

Projet Informatique 1^{ère} Année
CSC 3502



Application Android pour la ville intelligente

Mauriès Benjamin-William
De Logivière Louis
Oelhoffen Maxime
Lejot Florent

Enseignant responsable : **Afifi Hossam**

23/05/12

Sommaire

Introduction : Le Projet Wonderville	3
1. Cahier des charges : portail captif et page HTML	5
2. Déroulement du Projet.....	7
2.1. Installation de CoovaChili	9
2.2. TP : Création et utilisation d'un portail captif	12
2.2.1. Etape n°1 : Désactivation d'options du point d'accès	13
2.2.2. Etape n°2 : Mise en place du portail captif	16
2.3. Création d'une page HTML d'authentification	19
1. L'affichage de l'heure	20
2. Le compteur de visite	20
3. Le compteur d'utilisateurs du portail captif	21
4. Protection de la page infos.php.....	23
5. Champs de saisie	24
6. Incorporation de la page à Coovachilli.....	24
2.4. Vérifications des codes et tests	25
2.5. Hébergement sur le web et en local	27
3. Conclusion	28
4. Bibliographie	29
5. Annexes	30
5.1. Annexe 1 : Gestion de projet	30
5.1.1. Plan de charge	30
5.1.2. Planning Prévisionnel.....	31
5.2. Annexe 2 : Code source	32

Introduction : Le Projet Wonderville

Le projet Wonderville, projet de ville intelligente en partenariat avec la mairie de Montreuil, consiste à faire interagir les citoyens avec les organismes administratifs, mais également entre eux, en échangeant des informations sur la vie administrative, culturelle, sociale ou encore météorologique de la ville. On peut imaginer des applications dans des domaines divers, qu'ils soient culturels, évènementiels ou encore sportifs (achats de billets, réservations, abonnements, paiements...).

Cet échange d'informations se fait au travers de « borne », installées dans tout la ville, avec lesquels les utilisateurs peuvent échanger de l'information via un smartphone ou un ordinateur personnel. Les informations seront stockées sur chaque borne et sont destinées à être transmises au serveur central de la mairie.

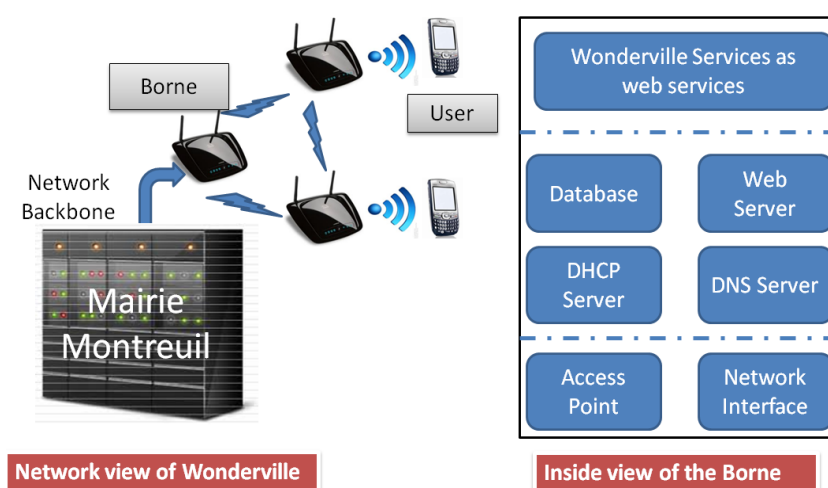


Schéma 1.1 : Interaction entre borne et réseau

En revanche, les bornes ne sont pas connectées au réseau Internet mais sont interconnectées entre elles par des liaisons radio. La plupart des connexions Internet, et notamment l'envoi des informations enregistrées dans les bornes vers le serveur central, se fait par l'intermédiaire de la connexion 3G des smartphones des utilisateurs. L'intérêt de ce procédé est qu'il permet aux bornes de rester indépendantes de tout système de connexion à Internet, coûteux, énergivore et contraignant.

L'interface entre les bornes et les smartphones pourra se faire au travers d'applications Android permettant la connexion au réseau et l'échange de données (en émission et réception) avec les différentes bornes.

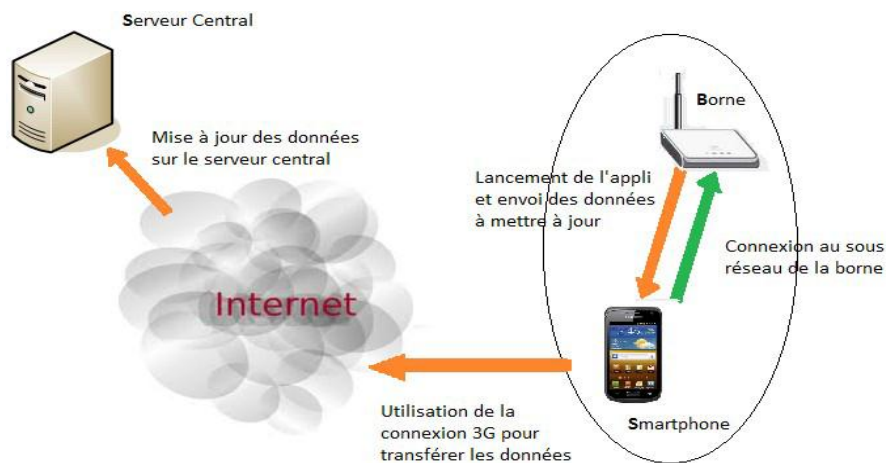


Schéma 2.1 : Connexion à la borne avec son smartphone

La connexion entre l'utilisateur et les bornes peut se faire via des liaisons Wi-Fi ou Bluetooth. Pour accéder aux informations comprises dans les bornes et éventuellement en publier, il est nécessaire que l'utilisateur s'identifie. Toute connexion non identifiée est donc à priori bloquée. Le blocage de la connexion et l'autorisation d'accès se font par l'intermédiaire d'un portail captif. Une fois identifiés et autorisés par le portail, les clients peuvent accéder au réseau des bornes protégé.

Ce système de portail captif est un système très courant pour contrôler l'accès à un certain réseau, particulièrement le réseau internet. Il est notamment présent au sein du campus de Telecom et Management SudParis (WifiT et Eduroam), mais également dans de nombreuses infrastructures telles que les gares ou encore les restaurants qui proposent des services de connexion à Internet.

1. Cahier des charges : portail captif et page HTML

Un portail captif est un système installé sur un serveur qui fait le lien entre le poste de connexion de l'utilisateur, le réseau auquel on souhaite se connecter et un serveur RADIUS contenant les informations d'identifications des utilisateurs enregistrés.

Dans le cadre de notre projet informatique, cette fonction sera assurée par le logiciel CoovaChilli dans un environnement Linux. Lorsqu'un utilisateur tente de se connecter à un réseau (nous considérerons ici le cas du réseau internet), le logiciel bloque l'accès au réseau et redirige l'utilisateur vers une page d'accueil où il lui est demandé de s'identifier.

Si l'utilisateur est d'ores-et-déjà enregistré, il rentre ses identifiant et mot de passe. Le portail captif communique alors avec le serveur RADIUS afin de vérifier les informations entrées. Il peut alors accepter la requête ou la refuser. Si l'utilisateur est reconnu, CoovaChilli lui autorise l'accès au réseau. Sinon, ce dernier est bloqué par le portail captif. Si l'utilisateur n'est pas enregistré, le portail captif peut ou non lui proposer la possibilité de s'inscrire sur le serveur RADIUS.

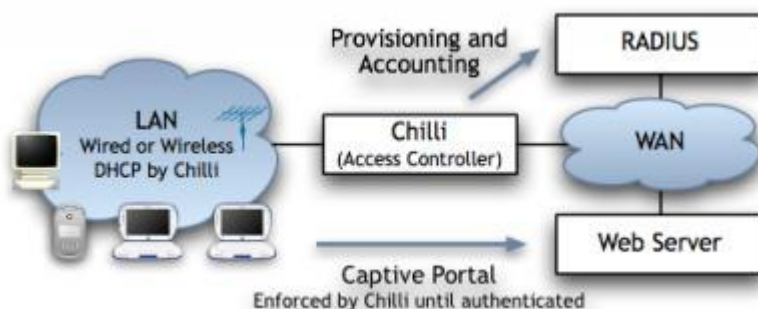


Schéma 2.1 : Interconnections et échanges entre les différents serveurs

Dans le projet Wonderville, il est nécessaire d'installer un portail captif sur la borne. Le prototype de borne réalisé dispose d'un système d'exploitation Linux modifié en raison des contraintes de dimensions de l'appareil. Ce système, relativement instable, est difficile à manipuler. C'est pourquoi notre principal objectif ne sera pas l'installation d'un portail captif sur la borne, mais sur un ordinateur personnel fonctionnant avec une distribution Linux Ubuntu standard.

