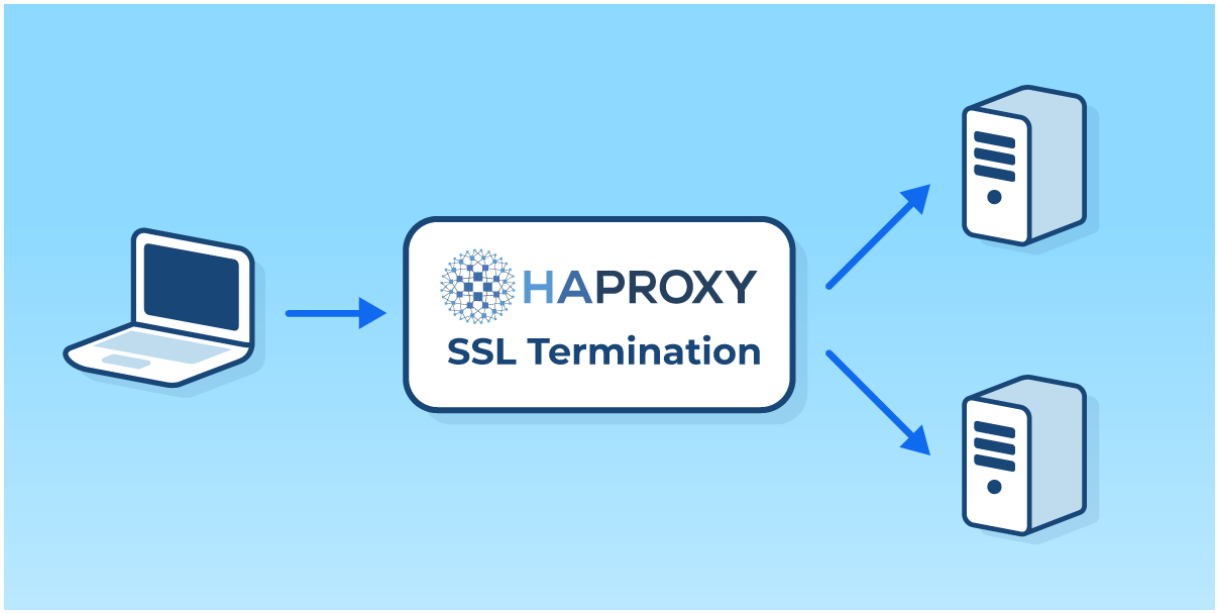
Intégration d’un certificat SSL dans Haproxy pour sécuriser les communications entre les clients et les serveurs



Réalisé par :

DJIMTONE Ngarndo Moguidbe

Encadré par :

Mr Antoine Millot

Table des matières

Introduction

Un certificat SSL sur un load balancer va nous permettre de sécuriser les communications entre les clients et les serveurs, en respectant les bonnes pratiques de sécurité réseau.

Pratique

Je vais tout d'abord me rendre sur mon haproxy et taper la commande qui va me permettre de générer une clé privée, clé que je vais personnaliser :

```
root@HAproxy:~# cd /etc/haproxy/
root@HAproxy:/etc/haproxy# openssl genrsa -out moguidbe.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................................................................
......+++++
.....+++++
e is 65537 (0x010001)
root@HAproxy:/etc/haproxy# _
```

Assurez-vous d'avoir vim, sinon vous pouvez l'installer avec la commande apt install vim.

