Keytab File

13 Jan 2015 - 11:26 | Version 21 | Oderholz | Domain

Dieses Topic beschreibt, was Keytabs (Keytabfiles) sind, wozu sie benötigt werden und wie sie **manuell** erstellt werden können.

Inhaltsverzeichnis:

- Einleitung
  - Kerberos
  - Principals, Tickets und Keytabs
- Keytabs (Keytabfiles)
- Schritt-Für-Schritt-Anleitungen mit Active Directory
  - Keytab für kerberisierten Dienst
  - Keytab für Batch-Job erstellen
  - Keytab unter Linux erstellen
  - Keytab überprüfen

Einleitung

Hier sollen nur kurz die wesentlichen Punkte aufgeführt werden, um ein minimales Hintergrundwissen für den Umgang mit Keytabfiles sicher zu stellen. Weitere Informationen gibt es unter anderem im Knowledge Forum: Kerberos and AFS

Kerberos

Kerberos bietet ein Verfahren zur **gegenseitigen Authentifizierung** durch einen **vertrauensvollen Dritten** (trusted third party mutual authentication) über eine **ungesicherte Verbindung**. Das heisst im Detail:

Gegenseitige Authentifizierung
: Es wird nicht nur der Client (aktiver Kommunikationspartner; stellt die Verbindungsanfrage) gegenüber dem (Anwendungs-)Server (passiver Kommunikationspartner; wartet auf eingehende Verbindungsanfragen) authentifiziert, sondern auch der Server gegenüber dem Client. Nach erfolgreichem Aufbau einer Kerberos-Session können beide Kommunikationspartner sicher sein, dass das jeweilige Gegenüber ist, wer/was es vorgibt zu sein.

Vertrauensvoller Dritte
: Eine dritte "Person", die an der Kommunikation nicht teil nimmt, der aber beide Kommunikationspartner (Client und Server) vertrauen. Bei Kerberos ist dies der "Kerberos-Server".

Ungesicherte Verbindung
: Für Kerberos ist es nicht erforderlich, dass die Kommunikationsverbindung zwischen Client, Anwendungs-Server und Kerberos-Server "abhörsicher" ist. Dies wird dadurch erreicht, dass die ausgetauschten Nachrichten jeweils selbst verschlüsselt werden.

Principals, Tickets und Keytabs

Ein Kerberos-Server benutzt kryptographische Schlüssel, um die Nachrichten an die Kommunikationspartner zu verschlüsseln. Diese muss er speichern. Damit der Kerberos-Server weiss, welcher Schlüssel zu welchem Kommunikationspartner gehört, bekommt jeder einen sogenannten "Principal" zugewiesen.

Die Nachrichten, die ausgetauscht werden, werden als "Tickets" bezeichnet. Es gibt zwei Arten von Tickets:

Service-Tickets
: Mit einem Service-Ticket kann ein Client sich gegenüber einem Anwendungs-Server ausweisen. Wenn das funktioniert, ist eine Kerberos-Session zwischen Client und Server hergestellt und beide sind gegenseitig authentifiziert. Ein Service-Ticket erhält der Client vom Kerberos-Server.

Ticket Granting Ticket (TGT)
: Damit ein Client vom Kerberos-Server Service-Tickets (s.o.) erhalten kann, muss er sich durch Benutzung seines kryptographischen Schlüssels "anmelden". (Der Schlüssel wird aus dem Passwort mathematisch abgeleitet.) Wenn sich der Client erfolgreich beim Kerberos-Server authentifiziert hat, erhält er daraufhin ein TGT. Mit diesem TGT kann er dann vom Kerberos-Server Service-Tickets beziehen. (Da der Client nur das TGT vorweisen muss und nicht erneut seinen Schlüssel verwenden muss, ermöglicht Kerberos damit "Single Sign-On" (SSO)).

Keytabs (Keytabfiles)

Keytabs (auch Keytabfiles genannt) dienen nur einem Zweck: Den kryptographischen **Schlüssel** eines Prinzipals für die **spätere Verwendung** abzuspeichern; es ist im Grunde gleichbedeutend, wie wenn Benutzername und Passwort abgespeichert würden. Dies wird immer dann benötigt, wenn eine **interaktive Eingabe des Passworts nicht möglich** ist.