Les objectifs de notre projet proposés par notre tuteur sont les suivants :

I - Installation d'un portail captif fonctionnel sur un ordinateur personnel

Installation du logiciel CoovaChilli sous une distribution Linux Ubuntu

II – Mettre en évidence son fonctionnement par l'intermédiaire d'une mise en pratique

Création d'un portail captif permettant ou non l'accès à Internet, communication avec un serveur radius, combinaisons des réseaux...

III - Réalisation d'une page HTML contenant diverses informations sur la borne :

Cette page devra contenir le nombre d'utilisateurs connectés au réseau via le portail captif. Il est demandé que ces informations soient protégées par un système d'identifiants et de mots de passe.

2. Déroulement du Projet

Le projet a beaucoup évolué dans le temps, notamment en raison d'un changement important de la direction donnée au projet. Suite à ce changement, l'idée de l'encadrant a été de nous laisser avancer étape par étape, sans considérer l'étape suivante. Nous avons donc chaque semaine un travail spécifique à réaliser. Une fois l'idée générale redéfinie au milieu du projet, nous avons pu avoir une vision plus globale du sujet, mais nous devions toujours attendre les explications de M. Afifi pour commencer à travailler sur l'étape suivante.

Nous avons suivi scrupuleusement le planning prévisionnel dans les toutes premières semaines, puis nous avons modifié notre organisation face aux difficultés que nous avons rencontrées au cours du projet.

6 février : Début projet, rencontre avec Afifi prise de conscience du sujet demandé

Deuxième semaine : Lecture des rapports des années passées pour la semaine suivante

Troisième semaine : - Explications de Mr Afifi quant au sujet à traiter

- Configuration du point d'accès, réinitialisation et désactivation des services

Quatrième semaine : Configuration de CoovaChilli pour créer le portail captif

Cinquième semaine : Premier problème, le makefile possède des erreurs. Impossible d'installer le logiciel

Sixième semaine : Deuxième problème, puisque l'archive ne marchait pas. Nous avons décidé de récupérer CoovaChilli sous forme de paquets. L'installation du paquet n'a pas marché et elle a endommagé le système de paquets du système Ubuntu installé sur l'ordinateur de Louis, il n'était plus possible d'installer aucun paquet sur cet ordinateur.

Ce projet, certes intéressant, n'était pas adapté à notre niveau. Les étapes d'installation du logiciel était trop complexe et nécessitait une connaissance approfondie du logiciel et du système Linux. Le makefile de l'archive téléchargé sur le site comportait des erreurs donc il était impossible d'installer CoovaChilli correctement et nous n'étions pas en mesure de corriger ces erreurs, n'ayant pas les connaissances et compétences requises. De plus l'architecture du logiciel CoovaChilli était beaucoup trop complexe. Cela nous entravait et nous empêchait d'avancer sur le projet. C'est pourquoi nous avons décidé de rencontrer Madame Taconet.

20 avril : Rencontre avec Mme Taconet qui a bien compris notre problème. Elle a décidé de s'entretenir avec M. Afifi pour trouver une solution adaptée à des élèves de 1^e année d'école d'ingénieur désireux de mettre en application leurs connaissances en C ce qui n'était pas le cas pour l'instant. (Pas de programme à écrire)

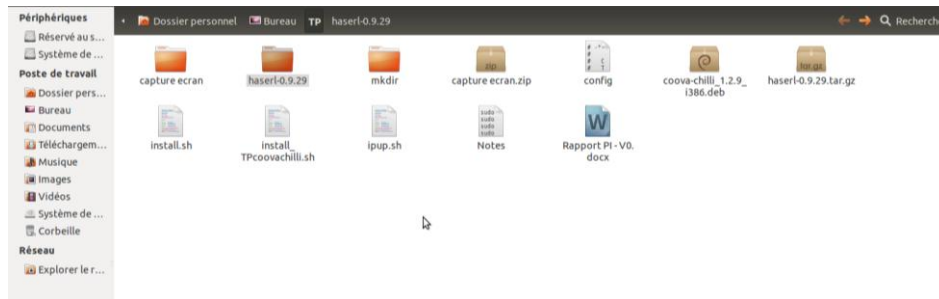
Vacances d'avril : Changement d'orientation. M. Afifi aidé de M. Aschiche Gupta a personnalisé une archive utilisable de CoovaChilli pour une installation sans erreur. Malheureusement, nous avons été confrontés à un nouvel obstacle. En effet, l'archive avait été conçue pour des ordinateurs 32bits mais l'ordinateur de Louis fonctionnait sur 64 bits. Nous avons passé toute une après-midi à chercher l'erreur et nous avons finalement installé une sorte de simulateur 32 bits sur l'ordinateur de Louis. Suite à cela CoovaChilli s'est installé sans problème. Nouvel objectif : Configuration d'une page HTML servant de page d'authentification administrateur en plus de celle du portail captif. Cette page pourrait servir à la personne authentifiée d'obtenir des informations sur le portail : nombre de visites, fréquence, mot de passe... Nécessité d'apprendre quelques bases des langages HTML, PHP, CSS et JavaScript pour la mise en page.

Rentrée vacances d'avril jusqu'au 7 mai : Réalisation de la page internet d'authentification en langage HTML et CSS

7 mai jusqu'au 14 mai : Codage d'un programme en C, et d'un compteur en PHP et CGI. Incrustation de ces scripts à la page HTML (nécessité d'hébergement ou d'un serveur virtuel)

2.1. Installation de CoovaChili

Tout d'abord on ouvre l'archive pour procéder à l'installation. Cette archive est différente de celle que l'on peut télécharger sur le site car elle a été personnalisée par nos tuteurs pour installer Coovachilli sans problème.



On procède à l'installation en lançant le fichier .sh. Ce fichier est un script shell qui exécute des commandes shell successivement.

```
louis@louis-Dell-System-XPS-L502X:~$ sudo ./install.sh
```

On copie le fichier de configuration personnalisé et le fichier IP dans le dossier /etc/chilli.

```
louis@louis-Dell-System-XPS-L502X:~$ cp config /etc/chilli
louis@louis-Dell-System-XPS-L502X:~$ cp ipup.sh /etc/chilli
```

On installe le logiciel Haserl qui permet de créer et de gérer les éléments web de la page d'authentification (de type cgi). On se place dans l'archive.

```
louis@louis-Dell-System-XPS-L502X:~$ cd haserl
```

On lance la configuration du logiciel, puis on peut commencer à installer.

```
louis@louis-Dell-System-XPS-L502X:~$ ./configure
```

```
louis@louis-Dell-System-XPS-L502X:~$ make
```

```
louis@louis-Dell-System-XPS-L502X:~$ make install
```

Si tout fonctionne, on peut lancer ou arrêter Coovachilli à l'aide des commandes suivantes :

```
louis@louis-Dell-System-XPS-L502X:~$ sudo /etc/init.d/chilli start
louis@louis-Dell-System-XPS-L502X:~$ sudo /etc/init.d/chilli stop
```

```

louis@louis-Dell-System-XPS-L502X: ~
louis@louis-Dell-System-XPS-L502X:~$ sudo /etc/init.d/chilli start
[sudo] password for louis:
Starting chilli: apres le modprobe

apres writeconf

apres le radius

apres ifconfig

eth0
chilli , chilli , /etc/chilli.conf, /usr/sbin/chilli
chilli.
louis@louis-Dell-System-XPS-L502X:~$ █

