Sécuriser les communications HTTPS

Configuration

Voici la configuration a ajouter dans les VirtualHost :

```
# Mise en place du HSTS
Header always set Strict-Transport-Security "max-age=63072000;
includeSubdomains;"

# Il est nécessaire d'activer SSL, sinon c'est http qui sera utilisé
SSLEngine On

# Les clefs du serveur :
SSLCertificateFile /path/to/file/fullchain.pem
SSLCertificateKeyFile /path/to/file/privkey.pem

# On autorise TLSv1.2, on rejette les autres
SSLProtocol -all +TLSv1.2

# On autorise uniquement les clefs de cryptage longue (high).
SSLCipherSuite HIGH:!kRSA:!kDHr:!kDHd:!kSRP:!aNULL:!3DES:!MD5
```

et il suffira d'activer les modules correspondants :

```
a2enmod ssl
a2enmod headers
```

Tests

Pour vérifier l'efficacité, lancer le test sur https://tls.imirhil.fr
Le projet est OpenSource, et disponible ici

La notation est simple :

De **A+** à **F** avec pour référence **A** qui équivaut a la recommandation de la RFC7525
M indique que le certificat ne correspond pas au bon nom de domaine
T indique que le certificat n'est pas validé par un certificat d'autorité root

Attention, le test prend la configuration des protocoles par le premier VirtualHost.

From:
https://wiki.virtit.fr/ - **VirtIT**

Permanent link:
https://wiki.virtit.fr/doku.php/kb:linux:apache2:securiser_les_communications_https

Last update: **2017/12/09 00:19**