



NE302 – Projet Réseau

Projet « HTTP Server » - part 1

1 Travail préparatoire

Lisez la documentation de l'outil netcat (dans un terminal, faire man netcat).

Une première pré-lecture de la RFC HTTP est préférable si le sujet vous est inconnu.

Une première pré-lecture de la spécification fastCGI est préférable si le sujet vous est inconnu.

2 Objectifs de la séance 1 – Projet HTTP

Cette séance 1 va permettre :

- la (re)découverte du protocole HTTP en utilisant un serveur web « lourd ».
- d'effectuer des captures réseaux vous permettant de mieux comprendre les spécifications des RFCs sur HTTP. Vous devrez conserver ces captures comme références pour vos travaux ultérieurs.
- d'expérimenter la configuration du serveur PHP en mode fastCGI, de prendre aussi des captures réseaux servant de références pour la compréhension des spécifications et la réalisation de la partie 4.

Pour les captures réseaux utilisez wireshark et sauvegardez les packet au format pcap.
N'hésitez pas à poser des questions pendant la séance....

3 Installation d'un serveur web « lourd »

Installez le serveur web avec le package apache2 avec les commandes suivantes :

```
#apt-get update
```

```
#apt-get install apache2
```

Vérifiez que le serveur écoute sur le port 80, avec la commande

```
# netstat -antp
```

créer le répertoire /var/www/html2 et copiez le fichier /var/www/html/index.html dedans, modifiez le contenu de manière à pouvoir le différencier.

Ajouter les lignes suivantes à la fin du fichier de configuration

```
/etc/apache2/sites-enabled/000-default
```

```
<VirtualHost *:80>
```

```
    ServerName www.toto.com
```

```
    DocumentRoot /var/www/html2
```

```
</VirtualHost>
```

et redémarrez le serveur :

```
# /etc/init.d/apache2 restart
```

Ajouter à la fin de la ligne dans le fichier /etc/hosts

```
127.0.0.1    jessie    www.toto.com
```

4 Découverte de HTTP avec netcat

Le protocole HTTP fonctionne au dessus du protocole TCP, il permet de manipuler des ressources distantes. Il est principalement utilisé pour le web. Le client envoie des requêtes avec le format suivant :

```
Ligne de commande (Commande, URL, Version de protocole)
En-tête de requête
[Ligne vide]
Corps de requête
```

La ligne de commande indique ce que le client attend, par exemple obtenir la page d'accueil : GET /index.html HTTP/1.0

L'entête de requête contient une succession de couples (variable : valeur) , par exemple User-Agent : Mozilla. Elle permet au client de donner des informations à son propos ou de préciser la requête.

La ligne vide est très importante elle distingue le header HTTP du reste du message.

Le corps de la requête peut contenir des données additionnelles (dans le cas de messages POST par exemple)

Le serveur répond avec un format similaire :

```
Ligne de statut (Version, Code-réponse, Texte-réponse)
En-tête de réponse
[Ligne vide]
Corps de réponse
```

La ligne de status indique au client si sa requête a réussi. Parmi les réponses possibles:

- HTTP/1.0 200 OK indique le succès de l'opération
- HTTP/1.0 404 Not Found, indique que la page demandée n'existe pas, etc...

L'entête permet au serveur de préciser la réponse par exemple en indiquant la taille et le type de la ressource demandée.

La ligne vide sépare l'entête de la ressource demandée dans le corps du message.

Naviguer sur votre station de travail avec un navigateur classique.

L'URL est http://127.0.0.1/index.html

Faites des captures réseaux, quelle version de HTTP est utilisée par votre navigateur.

Combien d'entête y a-t-il dans la requête ? A quoi servent-ils ?

Combien d'entête y a-t-il dans la réponse ?

Faites ensuite la même navigation avec l'utilitaire netcat, en tapant directement à la main les commandes HTTP (vous resterez avec le protocole HTTP/1.0).

Faites des captures réseaux en utilisant le protocole HTTP/1.0 et HTTP/1.1 en utilisant une

requête GET, quels sont les entêtes nécessaires pour obtenir une réponse de type 200 OK de la part du serveur ?

Essayez de trouver les entêtes permettant d'accéder au site correspondant à www.toto.com que vous avez ajouté en début de séance.

5 Utilisation de HTML

HTML est le format de données permettant de représenter les pages web.

Voici un exemple de fichier HTML :

```
<html>
<head>
  <title>
    Exemple de HTML
  </title>
</head>
<body>
  Ceci est une phrase avec un <a href="cible.html">hyperlien</a>.
  <p>
    Ceci est un paragraphe où il n'y a pas d'hyperlien.
  </p>
</body>
</html>
```

Ecrire une page HTML simple à votre convenance, et afficher ensuite cette page dans votre navigateur. Faites une page contenant une image.

6 Exercice 3 : Installation de PHP en mode fastCGI

Installez le serveur PHP5 en mode fpm (FastCGI Process Manager) :

```
#apt-get install php5-fpm
```

Modifiez la ligne 38 du fichier /etc/php5/fpm/pool.d/www.conf

```
listen = 9000
```

Relancez le processus, et vérifiez qu'il écoute sur le port 9000

```
# /etc/init.d/php5-fpm restart
```

```
#netstat -antp
```

Le processus PHP écoute sur le port 9000 les requêtes que devra relayer le serveur web pour les scripts CGI. Quand le serveur relaye les requêtes, il reprend les informations émises par le client HTTP dans les requêtes (POST ou GET), et les remet en forme en suivant les spécifications FastCGI pour les envoyer via la connexion TCP sur le port 9000.

Vous allez capturer ces échanges sur le port 9000. Mais avant cela il faut configurer le serveur apache pour rediriger les requêtes PHP. Ceci nécessite un package spécifique... et donc un peu de configuration supplémentaire. Pour cela ajouter dans le fichier /etc/apt/sources.list le mot clef non-free à

la fin des lignes :

```
deb http://ftp.fr.debian.org/debian/ jessie main non-free
deb-src http://ftp.fr.debian.org/debian/ jessie main non-free
deb http://security.debian.org/ jessie/updates main non-free
deb-src http://security.debian.org/ jessie/updates main non-free
```

puis installez le package :

```
# apt-get update
# apt-get install libapache2-mod-fastcgi
# a2enmod actions alias fastcgi
# /etc/init.d/apache2 restart
```

Configurez apache en ajoutant dans le fichier /etc/apache2/sites-enabled/000-default (dans la section virtualhost), les lignes suivantes :

(les plus curieux pourront regarder la doc de chaque directive...)

```
FastCGIExternalServer /var/www/fcgi -host 127.0.0.1:9000
AddHandler php-fastcgi .php
Action php-fastcgi /var/www/virtualpath virtual
Alias /virtualpath /var/www/fcgi
```

Créez le repertoire /var/www/fcgi dans lequel vous créez un fichier index.php contenant.

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

Bon tout ça pour ça.... Maintenant appelez <http://127.0.0.1/index.php> depuis un navigateur et prenez les captures réseaux sur le port 9000 pour observer les formats de messages fastCGI....