Wi-Fi & Eduroam: de la théorie à la pratique

Sommaire:

III. Radius : mise en oeuvre

- C’est quoi radius ?
- Radius: Fonctionnalités attendues
- Choix
- Mise en œuvre
  - ✓ base
  - ✓ proxy
- Exploitation
- Adaptation

Radius : c'est quoi ?...

- Le protocole RADIUS (Remote Authentication Dial-In User Service), mis au point initialement par Livingston, est un protocole d'authentification standard, défini par un certain nombre de RFC.
- Il s'agit du protocole de prédilection des fournisseurs d'accès à internet car il est relativement standard et propose des fonctionnalités de comptabilité permettant aux FAI de facturer précisément leurs clients.
- Basé sur le principe client/serveur, le client radius interroge un serveur RADIUS qui, relié à une base d'identification (base de données, annuaire LDAP, etc.), renvoie sa réponse. L'ensemble des transactions entre le client RADIUS et le serveur RADIUS est chiffré et authentifié grâce à un secret partagé.
- Il est à noter que le serveur RADIUS peut faire office de proxy, c'est-à-dire transmettre les requêtes du client à d'autres serveurs RADIUS "pères" ou "fils".

Radius : Fonctionnalités attendues

- **Le fonctionnement de RADIUS est basé sur un scénario proche de celui-ci:**
  - ✓ **Un utilisateur envoie une requête au NAS afin d'autoriser une connexion**
  - ✓ **Le NAS achemine la demande au serveur RADIUS,**
  - ✓ **Le serveur RADIUS consulte sa ou ses base(s) de données afin de négocier le méthode d'authentification et comparer les sésames de l'utilisateur. Le serveur RADIUS retourne enfin l'une des quatre réponses suivantes:**
    - **ACCEPT : l'identification a réussi,**
    - **REJECT : l'identification a échoué,**
    - **CHALLENGE : le serveur RADIUS souhaite des informations supplémentaires de la part de l'utilisateur et propose un « défi » (en anglais « challenge »),**
    - **CHANGE PASSWORD : le serveur RADIUS demande à l'utilisateur un nouveau mot de passe.**
- **Suite à cette phase dit d'authentification, débute une phase d'autorisation où le serveur retourne les autorisations de l'utilisateur.**
- **Peut également suivre une phase d'accounting (comptabilité) où est enregistré une trace de l'activité de l'utilisateur (login/logout)**

Radius : Fonctionnalités attendues

- Mode proxy (local et forward national => eduroam),
- Support, à minima, des méthodes EAP sécurisées:
  - ✓ Peap,
  - ✓ eap/tls,
  - ✓ eap/ttls.
- Logs:
  - ✓ Locaux,
  - ✓ Déportés via syslog.
- Interfaçage avec bases ldap locales et distantes:
  - ✓ OpenLdap (personnel et invités)
  - ✓ Active Directory (étudiants)

Radius : Choix

- Il existe plusieurs solutions logicielles libres:
  - ✓ FreeRadius (http://www.freeradius.org),
  - ✓ OpenRadius (http://www.openradius.net),
  - ✓ CistronRadius (http://www.radius.cistron.nl),
  - ✓ GNU-Radius (http://www.gnu.org/software/radius),
  - ✓ Yard-Radius (http://sourceforge.net/projects/yardradius),
- « LA » solution la plus aboutie et la plus active est freeRadius,
- Elle est utilisée depuis longtemps par un nombre important d'utilisateurs dont certains fournisseurs internet comme cegetel. (10 millions d'utilisateurs officiels déclarés sur freeradius.org)
- Elle est stable depuis longtemps (officiellement depuis moins d'un an),
- Son chef de projet est très réactif (parfois un peu trop ;-))

Radius : Mise en oeuvre

- **L'installation de freeRadius a été faite à partir des sources (tar.gz):**
  - ✓ **Télécharger l'archive depuis le site officiel (http://www.freeradius.org/getting.html),**
  - ✓ **Décompresser l'archive sous /usr/src par exemple,**
  - ✓ **Créer une arborescence dédiée à chaque instance de freeRadius sous /usr/local,**
  - ✓ **Configurer (./configure –prefix=xxxx) compileret installer freeRadius (make & make install) pour chaque instance,**
  - ✓ **Personnaliser chaque instance en fonction de sa fonction (proxy/backend), de sa source d'authentification (locale,ldap,...) et des méthodes d'authentification à supporter,**