```

Schéma 3.1.1 : Lancement de Coovachili

On peut vérifier que la liaison est bien faite avec l'ordinateur à l'aide d'ipconfig et de ping. Ce qui est surligné représente le tunnel entre les 2 ordinateurs :

```

louis@louis-Dell-System-XPS-L502X: ~
chilli , chilli , /etc/chilli.conf, /usr/sbin/chilli
chilli.
louis@louis-Dell-System-XPS-L502X:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 14:fe:b5:c0:d2:e0
          UP BROADCAST PROMISC  MTU:1500  Metric:1
          Packets reçus:0 erreurs:0 :0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:0 (0.0 B) Octets transmis:0 (0.0 B)
          Interruption:47 Adresse de base:0xa000

lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          Packets reçus:22 erreurs:0 :0 overruns:0 frame:0
          TX packets:22 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          Octets reçus:2568 (2.5 KB) Octets transmis:2568 (2.5 KB)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet adr:192.168.1.1  P-t-P:192.168.1.1  Masque:255.255.255.0
          UP POINTOPOINT RUNNING  MTU:1500  Metric:1
          Packets reçus:0 erreurs:0 :0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:100
          Octets reçus:0 (0.0 B) Octets transmis:0 (0.0 B)

wlan0     Link encap:Ethernet  HWaddr ac:72:89:3c:f0:96
          inet adr:172.21.119.11  Bcast:172.21.255.255  Masque:255.255.0.0
          adr inet6: fe80::ae72:89ff:fe3c:f096/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:2834 erreurs:0 :0 overruns:0 frame:0
          TX packets:1726 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:2299659 (2.2 MB) Octets transmis:336316 (336.3 KB)

louis@louis-Dell-System-XPS-L502X:~$ █

```

Schéma 3.1.2 : Vérification du bon établissement de la communication

```

louis@louis-Dell-System-XPS-L502X: ~
louis@louis-Dell-System-XPS-L502X:~$ ping
Usage: ping [-LRUbdnqrVvAD] [-c count] [-i interval] [-w deadline]
          [-p pattern] [-s packetsize] [-t ttl] [-I interface]
          [-M pmtudisc-hint] [-m mark] [-S sndbuf]
          [-T tstamp-options] [-Q tos] [hop1 ...] destination
louis@louis-Dell-System-XPS-L502X:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_req=1 ttl=64 time=0.105 ms
64 bytes from 192.168.1.1: icmp_req=2 ttl=64 time=0.072 ms
64 bytes from 192.168.1.1: icmp_req=3 ttl=64 time=0.072 ms
64 bytes from 192.168.1.1: icmp_req=4 ttl=64 time=0.073 ms
64 bytes from 192.168.1.1: icmp_req=5 ttl=64 time=0.072 ms
64 bytes from 192.168.1.1: icmp_req=6 ttl=64 time=0.071 ms
64 bytes from 192.168.1.1: icmp_req=7 ttl=64 time=0.074 ms
64 bytes from 192.168.1.1: icmp_req=8 ttl=64 time=0.071 ms
64 bytes from 192.168.1.1: icmp_req=9 ttl=64 time=0.073 ms
64 bytes from 192.168.1.1: icmp_req=10 ttl=64 time=0.073 ms
64 bytes from 192.168.1.1: icmp_req=11 ttl=64 time=0.074 ms
64 bytes from 192.168.1.1: icmp_req=12 ttl=64 time=0.066 ms
64 bytes from 192.168.1.1: icmp_req=13 ttl=64 time=0.070 ms
64 bytes from 192.168.1.1: icmp_req=14 ttl=64 time=0.072 ms
64 bytes from 192.168.1.1: icmp_req=15 ttl=64 time=0.065 ms
64 bytes from 192.168.1.1: icmp_req=16 ttl=64 time=0.064 ms
64 bytes from 192.168.1.1: icmp_req=17 ttl=64 time=0.074 ms
64 bytes from 192.168.1.1: icmp_req=18 ttl=64 time=0.064 ms
64 bytes from 192.168.1.1: icmp_req=19 ttl=64 time=0.073 ms
64 bytes from 192.168.1.1: icmp_req=20 ttl=64 time=0.072 ms
64 bytes from 192.168.1.1: icmp_req=21 ttl=64 time=0.072 ms
64 bytes from 192.168.1.1: icmp_req=22 ttl=64 time=0.073 ms
64 bytes from 192.168.1.1: icmp_req=23 ttl=64 time=0.074 ms
64 bytes from 192.168.1.1: icmp_req=24 ttl=64 time=0.073 ms
64 bytes from 192.168.1.1: icmp_req=25 ttl=64 time=0.075 ms
64 bytes from 192.168.1.1: icmp_req=26 ttl=64 time=0.090 ms
64 bytes from 192.168.1.1: icmp_req=27 ttl=64 time=0.075 ms
64 bytes from 192.168.1.1: icmp_req=28 ttl=64 time=0.076 ms
^C
--- 192.168.1.1 ping statistics ---
28 packets transmitted, 28 received, 0% packet loss, time 26998ms
rtt min/avg/max/mdev = 0.064/0.073/0.105/0.011 ms
louis@louis-Dell-System-XPS-L502X:~$

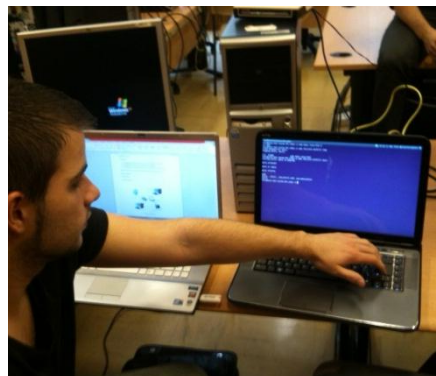
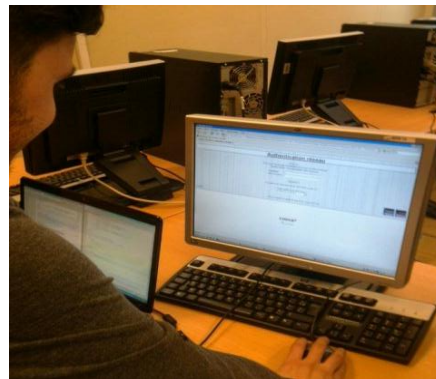
```

Schéma 3.1.3 : la fonction ping permet de vérifier la bonne réception des informations envoyées au serveur distant

2.2. TP : Création et utilisation d'un portail captif

Matériel pour le TP :

- 3 ordinateurs :
 - 1 ordinateur portable sous Ubuntu sur lequel est installé CoovaChilli
 - 1 ordinateur du campus qui fera office de serveur Radius
 - l'ordinateur de l'utilisateur (sous n'importe quel système d'exploitation)
- 1 point d'accès D-Link
- 4 câbles Ethernet
- 1 hub Ethernet
- Le réseau internet de l'école



2.2.1. Etape n°1 : Désactivation d'options du point d'accès

La première étape de ce TP nécessite au départ un ordinateur quelconque (de préférence celui sur lequel est installé le serveur CoovaChilli), ainsi que le point d'accès. Les deux appareils sont reliés en filaire.



Schéma 3.2.1.1 : Dispositif de connexion

La première difficulté que nous avons rencontrée fut d'établir la connexion effective entre les deux appareils. En effet, après avoir rentré dans le navigateur web de l'ordinateur l'adresse IP « 192.168.1.1 » du point d'accès, la connexion ne s'établissait pas. Après de nombreux essais, nous nous sommes aperçus qu'il fallait désactiver l'identification IEEE 802.1X dans les paramètres réseaux de l'ordinateur. Cette option empêchait la connexion au boîtier.

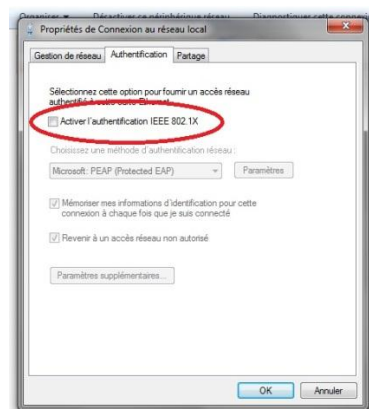


Schéma 3.2.1.2 : Désactivation de IEEE 802.1X

Une fois la connexion établie après avoir tapé l'adresse IP et les bons mots de passe, ici « admin », on peut agir sur le point d'accès au travers du navigateur web.

