

TP2

1 Protocole ARP

- Identifiez les adresses des machines voisines avec lesquelles des échanges récents ont eu lieu (table ARP, disponible par la commande `/usr/sbin/arp`, on peut utiliser l'option `-n` pour avoir les adresses IP plutôt que les noms de machines). Il doit y avoir au moins l'adresse du routeur (en `.254`, on verra dans la section routage ci-dessous).
- Vérifiez que lorsque vous émettez avec ping une requête ICMP echo vers une machine qui ne figure pas encore dans votre table ARP, cette machine apparaît dans la table ARP de votre machine et la vôtre, dans la table ARP de l'autre.
- Essayez la commande `ip neigh ls`; c'est la même chose en version plus moderne, et contient notamment aussi les voisins en IPv6.
- Essayez de faire un ping vers `8.8.8.8` et vers `10.0.252.4`. Pourquoi leurs adresses n'apparaissent pas dans la table ARP ?

2 Résolution de noms (DNS)

Mais au fait, comment la traduction `yahoo.com` → `98.139.180.149` fonctionne ? À l'aide d'un serveur DNS.

- Lisez le fichier `/etc/resolv.conf`, pourquoi y a-t-il plusieurs adresses IP ?
La ligne `search` permet d'éviter d'avoir à taper le nom de machine en entier. Essayez de taper `http://www/` tout court dans un navigateur web et observez comment cela est complété pour confirmer.
- Pour effectuer explicitement une résolution de nom, utilisez la commande `host` (ou éventuellement `nslookup`).
- Essayez de résoudre `yahoo.com`. Pourquoi y a-t-il plusieurs adresses IP ?
Observez également que cela retourne à la fois des adresses IPv4 et des adresses IPv6. Réessayez plusieurs fois. Il peut arriver que le résultat soit différent, pourquoi ?
- Parfois il est aussi utile d'ajouter des noms de machine à la main, lisez le fichier `/etc/hosts` (et le manuel).

3 Services au CREMI : LDAP & NFS

Quelle que soit la machine que vous utilisez au CREMI, elle vous connaît et retrouve vos fichiers. Comment cela se passe-t-il ?

- Pour l'identification et l'authentification, c'est le protocole LDAP qui est utilisé, la configuration est lisible dans le fichier `/etc/ldap.conf` (pas la peine de tout lire, ce qui nous intéresse est au tout début), pourquoi y a-t-il plusieurs serveurs (host)? Observez les adresses IP de ces serveurs.
Trouvez le numéro de port de ce service, vérifiez dans `/etc/services`, retrouvez-y également les ports `http`, `ssh`, `x11`, etc.
- Vos fichiers sont stockés sur un serveur NFS, utilisez la commande `df ~` pour repérer le nom du serveur, le chemin sur le serveur, et le chemin où cela apparaît sur votre machine.