Radius : Mise en oeuvre - base

- Pour la première instance (base du personnel), les personnalisations suivantes ont été faites:

- Fichier radiusd.conf:
  - ✓ Changement de l'utilisateur/groupe par défaut (sécurité):
    - Créer un groupe et un utilisateur ayant les droits sur l'arborescence créée.
  - ✓ Changement du port par défaut (toutes les instances doivent avoir leurs propres ports):
    - Ne pas utiliser des ports consécutifs car FR a besoin de plusieurs ports tcp.
  - ✓ L'attribut « reject-delay » a été positionné à '2‘,
  - ✓ L'attribut « proxy-requests » a été positionné à 'no‘:
    - On est dans un contexte de base arrière, dans le cas du proxy, on le positionnera, bien évidemment, sur 'yes‘.
  - ✓ Modules mschap et ldap activés et configurés:
    - Définir le host, manager+pwd,basedn, filtres (base-filter et filter),
    - Définir les propriétés tls si le serveur ldap n'est pas sur le même host,
    - Définir le nb de connexions et les timeout.
  - ✓ Rubrique 'authorise' paramétrée avec mschap, eap, files et ldap,
  - ✓ Rubrique 'authenticate' paramétrée avec mschap, ldap et eap,
  - ✓ Rubrique 'post-proxy' paramétrée avec eap.

Radius : Mise en oeuvre - base

- **Fichier clients.conf:**
  - ✓ **Ajout d'une rubrique 'client' ayant l'ip, le secret partagé et le nom du NAS WiFi:**

```
client A.B.C.D {
        secret = mon-secret-partage
        shortname = aruba5000-sm1
}
```

  - ✓ **Configurer la rubrique 'client' pour le proxy (localhost ou 127.0.0.1 si il est hébergé sur le même host:**

```
Client 127.0.0.1 {
        secret          = secret
        shortname = proxy-local
}
```

Radius : Mise en oeuvre - base

- **Fichier eap.conf:**
  - ✓ **Configurer le 'default-eap-type' sur ttls,**
  - ✓ **Configurer les modules tls (certificats), ttls et peap (ne pas oublier mschapv2),**
  - ✓ **Exemple:**

```
eap {
        default_eap_type = leap
        timer_expire     = 60
        ignore_unknown_eap_types = no
        cisco_accounting_username_bug = no
    tls {
        private_key_file = /usr/ssl/private/auth.u-bourgogne.fr.key
        certificate_file = /usr/ssl/certs/auth.u-bourgogne.fr.crt
        CA_file = /usr/ssl/certs/cachain.txt
        dh_file = /etc/1x/DH
        random_file = /etc/1x/random
        include_length = yes
    }
    peap {
        copy_request_to_tunnel = yes
        default_eap_type = mschapv2
    }
    mschapv2 {
    }
```

Exemple de configuration du proxy : (etc/raddb/proxy.conf)

