SERVEUR APACHE

Table des matières

INTRODUCTION AU COURS

Un peu d'histoire

Internet désigne l'interconnexion mondial des réseaux. Il est créé au milieu des années 60 au sein du ministère de la défense américaine.

Le World Wide Web désigne le système hypertexte constitué par l'ensemble des pages des serveurs utilisant le protocole http (Hypertexte Transfer Protocol). Il est créé en 1989 au CERN à Genève.

Quelques dates

1993 : Mosaic, 1er navigateur graphique

1994 : Netscape

1995 : Microsoft Internet Explorer

2004 : Mozilla Firefox

Le protocole http

1997, version 1.1, révisée en 1999

Le dénouement d'une connexion :

- Connexion du client au serveur
- Envoi d'une requête GET au serveur
- Réponse du serveur
- Envoi d'une seconde requête pour faire la connexion
- Deuxième réponse du serveur
- Fermeture de la connexion

Requêtes les plus courantes

- GET : pour récupérer un document
- HEAD : pour récupérer seulement l'entête
- POST : envoi de données au serveur
- PUT : envoi d'un fichier
- DELETE : suppression d'un fichier
- CONNECT : accès sécurisé (HTTPS)

Informations contenues dans les entêtes

Les entêtes contiennent de nombreuses informations :

- HOST : indique au serveur quel hôte virtuel on veut interroger
- USER-AGENT : nom du navigateur utilisé
- ACCEPT : le format du fichier accepté
- ACCEPT-LANGUAGE : langue acceptée
- AUTHORIZATION : les informations d'authentification sur le serveur
- CONNECTION : pour fermer la connexion
- DATE : date de la réponse
- SERVER : le logiciel du serveur
- CONTENT-TYPE : format des données envoyées
- CONTENT -LENGTH : taille du fichier envoyé

Catégories des codes réponses du serveur

- Du 100 au 199 : messages informatifs
- Du 200 au 299 : succès de la requête
- Du 300 au 399 : redirections
- Du 400 au 499 : problèmes
- 401 : requête non autorisée
- 404 : le fichier n'existe pas
- 405 : méthode GET ou POST incorrecte
- Du 500 au 599 : erreurs internes au serveur

La fondation apache

La Apache Software Fondation est une organisation à but non lucratif dont l'objet est de développer des logiciels libres. Elle est née en 1995 pour améliorer les serveurs HTTPS du NCSA. Elle n'a pas de salariés, seulement des bénévoles sur une centaines de projets dont HTTP Apache, LDAP Apache Directory (2002), OpenOffice.org (2011), SpamAssassin (2004) …

La licence Apache est une licence libre compatible avec la GNU/GLP, mais on n'est pas obligé de republier son code sous licence Apache : il peut rester propriétaire au sein de la formation.

Http apache

Version 2.2 en 2005, 2.4 en 2012

Plus de 80 % des serveurs dans le monde. Conception modulaire : par exemple le mod_ssl permet le HTTPS. Les hôtes virtuels sont inclus dans le module de base : core. Plusieurs modules concernent la journalisation (logs), l'authentification, etc.…

Des petites questions …

Fichier /etc/apache2/apache2.conf

Quelle directive spécifie le port d'écoute ? Quel est le port par défaut ?

- Le port d'écoute est le 80 pour Apache

Y-a-t-il d'autres fichiers de configuration changé par apache2.conf ? Lesquels ? Ou sont-ils ?

Quel est le répertoire racine dans lequel sont les documents HTML ?

/var/www/html

Quelle page Apache est renvoyée lorsque l'URL demandés correspond à un dossier ?

Comment sont traités les URL de la forme http://serveur/nom-dossier/fichier.html

INSTALLATION ET CONFIGURATION DE WAMP

Wamp = Windows Apache MySQL PHP

Apache : serveur Web libre le plus utilisé (>20%)

INSTALLATION

Récupérer les archives :

-Visual studio

-WampServer

On rajoute aussi un éditeur avec une coloration syntaxique : Note pad++ ou Visualstudio-code est autre.

Lancer WampServer : icône en bas à droite W qui doit passer en rouge, orange et vert. 

Vérifier qu'il est en français (clique droit) et tester (lancer localhost, clic gauche)

Localhost indique toujours la machine sur laquelle on se trouve : pas besoin de connaitre son adresse IP. Si la page web s'affiche avec la version du serveur ça marche !