4 Analyse de Trames avec Wireshark

4.1 Préambule

Voici les informations utiles sur la machine qui a servi à capturer ces traces réseaux. Ces informations sont obtenues avec les commandes Linux suivantes :

```
root@mydebian:~$ ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING, MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0    broadcast 10.0.2.255
    ether 08:00:27:0b:03:37 txqueuelen 1000 (Ethernet)
    RX packets 60 bytes 8043 (7.8 KiB)
    RX errors 0 dropped 0 overruns 0    frame 0
    TX packets 63 bytes 10204 (9.9 KiB)
    TX errors 0 dropped 0 overruns 0    carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0    frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0    carrier 0 collisions 0

root@mydebian:~$ route -n

Kernel IP routing table
Destination    Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0        10.0.2.2        0.0.0.0         UG    0      0      0 eth0
10.0.2.0       0.0.0.0         255.255.255.0   U    0      0      0 eth0

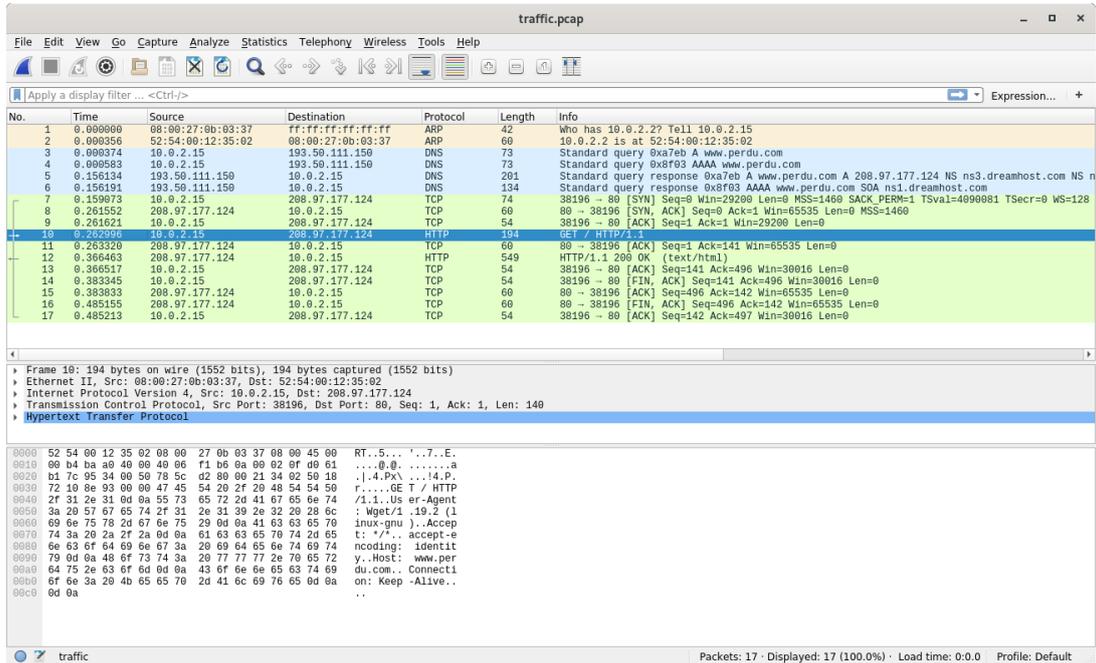
root@mydebian:~$ cat /etc/resolv.conf

nameserver 10.0.2.3
```

Vous devez reconnaître l'adresse IP de la machine utilisée (et son masque de réseau), l'adresse IP de la passerelle vers Internet, ainsi que l'adresse du serveur DNS local...

4.2 Prise en main de Wireshark

A l'aide de l'outil Wireshark, ouvrez le fichier [ping.pcap](#).



On voit apparaître dans la partie du haut une ligne pour chaque trame Ethernet que la carte réseau de la machine a traitée (en émission comme en réception). Ici on voit donc une séquence de trames avec un petit résumé qui indique la source, la destination, le protocole et quelques infos supplémentaires.

Lorsqu'une trame est sélectionnée, le contenu brut (i.e. octet par octet) apparaît dans la partie du bas, et une version décodée apparaît dans la partie du milieu. Il est alors possible d'inspecter cette trame en profondeur, couche par couche, en affichant à chaque niveau tous les détails sur le protocole utilisé. Cliquez sur \triangleright pour explorer un niveau.

4.3 Ping

La commande *ping* disponible sur toutes les plateformes permet de tester rapidement si une machine est joignable sur le réseau Internet à partir de son nom ou de son IP... Par exemple :

```
$ ping -4 -n www.google.com
```

```
PING www.google.com (172.217.19.132) 56(84) bytes of data.
64 bytes from 172.217.19.132: icmp_seq=1 ttl=63 time=15.4 ms
64 bytes from 172.217.19.132: icmp_seq=2 ttl=63 time=15.8 ms
64 bytes from 172.217.19.132: icmp_seq=3 ttl=63 time=15.7 ms
64 bytes from 172.217.19.132: icmp_seq=4 ttl=63 time=15.8 ms
64 bytes from 172.217.19.132: icmp_seq=5 ttl=63 time=15.8 ms
64 bytes from 172.217.19.132: icmp_seq=6 ttl=63 time=15.7 ms
64 bytes from 172.217.19.132: icmp_seq=7 ttl=63 time=15.7 ms
```

```
--- www.google.com ping statistics ---
```

```
7 packets transmitted, 7 received, 0% packet loss, time 26ms
rtt min/avg/max/mdev = 15.442/15.722/15.847/0.207 ms
```

Par défaut, la commande *ping* envoie une requête echo-request (protocole ICMP) chaque seconde et affiche en retour la réponse echo-reply avec quelques statistiques... Notamment, le temps affiché correspond au temps d'aller-retour du message (RTT = Round-Time Trip). On considère souvent que la Latence représente la moitié du RTT.

