Apache

Les exemples de commandes sont donnés pour Debian/Ubuntu.

Installation

Ubuntu

Installation paquet

```
sudo apt-get install apache2
```

Vérification version

```
apachectl -v
```

Redhat/CentOS

Installation paquet

```
sudo dnf install httpd
```

Démarrage / Activation

```
sudo systemctl start httpd
sudo systemctl enable httpd
```

Vérification version

```
httpd -v
```

Haut de page

Configuration générale apache

Ubuntu

Fichiers/Répertoires

/etc/apache2/apache2.conf	Fichier de configuration générale
/etc/apache2/ports.conf	Fichier contenant les ports à écouter, par défaut 80 (http) et 443 (https)
/etc/apache2/sites-available	Répertoire contenant les fichiers de conf pour les sites disponibles
/etc/apache2/sites-enabled	Répertoire contenant les liens vers fichiers de conf pour les sites activés (par *a2ensite*)
/etc/apache2/mods-available	Répertoire contenant les fichiers de conf des modules Apache disponibles
/etc/apache2/mods-enabled	Répertoire contenant les liens vers les modules Apache activés (par *a2enmod*)

Paramètres par défaut

- User : www-data
- Group : www-data

Redhat/CentOS

Fichiers/Répertoires

/etc/httpd/conf/httpd.conf	Fichier de configuration générale
/etc/httpd/conf.d	Répertoire contenant les fichiers de conf pour les sites disponibles
/etc/httpd/conf.modules.d	Répertoire contenant les fichiers de conf avec les modules Apache à charger

Paramètres par défaut

- User : apache
- Group : apache

selinux

En cas de problème de droits (Erreur 403) vérifier si selinux est activé, vérifier les logs et changer les droits des fichiers :

```
sestatus
tail -f /var/log/audit/audit.log
sudo chcon -R -t httpd_sys_content_t /var/www/html/
```

Haut de page

Configuration sites web

Les fichiers .conf du répertoire /etc/apache2/sites-available sont explorés dans l'ordre alphabétique.

Exemples

http

```
<VirtualHost *:80>
        ServerName www.site.com
        ServerAlias site.com
        DocumentRoot "/var/www/html/site"

        ErrorLog ${APACHE_LOG_DIR}/site.error.log
        CustomLog ${APACHE_LOG_DIR}/site.access.log combined
</VirtualHost>
```

https

Attention de penser à activer le module ssl avec la commande
sudo a2enmod ssl

```
<IfModule mod_ssl.c>
        <VirtualHost *:443>
                ServerName www.site.com
                ServerAlias site.com
                DocumentRoot "/var/www/html/site"

                ErrorLog ${APACHE_LOG_DIR}/site.error.log
                CustomLog ${APACHE_LOG_DIR}/site.access.log combined

                SSLEngine on
                SSLCertificateFile    /etc/ssl/certs/ssl-cert-snakeoil.pem
                SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
        </VirtualHost>
</IfModule>
```

Sécurisation

Redirection du http sur le https

```
<VirtualHost *:80>
        ServerName www.site.com

        ErrorLog ${APACHE_LOG_DIR}/site.error.log
        CustomLog ${APACHE_LOG_DIR}/site.access.log combined

        Redirect permanent / https://site.com/
</VirtualHost>
```

Interdire l'accès direct par l'adresse IP

```
<VirtualHost *:80>
        ServerName xxx.xxx.xxx.xxx
        <Directory />
                Deny from all
        </Directory>
</VirtualHost>

<VirtualHost *:443>
        ServerName xxx.xxx.xxx.xxx
        <Directory />
                Deny from all
        </Directory>
</VirtualHost>
```

VirtualHost par défaut

Dans l'exemple suivant on bloque l'accès pour les url non prévues.

```
<VirtualHost _default_:80>
        ErrorLog ${APACHE_LOG_DIR}/default-http.error.log
        CustomLog ${APACHE_LOG_DIR}/default-http.access.log combined
        <Directory /var/www/html>
            Require all denied
        </Directory>
</VirtualHost>

<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ErrorLog ${APACHE_LOG_DIR}/default-https.error.log
        CustomLog ${APACHE_LOG_DIR}/default-https.access.log combined

        SSLEngine on
```

```
        SSLCertificateFile    /etc/ssl/certs/ssl-cert-snakeoil.pem
        SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
        <Directory /var/www/html>
            Require all denied
        </Directory>
    </VirtualHost>
</IfModule>
```