Les diverses informations de configuration du point d'accès s'affichent. Il s'agit ici surtout de désactiver le protocole DHCP ainsi que la sécurité Internet.

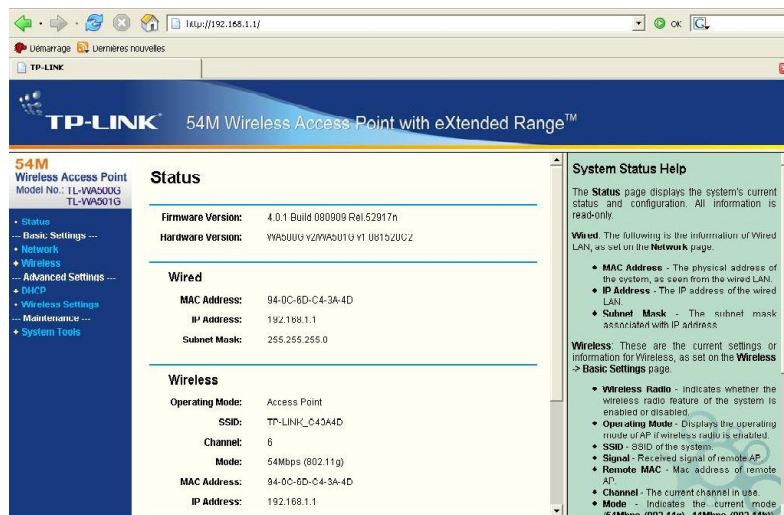


Schéma 3.2.1.3 : Configuration du point d'accès

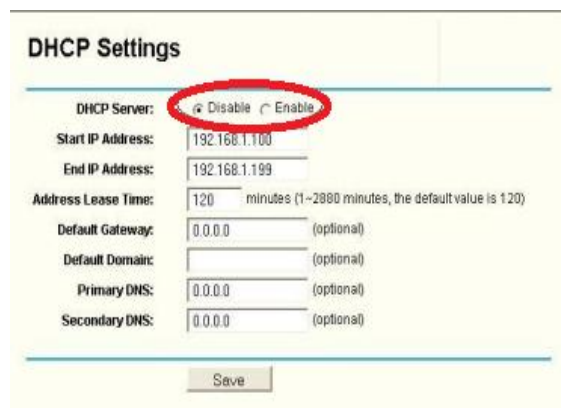


Schéma 3.2.1.4 : Désactivation du protocole DHCP



Schéma 3.2.1.4 : Désactivation de la sécurité internet

DHCP (Dynamic Host Configuration Protocol) est un protocole qui permet à un ordinateur qui se connecte sur un réseau d'obtenir dynamiquement (c'est-à-dire sans intervention particulière) sa configuration réseau.

Le protocole DHCP sert principalement à distribuer des adresses IP sur un réseau, mais il a été conçu au départ comme complément au protocole BOOTP (Bootstrap Protocol) qui est utilisé par exemple lorsque l'on installe une machine à travers un réseau (BOOTP est utilisé en étroite collaboration avec un serveur TFTP sur lequel le client va trouver les fichiers à charger et à copier sur le disque dur). Un serveur DHCP peut renvoyer des paramètres BOOTP ou de configuration propres à un hôte donné.

2.2.2. Etape n°2 : Mise en place du portail captif



Schéma 3.2.2.1 : Dispositif du TP souhaité

Afin de simuler au mieux le fonctionnement de la borne, nous avons voulu remplacer celle-ci par un ordinateur personnel, relié au réseau internet et au serveur radius, et permettant à un utilisateur tierce de se connecter. Pour des raisons de praticités, nous avons dû modifier les dispositifs de connexion entre les différents éléments pour s'adapter au matériel disponible et aux possibilités qu'il offrait. Nous avons donc créé un dispositif alternatif mais dont la simple différence réside dans les outils de connexion. Les résultats obtenus seront donc similaires et parfaitement applicable au dispositif le plus proche du dispositif réel.

Nous avons donc créé le dispositif suivant :



Schéma 3.2.2.2 : Dispositif du TP

La connexion entre l'utilisateur et le portail captif se fait par l'intermédiaire d'un câble Ethernet (utilisation d'un ordinateur fixe de salle de TP dépourvu de Wi-Fi). L'ordinateur personnel sur lequel est installé Coovachilli ne disposant que d'un unique port Ethernet, celui-ci sera relié au serveur radius et à internet au moyen d'un point d'accès Wi-Fi (passif) et d'un Hub qui établit une connexion Ethernet entre les 4 entités.

On désactive toute connexion de l'utilisateur et on tente de se connecter à Internet en ouvrant un quelconque navigateur internet.

- Si l'opération de lancement de Coovachilli n'a pas fonctionné, l'utilisateur échoue alors sur la page suivante :

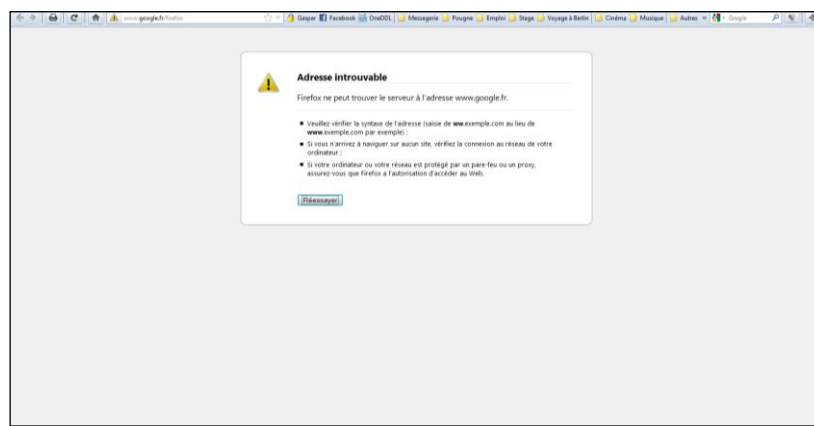


Schéma 3.2.2.3 : Problème de connexion

Cette erreur survient si l'ordinateur sous CoovaChilli n'est pas prêt, ou s'il est mal configuré, c'est-à-dire si la première étape a mal été effectuée.

- Si l'opération est un succès, l'utilisateur voit alors s'afficher cette page :

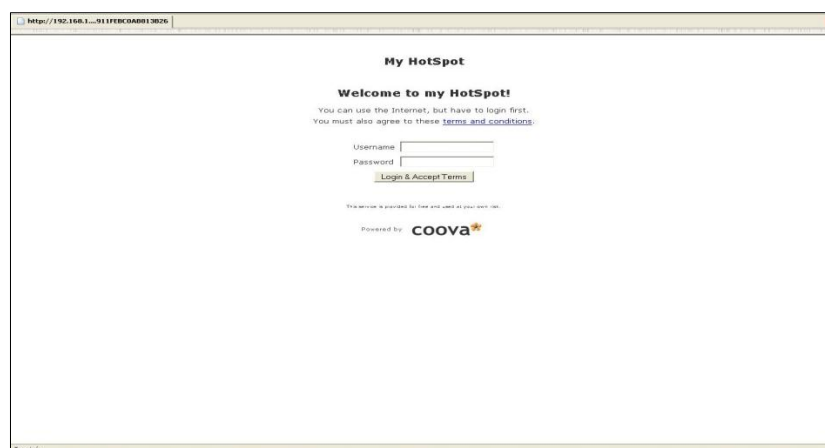


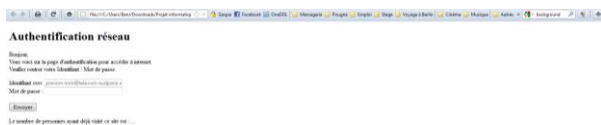
Schéma 3.2.2.4 : Page de CoovaChilli

Il se retrouve alors sur la page d'authentification mise en place par le logiciel CoovaChilli. S'il veut effectivement pouvoir surfer sur le web, l'utilisateur est amené à s'identifier en rentrant son nom d'utilisateur (Username), ainsi que son mot de passe (Password). Les informations rentrées dans ces formulaires sont alors envoyées au serveur Radius qui vérifie si l'utilisateur est enregistré dans la base données des utilisateurs autorisées. Si tel est le cas, le serveur radius envoie une réponse positive au portail captif qui « libère » la connexion internet à l'individu connecté. Le portail captif constitue donc une sécurité vis-à-vis du réseau pour empêcher tout individu inconnu de se connecter sur un serveur.

