K e r b e r o s

Single Sign On

Benutzerauthentisierung

Holger.Zuleger@hznet.de

> c

Was ist Kerberos ?

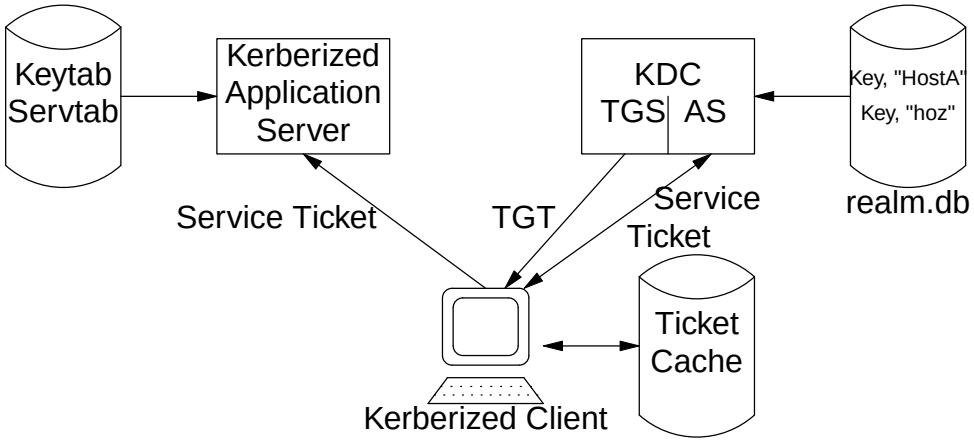
a. Dreiköpfiger Höllenhund aus der griechischen Mythologie

b. "Third party" Authentication System entwickelt am MIT
Ursprünglich auch für Authorization und Accounting gedacht
(Daher die 3 Köpfe)

Was ist Kerberos ? (2)

- Basiert auf symmetrischen Schlüsseln (DES, 3DES, u. a.)
  - ☞ schnell
- **Keine** Übertragung von Passwörtern
  - ☞ sicher
- Erlaubt Single Sign On
  - ☞ praktisch
- Über 15 Jahre alt
  - ☞ bewährt
- Wird stetig weiterentwickelt (KerberosV, GSSAPI, KINK)
  - ☞ aktuell
- Anwendungsbeispiele:
  - — Athena
  - — Benutzerauthentisierung bei Cisco Router und Switchen
  - — Windows 2000 Benutzerauthentisierung
  - — ssh

Überblick



KDC Key Distribution Center

AS Authentication Server

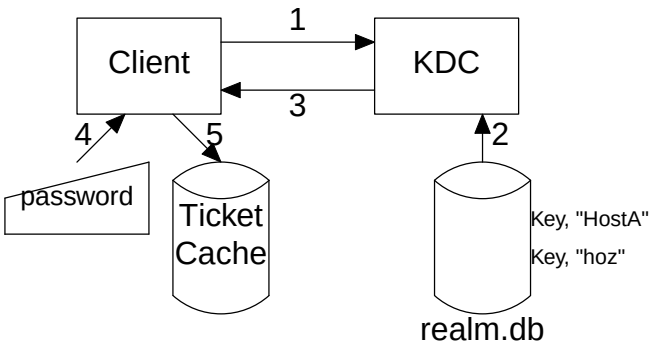
TGS Ticket Granting Server

Ticket Cache Ermöglicht Single Sign On

Keytab File Speichert Anwendungsschlüssel

Kerberized Application z.B. POP3-Server, Telnet-Server, SSH-Server, alle Dienste mit GSS-API Benutzerauthentisierung

Funktionsweise Ticket-Granting-Ticket (1)

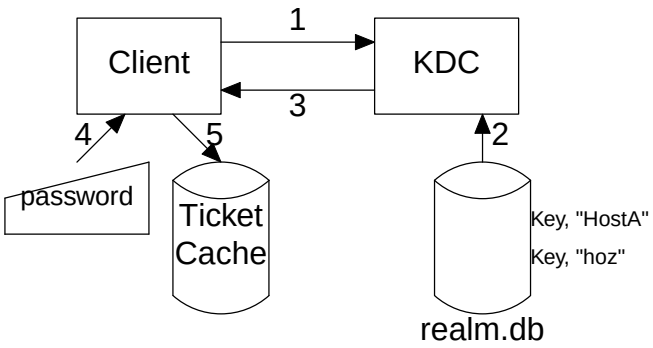


1. Ticket Request an das Key Distribution Center: "Ich bin hoz@HZNET.DE"! Gib mir ein Ticket Granting Ticket.

2. Holen des Benutzerschlüssels und des KDC-Serverkeys aus der Datenbank. Generieren eines SessionKeys.

3. Antwort vom KDC enthält zwei Informationen:
   a. Key: SessionKey + ServerID von KDC, verschl. mit Userpassword
   b. Ticket-Granting-Ticket zur Vorlage gegenüber KDC: Session Key + UserID von "hoz" verschlüsselt mit ServerKey

