

Comment intégrer des méthodes d'authentications hétérogènes grâce à Freeradius

Une présentation pour les Linux Days

Par Jérémie MATOS

03.06.2009

 **ELCA** *We make it work.*

Agenda

- Le protocole Radius
- Le serveur FreeRadius
- Cas 1 : Authentification multi-LDAP
- Cas 2 : Migration progressive grâce au mode proxy
- Cas 3 : Méthode d'authentification custom
- Cas 4 : Protéger sa propre application avec FreeRadius



- Protocole client-serveur pour l'authentification
- Standard de sécurité largement adopté par l'industrie
 - Equipements réseau
 - Serveurs d'authentification
- AAA
 - Authentification
 - Autorisation
 - Accounting

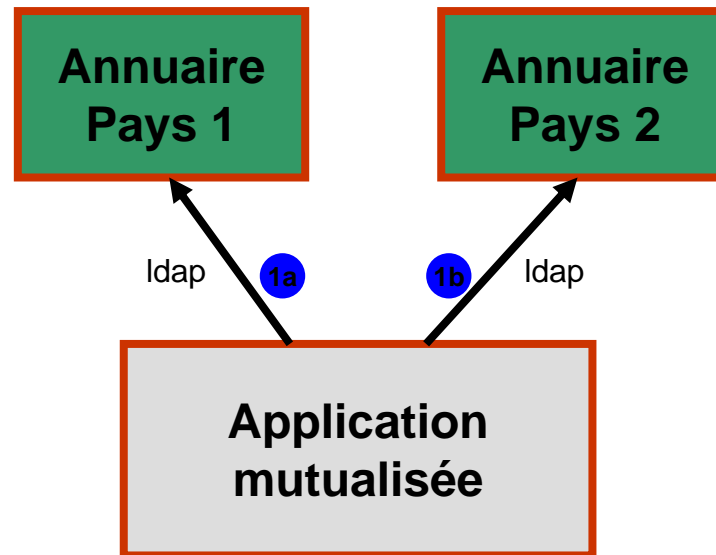
Le serveur Freeradius

- Serveur Radius le plus utilisé dans le monde
 - Plus de 50,000 déploiements dans le monde
 - Estimation : 33% de tous les accès Radius
- OpenSource
 - Tourne sur la plupart des plate-formes liées à Unix
- Robuste
 - Versions 1.0 et 2.0 stables. Actuellement en version 2.1
 - Failover et clustering disponible
 - Utilisé pour des déploiements > 10 millions d'utilisateurs
- Flexible
 - Chaînage de modules par configuration
 - Autorisation
 - Authentification
 - Accounting
 - Notion de virtual host depuis la version 2.0



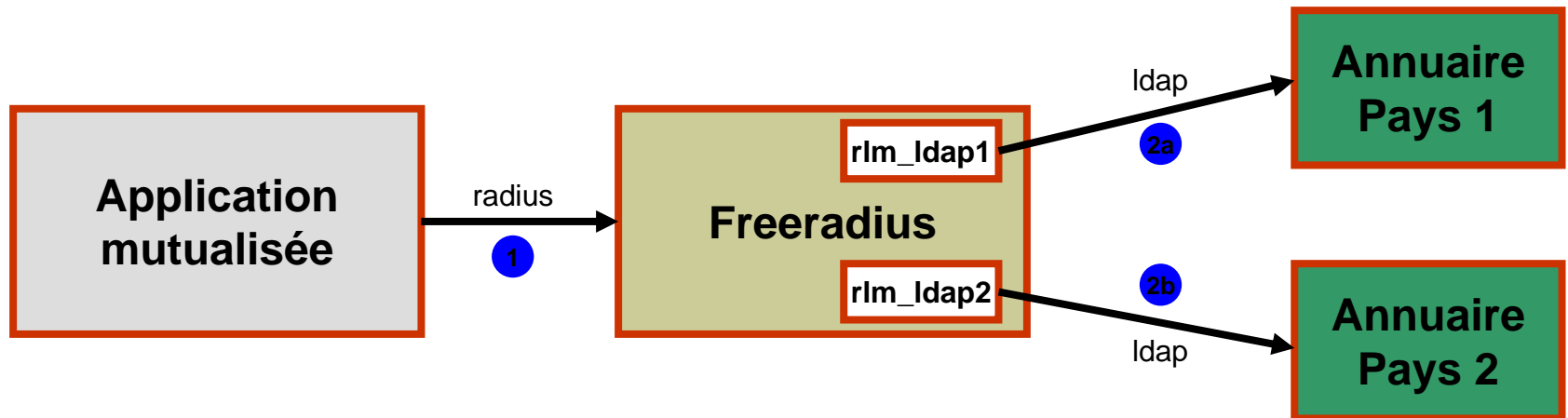
- ▶ Authentication multi-LDAP
- Migration progressive de méthode d'authentification
- Méthode d'authentification custom
- Protéger sa propre application avec FreeRadius

