# ALOHA LOAD BALANCER
## MANAGING SSL ON THE BACKEND & FRONTEND

**"APPNOTE" #0023 — MANAGING SSL ON THE BACKEND & FRONTEND**

*This application note is intended to help you implement SSL management on both the backend and the frontend (to encrypt data before connecting to the HTTPS server) within the ALOHA Load Balancer solution.*

### REQUIREMENTS

Users send a secure (HTTPS) request requiring layer 7 persistence and Web servers expect encrypted SSL connections only.

### PURPOSE

Ensure end-to-end security of requests while enabling layer 7 processing.

### COMPLEXITY

### CHANGELOG

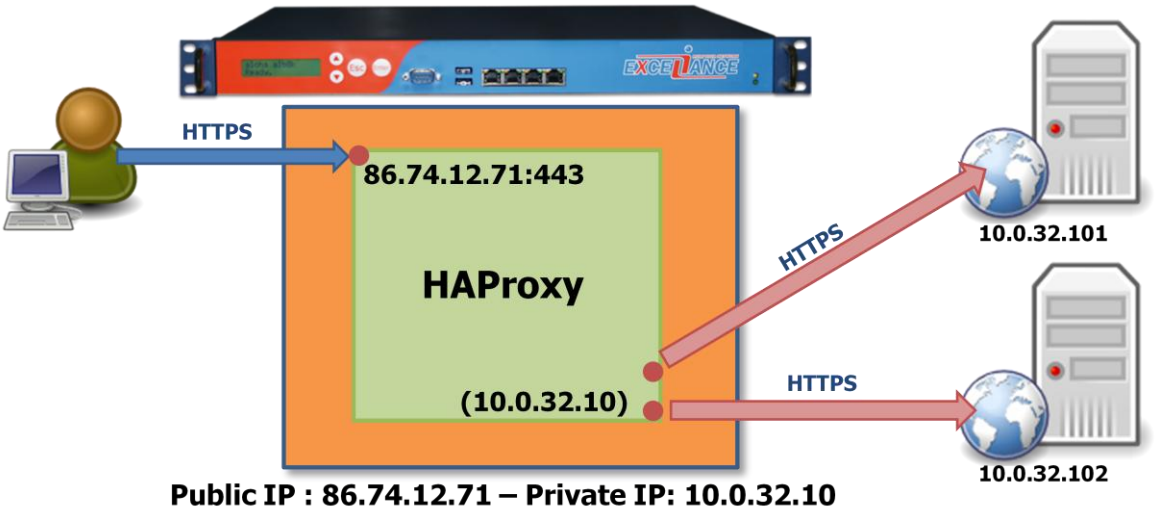
2013-01-02: update for ALOHA 5.5 and above

2011-10-21: update for ALOHA 3.7 to 5.0

2001-03-31: initial version

## ALOHA 5.5 AND ABOVE

### TARGET NETWORK DIAGRAM




### LB LAYER 7 AND SSL CONFIGURATION

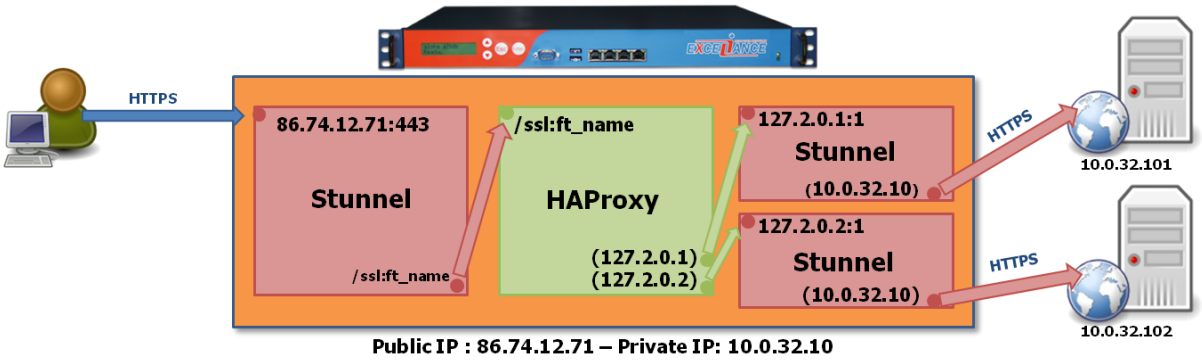
The traffic comes is ciphered between the client and the ALOHA and between the ALOHA and the server. In the ALOHA, **HAProxy** can access all the layer 7 protocol information in clear: this is useful to provide layer 7 persistence on a end to end ciphered connection.

```
######## 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 3.7 UNTIL 5.0

### TARGET NETWORK DIAGRAM




### EXTRACT OF THE SSL CONFIGURATION

You can directly access the **Stunnel** configuration from the **SSL** tab.