Funktionsweise Ticket-Granting-Ticket (2)

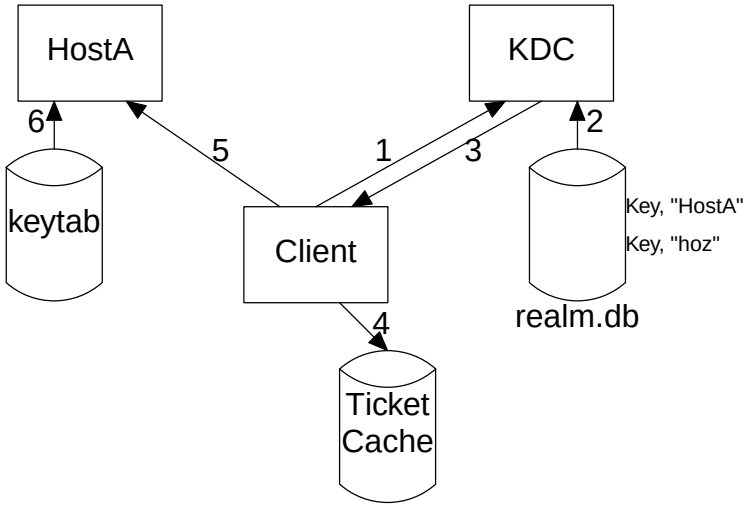


4. Eingabe des Benutzerpasswortes und entschlüsseln des Key-Paketes.
   ☞ Der Client ist im Besitz des Sessionkeys.

5. Speichern des Ticket-Granting-Ticket im Ticket Cache.

Ab diesem Zeitpunkt wird keine Passworteingabe mehr benötigt!

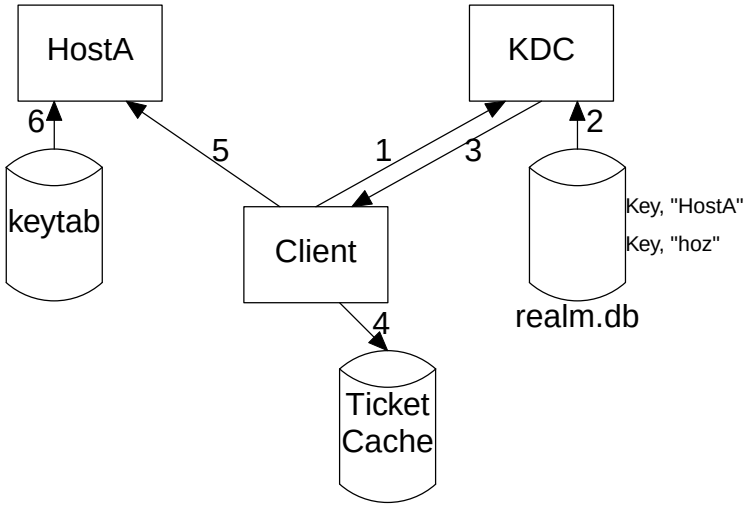
Funktionsweise Service-Ticket (1)



1. Ticket Request an Key Distribution Center mit Vorlage des TGT:
   Gib mir Ticket "service/hostA.hznet.de"

2. Suchen des Benutzer- und Serverschlüssels in Datenbank
   Generieren eines SessionKeys

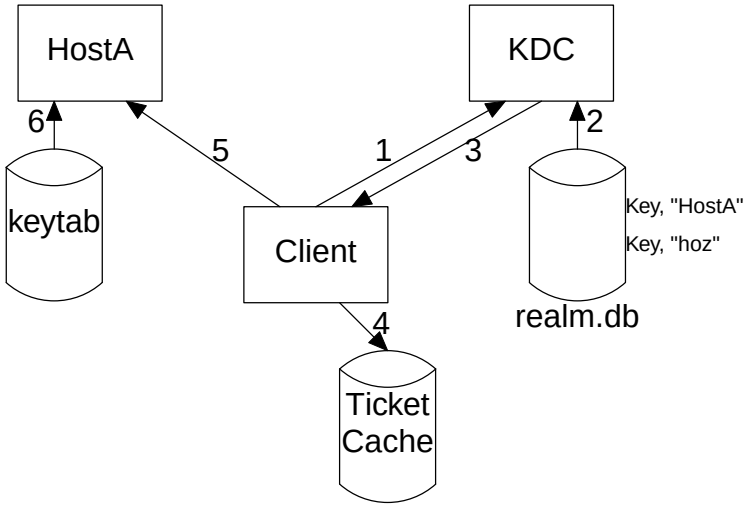
Funktionsweise Service-Ticket (2)