- Authentification par login/mot de passe
- Annuaire indépendants par pays
- On souhaite déployer une seule application métier au niveau mondial



- Réplication d'annuaires
 - Intrusif car on change le contenu d'un annuaire
 - Mise en place pas toujours évidente
 - Très délicat à tester
- Code spécifique
 - S'il s'agit d'une application avec le code source d'authentification disponible
 - Changement de la logique d'authentification pour vérifier dans les deux annuaires
 - Déploiement d'une nouvelle version à chaque changement de l'architecture d'authentification

- Freeradius
 - Chainage de 2 modules rlm_ldap par configuration
 - Chacun pointe vers un annuaire
 - Freeradius balaie les différentes méthodes d'authentification jusqu'à trouver une réponse valide





Authentication multi-LDAP

- ▶ Migration progressive de méthode d'authentification

Méthode d'authentification custom

Protéger sa propre application avec FreeRadius

Migration progressive de méthode d'authentification

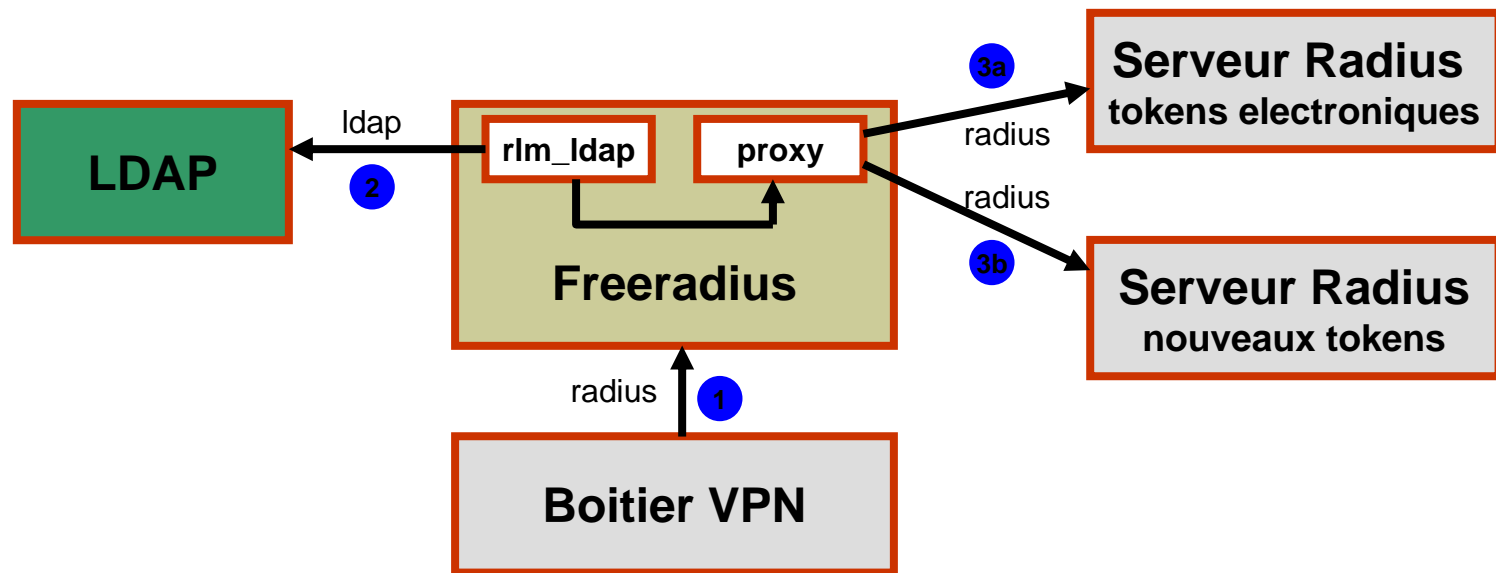
Problématique

- Accès VPN
- Tous les utilisateurs disposent du même token électronique
- Pour des raisons de coût, on souhaite changer de méthode d'authentification
- Pour la continuité du service, on souhaite remplacer les token au compte goutte : quand le token électronique expire
- Comportement impossible à configurer directement sur un équipement standard : si on veut utiliser des serveurs Radius différents, il faut au minimum configurer des services différents

Migration progressive de méthode d'authentification

Solution

- Contraintes
 - Stockage d'un attribut dans le profil utilisateur (e.g LDAP)
- Configuration
 - Récupération de l'attribut grâce à rlm_ldap
 - Ajout d'un suffixe au nom d'utilisateur en fonction de cet attribut
 - Utilisation du mode proxy pour transférer la requête au serveur Radius correspondant au suffixe