UTILISATION

Essayer d'accéder à votre serveur à partir d'une autre machine

http://IP_SERVEUR

Si cela ne marche pas il faut ouvrir le port http (port 80) dans le pare-feu

Dans le fichier de configuration httpd.conf qui se trouve ici

Voici l'intérieur du fichier :

```
<Directory />
    AllowOverride none
    Require all denied
</Directory>
```

AllowOverride none = interdit toute directive de configuration

Require all denied = interdit toute connexion externe

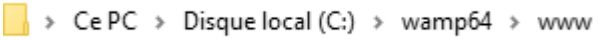
Nous devrons modifier ce code pour que le serveur puisse communiquer avec le client

```
<Directory />
    AllowOverride all
    order deny, allow
    allow from localhostallow from 192.168.*.*
</Directory>
```

Order donne l'ordre des directions d'accès

Allow from localhostallow from 192.168.*.* = autorise toute les ip de 192.168.*.*

Le dossier du serveur se trouve dans



Dans ce dossier, créer un fichier test1.html :

```
<html>
    <head>
        <meta charset = « UTF-8 » />
```

```
        <title> Teste du fichier HTML</title>
    </head>
<body>
    <h1> Ceci est un titre de niveau 1</h1>
    <h2>Ceci est un titre de niveau 2</h2>
    <div>la balise div permet d'écrire du texte</div>
</body>
</html>
```

INSTALLATION ET CONFIGURATION DE LAMP

Lamp = linux Apache MySql PhP

INSTALLATION APACHE

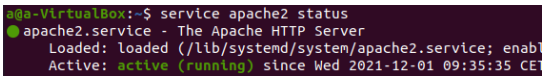
Raspberry :

- apt install apache 2

Sur Raspberry après avoir installé apache2 la commande pour le lancer :

-service apache2 start

-service apache2 status #pour afficher le statut du service



-service apache2 restart #pour reboot le service

-service apache2 stop #pour stopper le service

VERIFICATION

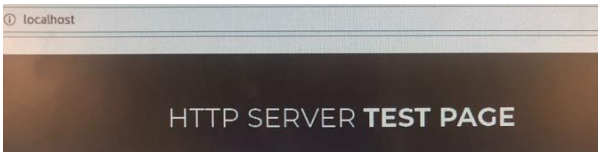
Sur une page internet lancer :

http://localhost

Si une page s'ouvre alors c'est que le service apache2 fonctionne

Si du côté du client il ne trouve pas le server on peut essayer d'ajouter cette commande ci-dessous

-sudo firewall-cmd –permanent –add-service=http



INSERER DU HTML DANS APACHE PAR LE RASBERRY

Pour insérer une page d'HTML dans un serveur apache il faut se rendre dans le dossier var

-cd /var/www/html

-sudo nano test1.html

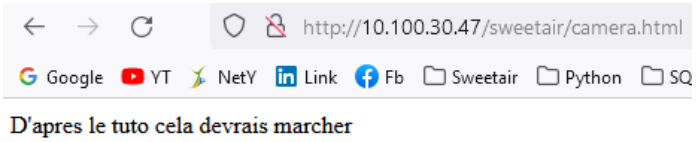
Ensuite une page va s'ouvrir et on va pouvoir écrire notre texte

VERIFICATION DE LA LECTURE HTML PAR LE SERVEUR

Pour vérifier si notre page html fonctionne bien nous allons sur une page internet et nous tapons

http://IP/MonFICHIER.html ou http://IP/NomDudossier/Fichier.html

Exemple http://10.100.30.47/sweetair/camera.html



PHP

Pour Raspberry rentrer la commande :

-apt install php

Nous allons rajouter un fichier de test dans le dossier /var/www/html : phpinfo.php
Sudo nano /var/www/html phpinfo.php #ecrire le code ci-dessous dans le fichier phpinfo.php

```
< ? php
        Phpinfo() ;
?>
```

vérification du php

Pour vérifier si le PHP fonctionne correctement nous rentrons l'adresse IP du serveur et nous rajoutons phpinfo.php comme s'en suit l'exemple http://10.100.30.47/phpinfo.php

PHP Version 7.4.25

System	Linux raspberrypi 5.10.63+ #1459 Wed Oct 6 16:40:27 BST 2021 armv6l
Build Date	Oct 23 2021 21:53:50
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-ffi.ini, /etc/php/7.4/apache2/conf.d/20-fileinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-json.ini, /etc/php/7.4/apache2/conf.d/20-mysqli.ini, /etc/php/7.4/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.4/apache2/conf.d/20-phar.ini, /etc/php/7.4/apache2/conf.d/20-posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20-shmop.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.ini, /etc/php/7.4/apache2/conf.d/20-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902,NTS
PHP Extension Build	API20190902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

