Mathrice - Lille - 21 octobre 2004

Authentification avec Kerberos

(ou remplacer NIS par Kerberos (+LDAP))

©
<bernard.perrot@univ-rennes1.fr>

Authentification « basique »

- Pourquoi est-ce que « tout le monde » utilise NIS ?
  - Facile à mettre en œuvre.
  - Facile à administrer (rien à faire...).
  - Supporte bien le facteur d'échelle (scalability).
  - Très largement supporté (très bonne interopérabilité).
- Mais que fait NIS ?
  - Il diffuse des tables.
  - Et C'EST TOUT !
- NIS ne FAIT PAS d'authentification :
  - Il ne fait que propager des enregistrements des fichiers `/etc/passwd`, `/etc/group` et `/etc/shadow`
  - Sur une machine autonome, NIS ne fait pas plus que s'il n'était pas là...

Authentification « basique »

- Sur un système Unix avec PAM, c'est le module pam_unix (et quelques autres pour des tâches plus spéciales) qui réalise l'authentification (c'est à dire vérifie que la signature du mot de passe entré est la même que celle stockée dans `/etc/shadow`).

- Si on fait de « l'authentification LDAP », on remplace NIS par LDAP pour la diffusion des informations d'identité,
  - Les communications sont plus sûres (SSL avec les clients).
  - Ajouts de toute une panoplie, bien utile, de services d'annuaire.
  - Mais rien de neuf en ce qui concerne la fonction d'authentification proprement dite.

Authentification « basique »

- Parmi les autres options rencontrées :
  - Copie (cp, scp, rsync,...) du passwd/shadow local :
    - Nécessite des mécanismes « root » pour communiquer, est-ce de confiance ?
    - Désynchronisation possible.
  - NIS+
    - Complexe.
    - Bien moins supporté que NIS.
    - Mort-né...
  - Quoi encore ... ?

Authentification

- Mais que voudrait-on ?
  - suppression des mots de passe clairs (ou chiffrés « faibles ») qui circulent sur le réseau (soit lors de séquence d'authentification, soit par fuite directe du contenu des tables de passwords).
  - Gestion intégrée d'une politique de mots de passe (longueur, entropie, durée de vie).
  - Gestion intégrée de l'expiration de comptes.
  - Auditing intégré.
  - SSO (Single Sign On).
  - Redondance du service d'authentification.
  - Compatible avec le maximum de systèmes clients.
  - Identifier des utilisateurs, mais aussi des machines
  - En fait un « vrai » service d'authentification, pas un seul « mécanisme » !
- Kerberos ?

Kerberos

- Points de départ :

  `http://web.mit.edu/kerberos/`
  `http://www.isi.edu/gost/info/kerberos/`

- Plusieurs RFC, dont les deux de base :
  - RFC-1510 (Kerberos Network Authentication Service V5, sept. 1993)
  - RFC-1964 (Kerberos V5 GSS-API mechanism, juin 1996)

- Plein de choses chez Microsoft, par exemple :

  `http://www.microsoft.com/windows2000/techinfo/howitworks/security/kerberos.asp`

  et beaucoup d'autres documents technique sur l'interopérabilié.

- Mais, Kerberos n'est-il pas un protocole déjà ancien d'un projet oublié (Athena) ? Pourquoi ressortir des vielles casseroles ?

Kerberos

- Juste un extrait de ce document pour démystifier :

  « *The Kerberos version 5 authentication protocol is the default for network authentication on computers with Windows 2000* »

  (nb: cela continue d'être vrai avec XP bien entendu)

- Donc, en envisageant Kerberos comme système d'authentification, on ne fait que s'aligner sur le protocole utilisé par l'immense majorité du parc (micro)informatique mondial actuel. C'est bien ou c'est mal, mais c'est tout sauf être exotique !
- C'est interopérable, complexe, mais pas spécialement compliqué (c'est à dire qu'il faudra lire la doc et y passer un peu de temps au début...)

