

**Chiffrement des communications**

Projet gsb.org

DUMAS Lucie

Table des matières

[SSL/TLS 3](#_Toc167559789)

[Qu’est-ce que SSL/TLS 3](#_Toc167559790)

[HTTPS 4](#_Toc167559791)

[Wordpress 4](#_Toc167559792)

[DokuWiki 6](#_Toc167559793)

[NextCloud 9](#_Toc167559794)

[Création de la base de données 9](#_Toc167559795)

[Configuration de l’outil NextCloud 9](#_Toc167559796)

[FTPS 16](#_Toc167559797)

# SSL/TLS

## Qu’est-ce que SSL/TLS

SSL (Secure Sockets Layer) et TLS (Transport Layer Security) sont des protocoles de sécurité qui permettent de sécuriser les communications sur un réseau, généralement Internet. Ils sont utilisés pour établir des connexions sécurisées entre un client (comme un navigateur web) et un serveur, assurant ainsi la confidentialité et l'intégrité des données échangées.

SSL a été le premier protocole de sécurité largement utilisé pour sécuriser les communications sur Internet. Cependant, en raison de vulnérabilités découvertes au fil du temps, SSL a été remplacé par TLS. Les versions spécifiques de SSL incluent SSL 2.0, SSL 3.0.

TLS est le successeur de SSL. Il a été développé pour remédier aux failles de sécurité découvertes dans les versions antérieures de SSL. TLS fonctionne de manière similaire à SSL en établissant une connexion sécurisée entre un client et un serveur. TLS est maintenant la norme utilisée pour sécuriser les communications sur Internet. Il existe différentes versions de TLS, telles que TLS 1.0, TLS 1.1, TLS 1.2, TLS 1.3, chacune apportant des améliorations en termes de sécurité et de performances.

Lorsqu'un navigateur se connecte à un site Web sécurisé, il utilise SSL ou TLS pour établir une connexion chiffrée. Cette connexion sécurisée garantit que les données transmises entre le navigateur et le serveur sont protégées contre l'interception par des tiers malveillants.

En résumé, SSL et TLS sont des protocoles de sécurité essentiels qui contribuent à assurer la confidentialité et l'intégrité des données lors de leur transmission sur InternetMise en place de ProFTPD

# HTTPS

Dans cette partie, nous devrons adapter nos différents services web afin qu’ils utilisent le protocole HTTPS.

## Wordpress

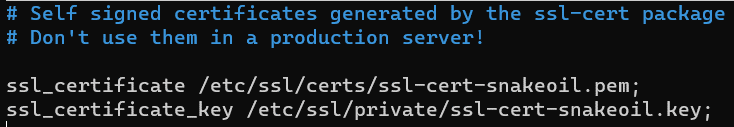
Pour permettre à Wordpress d'utiliser le protocole HTTPS, nous commençons par installer le paquet nécessaire :

apt update && apt upgrade

apt install ssl-cert -y

Cette commande installe automatiquement les clés et certificats nécessaires. Nous pouvons vérifier leur bonne installation en regardant le contenu du fichier /etc/nginx/snippets/snakeoil.conf :

cat /etc/nginx/snippets/snakeoil.conf



Notre certificat est stocké dans le fichier /etc/ssl/certs/ssl-cert-snakeoil.pem et notre clé est stockée dans le fichier /etc/ssl/privatessl-cert-snakeoil.key. Nous modifions maintenant les fichiers de configuration de nos sites internet en modifiant le bloc lié au protocole HTTP et en ajoutant un bloc lié au protocole HTTPS :

server {

listen 80;

listen [::]:80;

server\_name www.asie.gsb.org;

return 301 https://www.asie.gsb.org/;

}

server {

# SSL configuration

#

listen 443 ssl;

listen [::]:443 ssl;

include snippets/snakeoil.conf;

root /home/htdocs/gsb.org/asie/wordpress;

