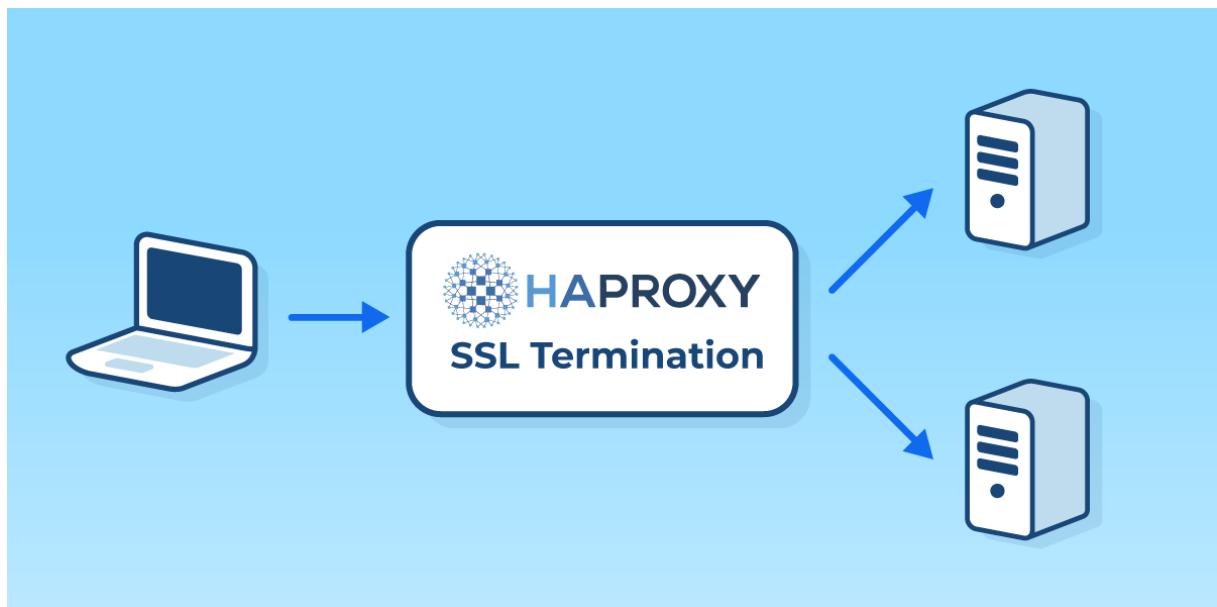


# **Intégration d'un certificat SSL dans Haproxy pour sécuriser les communications entre les clients et les serveurs**



**Réalisé par :**

**DJIMTONE Ngarndo Moguidbe**

**Encadré par :**

**Mr Antoine Millot**

# Table des matières

<u>Introduction</u> .....	1
<u>Pratique</u> .....	1
<u>Tests :</u> .....	5
<u>Conclusion :</u> .....	6

## 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-----
MIIEpQIBAACAQEAQNXDfCfmN1Bk1yul6F/tkHm2NFfPvvS9xRZIS0YafsVKPjON
oZyNpVPqJwgBmYbvyl+pgm/or37rXUtSwLzuYyLKuDhHlvJ0vUG7Pvu744cPmwJO
NPSeB5uRhRIdzusCmNxPSdRwOPHXsd4q4v5+rm0bKr4+TWfR5CQ/5SPncrJBMZjvg
pUCUbZz60Bns1/BcnubbyyDz7hQPjrSk1cH+CF/kKQVRwJ05hNXQ9+kRV/tVq3dq
1kuJx5GsNrm4JhpJszNagj6PP6v4Ms8TMzi0JPonm+mqksj0eunYh7AASODMD5xh
aRgC7n0p5vM1BA219U0MD/Fgnjwz/GkxGLck/wIDAQABAOIBAG/HHqZLUCqngfvH
KJcPM5wsSiO8/MiDBkS+rNxRLGdzvOSTgxhVKp4jgWmX718zq/rvkY1PA/F0iOT6
Ym5CZ11xA1+SfrVN9P1t0EcXqMJJf/ZzP03mCd416wyQ1Pj2ZNUgJXzo+roUpJq8
V/ZeAuC3I8e5uaebqKpsTfGXoxcHW1RubyAXi4dM0yk9d7k8u6y1H/TXX8BdvNyE
8rNJGYzLzJ4v6moGKETrQNT6Ece25FM0vLAHuVFu/M5CTEtXvf+y4FoHi2WzvCdQ
kMJDIKwZgvHX1c+I5hxGCG9LWpwwun8Vusu3kVwC8ZXv1KvsEAVacY4oUIRBHKn/
PsD853ECgYEA2PjMztvzcYWVaSgXidPzd3A17EvaU4uH9Z9faso/4xIX8yJwK0rs
kGzog7o/Ct8y5RN10rkG6JEKCGjWu30eUXV1n22T6PsDDkdKt3MDj5aKw9axD5YY
I1W13FQNSmDVLiyWzkMIohLtMjGMQePPMNISdbLsP+ecQdbumdjmIccCgYEAxzRX
AWqTsrLuEm2d6lqm5U3WyuLVq9aupScGc7o0jvh90yXkHcF9Kwrip0Es11BAsEC1
SuxrRM8BMUye1+sDVX7qHfYWQNN7qqifii96fiv/HcBtVjbzPBGsP7b6sA5k5Kpo
quvEowGdxrh9uRPKZ/uIywb0sqk+2qbJSbnwYukCgYEAhnJVZvFAft3J4aloPI3L
1f0ATPYLya0yig9DTwcYY3p4wL5MaexIvo3d3pJwMipwfsZx+AuTT1VhpChitzFD
rzJbunjKKJoyIIzorZhlYnpkbkoA2ammNtBpqqeivK4YK06R/CnplZSoymuNf97M
KnJh4fEHF+pLFzk6DLcSk00CgYEAqmRwz7tUHxTCY2bAioeEHrwYMIp6LoRx2r6o
3DCX519ScVwF5hXtc5ID0cEEFCcG/SqDLcWoc7EFeh/p9xMFRQpMQ9iNbGH5dB7M
aKz4ABiuHcrN02PBZu5PAikIAbQuRRImeaqu56eOR6Tj1X4CMjOL705EPN2b9zeX
EcNEhdECgYEAhb2vcTw2gwdoSyDfxd60cj7fooIgoRxleyN/QGGFFTZcd7d6ga0u
Kged1ZfRYp5wCx1T9qZVXZDdooGZROPo+qpdyBR416ueYjVGdd/6ugI1uR2Ji5n
qyZ/R7nZ1CuyTBeZXYA0NeN5/5E/Qfodoe4kh10/3GE0D1okuIZmvGg=
-----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 crée 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/ssl
ssh/ ssl/
root@HAproxy:~# cd /etc/ssl/private/
root@HAproxy:/etc/ssl/private# ls
moguidbe.pem
root@HAproxy:/etc/ssl/private# _

```

Avec vim :