3. Antwort vom KDC mit zwei Informationen:
   a. Key: SessionKey + ServerID "HostA", verschl. mit TGT-Sessionkey
   b. Ticket zur Vorlage gegenüber "HostA":
      Session Key + UserID von "hoz" verschl. mit ServerKey

4. Entschlüsseln des Key-Paketes mit Hilfe des TGT-Sessionkeys
   ☞ Der Client ist im Besitz des Sessionkeys.
   Speichern des Tickets im Ticket Cache.

Funktionsweise Service-Ticket (3)



5. Bilden eines *Authentication credentials* (Ticket + Authenticator) zur Authentisierung gegenüber dem Host.

6. Holen des Serverkeys aus der Keydatei und entschlüsseln des Ticket.
   ☞ Der Server ist im Besitz des Sessionkeys.

7. Überprüfung des Authenticators anhand des Sessionkeys.

Client und Server besitzen jetzt einen Sessionkey!
Mit diesem wird die weitere Kommunikation verschlüsselt abgewickelt

Kerberos Realm

- Ein administrativer Bereich zur Verwaltung von
  - Benutzerkennungen
  - Services auf Rechnern
- Benutzer ID's und Service ID's werden auch *Principals* genannt
- Eine Realm kann hierarchisch strukturiert sein
- Jede REALM hat einen eindeutigen Namen, angelehnt an DNS-Domainnamen
- Der Name der Realm wird per Übereinkunft in Großbuchstaben notiert `HZNET.DE`

Principals

- Principals sind Teilnehmer des Kerberos Authentisierungssystems
- Eindeutige Identifikation eines Benutzers oder eines Dienstes innerhalb einer REALM
- Muß **vor** der Nutzung in der REALM-Datenbank angelegt werden
- Besitzt eine ID und ein „Geheimnis“
- „Geheimnis“ ist nur dem Principal und dem KDC bekannt
- Principals werden durch `kadmin` erzeugt und in der KDC Datenbank gespeichert

Userprincipals

- Die ID besteht aus einem Namen, einem @-Symbol und einer REALM.

```
user@REALM
```

```
hoz@HZNET.DE
hzuleger@HZNET.DE
```

- Als "Geheimnis" wird ein Passwort verwendet.
- Passwort muß vom Benutzer geändert werden können (`kpasswd`)
- BenutzerID kann eine *Instance* beinhalten (durch Slash getrennt)

```
user/instance@realm
```

```
hoz@HZNET.DE
hoz/admin@HZNET.DE
hoz/routeradm@HZNET.DE
```

  Instanzen weisen Benutzer unterschiedliche Rollen zu

Serviceprincipals

- Die ID besteht aus einer Serviceangabe, gefolgt von einem Slash sowie dem vollqualifizierten Rechnernamen des Servers (FQDN).
- Als Serviceangabe sind u.a. `host`, `ftp` oder z.B. `pop` erlaubt.

```
service/dns.domain.name@REALM

host/max.hznet.de@HZNET.DE
ftp/max.hznet.de@HZNET.DE
```

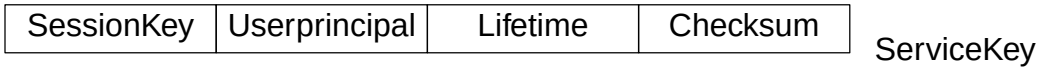
- Als "Geheimnis" wird ein Zufallswert verwendet.
- Das "Geheimnis" eines Serviceprincipals muß auf dem Server in der Datei `/etc/keytab` hinterlegt werden.
- Der Schlüssel eines Servers kann **nicht** geändert, sondern lediglich neu erzeugt werden.