# Add index.php to the list if you are using PHP

index index.html index.htm index.php index.nginx-debian.html;

server\_name www.asie.gsb.org;

location / {

# First attempt to serve request as file, then

# as directory, then fall back to displaying a 404.

try\_files $uri $uri/ =404;

}

location ~ ^/~(.+?)(/.\*)?$ {

alias /home/$1/public\_html$2;

index index.html index.htm;

autoindex on;

auth\_basic "Zone securisée - Authentification requise";

auth\_basic\_user\_file /etc/nginx/.htpasswd;

}

# pass PHP scripts to FastCGI server

location ~ \.php$ {

include snippets/fastcgi-php.conf;

# With php-fpm (or other unix sockets):

fastcgi\_pass unix:/run/php/php8.2-fpm.sock;

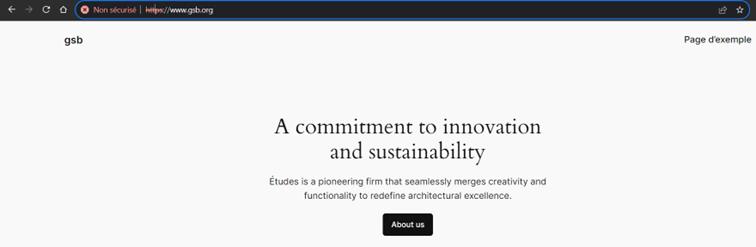
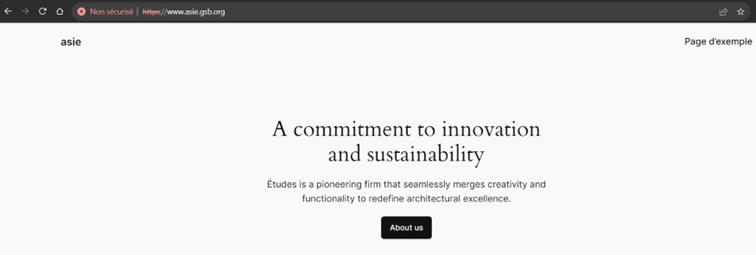
}

}

Nous appliquons ces changements dans le fichier de configuration de notre deuxième site et redémarrons notre service :

systemctl restart nginx

Nous pouvons maintenant utiliser notre navigateur pour accéder à nos sites afin de vérifier que ces derniers utilisent le protocole HTTPS :



## DokuWiki

Pour permettre à DokuWiki (et plus tard Nextcloud) d'utiliser le protocole HTTPS, nous commençons par installer le paquet nécessaire :

apt update && apt upgrade

apt install ssl-cert -y

Nous modifions le fichier de configuration de notre site DokuWiki :

server {

listen 80;

listen [::]:80;

server\_name documentation.asie.gsb.org;

return 301 https://documentation.asie.gsb.org/;

}

server {

server\_name documentation.asie.gsb.org;

listen 443 ssl;

listen [::]:443 ssl;

autoindex off;

#client\_max\_body\_size 15M;

#client\_body\_buffer\_size 128k;

index index.html index.htm index.php doku.php;

root /home/htdocs/asie.gsb.org/wiki/dokuwiki;

include snippets/snakeoil.conf;

location / {

try\_files $uri $uri/ @dokuwiki;

}

location ~ ^/lib.\*\.(gif|png|ico|jpg)$ {

expires 30d;

}

location = /robots.txt { access\_log off; log\_not\_found off; }

location = /favicon.ico { access\_log off; log\_not\_found off; }

location ~ /\. { access\_log off; log\_not\_found off; deny all; }

location ~ ~$ { access\_log off; log\_not\_found off; deny all; }

location @dokuwiki {

rewrite ^/\_media/(.\*) /lib/exe/fetch.php?media=$1 last;

rewrite ^/\_detail/(.\*) /lib/exe/detail.php?media=$1 last;

rewrite ^/\_export/([^/]+)/(.\*) /doku.php?do=export\_$1&id=$2 last;

rewrite ^/(.\*) /doku.php?id=$1 last;