La trace enregistrée se divise en trois étapes principales identifiées avec des couleurs différentes dans Wireshark :

1. trames 1-2 (ARP) : on cherche à découvrir l'adresse Ethernet de la passerelle, afin de connaître la porte de sortie du réseau local vers Internet ;
2. trames 3-4 (DNS) : on cherche à trouver l'adresse IP de la machine cible (www.google.com) ;
3. trames 5-18 (ICMP) : 7 pings et leurs réponses...

Répondez aux questions suivantes concernant la trace [ping.pcap](#) en vous aidant de Wireshark.

- A quelle adresse Ethernet est destinée la requête ARP (trame 1) émise par la machine cliente ? Il s'agit en fait de l'adresse de diffusion (broadcast). Elle ne correspond à aucune machine particulière ! A votre avis pourquoi doit on procéder ainsi ?
- La requête ARP WHO HAS (trame 1) cherche à trouver l'adresse Ethernet de la machine 10.0.2.2. Pourquoi cette machine et non pas la machine cible www.google.com ? Vérifiez l'adresse Ethernet destination utilisée pour envoyer la trame 3.
- Quel est le protocole de transport utilisé pour les échanges DNS (trames 3-4) ? Observez en détail la réponse DNS (section Answers) et découvrez ainsi l'adresse IP de la machine www.google.com retourné par le serveur DNS.
- Observez la première requête / réponse ICMP (trames 5-6) et observez la valeur du champs type dans l'en-tête ICMP...

4.4 Une page Web : je suis perdu !

Que se passe-t-il quand je consulte une page web (par exemple, <http://www.perdu.com>) sur Internet avec mon navigateur préféré ?

La trace enregistrée effectue les mêmes étapes préliminaires que dans la trace précédente : requêtes ARP et DNS... Concentrons nous ici sur la conversation TCP/IP (trames 7-17). On demande la page d'accueil "/" ou "/index.html" du serveur web (www.perdu.com) en utilisant le protocole HTTP au dessus de TCP, ce qui implique plusieurs étapes intermédiaires :

1. la connexion TCP en trois temps (trames 7-9) ;
2. la requête HTTP ainsi que la réponse incluant le code HTML (trames 10-13) ;
3. la phase de déconnexion (trames 14-17).

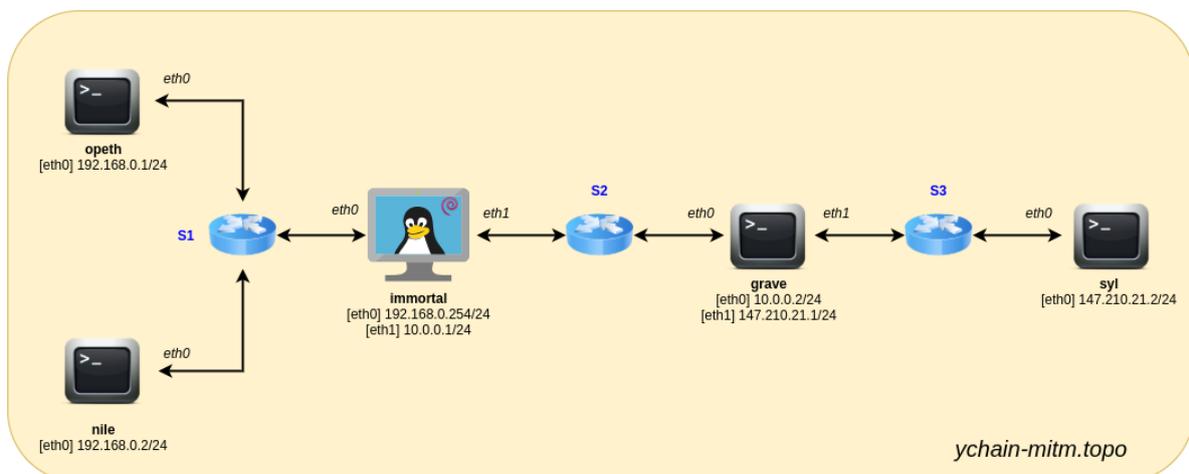
Répondez aux questions suivantes concernant la trace [http.pcap](#) en vous aidant de Wireshark.