Kerberos

- Pourquoi Kerberos ?
  - LDAP a été conçu pour publier de l'information (publique).
    - Il y a bien des ACLs pour protéger les champs tels que userPassword, mais...
- Kerberos permet une politique de mots de passe
  - Examen des nouveaux passwords contre dictionnaires, règles d'entropie.
  - Gestion de l'expiration des mots de passe
  - Kerberos réalise des authentifications, mais les passwords chiffrés ne quittent jamais le système (serveur) : moins de risque d'interception ou compromission
- Kerberos permet en plus :
  - Une compatibilité avec Windows
  - De l'authentification et chiffrement sur NFS
  - L'intégration d'AFS

Kerberos

- Avant que Microsoft n'en fasse son système d'authentification, la diffusion/culture de Kerberos était limitée :
  - Sans doute à cause de difficultés (légales pas techniques) liées à l'exportation de produit de cryptologie en dehors des USA (si conçus aux USA)
  - En France s'y est ajouté la difficulté jusqu'à assez récemment de pouvoir l'importer/l'utiliser « légalement ».
- La communauté des sites utilisant AFS (dans notre milieu, la Physique des Hautes Énergies, l'In2p3 en particulier) font exception et utilisent Kerberos depuis longtemps.
  (AFS = Andrew File System)
- C'est un système de type « tiers de confiance »
- Nb: kerberos utilise de la crypto, mais ne chiffre pas de données

Kerberos (clients)

- A l'origine, les applications voulant profiter de Kerberos devait être « Kerbérisée », c'est à dire contenir du code spécifique pour être un client « natif ».
- Parmi les applications kerbérisée, on peut citer :
  - Cyrus IMAP (serveur)
  - OpenLDAP (serveur)
  - OpenSSH (serveur et client)
  - Reflexion X
  - Eudora
  - Apple Mail.app (MacOSX)
  - Telnet
  - Cisco
  - mod-auth_krb
  - etc.

Kerberos (clients)

- La nécessité de disposer d'applications kerbérisée serait un frein à l'utilisation de Kerberos, mais ...
- ...PAM vient resoudre le problème !
- Sur un système « PAMifié », il existe un (plusieurs même) module pam_krb5 qui permet de rendre l'usage de Kerberos transparent : une simple configuration de PAM fera que Kerberos deviendra le système d'authentification de la machine pour toutes les applications utilisant PAM (quasi-toutes maintenant sur un Unix moderne).
- Cependant, il faudra des applications kerbérisée pour profiter des jetons (transmis par SSH par exemple).
- Si les applications PAMifiées font circuler les mots de passe en clair, on perd la sécurité procuré par Kerberos par rapport à cela (remède, ajouter la couche SSL à ces applications).

Kerberos (critiques)

- Pour :
  - Sécurité réellement renforcée.
  - Outils de gestion/administration du produit.
- Contre :
  - configuration/administration compliquée ?
    - Est-ce VRAIMENT plus compliqué que configurer/administrer un serveur HTTPS (configuration SSL) ? Qu'un serveur LDAP (S) ? Qu'un serveur W2K3 ?
  - Pas d'avenir ?
    - Que l'on aime ou pas Microsoft, ne peut-on supposer que parmi les possibilités, son choix a été vraiment pragmatique (et à noter que sur ce point, un des seuls, il n'a pas juger mieux, ni même commercial, de réinventer...)
    - Pensez-vous que Microsoft va changer une fois de plus de système d'authentification dans les prochains mois/années ?
  - Concept ancien (cryptographie à clés secrètes et pas publiques) :
    - bof... et si ça marche ? (et évolution possible à l'étude)

Kerberos (fonctionnement)

- Kerberos stocke des paires {username,password}.
  - Les « usernames » sont appelés « principals » dans Kerberos.
  - La base de donnée interne peut être assimilée à `/etc/shadow`.
- Les mots de passe (chiffré ou non) ne sont jamais transmis sur le réseau
- Quand un utilisateurs s'authentifie, il lui est délivré un « ticket »
  - Ces tickets ont une durée de vie (8H en général par défaut)
  - Utiles pour des usages tels que IMAP, AFS, NFS authentifié,... (fonctionnalité Single Sign On)
- Kerberos procure de l'authentification, PAS d'autorisation :
  - Il garanti que celui qui dit s'appeler X est bien X.
  - C'est au client de décider ce que X a le droit de faire
    - (même principe que la certification X509)

Kerberos (terminologie)

