

- [Accueil](#)
- Missions 1 à 5
 - [Mission 1 : Configuration réseau](#)
 - [Mission 2 : GPG](#)
 - [Mission 3 : Clonezilla](#)
 - [Mission 4 : BackupPC](#)
 - [Mission 5 : MariaDB](#)
- Missions 6 à 10
 - [Mission 6 : DHCP](#)
 - [Mission 7 : Failover](#)
 - [Mission 8 : DNS](#)
 - [Mission 9 : Nginx](#)
 - [Mission 10 : FTP](#)
- [Mission 11 : SSL/TLS](#)
- [Mission 12 : OPNsense](#)
- [Mission 13 : Zabbix](#)

SSL/TLS

I) 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 (Secure Sockets Layer):

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 (Transport Layer Security):

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 Internet

II) HTTPS

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

A) 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
```

```
# Self signed certificates generated by the ssl-cert package
# Don't use them in a production server!

ssl_certificate /etc/ssl/certs/ssl-cert-snakeoil.pem;
ssl_certificate_key /etc/ssl/private/ssl-cert-snakeoil.key;
```

Notre certificat est stocké dans le fichier `/etc/ssl/certs/ssl-cert-snakeoil.pem` et notre clé est stockée dans le fichier `/etc/ssl/private/ssl-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;
}

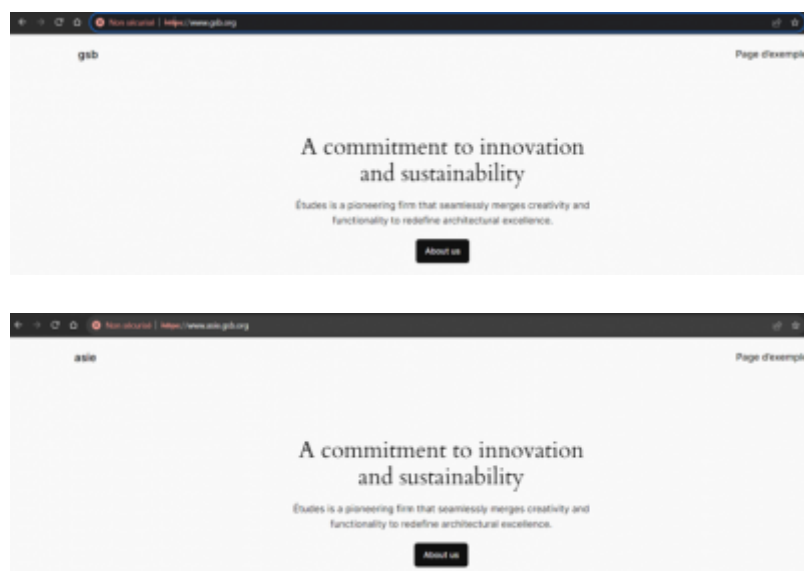
location ~ ^/~(.+?)(/.*)?$ {
    alias /home/$1/public_html$2;
    index index.html index.htm;
    autoindex on;
    auth_basic "Zone sécurisée - Authentification requise";
    auth_basic_user_file /etc/nginx/.htpasswd;
}
# pass PHP scripts to FastCGI server
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    # 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 :



B) 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;
    }

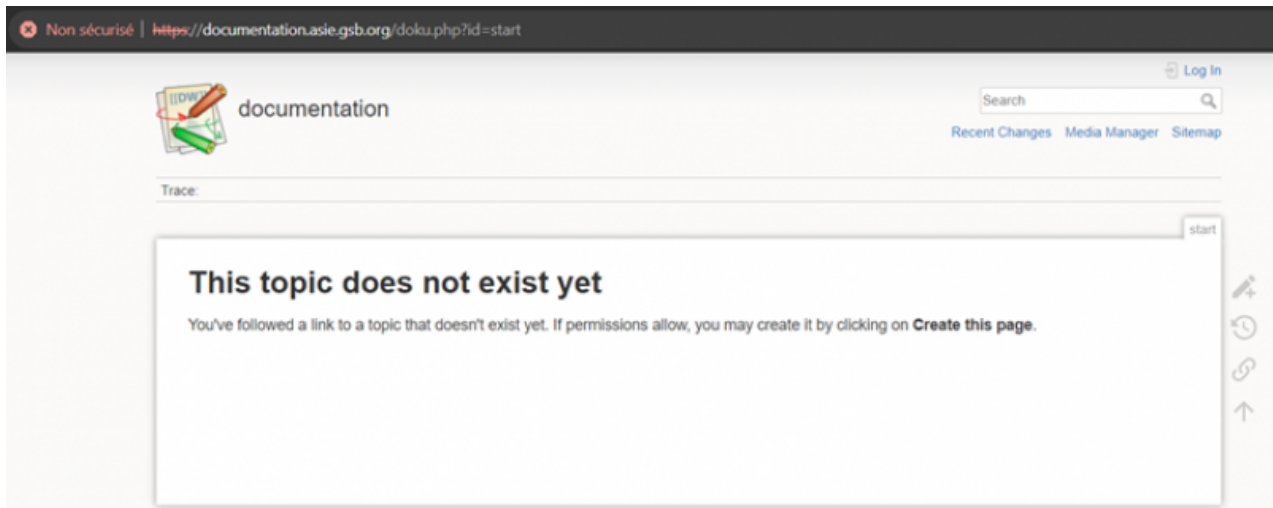
    location ~ /\.ht {
        deny all;
    }

}
```