MYSQL

On rajoute MySQL ou Maria DB

SQL : Standard Query Language (Language d'interrogation des bases de données)

INSTALLATION MYSQL

Raspberry :

-sudo apt install mariadb-server php-mysql

Redémarrer le service apache2

VERIFCATION SQL

Pour la vérification nous retournons sur notre page HTML et il faut descendre les tableaux, si MSQL s'affiche c'est que MySQL est bien installé comme ci-joint sur la photo

mysqli

Mysqli Support	enabled
Client API library version	mysqlnd 7.4.15
Active Persistent Links	0
Inactive Persistent Links	0
Active Links	0

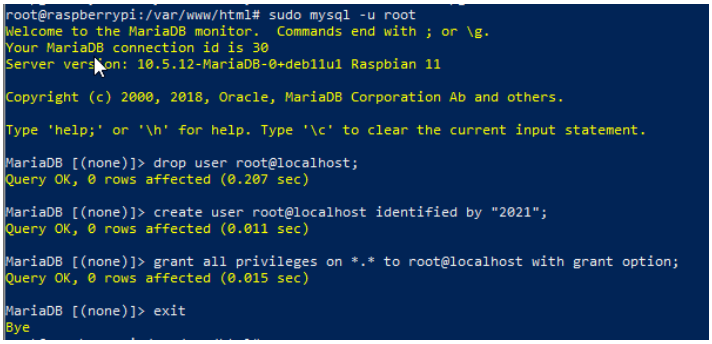
Directive	Local Value	Master V
mysqli.allow_local_infile	Off	Off
mysqli.allow_persistent	On	On
mysqli.default_host	*no value*	*no value*
mysqli.default_port	3306	3306
mysqli.default_pw	*no value*	*no value*
mysqli.default_socket	*no value*	*no value*
mysqli.default_user	*no value*	*no value*
mysqli.max_links	Unlimited	Unlimited
mysqli.max_persistent	Unlimited	Unlimited
mysqli.reconnect	Off	Off
mysqli.rollback_on_cached_plink	Off	Off

mysqlnd

mysqlnd	enabled
Version	mysqlnd 7.4.15
Compression	supported
core SSL	supported
extended SSL	supported

COMMANDE SQL

Commandes SQL pour accéder aux bases de données :



-sudo mysql -u root
>drop user root@localhost;
>create user root@localhost identified by « 2021 » ;
>grant all privileges on *.* to root@localhost with grant option ;
>exit

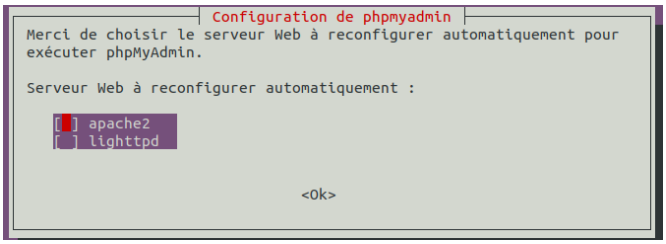
= Traduction informatique

#supprimer l'user root
#crer l'user root avec motdepasse 2021
#donner tout privilèges à la base, + user root toute option
#sortir

COMMANDE PHPMYADMIN

Pour gérer plus facilement les bases de données que dans une console, on ajoute PhpMyAdmin sur le serveur apache :