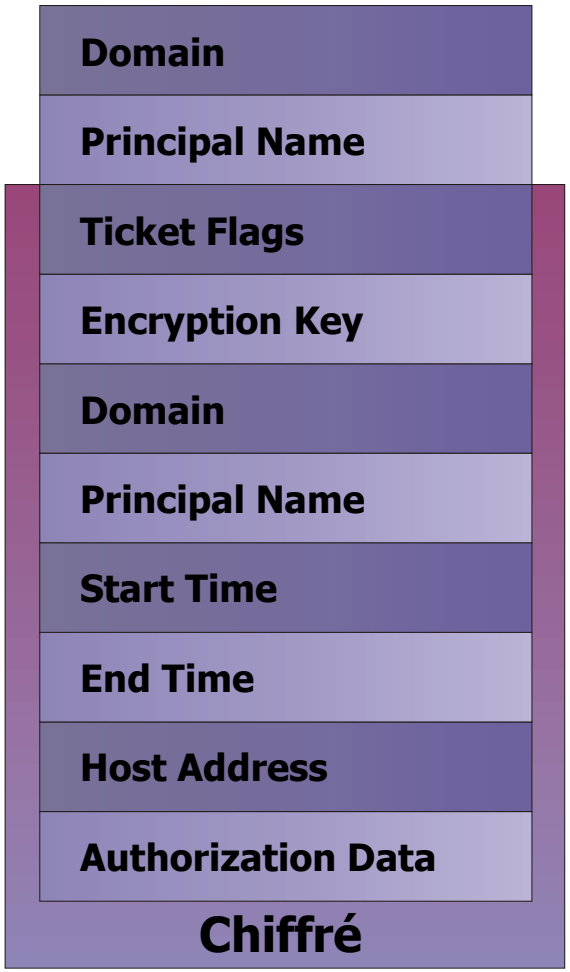
- « realm » = domaine d'authentification : typiquement, c'est un nom de domaine IP (en lettres capitales)
  - MATH.UNIV-RENNES1.FR
- « Principal » = trinôme (nom, instance, realm) : `nom/instance@realm`
  - Nom (primary) = nom d'utilisateur ou service
  - Instance = qualifie le nom (rôle/groupe)
  - perrot@MATH.UNIV-RENNES1.FR
  - perrot/admin
  - mon-pc/domain.tld
  - ldap/ldap.math.univ-rennes1.fr
- KDC (Key Distribution Center) : le serveur d'authentification Kerberos (machine dédiée hautement recommandé). Base de donnée des « principaux » et clés associées.

Kerberos (terminologie)

- Client : entité pouvant obtenir un ticket (utilisateur/hôte)
- Ticket : crédits (identité d'un client pour un service particulier)
- TGT (Ticket Granting Ticket) : ticket donné par l'AS (Authentication Service) permettant d'obtenir d'autres tickets (pour le même realm).
- Le KDC (Key Distribution Center) :
  - Responsable de la maintenance des clés maîtres et de la mise à disposition des tickets
  - L'AS donne au client une clé de session et un TGT
  - Distribue les clés de sessions et les tickets via le Ticket Granting Service (TGS)

Kerberos (ticket)

