L'AAA, késako ?

Bruno Bonfils, <asyd@solaris-fr.org>,

Novembre 2005

Introduction

Sous ce terme d'apparence barbare est regroupé l'ensemble des concepts suivants :

- Authentication (authentification)
- Authorization (autorisation)
- Accounting (comptabilité)

Ce papier à pour but d'expliquer rapidement les concepts qui se cachent derrière ces mots, et de parler des exemples les plus classiques. En effet, bien que le terme d'AAA soit principalement utilisé dans un contexte PPP et Radius, ce document abordera ces trois points pour une utilisation générique.

Table des matières

Authentication (authentification)

Cette opération valide une identité. Cela se traduit la plupart du temps par un couple identifiant / mot de passe, mais on peut également utiliser d'autres formes, comme un certificat (cf remarques), une carte à jeton, ou encore la biométrie (empreinte du pouce sur une certaine marque d'ordinateur portables), etc..

L'authentification est utilisée pour identifier de façon unique (login) un individu avec une information dont il est – censé – être le seul possesseur.

Fichier plat

On trouve plusieurs exemples de fichiers plats, le premier est /etc/passwd (couplé avec le fichier /etc/shadow de nos jours), le second sont les fichiers htpasswd utilisés – entre autre – par apache, qui contiennent simplement le couple identifiant:motdepasse.

PAM

PAM siginife Pluggable Authentication Module, soit en français : module d'authentification enfichable. En fait, c'est un ensemble de différents modules spécalisés. Ces différents modules ont pour rôle de confirmer un identifiant. Pour qu'une identitée soit validée, l'ensemble des modules marqués nécessaire (required) doit retourner une réponse positive, sinon l'identitée est invalide. L'avantage - pour un concepteur de logiciels - d'utiliser PAM est l'interface (API) unique de programmation. L'administrateur système peut donc, sans modifier les sources des applications, choisir sa méthode d'authentification.

Fonctionnement

On distingue quatres types d'opération :

- *auth* Vérifie l'idendité (et éventuellement changement d'identité)
- *account* Opérations supplémentaires (autorisation par exemple)
- *passsword* Mise à jour de mot de passe (modification du /etc/shadow)
- *session* Journalisation

Notez bien qu'un même module peut implémenter une ou plusieurs opérations. Par exemple, pam_unix permet l'authentification (*auth*) via /etc/password et /etc/shadow (si existant), le changement du mot de passe d'/etc/shadow (*password*) et mise à jour des fichiers *login records* (utmp / wtmp).

Exemples de modules PAM d'authentification

Voici une liste non exhaustive de module d'authentification pour PAM :

- *pam_unix* utilisation des utilisateurs systèmes, via NSS ou /etc/passwd et /etc/shadow
- *pam_ldap* interrogation d'un annuaire via le protocole LDAP
- *pam_mysql* interrogation d'une base de donnée type MySQL
- *pam_pgsql* interrogation d'une base de donnée type PostgreSQL
- *pam_krb5* Obtention / vérification d'un ticket Kerberos

Exemples de modules PAM additionnels

De plus, il permet l'utilisation d'autres modules optionnels, on peut citer par exemple :

- *pam_mkhomedir* permet la création du répertoire utilisateur (home directory) à la connexion si celui-ci n'existe pas
- *pam_cracklib* vérification de la complexité du mot de passe (lors de sa modification)
- *pam_moumt* montage automatique à la connexion

Historique

La sociétié Sun Microsystems est à l'origine de PAM pour son système d'exploitation Solaris. On le

trouve néanmoins – en standard – sur la grande majorité des systèmes Unix.