-sudo apt install phpmyadmin



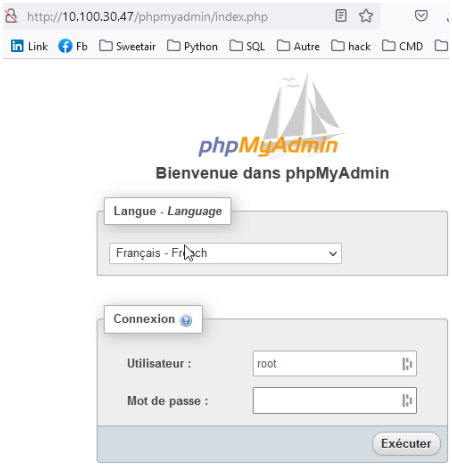
Créer un lien avec le serveur apache

-sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin

Vérification dans /var/www/html

Redémarrer le serveur Apache

Tester dans le navigateur : http//localhost/phpmyadmin

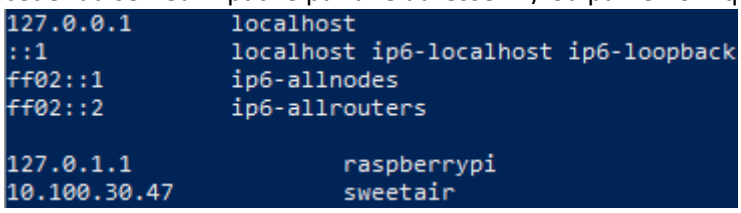


Fichier hosts (accès page apache d'un autre client)

Pour accéder au fichier HOSTS :

sudo nano /etc/hosts

Ce fichier va permettre d'accéder au serveur Apache par une adresse IP / ou par le nom que on leur donne



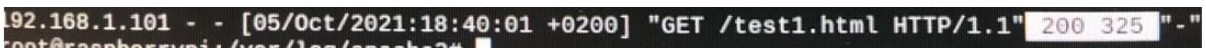
Exemple : dans la barre de recherche après avoir associé l'adresse IP de momodu38 avec son nom je tape

-http://sweetair #Cela nous ramène sur sa page apache

Logs des accès au page internet apache

Pour consulter les logs de la page internet, voir quel fichier a été ouvert ou autre on rentre la commande

-var/log/apache2/access.log



Pour l'exemple mathias s'est connecté sur ma page HTML afin que j'ai un log, nous pouvons voir que son adresse IP est 192.168.1.101 et qu'il a ouvert mon fichier test1.html

Ce qui est surligné en blanc ce sont les catégories de codes réponses du serveur, 200 = réussite + 325 = redirections

Création de dossier et fichier

Créer un dossier webftp dans var /www/html

-sudo mkdir /var/www/html/webftp

Faire une copie dedans du fichier test1.html
-cp test1.html /var/www/html/webftp

Une copie de test1 serra faite dans webftp
On peut y accéder avec le navigateur

Renommez le fichier en index.html
-mv test1.html index.html

Quand nous retournons sur la page web le fichier s'ouvre directement.

Pourquoi ?

le serveur Apache est configuré par défaut pour trouver dans un dossier le fichier index.html ou index.php. S'il n'existe pas Apache liste le dossier. Si on ne veut pas qu'Apache liste les dossiers alors il faut changer la configuration par défaut du dossier dans le fichier apache2.conf :

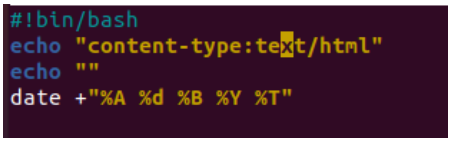
<directory /var/www/html/webftp>
Option -indexes
<directory>

Script cgi

On crée un script nommé date.cgi qui renvoie la date et l'heure dans une page html
-/var/www/cgi-bin

Le script :

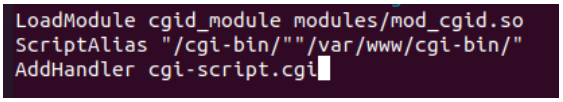
#!/bin/bash
echo "content-type:txtt/html"
echo ""
date +"%A %d %B %Y %T"



Et le rendre exécutable :
-chmod +x date.cgi

Ajouter une ligne de le httpd.conf (apache2.conf)

LoadModule cgid_module modules/mod_cgid.so
ScriptAlias "/cgi-bin/" "/var/www/cgi-bin/"
AddHandler cgi-script .cgi



Pour les Raspberry, le module CGI n'est pas chargé par défaut.
Vérifier qu'il se trouve dans /etc/apache2/mods-available # cgi.load

Vérifier qu'il n'est pas dans /etc/apache2/mods-enabled

```
ccess_compat.load  authz_user.load  filter.load       php7.4.load
lias.conf          autoindex.conf   mime.conf         reqtimeout.conf
lias.load          autoindex.load   mime.load         reqtimeout.load
uth_basic.load     deflate.conf     mpm_prefork.conf  setenvif.conf
uthn_core.load     deflate.load     mpm_prefork.load  setenvif.load
uthn_file.load     dir.conf         negotiation.conf  status.conf
uthz_core.load     dir.load         negotiation.load  status.load
uthz_host.load     env.load         php7.4.conf
@a-VirtualBox:/etc/apache2/mods-enabled$
```