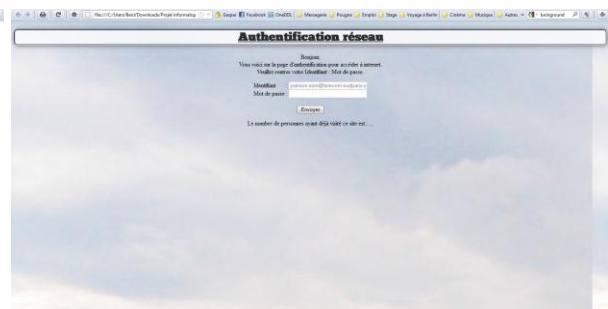
2.3. Création d'une page HTML d'authentification

Par la suite, notre nouvel objectif a été de créer une page HTML complémentaire à celle qui s'affiche à l'ouverture du portail captif. Sur cette page, il est souhaité qu'il puisse y figurer divers éléments tels que les logos de l'école, un système de contact, des questionnaires, un système pour afficher l'heure...

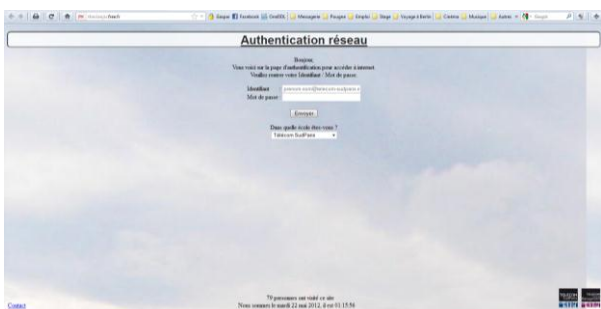
Cette page doit également contenir un compteur du nombre d'utilisateur connectés via le portail captif. Ces informations devront être protégées par un identifiant et un mot de passe d'administrateur. Nous y reviendrons par la suite. Voici différentes versions du site à travers lesquelles nous voyons son évolution.



Version 1 : Brute sans CSS



Version 2 : Avec CSS



Version 3 : Ajout de date, questionnaire, logos et contact



Version 4 : Ajout du compteur et des infos

La construction de ce site web a nécessité l'apprentissage de plusieurs langages et fonctionnalités :

- Le langage HTML pour la structure globale de la page
- Le CSS pour la mise en forme
- Le PHP pour exécuter quelques scripts tels que le compteur de visite ou encore le compteur d'utilisateurs connectés au portail (principale problématique)
- L'utilisation de fichiers .htaccess et .htpasswd pour protéger la page des informations
- La compréhension du JavaScript pour un script affichant la date

Dans les parties qui suivront, nous ne traiterons pas le détail du codage HTML et CSS. Ses codes sont néanmoins fournis en annexe.

1. L’affichage de l’heure

Il s’agit simplement ici de l’utilisation d’un script JavaScript existant appelé par les lignes suivantes :

```
<script type="text/javascript" src="date_heure.js"></script>
```

(contenu dans l’entête du fichier)

```
<span id="date_heure"></span>
```

```
<script type="text/javascript">window.onload = date_heure('date_heure');</script></article>
```

(pour exécuter le script)

2. Le compteur de visite

Le compteur de visite est effectué par l’intermédiaire d’un script PHP dont voici le code :

```
13
14 <p><?php
15 $fp = fopen("../compteur.txt", "r+");
16 $nbvisites = fgets($fp, 10);
17 if ($nbvisites=="") $nbvisites = 0;
18 $nbvisites++;
19 fseek($fp, 0);
20 fputs($fp, $nbvisites);
21 fclose($fp);
22 echo "$nbvisites personnes ont visité ce site";
23 ?></p>
```

Schéma 3.3.1 : script PHP du compteur de visites

Le principe de ce compteur est simple : il ouvre le fichier « compteur.txt » où est inscrit le nombre de personnes qui se sont connectées (fopen), récupère ce nombre de visites (fgets), incrémente ce nombre, se replace au début du fichier (fseek) pour inscrire le nouveau nombre (fputs) et l’afficher (echo).

3. Le compteur d'utilisateurs du portail captif

Sur l'ordinateur personnel sur lequel nous avons installé Coovachilli, chaque connexion d'un utilisateur se traduit par une inscription de son adresse MAC dans un fichier « syslog » situé dans le dossier /var/log de l'ordinateur. Ainsi, le nombre d'utilisateurs connectés correspond au nombre d'occurrences du mot « MAC » dans le fichier syslog.

L'installation de cette page n'ayant pas été faite sur l'ordinateur personnel, le script interprète le nombre d'itération du mot MAC dans un fichier syslog de test présent dans le répertoire www du site web. Le contenu de ce fichier test peut-être modifié pour tester le fonctionnement du script. Pour adapter ce script à l'utilisation faite avec Coovachilli, il suffit simplement de remplacement l'emplacement du syslog afin de prendre en considération le fichier syslog de l'ordinateur (en vérifiant que celui-ci possède les droits d'exécution nécessaires).

Le code du script est le suivant :

```
<p><?php
$fichier = fopen("syslog", "r+");
$compteur= 0;
while (!feof($fichier) ) {
    $s= fgets($fichier, 80);
    $compteur = $compteur + substr_count($s, 'MAC');
}
fclose($fichier);
echo "$compteur utilisateur(s) sont actuellement connecté(s)";
?></p>
```

Schéma 3.3.2 : script PHP du compteur de connectés

Celui-ci ouvre le fichier syslog. Il initialise un compteur d'occurrences de « MAC » à 0. Tant qu'on est pas arrivé à la fin du programme, on lit le fichier ligne par ligne. On teste la présence de MAC dans la ligne lue avec la fonction « substr_count ». Si celui-ci est présent, substr_count renvoie 1 (et incrémente le compteur), sinon 0 (elle laisse le compteur inchangée). On ferme ensuite le fichier syslog et on affiche le nombre d'utilisateurs connectés.

Le comptage de l'occurrence de « MAC » dans le fichier syslog peut se faire très rapidement avec la commande shell « grep MAC ./syslog | wc -l ». Néanmoins, nous nous sommes aperçu des difficultés que représentait l'exécution d'une telle commande au sein d'une page web. Certaines fonctions PHP permettent l'utilisation de commande SHELL (exec, shell_exec...) mais ne fonctionnait pas lors de nos tests.

Nous avons un temps pensé à exécuter cette fonction au travers d'une application cgi : ceci est cohérent avec le principe du cgi qui consiste à exécuter sur le serveur la fonction demandée (en l'occurrence, le serveur ou tourne coovachilli et qui possède le fichier syslog).

Pour cela, nous avons réalisé le programme C suivant :

```
1
2  #include <stdlib.h>
3  #include <stdio.h>
4  #define TAILLE_TAMPON 81
5  #include "cgl.h"
6
7
8  main() {
9
10     FILE *nbco;
11     char tampon[TAILLE_TAMPON];
12     int nb;
13     if((nbco = popen("grep MAC ./syslog | wc -l", "r")) == NULL)
14     {
15         perror("popen");
16         exit(0);
17     }
18     while((fgets(tampon, TAILLE_TAMPON, nbco)) != NULL) {
19         nb = strtol(tampon, NULL, 10);
20     }
21     pclose(nbco);
22     cgl_html_header();
23     cgl_html_begin("Nombre d'utilisateurs connectés");
24     printf("<pre>\n");
25     printf("Le nombre d'utilisateurs connectés est %d \n", nb);
26     printf("</pre>\n");
27     cgl_html_end();
28 }
29
30
```

Schéma 3.3.3 : fonction C réalisant le comptage du nombre de connectés

La compilation de ce programme, donnée par la ligne de code :

```
gcc -o programme.cgi programme.c -L. -lcgl
```

a permis la création d'un fichier programme.cgi fonctionnel. Néanmoins, nous n'avons pas réussi à exécuter ce programme sur notre page web qui téléchargeait le script au lieu de l'exécuter. D'où le report vers une solution PHP.

