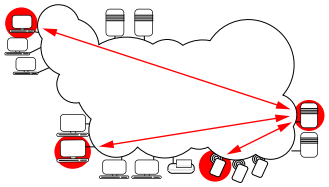
ARes - Lab n°2

Couche application (1) : Telnet, SSH, FTP, TFTP et Web

Lors du Lab n°1, vous avez appris comment utiliser la plateforme d'expérimentation et vous l'avez exploitée afin de générer et d'analyser des traces assez simples de la couche application, contenant du trafic web. Pour le Lab n°2, vous allez explorer la couche application beaucoup plus en détail, en étudiant les protocoles suivants : TELNET, SSH, FTP, SFTP, TFTP et HTTP. Pour chacun, vous allez générer du trafic réel que vous allez capturer et analyser avec l'outil `wireshark/tshark`. Vous utiliserez également le RFC de l'un de ces protocoles (FTP) afin de mieux comprendre son trafic.



1 Exercices d'échauffement (sans machine)

1. Qu'est-ce qu'un protocole applicatif ?

2. Quels programmes accédant au réseau utilisez-vous couramment ? Savez-vous quels sont les protocoles applicatifs que ces programmes utilisent ?

3. Sur quel modèle de communication s'appuient principalement les applications actuelles ? Comment identifier les rôles des participants pour les applications citées précédemment ?

4. Décrivez les grandes catégories d'applications utilisant les réseaux. Pour chacune d'elles, indiquez les besoins en termes de débit, de tolérance à la variation ce débit, de sensibilité aux pertes et de contraintes temporelles.

2 Connexion à distance

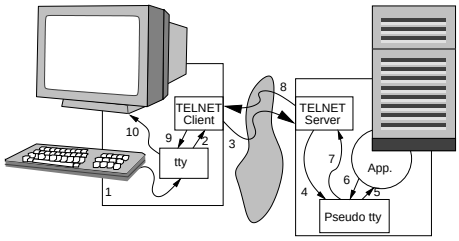
2.1 Rappels

1. Quelle est l'utilité des applications de connexion à distance *(remote login)* ?

2. Quels types d'informations sont échangés par ce genre d'application ?

3. Quelles contraintes peut poser ce type d'application ? Citez des exemples.

4. Quel type de service réseau doivent utiliser ces applications ?



2.2 Protocole TELNET

TELNET facilite la connexion à distance. Il est l'un des protocoles les plus anciens de l'environnement TCP/IP. (Le RFC 854 a été publié en 1983.) Il doit donc fonctionner avec un existant très important de machines et de terminaux et permettre la négociation de nombreux paramètres optionnels (généralement à l'ouverture de la communication) pour s'adapter aux besoins des deux extrémités.

L'hétérogénéité potentielle des deux hôtes impliqués dans l'échange nécessite un service de terminal virtuel, c'est-à-dire un encodage commun à travers le NVT (encodage proche de l'ASCII 7bits pour les caractères imprimables), évitant d'avoir à connaître la correspondance des caractères entre chaque type de destinataires (ce point est important car le NVT est également l'encodage habituel des applications textuelles de TCP/IP).

Le protocole TELNET repose sur une connexion réseau TCP (port serveur **23**) afin de garantir la fiabilité de l'échange. Le contrôle est dit *In-band* : les données circulent dans la même connexion que les informations de négociation. Comme la plupart des protocoles anciens de TCP/IP, il n'intègre aucun mécanisme de sécurité (pas de confidentialité).

Dans l'analyse qui va suivre, vous allez analyser les deux phases caractéristiques du protocole TELNET : négociation puis échange de données.