Comme nous pouvons le voir le CGI est bien dans AVAILABLE et pas dans ENABLED
Rajouter le lien

-ln -s /etc/apache2/mods- available /cgi.load cgi.load

```
x:~$ ln -s /etc/apache2/mods-available/cgi.load cgi.load
```

Modifier la configuration

ScriptAlias « cgi-bin » « /var/www/cgi.bin »
AddHandler cgi-script .cgi

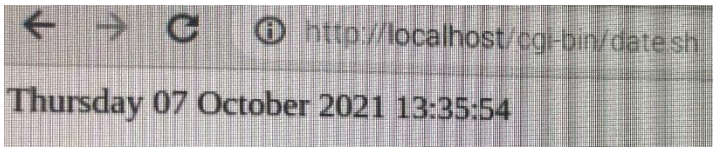
Redémarrer Apache

Modifier la configuration pour que les fichiers .sh soient exécutées comme des requêtes CGI
Rajouter au nano apache2.conf le .sh

Copier le fichier date.cgi et le coller en date.sh

```
a@a-VirtualBox:/var/www/cgi-bin$ sudo cp date.cgi date.sh
```

Redémarrer le serveur
Dans localhost taper date.sh



cgi utilisateur processus

Formulaire HTML simple : user.html
CGI pour afficher les processus utilisateurs

```
<html>
        <head>
                <title> user </title>
        </head>
        <body>
                <form method= "get" action="/cgi-bin/listeproc.sh">
                        <input type="text" name="user" />
                </form>
        </body>
</html>
```

10.1.40.178/user.html

Dans le champ de recherche, rechercher un utilisateur existant au serveur, pc local a la machine hôte

pi

Ensuite, il nous affiche tous les processus en cours d'utilisation de l'utilisateur

```
PID TTY          TIME CMD
690 ?        00:00:00 systemd
691 ?        00:00:00 (sd-pam)
701 ?        00:00:00 lxsession
709 ?        00:00:00 dbus-daemon
720 tty1     00:00:00 bash
749 ?        00:00:00 ssh-agent
773 ?        00:00:00 gvfsd
780 ?        00:00:00 gvfsd-fuse
785 ?        00:00:00 openbox
788 ?        00:00:00 lxpolkit
792 ?        00:00:04 lxpanel
797 ?        00:00:01 pcmanfm
805 ?        00:00:00 ssh-agent
807 ?        00:00:00 xcompmgr
818 ?        00:00:01 applet.py
```

Création répertoire prive

Créer un répertoire privé dans /var/www/html
On veut limiter son accès à un ensemble limité de comptes Apache
-créer le dossier et mettre un fichier html. Vérifier qu'il est accessible de partout

-créer dans le dossier un fichier caché .htaccess

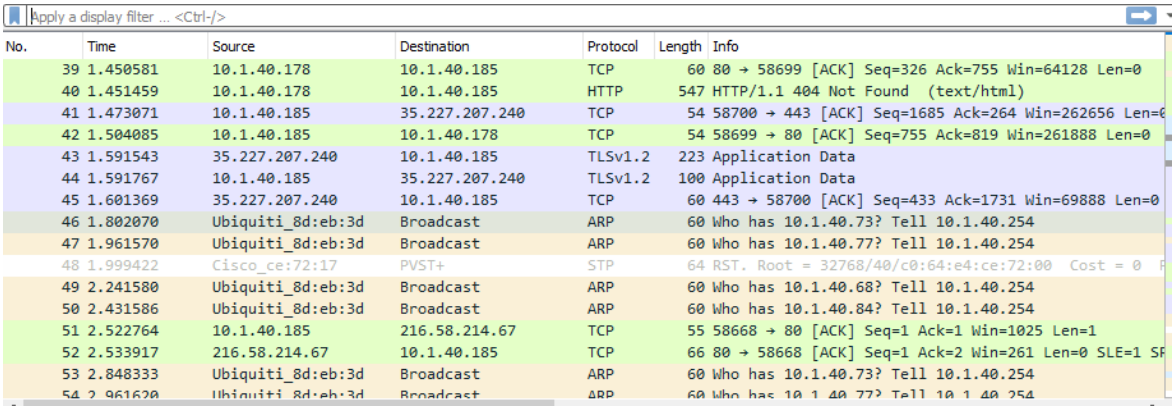
Tester avec le ls et le ls -a
Créer les comptes
-cd /etc/apache2
-htpasswd -c users admin