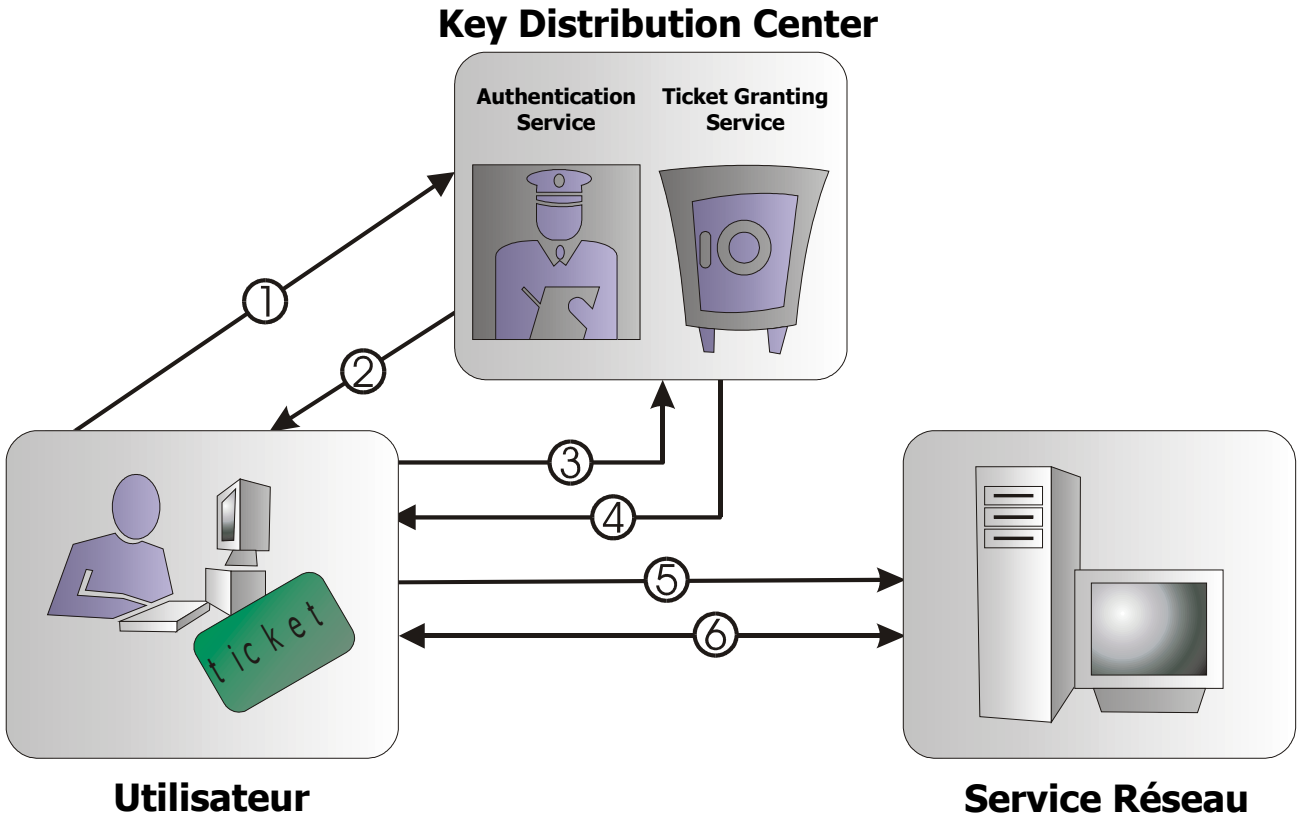
- Contenu d'un ticket :



Ce dessin et suivants honteusement piqué à N. Fischbach sur le site de securite.org (présentation Kerberos à l'OSSIR, 2001)

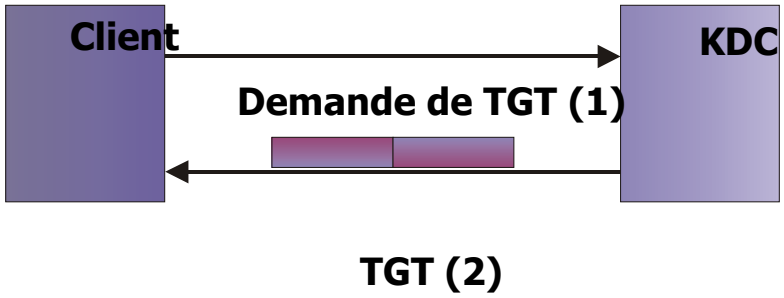
Kerberos (protocole)

- Échanges de tickets :



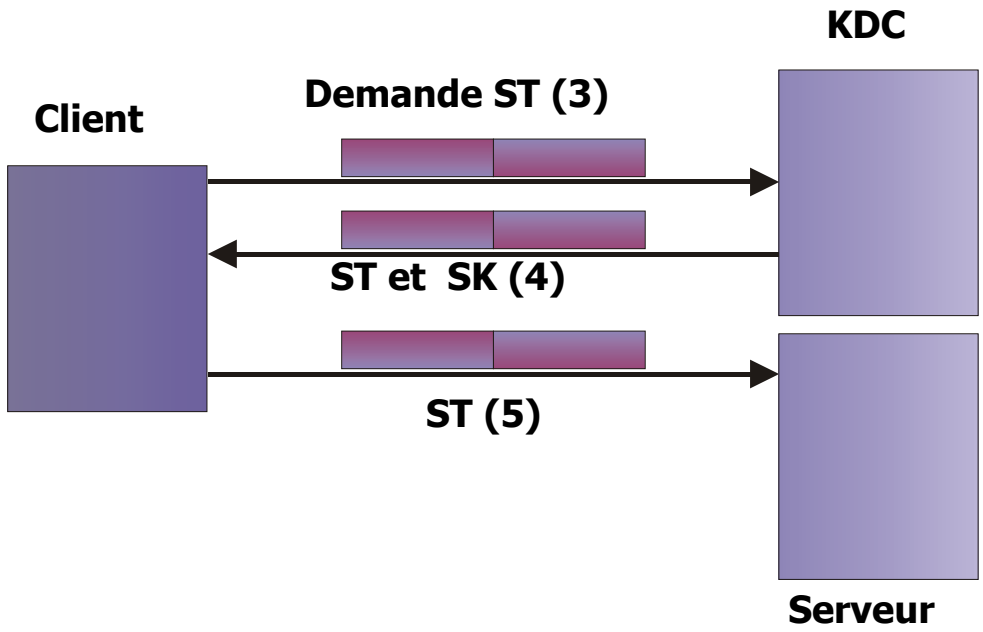
Kerberos (protocole)

