Evolution de l'authentification

Mardi 25 mai 2010

Emmanuel Blindauer
e.blindauer@unistra.fr
Direction Informatique
Université de Strasbourg

Plan

- {Qu'est ce que,Pourquoi} authentifier ?
- Les techniques et chiffrements
- Les (in)compatibilités
- Les possibilités
- Les évolutions à venir
- Conclusion

Q'est ce qu'authentifier?

- « Salut, je suis Bart, ouvre moi la porte »
  - « Euh... ??? Prouve le ! »
- Comment prouver son identité ?
- Comment vérifier la preuve ?
- Avoir eu des données communes précédemment
  - Des connaissances (mémorisé, façon de signer...)
  - Possession physique (token, photo, emprunte digitale, …)
- Après vérification, on sait qui est Bart. Par contre, ce qu'il peut faire ...

Q'est ce qu'authentifier?

- Certains token peuvent être changés (mot de passe, clef USB, ...)
- D'autres, plus ... difficilement (photo, empreintes digitales)
- Il faut qu'un changement de token soit répercuté
- Il ne faut pas qu'une tierce personne ait cette information

Pourquoi authentifier ?

- Pourquoi pas un compte générique pour tous ?
- LCEN Décret n°2006-358 du 24 mars 2006:
  Permettre d'identifier toute personne utilisant des réseaux électroniques
- Service personnalisé aux utilisateurs
  - Stockage
  - Profil adapté
  - ...

Quel niveau d'authentification ?

- Authentification basique : 1 token
- Authentification forte : 2 tokens ou plus
- Authentification sans divulgation d'information (vérification par calcul mathématique)

Les techniques – AuthN / AuthZ

- Authentification : « je suis Bart et je le prouve »
  - Une fois prouvé, Confiance en l'identité
- Autorisation : Bart peut il accéder au bar ?
  - Besoin d'un autre type d'information, relative au service
- Autres données : Quel est le bureau de Bart ? Son téléphone ? Son uid ?
- La séparation n'est pas toujours faite...

Plan

- {Qu'est ce que,Pourquoi} authentifier ?
- Les techniques et chiffrements
- Les (in)compatibilités
- Les possibilités
- Les évolutions à venir
- Conclusion

Technique : Flat file Unix 1

- LE fichier `/etc/passwd` :

```
bart:zf67.sLB9vFPE:101:100:Bart Simpson:/home/bart:/bin/bash
```

- A l'époque, un serveur, un utilisateur avec un accès complet, facile à gérer
- Champs séparés par « : », avec le mot de passe chiffré, et les informations relatives au systeme (uid, gid, nom, répertiore de travail, shell)
- Chiffrement « crypt »
- Bien pour un très petit parc, refermé sur lui même.

Technique : Flat file Unix 2

- Déport dans /etc/shadow (accès restreint)
- Chiffrements md5, sha1
- Début de séparation AuthN / AuthZ
- Multiples serveurs, NFS, besoin d'avoir les mêmes uid, de centraliser les mots de passes

Technique : NIS / NIS+ / NYS

- Développé par SUN
- Distribuer les fichiers /etc/passwd (entre autre) à travers le réseau
- Modèle de distribution client/serveur basé sur les RPC
- N'a jamais percé

Technique : Flat file et appels système

- Passer outre les fichiers locaux, rester compatible avec le format
- Appel système « getent »
- Remplacer la lecture de /etc/passwd par getent passwd
- Autres bases fichiers : groups, protocols ...
- Souplesse dans la configuration
- /etc/nsswitch.conf

Technique : LDAP

- LDAP : Annuaire ...d'utilisateurs
- Bart devient maintenant :

```
gecos: Bart Simpson
uidNumber: 101
gidNumber: 100
homeDirectory: /home/bart
loginShell: /bin/bash
uid: bart
userPassord: {CRYPT}zf67.sLB9vFPE
```

- Et pleins d'autres informations : email, télephone...
- On y retrouve au moins le contenu de /etc/passwd

Technique : LDAP

- Modèle Client Serveur
- Appels système getent fréquents : nscd
- Solution très répandu autour de openLDAP
- Stocke à la fois le mot de passe chiffré et les données additionnelles.

