



Séminaire

22-23 Novembre 2006



Sphynx



Historique (1)

- Sphynx 1.0
 - Sortie debut 2003
 - Même base que l'Amon-1.5
 - Utilisation freeswan-1.99
 - Générateur de configuration vpn en XML
 - Conforme au cahier des charges AGRIATES
 - possibilité de ne monter qu'un seul tunnel avec un Amon



Historique (2)

- Sphynx-1.1
 - Sortie debut 2004 en beta.
 - Basé sur une Mdk 9.1.
 - Au départ, pas d'évolution fonctionnelles puis,
 - Intégration de la gestion multi-tunnels vers Amon.
 - Passage à super-freeswan.
 - Prise en compte du nat transversal.
 - gestion des adresses privées entre un Amon et un routeur.
 - Ajout des produits permettant de gérer la haute disponibilité.
 - Ajout d'outils permettant de gérer le routage dynamique.
 - Évolutions noyau.



Bilan actuel

- Une cinquantaine de Sphynx déployés.
- Produit vieillissant :
 - problème de renouvellement des tunnels bloqués.
 - Lenteurs lors de la génération des fichiers de configuration (lié au nombre de tunnels).
 - Problèmes de stabilité ?
 - Plus de possibilités d'évolution.
 - Plus de mise à jour de sécurité.





Sphynx NG₍₁₎

- Version 1.98 livrée en juin 2006 en version Alpha
 - Basée sur une distribution Sourcemage
 - Passage sur un noyau 2.6
 - Couche ipsec intégrée au Noyau.
 - Abandon de Freeswan au profit de Strongswan.
 - Nouveau système de templates Eole.
 - Niveau fonctionnel quasi identique.
- Actuellement, finalisation de la migration sur Ubuntu.



- Strongswan ⁽¹⁾
 - Projet basé sur les dernières versions de freeswan
 - Intègre la gestion des certificats X509 v3
 - Support des tokens, smartCards ...
 - Meilleure prise en compte des crls.
 - Récupération et cache local des crls via http et ldap.
 - Support complet du protocole ocsf (Online Certificate Status Protocol).
 - Récupération automatique des crls (si champ URI présent dans le certificat).



- Strongswan (2)
 - Gestion de la politique de sécurité ipsec étendue pouvant-être basée sur :
 - des CA intermédiaires et des wildcards.
 - des groupes avec des attributs de certificats.
 - des ports ou des groupes de ports.
 - des ip ou des plages d'ip.
 - Insertion et suppression automatiques des règles de firewall.
 - Meilleure prise en compte des algorithmes de crypto
 - aes (jusqu'à 512), blowfish, towfish, serpent ...
 - Meilleur support du nat traversal.



Sphynx NG₍₄₎

- Strongswan ⁽³⁾
 - Intégration complète du Dead Peer Detection.
 - Commande permettant le démarrage et renouvellement des connexions plus rapidement.
 - Gestion statique et dynamique (en pull ou en push) des ip virtuelles.



- Évolutions

- Au niveau de la configuration :

- Intégration et gestion de la haute disponibilité (sans répartition de charge).
 - Gestion des vlans, des ip virtuels, des routes.
 - Intégration de la configuration du routage dynamique.

- Au niveau du générateur :

- Refonte totale de l'interface.
 - Interaction beaucoup plus forte avec Zephir.
 - Passe par une redéfinition préalable des besoins.



Sphynx NG₍₆₎

- Autres évolutions possibles :
 - Prise en charge ipv6.
 - Prise en charge mpls.
 - Qualité de service.
 - ...





Nouveaux dictionnaires⁽¹⁾

Fichier Mode

Eole

- general
- reseau
- vpn-pki

Nom de la machine	<input type="text" value="sphinx"/>	Prec	Def
Nom de domaine privé du reseau local	<input type="text" value="monreseau.lan"/>	Prec	Def
Nom de domaine académique sans le .fr (ex : ac-dijon)	<input type="text"/>	Prec	Def
adresse serveur NTP	<input type="text" value="pool.ntp.org"/>	Prec	Def
serveur de mise à jour	<input type="text" value="eole.ac-dijon.fr"/>	Prec	Def
serveur de référence pour les RPMs Eole/Amon	<input type="text" value="eole.ac-dijon.fr"/>	Prec	Def
Activation de la haute disponibilité oui/non	<input type="text" value="non"/>	Prec	Def





Nouveaux dictionnaires⁽²⁾

- Fichier dictionnaire en XML

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```
<creole>
```

```
<files>
```

```
  <file name='/etc/hosts'/>
```

```
  <file name='/etc/sudoers' mode="0440"/>
```

```
<file filelist='dyn_route' name='/etc/quagga/zebra.conf'/>
```

```
<file filelist='dyn_route' name='/etc/quagga/debian.conf'  
source='quagga.conf'/>
```