Ligne de commande à rentrer pour créer l'utilisateurs admin
Le -c permet de créer le fichier, pour rajouter des utilisateurs on peut enlever le -c
Ici nous venons de créer un nouvel utilisateur admin

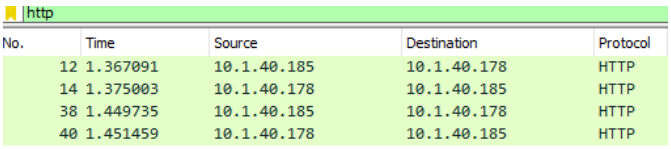
```
root@raspberrypi:/var/www/html# htpasswd -c users admin
New password:
Re-type new password:
Adding password for user admin
```

Wireshark

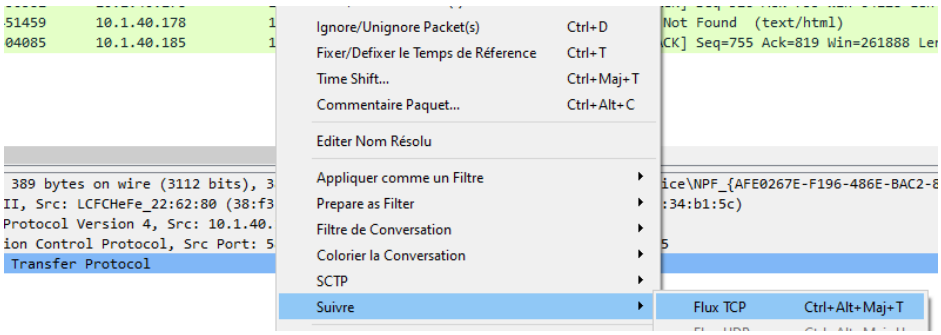
Vider le cache Firefox
Lancer une capture Wireshark est se connecter par l'adresses IP au serveur APACHE



C'est très illisible, pour réduire le nombre de ligne on tape dans la barre de recherche en haut http



Voici ce qu'il en reste on va les traiter une à une et regarder la ligne ou il y a écrit AUTHORIZATION

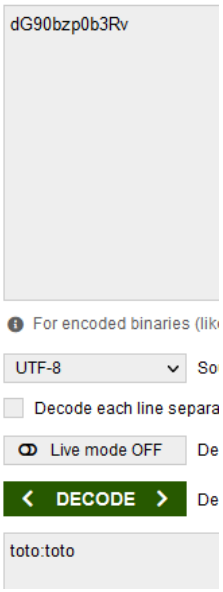


On clique droit /suivre/FluxTCP

```
*;q=0.8
Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Authorization: Basic dG90bzp0b3Rv

HTTP/1.1 200 OK
Date: Thu, 21 Oct 2021 07:22:09 GMT
Server: Apache/2.4.38 (Raspbian)
Last-Modified: Wed, 20 Oct 2021 08:31:52 GMT
ETag: "28-5cec49c2585a8"
Accept-Ranges: bytes
Content-Length: 40
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

<header>
```



Ici la ligne Authorization nous donne le mot de passe encodé en base 64
Pour le désencodé on va sur internet pour recherche un décodeur de base64 ➔

Serveur virtuel

Sert à héberger plusieurs site web ou local différents

1) Donner plusieurs noms à votre machine
2) Fichier etc/hosts : rajouter sweetair.org
3) Tester avec le navigateur

Configuration hôte virtuel dans apache2.conf

NameVirtualHost 10.x.x.x
<VirtualHost 10.x.x.x>
    DocumentRoot /var/www/html/sweetair
    ServerName sweetair.org
</VirtualHost>

HTTPS ou SSL

Par défaut, apache utilise le protocole non sécurisé http sur le port 80. Passons-le en HTTPS qui écoute sur le port 443

On active le mode ssl et le site par défaut default-ssl :