Technique : Hesiod

- Projet du MIT parallèle à Kerberos
- Stockage des données dans les DNS des domaines ou sous domaines dans un champs HS (Extension de bind)
- Requêtes DNS des clients pour récupérer des données équivalentes à /etc/passwd
- N'a pas percé

Technique : Kerberos

- Projet du MIT
- Utilisé ou intégré dans toutes les grandes infrastructures (Active Directory en particulier)
- Ne gère que l'authentification des utilisateurs

Le chiffrement

- But : Ne pas laisser lisible le mot de passe
- Méthodes: Crypt / MD5 / SHA1 / SMD5 / SSHA1
- Pas de moyen de décrypter (voulu)
- Vérification :
  - Le mot de passe du client est transmis au serveur si besoin
  - Chiffrement avec la même méthode que celui en base
  - Vérification de la concordance (ou non)

Le chiffrement

- LMHASH : Historique, Chiffrement LAN Manager: très faible sécurité (jusque XP)
- NTLM(v1) : stockage de LMHASH et NTHASH , mais reste impacté par la faiblesse du premier
- NTLMv2 (après NT4SP4) : Sécurisé, contient des informations « externes » : nom du domaine, heure...

Technique : Security Account Manager

- Stockage en tant que base de registre, chiffré par SYSKEY
- Fichier verrouillé
- Contient les Hash des mots de passe

Plan

- {Qu'est ce que,Pourquoi} authentifier ?
- Les techniques et chiffrements
- Les (in)compatibilités
- Les possibilités
- Les évolutions à venir
- Conclusion

Matrice d'authentification

	FlatFile Unix	Kerberos	Ldap	NTLM	Autre
Windows	Non	Oui(*)	Non (Oui)	Oui	Non (Oui)
Linux	Oui	Oui	Oui	Non (Oui)	Oui
Mac OSX	Oui	Oui	Oui	Non (Oui)	?

(in)Compatibilités

- Les données fonctionnelles
  - uid
  - SID
  - HOME
  - Les autorisations
- On en a besoin
- Il faut stocker ces informations de « comptes » même dans des structures centralisées
- Mais: Déporter la problématique Authentification

(in)Compatibilités

- Déployer à grande échelle :
  - Choisir un système d'authentification
    Centre unique pour tous les systèmes
  - Déporter s'il le faut les autorisations sur un autre stockage
  - Cuisine locale acceptée sur les parties fonctionnelles

(in)Compatibilités : samba / winbind

- Bases fonctionnelles : Comment les synchroniser ?
- pam_winbind : auth NTLM/Kerberos
- winbind : table de correspondance Active Directory / Unix : partie idmap
- Stockage : Fichier local, ldap, rid, Active Directory via SFU ou rfc2307,
- Implémentation via nsswitch

(in)Compatibilités : samba / winbind

- Fichier local : pas de service unix centralisé, peu de client (idmap.tbd)
- Plusieurs clients: allocation au premier venu sans synchronisation : bart sur le poste A : uid=100 et homer sur le poste B : uid=100
- RID : bijection connu entre le SID et uid, corrige cette problématique (Un domaine unique)
- GECOS : utilisation de templates

(in)Compatibilités : samba / winbind

- Stockage LDAP externe :
- Utilisation d'une « ou=ldmap »
- L'allocation faite par chaque client, à la volée
- Utilisation de template

(in)Compatibilités : samba / winbind

- Stockage Active Directory
- Utilisation des attributs dans l'AD via SFU
- Utilisation de la compatibilité rfc2307 (2003R2 et +)
- Personnalisation des champs possible

(in)Compatibilités : samba / winbind

- Affiner la solution idmap par rapport à l'existant.
- Une solution idmap est toujours liée à une authentification vers un Active Directory