Schéma 3.3.4 : Visuel du site web réalisé

4. Protection de la page infos.php

Les informations données par les scripts sont présentes sur une page infos.php dont nous voulons réduire l'accès. Pour cela, nous avons créé les fichiers .htaccess et .htpasswd, présents dans le répertoire de la page infos.php, qui limite l'accès de celle-ci aux utilisateurs enregistrés.

Le fichier .htaccess se présente de la manière suivante :

```
AuthName "Veuillez saisir vos identifiants"

AuthType Basic

AuthUserFile "C:\wamp\www\Authentification\infos\.htpasswd"

Require valid-user
```

Le fichier .htpasswd appelé, qui contient les mots de passe, se présente de la sorte :

```
admin:admin

maxime:intervenir
```

On peut noter ici que les mots de passe ne sont pas cryptés. Ceci est dû au fait que nous avons utilisé un serveur free pour faire fonctionner nos programmes PHP, puis un serveur Apache pour les faire fonctionner en local. Tous deux requièrent des mots de passe non cryptés pour fonctionner. Le serveur free nécessite également un autre fichier .htaccess (et non .htaccess) qui contient la simple ligne « deny from all » et protège ainsi le fichier .htpasswd contenant les mots de passe.

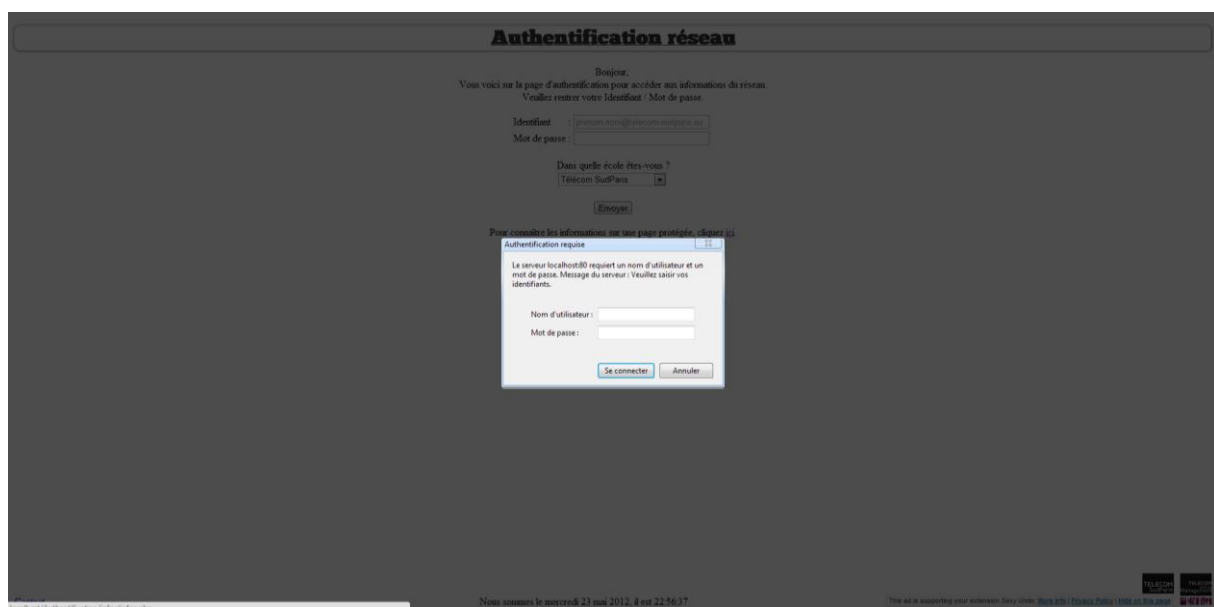


Schéma 3.3.5 : Demande d'identification

5. Champs de saisie

Nous avons également tenu à ajouter des champs de saisie qui permettront à un client potentiel de paramétrer par la suite un service d'identification reposant sur une base de données SQL. Les champs créés n'ont pour l'instant aucun effet sur la page et sur l'identification, mais on peut imaginer à l'avenir une utilisation de ses champs pour transmettre des informations au serveur ou simplement s'identifier.

6. Incorporation de la page à Coovachilli

Dans un premier temps, nous souhaitions modifier la page de redirection de Coovachilli pour afficher les informations demandées. Pour des raisons inconnues, cette page est restaurée après toute modification, probablement lors de chaque exécution de Coovachilli : nous avons donc dû créer une page indépendante. Afin de disposer des deux pages (Coovachilli et la page d'informations) nous avons redirigé les connexions des utilisateurs sur une page divisée en deux parties :

The screenshot shows a Mozilla Firefox browser window. The top part of the window displays the 'Authentication réseau' page. It has a title bar with 'Mozilla Firefox' and a menu bar with 'Fichier', 'Edition', 'Affichage', 'Aller à', 'Marque-pages', 'Outils', and '?'. The address bar shows a URL: 'http://192.168.1.1:3990/www/coova.html?loginurl=http%3a%2f%2f192.168.1.1%3a4990%2fwww%2flogin.ch%3fres%3dnotyet%26uamp%3d192.168.1.1%26uampor'. The page content includes a greeting 'Bonjour,' followed by instructions: 'Vous voici sur la page d'authentification pour accéder à internet. Veuillez rentrer votre Identifiant / Mot de passe.' There are input fields for 'Identifiant' and 'Mot de passe', and an 'Envoyer' button. Below this, it says 'Le nombre de personnes ayant déjà visité ce site est : ...' and a dropdown menu for 'Dans quelle école êtes-vous ?' with 'Télécom SudParis' selected. At the bottom, it says 'Nous sommes le mardi 22 mai 2012, il est 16:23:33' and has a 'Contact' link. The bottom part of the browser window shows the 'My HotSpot' page. It has a title 'My HotSpot' and a subtitle 'Welcome to my HotSpot!'. The text says 'You can use the Internet, but have to login first. You must also agree to these [terms and conditions](#).' There are input fields for 'Username' and 'Password', and a 'Login & Accept Terms' button. At the bottom, it says 'This service is provided for free and used at your own risk.' and 'Powered by coova'. The browser's status bar at the very bottom shows 'Terminé', 'Démarrer', 'Mozilla Firefox', 'Connexions réseau', and the time '16:23'.

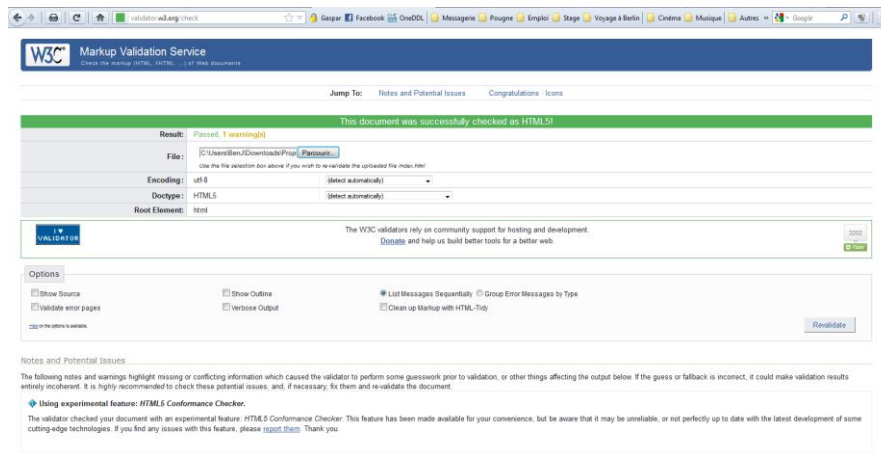
Schéma 3.3.6 : redirection de Coovachilli : informations et identification

(la version de la page HTML est ici une version non définitive)

2.4. Vérifications des codes et tests

Test de notre page HTML : Site <http://validator.w3.org/>

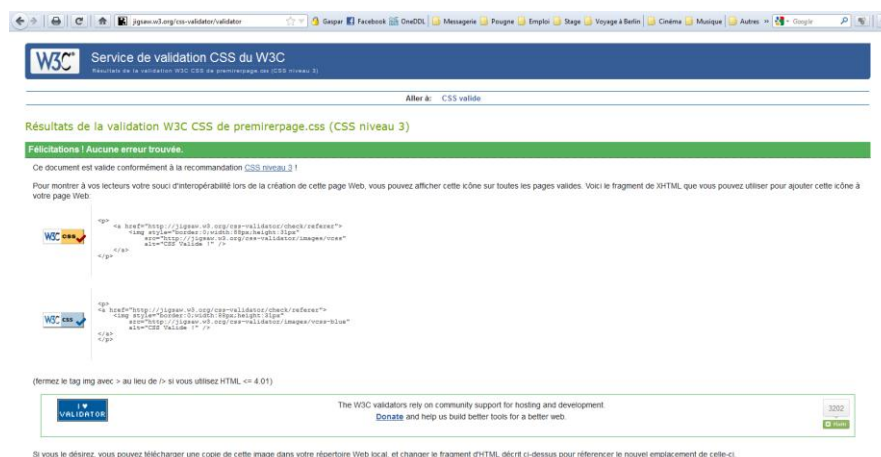
Lors du premier test HTML, nous avons 8 erreurs, principalement dues à des problèmes de syntaxes ou d'ordres de fermetures de balise, qui n'empêchaient néanmoins pas un bon affichage par le navigateur. Après multiples corrections, la page était correctement codée.



