Sichere Authentizierung mit Kerberos 5

Andreas Hirczy

TU Graz
Institut für Theoretische Physik – Computational Physics

Grazer Linuxtage 2007
19. Mai. 2007

Vorstellung

Meine Installation umfasst das **Institut für Theoretische Physik – Computational Physics** an der TU Graz und einen **Computerlehr- und Arbeitsraum** für Mathematik- und Physik-Studenten.

Wir betreiben etwa 80 PCs und 10 Notebooks unter Linux und haben zur Zeit 800 Useraccounts, von denen im Regelfall höchstens 50 gleichzeitig aktiv sind.

Vorstellung

Meine Installation umfasst das **Institut für Theoretische Physik – Computational Physics** an der TU Graz und einen **Computerlehr- und Arbeitsraum** für Mathematik- und Physik-Studenten.

Wir betreiben etwa 80 PCs und 10 Notebooks unter Linux und haben zur Zeit 800 Useraccounts, von denen im Regelfall höchstens 50 gleichzeitig aktiv sind.

Der Arbeitsschwerpunkt liegt bei der mathematischen Behandlung von Fragestellungen vor allem im Bereich der **Vielteilchenphysik** und der **Plasmaphysik** – dabei kommen sowohl analytische als auch numerische Verfahren zum Einsatz.

Vorstellung

Meine Installation umfasst das **Institut für Theoretische Physik – Computational Physics** an der TU Graz und einen **Computerlehr- und Arbeitsraum** für Mathematik- und Physik-Studenten.

Wir betreiben etwa 80 PCs und 10 Notebooks unter Linux und haben zur Zeit 800 Useraccounts, von denen im Regelfall höchstens 50 gleichzeitig aktiv sind.

Der Arbeitsschwerpunkt liegt bei der mathematischen Behandlung von Fragestellungen vor allem im Bereich der **Vielteilchenphysik** und der **Plasmaphysik** – dabei kommen sowohl analytische als auch numerische Verfahren zum Einsatz.

Sicherheit ist wichtig – aus purem Egoismus!

Wer liefert Kerberos? Geschichte?

Die Refenzimplementierung stammt aus dem Projekt Athena vom MIT. Im Zuge der Ablösung der wenigen Mainframes durch viele Workstations, deren Besitzern/Benutzern weitgehende Freiheiten auf Ihren Maschinen zugestanden werden sollte und die dezentral gewartet wurden, war es notwendig, ein verteiltes System zur Zugangskontrolle zu implementieren.

Wer liefert Kerberos? Geschichte?

Die Refenzimplementierung stammt aus dem Projekt Athena vom MIT. Im Zuge der Ablösung der wenigen Mainframes durch viele Workstations, deren Besitzern/Benutzern weitgehende Freiheiten auf Ihren Maschinen zugestanden werden sollte und die dezentral gewartet wurden, war es notwendig, ein verteiltes System zur Zugangskontrolle zu implementieren.

Nach einigen internen Prototypen wurde 1989 Kerberos 4 vorgestellt und später (2000?) durch Kerberos 5 abgelöst.

Wer liefert Kerberos? Geschichte?

Die Refenzimplementierung stammt aus dem Projekt Athena vom MIT. Im Zuge der Ablösung der wenigen Mainframes durch viele Workstations, deren Besitzern/Benutzern weitgehende Freiheiten auf Ihren Maschinen zugestanden werden sollte und die dezentral gewartet wurden, war es notwendig, ein verteiltes System zur Zugangskontrolle zu implementieren.

Nach einigen internen Prototypen wurde 1989 Kerberos 4 vorgestellt und später (2000?) durch Kerberos 5 abgelöst.

Aus dem MIT-Code wurden vor allem durch die Exportbeschränkungen der USA für kryptografische Produkte die Bildung einige Derivate (Bones, eBones und vor allem Heimdal) angeregt. Heute ist Kerberos 5 darüberhinaus in Routern von Cisco und den Betriebsystemen von Sun, Apple und Microsoft bereits enthalten.

Die 3 großen As

Im wesentlichen erwartet man sich drei Funktionen bei der Verwaltung von Benutzern und Services in einem Netzwerk:

- **Authentizierung** - Wer meldet sich an?
- **Authorisierung** - Was darf er?
- **Audit** - Wir schreiben auf, wer bei der Türe reinkommt!