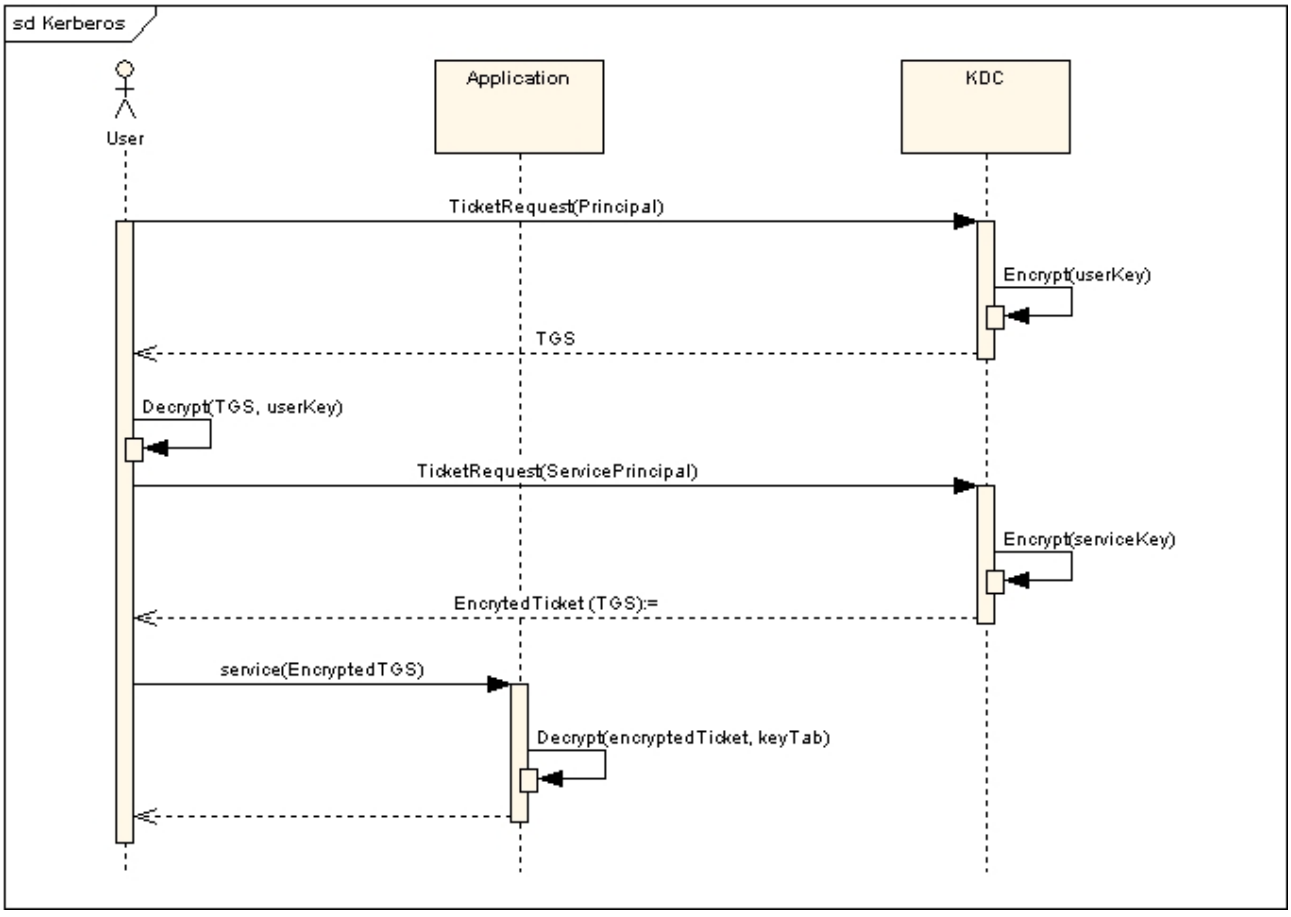
JAAS

JAAS, pour Java Authentication and Authorization Services est une implémentation de PAM pour le language Java. En plus de son fonctionnement identique à PAM, il fournit également d'autres méthodes, comme doAs (appel de méthode sous une identitée spécifique).

Kerberos

Kerberos est un protocole originaire du MIT (Massachusetts Institute of Technology), son fonctionnement repose sur un système de tickets.

