EOLE AD DC - Tâche #37547

Scénario # 37406 (Nouveau): Porter Sphynx en 2.11

etude

07/07/2026 10:25 - Ludwig Seys

Statut:	En cours	**Début:**	07/07/2026
Priorité:	Normal	**Echéance:**	
Assigné à:	Ludwig Seys	**% réalisé:**	90%
Version cible:	Carnet Cadoles - MEN	**Temps estimé:**	0.00 heure

Description

Historique

#1 - 07/07/2026 10:26 - Ludwig Seys

- Statut changé de Nouveau à En cours

#2 - 07/07/2026 13:57 - Ludwig Seys

- % réalisé changé de 0 à 90

1. Conclusions étude Sphynx 2.11

Attention décision à prendre !! (en fin de ce pavé)

2. corosync 3.1.7 -> 3.1.9

Aucun changement bloquant identifié à ce stade.

2.1 Les changements relevés concernent principalement :

- le déplacement de la policy D-Bus `corosync-notifyd` vers `/usr/share/dbus-1/system.d` ;
- la suppression de l'ancien conffile `corosync-notifyd` ;
- des ajustements de tests Debian/Ubuntu ;
- un correctif de fallback `knet` ;
- des corrections mineures upstream.

2.2 Actions :

- vérifier que Sphynx ne référence pas en dur l'ancien chemin D-Bus de `corosync-notifyd` ;
- vérifier que `corosync-notifyd` n'est pas templatisé ou post-traité par EOLE ;
- vérifier que la configuration générée ne contient pas de timers ping invalides ;
- vérifier qu'un reload corosync ne remonte pas d'erreur crypto/authkey ;
- vérifier `corosync-quorumtool ~~s` sur Sphynx 2.11 ;~~
vérifier le cluster avec `corosync-cfgtool -s` et `systemctl status corosync pacemaker`.

3. FRR 8.4.4 -> 10.5.1

La montée de version est significative.

3.1 Les changements relevés concernent principalement l'empaquetage :

- suppression d'anciens traitements sysvinit/quagga/zebra ;
- réécriture des maintainer scripts ;
- ajustements de logs Ubuntu ;
- désactivation du support Lua côté Ubuntu ;
- création/usage de `/var/lib/frr`.

3.2 Actions :

Aucun changement bloquant n'est identifié à ce stade dans le changelog Debian/Ubuntu pour un usage Sphynx standard, sous vérification des points suivants :

- vérifier si Sphynx utilise encore un script init SysV ou uniquement systemd ;
- vérifier la présence et l'état du service `frr.service` ;
- vérifier le fichier `/etc/frr/daemons` généré par EOLE ;

- vérifier le fichier `/etc/frr/frr.conf` généré par EOLE ;
- vérifier les commandes `vtysh` utilisées dans les scripts EOLE ;
- vérifier que le support Lua n'est pas utilisé ;
- vérifier que le module IRDP n'est pas utilisé ;
- vérifier les droits et l'existence de `/var/lib/frr` ;
- vérifier les droits et la rotation des logs dans `/var/log/frr` ;
- vérifier que les daemons utilisés démarrent, par exemple `zebra`, `staticd`, `ospfd`, etc.

4. strongSwan 5.9.13 -> 6.0.4

La montée de version est sensible pour Sphynx.

4.1 Changements identifiés

Les changements Debian/Ubuntu montrent plusieurs points de vigilance :

- le métapaquet Debian strongswan bascule vers strongswan-swanctl ;
- Sphynx utilise encore le modèle historique strongswan-starter / ipsec.conf ;
- strongswan-charon et charon-systemd sont désormais en conflit ;
- plusieurs plugins changent de paquet, notamment kdf, xcbc et fips-prf vers libstrongswan-extra-plugins ;
- les plugins crypto par défaut évoluent en strongSwan 6.0 ;
- les comportements autour des certificats/CRL doivent être validés.

4.2 Retours observés sur Sphynx 2.11

Des blocages sont déjà identifiés à ce stade, notamment via les tests Jenkins et l'instance cassée.

Sur Sphynx 2.11, strongswan-starter est bien présent et activé :

```
strongswan-starter.service - strongSwan IPsec IKEv1/IKEv2 daemon using ipsec.conf
Loaded: loaded (/usr/lib/systemd/system/strongswan-starter.service; enabled; preset: enabled)
Active: inactive (dead)
```

Le service a démarré charon, puis a été stoppé proprement pendant l'instance :

```
charon started
SIGINT received, shutting down
ipsec starter stopped
strongswan-starter.service: Deactivated successfully
```

On trouve également des fichiers liés à charon-systemd dans les chemins Creole/EOLE :

```
/etc/strongswan.d/charon-systemd.conf
/usr/share/eole/creole/distrib/charon-systemd.conf
/var/lib/creole/charon-systemd.conf
```

Il faut donc clarifier le modèle réellement attendu pour Sphynx 2.11 :

- soit conserver ipsec.conf + strongswan-starter ;
- soit migrer vers swanctl / charon-systemd.