2.2.1 Capture d'un trafic TELNET

Cette première capture a pour but de percevoir la nature du trafic TELNET. Sur la plateforme d'expérimentation, la topologie 1 a été configurée (postes clients et serveur sur le même LAN). Réalisez la capture de trafic TELNET à l'aide du logiciel `wireshark` ou `tshark` :

- A partir du poste PPTI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles
  - fenêtre 1, accédez à la vm**N**1 (hôte "client") via SSH vers `etudiant@10.0.7.`**N**1 (utilisez `-Y` pour la redirection X11 d'un client graphique)
  - fenêtre 2, accédez à la vm**N**2 (hôte "sonde") via SSH vers `etudiant@10.0.7.`**N**2 (utilisez `-Y` pour la redirection X11 de `wireshark`)
  - fenêtre 3, accédez à la vm**N**3 (hôte "serveur") via SSH vers `etudiant@10.0.7.`**N**3

- Vérifiez que TELNET tourne sur la VM serveur (fenêtre 3)
  - recherchez le processus du serveur TELNET, tapez `ps aux | grep telnetd` ou `inetd` (avec la configuration adéquat du service TELNET dans le fichier `/etc/inetd.conf`)
  - visualisez la configuration des interfaces, en particulier celle sur le LAN d'étude (`/sbin/ifconfig eth1`) pour vérifier l'adresse IP du serveur pour la connexion du client (devrait être `10.`**N**`.1.`**N**3)
- Lancez l'analyseur sur `10.0.7.`**N**2 (fenêtre 2)
  - si vous lancez l'analyseur graphique, tapez : `wireshark`, puis initier la capture sur l'interface `eth1`, comme indiqué précédemment
  - si vous êtes en mode texte, tapez `tshark -i eth1 -w <maCapture>`
- Démarrez un client TELNET sur `10.0.7.`**N**1 (fenêtre 1)
  - lancez le client en établissant une connexion vers le serveur, tapez : `telnet 10.`**N**`.1.`**N**3
  - identifiez-vous avec le login `etudiant` et le mot de passe correspondant puis tapez quelques commandes UNIX avant de terminer votre session TELNET (fermeture de la session par la commande `exit`)
- Observez la capture réalisée dans la fenêtre de `wireshark` (en la rapatriant préalablement si vous avez utilisé `tshark`)
- **Filtrez le trafic afin de ne conserver que celui relatif à TELNET** (filtre = `telnet`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

2.2.2 Analyse de la négociation TELNET

Les négociations ont principalement lieu au début de la connexion. Elles sont composées de commandes qui peuvent être envoyées dans chaque sens. Chaque commande démarre par la **commande d'échappement** qui est codée sur 1 octet (début d'une commande) : `IAC`=`0xff`. Les **commandes de négociation** d'option (sur 1 octet) sont immédiatement suivies de la valeur de l'option (sur 1 octet) : `WILL`=`0xfb` (indique ce qu'une entité va faire), `WONT`=`0xfc` (ne pas faire), `DO`=`0xfd` (demande à l'autre entité de faire), `DON'T`=`0xfe` (demande de ne pas faire). Exemple :

`IAC`, `DO`, `24`

Les **sous-options** sont transmises (après une demande à l'aide d'un `WILL` puis d'une confirmation avec un `DO`) entre les deux commandes (sur 1 octet) suivantes : `SB`=`0xfa` et `SE`=`0xf0`. Une sous-option se compose du code de l'option sur 1 octet, 1 octet nul puis la valeur de l'option. Exemple :

`IAC`, `SB`, `24`, `0`, `'V'`, `'T'`, `'2'`, `'2'`, `'0'`, `IAC`, `SE`

$(Value)_{10}$	**Option name**
1	Echo
3	Suppress Go Ahead
5	Status
6	Timing Mark
24	Terminal Type
31	Negotiate About Window Size
32	Terminal Speed
33	Remote Flow Control
34	Linemode
35	X Display Location
36	Environment variables
39	New Environment Option
...	

Dans la capture réalisée, essayez de trouver la signification des différents paramètres négociés en observant seulement la partie présentant les octets en hexadécimal de `wireshark`.

1. Analysez les différentes options et sous-options échangées.

2. Trouvez combien de temps dure la négociation.

2.2.3 Analyse de l'échange de données TELNET

Passez les quelques trames de négociation.

