**2012**



**08/10/2012**

sTunnel 4.53 - Dossier d’installation

Version 1

**Informations sur le document**

|  |  |
| --- | --- |
| Auteur | Guillaume MANDIER |
| Description | Dossier d’installation de Stunnel version 4.53 |

**Revue du présent document**

| Nom | Fonction | Date |
| --- | --- | --- |
|  |  |  |

**Validation du présent document**

| Nom | Fonction | Date |
| --- | --- | --- |
|  |  |  |

**Liste de Diffusion**

| Nom | Fonction |
| --- | --- |
|  |  |
|  |  |

**Versions**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Responsable | Nature des modifications |
| V1.0 | 08/10/2012 | G.MANDIER | Création du document |
| V1.1 | 07/07/15 | H.TAI | Précision sur la configuration et les certificats |
| V1.2 | 08/07/2016 | J. GONCALVES |  |
| V1.3 | 16/01/2016 | D. Barat | Installation sur la couche prez |

Sommaire

[I. Pré-requis à l’installation de sTunnel 4](#_Toc337467828)

[1. Vérification de la présence d’OpenSSL 4](#_Toc337467829)

[2. Installation OpenSSL 4](#_Toc337467830)

[II. Installation de sTunnel 4.53 5](#_Toc337467831)

[1. Déploiement du package 5](#_Toc337467832)

[2. Fichier de configuration 5](#_Toc337467833)

[3. Récupération du certificat 5](#_Toc337467834)

[III. Test de fonctionnement de sTunnel 6](#_Toc337467835)

[1. Démarrage de sTunnel 6](#_Toc337467836)

[IV. ANNEXE : Installation test de sTunnel 6](#_Toc337467837)

[1. Installation de sTunnel sur machine de test 6](#_Toc337467838)

# Pré-requis à l’installation de sTunnel

## Package

Le package complet est présent sous S:\84-ESL\50-Projets\PROJETS\C0870 - LOT1 - CT IPS 2015 - CONVERGENCE\Expiration des AC

## Vérification de la présence d’OpenSSL

Pour fonctionner, sTunnel a besoin d’OpenSSL pour le chiffrement. Il est donc nécessaire ce que celui-ci soit installer sur le serveur où le package de sTunnel va être déployé.

Pour vérifier si OpenSSL est installé ou non, il faut tester la commande openssl :

openssl.JPG

Si OpenSSL n’est pas installé, vous aurez ça :

openssl_2.JPG

Si OpenSSL est installé, passer au step II.

Sinon :

*root@parva2409736:/opt/stunnel-4.53/openssl-1.0.1c$ openssl*

*OpenSSL> version*

*OpenSSL 1.0.1c 10 May 2012*

## Installation OpenSSL

Pour installer OpenSSL, suivez les instructions suivantes :

1. Aller sous */opt/stunnel-4.53/openssl-1.0.1c*
2. Lancer les commandes suivantes :

*# ./config  
# make  
# make test (optionnel)  
# make install*

Par défaut, OpenSSL est installé dans le répertoire /usr/local/ssl

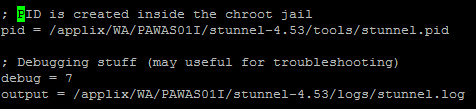
# Installation de sTunnel 4.53

## Déploiement du package

Le package source se situe dans le répertoire /opt.

Lorsque vous l’avez récupéré, copiez le dans le /opt du serveur où va être installé sTunnel puis exécutez les commandes suivantes :  
- gunzip stunnel-4.53.tar.gz  
- tar xvf stunnel-4.53.tar

## Fichier de configuration

Allez dans le répertoire /opt/stunnel-4.53/tools puis modifiez le fichier stunnel.conf selon l’environnement. Il y a 3 choses à vérifier :  
- Les chemins pour le fichier pid (permet de killer sTunnel en cas de besoin) et le fichier de log :  
  
Après avoir déclaré les chemins d’accès, ne pas oublier de les créer.