Was kann Kerberos?

- Identität eines *principals* bestätigen

  Benutzer: ahi@ITP.TUGRAZ.AT

  Host: host/faeppc35.tu-graz.ac.at@ITP.TUGRAZ.AT

  Service: afs/itp.tugraz.at@ITP.TUGRAZ.AT

Was kann Kerberos?

- Identität eines *principals* bestätigen

  Benutzer: ahi@ITP.TUGRAZ.AT

  Host: host/faeppc35.tu-graz.ac.at@ITP.TUGRAZ.AT

  Service: afs/itp.tugraz.at@ITP.TUGRAZ.AT

- kein unverschlüsseltes Passwort wird übers Netzwerk gesendet

Was kann Kerberos?

- Identität eines *principals* bestätigen

  Benutzer: ahi@ITP.TUGRAZ.AT

  Host: host/faeppc35.tu-graz.ac.at@ITP.TUGRAZ.AT

  Service: afs/itp.tugraz.at@ITP.TUGRAZ.AT
- kein unverschlüsseltes Passwort wird übers Netzwerk gesendet
- nur einer Instanz muß Vetrauen entgegengebracht werden - dem *Key Distribution Center* (KDC)
- gegenseitige Authentizierung zwischen Client und Server

Was kann Kerberos?

- Identität eines *principals* bestätigen

  Benutzer: ahi@ITP.TUGRAZ.AT
  Host: host/faeppc35.tu-graz.ac.at@ITP.TUGRAZ.AT
  Service: afs/itp.tugraz.at@ITP.TUGRAZ.AT

- kein unverschlüsseltes Passwort wird übers Netzwerk gesendet
- nur einer Instanz muß Vetrauen entgegengebracht werden - dem *Key Distribution Center* (KDC)
- gegenseitige Authentizierung zwischen Client und Server
- *Single Sign On* (SSO)

Was kann Kerberos nicht?

- Kerberos bietet **nur Authentizierung** – Authorisierung und Logging müssen von anderen Systemen (PAM, . . .) übernommen werden.
- Kerberos ist **kein Verzeichnissystem** wie z.B. NIS, LDAP oder Active Directory.

Begriffe

principal wird von Kerberos als eine Einheit gesehen und authentiziert. Der *principal* setzt sich aus mehreren Komponenten und dem *realm* zusammen - eine Komponente ist typisch für Benutzer (ahi@ITP.TUGRAZ.AT), zwei Komponenten werden für Service- und Host-Tickets verwendet (host/faeppc35.tu-graz.ac.at@ITP.TUGRAZ.AT), mehr als zwei Komponenten sind möglich, aber nicht üblich.

Begriffe

principal wird von Kerberos als eine Einheit gesehen und authentiziert. Der *principal* setzt sich aus mehreren Komponenten und dem *realm* zusammen - eine Komponente ist typisch für Benutzer (ahi@ITP.TUGRAZ.AT), zwei Komponenten werden für Service- und Host-Tickets verwendet (host/faeppc35.tu-graz.ac.at@ITP.TUGRAZ.AT), mehr als zwei Komponenten sind möglich, aber nicht üblich.

realm die administrative Organisationseinheit

Begriffe

principal wird von Kerberos als eine Einheit gesehen und authentiziert. Der *principal* setzt sich aus mehreren Komponenten und dem *realm* zusammen - eine Komponente ist typisch für Benutzer (ahi@ITP.TUGRAZ.AT), zwei Komponenten werden für Service- und Host-Tickets verwendet (host/faeppc35.tu-graz.ac.at@ITP.TUGRAZ.AT), mehr als zwei Komponenten sind möglich, aber nicht üblich.

realm die administrative Organisationseinheit

ticket der virtuelle Fahrschein

Begriffe

principal wird von Kerberos als eine Einheit gesehen und authentiziert. Der *principal* setzt sich aus mehreren Komponenten und dem *realm* zusammen - eine Komponente ist typisch für Benutzer (ahi@ITP.TUGRAZ.AT), zwei Komponenten werden für Service- und Host-Tickets verwendet (host/faeppc35.tu-graz.ac.at@ITP.TUGRAZ.AT), mehr als zwei Komponenten sind möglich, aber nicht üblich.

realm die administrative Organisationseinheit

ticket der virtuelle Fahrschein

Ticket Granting Ticket (TGT) ist ein *service ticket*, das Zugang zum TGS gibt, damit können ohne Passwort andere Service-Tickets erworben werden.