1. Quand démarre l'émission de données TELNET ?

2. Quelles transmissions sur le réseau la frappe d'un caractère par l'utilisateur génère-t-elle lors d'une session TELNET ?

3. Que pensez-vous de l'efficacité du protocole ?

4. Quel est le degrès d'interactivité ?

5. Quelles informations sont véhiculées dans l'échange de données ?

2.2.4 Trace TELNET longue distance (*facultatif... à traiter de manière autonome si vous êtes nettement en avance par rapport au reste du groupe*)

A partir d'une trace contenant une heure de trafic longue distance entre le *Lawrence Berkeley Laboratory* et le reste du monde en janvier 1994, retrouvez des exemples de communications TELNET.

Ces traces , initialement au format `tcpdump` (le format standard de trace utilisé aussi par `wireshark`/`tshark`), ont été converties en ASCII en prenant soin de renuméroter les adresses IP et de supprimer le contenu des paquet[1].

En voici un extrait :

```
8.430376 22 21 23 33281 1
8.437539 3 4 3930 119 47
8.442644 4 3 119 3930 15
8.454895 26 11 4890 23 1
8.459398 5 2 14037 23 0
8.469004 4 23 4464 119 512
```

La première colonne contient une estampille temporelle relative au début de la capture (exprimée en secondes), les deux colonnes suivantes sont les adresses sources et destinations renumérotées par ordre d'apparition, ensuite se trouvent les numéros de port puis la taille des données (en octets).

Chargez la trace `tme2-lbl.txt.gz`, soit à partir du répertoire `/Infos/lmd/2022/master/ue/MU4IN001-2022oct`, soit sur la page `http://www-npa.lip6.fr/~fourmaux/Traces/labV8.html`, vers un répertoire **local** (ex : `/tmp`)[2]. Puis à l'aide des outils UNIX standard (`awk`, `perl`, `sed`...), isolez un des flots TELNET et identifiez ses caractéristiques typiques. **Ne demandez pas à votre encadrant d'aide sur ces outils, il est là pour répondre à vos questions liées au réseau**.

[1] The trace lbl-pkt-4 ran from 14 :00 to 15 :00 on Friday, January 21, 1994 (times are Pacific Standard Time) and captured 1.3 million TCP packets, the dropping about 0.0007 of the total. The tracing was done on the Ethernet DMZ network over which flows all traffic into or out of the Lawrence Berkeley Laboratory, located in Berkeley, California. The raw trace was made using tcpdump on a Sun Sparcstation using the BPF kernel packet filter. Timestamps have microsecond precision. The trace has been "sanitized" using the sanitize scripts. This means that the host IP addresses have been renumbered, and all packet contents removed. The trace was made by Vern Paxson (vern@ee.lbl.gov). The trace may be freely redistributed.

[2] La taille de la trace étant particulièrement importante, si vous travaillez sur votre compte qui est monté par NFS vous obtiendrez des temps de réponse très mauvais.

1. Réalisez un chronogramme rapide de quelques échanges TELNET se trouvant dans la trace. Que pouvez-vous dire de l'interactivité ?
2. La sonde est-elle proche de l'émetteur ?
3. Pouvez-vous faire des hypothèses sur le type d'informations échangées ?

2.2.5 Sans la plateforme...

En cas de problème d'accès à la plateforme d'expérimentation ou si vous souhaitez réviser sur une autre machine, vous pouvez télécharger la trace `tme2-tel.dmp` (similaire à celle capturée précédemment) soit à partir du répertoire `/Infos/lmd/2022/master/ue/MU4IN001-2022oct`, soit sur la page web `http://www-npa.lip6.fr/~fourmaux/Traces/labV8.html`, puis l'analyser avec le logiciel `wireshark` (sans avoir besoin des droits d'administrateur).

