ALOHA LOAD BALANCER
MISE EN ŒUVRE DU SSL – BACKEND & FRONTEND

« APPNOTES » #0023 — MISE EN ŒUVRE SSL BACKEND ET FRONTEND

Cette note applicative a pour vocation de vous aider à implémenter la gestion du SSL sur le backend et sur le frontend (déchiffrement puis chiffrement des données avant connexion vers un serveur HTTPS) au sein de la solution ALOHA Load Balancer

CONTRAINTE

Les utilisateurs émettent une requête sécurisée (HTTPS) qui nécessitent de la persistance au niveau 7 et les serveurs Web attendent exclusivement des connexions SSL chiffrées.

OBJECTIF

Permettre que les requêtes soient sécurisées de bout en bout tout en assurant un traitement de niveau 7.

COMPLEXITE

CHANGELOG

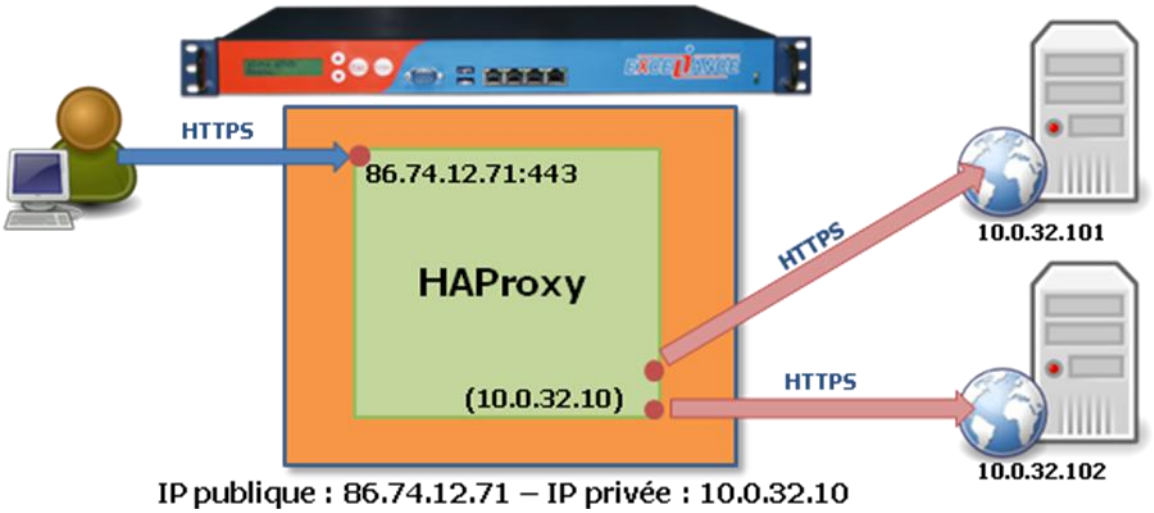
2013-01-02: Mise à jour pour ALOHA 5.5 et supérieur

