Kerberos und NFSv4

Alexander Kaiser

AG Technische Informatik

27. November 2012

Übersicht

Geschichte

Kerberos

- verteilter Authentifizierungsdienst für offene und unsichere Netzwerke
- ermöglicht *Single Sign-on*
- entwickelt von Steve Miller und Clifford Neuman am MIT
- basiert auf dem Needham-Schroeder-Protokoll
  - Schlüsselaustausch und Authentifikation
  - Grundlage: symmetrisches Kryptosystem
- Teil des Anthena-Projekts (u.a. X Windows System)
- erster Einsatz am MIT 1983
- aktuelle Version: Kerberos V
- offener Standard (RFC 4120)
- populäre Implementationen: MIT Kerberos, Heimdal (KTH)

Geschichte

NFS

- verteiltes Netzwerk-Dateisystem
- Einsatzgebiete: Thin Clients, verteilte `$HOME`s
- entwickelt von Sun Microsystems (1984)
- Vorteil gegenüber bspw. FTP: Transparenz
- offener Standard seit Version 2 (1989, RFC 1094)
- aktuelle Version: NFSv4 (RFC 3530) [NFSv4.1 (RFC 5661)]
- Implementation meist Teil des BS

Übersicht

Grundlagen

Terminologie

Realm Kerberos-Domäne, meist in Großbuchstaben (z.B. FOO.BAR.ORG)

Principal Name eines Kerberos-Nutzers (Person, Host oder Service) (z.B. juser@FOO.BAR.ORG, host/somehost@FOO.BAR.ORG)

Ticket Instanz einer Sitzung, wird nach erfolgreicher Anmeldung in einem *Ticket Cache* (meist einer Datei) abgelegt

Key Distribution Center (KDC) Server, der für die Ausstellung von *Tickets* in seinem *Realm* verantwortlich ist

Realms & Principals

Realms

- typischerweise Name der DNS-Domäne
- Zuordnung zwischen Realms und DNS-Domänen festlegbar

Principals

Benutzer	UNIX-Benutzername
Clients	host/<*hostname*>
Services	<*service*>/<*hostname*>

Bemerkung: <*hostname*> ist immer der FQDN des Clients! (setzt DNS voraus)

Kerberos-Tickets

Allgemeines

TGT *Ticket Granting Ticket*, initiales Ticket; wird typischerweise beim Login ausgestellt

Service-Tickets werden bei Service-Benutzung gezogen

Ticket-Cache Ablage für Tickets, benutzerspezifisch

Gültigkeit

- Tickets haben eine zeitlich begrenzte Gültigkeit
- Verlängerung von Tickets möglich, solange sie noch Gültig sind (ohne Passworteingabe)
- neues TGT beantragen durch Passworteingabe

Kerberos-KDC

Allgemeines

- Kommunikation erfolgt über TCP-Port 88
- sollte auf einem besonders abgesicherten Rechner laufen