2.3 Protocole SSH

Ce protocole remplace généralement TELNET car c'est un protocole de connexion à distance intégrant des mécanismes de sécurité : SSH garantit l'authentification, la confidentialité et l'intégrité des communications. SSH utilise une connexion TCP sur le port serveur **22**.

De nombreuses possibilités d'utilisation sont associées à SSH pour profiter de ses mécanismes de sécurité : multiplexage de plusieurs flux dans une connexion, utilisation d'une connexion SSH comme couche transport pour d'autres applications (vous pouvez, par exemple, créer une connexion SSH entre votre machine résidentielle et le serveur d'accès de l'université, non seulement pour réaliser une connexion à distance textuelle, mais également pour rediriger du trafic entre un client applicatif local et un serveur du centre de calcul de l'université)...

Dans l'analyse qui va suivre, vous allez étudier le protocole SSH en essayant d'effectuer les mêmes interactions que précédemment au niveau de l'utilisateur.

2.3.1 Capture d'un trafic SSH

Cette troisième capture a pour but de percevoir la nature du trafic SSH. Sur la plateforme d'expérimentation, toujours sur la topologie 1 (postes client et serveur sur le même LAN), réalisez la capture de trafic SSH à l'aide du logiciel `wireshark/tshark` :

- A partir du poste PPTI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles.
- Vérifiez que le serveur SSH (`sshd`) tourne sur `10.0.7.`**N**3 (fenêtre 3)
- Démarrez la capture en lançant l'analyseur sur sur l'interface `eth1` de l'hôte `10.0.7.`**N**2 (fenêtre 2)
- Démarrez un client SSH sur `10.0.7.`**N**1 (fenêtre 1)
  - lancez le client en établissant une connexion vers le serveur, tapez : `ssh 10.`**N**`.1.`**N**3 (réseau d'expérimentation)
  - identifiez-vous avec le login `etudiant` et le mot de passe correspondant puis tapez les quelques commandes exécutées précédemment lors des captures TELNET
- Observez la capture réalisée dans la fenêtre de `wireshark` (en la rapatriant préalablement si vous avez utilisé `tshark`)
- **Filtrez le trafic afin de ne conserver que celui relatif à SSH** (filtre = `ssh`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

2.3.2 Analyse de l'échange SSH

1. Qu'observez-vous au début de l'échange ?

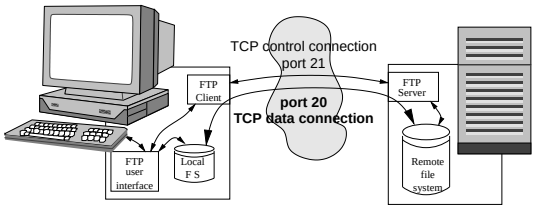
2. Pouvez-vous observer des différences avec TELNET (l'échange réalisé est identique aux précédents en terme de données utilisateur) ?

2.3.3 Trace SSH longue distance (*facultatif... à traiter de manière autonome si vous êtes en avance par rapport au reste du groupe*)

1. Toujours par rapport à la trace `tme2-lbl.txt.gz` chargée précédemment, identifiez des communications SSH.

2.3.4 Sans la plateforme...

En cas de problème d'accès à la plateforme d'expérimentation ou si vous souhaitez réviser sur une autre machine, vous pouvez télécharger la trace `tme2-ssh.dmp` (similaire à celle capturée précédemment) soit à partir du répertoire `/Infos/lmd/2022/master/ue/MU4IN001-2022oct`, soit sur la page web `http://www-npa.lip6.fr/~fourmaux/Traces/labV8.html`, puis l'analyser avec le logiciel `wireshark` (sans avoir besoin des droits d'administrateur).