2011-10-21: Mise à jour pour ALOHA 3.7 à 5.0

2001-03-31: Version initiale

ALOHA VERSION 5.5 ET SUPERIEUR

SCHEMA CIBLE



EXTRAIT DE LA CONFIGURATION LB LAYER 7 ET SSL

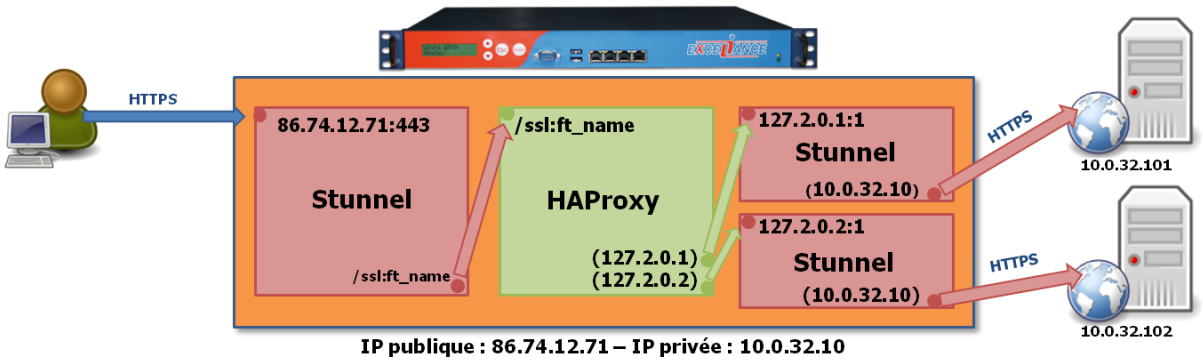
Le trafic arrive chiffré sur le frontend et ressort chiffré vers les serveurs. Entre les deux, **HAProxy** a accés en clair à toutes les informations du protocole http : utile pour faire de la persistance niveau 7 sur une connexion chiffrée de bout en bout.

```
######## The first public address as seen by the clients
frontend frt
  bind  86.74.12.71:443 ssl crt domain.com
  mode http
  log global                    # use global log parameters
  option httplog                # Enable HTTP logging
  maxconn 4000                  # max conn per instance
  timeout client 25s            # maximum client idle time (ms)
  default_backend bck           # send everything to this backend by default

####### This backend manages the servers and the load balancing algorithm
backend bck
  balance roundrobin            # roundrobin | source | uri | leastconn
  mode http
  log global                    # use global log parameters
  option httplog                # Enable HTTP logging
  cookie  SERVERID insert indirect nocache      # provide persistence with cookie
  option  httpchk HEAD /        # how to check those servers
  option forwardfor except 127.0.0.1/8      # add X-Forwarded-For except local
  timeout server 25s            # max server's response time (ms)
  server srv1 10.0.32.101:443 ssl cookie s1 weight 10 maxconn 100 check
  server srv2 10.0.32.102:443 ssl cookie s2 weight 10 maxconn 100 check
```

ALOHA VERSION 3.7 JUSQU'A 5.0

SCHEMA CIBLE



EXTRAIT DE LA CONFIGURATION SSL

La configuration de **Stunnel** est accessible directement dans l'onglet **SSL**.

```
; Service-level configuration for frontend
; forward clear requests to haproxy on 127.1.0.x
; and add the xforwarded-for header.
[ssl_frontend]
client = no
key =  /etc/ssl/frontends/SSLfrontend/key.pem
cert = /etc/ssl/frontends/SSLfrontend/crt.pem
accept  = 86.74.12.71:443
connect = /ssl:ft_name
xforwardedfor = yes

; Service-level configuration for backend
; receive haproxy traffic on 127.2.0.x
[ssl_backend_1]
client = yes
accept = 127.2.0.1:1
connect = 10.0.32.101:443

[ssl_backend_2]
client = yes
accept = 127.2.0.2:1
connect = 10.0.32.102:443
```

Vous n'avez besoin de préciser que quelques paramètres lors de l'implémentation d'un proxy SSL :

- `client = no` : place stunnel en mode server
- `key / cert` : chemin d'accès à la clef et au certificat de ce proxy SSL
- `accept` : adresse IP externe où stunnel attend les connections clients
- `connect` : socket unix interne où le traffic est envoyé en clair

Afin de chiffrer le traffic vers un serveur, il faut aussi créer un proxy.
Il doit y avoir autant de proxy que de serveur, chacun écoutant sur une adresse IP différente.
[ssl_backend_1] et **[ssl_backend_2]**

- `client = yes` : place stunnel en mode client
- `accept` : adresse IP interne où stunnel attend les connections de HAProxy
- `connect` : adresse IP du serveur où le traffic est envoyé chiffré.