Vérification HTML

Test de notre fichier CSS : Site <http://jigsaw.w3.org/css-validator/>

Lors du premier test CSS, nous avons 6 erreurs, qui étaient quant à elles dues principalement à des fautes de frappes et à des erreurs de types.



Vérification CSS

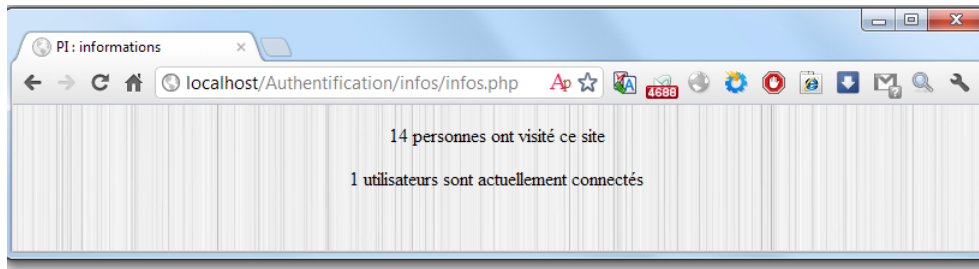
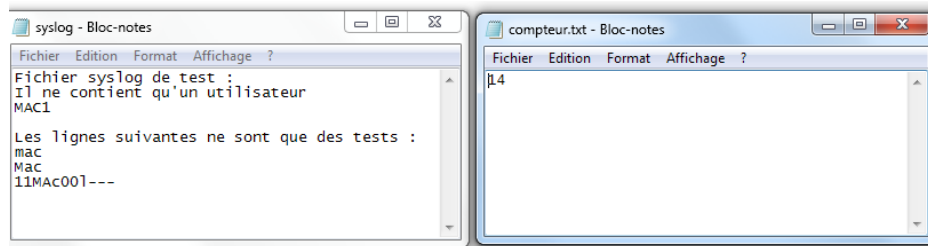


Schéma 3.5.1 : affichage de la page infos.php après authentification

Les compteurs renvoient 14 et 1 correspondant au fichier syslog (test) et compteur.txt suivant :



Remarque : le fichier compteur.txt correspond à l'ensemble du site et est donc unique, hébergé dans le dossier source www. Chaque page possède son propre fichier syslog, pour les raisons évoquées ci-dessus concernant les changements de chemin vers un fichier syslog unique intégré aux répertoires système du serveur Coovachilli.

2.5. Hébergement sur le web et en local

L'interprétation des scripts php nécessite l'utilisation d'un serveur web. Pour plus de flexibilité, nous avons utilisé deux serveurs web :

- Un serveur Free pour mettre le site en ligne (dont quelques spécificités sur les mots de passe ont été vues précédemment)
 - Un serveur local Apache pour exécuter le site sur les machines sans connexion
- C'est par ailleurs, la solution répondant à notre problématique.

L'hébergement du site se fait par l'utilisation du logiciel Filezilla pour transférer les fichiers en FTP sur l'hôte ftpperso.free.fr. Le site web est donc présent à l'adresse <http://theclanjsr.free.fr> (adresse personnelle d'un des membres de l'équipe).

free



Afin d'utiliser la page HTML en local, il est nécessaire d'installer un serveur local. Nous avons opté pour le logiciel WAMP sous windows qui permet l'utilisation d'un serveur Apache. Après quelques difficultés (notamment résoudre le problème de l'utilisation du port 80 utilisé par à la fois le logiciel Skype, un serveur web Microsoft IIS et le serveur web Apache concerné), nous avons réussi à exécuter la page localement sur un de nos ordinateurs personnels. Celle-ci est entièrement fonctionnelle.



3. Conclusion

Le concept de portail captif est fondamental dans le cadre de notre projet. Nous avons donc dû assimiler, au travers des explications de notre tuteur et à la lecture des divers documents réalisés dans le cadre du Projet Wonderville, le fonctionnement et les spécificités d'un tel système.

Nous avons ainsi été menés à mettre en place un portail captif au moyen de l'outil informatique CoovaChilli. Notre travail a nécessité l'utilisation de plusieurs types d'équipements : un point d'accès WiFi, un ordinateur configuré en serveur Radius, un autre ordinateur qui fait fonctionner le portail captif par l'intermédiaire du logiciel CoovaChilli, ainsi que l'appareil de l'utilisateur.

La réalisation de la page HTML nous a permis à tous d'apprendre les bases de nombreux langages de programmation de site WEB tels que les langages HTML, CSS et PHP. Elle nous a également permis de savoir comment protéger une page web via les fichiers .htaccess et .htpasswd. La réalisation de l'application CGI nous a également permis de mieux appréhender ces outils sans toutefois réussir à finaliser leur utilisation.

Ce projet nous a permis de gérer plusieurs langages et différentes technologies. Nous nous avons donc consultés de nombreux tutoriels et documentations pour se sensibiliser à ces outils et cela fut très formateur, malgré le temps perdu à se documenter.

Ce qui nous a peut-être manqué tout au long de ce projet, c'est le manque de codage personnel à réaliser. Faute d'un objectif final clair et prédéfini, il nous a été impossible de mener une démarche descendante pour ce projet et de diviser le travail à effectuer. Si cette division s'est faite plus naturellement après la réorientation, le manque de travail que chacun pouvait fournir seul a fortement ralenti la progression de ce projet. Le travail de codage est arrivé tardivement dans l'évolution du projet, qui a d'ailleurs subi un changement significatif en cours de route.

Pour aller, un peu plus loin, nous aurions aimé pouvoir mettre en place une base de données SQL plus convenante à la page d'identification. L'intégration de cette page au sein de Coovachilli aurait également été souhaitable, malgré les soucis de restauration de page. Avec plus de temps et une réorientation plus anticipée, nous aurions peut-être eu la possibilité de tester l'application cgi créée. En outre, nous avons étudié ici les seules connexions de type filaire ou Wifi, mais il aurait été intéressant de porter de l'attention à la connexion Bluetooth. Enfin, malgré les contraintes physiques de la borne, il aurait été intéressant d'installer un portail captif sur la borne afin de vérifier le fonctionnement d'un tel système.