3 Transfert de fichiers

3.1 Protocole FTP

3.1.1 Etude du RFC 959 (sans machine)

Dans cette section nous allons travailler sur un **RFC** *(Request For Comments)* produit par les groupes de travail de l'**IETF** *(Internet Engineering Task Force)* afin d'assurer la standardisation des protocoles de l'Internet. Le **RFC 959** présente une forme classique et donne un aperçu de ce type de document. Récupérez le **RFC 959** sur le site dédié de l'IETF :

- démarrez un navigateur et accédez à la page `http://www.rfc-editor.org/`
- cliquez sur RFC SEARCH et précisez le terme "FTP" pour démarrer la recherche
- sélectionnez le document **RFC 959** dans le résultat de la recherche

Ouvrez ce document et parcourez en rapidement le contenu, puis répondez aux questions suivantes :

1. Que pouvez-vous dire sur la forme du document ? Quelles sont les différentes sections abordées dans ce document ?

2. Précisez l'architecture de communication de FTP. Pourquoi dit-on que les informations de contrôle circulent "hors-bande" ?

3. Quelles sont les différentes commandes à la disposition du client ?

4. Pouvez-vous citer les différents types d'erreur que peut signaler FTP ? Comment les informations d'erreur sont-elles transmises ?

3.1.2 Capture d'un trafic FTP

Cette capture a pour but de percevoir la nature du trafic FTP. Sur la plateforme d'expérimentation, toujours sur la topologie 1 (postes client et serveur sur le même LAN), réalisez la capture de trafic FTP à l'aide du logiciel `wireshark/tshark` :

- A partir du poste PPTI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles.
- Vérifiez que le serveur FTP (`ftpd`) tourne sur `10.0.7.`**N**3 (fenêtre 3)

- Démarrez la capture en lançant l'analyseur sur sur l'interface `eth1` de l'hôte `10.0.7.`**N**2 (fenêtre 2)
- Démarrez un client FTP sur `10.0.7.`**N**1 (fenêtre 1)
  - lancez le client en établissant une connexion vers le serveur, tapez : `ftp 10.`**N**`.1.`**N**3 (réseau d'expérimentation)
  - identifiez-vous avec le login `etudiant` et le mot de passe correspondant
  - déplacez-vous dans le système de fichiers du serveur (commandes de l'interface utilisateur `pwd`, `cd` et `dir`)
  - choisissez un fichier et transférez le sur la machine client (commandes de l'interface utilisateur `get`)
  - terminez l'échange (commandes de l'interface utilisateur `quit`)
- Observez la capture réalisée dans la fenêtre de `wireshark` (en la rapatriant préalablement si vous avez utilisé `tshark`)
- **Filtrez le trafic afin de ne conserver que celui relatif à FTP** (filtre = `ftp or ftp-data`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

3.1.3 Analyse de la connexion FTP de contrôle

Retrouvez la correspondance entre les messages échangés sur le réseau (sur la connexion de contrôle de FTP) et ceux affichés par l'application (interface utilisateur sur le client).

1. Par définition, qui initie la communication entre le client et serveur ? Peut-on l'observer dans la capture ?

2. Quelle commande du protocole FTP identifie l'utilisateur ? Dans la capture, quelle information pouvez-vous observer sur celui-ci ?

3. Quelle est la commande qui authentifie ensuite l'utilisateur ? Le mot de passe apparaît-il en clair sur le réseau ?

4. Quel est l'intérêt de la commande suivant l'authentification ?

5. A quoi sert la commande `PORT` ? Analysez ses paramètres. Pourquoi est-elle émise à ce point de l'échange ?

6. La commande `LIST` permet d'obtenir la liste des fichiers du répertoire courant au niveau du serveur. Pourquoi est-elle suivie de deux messages envoyés par le serveur ?

7. Quelles sont les autres commandes que vous observez ? A quoi servent-elles ?

8. A quels moments de la transaction ont lieu les transferts de fichiers ?

3.1.4 Analyse de la connexion FTP de données

1. A quoi correspondent les données échangées sur les connexions de tranfert de données ?

2. Sur quels ports ces données sont-elles envoyées ?

3. Quelle synchronisation observez-vous entre les messages sur la connexion de contrôle et ceux de la connexion de données ?