On va taper vim [nom de notre clé], vim moguidbe.key

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAqNXDfCfmN1Bk1ywL6F/tkHm2NFfPvvS9xRZISOYafsVKPjON
oZyNpVPqJwgBmYbvyL+pgm/or37rXUtSwLzuYyLKuDHhlvJOvUG7Pvu744cPmwJO
NPSeB5whRIdzusCmNxPSdRWoPHXsd4q4v5+rmObKr4+TWfR5CQ/5SPncrJBMZjvg
pUCUbZz60Bns1/BcnubbyyDz7hQPjrSklcH+CF/kKQVRwJO5hNXQ9+kRV/tVq3dq
1kujx5GsNrm4JhpJszNagj6PP6v4Ms8TMzi0JPonm+mqksj0eunYh7AASODMD5xh
aRgC7nOp5vMlBA219UOMD/Fgnjwz/GkxGLck/wIDAQABAoIBAG/HHqZLUCqngfvH
KJcPM5wsSIo8/MiDBkS+rNxRLGdzvOSTgxhVKp4jgWmX7l8zq/rvkY1PA/FOiOT6
Ym5CZ11xAl+SfrVN9P1t0EcxqMJJf/ZzPO3mCd4l6wyQ1Pj2ZNUgJXzo+roUpJq8
V/ZeAuC3I8e5uaebqKpsTfGXoxcHW1RubyAXi4dMOyk9d7k8u6ylH/TXX8BdvNyE
8rNJGYzLzJ4v6moGKETrQNt6Ece25FMOvLAHuVFu/M5CTEtXvf+y4FoHi2WzvCdQ
kMJOIKwZgvHXlc+I5hxGCG9LWpwvun8Vusu3kVwC8ZXv1KvsEAVacY4oUIRBHKn/
PsD853ECgYEA2PjMztvzcYWVaSgXidPzd3A17EvaU4uH9Z9faso/4xIX8yJWkOrs
kGzog7o/Ct8y5RN1OrkG6JEKCGjWu3OeUXV1n22T6PsDDkdKt3MDj5aKw9axD5YY
I1W13FQNSmDVLiyWzkMIohLtMjGMQePPMNISdbLsP+ecQdbumdjmIccCgYEAxzRX
AWqTsrLuEmZd6lqm5U3WywLVq9aupScGc7oOjvh9OyXkHcF9KWripOEs11BASeC1
5uxrRM8BMUye1+sDVX7qHfYWQNN7qqifii96fiv/HcBtVjbzPBGsP7b6sA5k5Kpo
qvuEowGdxrh9uRPKZ/uIywbOsqk+2qbJSbnwYwkCgYEAhnJVZvFAft3J4aloPI3L
1fOATPYLyaOyiG9DTwcYY3p4wL5MaexIvo3d3pJwMipwfSZx+AuTTlVhpChitzFD
rzJbunjKKJoyIIzorZhlYnpkbkoA2ammNtBpqqeivK4YKO6R/CnplZSoymuNf97M
KnJh4fEHF+pLFzk6DLcskOOCgYEAqmRwz7tUHxTCY2bAioeEHrwYMIp6LoRx2r6o
3DCX519ScVwF5hXtc5IDOcEEFCCG/SqDLcWoc7EFeh/p9xMFRQpMQ9iNbGH5dB7M
aKz4ABiuHcrNO2PBZu5PAikIAbQuRRImeaqu56eoR6TjlX4CMjOL7O5EPNZb9zeX
EcNEhdECgYEAhbZvcTw2gwdoSyDfxd6Ocj7fooIgoRxleyN/QGGFFTZcd7d6gaOu
Kged1ZfRYp5wCxlT9qZVXZDdooGZRopFo+qpdyBR4l6ueYjVGdd/6ugI1uR2Ji5n
qyZ/R7nZlCuyTBeZXyA0NeN5/5E/Qfodoe4kh1O/3GEOD1okuIZmvGg=
-----END RSA PRIVATE KEY-----
~
```

Maintenant qu'on a la clé, on va créer une signature de certificat csr, on aura juste besoin de spécifier la clé pour laquelle on créer la signature.

```
root@HAproxy:/etc/haproxy# openssl req -new -key moguidbe.key -out moguidbe.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
root@HAproxy:/etc/haproxy# ls
errors  haproxy.cfg  haproxy.cfgo  haproxy.cfg.save  moguidbe.csr  moguidbe.key
root@HAproxy:/etc/haproxy#
```

Le fichier moguidbe.csr a donc été créé.

On va à présent créer un certificat auto-signé avec cette commande :

```
root@HAproxy:/etc/haproxy# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout moguidbe.key -out moguidbe.crt
Generating a RSA private key
...................+++++
.....................................+++++
writing new private key to 'moguidbe.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
root@HAproxy:/etc/haproxy#
root@HAproxy:/etc/haproxy# ls
errors  haproxy.cfg  haproxy.cfgo  haproxy.cfg.save  moguidbe.crt  moguidbe.csr  moguidbe.key
root@HAproxy:/etc/haproxy# _
```

On va à présent avec la commande cat fusionner la clé et la signature pour obtenir un fichier moguidbe.pem :