- Le bloc concernant le service ldaps, les ports et host sont à modifier selon les environnements :

Pour qu’une application puisse utiliser le LDAPs, il faut placer en amont un tunnel serveur dont le but sera de décrypter le flux ldaps envoyé par l’application :

CONFIG HORS PROD :   
*[ldapsREFSGClient]*

*;verify = 4*

*client = yes*

*accept = 1235*

*sslVersion = TLSv1*

*options = NO\_SSLv2*

*options = NO\_SSLv3*

*connect = auth-refsg-int.staging.echonet:636*

*[ldapsREFSGServer]*

*client = no*

*cert = /opt/stunnel-4.53/tools/stunnel6.pem*

*key = /opt/stunnel-4.53/tools/stunnel.key*

*accept = 1234*

*connect = 1235*

*[httpsREFOG]*

*client = yes*

*accept = 80*

*sslVersion = TLSv1*

*options = NO\_SSLv2*

*options = NO\_SSLv3*

*protocol = connect*

*protocolAuthentication = basic*

*protocolHost = ws-qds.group.echonet:443*

*protocolUsername = XXXX => A renseigner*

*protocolPassword = XXXX => A renseigner*

*connect = vip-proxy-ass.fr.net.intra:8080*

CONFIG PROD :

*[ldapsRE FSGClient]*

*;verify = 4*

*client = yes*

*accept = 1235*

*sslVersion = TLSv1*

*options = NO\_SSLv2*

*options = NO\_SSLv3*

*connect = auth-refsg.group.echonet:636*

*[ldapsREFSGServer]*

*client = no*

*cert = /opt/stunnel-4.53/tools/stunnel6.pem*

*key = /opt/stunnel-4.53/tools/stunnel.key*

*accept = 1234*

*connect = 1235*

*[httpsREFOG]*

*client = yes*

*accept = 80*

*sslVersion = TLSv1*

*options = NO\_SSLv2*

*options = NO\_SSLv3*

*protocol = connect*

*protocolAuthentication = basic*

*protocolHost = ws-qds.group.echonet:443*

*protocolUsername = XXXX => A renseigner*

*protocolPassword = XXXX => A renseigner*

*connect = vip-proxy-ass.fr.net.intra:8080*

L'option verify doit prendre la valeur 0 pour désactiver la vérification du certificat client

Le paramètre accept détermine le serveur (localhost) et le port d'entrée de la communication a rediriger (dans notre exemple, le port 1234 n’est pas utilisé sur le serveur).

Le paramètre connect détermine le serveur (localhost) et le port sur lequel rediriger la communication

Pour la connexion au RefOG, nous avons dû modifier le fichier /etc/hosts comme ceci (Attention faire une sauvegarde au préalable) :

Hors prod :

*127.0.0.1 localhost loopback refogsearch-rec.staging.echonet # loopback (lo0) name/address*

Prod :

*127.0.0.1 localhost loopback ws-qds.group.echonet # loopback (lo0) name/address*

## Génération/Intégration de certificats

Cette partie n’est pas à réaliser si les fichiers ca.srl, ca.key, ca.crt stunnel.pem et stunnel.key sont présents sous /opt/stunnel-4.53/tools

En pré-requis il faut générer dans l'ordre à l'aide de openssl :

- Une clé privée pour le tunnel serveur ==> stunnel.key

*# openssl genrsa 1024 > stunnel.key*

- A partir de cette clé privée générer une demande de signature de certificat (CSR certificate signature request) ==> stunnel.pem

*# openssl req -new -key stunnel.key > stunnel.pem*

- Une clé privée pour notre propre authorité de certification ==> ca.key

*# openssl genrsa -des3 1024 > ca.key*

- A partir de cette clé privée, générer le certificat de notre authorité de certification (auto signé) ==> ca.crt

*# openssl req -new -x509 -days 3650 -key ca.key > ca.crt*

- Signer la demande de certificat par notre authorité de certification afin de générer notre propre certificat serveur ==> stunnel6.pem