3.1.5 Trace FTP longue distance (*facultatif... à traiter de manière autonome si vous êtes en avance par rapport au reste du groupe*)

1. Toujours par rapport à la trace `tme2-lbl.txt.gz` chargée précédemment, identifiez des communications FTP avec les connexions `FTP` et `FTP-DATA` associées.
2. Tracez le chronogramme.
3. Que pouvez vous dire de l'interactivité par rapport à TELNET ?

3.1.6 Sans la plateforme...

En cas de problème d'accès à la plateforme d'expérimentation ou si vous souhaitez réviser sur une autre machine, vous pouvez télécharger la trace `tme2-ftp.dmp` (similaire à celle capturée précédemment) soit à partir du répertoire `/Infos/lmd/2022/master/ue/MU4IN001-2022oct`, soit sur la page web `http://www-npa.lip6.fr/~fourmaux/Traces/labV8.html`, puis l'analyser avec le logiciel `wireshark` (sans avoir besoin des droits d'administrateur).

3.2 Protocoles SCP/SFTP

Plusieurs protocoles permettent la transmission de fichier sécurisée. L'application `scp` est un client de la suite logicielle associée à SSH (transfert de fichiers à un serveur `sshd`). `scp` s'utilise de manière similaire à `rcp` (copie à distance de la famille des `r*`-commandes UNIX). Il existe également le client `sftp` qui conserve un mode de fonctionnement similaire à FTP dans un tunnel SSH (un serveur spécifique `sftp-server` lui est dédié).

3.2.1 Capture d'un trafic SCP/SFTP

Cette capture a pour but de percevoir la nature du trafic associé à un transfert de fichier avec un protocole sécurisé. Sur la plateforme d'expérimentation, toujours sur la topologie 1 (postes client et serveur sur le même LAN), réalisez la capture de trafic SCP ou SFTP[3] à l'aide du logiciel `wireshark/tshark` :

- A partir du poste PPTI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles.
- Vérifiez que le serveur SCP (`sshd`) ou SFTP (`sftp-server`) tourne sur `10.0.7.`**N**3 (fenêtre 3)
- Démarrez la capture en lançant l'analyseur sur sur l'interface `eth1` de l'hôte `10.0.7.`**N**2 (fenêtre 2)
- Démarrez un client SCP ou SFTP sur `10.0.7.`**N**1 et récupérez à partir du serveur le fichier précédemment transféré (fenêtre 1)

  SCP tapez : `scp etudaint@10.`**N**`.1.`**N**`3:<fichier_distant> <fichier_local>` puis authentifiez-vous

  SFTP tapez : `sftp 10.`**N**`.1.`**N**3 puis authentifiez-vous et tapez les quelques commandes exécutées précédemment lors de la capture FTP
- Observez la capture se réaliser dans la fenêtre de `wireshark` (en la rapatriant préalablement si vous avez utilisé `tshark`)
- **Filtrez le trafic afin de ne conserver que celui relatif à SCP ou SFTP** (filtre = `ssh`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

[3] Nous notons SCP ou SFTP les trafics associé aux applications `scp` ou `sftp`.

3.2.2 Analyse de l'échange SCP/SFTP

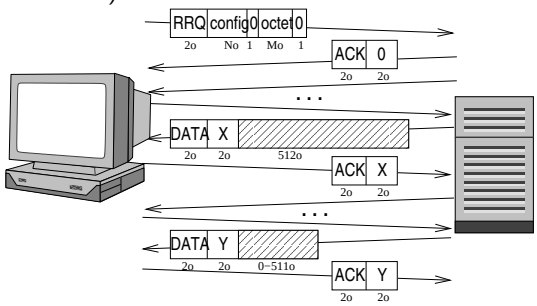
1. Rappelez le fonctionnement des protocoles SCP et SFTP.