```
root@HAproxy:~# cd /etc/haproxy/
root@HAproxy:/etc/haproxy# ls
errors  haproxy.cfg  haproxy.cfgo  haproxy.cfg.save  moguidbe.crt  moguidbe.csr  moguidbe.key
root@HAproxy:/etc/haproxy# cat moguidbe.key moguidbe.crt >> /etc/ssl/private/moguidbe.pem




root@HAproxy:/etc/haproxy# ls
errors  haproxy.cfg  haproxy.cfgo  haproxy.cfg.save  moguidbe.crt  moguidbe.csr  moguidbe.key
root@HAproxy:/etc/haproxy# cd
root@HAproxy:~# cd /etc/ss
ssh/ ssl/
root@HAproxy:~# cd /etc/ssl/private/
root@HAproxy:/etc/ssl/private# ls
moguidbe.pem
root@HAproxy:/etc/ssl/private# _
```

Avec vim :

On va enfin ajouter cette certification dans le fichier de configuration d’haproxy tout en faisant une redirection vers le site sécurisé:

```
frontend http_front
  bind *:80
  bind *:443 ssl crt /etc/ssl/private/moguidbe.pem
  redirect scheme https if !{ ssl_fc }
  mode http
  balance roundrobin
  default_backend http_back
```

Après avoir modifier le fichier de configuration, je redémarre le service haproxy et je vérifie son état pour enfin vérifier si le fichier haproxy.cfg est valide :

```
root@HAproxy:~# service haproxy reload
root@HAproxy:~# service haproxy status
● haproxy.service - HAProxy Load Balancer
     Loaded: loaded (/lib/systemd/system/haproxy.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2024-04-08 12:56:49 CEST; 3min 28s ago
       Docs: man:haproxy(1)
             file:/usr/share/doc/haproxy/configuration.txt.gz
    Process: 520 ExecStartPre=/usr/sbin/haproxy -f $CONFIG -c -q $EXTRAOPTS (code=exited, status=0/SUCCESS)
    Process: 570 ExecReload=/usr/sbin/haproxy -f $CONFIG -c -q $EXTRAOPTS (code=exited, status=0/SUCCESS)
    Process: 571 ExecReload=/bin/kill -USR2 $MAINPID (code=exited, status=0/SUCCESS)
   Main PID: 528 (haproxy)
      Tasks: 2 (limit: 2354)
     Memory: 10.8M
     CGroup: /system.slice/haproxy.service
             ├─528 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -sf 529 -x /run/haproxy/admin.sock
             └─572 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -sf 529 -x /run/haproxy/admin.sock

avril 08 13:00:09 HAproxy haproxy[528]: [WARNING] 098/130009 (528) : parsing [/etc/haproxy/haproxy.cfg:44] : 'balance' ignored b
avril 08 13:00:09 HAproxy haproxy[528]: [WARNING] 098/130009 (528) : Setting tune.ssl.default-dh-param to 1024 by default, if yo
avril 08 13:00:09 HAproxy haproxy[528]: Proxy statut started.
avril 08 13:00:09 HAproxy haproxy[528]: Proxy statut started.
avril 08 13:00:09 HAproxy haproxy[528]: Proxy http_front started.
avril 08 13:00:09 HAproxy haproxy[528]: Proxy http_front started.
avril 08 13:00:09 HAproxy systemd[1]: Reloaded HAProxy Load Balancer.
avril 08 13:00:09 HAproxy haproxy[528]: Proxy http_back started.
avril 08 13:00:09 HAproxy haproxy[528]: Proxy http_back started.
avril 08 13:00:10 HAproxy haproxy[528]: [WARNING] 098/130009 (528) : Former worker 529 exited with code 0

root@HAproxy:~# haproxy -c -f /etc/haproxy//haproxy.cfg
[WARNING] 098/132759 (599) : parsing [/etc/haproxy//haproxy.cfg:44] : 'balance' ignored because frontend 'http_front' has no bac
kend capability.
[WARNING] 098/132759 (599) : Setting tune.ssl.default-dh-param to 1024 by default, if your workload permits it you should set it
 to at least 2048. Please set a value >= 1024 to make this warning disappear.
Configuration file is valid
root@HAproxy:~#
```

Tests :

Quand on se rend sur l'IP de l'Haproxy (le front) on a un message qui nous prévient :

Bienvenue sur le Serveur Web 2 - Test

Créer un ticket GLPI

Cliquez sur le lien ci-dessous pour créer un nouveau ticket :

Créer un ticket

La charge est bien repartie entre les deux serveurs web et les utilisateurs peuvent y accéder de manière sécurisée.

Conclusion :

On a pu sécuriser le trafic entre les utilisateurs et les serveurs web grâce au certificat. Tout en bloquant l'accès à l'interface.