eole-radius - Tâche #36376

Scénario # 35809 (Terminé (Sprint)): EOLE 2.10 : refonte de eole-freeradius

Exploration des implémentations possibles

08/01/2025 09:49 - Benjamin Bohard

Statut:	Fermé	**Début:**	01/10/2022
Priorité:	Normal	**Echéance:**	
Assigné à:	Benjamin Bohard	**% réalisé:**	100%
Version cible:	Livraison Cadoles - MEN 31/12/2024 (100)	**Temps estimé:**	0.00 heure
		Temps passé:	0.00 heure

Description

Historique

#1 - 08/01/2025 09:49 - Benjamin Bohard

- Statut changé de Nouveau à En cours

#2 - 08/01/2025 11:12 - Benjamin Bohard

En repartant des constats[1] des précédentes études, le travail d'implémentation s'efforce de répondre aux problématiques suivantes :
- la configuration proposée par EOLE ne doit pas empêcher une configuration alternative et vice et versa (éviter l'écrasement des fichiers notamment ;
- la configuration proposée par EOLE doit permettre le fonctionnement des cas d'usage identifiés ;

Plusieurs difficultés ont été identifiées dans la mise en œuvre d'une configuration adéquate

Isolation de la configuration

Pour les cas où la configuration proposé par EOLE ne convient pas à l'infrastructure de déploiement, il serait utile de pouvoir basculer sur un mode manuel.
Une variable de configuration permet de déclarer le mode voulu. Dans le cas manuel, la configuration doit être effectuée par l'administrateur en utilisant les mécanismes classiques de freeradius (déclaration du serveur virtuel, configuration et activation des modules).
Dans le cas de la configuration proposée par EOLE, le choix des modules serait fixe et leur configuration se ferait à la marge (authentification des clients par certificat ou non, ntlm_auth ou winbind, etc.).

Pour les modules adoptant la syntaxe classique de freeradius (ceux gérés dans les dossiers mods-available et mods-enabled), l'isolation est faisable en utilisant des dossiers séparés par mode, par exemple. Une approche similaire peut être utilisée également pour le dossier spécial mods-config. Le dossier policy.d, par contre, ne semble pas pouvoir être dupliqué. Il faut également noté que quelques configurations sont des liens symboliques vers d'autres dossiers.

Un autre point est la distinction de configurations qui pourraient être transverses aux deux modes, comme la déclaration des NAS, les filtres utilisateurs ou la configuration LDAP.

Modularité

Un test de configuration modulaire (activation des modules à la demande) s'est avérée trop complexe à consolider, au vu des dépendances entre modules.
L'idée était d'étendre le cas d'usage identifié avec l'activation de l'accounting à la demande.
Dans la configuration actuelle, l'accounting est identifié en opposition au mode 802.1x. Dans le principe de fonctionnement de freeradius, il semble plus justifié d'implémenter l'accounting, non pas en opposition avec le reste des fonctionnalités, mais comme une option qui peut être désactivée[2].
De la même manière, permettre le choix des modules utilisés pour l'autorisation et l'authentification pourrait faciliter l'adaptation du fonctionnement du service par l'administrateur. Cependant, l'interdépendance des modules pour les phases d'autorisation et d'authentification rend difficile la présentation des variables (dans l'application de configuration du module notamment).
Dans le cas du mode manuel, où la configuration de certains modules comme le LDAP via l'application de configuration du module pourrait être pertinente, il est compliqué de s'assurer que ces valeurs soient utilisées tout en obligeant à faire le reste de la configuration à la main.

Comme pour le point précédent, certains morceaux de configuration se dégagent du lot et semblent pouvoir être proposés à la configuration indépendemment du mode, comme la déclaration des NAS et les filtres utilisateurs.

Cas d'usages

Les cas d'usage identifiés sont l'autorisation auprès de l'annuaire (maintenant sous forme AD), l'activation de l'accounting et l'identification par certificat des deux extrêmités.