Pour un portage minimal, le modèle ipsec.conf + strongswan-starter semble à conserver.

4.3 Blocage actuel de l'instance

L'instance Sphynx 2.11 échoue dans le posttemplate ARV :

```
run-parts: executing /usr/share/eole/posttemplate/00-arv instance
Ajout du serveur sphynx...
IndexError: list index out of range
Exception: error in add_cred_auth: list index out of range
run-parts: /usr/share/eole/posttemplate/00-arv exited with return code 1
Erreur : posttemplate
```

Le crash se produit lors de l'ajout du serveur dans ARV, au moment de l'extraction du sujet du certificat :

```
arv/db/edge.py, add_credential_auth()
cert.get_subject(cert=credential)
creole/cert.py, get_subject()
regexp_get_subject.findall(ret)[0]
```

ça nous oriente vers un problème de compatibilité autour de la génération/lecture des certificats avec l'environnement 2.11, peut-être lié aux changements Python/OpenSSL/strongSwan/certificats.

4.4 Actions

Vérifier explicitement les dépendances du paquet EOLE.

Si le modèle ipsec.conf + strongswan-starter est conservé, les dépendances EOLE doivent sans doute inclure explicitement :

- strongswan-starter ;
- strongswan-charon ;
- libstrongswan ;
- libstrongswan-standard-plugins ;
- libstrongswan-extra-plugins si des plugins déplacés sont nécessaires, notamment kdf, xcbc, fips-prf ou autres.

Il faut également :

- vérifier que charon-systemd n'est pas installé, activé ou configuré à la place de strongswan-starter ;
- vérifier les fichiers générés par Creole autour de charon-systemd ;
- vérifier les plugins strongSwan réellement chargés au démarrage ;
- analyser les logs Jenkins en échec ;
- reproduire localement les commandes Jenkins sur Sphynx 2.11 ;
- investiguer le crash ARV dans add_credential_auth() / cert.get_subject().

5. conclusion

À ce stade, corosync ne présente pas de changement bloquant identifié et FRR nécessite surtout une validation de compatibilité des fichiers générés et des commandes vtysh.

Le point le plus sensible est strongSwan, car la version 6.0 modifie le modèle de paquets, les plugins et le positionnement entre strongswan-starter, swanctl et charon-systemd.

Le blocage immédiat de l'instance Sphynx 2.11 se situe toutefois dans ARV, lors de l'extraction du sujet du certificat dans cert.get_subject(). Les modifications les plus importantes risquent donc de concerner :

- arv, pour corriger la compatibilité certificat/Python/OpenSSL ;
- conf-sphynx, pour ajuster les dépendances et la configuration strongSwan générée.

6. Choix à faire :

La montée vers Ubuntu 26.04 / strongSwan 6.0 pose un choix d'orientation.

Le modèle historique utilisé par Sphynx repose sur :

- ipsec.conf ;
- ipsec.secrets ;
- strongswan-starter ;
- l'interface legacy stroke.

Le modèle recommandé par strongSwan 6.0 et suivi par le packaging Debian/Ubuntu actuel repose plutôt sur :

- swanctl.conf ;
- swanctl ;
- charon-systemd ;
- l'interface vici.

Pour suivre l'usage recommandé avec Ubuntu 26.04, il faudrait donc migrer Sphynx vers swanctl / charon-systemd.

Conserver ipsec.conf + strongswan-starter reste possible comme choix de compatibilité EOLE à court terme, mais ce choix maintient Sphynx sur un modèle historique/déprécié et impose un travail à suivre sur les dépendances côté EOLE.

6. 1 Scénarios possibles

6.1.1 Scénario A — Migration recommandée

Migrer Sphynx vers le modèle actuel strongSwan :

- générer une configuration `swanctl.conf` ;
- migrer les secrets depuis `ipsec.secrets` vers le format attendu par `swanctl` ;
- utiliser `swanctl --load-all` ou les mécanismes systemd associés ;
- utiliser `charon-systemd` ;
- valider que `strongswan-starter` n'est plus nécessaire ;
- adapter les scripts EOLE qui utilisent `ipsec`, `ipsec status`, `ipsec reload`, etc. ;
- vérifier les plugins nécessaires côté `libstrongswan-*`.

6.1.2 Scénario B — Compatibilité court terme

Conserver temporairement le modèle historique :

- conserver `ipsec.conf` ;
- conserver `ipsec.secrets` ;
- dépendre explicitement de `strongswan-starter` ;
- dépendre explicitement de `strongswan-charon` ;
- éviter l'installation/activation de `charon-systemd` ;
- vérifier que le plugin `stroke` et les plugins déplacés sont bien disponibles ;
- documenter que ce choix s'écarte du modèle recommandé pour strongSwan 6.0.