- Obtention d'un TGT (1 + 2)
  - 1 : demande d'un TGT
  - 2 : TGT (déchiffré avec le hash du mot de passe utilisateur)



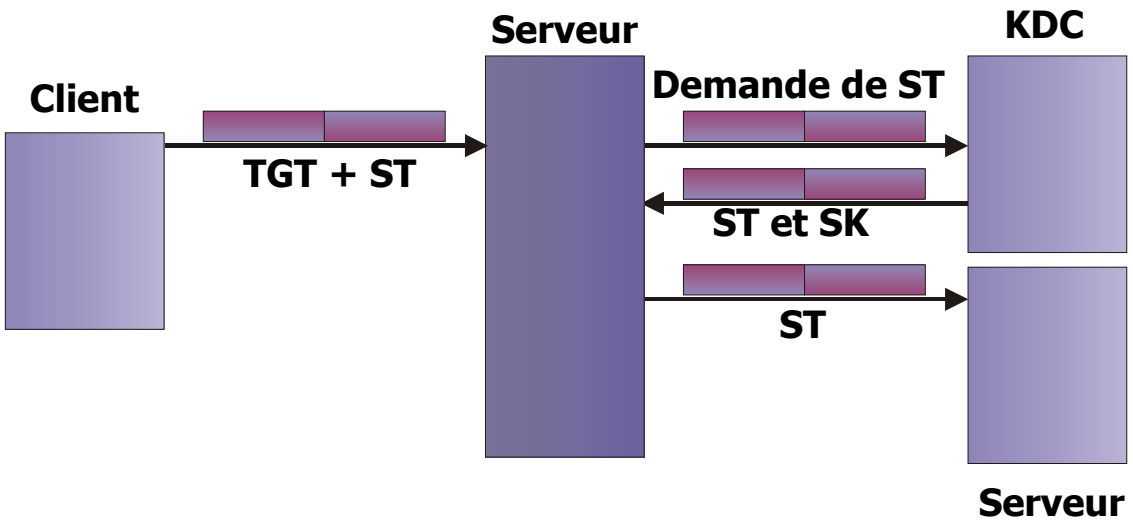
Kerberos (protocole)

- Obtenir et utiliser un Service Ticket (3 + 4 + 5) :
  - 3 : Demande de ST (avec le TGT)
  - 4 : obtention ST et clé de session
  - 5 : ST pour l'authentification



Kerberos (protocole)

- Délégation de l'authentification :



Kerberos

- Théorie, fonctionnement :
  - Voir la documentation indiquée en référence pour détails complémentaires, le temps imparti à cet exposé ne suffira pas sinon...
- Il y a un dispositif pour permettre l'authentification croisée avec d'autres Realm (grâce à un partage de clés de chiffrement).
- Il y a un dispositif pour convertir les ticket v5 et tickets v4 (krb524), nécessaire si on doit utiliser des outils kerbérisés v4 et pas v5 (AFS en particulier).
- Système d'ACLs assez élaboré pour les droits des administrateurs (permet de déléguer)

Kerberos (ports)

- Si le trafic Kerberos doit traverser un garde-barrière, il faut veiller au filtrage des ports suivants utilisés par le protocole :
  - 88/tcp/udp : clients <-> KDCs (trafic régulier d'authetification)
  - 464/tcp/udp : kpasswd (ancien protocole) (pour Windows ?)
  - 749/tcp : clients -> master KDC (kpasswd, administration)
  - 754/tcp : master KDC -> slave KDCs (replica database)
  - 4444/udp : traduction ticket Kerberos v5 vers Kerberos v4 (krb524)

  - Kerberos 4 utilise les ports 750/tcp/udp, 751/tcp/udp et 761/tcp (et quelques autres encore), mais il est désormais fort déconseillé d'utiliser Kerberos v4, et encore moins de laisser passer son trafic au delà du garde-barrière !