4. Bibliographie

Documents :

- Rapport de M. Aschiche
- Manuels sur Ubuntu

Internet :

- Le Site du zéro : <http://www.siteduzero.com/>
- CoovaChilli France : <http://doc.ubuntu-fr.org/coovachilli>
- Portail captif : http://doc.ubuntu-fr.org/portail_captif
- Documentation PHP : <http://php.net>
- Ressources Moodle (notamment le module Technologie web)

5. Annexes

5.1. Annexe 1 : Gestion de projet

5.1.1. Plan de charge

PLAN DE CHARGES PRÉVISIONNEL							SUIVI D'ACTIVITÉS (Charge Consommée)						
Description de l'activité	Charge en %	Charge en H	Charge en H / Participant				Charge en %	Charge en H	Charge en H / Participant				
			M	F	B	L			M	F	B	L	
Total	100%	205	51	52	51	51	110%	225	57	55	56	57	
Gestion de projets	28%	58	15	15	14	14	32%	73	16	19	19	19	
Réunion de lancement	2%	4	1	1	1	1	2%	4	1	1	1	1	
Planning prévisionnel et Suivi d'activités	2%	4	1	1	1	1	4%	8	2	2	2	2	
Réunions de suivi	20%	40	10	10	10	10	21%	48	12	12	12	12	
Rédaction	5%	10	3	3	2	2	6%	13	1	4	4	4	
Spécification	4%	8	2	2	2	2	4%	10	3	3	2	2	
Définition des fonctionnalités	4%	8	2	2	2	2	4%	10	3	3	2	2	
Conception préliminaire	14%	29	6	7	7	9	19%	42	12	7	9	14	
Installation et réglages coovachilli	1%	2				2	4%	9			2	7	
Apprentissage des bases des langages php et html	6%	12	2		5	5	6%	13	5		4	4	
Apprentissage des bases des langages CGI et CSS	3%	7	2	5			4%	8	4	4			
cours sur portail captif	4%	8	2	2	2	2	5%	12	3	3	3	3	
Conception détaillée	0%	0	0	0	0	0	0%	0	0	0	0	0	
Définition des structures de données	0%	0					0%	0					
Définition des fonctions	0%	0					0%	0					
Définition des tests unitaires	0%	0					0%	0					
Auto-formation Gtk et Glade	0%	0					0%	0					
Maquettage des IHM (ascii, web...)	0%	0					0%	0					
Codage	39%	80	20	20	20	20	28%	64	16	16	18	14	
Ecriture du code html des page d'accueil	1%	20	5	5	5	5	8%	17	4	3	5	5	
php	10%	20	5	5	5	5	7%	15	4	3	5	3	
CSS	10%	20	5	5	5	5	7%	16	4	5	4	3	
CGI	10%	20	5	5	5	5	7%	16	4	5	4	3	
Intégration	7%	14	4	4	4	2	7%	16	5	5	3	3	
Intégration des codes php,CSS et CGI au code HTML	4%	8	3	3	1	1	4%	10	3	3	2	2	
Tests d'intégration	3%	6	1	1	3	1	3%	6	2	2	1	1	
Soutenance	8%	16	4	4	4	4	9%	20	5	5	5	5	
Préparation de la soutenance	6%	12	3	3	3	3	7%	16	4	4	4	4	
Soutenance	2%	4	1	1	1	1	2%	4	1	1	1	1	

Document également disponible au format excel.

5.1.2. Planning Prévisionnel

6 février : Début projet, rencontre avec Afifi prise de conscience du sujet demandé

Deuxième semaine : Lecture des rapports des années passées pour la semaine suivante

Troisième semaine : - Explications de Mr Afifi quant au sujet à traiter

- Configuration du point d'accès, réinitialisation et désactivation des services

Quatrième semaine : Configuration de CoovaChilli pour créer le portail captif

20 avril : Réorientation Projet par Mme Taconet. Nouveau Planning prévisionnel.

Vacances d'avril : Nouvel objectif : Configuration d'une page HTML servant de page d'authentification administrateur en plus de celle du portail captif. Cette page pourrait servir à la personne authentifiée d'obtenir des informations sur le portail : nombre de visites, fréquence, mot de passe... Nécessité d'apprendre quelques bases des langages HTML, PHP, CSS et JavaScript pour la mise en page.

Rentrée vacances d'avril jusqu'au 7 mai : Réalisation de la page internet d'authentification en langage HTML et CSS

7 mai jusqu'au 14 mai : Codage d'un programme en C, et d'un compteur en PHP et CGI.

5.2. Annexe 2 : Code source

Codage HTML

```
Fichier  Edition  Recherche  Affichage  Encodage  Langage  Paramétrage  Macro  Exécution  Documents  ?
[Icons]
premierpage.css  index.html

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8" />
5      <link rel="stylesheet" type="text/css" href="premierpage.css" />
6      <script type="text/javascript" src="date_heure.js"></script>
7      <title>PI : authentification</title>
8  </head>
9
10 <body>
11 <h1> Authentication réseau </h1>
12 <p>Bonjour, <br/>Vous voici sur la page d'authentification pour accéder à internet.<br/>Veuillez rentrer votre Identifiant / Mot de passe. </p>
13 <form method="post" action="traitement.php">
14 <p>
15     <label for="pseudo">Identifiant <em>nnn</em>: </label>
16     <input type="text" name="pseudo" id="pseudo" placeholder="prenom.nom@telecom-sudparis.eu" size="30" maxlength="10" />
17
18     <br />
19     <label for="pass">Mot de passe :</label>
20     <input type="password" name="pass" id="pass" size="30" />
21     <br /> <br />
22     <input type="submit" value="Envoyer" />
23 </p>
24 </form>
25 <form method="post" action="traitement.php">
26 <p>
27     <label for="ecole">Dans quelle école êtes-vous ?</label><br />
28     <select name="ecole" id="ecole">
29         <optgroup label="Professeur">
30             <option value="sudparis">Télécom SudParis</option>
31             <option value="management">Télécom Management</option>
32             <option value="chercheur">Chercheur</option>
33         </optgroup>
34         <optgroup label="Etudiant">
35             <option value="sudparis">Télécom SudParis</option>
36             <option value="management">Télécom Management</option>
37         </optgroup>
38     </select>
39 </p>
40
41 </form>
42 <aside>  </aside>
43 <div class="date">
44     <span id="date_heure"></span>
45     <script type="text/javascript">window.onload = date_heure('date_heure');</script>
46 </div>
47 <footer>
48 <a href="mailto:bw.mauries@orange.fr">Contact</a>
49 </footer>
50 <header><a href="http://www.telecom-em.eu"> </a></header>
51 <div><a href="http://www.telecom-sudparis.eu"></a>
52 </div>
53 </body>
54 </html>
```


Codage CSS

```
Fichier  Edition  Recherche  Affichage  Encodage  Langage  Paramétrage  Macro  Exécution  Documents  ?

premierpage.css  index.html

1  @font-face
2  {
3      font-family: 'Chunkfive';
4      src: url('Chunkfive.otf') format('opentype');
5  }
6
7  h1
8  {
9      color: black;
10     background-color: white;
11     text-decoration: underline;
12     text-align: center;
13     border: groove;
14     box-shadow: 1px 1px 12px #555;
15     opacity: 0.2;
16     font-family: 'Chunkfive', Arial, serif;
17     border-radius: 10px;
18     opacity: 0.8;
19 }
20
21 p
22 {
23     color: black;
24     text-align: center;
25 }
26
27 body
28 {
29     background-image: url("background.jpg")
30 }
31
32 header
33 {
34     position: fixed;
35     right: 10px;
36     bottom: 10px;
37 }
38
39 div
40 {
41     position: fixed;
42     right: 70px;
43     bottom: 10px;
44 }
45
46 footer
47 {
48     position: fixed;
49     left: 10px;
50     bottom: 10px;
51 }
52
53
54 em
55 {
56     opacity: 0
57 }
58
59 .date
60 {
61     position: fixed;
62     bottom : 10px;
63     right: 35%;
64     left: 39%;
65 }
66
67 section
68 {
69     position: fixed;
70     bottom : 70px;
71     right: 45%;
72     left: 29%;
73 }
74
75 aside
76 {
77     position: fixed;
78     right: 36%;
79     left: 29%;
80 }
81
```

Codage HTML et PHP (infos.php)

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="utf-8" />
5     <link rel="stylesheet" href="premierpage.css" />
6     <script type="text/javascript" src="date_heure.js"></script>
7     <title>PI : informations</title>
8 </head>
9
10 <body>
11
12
13
14 <p><?php
15 $fp = fopen("../compteur.txt", "r+");
16 $nbvisites = fgets($fp, 10);
17 if ($nbvisites=="") $nbvisites = 0;
18 $nbvisites++;
19 fseek($fp, 0);
20 fputs($fp, $nbvisites);
21 fclose($fp);
22 echo "$nbvisites personnes ont visité ce site";
23 ?></p>
24 <p><?php
25 $fichier = fopen("syslog", "r+");
26 $compteur = 0;
27 while (!feof($fichier) ) {
28     $s = fgets($fichier, 80);
29     $compteur = $compteur + substr_count($s, 'MAC');
30 }
31 fclose($fichier);
32 echo "$compteur utilisateurs sont actuellement connectés";
33 ?></p>
34 </body>
35
36 </html>
```