```

-----BEGIN PRIVATE KEY-----
MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBAKwggSiAgEAAoIBAQC07eZU5z+BNXn2
JCNJsvTkqAQf7SI0XLp39jVwY09sIrw5IGTaYaCsEQeImXv1z8RBrmcY7PdFRopv
7nwuxeniJ4swy1w1JX5u6IuU0X1HGMQixkViJjHTR0Q0Vws2dufLhwq4T0zH0E4s
EcIKVZutbf5fa0HY4csVWU5B17zut3czDokYqc1aLP01HzNroHVF+0ySivcF4iDB
oEXY1bMsv9R1ADvfQtWBEQKjwkSK43s34fnABLWrMv2Q2QQW0/Jig1b8JmtZgS/Z
HK8ikDoDDiJjspeFBL5HgZAGNfjLe5b0CMgRoNJIIsUqKgqjrBVun0b0f2zHTNxpnb
oub1yshAgMBAAGCggEAb/pBMNECc8SWHCnY9bVFbfMyw10wdLnjSnWeMrdDI4kH
908+PA5qM7RCd1+P4d3Z+FCpTDro+vIiog323WSQD77HZJQqS/wZM1ccdfmE9uS0
F1WhAdJA0oQECWPIfCfxY/zLPK+Xad9a8113P5AmTpsH8Qpf8YwldambSpLsQs1N
VrMmY8Qfi/LNyCRz2bnjPJ9G0tenz0+1pfApib/7DkfH30r98mdMsQJBWtdDmhws
KDL60j+mqMxVd9+3CBmuvrFwg6TM1z0xnQcfsU1prH3TSQ1kYMDGJdjHpS3JW5Am
3habK32VEYeWVQ7AHqgDDok/X68Eu/e3kyoEzaCCMQKBgQDZNSV+R05XzhLuH4/b
7tTW79QpKQYkwa/ZMKj/BNzB1B+buWp0AnQAMyZB+zqizyFNrpjEGUCGMt0aYoQh
m5z7YReHgIMNcuwvphV/9h/YJwKioHgw/rj/qCe0F5yxYr6ZgJvFmn3QkUnTHT0Z
ypj2S9WeZ5GfRycadMzac5qpFQKBgQDVPHa0i/PyJkn3H0oB1+1B6HGHaR1o1U/K
1Cpb2iRML7EKVrJKQL1a84sAN4iVxKnZWEMQ/63/BZCy6NGHZK0BpXNjrGjUEuu
Mv6rYx1cdn4aWYQy0RRkv9ZVlaxuT5cPxyUGir+CTK3f0Lb4Xwazaff8Z5/KfGHG
mtNRM+tk3QKBgAr1t7PWMAAHvrtjqfWgX3bhIWvP0xes6x6ia1ghD1JLKirIsuaP
GPLJnInBSPfSHIGH0Njdy3bwJd2hA7AL9pxGCsApr0iFdR80yTzaSs61ZIRyaJP+
w51P/muKkznzEkzXpT9zGMzV/TYySNx2ZaY1t8nyuE0ddkDAADixipjhAoGAB0nX
/NPLwrsMTCXw220Xm+Kv+ErFU3PHBme6Cump0/CvwISpVbbRIo2MxbRB42qWeKi+
buzkB+XV3PKY4gygtwaCmEQITkywSP0sGby2fY1CTGGEzaS099StBopiputzhaA
Z7jWUt3ME5Ucq+CTmg5FK5tfGsy1iuQ/1MaxhQECgYAYRti921kLdx8xtvJZf7v
I1PKP405qoT6I7rSTJ6W8SubX3I+V4estzaZG87Xd6t8KFm+mxynZdQZ/SX8fC8Q
WpJAmUJB87LzaTOMHSS7nY6kccqbeU0ageuDAPKC5HB812Vo5t9aK1o2Nuut531
PqInkjZ4+BaMhDdMK3VMSQ==
-----END PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIIDazCCA10gAwIBAgIUC2GImKIZDDCK2WI/R9NvAfYt6i0wDQYJKoZIhvcNAQEL
BQAwRETELMAKGA1UEBhMCQVUxEzARBgNVBAgMCInvbWUtU3RhdGUxITAfBgNVBAoM
GE1udG9ybmV0IFdpZGdpdHMGUHR5IEExODAEFw0yNDA0MDgxMDQxMzZaFw0yNTA0
MDgxMDQxMzZaMEUxCzAJBgNVBAYTAkFVMRMwEQYDVQQIDApTb211LVNOYXRIMSEw
HwYDVQQKDBhJbnR1cm5ldCBXaWRnaXRzIFB0eSBMdGQwggE1MA0GCSqGSIb3DQEB
AQUAA4IBDwAwggEKAoIBAQC07eZU5z+BNXn2JCNJsvTkqAQf7SI0XLp39jVwY09s