Kerberos (attaques)

- Compromission (root) du serveur hébergeant le KDC :
  - C'est évident... pas un problème spécifique à Kerberos, mais Kerberos permet (recommande) une machine dédiée (pas de comptes utilisateurs) au KDC, et donc mieux maîtrisée et surveillée.
- Compromission du compte administrateur :
  - C'est évident... pas un problème spécifique à Kerberos non plus...
- Compromission (root) d'un serveur :
  - Un accès root sur un serveur permet de compromettre les services, y compris Kerbérisés de ce serveur. Mais cela ne permet pas de casser le protocole mais permet des attaques par force brute ou dictionnaire sur les tickets.

Kerberos (attaques)

- Compromission (root) d'une machine cliente :
  - Vis à vis de Kerberos proprement dit, les tickets ayant une durée de vie limitée, l'impact est faible. Évidement, la machine compromise peut devenir une backdoor (keylogger par exemple) et l'attaque s'amplifier.
- Compromission d'un compte utilisateur :
  - Si c'est un ticket qui est compromis, il ne peut être utilisé que pendant sa durée de vie
  - Si le password utilisateur est compromis... c'est un problème général de tous les systèmes d'authentification sauf les OTP !!!
- Deni de service sur les KDCs :
  - Possible, mais pas une spécificité...

Kerberos (attaques)

- Attaquant autorisé, social engineering, ... :
  - Problème non spécifique...
- Trous de sécurité dans l'implémentation :
  - Aucun logiciel n'est parfait.
  - Mais ne semble vraiment pas la cause d'insécurité majeure en ce moment !
- Force brute, attaques à dictionnaire :
  - Possibilité sur Kerberos v4 (sur les tickets)
  - La v4 utilise du simple DES (« fragile » si attaquant **très** motivé)
  - Contre-mesures dans la v5 (pré-authentification, 3DES, RC4)
    - Il faut que le parc soit homogène à niveau
    - Windows reconnaît RC4 mais pas le 3DES
  - Si le KDC ets sur une machine dédiée, les mots de passe (chiffrés) ne sont jamais diffusés contrairement aux méthodes « classiques », c'est donc moins pire.

Kerberos (attaques)

- Rejeu :
  - Il est envisageable d'intercepter un ticket et le rejouer
  - Attaque très complexe (contre-mesures)

- Man-in-the-middle :
  - Problème commun à tous les protocoles réseau...
  - Kerberos dispose d'un mécanisme interne pour lutter contre cette attaque
  - Mais reste possible (si par exemple on n'utilise pas les tickets mais seulement les passwords comme le fait PAM)
  - Bon, IPv4 est fragile de toute façon !

Kerberos (attaques)

- Les contre-mesures classiques aux attaques citées :
  - Utiliser la pré-authentification.
  - Protéger efficacement les serveurs, et surtout les KDCs.
  - Avoir une politique de mots de passe :
    - Expiration (durée de vie des passwords)
    - Historique (non-réutilisation)
    - Entropie contrôlée (casse des caractères, chiffres, caractères spéciaux, ...)
    - Le KDC du MIT permet de faire tout cela
  - Auditer, logguer

Kerberos (installation)

- Plusieurs implémentations possible sous Unix :
  - MIT
  - Heimdal
  - SEAM
- Prenez MIT, vous ne prendrez pas de risque !
- Windows Domain Controller en est une aussi...
- Configuration :
  - Hors limite du temps de cet exposé
  - Parfaitement bien détaillée dans la documentation de référence

