

Troubleshooting WEB

Situation 1

1) Contexte

Le site web interne de l'entreprise qui vous consulte a toujours été accessible via `www.formation.lab` ou bien `192.168.0.4`. Malheureusement, ce site n'est plus accessible aujourd'hui. Pourriez-vous jeter un coup d'œil afin de trouver une solution au problème?

2) Outils utilisé & leur utilisation dans mon cas

- **Links** : Cette commande permet d'ouvrir un site web à partir d'une URL en ligne de commande. Cela confirme le bon fonctionnement de la connexion et s'assure que le serveur WEB fonctionne bien . *Mon Utilisation* : J'ai utilisé cette commande pour savoir si mon site web était accessible ou pas à partir du client.

3) Symptôme collecté

Après le lancement de la simulation ainsi que du service DHCP, DNS et SOA sans souci, j'ai voulu lancer le serveur web avec cette commande : `apache2ctl start`, mais celui-ci n'a pas fonctionné :

```
root@www:/# apache2ctl start
AH00534: apache2: Configuration error: No MPM loaded.
```

Suite à cela, j'ai voulu prendre un peu d'avance et vérifier que mes fichiers de log et `index.html` étaient bel et bien présents afin de ne plus avoir à les vérifier plus tard. Ils étaient bien tous les deux présents. Pour poursuivre mon débogage, je suis passé par le dossier de configuration d'Apache à `etc/apache`. À cet endroit, j'ai remarqué qu'il manquait le dossier `mods-enabled` qui contient tous les liens vers les fichiers des extensions spécifiques au serveur.

Pour remédier à cela, j'ai créé le dossier avec la commande `mkdir mods-enabled` pour ensuite exécuter ces deux commandes du TP8 afin d'activer toutes les extensions sur le serveur web : `a2enmod mpm_prefork` et `a2enmod authz_core`.

Une fois cela fait, je suis passé dans le fichier de configuration de base d'Apache nommé `apache2.conf`. Dans celui-ci, j'ai remarqué qu'Apache écoutait sur le port 8080 au lieu de 80. Le port 8080 est destiné à la connexion de périphériques tels que souris ou clavier et non à un serveur web. En revanche, le port 80 est celui par défaut du protocole HTTP, permettant de ne pas avoir à spécifier le port dans les URL.

Pour finir, j'ai voulu relancer Apache pour voir ce qu'il allait me dire. Au lancement de celui-ci, j'ai reçu un message d'erreur plus explicite que le premier, m'indiquant qu'aucune permission n'était spécifiée dans le fichier de configuration d'Apache.

```
root@www:/etc/apache2# apache2ctl start
[Tue Dec 12 23:36:24.839437 2023] [core:emerg] [pid 1065] (22)Invalid argument:
AH00024: Couldn't set permissions on the mpm-accept mutex; check User and Group
directives
(22)Invalid argument: could not create accept mutex
AH00015: Unable to open logs
Action 'start' failed.
```

4) Explication du bug

Le problème réside dans le fait qu'aucun utilisateur certaines autorisations n'est spécifié dans la configuration d'Apache.

5) Proposition de solution

Pour résoudre cela on peut réouvrir le fichier *apache.conf* et rajouter ces deux lignes

```
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}
```

Ces deux lignes permettent de spécifier sous quel utilisateur le serveur Apache doit fonctionner et donc récupérer ses permissions. Les variables entre crochets dirigent vers un fichier qui stocke de nombreuses variables, dont les deux utilisées dans l'image ci-dessus. Les deux dirigent vers la valeur "www-data", qui est un utilisateur ayant les permissions nécessaires pour lire les fichiers.

Une fois ces deux lignes ajoutées, on peut relancer le serveur Apache sans problème et vérifier au niveau du client si tout fonctionne avec la commande *links http://www.woodytoys.lab* pour ouvrir le site en ligne de commande.

```
Apache2 Ubuntu
Ubuntu Logo Apache2 Ubuntu Default Page
It works!
```

Situation 2

1) Contexte

La page d'accueil du site web interne de l'entreprise qui vous consulte a toujours été accessible depuis www.formation.lab ou encore 192.168.0.4. Aujourd'hui, ce lien ne donne plus de résultats.... Étrange car le site reste pourtant accessible via 192.168.0.4/index.html. Qu'a-t-il pu se passer ?