*# openssl x509 -req -in stunnel.pem -out stunnel.pem -days 365 -CA ca.crt -CAkey ca.key -CAcreateserial -CAserial ca.srl*

|  |
| --- |
|  |

## Ajout du certificat dans le catalogue WAS :

Le certificat ca.crt doit être ajouté dans les catalogues suivants :

* WAS6 : DummyServerTrustFile.jks sous /apps/WebSphere/profiles6/node/etc

La commande à exécuter pour intégrer le certificat est la suivante :

*# gsk7cmd -cert -add -file ca.crt -db “Nom\_catalogue” -pw WebAS -label "stunnel"*

## Modification de la ressource WAS :

Afin que l’application passe par le STUNNEL et ne fait pas l’appel directement à l’application cible, il faut modifier les ressources ou fichiers de configurations qui réalisent les appels comme ceci :

LDAPs (REFSG) :

* modifier « *auth-refsg-int.staging.echonet:636* » par « *localhost:1234* »
* modifier « *host=https://refogsearch-rec.staging.echonet&service=RefogWS&appuid=user\_CARDIF&apppwd=Euncv~15&proxyhost=vip-proxy-ass.fr.net.intra&proxyport=8080&proxyuser=USRw56&proxypsw=Witbe%2656&language=FR&appelService=true* » par « *http://refogsearch-rec.staging.echonet:80&service=RefogWS&appuid=user\_CARDIF&apppwd=Euncv~15&language=FR&appelService=true* »

Attention : C’est un exemple pris de l’application SESAME SERVICE il se peut que l’appel soit réalisé différemment par exemple dans un fichier de configuration tel que « CINRJServer.properties ».

Exemple pour un fichier CINRJServer.properties :

*site.pool.ldap.ldap\_qualif.poolname=Poolcardif*

*#site.pool.ldap.ldap\_qualif.servername=auth-refsg-int.staging.echonet*

*site.pool.ldap.ldap\_qualif.servername=localhost*

*#site.pool.ldap.ldap\_qualif.serverport=636*

*site.pool.ldap.ldap\_qualif.serverport=1234*

*site.pool.ldap.ldap\_qualif.username=cn=user\_CIN,ou=Authentication\_Standard,ou=APPLICATIONS,o=GROUP*

*site.pool.ldap.ldap\_qualif.password=mdpCIN\*15*

*site.pool.ldap.ldap\_qualif.min\_connection=1*

*site.pool.ldap.ldap\_qualif.max\_connection=5*

*site.pool.ldap.ldap\_qualif.timeout=7000*

*# Service REST RDI*

*cinrj.rdi.app.uid=user\_CARDIF*

*cinrj.rdi.app.pwd=Euncv~15*

*#cinrj.rdi.service.host=https://refogsearch-rec.staging.echonet*

*cinrj.rdi.service.host=http://refogsearch-rec.staging.echonet*

*cinrj.rdi.service.name=RefogWS*

*cinrj.rdi.language=FR*

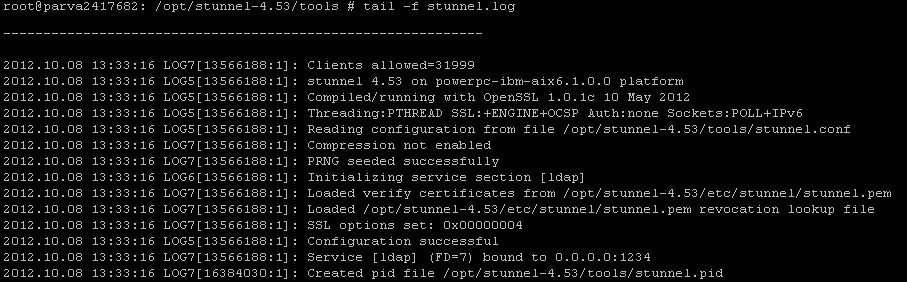
# Test de fonctionnement de sTunnel

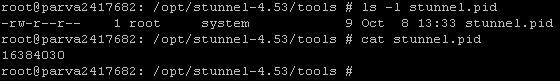
## Démarrage de sTunnel

Pour démarrer sTunnel, il faut utiliser les commandes suivantes :  
- /opt/stunnel-4.53/bin/stunnel /opt/stunnel-4.53/tools/stunnel\_REFSG.conf