Irw5IGTaYaCsEQeImXv1z8RBrmcY7PdFRopv7nwuxeniJ4swy1w1JX5u6IuU0X1H
GMQixkViJjHTR0Q0Vws2dufLhwq4T0zH0E4sEcIKVZutbf5fa0HY4csVWU5B17zu
t3czDokYqc1aLP01HzNroHVF+0ySivcF4iDBoEXY1bMsv9R1ADvfQtWBEQKjwkSK
43s34fnABLWrMv2Q2QQW0/Jig1b8JmtZgS/ZHK8ikDoDDiJjspeFBL5HgZAGNfjL
e5b0CMgRoNJIIsUqKgqjrBVun0b0f2zHTNxpnboub1yshAgMBAAGjUzBRMB0GA1Ud
DgQWBBS7mkyEkTNweeNpnEteakVAKWNxKTAfBgNVHSMEGDAwBS7mkyEkTNweeNp
nEteakVAKWNxKTAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQCm
LMVRUD5rzP1hd0A1E1i00FzP9bjmQGejxkG6mBRh+bbWdRSPXerTVa9tWJ/5qFfF
HA70MBK3uqxGz144FzaGjypjbOPvk7HBjaatGAqYo2QaAAVhJt51Zux4c+dSF2J
F9kSvoIabZuXcJKh7Bq1+TV/CorY44cw+GVMQCC5NR+brQPkSseHhA/WLqcxH8u
hoLGbog1NYPi5b0NVGnd+S6VrMDYYtrYsx6mGhGkhodV8bXPNLXDsiU3GK5yGWe2
CK0PzE+yW78A1EJ/kC5EnEPodZixYbVjE5gVoHYoBzS1FD6fHiY7+gegrgrxq2en
"/etc/ssl/private/moguidbe.pem" 49L, 2949C

```

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@HAproux:~# service haproxy reload
root@HAproux:~# 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 $EXTRA_OPTS (code=exited, status=0/SUCCESS)
  Process: 570 ExecReload=/usr/sbin/haproxy -f $CONFIG -c -q $EXTRA_OPTS (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 HAproux haproxy[528]: [WARNING] 098/130009 (528) : parsing [/etc/haproxy/haproxy.cfg:44] : 'balance' ignored b
avril 08 13:00:09 HAproux haproxy[528]: [WARNING] 098/130009 (528) : Setting tune.ssl.default-dh-param to 1024 by default, if yo
avril 08 13:00:09 HAproux haproxy[528]: Proxy statut started.
avril 08 13:00:09 HAproux haproxy[528]: Proxy statut started.
avril 08 13:00:09 HAproux haproxy[528]: Proxy http_front started.
avril 08 13:00:09 HAproux haproxy[528]: Proxy http_front started.
avril 08 13:00:09 HAproux systemd[1]: Reloaded HAProxy Load Balancer.
avril 08 13:00:09 HAproux haproxy[528]: Proxy http_back started.
avril 08 13:00:09 HAproux haproxy[528]: Proxy http_back started.
avril 08 13:00:10 HAproux haproxy[528]: [WARNING] 098/130009 (528) : Former worker 529 exited with code 0

root@HAproux:~# 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@HAproux:~#

```

## 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 répartie 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.