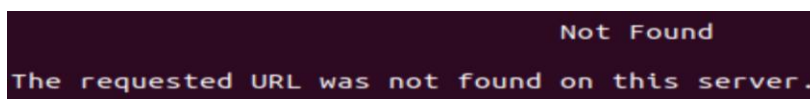
2) Outils utilisé & leur utilisation dans mon cas

- **Links** : Cette commande permet d'ouvrir un site web à partir d'une URL en ligne de commande. Cela confirme le bon fonctionnement de la connexion et s'assure que le serveur WEB fonctionne bien . *Mon Utilisation* : J'ai utilisé cette commande pour savoir si mon site web était accessible ou pas à partir du client.
- **Wireshark** : cette application permet d'analyser les paquets qui sont envoyés sur le réseau et de lire leur contenu. Cela facilite la détection d'erreurs dans les protocoles applicatifs. *Mon Utilisation* : J'ai utilisé Wireshark pour analyser le protocole html et comprendre qu'était la différence entre www.formation.lab et www.formation.lab/index.html au début du programme ou le deuxième lien ne fonctionnait pas mais pas le deuxième.

3) Symptôme collecté

Après le lancement de la simulation ainsi que du service DHCP, DNS, SOA et du serveur WEB sans souci, j'ai voulu expérimenter ce que la cliente disait ne pas fonctionner.

Mais avant cela, j'ai lancé une trace Wireshark pour écouter ce que le client reçoit et envoie, afin de pouvoir ensuite l'analyser. Suite à cela, j'ai exécuté la commande `link http://www.formation.lab` et `link http://www.formation.lab/index.html` chez le client. En effet, seule la deuxième commande affiche le site correctement. Le premier lien sans le `/index.html` indique que le fichier est introuvable sur le serveur.



Suite à cela, j'ai donc été voir la trace Wireshark que j'ai enregistré lors des requête pour les analyser.

Pour la requête avec l'URL <http://www.formation.lab/index.html> on peut voir en suivant le Stream que le client (rouge) a fait une requête *GET* avec comme route de fichier */index.html* et le protocole *HTMLI.1*, suite à cela on à l'host qui spécifie le nom de domaine du serveur avec www.formation.lab. Ensuite le serveur (bleu) va répondre par une requête 200 qui veut dire que tout est bon avec le contenu du fichier par la suite.

```
GET /index.html HTTP/1.1
Host: www.formation.lab
User-Agent: Links (2.20.2; Linux 5.4.0-91-generic x86_64; GNU C 9.2.1; text)
Accept: */*
Accept-Language: en,*;q=0.1
Accept-Encoding: gzip, deflate, br, zstd, bzip2, lzma, lzma2, lzip
Accept-Charset: us-ascii,ISO-8859-1,ISO-8859-2,ISO-8859-3,ISO-8859-4,ISO-8859-5,ISO-8859-6,ISO-8859-7,ISO-8859-8,ISO-8859-9,ISO-8859-10,ISO-8859-13,ISO-8859-14,ISO-8859-15,ISO-8859-16,windows-1250,windows-1251,windows-1252,windows-1256,windows-1257,cp437,cp737,cp850,cp852,cp866,x-cp866-u,x-mac,x-mac-ce,x-kam-cs,koi8-r,koi8-u,koi8-ru,TCVN-5712,VISCII,utf-8
Connection: keep-alive

HTTP/1.1 200 OK
Date: Wed, 13 Dec 2023 00:37:44 GMT
Server: Apache/2.4.41 (Ubuntu)
Last-Modified: Sun, 24 Oct 2021 10:35:55 GMT
ETag: "2aa6-5cf16cf2398c0"
Accept-Ranges: bytes
Content-Length: 10918
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
```