EXTRAIT DE LA CONFIGURATION LB NIVEAU7

Après modification de la configuration de **Stunnel** et l'implémentation du(des) certificat(s), il ne reste plus qu'à modifier celle du niveau 7, accessible directement dans l'onglet **LB niveau7**.

Pour la partie client, il convient d'ajouter le chemin vers la socket unix d'écoute d'HAProxy qui devra être identique aux paramètres **connect** définis dans la configuration SSL avec le mot clé **accept-proxy**.

Pour la partie serveur, il faut modifier les adresses des serveurs de destination qui devront être identiques aux adresses IP des instances de **Stunnel** définies au niveau des paramètres **connect** du bloc backend dans la configuration du SSL.

```
######## The first public address as seen by the clients
frontend frt
  bind  /ssl:ft_name accept-proxy    # socket unix d'écoute
  mode http
  log global                    # use global log parameters
  option httplog                # Enable HTTP logging
  maxconn 4000                  # max conn per instance
  timeout client 25s            # maximum client idle time (ms)
  default_backend bck           # send everything to this backend by default

####### This backend manages the servers and the load balancing algorithm
backend bck
  balance roundrobin            # roundrobin | source | uri | leastconn
  mode http
  log global                    # use global log parameters
  option httplog                # Enable HTTP logging
  cookie  SERVERID insert indirect nocache      # provide persistence with cookie
  option  httpchk HEAD /        # how to check those servers
  option forwardfor except 127.0.0.1/8      # add X-Forwarded-For except local
  fullconn 4000                 # dynamic limiting below
  timeout server 25s            # max server's response time (ms)
  server srv1 127.2.0.1:1 cookie s1 weight 10 maxconn 100 check inter 1000 fall 3
  server srv2 127.2.0.2:1 cookie s2 weight 10 maxconn 100 check inter 1000 fall 3
```

DEMARRAGE DU SERVICE STUNNEL

IMPORTANT

En cas de première configuration du SSL, un message d'avertissement indique que le service **Stunnel** n'est pas démarré. Dans l'onglet **Service**, éditez la configuration du service **Stunnel** en cliquant sur le bouton «stunnel options».

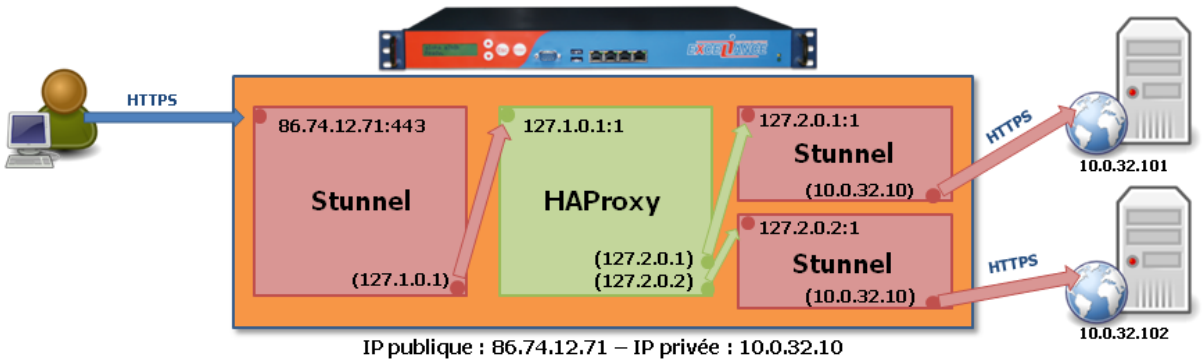
Il suffit de commander la ligne **no autostart** :

```
service stunnel
    ############## The SSL tunnel Daemon
    # no autostart
```

Il ne reste plus qu'à démarrez le service en cliquant sur le bouton «démarrer».