- /opt/stunnel-4.53/bin/stunnel /opt/stunnel-4.53/tools/stunnel\_REFOG.conf

## Vérification de stunnel

Le fichier log (stunnel\*.log) dans /applix/WA/PAWAS01I/stunnel-4.53/logs permet de vérifier le déroulement du démarrage :  


Et si celui-ci s’est bien déroulé, le fichier stunnel\*.pid doit être créé dans le répertoire /applix/WA/PAWAS01I/stunnel-4.53/tools :  


# Installation couche présentation (Cas appel SPOC pour SesameService)

## Installation de sTunnel sur la machine de présentation

Il faut reprendre l’installation comme décrite au [Déploiement du package](#_Déploiement_du_package), [Fichier de configuration](#_Fichier_de_configuration).

Créer un fichier de conf stunnel pour SPOC :

cd /opt/stunnel-4.53/tools

cp stunnel.conf stunnel\_SPOC.conf

Vérifier si le process est bien present : [Vérification de stunnel](#_Vérification_de_stunnel)

Ensuite ajouter une section **[httpsSPOC]** au fichier /opt/stunnel-4.53/tools/stunnel\_SPOC.conf comme ci-dessous, attention les valeurs protocolHost, protocolUsername, protocolPassword, connect sont à adapter suivant les environnements :

[httpsSPOC]

client = yes

accept = 80

sslVersion = TLSv1

options = NO\_SSLv2

options = NO\_SSLv3

protocol = connect

protocolAuthentication = basic

protocolHost = spoc-project-assurance.staging.echonet:443

protocolUsername = USRw56

protocolPassword = Witbe&56

connect = vip-proxy-ass.fr.net.intra:8080

Puis démarrer le stunnel avec la commande suivante :

/opt/stunnel-4.53/bin/stunnel /opt/stunnel-4.53/tools/stunnel\_SPOC.conf

## Modification fichier /etc/hosts de la couche service

Il faut ajouter la ligne ci-dessous au fichier /etc/hosts sur le serveur de la couche applicative :

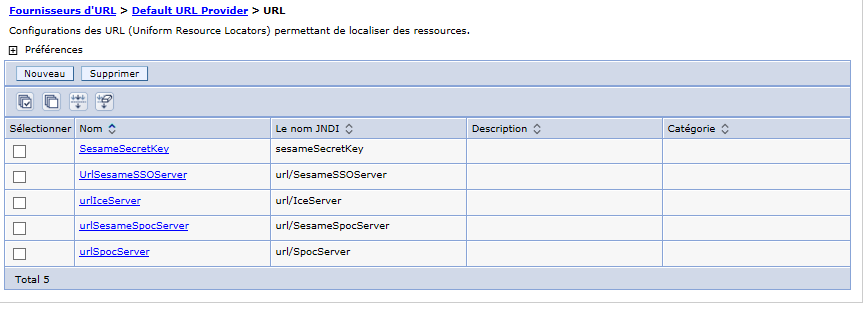
172.17.209.75 parva2410802 spoc-project-assurance.staging.echonet