Dafür gibt es 2 Anwendungsfälle:

1. Batch-Jobs: Batch-Jobs laufen ohne Benutzerinteraktion. Wenn ein solcher Job auf einen kerberisierten Dienst zugreifen muss (z.B. auf AFS zugreifen oder eine kerberisierte Webseite im Intranet aufrufen), muss ein Service-Ticket bezogen werden können. (Dazu bedarf es eines TGTs und dazu wiederum dem Schlüssel, der im Keytab gespeichert ist).
2. Kerberisierter Service: Damit ein Dienst (Prozess) ein Service-Ticket verstehen kann, muss er auf die (speziell für ihn) verschlüsselten Daten in dem Ticket zugreifen können. Dazu muss der Dienst auf seinen kryptographischen Schlüssel zugreifen können. (Da ein Dienst im Hintergrund läuft, kann aber kein Passwort eingegeben werden.)

Sicherheitshinweis: Da es sich bei Keytabs um **abgespeicherte Zugangsdaten** handelt, sollten die **Zugriffsrechte möglichst stark eingeschränkt** werden. (Unter Unix/Linux heisst das: `chown` auf User, unter dem der Daemon/Batch-Job läuft und anschliessend `chmod 400`.)

Schritt-Für-Schritt-Anleitungen mit Active Directory

Es müssen immer folgende Schritte erledigt werden:

1. Kerberos-Prinzipal und Schlüssel registrieren
2. Keytab generieren

Keytab für kerberisierten Dienst

1. Benötigte Informationen einholen
    - Wer ist der Verantwortliche/die Kontaktperson?
    - Welcher Dienst wird kerberisiert?
        - host: Login (z.B. ssh, telnet, rsh) Kleinschreibung ist wichtig!
        - HTTP: Webserver
        - CIFS: Windows-File-Server
    - Kurze Beschreibung der Aufgabe des Dienstes
    - Host-Name (DNS) **ACHTUNG:** Der Hostname muss auch für **DNS Reverse-Lookup** funktionieren

(daher z.B. wlb1.psi.ch und **nicht** intranet.psi.ch)

2. Benutzerkonto erstellen
    - OU: `/it/services/kerberos` (`ou=kerberos,ou=services,ou=it,dc=d,dc=psi,dc=ch`)
    - Namensschema für Benutzerkonto: `$hostname_$service` (z.B. `wlb1_http`) - bei Full Name eintragen
    - Beschreibung eintragen (inkl. Verantwortlichen / Kontaktperson)
    - Option "User cannot change password" setzen
    - Option "Password never expires" setzen
    - Keine Mailbox erstellen!
    - Nach Erstellung Description des Users anpassen (z. B. "Linux Server (Kapeller Rene)")
3. Prinzipal registrieren und Keytab erstellen
    - Kerberos-Prinzipal für Dienst (im Windows-Jargon "SPN" = Service Principal Name) ableiten nach Schema: `$service/$hostname@D.PSI.CH`
    - `ktpass.exe` aufrufen mit entsprechenden Parametern
        - `-princ ...`: Angabe des Kerberos-Prinzipals
        - `-mapuser ...`: Angabe des Benutzerkontos
        - `+rndPass +setPass`: Zufallspasswort erzeugen und zuweisen
        - `-ptype KRB5_NT_PRINCIPAL`: Typ des Kerberos-Prinzipals
        - `-out ...`: Pfad zur Datei (Keytab), die erstellt werden soll
4. Keytab auf sicherem Übertragungsweg zustellen
    - In Praxis ausreichend: Verschlüsseltes ZIP-File erstellen und per Email verschicken, aber Passwort nicht in Email mitschicken.
5. Verantwortlichen / Kontaktperson über Vertraulichkeit des Keytabs informieren

Beispiel:

```
C:\Temp> ktpass.exe -princ HTTP/myserver.psi.ch@D.PSI.CH -mapuser PSICH\myserver_http -out
C:\Temp\myserver_http.keytab +rndPass +setPass -ptype KRB5_NT_PRINCIPAL
```

Mit bekanntem Passwort:

```
ktpass.exe -princ HTTP/ispd03.psi.ch@D.PSI.CH -mapuser PSICH\svcusr-webhosting -out
C:\Temp\ispd03_http.keytab -ptype KRB5_NT_PRINCIPAL /pass *
Enter the  password: ...
```

Keytab für Batch-Job erstellen

1. Benötigte Informationen einholen
    - Wer ist der Verantwortliche/die Kontaktperson?
    - Kurze Beschreibung der Aufgabe des Batch-Jobs
2. Benutzerkonto erstellen (Service-User)
    - OU: `/psi/users/nonpersons` (`ou=nonpersons,ou=users,ou=psi,dc=d,dc=psi,dc=ch`)
    - Namensschema für Benutzerkonto: `svcusr-...` (z.B. `svcusr-sinq_cronjobs`)
    - Beschreibung eintragen (inkl. Verantwortlichen / Kontaktperson)
    - Option "User cannot change password" setzen
    - Option "Password never expires" setzen
3. Prinzipal registrieren und Keytab erstellen
    - Kerberos-Prinzipal für normale Benutzerkonten ist automatisch im AD registriert
    - `ktpass.exe` aufrufen mit entsprechenden Parametern