2. Pouvez-vous faire la correspondance entre les messages du terminal et les trames échangées ?

3. Quelles différences constatez-vous avec FTP ?

3.2.3 Sans la plateforme...

En cas de problème d'accès à la plateforme d'expérimentation ou si vous souhaitez réviser sur une autre machine, vous pouvez télécharger la trace `tme2-scp.dmp` (similaire à celle capturée précédemment) soit à partir du répertoire `/Infos/lmd/2022/master/ue/MU4IN001-2022oct`, soit sur la page web `http://www-npa.lip6.fr/~fourmaux/Traces/labV8.html`, puis l'analyser avec le logiciel `wireshark` (sans avoir besoin des droits d'administrateur).



3.3 Protocole TFTP

3.3.1 Capture d'un trafic TFTP

Cette dernière capture a pour but de percevoir la nature du trafic TFTP. Sur la plateforme d'expérimentation, toujours sur la topologie 1 (postes client et serveur sur le même LAN), réalisez la capture de trafic TFTP à l'aide du logiciel `wireshark/tshark` :

- A partir du poste PPTI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles.
- Vérifiez que le serveur TFTP (`tftpd`) tourne sur `10.0.7.`**N**3 et qu'un répertoire est configuré pour réaliser les transferts (fenêtre 3). Observez les fichiez disponibles dans ce répertoire.
- Démarrez la capture en lançant l'analyseur sur sur l'interface `eth1` de l'hôte `10.0.7.`**N**2 (fenêtre 2)
- Démarrez un client TFTP sur `10.0.7.`**N**1 (fenêtre 1)
  - lancez le client en accédant au serveur, tapez : `tftp 10.`**N**`.1.`**N**3 (réseau d'expérimentation)
  - récupérez un fichier du serveur avec la commande `get`
  - terminez l'échange avec la commande `quit`
- Observez la capture réalisée dans la fenêtre de `wireshark` (en la rapatriant préalablement si vous avez utilisé `tshark`)
- **Filtrez le trafic afin de ne conserver que celui relatif à TFTP** (filtre = `tftp`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

3.3.2 Analyse de l'échange TFTP

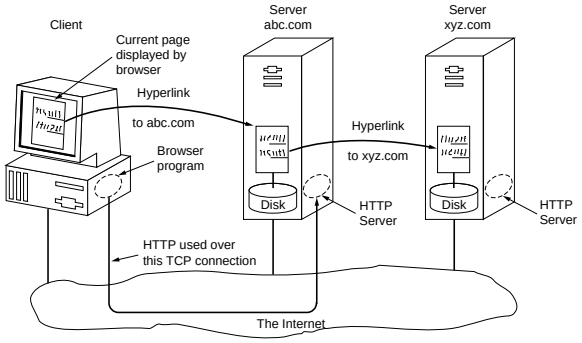
1. Rappelez le fonctionnement du protocole TFTP.

2. Pouvez-vous faire la correspondance entre les messages du terminal et les trames échangées ?

3. Quelles différences constatez-vous avec FTP ?

3.3.3 Sans la plateforme...

En cas de problème d'accès à la plateforme d'expérimentation ou si vous souhaitez réviser sur une autre machine, vous pouvez télécharger la trace `tme2-tft.dmp` (similaire à celle capturée précédemment) soit à partir du répertoire `/Infos/lmd/2022/master/ue/MU4IN001-2022oct`, soit sur la page web `http://www-npa.lip6.fr/~fourmaux/Traces/labV8.html`, puis l'analyser avec le logiciel `wireshark` (sans avoir besoin des droits d'administrateur).



4 Trafic web

4.1 Exercices (sans machine)

1. Expliquez les étapes nécessaires à la récupération d'une page web. Supposez que vous souhaitez récupérer une page composée d'un fichier HTML indiquant deux objets de taille réduite stockés sur le même serveur. En négligeant les temps de transmission de ces objets, indiquez le délai nécessaire pour obtenir la page. Illustrez vos réponses avec un chronogramme.

2. Quelles optimisations prévues par HTTP 1.1 utilisent les serveurs web actuels pour réduire la latence des échanges ? En reprenant l'exemple précédent, illustrez vos réponses avec des chronogrammes.

