

Réseaux 1

TP 3 - HTTP, Serveurs et Clients W3

Objectif : tester le protocole HTTP, visualiser et analyser la configuration du serveur httpd Apache sur nyx

1. Le protocole HTTP

Dans la suite, vous ouvrirez une session telnet sur nyx depuis votre PC, ensuite vous contacterez le serveur Apache httpd sur nyx en utilisant la commande telnet de UNIX.

Requêtes et Réponses

- 1.1. Demander le document `/test0.html` en spécifiant le protocole HTTP/1.0 (la requête doit se terminer par une ligne vide)
- 1.2. Demander uniquement les en-têtes du document `/test0.html`
- 1.3. Quelle est le type (ou la marque) du serveur ?
- 1.4. Demander les méthodes de requête disponible pour le document `/test0.html`
- 1.5. Demandez la page `/test_img.html` contenant des images au format gif. Avez-vous récupéré le code des images ? D'après vous, quelles étapes sont suivies par Netscape ou Internet Explorer pour afficher une page contenant des images.

Erreurs du client

- 1.6. Obtenir des réponses avec un code statut indiquant une erreur du client.
 - Essayer une requête incorrecte (Erreur client 400 Bad Request)
 - Essayer une ressource inexistante (Erreur client 404 Not Found)

Erreurs du serveur

- 1.7. Obtenir une réponse avec un code statut indiquant une erreur du serveur.
 - Essayer une méthode non implémentée (Erreur serveur 501 Method Not Implemented)

Connexion persistante

Les versions 0.9 et 1.0 du protocole HTTP nécessitent une nouvelle connexion TCP à chaque requête, la version actuelle HTTP/1.1 (RFC 2616) autorise plusieurs échanges de requêtes/réponses via une unique connexion (notion de connexion persistante ou *keep alive*).

- 1.8. Essayer les trois échanges (requêtes/réponses) présentés ci-dessous et effectués successivement sur la même connexion TCP, dans la troisième requête nous avons ajouté le champ d'en-tête `Connection : close` afin de demander la fermeture de la connexion après la réponse (il faut taper les commandes rapidement, la connexion étant fermée automatiquement au bout de quelques secondes d'inactivité)

Requêtes conditionnelles

- 1.9. Demander le document `/test0.html` s'il a été modifié depuis la date de votre dernière consultation.

2. Analyser la configuration du serveur Apache

Le serveur **Apache** sur *nyx* est installé dans `/etc/apache-ssl`

Les fichiers de configuration se trouvent dans le sous répertoire `conf`

- `httpd.conf` contient les directives de configuration générale
- `srm.conf` contient les directives concernant les ressources du serveur
- `access.conf` contient les directives concernant la politique d'accès au serveur.

Dans la nouvelle version 1.3 de Apache, les contenus des trois fichiers sont regroupés dans `httpd.conf`

Configuration générale

Consultez le fichier **httpd.conf** pour répondre aux questions suivantes

- 2.1. Comment le démon `httpd` est-il lancé sur *nyx* ?
- 2.2. Sur quel port le serveur `httpd` est-il à l'écoute ?
- 2.3. Quel est le *user id* utilisé par le processus `httpd` répondant à la requête d'un client ?

Ressources du serveur

– Emplacement des documents HTML

- 2.4. Où sont stockés les documents HTML diffusé par notre serveur W3 (pour chaque serveur virtuel) ?
- 2.5. Où sont stockés les documents HTML des utilisateurs sur *nyx* ?
- 2.6. Quelles sont les protections UNIX que vous devez affecter aux documents HTML dans votre répertoire personnel afin qu'ils soient accessibles par le serveur W3 ?

– Gestion des listes de fichiers

- 2.7. Quel est le nom de fichier pris par défaut lorsqu'une requête porte sur un répertoire ?

– Emplacement des scripts CGI

- 2.8. Où sont stockés les scripts CGI du serveur `www_iutinfo.unice.fr` ?

– Traduction des URLs

- 2.9. Indiquer les URLs permettant de référencer sur `www-iutinfo` les documents HTML suivants :
 - le document `doc1.html` dans l'espace web du serveur `www-iutinfo.unice.fr`
 - le document `doc2.html` dans le répertoire web de l'utilisateur `fred`.
- 2.10. Indiquer les URLs permettant de référencer sur `www_iutinfo` les scripts CGI suivants :
 - le script `script1.sh` dans le répertoire `cgi-bin` du serveur,
 - le script `script2.cgi` dans l'espace web de l'utilisateur `fred`.
- 2.11. Indiquer comment le serveur `httpd` sur *nyx* traduit les requêtes suivantes (sans préciser un nom de serveur virtuel)
 1. GET /
 2. GET /~fred/
 3. GET /~fred/doc.html
 4. GET /cgi-bin/script.sh
 5. GET /~fred/script.cgi

Mettre en place votre espace WEB sur nyx

- 2.12. Testez dans votre navigateur l'url : <http://nyx.unice.fr/~my-username> (où *my-username* représente votre nom d'utilisateur sur *nyx*) après chacune des étapes décrites en dessous, observez bien les différentes réponses successives du serveur
1. se loger par telnet sur nyx
 2. créez dans votre répertoire \$HOME sur nyx un répertoire nommé web

```
nyx$ mkdir web
```
 3. changez les permissions :

```
nyx$ chmod o+x .  
nyx$ chmod o+rx web
```
 4. créez dans votre répertoire web un fichier index.html

```
nyx$ cd web  
nyx$ cat > index.html  
<h1>bienvenue chez moi</h1>  
^D
```
 5. changez les permissions de index.html

```
nyx$ chmod o+r index.html
```

Politique d'accès au serveur

– Protection des accès par domaines

- 2.13. Créer un répertoire R1 dans votre répertoire web, puis interdire tous les accès dans R1, sauf aux utilisateurs venant de votre station NT, vérifiez.

– Protection des accès par nom d'utilisateur et mot de passe

- 2.14. Créer un répertoire R2 dans votre répertoire web personnel, puis autoriser l'accès dans R2 uniquement à *david* avec le mot de passe *david**** et à *steve* avec le mot de passe *steve**** . Vérifiez.

– Trace des accès clients

- 2.15. Dans quel fichier sont notifiés tous les accès des clients à notre serveur *www-iutinfo* ?
- 2.16. Consulter les 10 derniers accès à notre serveur *www-iutinfo* (utiliser la commande *tail* de UNIX).