Begriffe

principal wird von Kerberos als eine Einheit gesehen und authentiziert. Der *principal* setzt sich aus mehreren Komponenten und dem *realm* zusammen - eine Komponente ist typisch für Benutzer (ahi@ITP.TUGRAZ.AT), zwei Komponenten werden für Service- und Host-Tickets verwendet (host/faeppc35.tu-graz.ac.at@ITP.TUGRAZ.AT), mehr als zwei Komponenten sind möglich, aber nicht üblich.

realm die administrative Organisationseinheit

ticket der virtuelle Fahrschein

Ticket Granting Ticket (TGT) ist ein *service ticket*, das Zugang zum TGS gibt, damit können ohne Passwort andere Service-Tickets erworben werden.

Authentication Server (AS) vergibt das Ticket Granting Ticket (TGT)

Ticket Granting Server (TGS) vergibt Tickets mit Hilfe des TGT

Key Distribution Center (KDC) AS + TGS

Konfiguration der Server

Es ist wesentlich, das KDC besonders zu schützen, weil eine Kompromittierung dieser Geräte alle Accounts verdächtig macht – und 800 Benutzern neue Passworte zu geben ist eine reichlich schmerzhafte Angelegenheit.

Konfiguration der Server

Es ist wesentlich, das KDC besonders zu schützen, weil eine Kompromittierung dieser Geräte alle Accounts verdächtig macht – und 800 Benutzern neue Passworte zu geben ist eine reichlich schmerzhafte Angelegenheit. Zur guten Praxis der Server-Absicherung zählen jedenfalls:

- vermeiden unnützer Services
- OpenBSD oder Hardened Linux (SE Linux, Bastille Linux, grsecurity, . . .)

Konfiguration der Server

Es ist wesentlich, das KDC besonders zu schützen, weil eine Kompromittierung dieser Geräte alle Accounts verdächtig macht – und 800 Benutzern neue Passworte zu geben ist eine reichlich schmerzhafte Angelegenheit. Zur guten Praxis der Server-Absicherung zählen jedenfalls:

- vermeiden unnützer Services
- OpenBSD oder Hardened Linux (SE Linux, Bastille Linux, grsecurity, . . .)

Es ist möglich, das KDC redundant auszuführen; die Informationen werden periodisch auf den jeweils nachgeordneten Server kopiert. Für die Authentizierung funktioniert Failover sofort; für administrative Aufgaben muß der zuständige Server durch andere Maßnahmen geschaltet werden.

Konfiguration der Server

Es ist wesentlich, das KDC besonders zu schützen, weil eine Kompromittierung dieser Geräte alle Accounts verdächtig macht – und 800 Benutzern neue Passworte zu geben ist eine reichlich schmerzhafte Angelegenheit. Zur guten Praxis der Server-Absicherung zählen jedenfalls:

- vermeiden unnützer Services
- OpenBSD oder Hardened Linux (SE Linux, Bastille Linux, grsecurity, ...)

Es ist möglich, das KDC redundant auszuführen; die Informationen werden periodisch auf den jeweils nachgeordneten Server kopiert. Für die Authentizierung funktioniert Failover sofort; für administrative Aufgaben muß der zuständige Server durch andere Maßnahmen geschaltet werden.

Der Bedarf an Rechenleistung ist für kleiner Zellen sehr gering - für uns reicht 2 $\times$ Mini-ITX mit VIA EPIA-VE500 (533 MHz) mit CompactFlash statt Festplatte unter OpenBSD völlig: Load ist beinahe immer unter 0,2.