Principe (simplifié) de fonctionnment



Les protocoles supportés par Kerberos

- LDAP
- Telnet
- FTP
- Et dans une moindre mesure HTTP

Kerberos dans le monde du Web

Les navigateurs Internet Explorer, Mozilla et dérivés peuvent négocier l'échange d

Quelques solutions

- CAS (Central Authentication Services)

SASL

SASL signifie Simple Authentication and Security Layer, ou dans la langue de Molière couche simple (sic) d'authentification et de sécurité. On peut – dans une certaine mesure – comparer SASL à PAM. En effet, cela reste une méthode (selon le point de vue du développeur) d'appel unique pour authentifier un utilisateur. Néanmoins, à la différence de PAM, SASL est standardisé via la RFC 2222, et surtout, il rajoute des mécanismes d'appel (handlers) relatif à la sécurité (TLS par exemple) permettant au développeur de s'affranchir de la méthode d'obtention du mot de passe.

La principale (unique ?) implémentation libre pour le monde Unix est Cyrus SASL.

Listes des backends disponibles pour Cyrus SASL (directement ou via saslauthd)

- pam
- sasldb
- SQL (MySQL et PostgreSQL)
- LDAP

Remarques

Authentification basé sur certificats

Aujourd'hui, le couple identifiant / mot de passe peut être remplacé par un certificat, dans ce cas, il tient lieu d'identifiant et on parle alors d'authentification basée X509 (ou certificat). Le mot de passe n'existe plus (cf remarque), et est remplacé par diverses opérations (optionnelles ou obligatoire) sur le certificat, on trouve notamment :

- vérifier que le certificat est signé par une CA de confiance (opération obligatoire)
- une éventuelle vérification sur la profondeur de signature
- une éventuelle vérification du certificat dans la CRL associé à celui-ci ou au certificat de l'autorité

Remarque

Les différentes applications (navigateur, lecteur de courriel, explorateur de fichiers) protégent leurs trousseaux de *clées* (keyring en anglais, une clée étant un composant d'un certificat) par un mot de passe.

Authorization (autorisation)

L'opération d'autorisation consiste à dire si un utilisateur a le droit d'accéder à une resource précise, application, ou partie d'application (mode administrateur). Il n'existe pas d'interface (en dehors de JAAS) générique pour cette opération.

PAM

Il est possible dans un module, d'utiliser le nom du service (l'appelant du module donc) pour une opération d'autorisation. Malheureusement, peu de module utilise vraiment cette fonctionnalitée (cf remarque). Si vous êtes intéréssés pour une autorisation avancée avec PAM, jetter un oeil à

pam_eaccess (en cours de dévelopement).

LDAP

Si vos applications permettent l'utilisation de LDAP pour l'authentification, si vous pouvez définir vous même vos filtres, vous pouvez utiliser un attribut (de préférence multivaleur) pour définir les droits des utilisateurs.

Exemple d'authentification et autorisation LDAP avec apache

```
<Location "/admin">
  AuthType Basic
  AuthName "Admin”
  AuthLDAPURL ldap://ldap.solaris-fr.org/ou=extranet,dc=solaris-fr,dc=org?cn?sub?(host=admin)
  AuthLDAPBindDN cn=Apache,ou=Security Users, dc=solaris-fr, dc=org
  AuthLDAPBindPassword secret
</Location>
```

En utilisant le module *mod_auth_ldap*, ces directives permettent l'authentification (via vérification de l'identifiant et du mot de passe) et l'autorisation (via le filtre sur l'attribut host). En utilisant ce mécanisme, un administrateur est en mesure d'utiliser une seule branche (ici ou=extranet) d'un annuaire (accessible via le protocole LDAP) tout en rajoutant/supprimant des autorisations via la modification de l'attribut host.

Accounting (journalisation)

Cette opération consiste à garder une trace des différentes actions d'un utilisateur. Un fichier de log (de type serveur web) est une forme de journalisation.

Glossaire

CA	Certification Authority
CRL	Certificate Revocation List
PAM	Pluggable Authentication Module
SASL	Simple Authentication and Security Layer (RFC 2222)
LDAP	Lightweight Directory Access Protocol (RFC 1777)
GSSAPI	Generic Security Service API (RFC 2078)