-a2enmod ssl

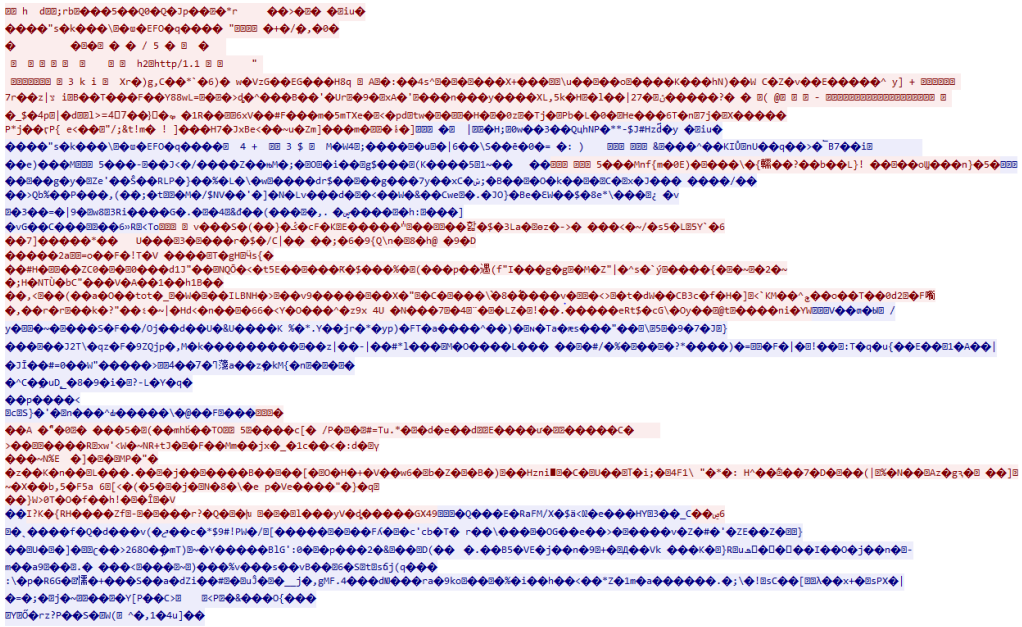
-a2ensite default-ssl

Redémarrer le serveur apache2

Vérifier qu'il fonctionne aussi en https et contrôler le chiffrement avec WireShark

No.	Time	Source	Destination	Protocol	Length	Info
27	4.512574	10.1.40.185	10.1.40.178	TCP	66	64521 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
28	4.513713	10.1.40.178	10.1.40.185	TCP	66	443 → 64521 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
29	4.513875	10.1.40.185	10.1.40.178	TCP	54	64521 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
30	4.517292	10.1.40.185	10.1.40.178	TLSv1.3	675	Client Hello
31	4.518187	10.1.40.178	10.1.40.185	TCP	60	443 → 64521 [ACK] Seq=1 Ack=622 Win=64128 Len=0
32	4.524879	10.1.40.178	10.1.40.185	TLSv1.3	294	Server Hello, Change Cipher Spec, Application Data, Application Data
33	4.525650	10.1.40.185	10.1.40.178	TLSv1.3	118	Change Cipher Spec, Application Data
34	4.526118	10.1.40.178	10.1.40.185	TCP	60	443 → 64521 [ACK] Seq=241 Ack=686 Win=64128 Len=0
35	4.527131	10.1.40.178	10.1.40.185	TLSv1.3	325	Application Data
36	4.531084	10.1.40.185	10.1.40.178	TLSv1.3	593	Application Data
37	4.532015	10.1.40.178	10.1.40.185	TCP	60	443 → 64521 [ACK] Seq=512 Ack=1225 Win=64128 Len=0
38	4.535285	10.1.40.178	10.1.40.185	TLSv1.3	401	Application Data
39	4.579659	10.1.40.185	10.1.40.178	TCP	54	64521 → 443 [ACK] Seq=1225 Ack=859 Win=261888 Len=0
41	4.660449	10.1.40.185	10.1.40.178	TLSv1.3	493	Application Data
42	4.661096	10.1.40.178	10.1.40.185	TCP	60	443 → 64521 [ACK] Seq=859 Ack=1664 Win=64128 Len=0
43	4.661817	10.1.40.178	10.1.40.185	TLSv1.3	570	Application Data
44	4.703637	10.1.40.185	10.1.40.178	TCP	54	64521 → 443 [ACK] Seq=1664 Ack=1375 Win=261376 Len=0

Une fois la bonne tram trouvée nous pouvons remarquer, quelle est totalement illisible



Nous n'avons pas eu besoin de générer le certificat ssl : il y en a un par défaut, valable 10 ans
Trouvez le dans le fichier :
/etc/apache2/sites-avaible/default-ssl

Votre site fonctionne dans les 2 modes, enlever le site http
-a2dissite default

On veut rediriger automatiquement toute les requêtes http vers HTTPS. Modifier le fichier /etc/apache2/sites-avaible/default
Redirect permanent
https://votredomaine.fr

