerne således slipper for at skulle taste et password ind, hver gang han ønsker at bruge en Kerberos service.
Hvis TGT (Ticket Granting Ticket) bliver kompromiset, af en hacker kan hackeren kun være ”Forklædt” så længe som ”billetten” gælder.

Kerberos virkemåde.



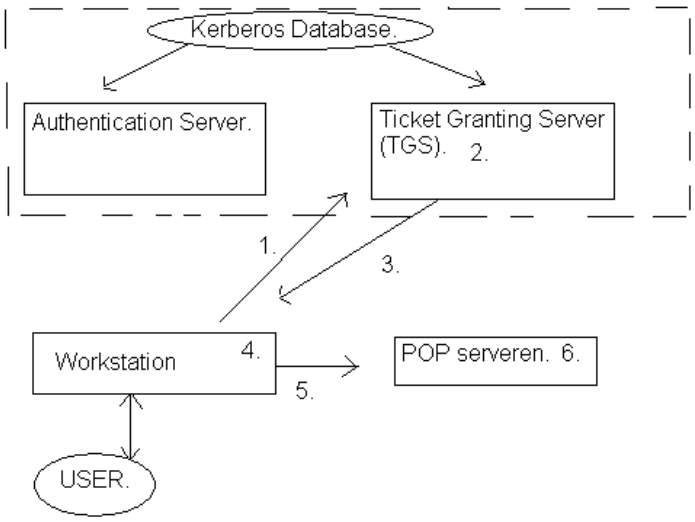
1. Brugeren indtaster sit brugernavn ved login på Workstation.
2. Inden brugeren bliver spurgt om password sendes en pakke til Authentication serveren:
   Besked = [Brugernavn, TGS–navn]

3. Authentication serveren finder en krypteringsnøgle for brugernavn og TGS.
4. Authentication serveren laver en ”billet” som:
   Sealed-ticket = crypt(TGS-key,[brugernavn,TGS-navn,Workstation-adressse, TGS-session-nøgle])
   Authentication serveren kryptere nu sealed-ticket og en besked bygges:
   Sealed-message=crypt(brugerens-krypteret-password ,[TGS-session-nøgle,ticket])
   Og denne besked bliver sendt tilbage til brugerens Workstation.

5. Nu modtager Workstation så beskeden fra Authentication serveren og pakken beder om brugerens password. Brugerens password krypteres, og det krypterede password bruges nu som nøgle til dekryptere beskeden med. Nu har Workstation fået en sealed ticket og en TGS-session-key. ASCII password slettes fra hukommelsen og brugeren lukkes ind.

Her slutter første del med logon sekvensen, men hvordan fungere det så når brugeren vil i kontakt med en Kerberos service. Jeg har som eks. Valgt en pop-server til mails.

Kerberos virkemåde i forbindelse med en POP3 mail server.



1. Workstation laver en bekræfter (authenticator)
   Authenticator=crypt(TGS-session-nøgle,[brugernavn,workstation-adresse, klokkeslet])
   Og en besked:
   Besked=[sealed-ticket,sealed-authenticatior,pop-serveren]
   Denne besked sendes så til TGS serveren.
2. TGS serveren modtager beskeden fra Workstation og dekrypterer sealed-ticket (den var krypteret med TGS´s serverens egen nøgle). Fra ticketen kender TGS nu session-nøgle, og med den kan den dekryptere sealed authenticator. TGS serveren laver nu et lille check (ticket.brugernavn=authenticator.brugernavn samt TGS navnet, samt ticket.workstation-adresse=authenticator.workstation-adresse=socket.workstation-adresse. Endelig checker den tiden i authenticator den skal passe nogenlunde. TGS serveren finder nu pop-serverens nøgle fra kerberos databasen.
3. TGS laver en ny ”billet” (ticket):
   Sealed-ticket=crypt(pop-server-nøgle,[brugernavn,pop-server-navn,Workstation-adresse,pop-session-nøgle])
   Nu laves en ny besked som sendes tilbage til Workstation:
   Besked=crypt(TGS-session-key,[pop-session-key,sealed-ticket])
4. Workstation modtager beskeden, og kan dekryptere den med TGS-session-nøgle, og har nu en pop-session-nøgle.
5. Igen laver Workstation en bekræfter (Authenticator):
   Sealed-authenticator=crypt(pop-session-nøgle,[brugernavn,Workstation-adresse,klokkeslet])
   Og en ny besked laves:
   Besked=[sealed-ticket,sealed-authenticator,pop-server]
   Denne besked krypteres ikke da kun indeholder krypterede pakker og et offenligt navn.

Pakken sendes til pop-serveren.

6. pop-serveren modtager beskeden fra Workstation og dekrypterer sealed-ticket med sin egen nøgle, og finder derved pop-session-nøglen og kan dermed dekryptere sealed-autenticator. Pop-serveren kan nu lave samme verificering som skete i punkt 7.