        - `-princ ...`: Angabe des Kerberos-Prinzipals
        - `-mapuser ...`: Angabe des Benutzerkontos (kann weggelassen werden, aber dann gibt `ktpass.exe` eine **Warnung** aus. Diese **darf ignoriert werden**. Passwort-Setzen (s.u.) kann dann aber nicht integriert werden.)
        - `+rndPass +setPass`: Zufallspasswort erzeugen und zuweisen (kann weggelassen werden, wenn das Passwort bekannt ist/sein muss)
        - `-ptype KRB5_NT_PRINCIPAL`: Typ des Kerberos-Prinzipals
        - `-out ...`: Pfad zur Datei (Keytab), die erstellt werden soll
4. Keytab auf sicherem Übertragungsweg zustellen
    - In Praxis ausreichend: Verschlüsseltes ZIP-File erstellen und per Email verschicken, aber Passwort nicht in Email mitschicken.
5. Verantwortlichen / Kontaktperson über Vertraulichkeit des Keytabs informieren

Beispiel: Mit Zufallspasswort

```
C:\Temp> ktpass.exe -princ svcusr-my_batch_job@D.PSI.CH -mapuser PSICH\svcusr-my_batch_job -
out C:\Temp\svcusr-my_batch_job.keytab +rndPass +setPass -ptype KRB5_NT_PRINCIPAL
```

Beispiel: Mit bekanntem Passwort

```
C:\Temp> ktpass.exe -princ svcusr-my_batch_job@D.PSI.CH -ptype KRB5_NT_PRINCIPAL -out
C:\Temp\svcusr-my_batch_job.keytab -pass *
NOTE: creating a keytab but not mapping principal to any user.
For the account to work within a Windows domain, the
principal must be mapped to an account, either at the
domain level (with /mapuser) or locally (using ksetup)
If you intend to map svcusr-my_batch_job@D.PSI.CH to an account through other means
or don't need to map the user, this message can safely be ignored.
Type the password for svcusr-my_batch_job:
Type the password again to confirm:
WARNING: pType and account type do not match. This might cause problems.
Key created.
Output keytab to svcusr-my_batch_job.keytab:
Keytab version: 0x502
keysize 47 svcusr-my_batch_job@D.PSI.CH ptype 1 (KRB5_NT_PRINCIPAL) vno 1 etype 0x17 (RC4-
HMAC) keylength 16 (0xae4ca1304b999690711715010d232ebc)
```

Keytab unter Linux erstellen

Damit ein Keytabfile unter Linux erstellt werden kann, muss

- Das Benutzerkonto im AD bereits existieren
- Der Prinzipal bereits für das Benutzerkonto registriert sein
- Das Passwort des Benutzerkontos bekannt sein

```
[lauk@llc2 ~]$ ktutil
ktutil:  add_entry -password -p svcusr-my_batch_job@D.PSI.CH -k 1 -e RC4-HMAC
Password for svcusr-my_batch_job@D.PSI.CH:
ktutil:  write_kt svcusr-my_batch_job.keytab
ktutil:  quit
```

Keytab überprüfen

Ein Keytab kann überprüft werden, indem versucht wird, vom Kerberos-Server ein TGT für einen Prinzipal zu erhalten. (Dazu sollte möglichst ein temporärer Credential-Cache verwendet werden.)

```
[lauk@llc2 ~]$ kinit -k -t svcusr-my_batch_job.keytab -c /tmp/test_cc svcusr-
my_batch_job@D.PSI.CH
[lauk@llc2 ~]$ klist -c /tmp/test_cc
Ticket cache: FILE:/tmp/test_cc
Default principal: svcusr-my_batch_job@D.PSI.CH

Valid starting     Expires            Service principal
05/12/14 12:53:57  05/12/14 22:53:57  krbtgt/D.PSI.CH@D.PSI.CH
        renew until 05/19/14 12:53:57
[lauk@llc2 ~]$ kdestroy -c /tmp/test_cc
[lauk@llc2 ~]$ rm /tmp/test_cc       # double check that credential cache was deleted
rm: cannot remove `/tmp/test_cc': No such file or directory
```

Revision 21, 13 Jan 2015 11:26:29