- Considérons la première trame TCP qui ouvre la connexion (trame 7). Trouvez dans l'en-tête TCP le port source et le port de destination. Ce dernier est standard pour tous les serveur web (80). A quoi correspond le flag SYN dans cette en-tête ?
- Identifiez dans la conversation TCP les trames correspondant à la requête HTTP et à la réponse HTTP...
- Dans l'en-tête de la requête HTTP, on observe sur la première ligne qu'il s'agit de la requête `GET / HTTP/1.1`. Identifiez le rôle des champs suivants : User-Agent, Host, Connection.
- La réponse HTTP commence par la ligne suivante "HTTP/1.1 200 OK" qui indique que tout s'est bien passé (code 200). Vous devez déjà connaître le fameux code d'erreur 404 (cf. [liste des codes HTTP](#)).
- Observez maintenant les différents champs dans la réponse HTTP et en déduire le logiciel serveur, la longueur et le type de contenu dans cette réponse.
- Immédiatement après l'en-tête HTTP, vous pouvez identifier le code HTML de la page web : `<html>...</html>`.
- Trames 7-16 : Pour lire plus facilement la conversation TCP, vous pouvez faire un "clic droit" sur un des paquets TCP et sélectionner *Follow → TCP Stream* dans le menu déroulant. Notez qu'il est possible de reconstruire précisément le fil de la conversation grâce aux numéros de séquence (en octets) qui se trouve dans l'en-tête TCP.

5 Capture de Trames avec Wireshark

Notez qu'il faut absolument avoir les droits *root* pour pouvoir capturer tout le trafic réseau sur une interface réseau, comme *eth0*. Pour ce faire, nous allons travailler dans le réseau virtuel *QemuNet* ci-dessous. Ce réseau se compose de plusieurs machines Linux/Debian, déjà configurés, dont la machine *immortal* dispose d'une session graphique avec l'outil Wireshark.

Pour vous connecter aux machines, il faut utiliser le login *root* : pas de mot de passe, sauf pour les sessions graphiques où le mot de passe est *root*. En outre, vous disposez d'un compte utilisateur (login : *toto* et password : *toto*) sur toutes les machines pour faire des tests. Par ailleurs, chaque machine dispose de différents services : *www*, *ssh*, ...



Lancez la session QemuNet de la façon suivante au CREMI :

```
/net/ens/qemunet/qemunet.sh -s /net/ens/qemunet/demo/ychain-mitm.tgz
```

- Pour commencer, connectez vous en tant que *root* sur *opeth* (ou *nile* qui joue un rôle identique) et utilisez la commande *nmap* pour découvrir les services ouverts sur la machine *syl* à l'autre bout du réseau.
- Connectez vous maintenant en tant que *root* sur *immortal* (session graphique) et lancez une capture avec l'outil Wireshark sur l'interface *eth0* uniquement.
- Lancez *tcpdump* sur la machine *grave* (en mode texte) afin d'afficher le trafic qui circule entre *opeth* et *syl*.
- Depuis la machine *opeth*, testez (en ligne de commande) les différents services de *syl*. Commencez par *ping*. Vérifiez que vous comprenez toutes les étapes de la trame capturée dans Wireshark.
- Toujours sur *opeth*, utilisez la commande *telnet* pour vous connecter à la machine *syl* en utilisant le login *toto* (mot de passe : *toto*) : `telnet syl`. Pour vous déconnecter, tapez la commande *exit*. Cherchez le mot de passe *toto* dans la trace capturée !
- Bonus : Faire de même avec la commande *ssh* : `ssh toto@syl`. SSH (ou *Secure SHell*) est la version moderne et sécurisée de Telnet... Pouvez-vous voir le mot de passe ? Sur quel protocole au-dessus de TCP repose cette commande ?
- Récupérez la page d'accueil du serveur web de *syl* avec la commande en ligne *curl* ou *wget* : `wget http://syl`. D'un point de vue réseau, cette commande effectue le même travail qu'un navigateur web comme Chrome ou Firefox... Observez la réponse HTTP, dont vous pouvez analyser l'en-tête et le corps : une simple page HTML en texte.