Konfiguration der Server (2)

kdc.conf

```
[kdcdefaults]

[realms]
ITP.TUGRAZ.AT = {
    master_key_type = des-cbc-crc
    supported_enctypes = aes256-cts-hmac-sha1-96:normal arcfour-hmac-md5:normal des
    max_life = 7d
    max_renewable_life = 31d
    dict_file = /usr/share/dict/words
}
```

Konfiguration der Clients

/etc/krb5.conf

```
[libdefaults]
   debug = true
   default_realm = ITP.TUGRAZ.AT
   default_tkt_enctypes = aes256-cts-hmac-sha1-96 des3-cbc-sha1 des-cbc-crc
   default_tgs_enctypes = aes256-cts-hmac-sha1-96 des3-cbc-sha1 des-cbc-crc

   afs_cells = itp.tugraz.at
```

Konfiguration der Clients (2)

/etc/krb5.conf (...)

```
[appdefaults]
   forwardable      = true
   forward          = true
   renewable        = true
   encrypt          = true
   krb4_get_tickets = false
   krb4_convert     = false
   krb5_get_tickets = true
   ticket_lifetime  = 86400
   renew_lifetime   = 2678400

   pam = {
      max_timeout = 2
      timeout_shift = 2
      initial_timeout = 1
      afs_cells = itp.tugraz.at
   }
```

Konfiguration der Clients (3)

/etc/krb5.conf (...)

```
[realms]
   ITP.TUGRAZ.AT = {
      kdc = faepkrb1.tu-graz.ac.at
      kdc = faepkrb2.tu-graz.ac.at
      admin_server = faepkrb1.tu-graz.ac.at:749
      default_domain = tu-graz.ac.at
   }

[domain_realm]
   tu-graz.ac.at = ITP.TUGRAZ.AT
  .tu-graz.ac.at = ITP.TUGRAZ.AT
   tugraz.at = ITP.TUGRAZ.AT
  .tugraz.at = ITP.TUGRAZ.AT

[logging]
   kdc = FILE:/var/log/krb5kdc.log
   admin_server = FILE:/var/log/kadmin.log
   default = FILE:/var/log/krb5lib.log
```

Erzeugen des Ticket Granting Ticket (TGT)

Mit dem Programm `kinit` wird eine Anfrage an den KDC/AS geschickt, um ein TGT zu erhalten. Der KDC baut ein Ticket (Client-Adresse, zufälliger *session key*, Zeitstempel, ...), verschlüsselt einen Teil dieser Informationen symmetrisch mit dem Password K_c des Benutzers und sendet sie zurück an `kinit`.

Erzeugen des Ticket Granting Ticket (TGT)

Mit dem Programm `kinit` wird eine Anfrage an den KDC/AS geschickt, um ein TGT zu erhalten. Der KDC baut ein Ticket (Client-Adresse, zufälliger *session key*, Zeitstempel, ...), verschlüsselt einen Teil dieser Informationen symmetrisch mit dem Password K_c des Benutzers und sendet sie zurück an `kinit`.

`kinit` fragt den Benutzer nach dem Passwort (oder liest es aus einer Keytab-Datei), entschlüsselt damit das Ticket und erhält den *session key* $K_{c,tgs}$, der für die Verschlüsselung der weiteren Kommunikation mit dem KDC dient. Das Passwort K_c wird nun nicht mehr gebraucht; der Speicherbereich kann vom Client überschrieben werden.

Erzeugen des Ticket Granting Ticket (TGT)

Mit dem Programm `kinit` wird eine Anfrage an den KDC/AS geschickt, um ein TGT zu erhalten. Der KDC baut ein Ticket (Client-Adresse, zufälliger *session key*, Zeitstempel, ...), verschlüsselt einen Teil dieser Informationen symmetrisch mit dem Password K_c des Benutzers und sendet sie zurück an `kinit`.

`kinit` fragt den Benutzer nach dem Passwort (oder liest es aus einer Keytab-Datei), entschlüsselt damit das Ticket und erhält den *session key* $K_{c,tgs}$, der für die Verschlüsselung der weiteren Kommunikation mit dem KDC dient. Das Passwort K_c wird nun nicht mehr gebraucht; der Speicherbereich kann vom Client überschrieben werden.

Das TGT hat eine festgelegte Gültigkeit von meist 10 Stunden; ist aber in einem gewissen Rahmen verlängerbar.

Verbindung mit einem Service

Ein Programm mit Kerberosunterstützung soll Verbindung zu einem anderen Computer (z.B. `itp.tugraz.at` aka. `faepsv.tu-graz.ac.at`) aufnehmen, der ein Service (z.B. WWW) anbietet. Der Client stellt fest, daß dazu ein Service-Key `HTTP/faepsv.tu-graz.ac.at@ITP.TUGRAZ.AT` notwendig ist, der noch nicht vorliegt.