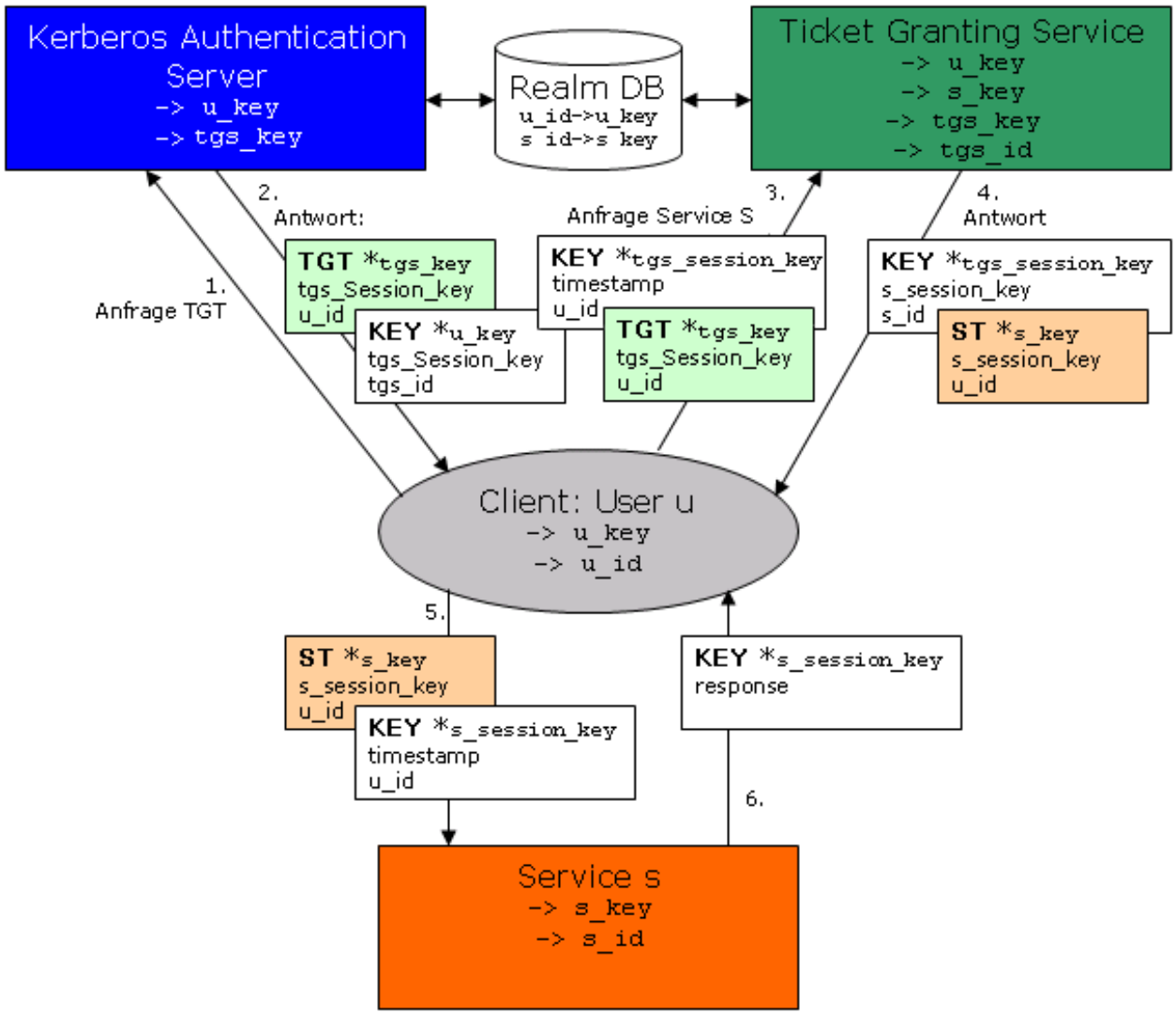
Bestandteile

Realm Database enthält Principal-Keys

Authentication Server stellt TGTs aus

Ticket Granting Service stellt weitere Service-Tickets aus

Kerberos-Authentifikation (Ablauf)



(Quelle: Wikipedia)

Beispiel: LDAP-Query I

```
$ klist
Ticket cache: FILE:/tmp/krb5cc_13478_WvlRv73546
Default principal: juser@FOO.BAR.ORG

Valid starting    Expires           Service principal
25/11/2012 16:37  26/11/2012 16:37  krbtgt/FOO.BAR.ORG@FOO.BAR.
        renew until 09/12/2012 16:37
$ ldapwhoami
SASL/GSSAPI authentication started
SASL username: juser@FOO.BAR.ORG
SASL SSF: 56
SASL data security layer installed.
dn:uid=juser,ou=people,dc=foo,dc=bar,dc=org
```

Beispiel: LDAP-Query II

```
$ klist
Ticket cache: FILE:/tmp/krb5cc_13478_WvlRv73546
Default principal: juser@FOO.BAR.ORG

Valid starting    Expires           Service principal
25/11/2012 16:37  26/11/2012 16:37  krbtgt/FOO.BAR.ORG@FOO.BAR.
        renew until 09/12/2012 16:37
25/11/2012 18:58  26/11/2012 16:37  ldap/ldap1.foo.bar.org@FOO.
        renew until 25/11/2012 18:58
```