Nous redémarrons notre service :

```
systemctl restart nginx
```

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



III) NextCloud

A) 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 'adminnextcloud'@'%' identified by 'password';

# Attribution des privilèges
GRANT ALL PRIVILEGES ON nextcloud.* TO 'adminnextcloud'@'%';

# Mise à jour des droits
FLUSH PRIVILEGES;
```

```
MariaDB [(none)]> CREATE DATABASE nextcloud;
Query OK, 1 row affected (0,000 sec)

MariaDB [(none)]> CREATE USER 'adminnextcloud'@'%' identified by 'password';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON nextcloud.* TO 'adminnextcloud'@'%';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,001 sec)
```

B) 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 téléchargeons Nextcloud depuis notre nouveau dossier :

```
curl -o nextcloud.zip  
https://download.nextcloud.com/server/releases/latest.zip
```

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;

    # HSTS settings
    # WARNING: Only add the preload option once you read about
    # the consequences in https://hstspreload.org/. This option
    # will add the domain to a hardcoded list that is shipped
    # in all major browsers and getting removed from this list
    # could take several months.
    #add_header Strict-Transport-Security "max-age=15768000;
includeSubDomains; preload" always;

    # 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>

    # Pagespeed is not supported by Nextcloud, so if your server is
built
    # with the `ngx_pagespeed` module, uncomment this line to disable
it.
    #pagespeed off;

    # The settings allows you to optimize the HTTP2 bandwitdth.
    # See
https://blog.cloudflare.com/delivering-http-2-upload-speed-improvement
s/
    # for tuning hints
    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 {
    # The rules in this block are an adaptation of the rules
    # in `.htaccess` that concern `/.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; }

    # Let Nextcloud's API for `/.well-known` URIs handle all other
    # requests by passing them to the front-end controller.
    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 sécurisée - 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 :

Créer un compte administrateur

Nom d'utilisateur

Mot de passe

Stockage & base de données ▾
Répertoire des données

Configurer la base de données

Seul(e) MySQL/MariaDB est disponible. Installez et activez les modules PHP additionnels adéquats pour choisir d'autres types de base de données.

Consultez la documentation pour plus de détails. [↗](#)

Utilisateur de la base de données

Mot de passe de la base de données

Nom de la base de données

Hôte de la base de données

Veuillez spécifier le numéro du port avec le nom de l'hôte (ex: localhost:5432).

Installer



Nous avons maintenant terminé la configuration de NextCloud.

IV) FTPS

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

```
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
```

```
# This is used for FTPS connections
#
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 pour décommenter la ligne

suivante et l'adapter à notre plage de ports (par défaut la plage de ports est de 1024 à 65000) :

```
PassivePorts 55000 60000
```

```
# In some cases you have to specify passive ports range to by-pass
# firewall limitations. Ephemeral ports can be used for that, but
# feel free to use a more narrow range.
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

TLRSACertificateFile                    /etc/proftpd/ssl/ftpcert.pem
TLRSACertificateKeyFile                 /etc/proftpd/ssl/ftpkey.key
</IfModule>
```

```
<IfModule mod_tls.c>
TLSEngine                                on
TLSLog                                  /var/log/proftpd/tls.log
#TLSProtocol                            SSLv23
#
# Server SSL certificate. You can generate a self-signed certificate using
# a command like:
#
# openssl req -x509 -newkey rsa:1024 \
#             -keyout /etc/ssl/private/proftpd.key -out /etc/ssl/certs/proftpd.crt
#             -nodes -days 365
#
# The proftpd.key file must be readable by root only. The other file can be
# readable by anyone.
#
# chmod 0600 /etc/ssl/private/proftpd.key
# chmod 0640 /etc/ssl/private/proftpd.key
#
TLRSACertificateFile                    /etc/ssl/certs/proftpd.crt
TLRSACertificateKeyFile                 /etc/ssl/private/proftpd.key
```

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>
```

```
<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
    Unask          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
```

```
# Install proftpd-mod-crypto to use this module for TLS/SSL support.
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
```

```
-----
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]:FR
State or Province Name (full name) [Some-State]:Haute-Vienne
Locality Name (eg, city) []:Limoges
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Beaupeyrat
Organizational Unit Name (eg, section) []:SIO
Common Name (e.g. server FQDN or YOUR name) []:ftp
Email Address []:luciedunas24438pro@gmail.com
```

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 apparaît.



From:
<https://sisr2.beaupeyrat.com/> - Documentations SIO2 option SISR

Permanent link:
<https://sisr2.beaupeyrat.com/doku.php?id=sisr2-asie:mission11>

Last update: **2024/04/02 15:06**

