ALOHA LOAD BALANCER
MANAGING SSL ON THE BACKEND

"APPNOTE" #0022 — MANAGING SSL ON THE BACKEND

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

CONSTRAINT

The Web servers expect encrypted SSL connections only.

PURPOSE

Enable unsecure (HTTP) requests intended for Web servers to arrive to their destination and transparently for users.

COMPLEXITY

VERSIONS CONCERNED

V 3.x and later

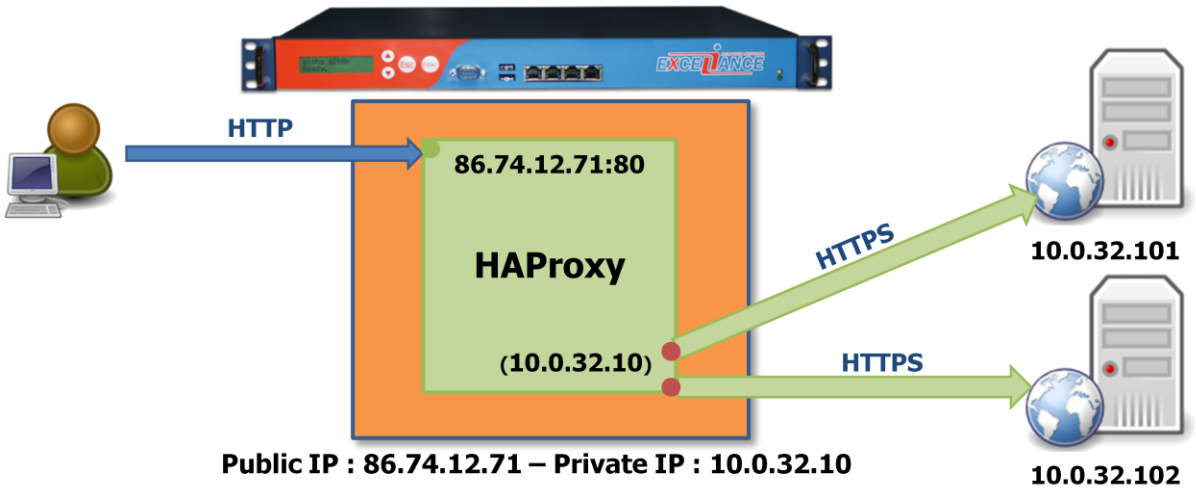
CHANGELOG

2013-01-02 : Update for ALOHA 5.5

2011-10-24 : Initial version

ALOHA 5.5 AND ABOVE

DIAGRAM



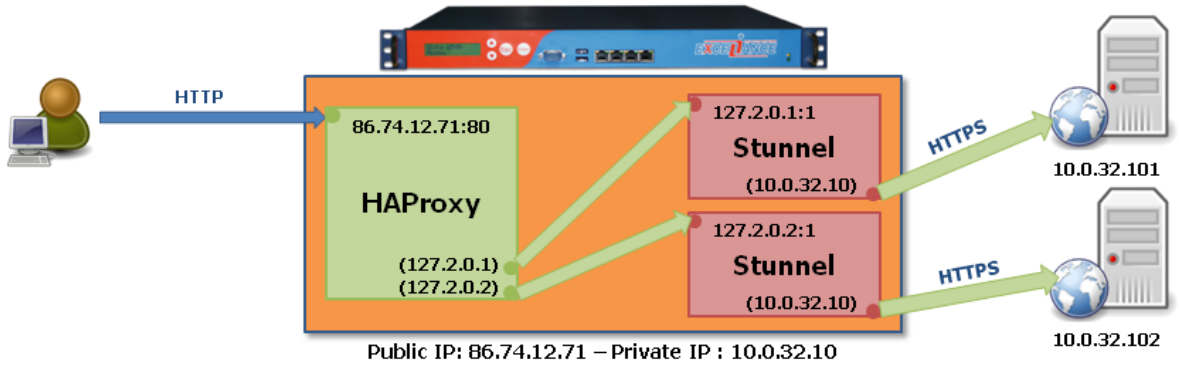
LB LAYER 7 AND SSL CONFIGURATION

The only configuration change is located on the **server** line, where we add the keyword **ssl** which tells **HAProxy** to establish a ciphered connection to the server.

```
######## The first public address as seen by the clients
frontend frt
  bind  86.74.12.71:80               # 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 (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 s1 weight 10 maxconn 100 check
```

DIAGRAM



SSL CONFIGURATION EXTRACT

```
; 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:

- the operating mode: client or non-SSL (in this case, 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 redirection port for requests to the Web server

THE LB LEVEL7 CONFIGURATION EXTRACT

```
######## The first public address as seen by the clients
frontend frt
  bind  86.74.12.71:80              # 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 level 7; you can access that configuration directly from the LB level7 tab.

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 SSL configuration.

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 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.