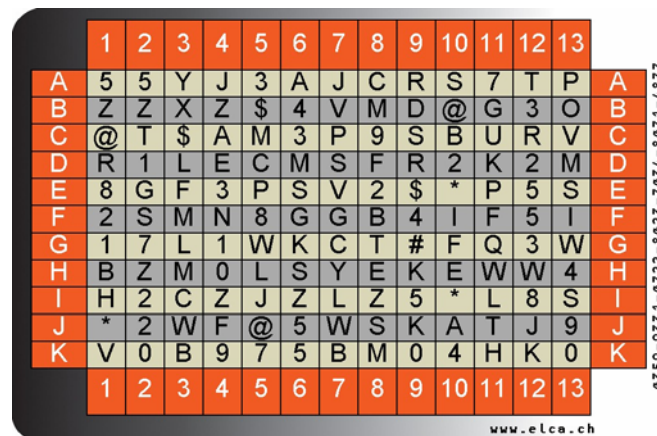
Authentication multi-LDAP

Migration progressive de
méthode d'authentification

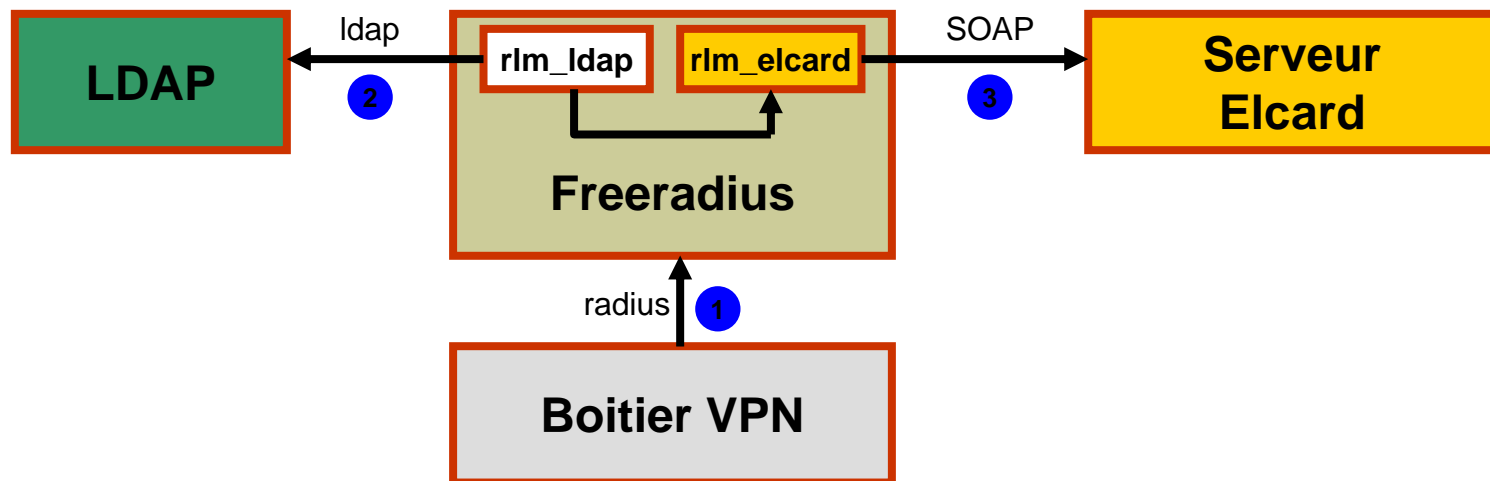
- ▶ Méthode d'authentification
custom

Protéger sa propre application
avec FreeRadius

- On souhaite intégrer une nouvelle méthode d'authentification
 - Qui n'a pas de module FreeRadius existant
 - Dont le fabricant ne propose pas de serveur Radius
- Exemple : carte à chemin Elcard
 - Authentification en deux étapes
 - Login/password dans un annuaire LDAP
 - Puis le serveur Elcard propose un défi et vérifie la réponse correspondante



- Créer un nouveau module pour Freeradius : rlm_elcard



- Implémentation en C
 - Définition des méthodes supportées
 - Méthode instantiate (et detach) pour l'initialisation
 - Méthode authorize
 - On configure le module rlm_ldap pour vérifier le login/password
 - On vérifie dans le module Elcard que le module rlm_ldap a bien validé le login/password
 - Méthode authenticate
 - Récupération d'un défi pour l'utilisateur à partir du serveur Elcard
 - Vérification de la réponse de l'utilisateur sur le serveur Elcard
- Interface avec les web services du serveur Elcard
 - Utilisation de la librairie gSoap



Authentification multi-LDAP

Migration progressive de
méthode d'authentification

Méthode d'authentification
custom

- ▶ Protéger sa propre application
avec FreeRadius

Protéger sa propre application avec FreeRadius

Problématique

- On dispose
 - D'une infrastructure FreeRadius existante
 - Ou éventuellement un autre serveur Radius, que l'on sait faire évoluer simplement vers une plate-forme Freeradius si besoin grâce au mode proxy