Verbindung mit einem Service

Ein Programm mit Kerberosunterstützung soll Verbindung zu einem anderen Computer (z.B. `itp.tugraz.at` aka. `faepsv.tu-graz.ac.at`) aufnehmen, der ein Service (z.B. WWW) anbietet. Der Client stellt fest, daß dazu ein Service-Key `HTTP/faepsv.tu-graz.ac.at@ITP.TUGRAZ.AT` notwendig ist, der noch nicht vorliegt.

Der Client schickt eine Anfrage zum KDC/TGS und erhält ein neues Ticket mit einem weiteren *session key* $K_{c,s}$, das mit dem *session key* $K_{c,tgs}$ aus dem TGT verschlüsselt ist.

Verbindung mit einem Service (2)

Ein zweites Ticket, das mit dem passenden *service key* K_s verschlüsselt wurde, wird ebenfalls zum Client gesendet. Der Client bildet einen *authenticator* (Zeitstempel, verschlüsselt mit dem neuen *session key* $K_{c,s}$), der gemeinsam mit dem zweiten Ticket an den (Applikations)-Server geschickt wird.

Verbindung mit einem Service (2)

Ein zweites Ticket, das mit dem passenden *service key* K_s verschlüsselt wurde, wird ebenfalls zum Client gesendet. Der Client bildet einen *authenticator* (Zeitstempel, verschlüsselt mit dem neuen *session key* $K_{c,s}$), der gemeinsam mit dem zweiten Ticket an den (Applikations)-Server geschickt wird.

Am Server wird das Ticket mit K_s ausgepackt; mit dem darin enthaltenen $K_{c,s}$ kann der *authenticator* entpackt werden.

Verbindung mit einem Service (2)

Ein zweites Ticket, das mit dem passenden *service key* K_s verschlüsselt wurde, wird ebenfalls zum Client gesendet. Der Client bildet einen *authenticator* (Zeitstempel, verschlüsselt mit dem neuen *session key* $K_{c,s}$), der gemeinsam mit dem zweiten Ticket an den (Applikations)-Server geschickt wird.

Am Server wird das Ticket mit K_s ausgepackt; mit dem darin enthaltenen $K_{c,s}$ kann der *authenticator* entpackt werden.

Die erfolgreiche Entschlüsselung des *authenticators* beweist Client und Server die Authentizität des jeweiligen Kommunikationspartners.

Benutzerkommandos

kinit	erzeugt ein TGT Ticket
klist	zeigt vorhandene Tickets im Cache an
kpasswd	ändert das Passwort am KDC
kdestroy	löscht den Cache mit den Tickts

Die Kommunikation mit dem KDC funktioniert über die verschiedenen Implementierungen hinweg.

Anatomie eines Ticket: klist -f

```
klist: You have no tickets cached
Ticket cache: FILE:/tmp/krb5cc_997_umFh44
Default principal: ahi@ITP.TUGRAZ.AT

Valid starting     Expires            Service principal
05/18/07 14:42:05  05/19/07 14:42:05  krbtgt/ITP.TUGRAZ.AT@ITP.TUGRAZ.AT
renew until 05/25/07 14:42:05, Flags: FRIA
05/18/07 14:42:05  05/19/07 14:42:05  afs/itp.tugraz.at@ITP.TUGRAZ.AT
renew until 05/25/07 14:42:05, Flags: FRAT
05/18/07 15:50:04  05/19/07 14:42:05  host/faeppc35.tu-graz.ac.at@ITP.TUGRAZ.AT
renew until 05/25/07 14:42:05, Flags: FRAT
05/18/07 16:09:27  05/19/07 14:42:05  HTTP/faepsv.tu-graz.ac.at@ITP.TUGRAZ.AT
renew until 05/25/07 14:42:05, Flags: FRAT


Kerberos 4 ticket cache: /tmp/tkt997
```

Kommando für Administratoren

Während die vorhandenen Implementierungen aus Benutzersicht kompatibel sind, gilt dies für die Administrations-Schnittstelle nicht.