}

location ~ \.php$ {

try\_files $uri =404;

fastcgi\_pass unix:/var/run/php/php8.2-fpm.sock;

fastcgi\_index index.php;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

include /etc/nginx/fastcgi\_params;

fastcgi\_param QUERY\_STRING $query\_string;

fastcgi\_param REQUEST\_METHOD $request\_method;

fastcgi\_param CONTENT\_TYPE $content\_type;

fastcgi\_param CONTENT\_LENGTH $content\_length;

fastcgi\_intercept\_errors on;

fastcgi\_ignore\_client\_abort off;

fastcgi\_connect\_timeout 60;

fastcgi\_send\_timeout 180;

fastcgi\_read\_timeout 180;

fastcgi\_buffer\_size 128k;

fastcgi\_buffers 4 256k;

fastcgi\_busy\_buffers\_size 256k;

fastcgi\_temp\_file\_write\_size 256k;

}

location ~ /(data|conf|bin|inc)/ {

deny all;

}

location ~ /\.ht {

deny all;

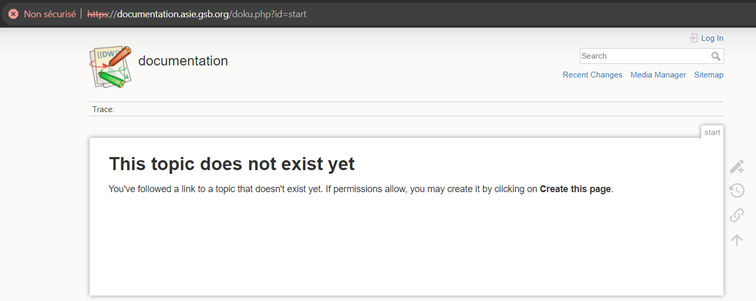
}

}

Nous redémarrons notre service :

systemctl restart nginx

A partir de notre navigateur, nous nous assurons que notre DokuWiki utilise le protocole HTTPS :



# NextCloud

## Création de la base de données

Pour pouvoir utiliser l'outil NextCloud, nous devons dans un premier temps configurer la base de données de NextCloud. Pour ce faire, nous allons créer une base de données, créer un utilisateur et attribuer des droits à cet utilisateur :

# Création de la base de données

CREATE DATABASE nextcloud;

# Création de l'utilisateur

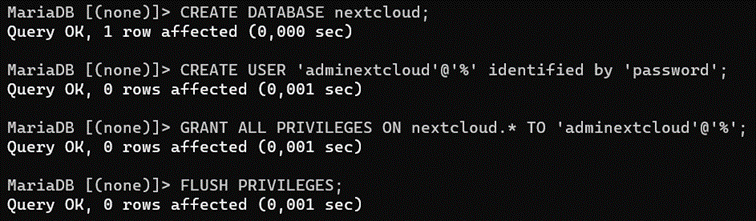
CREATE USER 'adminextcloud'@'%' identified by 'password';

# Attribution des privilèges

GRANT ALL PRIVILEGES ON nextcloud.\* TO 'adminextcloud'@'%';

# Mise à jour des droits

FLUSH PRIVILEGES;



## Configuration de l’outil NextCloud

Nous allons maintenant installer le service NextCloud. Pour ce faire, nous commençons par créer le dossier qui contiendra notre cloud :

mkdir -p /home/htdocs/gsb.org/intranet

Nous décompressons le dossier téléchargé précédemment :

unzip nextcloud.zip

Nous donnons les droits du fichier à l’utilisateur www-data :

chown -R www-data:www-data /home/htdocs/gsb.org/intranet/nextcloud

chmod 744 nextcloud

Nous créons ensuite notre fichier de configuration /etc/nginx/sites-available/intranet.asie.gsb.org :

nano /etc/nginx/sites-available/intranet.asie.gsb.org

upstream php-handler {

#server 127.0.0.1:9000;

server unix:/var/run/php/php8.2-fpm.sock;