Mais pour la requête avec l'URL <http://www.formation.lab>, en resuivant le Stream. On peut voir que le client (rouge) envoie une requête *GET* avec comme protocole *HTMLI.1* et comme host www.formation.lab. Suite à cela le serveur web (bleu) va lui répondre qu'il n'a pas trouvé la ressource avec l'erreur 404 avec l'HTML de la page d'erreur par la suite.

```
GET / HTTP/1.1
Host: www.formation.lab
User-Agent: Links (2.20.2; Linux 5.4.0-91-generic x86_64; GNU C 9.2.1; text)
Accept: */*
Accept-Language: en,*;q=0.1
Accept-Encoding: gzip, deflate, br, zstd, bzip2, lzma, lzma2, lzip
Accept-Charset: us-ascii,ISO-8859-1,ISO-8859-2,ISO-8859-3,ISO-8859-4,ISO-8859-5,ISO-8859-6,ISO-8859-7,ISO-8859-8,ISO-8859-9,ISO-8859-10,ISO-8859-13,ISO-8859-14,ISO-8859-15,ISO-8859-16,windows-1250,windows-1251,windows-1252,windows-1256,windows-1257,cp437,cp737,cp850,cp852,cp866,x-cp866-u,x-mac,x-mac-ce,x-kam-cs,koi8-r,koi8-u,koi8-ru,TCVN-5712,VISCII,utf-8
Connection: keep-alive

HTTP/1.1 404 Not Found
Date: Wed, 13 Dec 2023 00:37:39 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Length: 196
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1
```

4) Explication du bug

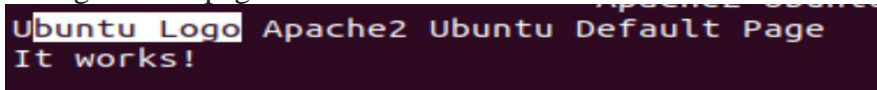
Les soucis dans ce cas ci est que le fichier *index.html* n'a pas été indiqué en tant que fichier par défaut dans la configuration d'apache.

5) Proposition de solution

Pour résoudre ce soucis, on doit aller modifier dans le fichier *apache2.conf* et rajouter ces lignes pour résoudre ce soucis :

```
<Directory /var/www/html/www>
    DirectoryIndex index.html
    Options FollowSymLinks
    Require all granted
</Directory>
```

Une fois cela fait, on peut relance le serveur WEB avec la commande *apache2ctl start* et passer du coter client pour tester la commande *link* <http://www.formation.lab> et regarder s'il est bien redirigé vers la page index.html.



On peut donc aussi analyser cela avec la trace Wireshark :

```
GET / HTTP/1.1
Host: www.formation.lab
User-Agent: Links (2.20.2; Linux 5.4.0-91-generic x86_64; GNU C 9.2.1; text)
Accept: */*
Accept-Language: en,*;q=0.1
Accept-Encoding: gzip, deflate, br, zstd, bzip2, lzma, lzma2, lzip
Accept-Charset: us-ascii,ISO-8859-1,ISO-8859-2,ISO-8859-3,ISO-8859-4,ISO-8859-5,ISO-8859-6,ISO-8859-7,ISO-8859-8,ISO-8859-9,ISO-8859-10,ISO-8859-13,ISO-8859-14,ISO-8859-15,ISO-8859-16,windows-1250,windows-1251,windows-1252,windows-1256,windows-1257,cp437,cp737,cp850,cp852,cp866,x-cp866-u,x-mac,x-mac-ce,x-kam-cs,koi8-r,koi8-u,koi8-ru,TCVN-5712,VISCII,utf-8
Connection: keep-alive

HTTP/1.1 200 OK
Date: Wed, 13 Dec 2023 01:43:23 GMT
Server: Apache/2.4.41 (Ubuntu)
Last-Modified: Sun, 24 Oct 2021 10:35:55 GMT
ETag: "2aa6-5cf16cf2398c0"
Accept-Ranges: bytes
Content-Length: 10918
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
```

Celle-ci nous indique bien que nous faisons une requête sans qu'aucun fichier ne soit spécialement précisé et pourtant tout s'est bien passé (200).