Les évolutions de la configuration actuelle qui semblent nécessaires sont :
- implémenter l'accounting comme une option plutôt qu'un mode ;
- conditionner les modules selon la volonté d'identifier par certificat (elle n'est possible qu'avec un nombre restreint de modules d'autorisation) ;
- donner le choix sur l'utilisation de ntlm_auth ou winbind pour la connexion à l'annuaire.

Ces cas d'usage ne déterminent cependant pas si il faut limiter le nombre de modules chargés au strict minimum (au risque d'avoir une configuration adaptée à un seul environnement d'exécution) ou si il faut permettre une certaine liberté d'activation et désactivation.

En résumé

La principale difficulté pour l'intégration de freeradius, comme rapportée dans les précédentes études, est la méconnaissance précise des cas d'usage et la complexité à implémenter une configuration flexible (sans pouvoir la tester convenablement).

Pour l'instant, les efforts ont porté sur la mise en œuvre d'une certaine flexibilité mais la formule adéquate n'a pas encore été trouvée. La cohérence globale de la configuration, avec les dépendances complexes, est le point bloquant pour l'instant.

---

[1] - la configuration est fortement dépendante de l'infrastructure, hors du scope d'EOLE, et les cas d'usage sont peu ou pas connus ;
- la configuration de freeradius mixe des fichiers explicitement chargés avec des fichiers implicitement chargés, des interdépendances entre ces fichiers existants par ailleurs.

[2] la finalité première de l'accounting est de pouvoir facturer le temps de connexion et le consommation de données. Le mécanisme prend des ressources inutilement si ces mesures ne sont pas pertinentes.

#3 - 09/01/2025 12:35 - Laurent Gourvenec

Suite à l'analyse de Benjamin, il semble que <b>tous<b> les fichiers de conf pour freeradius peuvent aller dans /etc/freeradius, ce qui facilite beaucoup de choses.
Voici le modèle visé actuellement :
Coté gen_config, dans l'onglet freeradius on aurait

```
Choix modèle (liste déroulante, noms à revoir) :
  - A
  - B
  - C
Variables pour l'annuaire
Variables pour le NAS
Variables pour d'autres contraintes externes communes à la plupart des modèles
```

Coté dico, déclaration des noms de modèle, ajout de contraintes sur les variables pour les désactiver selon le modèle choisi et pour désactiver la génération de fichiers templatisés. Par exemple, un modèle n'utilisant pas l'authentification LDAP (exemple bidon) implique qu'il ne faut pas templatiser le fichier de config ldap pour freeradius. Autrement, freeradius ne démarrera pas.

Coté templates, une poignée de fichiers pour des fichiers ayant besoin de templatisation. Ces fichiers représentent les "contraintes externes".

Coté fichiers statiques, c'est à dire le très gros de la conf, on partirait sur

```
/usr/share/eole/freeradius/
 - A/modules.enabled/
```

```
- A/conf.xml (nom bidon)
- B/ ...
- C/ ...
- README (avec un lien vers la doc pour ajouter un modèle)
- _base/ ... Avec les fichiers de conf de base de freeradius car ils sont plutôt complet et surtout commentés
! Pratique pour ajouter son propre modèle
```

Ces fichiers sont copiés en post-template dans /etc/freeradius

Pour ajouter un modèle, il faut donc ajouter un répertoire dans /usr/share/eole/freeradius et ajouter un dico local pour déclarer le modèle et écraser les contraintes.

#4 - 23/01/2025 09:01 - Benjamin Bohard

- Statut changé de En cours à À valider

#5 - 12/02/2025 17:31 - Emmanuel GARETTE

- Statut changé de À valider à Résolu

#6 - 29/03/2025 17:48 - Joël Cuissinat

- Statut changé de Résolu à Fermé

- % réalisé changé de 0 à 100

- Restant à faire (heures) mis à 0.0