}

# Set the `immutable` cache control options only for assets with a cache busting `v` argument

map $arg\_v $asset\_immutable {

"" "";

default "immutable";

}

server {

listen 80;

listen [::]:80;

server\_name intranet.asie.gsb.org;

return 301 https://$server\_name$request\_uri/;

}

server {

listen 443 ssl;

listen [::]:443 ssl;

server\_name intranet.asie.gsb.org;

# Path to the root of your installation

root /home/htdocs/gsb.org/intranet/nextcloud;

# Prevent nginx HTTP Server Detection

server\_tokens off;

include snippets/snakeoil.conf;

# set max upload size and increase upload timeout:

client\_max\_body\_size 512M;

client\_body\_timeout 300s;

fastcgi\_buffers 64 4K;

# Enable gzip but do not remove ETag headers

gzip on;

gzip\_vary on;

gzip\_comp\_level 4;

gzip\_min\_length 256;

gzip\_proxied expired no-cache no-store private no\_last\_modified no\_etag auth;

gzip\_types application/atom+xml text/javascript application/javascript application/json application/ld+json applica>

client\_body\_buffer\_size 512k;

# HTTP response headers borrowed from Nextcloud `.htaccess`

add\_header Referrer-Policy "no-referrer" always;

add\_header X-Content-Type-Options "nosniff" always;

add\_header X-Frame-Options "SAMEORIGIN" always;

add\_header X-Permitted-Cross-Domain-Policies "none" always;

add\_header X-Robots-Tag "noindex, nofollow" always;

add\_header X-XSS-Protection "1; mode=block" always;

# Remove X-Powered-By, which is an information leak

fastcgi\_hide\_header X-Powered-By;

include mime.types;

index index.php index.html /index.php$request\_uri;

# Rule borrowed from `.htaccess` to handle Microsoft DAV clients

location = / {

if ( $http\_user\_agent ~ ^DavClnt ) {

return 302 /remote.php/webdav/$is\_args$args;

}

}

location = /robots.txt {

allow all;

log\_not\_found off;

access\_log off;

}

location ^~ /.well-known {

location = /.well-known/carddav { return 301 /remote.php/dav/; }

location = /.well-known/caldav { return 301 /remote.php/dav/; }

location /.well-known/acme-challenge { try\_files $uri $uri/ =404; }

location /.well-known/pki-validation { try\_files $uri $uri/ =404; }

return 301 /index.php$request\_uri;

}

location ~ ^/(?:build|tests|config|lib|3rdparty|templates|data)(?:$|/) { return 404; }

location ~ ^/(?:\.|autotest|occ|issue|indie|db\_|console) { return 404; }

location ~ \.php(?:$|/) {

# Required for legacy support

rewrite ^/(?!index|remote|public|cron|core\/ajax\/update|status|ocs\/v[12]|updater\/.+|ocs-provider\/.+|.+\/ric>

fastcgi\_split\_path\_info ^(.+?\.php)(/.\*)$;

set $path\_info $fastcgi\_path\_info;

try\_files $fastcgi\_script\_name =404;

include fastcgi\_params;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

fastcgi\_param PATH\_INFO $path\_info;

fastcgi\_param HTTPS on;

fastcgi\_param modHeadersAvailable true; # Avoid sending the security headers twice

fastcgi\_param front\_controller\_active true; # Enable pretty urls

fastcgi\_pass php-handler;

fastcgi\_intercept\_errors on;

fastcgi\_request\_buffering off;

fastcgi\_max\_temp\_file\_size 0;

}

# Serve static files

location ~ \.(?:css|js|mjs|svg|gif|png|jpg|ico|wasm|tflite|map)$ {

try\_files $uri /index.php$request\_uri;

add\_header Cache-Control "public, max-age=15778463, $asset\_immutable";

access\_log off; # Optional: Don't log access to assets

location ~ \.wasm$ {

default\_type application/wasm;