```
</files>
```





Nouveaux dictionnaires⁽³⁾

```
<variables>  
  <family name='general'>  
    <variable name='nom_machine' type='string'  
description='Nom de la machine' >  
      <value>sphynx</value>  
    </variable>  
    <variable name='nom_domaine_local' type='string'  
description='Nom de domaine privé du reseau local' >  
      <value>monreseau.lan</value>  
    </variable>
```





Nouveaux dictionnaires⁽⁴⁾

```
<constraints>
  <fill name='calc_network'
target='adresse_network_int0'>
    <param type='eole'
name='ip'>adresse_ip_int0</param>
    <param type='eole'
name='netmask'>adresse_netmask_int0</param>
  </fill>
  <check name='obligatoire' target='nom_machine'/>
  <check name="valid_enum" target='haute_dispo'>
    <param>['non','maitre','esclave']</param>
  </check>
</constraints>
```



Eole Nouveaux dictionnaires (5)

Generate_xml
RESET

FILE
VARIABLE
FILELIST
FAMILY
FUNCTION
GROUP

FUNCTION

valid_ip

CHOOSE A FUNCTION
check-valid_ip
Valider

FUNCTION
Identifiant 1
[supprimer la fonction](#)
name : valid_ip type : function
source : None

Gestion des Targets

Liste des targets
target : adresse_network_int2 [Enlever le target](#)

Ajouter un target:

VARIABLE	<input type="text" value="adresse_netmask"/>	Valider	FAMILY	<input type="text" value="vide"/>	Valider
FILE	<input type="text" value="/sbin/lance.firewa"/>	Valider	FILELIST	<input type="text" value="vide"/>	Valider

Gestion des Parametres

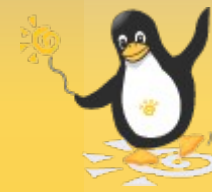
Liste des parametres :

Ajouter un parametre:

Variable	VARIABLE	<input type="text" value="adresse_netmask"/>	NOM	<input type="text" value="Nom du param"/>	Valider
Constante	VALEUR	<input type="text" value="Value du param"/>	NOM	<input type="text" value="Nom du param"/>	Valider

SOURCE
 Valider

Rechercher : grou Respecter la casse





Nouveaux dictionnaires (6)

http://localhost:7080/db/variable/1

WebSVN - Eole - Révision 5030 - /... Cours de Python : 6) Fichiers www.kieranholland.com Meet St... μTidylib, the TidyLib Python wrap...

[Generate xml](#)
[RESET](#)

- FILE
- VARIABLE
- FILELIST
- FAMILY
- FUNCTION
- GROUP

VARIABLE

adresse_netmask_vlan_int0
nom_domaine_local
adresse_network_int2
adresse_network_int0
adresse_network_int1
url_crl1
serveur_ref
vpn_mode
url_crl2
adresse_broadcast_int1
adresse_broadcast_int0
nom_machine
adresse_netmask_vlan_int1
nom_machine_esclave
ssh_int0
ssh_int1
adresse_broadcast_vlan_int0
adresse_ip_vlan_int0
alias_ip_int1
alias_ip_int0
adresse_ip_int2
int0_method
adresse_ip_maitre_int0
adresse_ip_maitre_int2
nom_machine_maitre
serveur_maj

VARIABLE

Identifiant 1

[supprimer variable](#)

name : adresse_netmask_vlan_int0 **family :** [reseau](#)
value : [u'255.255.255.0'] **type :** netmask **description :** Masque de sous reseau de l'interface dans ce vlan
multi : False **hidden :** False **help :** None

Fonction de controle

Liste des fonctions associees

Ajoutez :

FUNCTION CHECK valid_ip ajouter
FUNCTION CONDITION hidden_if_not_in ajouter
FUNCTION FILL calc_network ajouter

NAME	FAMILY	DESCRIPTION	VALUE
adresse_netmask_v	reseau	Masque de sous re	[u'255.255.255.0']
TYPE	HIDDEN	MULTI	HELP
ip	False	False	

Rechercher : grou Respecter la casse





Nouveaux dictionnaires⁽⁷⁾

- Exemple d'un fichier de configuration :

```
%if %%ssh_int0 == "oui"
```

```
%for %%reseau_ssh0 in %%ip_ssh_int0
```

```
%if %%reseau_ssh0.netmask_ssh_int0 == "255.255.255.255"
```

```
sshd:%%reseau_ssh0
```

```
%else if %%reseau_ssh0.netmask_ssh_int0 != "255.255.255.255"
```

```
sshd:%%reseau_ssh0/%%reseau_ssh0.netmask_ssh_int0
```

```
%end if
```

```
%end for
```

```
%end if
```





Merci de votre attention.