Mit `kadmin` (und dem richtigen Ticket) kann man von den Clients aus das Verhalten der Server steuern; `kadmin.local` bietet ein Abkürzung, wenn man mit Root-Rechten am KDC arbeitet. Die wichtigsten Möglichkeiten sind:

- anzeige verfügbarer *principals*
- erzeugen, löschen und modifizieren von *principals*
- ändern von Passworten
- vorgeben von *policies*
- erzeugen und löschen von *keytabs*

Kerberos am Mac

- Kerberos wird bei MacOS X mitgeliefert
- Konfiguration wie in Linux/Unix, allerdings in der Datei `/Library/Preferences/edu.mit.Kerberos`; in neueren Versionen angeblich auch in `/etc/krb5.conf`.
- GUI: `/System/Library/CoreServices/Kerberos`

Kerberos am Mac

- Kerberos wird bei MacOS X mitgeliefert
- Konfiguration wie in Linux/Unix, allerdings in der Datei `/Library/Preferences/edu.mit.Kerberos`; in neueren Versionen angeblich auch in `/etc/krb5.conf`.
- GUI: `/System/Library/CoreServices/Kerberos`
- Login mit Kerberos: in der Datei `/private/etc/authorization` sucht man den Eintrag `<key>system.login.console</key>`, danach `<key>mechanisms</key>` und tauscht das nächste `<string>authinternal</string>` gegen `<string>builtin:krb5authnoverify,privileged</string>` aus.

Kerberos am Mac

- Kerberos wird bei MacOS X mitgeliefert
- Konfiguration wie in Linux/Unix, allerdings in der Datei `/Library/Preferences/edu.mit.Kerberos`; in neueren Versionen angeblich auch in `/etc/krb5.conf`.
- GUI: `/System/Library/CoreServices/Kerberos`
- Login mit Kerberos: in der Datei `/private/etc/authorization` sucht man den Eintrag `<key>system.login.console</key>`, danach `<key>mechanisms</key>` und tauscht das nächste `<string>authinternal</string>` gegen `<string>builtin:krb5authnoverify,privileged</string>` aus.
- Doku: `http://web.mit.edu/macdev/KfM/Common/Documentation/`
- Mac OS X Kerberos Extras bietet die notwendigen CFM Support Libraries für Applikationen (Eudora,...) in `/Applications/Utilities`.

Microsoft Windows

Microsoft verwendet für die Authentizierung in Active Directory ebenfalls Kerberos, die notwendigen Routinen sollten daher ab Windows 2000 vorhanden sein. Ich habe auf Windows trotzdem immer die Variante *MIT Kerberos for Windows* installiert.

Microsoft Windows

Microsoft verwendet für die Authentizierung in Active Directory ebenfalls Kerberos, die notwendigen Routinen sollten daher ab Windows 2000 vorhanden sein. Ich habe auf Windows trotzdem immer die Variante *MIT Kerberos for Windows* installiert.

Zur Konfiguration kann man die Einstellungen von Unix einspielen, die unter dem Namen `krb5.ini` in `C:/Windows/` abgelegt werden.

Die grafische Benutzeroberfläche *Network Identity Manager* ist recht intuitiv zu bedienen und erlaubt auch die Konfiguration einiger Eigenschaften von Kerberos.

Single Sign On - SSH

Bei einigermaßen aktuellem OpenSSH funktioniert die Konfiguration:

/etc/ssh/sshd_config

```
GSSAPIAuthentication yes
```

Single Sign On - SSH

Bei einigermaßen aktuellem OpenSSH funktioniert die Konfiguration:

/etc/ssh/sshd_config

```
GSSAPIAuthentication yes
```

/etc/ssh/ssh_config

```
GSSAPIAuthentication yes
GSSAPIDelegateCredentials yes
```

Zusätzlich müssen Host-Keys (in /etc/krb5.keytab) für die beiden beteiligten Maschinen vorhanden sein und das TGT muß *forwardable* sein.

Single Sign On - IMAP

- WU imapd
- Mozilla Thunderbird ($\geq$ 1.5) – (Account Settings - Use Secure Authentication)
- pine
- mutt – set imap_authenticators="gssapi"
- Apple Mail

Voraussetzung sind Host-Key und Service-Key (`imap/<server>@<realm>`) für den IMAP-Server.