Commandes

```
sudo apache2ctl -M                ⇒ liste les modules Apache chargés
sudo a2enmod <module>             ⇒ activation module
sudo a2ensite <fichier>           ⇒ activation site
sudo apachectl configtest         ⇒ test de la configuration
sudo systemctl reload apache2     ⇒ recharge la configuration Apache
sudo systemctl restart apache2    ⇒ redemarre le service Apache
```

En cas de problème penser à vérifier qu'il n'y a pas un firewall qui bloque le flux.

Haut de page

Utilisation SSL

Activation

Ubuntu

```
sudo a2enmod ssl
```

Redhat/CentOS

```
sudo dnf install mod_ssl
```

Double authentification

Ajouter une directive *SSLProxyMachineCertificateFile* pour pointer sur le fichier contenant le certificat du client.

Générer un certificat autosigné

Installer openssl

```
sudo apt-get install openssl
```

Générer une clef privée

```
cd /etc/ssl
sudo openssl genrsa -out cle-privee.pem 2048
```

Générer une demande de signature de certificat (Certificat Signing Request)

```
sudo openssl req -new -key cle-privee.pem -out demande-csr.pem
```

Attention de bien répondre avec l'URL www.site.com pour le champ «Common Name»

Visualiser le résultat

```
openssl req -text -noout -in demande-csr.pem
```

Générer le certificat auto-signé pour 365 jours

```
sudo openssl x509 -req -days 365 -in demande-csr.pem -signkey cle-privee.pem
-out certificat.pem
```

Visualiser certificat

```
openssl x509 -noout -text -in certificat.pem
```

Générer un certificat signé par une autorité bidon

Générer la Clé Privée de la CA

```
openssl genrsa -out ca.key 2048
```

Créer un Certificat pour la CA

```
openssl req -x509 -new -nodes -key ca.key -sha256 -days 3650 -out ca.crt
```

Créer la Clé Privée pour le Serveur

```
openssl genrsa -out serveur.key 2048
```

Créer une Demande de Signature de Certificat (CSR) pour le Serveur

```
openssl req -new -key serveur.key -out serveur.csr
```

> Attention de bien répondre avec l'URL www.site.com ou l'adresse IP pour le champ «Common Name»

Signer le Certificat du Serveur avec la CA factice

```
openssl x509 -req -in serveur.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out serveur.crt -days 365 -sha256
```

- Si besoin de générer un certificat avec SAN (Subject Alternative Names), il suffit d'ajouter l'option **-extfile <fichier.ext>** pour pointer sur un fichier avec les infos :

```
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment,
dataEncipherment
subjectAltName = @alt_names
[alt_names]
IP.1 = 13.36.137.5
IP.2 = 10.1.3.23
IP.3 = 127.0.0.1
DNS.1 = localhost
DNS.2 = e2e-r23-ngrfid
```

Combiner la Clé Privée et le Certificat du Serveur

```
cat serveur.key serveur.crt > serveur.pem
```

Installer le Certificat CA sur les Clients (si nécessaire)

Pour que les clients reconnaissent le certificat auto-signé comme “fiable”, il faut installer le fichier

ca.crt sur chaque client ou application qui se connectera au serveur.

```
sudo cp ca.crt /usr/local/share/ca-certificates/
sudo update-ca-certificates
```

Générer un certificat Let's Encrypt

Installation certbot

- Sur Ubuntu 16.04 :

```
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:certbot/certbot
sudo apt-get update
sudo apt-get install python-certbot-apache
```

- Sur Ubuntu 20.04 :

```
sudo snap install core
sudo snap install --classic certbot
sudo ln -s /snap/bin/certbot /usr/bin/certbot
sudo apt-get install python3-certbot-apache
```

- Sur Redhat 9

```
sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-
latest-9.noarch.rpm
sudo dnf install certbot python3-certbot-apache
```

Génération certificat

```
sudo certbot --apache certonly
```

L'option certonly évite la modification automatique des fichiers de conf. Il faut donc les modifier manuellement pour indiquer les chemins vers les fichiers contenant les clefs.