}

}

location ~ \.woff2?$ {

try\_files $uri /index.php$request\_uri;

expires 7d; # Cache-Control policy borrowed from `.htaccess`

access\_log off; # Optional: Don't log access to assets

}

# Rule borrowed from `.htaccess`

location /remote {

return 301 /remote.php$request\_uri;

}

location / {

try\_files $uri $uri/ /index.php$request\_uri;

}

location ~ ^/~(.+?)(/.\*)?$ {

alias /home/$1/public\_html$2;

index index.html index.htm;

autoindex on;

auth\_basic "Zone securisée - Authentification requise";

auth\_basic\_user\_file /etc/nginx/.htpasswd;

}

}

Nous créons un lien symbolique du fichier de configuration entre le dossier des sites disponibles et le dossier des sites activés :

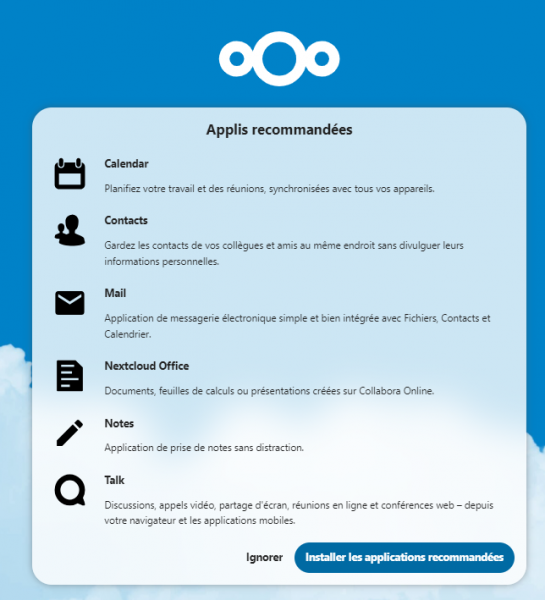
ln -s /etc/nginx/sites-available/intranet.asie.gsb.org

Nous redémarrons notre service :

systemctl restart nginx

Nous nous rendons sur l'interface web de configuration de NextCloud sur l'adresse intranet.asie.gsb.org et renseignons les informations comme le répertoire des données, les informations sur l'utilisateur NextCloud et les informations sur l'utilisateur de la base de données :





Nous avons maintenant terminé la configuration de NextCLoud.

# FTPS

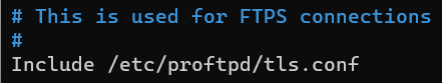
Dans cette partie, nous mettrons en place FTPS pour notre outil ProFTPD. Dans un premier temps, nous commençons par télécharger un paquet nécessaire :

apt update && apt upgrade

apt install proftpd-mod-crypto

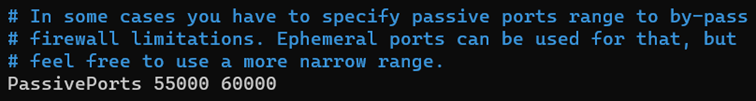
Nous modifions le fichier /etc/proftpd/proftpd.conf Pour inclure le fichier /etc/proftpd/tls.conf :

Include /etc/proftpd/tls.conf



Dans ce même fichier, nous allons activer le mode passif du serveur FTP. Le mode passif permet d'utiliser une plage de ports différents du port de transfert de fichiers (par défaut le port 20). Nous réserverons ainsi la plage de ports 55000 à 60000. Nous décommentons la ligne suivante et l'adaptons à notre plage de ports (par défaut la plage de ports est de 1024 à 65000) :

PassivePorts 55000 60000



Nous modifions maintenant les données du fichier /etc/proftpd/tls.conf :