Single Sign On - WWW

- Mozilla Firefox ($\geq$ 1.5) oder Internet Explorer ($\geq$ 6.0)
- Apache mit Modul mod_auth_kerb

Single Sign On - WWW

- Mozilla Firefox ($\geq$ 1.5) oder Internet Explorer ($\geq$ 6.0)
- Apache mit Modul mod_auth_kerb

Konfiguration am Server

```
AuthType Kerberos
AuthName "Institute of Theoretical and Computational Physics"
Krb5Keytab  /etc/krb5.keytab.apache
require valid-user
```

Single Sign On - WWW

- Mozilla Firefox ($\geq$ 1.5) oder Internet Explorer ($\geq$ 6.0)
- Apache mit Modul mod_auth_kerb

Konfiguration am Server

```
AuthType Kerberos
AuthName "Institute of Theoretical and Computational Physics"
Krb5Keytab  /etc/krb5.keytab.apache
require valid-user
```

Der Serverbereich sollte durch TLS/SSL abgesichert werden, weil das Apache-Modul ein lokales Ticket erzeugt, wenn der Client kein Ticket vorweisen kann – dann wird nach Username und Passwort gefragt. Benötigt wird zusätzlich ein Servicekey `HTTP/<server>@<realm>`.

Beliebte Fallen

Beim Umgang mit Kerberos tappt man gelegentlich in einige Fallen.

- Uhren sind nicht synchronisiert – der *authenticator* hat eine kurze Lebensdauer von typischerweise 5 Minuten, um *replay Attacken* zu erschweren. Empfehlenswert ist die Synchronisation der Beteiligetn Maschinen mit zuverlässigen NTP-Servern.

Beliebte Fallen

Beim Umgang mit Kerberos tappt man gelegentlich in einige Fallen.

- Uhren sind nicht synchronisiert – der *authenticator* hat eine kurze Lebensdauer von typischerweise 5 Minuten, um *replay Attacken* zu erschweren. Empfehlenswert ist die Synchronisation der Beteiligetn Maschinen mit zuverlässigen NTP-Servern.
- NAT – die Tickets enthalten im Normalfall die Netzwerkadresse das Clients, um *spoofing* zu verhindern; das gibt spätestens bei mehr als einem Client hinter einem Router mit *network address translation* ein Problem. Als Hilfe in diesen Fällen kann ein *address-less* TGT (`kinit -A`) erzeugt werden.

Zukunft — work in progress

Wie jede gesunde Infrastruktur ist auch Kerberos am Wachsen und Verändern:

- asymmetrische Kryptografie (*public keys*, RFC 4556)
- *smart cards* und *tokens*

Zukunft — work in progress

Wie jede gesunde Infrastruktur ist auch Kerberos am Wachsen und Verändern:

- asymmetrische Kryptografie (*public keys*, RFC 4556)
- *smart cards* und *tokens*
- Administrationsinterface soll standardisiert werden

Zukunft — work in progress

Wie jede gesunde Infrastruktur ist auch Kerberos am Wachsen und Verändern:

- asymmetrische Kryptografie (*public keys*, RFC 4556)
- *smart cards* und *tokens*
- Administrationsinterface soll standardisiert werden
- effizienteres Propagieren der DB auf redundante Server
- mehr Unit- und Protokoll-Tests (*validation suite*)

Literatur — weitere Informationen


MIT Kerberos Team.
Kerberos: The Network Authentication Protocol.
WWW - http://web.mit.edu/kerberos/www/.


Jason Garman.
Kerberos: The Definitive Guide.
O'Reilly, Sebastobol, CA 95472, 1 edition, 2003.


Brian Tung.
The Moron's Guide to Kerberos.
WWW - http://www.isi.edu/~brian/security/kerberos.html, 2007.


Jim Rome.
How to Kerberize your site.
WWW - http://www.ornl.gov/~jar/HowToKerb.html.


Clifford Neuman, Tom Yu, Sam Hartman, and Ken Raeburn.
The kerberos network authentication service (v5).
RFC 4120 - http://www.ietf.org/rfc/rfc4120.txt, 7 2005.

Danke für Ihre Aufmerksamkeit!

Andreas Hirczy
http://itp.tugraz.at/~ahi/
mailto:ahi@itp.tugraz.at

Diese Präsentationsunterlage finden Sie auf http://itp.tugraz.at/~ahi/VO/2007-05_GLT_kerberos.pdf und demnächst auch auf http://linuxtage.at/.

Fragen?