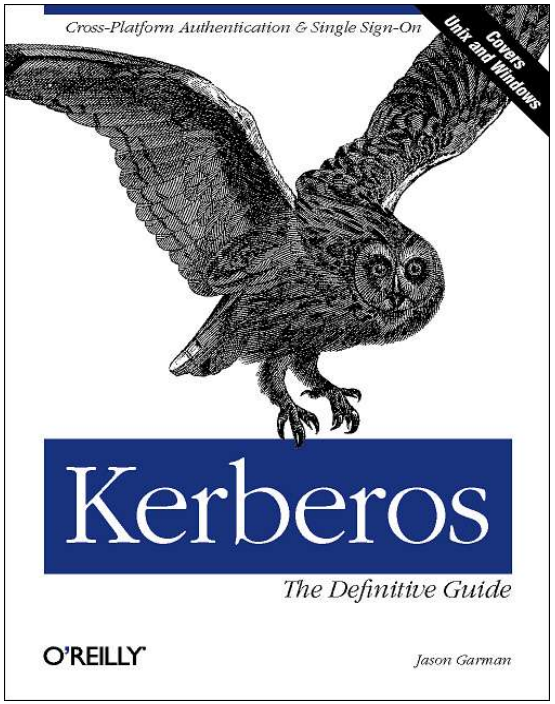
Kerberos (contraintes)

- Quelques commandes utilisateurs nouvelles :
  - (yp)passwd -> kpasswd, kinit, klog, ...
- La sécurité/fiabilité dépend de service réseau de qualité :
  - DNS évidement
  - Service de temps : tous les clients doivent être synchrone, sinon, risque de distorsion et des autorisations risquent d'échouer
  - À contrario, une attaque réussie sur le service de temps doit permettre des rejeux de tickets expirés.
- Un password expiré bloque le compte :
  - i.e. : on n'a pas la possibilité de se connecter très en retard pour au moins n'avoir que le droit de changer son password
  - Prévoir un mécanisme pour avertir les utilisateurs à l'avance,
  - Sinon, seul un admin du realm pourra débloquer la situation.
    - Souci si vous êtes seul capable de le faire et en congés...

Kerberos

- Documentations :
  - Beaucoup sur le Web…
  - Chez O'Reilly :

    Kerberos: The Definitive Guide
    (*cover Unix and Windows*)
    August 2003
    ISBN: 0-596-00403-6



Hesiod

- Si on veut migrer de NIS à Kerberos (et avant/sans utiliser LDAP), que fait-on avec les tables `passwd` et `group` (Kerberos s'occupe de `shadow`) ?
- Low-tech : cp, scp/sftp, rsync, ...
- On peut garder NIS pour cela, pas de problème de sécurité, mais a-t-on encore envie de dépendre de NIS pour la stabilité de son système ?
- On peut tenter HESIOD...

Hesiod

- C'est un autre protocole original du projet Athena
- Principe : utiliser le DNS pour diffuser les tables `passwd` et `group` (au sens `/etc/nsswitch.conf`)
- Bien que peu connu, supporté depuis longtemps par Bind
- Utilise des enregistrements `TXT` (nom -> record) et `CNAME` (ID -> nom)
- La classe prévue est `HS`, mais peu aussi être `IN`
  (à indiquer dans le `/etc/hesiod.conf`)
- Par exemple :

```
perrot.passwd  TXT    "perrot:x:123:789:Bernard Perrot:/users/perrot:/bin/bash"
perrot.group   TXT    "perrot:x:789:"
123.uid        CNAME  perrot.passwd
789.gid        CNAME  perrot.group
admin.group    TXT    "admin:x:900:root,perrot"
900.gid        CNAME  admin.group
```

Hesiod

- Pas besoin d'être par ailleurs serveur DNS primaire ou esclave, on peut très bien démarrer un Bind dédié à cela (pas « déclaré » au sens NIC, filtré)
- Aisé de construire (automatiser) le fichier de zone à partir des fichiers standards `/etc/passwd` et `/etc/group`
  (je peux fournir le script)
- Prévu (en option et/ou en commentaire) depuis longtemps dans votre `/etc/nsswitch.conf`, regardez...
- Intégré à la glibc
- Ca marche et ne provoque pas plus d'échanges réseau (pas moins non plus) qu'un NIS.