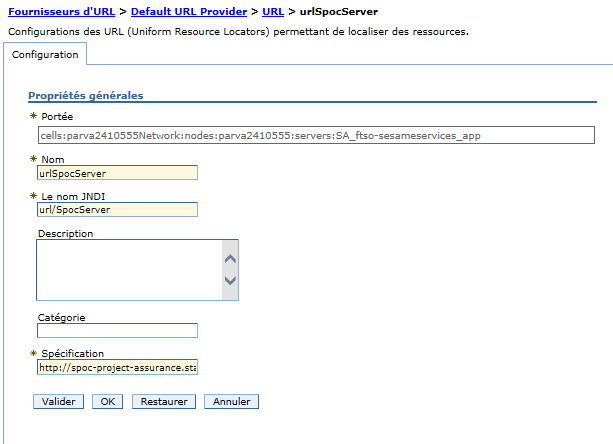
@IP serveur de présentation | nom du serveur de présentation | url de spoc

Ces données sont les valeurs de la recette.

## Modification de la ressource URL /etc/hosts de la couche service

Il faut se connecter au DMGR et modifier la ressource url **urlSpocServer** du SA\_ftso-sesameservices\_app :

Et changer l’url de spoc en **http** au lieu d’**https** :



Il faut effectuer un arrêt relance du SA une fois la modification effectuée.

# Modification crontab / supervision process

Le process stunnel doit être supervisé, voici ce qu’il faut rajouter dans la crontab de root:

##################### STUNNEL

0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46,48,50,52,54,56,58 \* \* \* \* /opt/stunnel-4.53/test\_stunnel.ksh

Et le script test\_stunnel.ksh associé :

#!/bin/ksh

DATE=$(date "+%Y%m%d")

DATE2=$(date "+%Y-%m-%dT%H-%M-%S")

CHEMIN\_LOGS=/applix/WA/PAWAS01I/stunnel-4.53/logs

CLIENT=SPOC

LOG=status\_stunnel\_${CLIENT}\_$DATE.log

echo "###################### Debut du test ############################" >> $CHEMIN\_LOGS/$LOG

echo $DATE2 >> $CHEMIN\_LOGS/$LOG

ps -ef | grep stunnel | grep tools >> $CHEMIN\_LOGS/$LOG

RETOUR\_STUNNEL=$?

if [ $RETOUR\_STUNNEL -ne 0 ]

then

echo "Le process stunnel est tombe, relance en cours..." >> $CHEMIN\_LOGS/$LOG

/opt/stunnel-4.53/bin/stunnel /opt/stunnel-4.53/tools/stunnel\_SPOC.conf

sleep 2

ps -ef | grep stunnel | grep tools >> $CHEMIN\_LOGS/$LOG

RETOUR\_STUNNEL2=$?

if [ $RETOUR\_STUNNEL2 -ne 0 ]

then

echo "Le stunnel ne se relance pas!" >> $CHEMIN\_LOGS/$LOG

else

echo "Le stunnel est relance" >> $CHEMIN\_LOGS/$LOG

fi

else

echo "Le process stunnel est bien present!" >> $CHEMIN\_LOGS/$LOG

fi

echo $DATE2 >> $CHEMIN\_LOGS/$LOG

echo "######################## Fin du test #############################" >> $CHEMIN\_LOGS/$LOG

# ANNEXE : Installation test de sTunnel

## Installation de sTunnel sur machine de test

Le package utilisé est celui de sTunnel en version 4.53 qui se trouve sous S:\84-ESL\50-Projets\PROJETS\C0870 - LOT1 - CT IPS 2015 - CONVERGENCE\Expiration des AC

Pré-requis :  
- OpenSSL installé  
- Compilateur C (gcc) présent sur la machine

Etapes à suivre :  
- mkdir/opt/stunnel/  
- gunzip -c stunnel-4.53.tar.gz > stunnel-4.53.tar  
- tar xvf stunnel-4.53.tar  
- mv (-R) /opt/stunnel/stunnel-4.53 /opt/stunnel-4.53  
- Puis dans le répertoire /opt/stunnel-4.53 :   
- Compiler en précisant le répertoire de ssl puis le répertoire d'installation de stunnel  
- ./configure --with-ssl=/usr/local/ssl/ --prefix=/opt/stunnel-4.53  
- make  
- make install