Zusammenfassung (Kerberos) I

Features

- Verschlüsselung des Datenverkehrs
- Aushandlung des Verfahrens
  - Verschlüsselungsverfahren (MIT): DES, 3DES, AES und RC4
  - Hash-Verfahren (MIT): MD5, SHA-1, HMAC und CRC32
- Delegation von Credentials (*Ticket Forwarding*)

Nachteile

- KDC $\rightarrow$ *Single Point of Failure*
- strikte Zeit-Vorgaben: Uhren aller beteiligten Rechner müssen synchron sein (nicht mehr als 5 Min. Abweichung)
- Admin-Protokoll nicht Teil des Standards
- Service-Principals gebunden an Host-Namen

Zusammenfassung (Kerberos) II

Linux-Unterstützung

`pam_krb5` PAM-Modul für Kerberos

GSSAPI *General Security Service API* (Abstraktion)

SASL-Plugin Kerberos-Authentifikation über GSSAPI

`krb5-auth-dialog` grafisches Widget, welches über die Gültigkeit von Tickets informiert

"kerberisierte" Dienste

- SSH (OpenSSH)
- LDAP (OpenLDAP via SASL)
- IMAP (Dovecot)
- SMTP (Sendmail)
- HTTP (apache)
- AFS
- NFSv4

Übersicht

Nachteile von NFSv3

- Authentifizierung auf Client-Ebene
  $\rightarrow$ ist nur so sicher wie das Netzwerk
- basiert auf UNIX-UIDs / GIDs
  $\rightarrow$ müssen auf Server und Clients übereinstimmen
- Mount-Protokoll (Abbildung von Pfaden auf File-Handles) und Lock-Protokoll sind nicht Teil des Standards
- Aushandlung von Ports über den *Portmapper*
  Probleme bei Firewalls

Neuerungen in NFSv4

- Mount- und Lock-Protokoll Teil des Standards
  $\rightarrow$ NFSv4 nicht zustandslos!
- Transport über TCP, standardmäßig Port 2049
- starke Sicherheit durch Kerboros-Integration (RPCSEC_GSS)
- *Pseudo File System*
- Abstraktion von UIDs / GIDs via *ID-Mapper*
- ACLs sind Teil des Protokolls
- Pfadnamen UTF-8 kodiert

Beispiel: Pseudo File System

/etc/exports

```
/export/        myclient(rw,sync,no_subtree_check,fsid=0)
/export/foo     myclient(rw,sync,no_subtree_check)
```

Pseudo FS

```
          /   (/export)
          |
          |
       /foo (/export/foo)
```

Kerberos-Integration

Sicherheitsstufen

Authentication Benutzerauthentifizierung durch Kerberos [sec=krb5]

Integrity zusätzliche Berechnung eines MAC (*Message Authentication Code*) [sec=krb5i]

Privacy zusätzliche Verschlüsselung der gesamten NFS-Nachricht (exklusive Header) [sec=krb5p]

Beispiel-Rechnung

sec=Wert	Durchsatz [MB/s]	Verlust	CPU-Auslastung
sys	5.40	–	69%
krb5	5.26	2.6%	70%
krb5i	4.44	17.7%	77%
krb5p	1.45	73.1%	99%

Zusammenfassung (NFSv4)

Features

- sichere Authentifizierung
- Transport erfolgt über ein TCP-Port
  $\rightarrow$ Firewalls, SSH-Tunnel
- native ACLs

Nachteile

- keine automatische Aushandlung der Sicherheitsstufe [Linux]
- schlechte Performance bei krb5i und krb5p [Linux]
- momentan nur DES- und CRC32-Unterstützung [Linux]
- ältere Kernel (2.6.18): Client-seitige Hänger bei abgelaufenen Tickets [Linux]
- keine lokalen bind-Mounts durch Autofs möglich

Übersicht

- NFSv4.1
  - *Session Key*
  - Load-Balancing durch pNFS (*parallel NFS*)
  - Kompatibilität zu Windows-ACLs
- SSSD (*System Security Services Daemon*)
  - vereinheitlichter Zugriff auf Indetitäts- und Authentifizierungsdienste
  - Back-Ends für Kerberos, LDAP, etc.
  - Integration in Active Directory
  - ermöglicht Offline-Authentifizierung

Fragen?