3. Une autre possibilité pour réduire le temps de réponse est l'utilisation de la mise en mémoire cache. Décrivez où intervient ce mécanisme et pour quels types d'objets il est intéressant.

4.2 Protocole HTTP

4.2.1 Capture d'un trafic HTTP

Cette dernière capture a pour but de percevoir la nature du trafic HTTP. Sur la plateforme d'expérimentation, toujours sur la topologie 1 (postes client et serveur sur le même LAN), réalisez la capture de trafic SSH à l'aide du logiciel `wireshark/tshark` :

- A partir du poste PPTI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles.
- Vérifiez que le serveur HTTP (`apache2`) tourne sur `10.0.7.`**N**3 (fenêtre 3)
- Démarrez la capture en lançant l'analyseur sur sur l'interface `eth1` de l'hôte `10.0.7.`**N**2 (fenêtre 2)
- Utilisez un client HTTP sur `10.0.7.`**N**1 (fenêtre 1)
  - lancez le client, tapez : `firefox` ouvrez dans le navigateur la page `http://10.`**N**`.1.`**N**3
  - ou, en mode textuel : `wget -p --no-proxy http://10.`**N**`.1.`**N**3
- Observez la capture réalisée dans la fenêtre de `wireshark` (en la rapatriant préalablement si vous avez utilisé `tshark`)
- **Filtrez le trafic afin de ne conserver que celui relatif à HTTP** (filtre = `http`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

4.2.2 Analyse de l'échange HTTP

1. Observez-vous le mécanisme de récupération d'une page présenté précédemment ?

2. Quels paramètres sont négociés entre le client et le serveur ?

3. Quelles optimisations sont mises en œuvre pour accélérer le rapatriement des pages web ?

4. Pouvez-vous afficher la page web à partir de la capture ?

4.2.3 Création d'une nouvelle page web (*facultatif... à traiter si vous êtes en avance*)

Pour capturer un peu plus d'informations relatives aux échanges HTTP, vous pouvez rendre opérationnelle la page web du compte `etudiant` de la VM "serveur".

Pour cela, créez le répertoire `public_html` avec les droits adéquats dans le répertoire racine de l'utilisateur de ce compte. Utilisez les commandes UNIX suivantes : `cd ; mkdir ~/public_html ; chmod 755 ~/public_html`

Dans le répertoire `public_html` peuvent être mis des fichiers qui seront accessibles directement via le navigateur à l'URL suivante : `http://10.`**N**`.1.`**N**`3/~etudiant/`.

Créez une page web très simple composée de :

- 1 page HTML simpliste (voir sur le web comment taper les quelques lignes de code nécessaires)
- 3 petites images (à tranférer via SCP du poste PPTI vers le même répertoire).

Réalisez à nouveau la capture de la partie 4.2.1 en accédant cette fois à la page que vous venez de créer et répondez ensuite aux questions de la partie 4.2.2.

4.2.4 Sans la plateforme...

En cas de problème d'accès à la plateforme d'expérimentation ou si vous souhaitez réviser sur une autre machine, vous pouvez télécharger la trace `tme2-htt.dmp` (similaire à celle capturée précédemment) soit à partir du répertoire `/Infos/lmd/2022/master/ue/MU4IN001-2022oct`, soit sur la page web `http://www-npa.lip6.fr/~fourmaux/Traces/labV8.html`, puis l'analyser avec le logiciel `wireshark` (sans avoir besoin des droits d'administrateur).

5 Avant de quitter la salle

- Si vous avez enregistré des captures sur la VM "sonde", n'oubliez pas de les rapatrier sur votre compte utilisateur de la PPTI. Tapez la commande suivante dans un terminal local du PC de la PPTI : `scp etudiant@10.0.7.N2:<trace> <dest>`
- Avant de terminer vos connexions sur les équipements de la plateforme, supprimez vos fichiers et modifications effectuées afin de retrouver l'état initial.