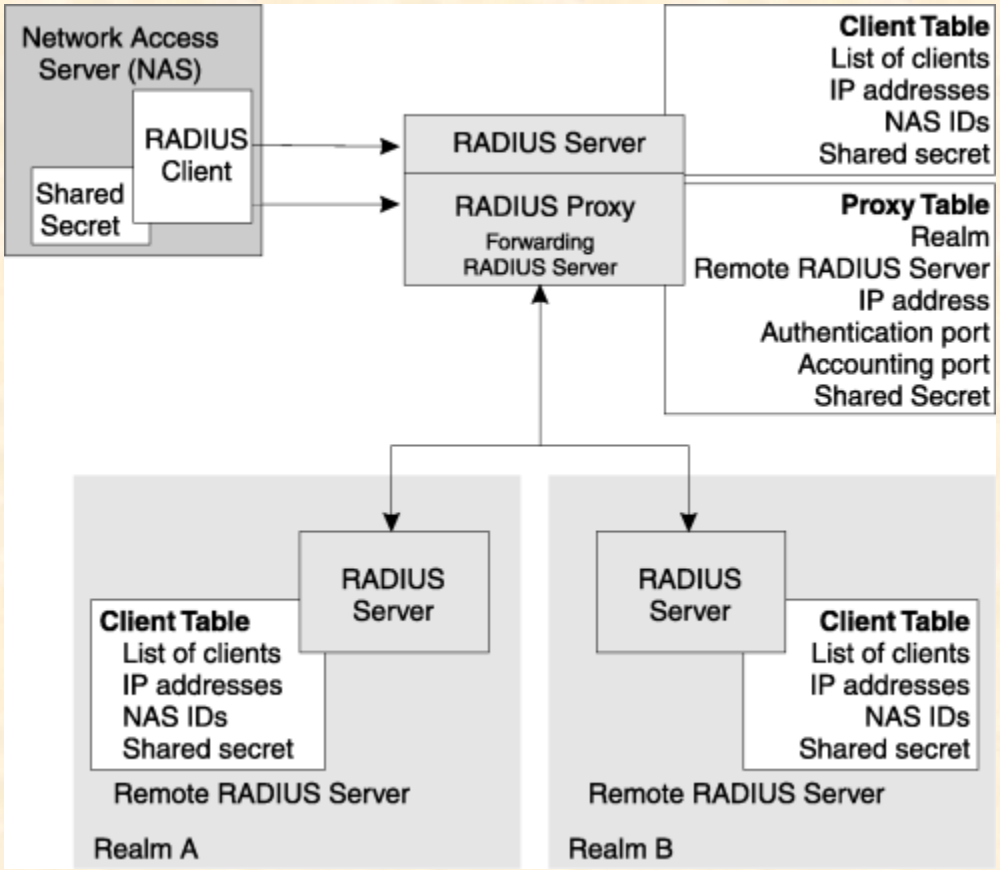
```
Realm  u-bourgogne.fr {
        type            = radius
        authhost        = localhost:3812
        accthost        = localhost:3813
        secret          = xxxxxxxx
        nostrip
}

realm etu.u-bourgogne.fr {
        type            = radius
        authhost        = 172.17.128.145:1812
        accthost        = 172.17.128.145:1813
        secret          = xxxxxxxxx
}
realm dijon.iufm.fr {
        type            = radius
        authhost        = localhost:2812
        accthost        = localhost:2813
        secret          = xxxxxxxx
}
```

```
realm NULL {
        type            = radius
        authhost        = localhost:3812
        accthost        = localhost:3813
        secret          = xxxxx
        nostrip
}
realm DEFAULT {
        type            = radius
        authhost        = rad1.eduroam.fr:1812
        accthost        = rad1.eduroam.fr:1813
        secret          =  xxxxxxxxx
        nostrip
}
realm DEFAULT {
        type            = radius
        authhost        = rad2.eduroam.fr:1812
        accthost        = rad2.eduroam.fr:1813
        secret          = xxxxxxxxx
        nostrip
}
```

Radius : Exploitation

- Il a été mis en place un proxy local (auth.u-bourgogne.fr) qui abrite plusieurs « realms »:
  - ✓ u-bourgogne.fr
  - ✓ etu.u-bourgogne.fr
  - ✓ dijon.iufm.fr
- En fonction du realm saisi dans le client (identifiant@realm.fr), l'authentification est soumise à l'un des radius locaux qui lui-même va vérifier les sésames sur l'un des annuaires locaux,
- Si le realm fourni est inconnu en local, on envoie la requête au proxys nationaux « arredu ».



Radius : Adaptations

- Il a également été fait quelques petites adaptations du schéma ldap afin de pouvoir « remonter » certains attributs ldap par radius,
- Ces « adaptations » sont faites via le fichier 'ldap.attrmap‘,
- Dans la ligne suivante, on remonte un attribut radius nommé 'Filter-Id' qui contient la valeur de l'attribut ldap 'eduPersonPrimaryAfilliation' ; ceci permet de positionner, via le contrôleur Aruba et les rôles précedemment créés, les utilisateurs dans un vlan donné en fonction de leur statut vis à vis de l'Université (etudiants, personnels, invités):

replyItem Filter-Id eduPersonPrimaryAffiliation

Merci de votre attention…

Questions – réponses