Hesiod

- /etc/hesiod.conf

```
rhs=.domain.tld
lhs=.hesiod
classes=IN,HS
```

<-- on indique l'ordre là

- /etc/nsswitch.conf

```
passwd:     files hesiod
group:      files hesiod
```

- Interrogation manuelle (exemple):

```
% dig txt in perrot.passwd.hesiod.domain.tld
```

qui retourne :

```
;; ANSWER SECTION:
perrot.passwd.hesiod.domain.tld. 86400 IN TXT
   "perrot:x:123:789:Bernard Perrot:/users/perrot:/bin/bash"
```

Hesiod

- Compléments d'informations :
  - Documentations de Bind
  - `README.hesiod` de la glibc
  - http://www.mit.edu:8001/afs/athena.mit.edu/astaff/project/hesioddev/hesiod.dist.old/
  - http://www.linux.ncsu.edu/lug/linux-at-ncsu-faq/x81.html
- Peu connu, peu utilisé... n'utilise pourtant que du très standard et classique que l'on a forcement sur sa machine (glibc, Bind)...
  - Technique pas spécialement jeune, mais NIS l'est-il plus ?
  - Pourquoi ? Tombé dans l'oubli avec Athena ?
  - Effets de bord (que je ne connais pas) ?
  - L'interrogation d'un serveur Bind échoue avec des timeout « longs » : si le serveur est en panne, les clients vont ramer...

Migration

- La migration d'une authentification « classique » à Kerberos pose le problème de la constitution initiale de la base d'utilisateurs.
  - Pas un soucis pour la plupart des informations, sauf...
  - ... le mot de passe :
    - Classiquement, c'est un hash non réversible qui est stocké (shadow)
    - Il faudrait un « clair » pour créer le compte utilisateur...
  - Solutions possibles :
    - Repartir de zéro (les utilisateurs se ré-enregistrent) : même si techniquement possible, peut être socialement mal accepté !
    - pam_migrate_krb5 : un module PAM stackable qui créé un principal après identification classique.
    - pam_storepw (modifié BP) : un module PAM stackable qui stocke les passwords utilisateurs chiffré contre une clé publique (donc invulnérables sur la machine exécutant le module qui n'a pas la clé privée). On laisse tourner quelques temps, puis on peut (script, pas obligé de « lire ») automatiser la création des comptes.

LDAP

- Objectif :
  - Savoir gérer cet enregistrement :

    ```
    userPassword: {KERBEROS}untel@domain.tld
    ```

    plutôt que :

    ```
    userPassword: {crypt}$1$MPWwsxra$X4u4jVrw/65cSyFPgh2
    ```

  - C'est à dire que la base LDAP ne contiennent plus de mot de passe (même chiffrés/hashés), mais délègue.
    - (nb: ne se résumera pas qu'à l'enregistrement indiqué quand même).

LDAP

- L'authentification avec Kerberos+LDAP va se dérouler ainsi :
  - Le client (PAMifié) va s'authentifier auprès du serveru Kerberos, qui va retourner un ticket
  - Ce ticket va être utilisé par le serveur LDAP (car OpenLDAP est Kerbérisé) pour achever la phase d'authentification (au lieu de retourner le hash du password dans le cas d'une « authentification LDAP» sans Kerberos).
  - Et voilà...
- Il va y avoir « disjonction » entre la config de nsswitch.conf et pam.d :

  `passwd: files ldap` (`/etc/nsswitch.conf`) (idem pour `group` et `shadow`)

  ...

  `auth sufficient /lib/security/pam_krb5.so` (`/etc/pam.d/system-auth`)

  contrairement à l'habitude.
- Kerberos est donc un plus apporté à l'usage de LDAP comme système unificateur de l'authentification, pas un concurrent.

LDAP + Kerberos

- Mise en oeuvre / configuration :
  - Hors limite du temps de cet exposé...
  - Assez technique, lassant pour un auditoire faiblement près à franchir le pas...
  - Fort bien détaillé dans la documentation de référence indiquée.
- Des précautions (pas d'impossibilité) pour que Windows puisse utiliser le service.
- Dispositif assez high-tech, donc forcement plus fragile que pam_unix + {password,group,shadow} synchronisés par scp ou rsync...mais plus sûr...

LDAP

- Compléments de documentations :
  - « Replacing NIS with Kerberos and LDAP » :

    `http://www.fb.net/jheiss/krbldap/`
  - Kerberos (O'Reilly)
  - « Comptes réseau avec LDAP et Kerberos V » :

    `Red Hat Magazine (France) 2003 n°1, pp 38-44`

Des questions ?