ALOHA VERSION 3.6 ET INFERIEUR

SCHEMA CIBLE



EXTRAIT DE LA CONFIGURATION SSL

```
; Service-level configuration for frontend
; forward clear requests to haproxy on 127.1.0.x
; and add the xforwarded-for header.
[ssl_frontend]
client = no
key =  /etc/ssl/frontends/SSLfrontend/key.pem
cert = /etc/ssl/frontends/SSLfrontend/crt.pem
accept  = 86.74.12.71:443
connect = 127.1.0.1:1
xforwardedfor = yes

; Service-level configuration for backend
; receive haproxy traffic on 127.2.0.x
[ssl_backend_1]
client = yes
accept = 127.2.0.1:1
connect = 10.0.32.101:443

[ssl_backend_2]
client = yes
accept = 127.2.0.2:1
connect = 10.0.32.102:443
```

La configuration de Stunnel est accessible directement dans l'onglet SSL.
Lors d'une implémentation du SSL en mode frontend, seuls quelques paramètres sont à renseigner :

[ssl_frontend]

- le mode de fonctionnement : le module Stunnel ne devra pas être configuré en mode client mais en mode serveur. L'option «client = no» devra être choisie,
- les chemins de la clé et du certificat créés à l'aide de l'assistant (cf: howto-0020-Mise-en-oeuvre-du-SSL-0912-fr.pdf),
- l'adresse et le port d'écoute liés à un certificat SSL,
- l'adresse et le port de redirection des requêtes à destination d'HAProxy.

[ssl_backend_1] et [ssl_backend_2]

- le mode de fonctionnement : le module Stunnel devra être configuré en mode client. L'option «client = yes» devra être choisie,
- l'adresse et le port d'écoute des requêtes en provenance d'HAProxy,
- l'adresse et le port de redirection des requêtes à destination du serveur Web.l'adresse et le port de redirection des requêtes à destination du serveur Web.

EXTRAIT DE LA CONFIGURATION LB NIVEAU7

```
######## The first public address as seen by the clients
frontend frt
  bind  127.1.0.1:1                # address:port to listen to
  mode http
  log global                       # use global log parameters
  option httplog                   # Enable HTTP logging
  maxconn 4000                     # max conn per instance
  timeout client 25s               # maximum client idle time (ms)
  default_backend bck              # send everything to this backend by default

####### This backend manages the servers and the load balancing algorithm
backend bck
  balance roundrobin               # roundrobin | source | uri | leastconn
  mode http
  log global                       # use global log parameters
  option httplog                   # Enable HTTP logging
  cookie  SERVERID insert indirect nocache      # provide persistence with cookie
  option  httpchk HEAD /                        # how to check those servers
  option forwardfor except 127.0.0.1/8          # add X-Forwarded-For except local
  timeout server 25s                            # max server's response time
  server srv1 127.2.0.1:1 cookie s1 weight 10 maxconn 100 check inter 1000 fall 3
  server srv2 127.2.0.2:1 cookie s2 weight 10 maxconn 100 check inter 1000 fall 3
```

Après modification de la configuration de Stunnel et l'implémentation du(des) certificat(s), il ne reste plus qu'à modifier celle du niveau 7 qui est accessible directement dans l'onglet LB niveau7.

Il convient de rajouter alors l'adresse et le port d'écoute d'HAProxy qui devront être identiques aux paramètres «connect» du bloc frontend et de modifier les adresses des serveurs de destination qui devront être identiques aux adresses IP des instances de Stunnel définies au niveau des paramètres «connect» du bloc backend dans la configuration du SSL.

DEMARRAGE DU SERVICE STUNNEL

IMPORTANT

En cas de première configuration du SSL, un message d'avertissement indique que le service «Stunnel» n'est pas démarré. Dans l'onglet Service, éditez la configuration du service Stunnel en cliquant sur le bouton «stunnel options».

```
service stunnel
   ############# The SSL tunnel Daemon
   # config <dir>      : daemon configuration file config /etc/stunnel/stunnel.conf
   # no autostart   # commenter le no devant autostart
```

Il ne reste plus qu'à démarrez le service en cliquant sur le bouton «démarrer».