Technique : Open Directory

- MacOS X : intégration forte openLDAP et MIT Kerberos
- Utilisation de schema spécifique (disponible)
- Outil d'administration graphique
- Stockage des mots de passe dans Kerberos ou dans Apple Password server (stockage SASL)

Technique : MacOSX

- Autres possibilités offertes :
  - Authentification LDAP ou Kerberos
  - Autorisation LDAP
- Liaison avec samba (v3)
  - Intégration classique dans un Active Directory

Technique : Linux

- PAM : séparation authentification, autorisation
- NSS : Récupération des informations fonctionnelles
- Tout ce qui est documenté et techniquement possible peut (sans doute) être réalisé

Technique : Windows

- Seul l'authentification est « délégeable »
  - GINA (jusque XP) : bibliothèque à remplacer (pgina 1, novell)
  - Credential provider (vista, 7): pgina 2 ...
  - Ou approbation Kerberos
- Partie fonctionnelle liée à Active Directory ou à SAM
- pGina : LDAP, Radius, ssh, imap, slashdot, …
- Modification possible...

Technique

- NTLM désactivé dans Vista et successeur
- NTLM utilisé si Kerberos bloqué, dans les « Workgroup » , utilisation d'une IP pour désigner un serveur
- Kerberos présent des Active Directory
  - AES pour 2008 et +
  - RC4-HMAC pour 2003 et +
  - DES désactivé après 2008R2

Technique : SSO

- S'authentifier une seule fois, et ensuite être reconnu sans intervention de l'utilisateur
- Nécessite un stockage de l'information, chiffré
- Kerberos : Informations dans les Tickets
- SSPI (Windows) : Stockage par le système d'un hash
- CAS, CoSign, SAML, Shibboleth ...

Plan

- {Qu'est ce que,Pourquoi} authentifier ?
- Les techniques et chiffrements
- Les (in)compatibilités
- Les possibilités
- Les évolutions à venir
- Conclusion

Les évolutions possibles

- Tout change … GINA par exemple
- Authentifier les utilisateurs via le réseau ? 801.1x
- Authentifier les utilisateurs sur un service web ?
- Passer à des token physique (réduire le social engineering) ?
- Passer à des authentifications fortes ?

Les évolutions possibles : web

- OpenID
  - Participants : Entreprises, web 2.0, ...
  - Serveurs décentralisés (à la facon xmpp ou smtp)
  - Authentification et partage d'informations
  - Adopté peu à peu par les services web (Google, Live, Facebook, Twitter, …) et les fournisseurs de services (Yahoo, Google, Orange, …)
- Liberty Alliance (SAML)
  - Participants : Monde industriel, entreprises, banques...

Les évolutions possibles : web

- Shibboleth
- Participants : monde universitaire
  - Décentralisation des serveurs
  - Utilisation de CAS pour les authentifications

Evolutions possibles : WS Federation

- Windows Cardspace (XP et plus) : Stockage des propriétés relatives à certains sites
- Permet de configurer le type d'informations transmises
- Sans contrainte de token, donc compatible OpenID / SAML

Evolutions possibles : Web

- Oauth
  - En complement de OpenID
  - Utilisé chez Google / Yahoo
  - Donne accès à certaines informations via des fournisseurs tiers

Plan

- {Qu'est ce que,Pourquoi} authentifier ?
- Les techniques et chiffrements
- Les (in)compatibilités
- Les possibilités
- Les évolutions à venir
- Conclusion

Conclusion

- De « Flat file » à Shibboleth .... 40 ans
- Complexification croissante
- Le SSO est une demande forte des utilisateurs
- Sécurité : quel évolution des utilisateurs en 40 ans ?
- Quel est la sureté pratique de la réinitialisation d'un mot de passe d'un compte ?
- Faut il aller vers une authentification forte ?

Conclusion

- Combien de temps pour déployer une solution ?
- Quels solutions sécurisés pour des utilisateurs ultra mobiles ?

Conclusion

- Questions