Génération certificat pour plusieurs url

```
sudo certbot --apache certonly –d url1,url2,...
```

Renouvellement certificat

```
sudo certbot  --standalone  certonly -d url1,url2,...
```

certbot renouvelle automatiquement les certificats à l'aide de la crontab (voir le fichier /etc/cron.d/certbot)

Liste

```
sudo certbot certificates
```

Suppression

```
sudo certbot delete --cert-name <nom>
```

Laisser le port 80 fermé

Pour ne pas laisser le port 80 ouvert :

- Supprimer la ligne Listen 80 du fichier /etc/apache2/ports.conf
- ajouter l'option --standalone à la commande certbot pour qu'il gère lui-même le port 80

```
sudo certbot -q renew --dry-run --standalone
```

Vérification connexion SSL

```
openssl s_client -connect @IP:port
```

Vérification expiration certificat

Il est possible d'utiliser le script ssl-cert-check pour vérifier l'expiration des certificats.

Vérification certificat local

```
ssl-cert-check -d /etc/letsencrypt/live/*/*.pem
```

Vérification certificat site distant

```
ssl-cert-check -n -p 443 -s <url>
```

Haut de page

Sécurisation des entêtes http

Source

https://geekflare.com/http-header-implementation/
https://developers.google.com/web/fundamentals/security/csp
https://developer.mozilla.org/fr/docs/Web/HTTP/Headers

Paramètres

Strict-Transport-Security

Force la communication en utilisant HTTPS au lieu de HTTP.

X-Frame-Options

Indique si le navigateur est autorisé à afficher une page dans un <frame>, <iframe> ou <object>.

- DENY : interdiction complète
- SAMEORIGIN : autorise uniquement depuis le même site
- ALLOW-FROM : autorise depuis un site donné

X-Content-Type-Options

Désactive le repérage MIME et force le navigateur à utiliser le type donné dans Content-Type.

X-XSS-Protection

Active le filtrage de script intersite.

Content-Security-Policy

Contrôle les ressources que l'agent utilisateur est autorisé à charger pour une page donnée.

Paramètres :

- default-src : valeur par défaut des directives de récupération qui ne sont pas définies explicitement.
- script-src : sources valides pour les fichiers JavaScript.
- img-src : sources valides pour les images et les favicons.
- media-src : sources valides pour les ressources média des éléments <audio> et <video>.
- frame-src : sources valides pour les éléments qui représentent des contextes de navigation

imbriqués, tels que <frame> et <iframe>.
- font-src : sources valides pour les polices de caractères chargées depuis @font-face.
- style-src : sources valides pour les feuilles de styles.

Valeurs :

- 'self' : uniquement depuis le même site
- 'unsafe-inline' : à éviter
- 'unsafe-eval' : à éviter
- data: : à éviter
- domaine.xx : uniquement depuis un domaine
- *.domaine.xx : depuis tous les sous domaines
- https://domaine.xx : uniquement en https depuis le domaine
- https://*.domaine.xx : uniquement en https depuis tous les sous domaines

X-Permitted-Cross-Domain-Policies

Spécifie si un fichier de règlementation interdomaines (crossdomain.xml) est autorisé. Ce fichier peut définir une règle pour accorder aux clients (comme Adobe Flash Player, Adobe Acrobat, Microsoft Silverlight ou Apache Flex) la permission de gérer des données entre domaines qui seraient autrement restreintes à cause de Same-Origin Policy.

Referrer-Policy

Indique quelles informations de provenance envoyées dans l'en-tête Referer doivent être incluses dans les requêtes effectuées.

Expect-CT

Permet de contrôler de manière stricte ou non l'adhérence aux règles de transparence des certificats, permettant ainsi de limiter les utilisations frauduleuses du certificat associé au site grâce à une vérification publique.