- On souhaite protéger une application sensible
 - Au niveau de la session
 - Mais aussi pour la validation de transactions sensibles
 - Codée en Java

- Utilisation de la librairie JRadius
 - Composant serveur
 - Qui permet de coder un module en Java
 - Qui s'intègre à Freeradius par l'intermédiaire du module rlm_jradius
 - Composant client
 - Qui permet d'appeler un serveur Radius pour une demande d'authentification
- Encapsuler dans une méthode doAuthentication() les appels à JRadius
 - L'envoi de l'Acces-Request au serveur
 - La gestion du code de retour "Access-Accept" ou "Access-Reject"
 - Eventuellement la gestion du code de retour "Access-Challenge" si le serveur propose un défi/réponse supplémentaire

Conclusion

Freeradius permet

- De centraliser une infrastructure d'authentification hétérogène
- Rapidement
- De manière très flexible
- Tout en étant robuste

Des questions ?

Merci de votre attention

Pour plus d'informations
veuillez contacter:



ELCA

Jérémy MATOS

Senior Software Engineer

jeremy.matos@elca.ch

Av. de la Harpe 22-24

CH-1001 Lausanne

Tél +41 21 613 21 11

Lausanne | Zürich | Bern | Genève | London | Paris | Ho Chi Minh City

- Freeradius
 - <http://freeradius.org/>
 - Wiki : http://wiki.freeradius.org/Main_Page
- Protocole radius
 - Liens vers les différents RFC : <http://en.wikipedia.org/wiki/RADIUS>
- JRadius : <http://coova.org/wiki/index.php/JRadius>
 - Partie serveur : <http://coova.org/wiki/index.php/JRadius/WithFreeRADIUS>
 - Partie cliente : <http://coova.org/wiki/index.php/JRadius/ClientAPI>
- gSoap : <http://www.cs.fsu.edu/~engelen/soap.html>