<IfModule mod\_tls.c>

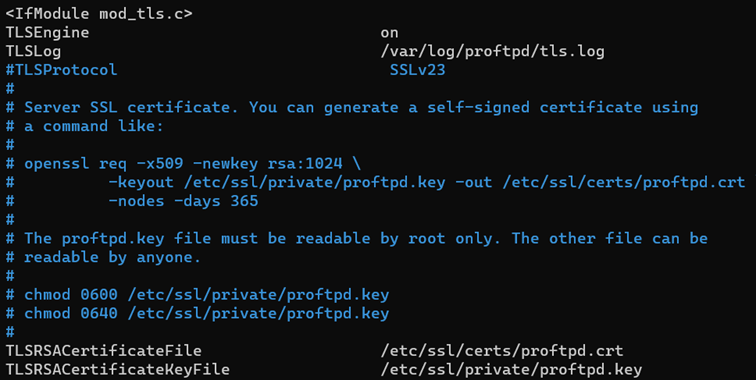
TLSEngine on

TLSLog /var/log/proftpd/tls.log

TLSRSACertificateFile /etc/proftpd/ssl/ftpcert.pem

TLSRSACertificateKeyFile /etc/proftpd/ssl/ftpkey.key

</IfModule>



Nous ajoutons ce nouveau module dans les Virtual Hosts (/etc/proftpd/virtuals.conf) :

<VirtualHost ftpin.asie.gsb.org>

ServerAdmin admin@gsb.org

ServerName "FTP INTRANET"

User intra

Group intra

Include /etc/proftpd/tls.conf

# Pour n'autoriser que le groupe extra

<Limit LOGIN>

Order allow, Deny

Allowgroup intra

Deny from all

</Limit>

# Pour les permissions lors de la création de fichier/repertoire

PassivePorts 55000 60000

Umask 022

TransferLog /var/log/proftpd/xfer/ftp-intranet.gsb.org

MaxLoginAttempts 10

DefaultRoot /srv/ftp/intranet

AllowOverwrite yes

</VirtualHost>

<VirtualHost ftpex.asie.gsb.org>

ServerAdmin admin@gsb.org

ServerName "FTP EXTRANET"

User extra

Group extra

Include /etc/proftpd/tls.conf

# Pour n'autoriser que le groupe extra

<Limit LOGIN>

Order Allow,Deny

Allowgroup extra

Deny from all

</Limit>

<Limit WRITE>

DenyAll

</Limit>

# Pour les permissions lors de la création de fichier/repertoire

PassivePorts 55000 60000

Umask 022

TransferLog /var/log/proftpd/xfer/ftp-intranet.gsb.org

MaxLoginAttempts 10

DefaultRoot /srv/ftp/extranet

AllowOverwrite yes

</VirtualHost>

Nous modifions le fichier /etc/proftpd/modules.conf afin d’activer le module gérant SSL/TLS :

LoadModule mod\_tls.c

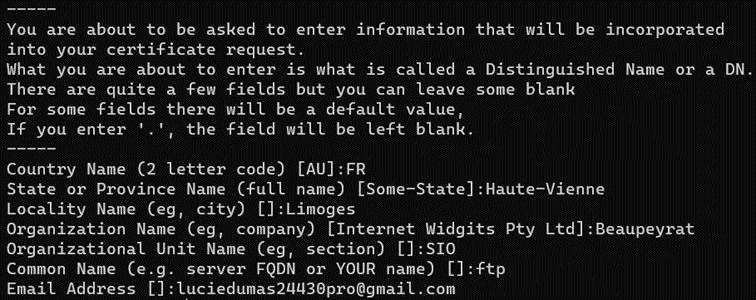


Enfin, nous devons créer un certificat SSL afin de sécuriser notre connexion :

mkdir /etc/proftpd/ssl

DIR=/etc/proftpd/ssl/

openssl req -x509 -newkey rsa:4096 -nodes -keyout $DIR/ftpkey.key -out $DIR/ftpcert.pem -days 365



Nous pouvons redémarrer le service :

systemctl restart proftpd

 Nous pouvons utiliser l'outil FileZilla sur notre client pour vérifier que le protocole FTPS est bien en place. Pour ce faire, nous nous connectons au serveur de fichiers. Un message d'avertissement concernant le certificat apparait.