KDC (Key Distribution Center)

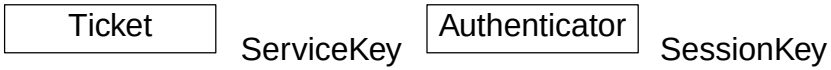
- KDC besteht aus *authentication Server* (AS) und *Ticket granting Server* (TGS)
- KDC muß besonders geschützt werden
- Enthält die Datenbank mit den User- und Serverprincipals der Realm
- Aus Redundanzgründen auch weitere KDC-Server (Slaves) möglich
- Die Slaves gleichen die Datenbank über ein Protokoll mit dem Master KDC ab.

Ticket

- Gegenseitige Authentisierung von Benutzer und Dienst
  Mutual Authentication
- Enthält SessionKey, UserPrincipal, und Lebensdauer (Gültigkeit)
  Das Ticket ist verschlüsselt mit dem Serviceschlüssel des Servers



- Wird vom Client beim KDC für einen bestimmten Dienst angefordert
- Wird in einem lokalen Ticketcache befristet gespeichert
- Wird vor der Authentisierung um Zeitstempel (*Authenticator*) ergänzt.
- Ticket plus Authenticator ist ein *Authentication credential*



Ticket Granting Ticket (TGT)

- TGT wird einmal beim Authentication Server (AS) angefordert
  Benutzer authentisiert sich durch Passwort
- TGT ist ein Service-Ticket für den Ticket Granting Server (TGS)
- Automatische Beantragung beim Anmelden oder manuell durch `kinit`
- TGT hat default Lifetime von 8 Stunden (änderbar)
  Wird im Ticketcache des Client gespeichert
- Für Authentisierung bei Anforderung von Servicetickets am TGT
  Der Benutzer muß sein Passwort nicht noch einmal eingeben
- Single-Sign-On
- Beim Abmelden wird der Ticketcache automatisch gelöscht.

Beispiele (Anmelden)

- Holen eines Ticket-Granting-Ticket

```
$ kinit hoz
Password for hoz@HZNET.DE:<password>
```

- Anzeige des Ticket Cache

```
$ klist -f
Ticket cache: FILE:/tmp/krb5cc_3157
Default principal: hoz@HZNET.DE

Valid starting     Expires            Service principal
12/20/01 12:48:29  12/20/01 22:48:29  krbtgt/HZNET.DE@HZNET.DE
        Flags: FI
```

Beispiele (telnet)

- Telnetverbindung zu einem Rechner aufbauen

```
$ telnet -a xt4.hznet.de
Trying 10.128.4.174...
Connected to xt4.hznet.de (10.128.4.174).
Escape character is '^]'.
[ Kerberos V5 accepts you as ''hoz@HZNET.DE'' ]
Welcome to SuSE Linux 7.3 (i386) - Kernel %r (%t).

*** Connection not encrypted! Communication may be eavesdropped.
hoz@hma:~> exit
```

- Jetzt das ganze mit verschlüsselter Datenkommunikation:

```
$ telnet -x -a xt4.hznet.de
Trying 10.128.4.174...
Connected to xt4.hznet.de (10.128.4.174).
Escape character is '^]'.
Waiting for encryption to be negotiated...
[ Kerberos V5 accepts you as ''hoz@HZNET.DE'' ]
done.
Welcome to SuSE Linux 7.3 (i386) - Kernel %r (%t).
hoz@hma:~> exit
```

Beispiele (Ticketcache)

- Anzeige des Ticketcache

```
$ klist -f
Ticket cache: FILE:/tmp/krb5cc_3157
Default principal: hoz@HZNET.DE

Valid starting     Expires            Service principal
12/20/01 12:48:29  12/20/01 22:48:29  krbtgt/HZNET.DE@HZNET.DE
        Flags: FI
12/20/01 12:54:27  12/20/01 22:48:29  host/xt4.hznet.de@HZNET.DE
        Flags: F
```

- Ticket cache leeren

```
$ kdestroy
$ klist -f
```

Voraussetzungen für den Einsatz von Kerberos

- Zeitsynchronisierte Systeme (max. 5 Minuten Differenz)
- Client- und Serversysteme müssen korrekt im DNS eingetragen sein
- KDC’s müssen speziell gesicherte Systeme sein
- Mindestens zwei KDC’s (Master, Slave).
- Ausschließlich Kerberos V verwenden
- Anforderungen für Cisco Router
  - IOS-Image mit Kerberos Support (Enterprise, IPsec oder Telco Feature Set)
  - DES oder 3DES notwendig für verschlüsselte Telnetverbindung (Authentisierung auch ohne DES-Image).
- Backupkonzept für REALM Datenbank!

CONTENTS