```
; 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
```

You only need to specify a few parameters when implementing SSL in frontend mode:

[ssl_frontend]

- the operating mode: the Stunnel module have to be configured in server mode. That's why you have to set up the **client = no** option
- the paths for the key and the certificate created using the wizard (see howto-0020-Implementing-SSL-0912-fr.pdf)
- the address and listening port linked to an SSL certificate
- the unix socket to forward traffic to HAProxy

[ssl_backend_1] and [ssl_backend_2]

- the operating mode: the Stunnel module must be configured in client mode. That's why you have to set up the **client = yes** option.
- **accept**: the listening address and port for incoming traffic from HAProxy.
- **connect**: the address and port ofthe web server to send encrypted traffic to.

## LB LEVEL7 CONFIGURATION EXTRACTION

After modifying the certificate installation and the **Stunnel** configuration update, you need to modify the layer 7 configuration.

You can access that configuration directly from the LB WUI, through the **LB layer7** tab.

```
######## The first public address as seen by the clients
frontend frt
  bind  /ssl:ft_name accept-proxy    # unix socket 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
  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
```

You must add the unix socket HAProxy listening port and they must be identical to the **connect** parameters of the **SSL frontend** block.
You also need to modify the addresses of the servers; they must be identical to the IP addresses of the **Stunnel** instances defined in the **connect** parameters of the **SSL backend** block.

## STARTING STUNNEL SERVICE

**IMPORTANT**

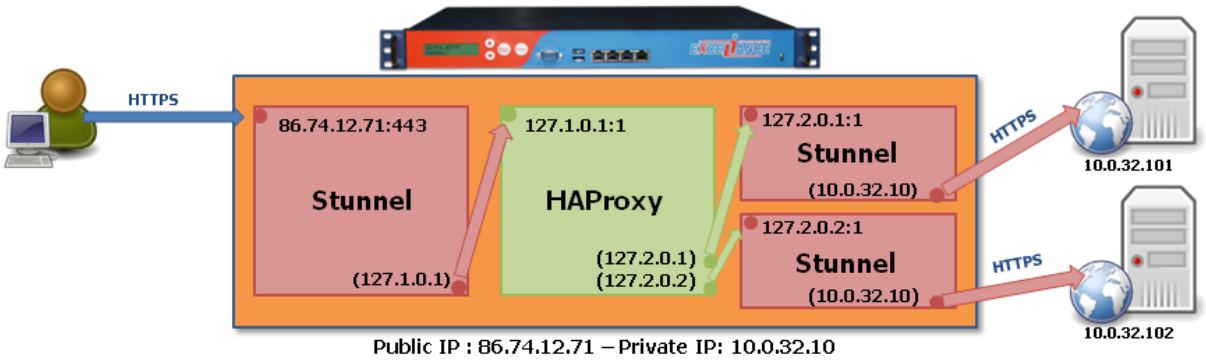
When you first configure SSL, a warning message indicates that the **Stunnel** service has not started.
In the **Service** tab, edit the **Stunnel service** configuration by clicking on the stunnel options button.
Just comment out the line **no autostart**:

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

Now you simply need to start the service by clicking the **start** button.

## ALOHA 3.6 AND BELOW

### TARGET NETWORK DIAGRAM




### EXTRACT OF THE SSL CONFIGURATION

```
; 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
```

You can directly access the Stunnel configuration from the SSL tab.
You only need to specify a few parameters when implementing SSL in frontend mode:

[ssl_frontend]

- the operating mode: the Stunnel module should not be configured in client mode, but in server mode. Therefore you should choose the "client = no" option
- the paths for the key and the certificate created using the wizard (see howto-0020-Implementing-SSL-0912-fr.pdf)
- the address and listening port linked to an SSL certificate
- the address and redirection port for requests to HAProxy

[ssl_backend_1] and [ssl_backend_2]

- the operating mode: the Stunnel module must be configured in client mode. Therefore you should choose the "client = yes" option.
- the address and redirection port for requests from HAProxy.
- the address and the redirection port of requests to the Web server. The address and redirection port of request for the Web server.

## EXTRACT OF THE LB LEVEL7 CONFIGURATION

```
######## 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
  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
```

After modifying the Stunnel configuration and the implementation of the certificate(s), all that remains is to modify the configuration of layer 7; you can access that configuration directly from the LB layer7 tab.

You must add the address and the HAProxy listening port and they must be identical to the "connect" parameters of the frontend block. You also need to modify the addresses of the destination servers; they must be identical to the IP addresses of the Stunnel instances defined in the "connect" parameters of the backend block in the SSL configuration.

## STARTING THE STUNNEL SERVICE

**IMPORTANT**

When you first configure SSL, a warning message indicates that the "Stunnel" service has not started. In the Service tab, edit the configuration of the Stunnel service by clicking the "stunnel options" button.

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

Now you simply need to start the service by clicking the "start" button.