MySQL

MySQL est un SGBD (system de gestion de base de données) libre, complémentaire de PHP pour créer des sites Web dynamiques, c'est-à-dire dont les pages sont fabriquées à la demande

Il existe d'autres SGBD, Microsoft SQL server, oracle, PostgreSQL sont des SQL libres exploitant le langage SQL. Suite au rachat de la société MySQL par Sun Microsystems en 2009, puis par oracle , le projet libre s'est transformé en MariaDB soutenu par une fondation indépendante

Console MySQL

-MySQL -u root -p

Créons une base films et donnons lui un administrateur

```
CREATE DATABASE Films ;
GRANT ALL PRIVILEGES ON Films.* TO adminFilms@localhost IDENTIFIED BY ‘ Films ' ;
```

On se déconnecte de root et on se reconnecte comme admin films :

```
Exit
```

```
mysql -u adminFilms -p Films
```

Créons une première table

```
CREATE TABLE FilmSimple (
titre           VARCHAR(30),
annee           INTEGER
) ;
```

La commande DESC FilmSimple ; permet d’afficher la structure de la table et la commande DROP TABLE permet de le supprimer.

Pour remplir la table :

```
INSERT INTO FilmSimple
(titre, annee)
VALUES
('Pulp Fiction', 1996) ,
('Alien', 1979) ,
('Titanic', 1997) ,
('Eiffel', 2021) ;
```

On peut donc maintenant interroger la base de données par des requêtes. Par exemple :

```
SELECT titre FROM FilmSimple WHERE annee='1997' ;
```

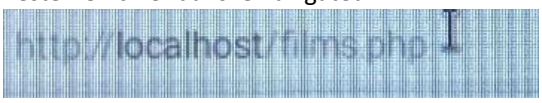
Lien avec PHP

Comment utiliser la base Films ?

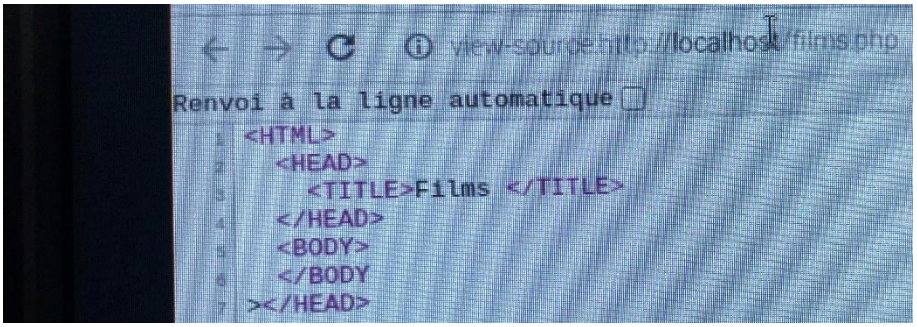
En PHP :

```
< ?php
Echo « <HTML>  » ;
Echo « <HEAD>  » ;
Echo « <TITLE>  Films </TITLE>»  ;
Echo « </HEAD>  » ;
Echo « <BODY>  » ;
Echo « </BODY>  » ;
Echo « </HTML>  » ;
?>
```

Tester le fichier dans le navigateur



Avec le CRL + U



Avec le PHP/SQL :

On insert ce texte dans le BODY

```
$connexion = mysqli_connect(localhost, adminFilms, motdepasse, Films) ;
$requete = "select * from FilmSimple" ;
$resultat = mysqli_query($connexion, $requete) ;
```

```
while ($film = mysqli_fetch_assoc($resultat) {
echo "<P>" . $film['titre'] . ", paru en " . $film['annee'] . "</P>\n" ;
}
mysqi_close() ;
```

```php
<?php
echo "<HTML>\n";
echo "  <HEAD>\n";
echo "    <TITLE>Films </TITLE>\n";
echo "  </HEAD>\n";
echo "  <BODY>\n";
$connexion = mysqli_connect(localhost, adminFilms, Films, Films);
$requete = "select * from FilmSimple ";
$resultat = mysqli_query($connexion, $requete);
while ($film = mysqli_fetch_assoc($resultat)) {
echo "<P>" .$film['titre'].",paru en ".$film['annee']. "</P>\n";
}
mysqli_close();
echo "  </BODY>\n";
echo "</HEAD>\n";
?>
```

On vérifie par la page web :