Implémentation

```
sudo a2enmod headers
sudo systemctl reload apache2
```

```
Header always set Strict-Transport-Security "max-age=15552001;
includeSubDomains;"
Header always append X-Frame-Options SAMEORIGIN
Header set X-Content-Type-Options nosniff
Header set X-XSS-Protection "1; mode=block"
Header set Content-Security-Policy "default-src 'self' 'unsafe-inline'
*.lab.acs.altran.fr; script-src 'self' 'unsafe-inline' 'unsafe-eval'
*.lab.acs.altran.fr"
```

```
Header set X-Permitted-Cross-Domain-Policies "none"
Header set Referrer-Policy "no-referrer"
Header set Expect-CT 'enforce, max-age=43200'
```

Debug

Pour trouver ce qui cloche lorsque la page ne s'affiche plus correctement, utiliser les outils de developpement Web du navigateur (Appuyer sur les touches Ctrl⇧ ShiftI).

> Ne pas oublier de vider le cache du navigateur si les modifications ne semble pas prises en compte.

Vérification

- https://gf.dev/secure-headers-test
- https://observatory.mozilla.org/
- https://www.ssllabs.com/ssltest/

Haut de page

Restriction d'accès

Directives

```
<Directory /var/www/html>
    AuthName 'Private'
    AuthType Digest
    AuthDigestDomain /
    AuthDigestProvider file
    AuthUserFile /var/www/data/.htdigest
    Require valid-user
</Directory>
```

Activation auth_digest

```
sudo a2enmod auth_digest
```

Création fichier htdigest

```
htdigest -c /var/www/data/.htdigest 'Private' 'vfc-training'
```

Haut de page

Reverse Proxy

Module supplémentaires

```
sudo a2enmod proxy proxy_http
```

Fichier de configuration

```
<IfModule mod_ssl.c>
        # site avec 2 chemins d'accès vers 2 machines différentes
        <VirtualHost *:443>
                ServerName site.fr
                ErrorLog ${APACHE_LOG_DIR}/site.error.log
                CustomLog ${APACHE_LOG_DIR}/site.access.log combined

                ProxyPreserveHost On
                ProxyRequests Off
                ProxyVia Off
                ProxyPass /chemin1 https://@IP1:port1/
                ProxyPassReverse /chemin1 https://@IP1:port1
                ProxyPass /chemin2 https://@IP2:port2/
                ProxyPassReverse /chemin2 https://@IP2:port2

                SSLEngine on
                SSLCertificateFile    /etc/ssl/certs/ssl-cert-snakeoil.pem
                SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
                SSLProxyEngine On
                SSLProxyCheckPeerCN Off
                SSLProxyCheckPeerName Off
                SSLProxyCheckPeerExpire off
                SSLProxyVerify none
        </VirtualHost>
        # websocket avec rewrite
        <VirtualHost *:443>
                ServerName domaine2.fr
                ErrorLog ${APACHE_LOG_DIR}/domaine2.error.log
                CustomLog ${APACHE_LOG_DIR}/domaine2.access.log combined

                ProxyPreserveHost On
                ProxyRequests Off
                ProxyVia Off
                ProxyPass / https://xxx.xxx.xxx.xxx:port2/
                ProxyPassReverse / https://xxx.xxx.xxx.xxx:port2/
```

```
                RewriteEngine On
                RewriteCond %{HTTP:Upgrade} =websocket [NC]
                RewriteRule ^/(.*)    wss://xxx.xxx.xxx.xxx:port2/$1 [P,L]

                SSLEngine on
                SSLCertificateFile    /etc/ssl/certs/ssl-cert-snakeoil.pem
                SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
                SSLProxyEngine On
                SSLProxyCheckPeerCN Off
                SSLProxyCheckPeerName Off
                SSLProxyCheckPeerExpire off
                SSLProxyVerify none
        </VirtualHost>
</IfModule>
```

> La redirection vers le serveur n'est pas obligatoirement en https (si dans le cadre d'un réseau interne). Le tunnel crypté peut s'arrêter au reverse proxy.

Dépannage

[proxy:error] Permission denied: AH00957

Sur Redhat SELinux bloque la connexion, il faut passer la commande suivante :

```
sudo setsebool -P httpd_can_network_connect 1
```

Haut de page