M1 RES - Travaux sur machine encadrés 1/10

Introduction à l'analyse de trames

1 Analyse manuelle d'une trame capturée sur un réseau Ethernet

Etudions une trame observée sur un réseau local Ethernet. Cette trace est obtenue à l'aide d'un analyseur multiprotocolaire (un *sniffer*, tel l'outil `tcpdump` utilisé sur une machine connectée à ce réseau). Les traces sont habituellement présentées selon trois colonnes :

❶	❷	❸
0000	00 01 02 a5 fb 3a 00 01 02 a5 fc 8d 08 00 45 00	...?û:.. .?ü...E.
0010	00 3c ec 26 40 00 40 06 cc cd 0a 21 b6 b6 84 e3	.<ì&@.@. ÌÍ.!ll.ã
0020	.. ..	.. ..

- ❶ indique, avec 4 chiffres hexadécimaux, le **rang** du premier octet de la ligne courante dans la trame ;
- ❷ affiche la **valeur hexadécimale** de 16 octets capturés (un octet est codé sur deux chiffres hexadécimaux) ;
- ❸ représente les caractères ASCII correspondants aux 16 octets de la seconde colonne (la correspondance n'est significative que lorsque du texte lisible se trouve encodé dans ces octets).

Les trames Ethernet présentées ne comportent ni préambule, ni CRC.
Veillez à respecter les conventions de représentation :
- **Adresses Ethernet** : hexadécimale double pointée (ex : `00:50:04:ef:6b:18`)
- **Type Ethernet** : hexadécimale (ex : `0x0806`)
- **Adresses IP** : décimale pointée (ex : `10.1.1.3`)
- **Numéro de protocole** et **numéro de port** : décimale (ex : `17`)

Trace à analyser :

```
0000  00 01 02 a5 fb 3a 00 01  02 a5 fc 8d 08 00 45 00   ...?û:.. .?ü...E.

0010  00 3c ec 26 40 00 40 06  cc cd 0a 21 b6 b6 84 e3   .<ì&@.@. ÌÍ.!¶¶.ã

0020  3c 0d 0e b5 00 50 a9 55  92 64 00 00 00 00 a0 02   <..?.P©U .d....?.

0030  3e bc a3 74 00 00 02 04  05 b4 04 02 08 0a 08 39   >¼£t.... .´.....9

0040  91 16 00 00 00 00 01 03  03 00                     ........ ..
```

Les structures des principaux protocoles rencontrés sont rappelées dans la suite (page 3).

1. Représentez la structure de la trame en dessinant directement les délimitations sur la trace à analyser.
2. Quelles informations de niveau liaison observez vous ?
3. Représentez la structure du paquet directement sur la trace à analyser.
4. Le paquet contient-il des options ? Justifiez ?
5. Quelles sont la source et le destinataire du paquet ?
6. Représentez la structure des données transportées par le paquet directement sur la trace.
7. Quel est le protocole de transport utilisé ? Quels sont les ports utilisés ? quelles sont leurs significations ?
8. *Il n'y a pas de documentation correspondant à la couche application à la fin du document,* malgré cela, pouvez vous observer des informations associées à ce niveau dans la trace ?

2 Utilisation de l'analyseur de trame `ethereal`

Pour analyser les informations circulant sur les réseaux, les administrateurs disposent donc de *sniffer*. Cet outil se présente sous la forme d'un équipement pouvant se connecter directement sur le réseau ou d'un logiciel installé sur un ordinateurs reliés au réseau à analyser.

Lorsque le réseau à étudier est de type à **médium partagé** — Ethernet par exemple — l'interface de toute machine connectée "voit" potentiellement tout le trafic échangé sur le réseau local. Pour ne pas juste "voir", mais "regarder" explicitement le trafic afin de récupérer les trames observées (y compris celles qui ne lui sont pas destinées) pour les analyser, un mode de "promiscuité" est habituellement disponible sur l'interface réseau. Ce mode ne perturbe ni le trafic du réseau, ni celui de la machine support. Cela permet ainsi d'ajouter la fonction "*sniffer*" à cette dernière avec un logiciel adéquat.

Le logiciel `ethereal`[1] est un analyseur de protocoles (*sniffer*). Celui-ci peut utiliser directement l'interface de votre machine pour réaliser la capture de toutes les informations circulant sur le réseau local sur lequel vous êtes connecté[2]. Pour des raisons évidentes de sécurité, nous vous fournirons des captures déjà réalisées. Vous utiliserez alors la fonction principale de l'outil : l'analyse multiprotocolaire.

2.1 Introduction à `ethereal`

Sur une machine de la salle de T.M.E. sous le système **GNU/Linux**, accédez à votre compte utilisateur. Dans un terminal, exécutez la commande `ethereal`[3] (et avec la commande `man ethereal` obtenez les informations de base). Initialement l'écran est vide car aucune capture n'a été réalisée ou chargée. Cliquez sur le menu **File** et sélectionnez **Open**. Une fenêtre de sélection de fichier "*Ethereal : Open Capture Files*" apparaît. Sélectionnez le fichier :

`/Infos/lmd/2004/master/ue/res-2004oct/tme1.dmp.gz`

Ne pas spécifier de filtres dans le champ *Filter* (nous y reviendrons plus loin). Désactivez : ❑*Enable MAC name resolution*, ❑*Enable network name resolution* et ❑*Enable transport name resolution*. Cliquez finalement sur **Open** : La trace d'une capture précédemment réalisée est chargée et vous aller pouvoir l'analyser.

1. Décrivez le contenu des trois fenêtres proposées par `ethereal`.
2. Dans quels formats sont représentées les données de la troisième fenêtre ?
3. Quels sont les différents protocoles que vous pouvez observer dans la capture affichée ?

2.2 Filtres d'affichage `ethereal`

1. A l'aide du `man`, décrivez la syntaxe utilisée.
2. Combien de protocoles est capable d'analyser la version d'`ethereal` que vous utilisez
3. Décrivez un filtre qui ne sélectionne que les trames ARP de/vers l'interface avec l'adresse MAC `00:10:a4:86:2d:0b`. Pour vous aider, le menu **Analyse** propose **Display Filters...** qui affiche une fenêtre d'édition de filtre. Le bouton **+Expression** permet une aide à la création de l'expression correspondante. Attention, après avoir entré le nom du filtre et l'expression, il faut ensuite cliquer sur **Nouveau** pour valider la création du filtre. Appliquer ce filtre d'affichage pour n'observer que les trames correspondantes.
4. Supprimer le filtre précédent et coloriez en vert les trames ARP.

2.3 Analyse d'un trafic HTTP

Dans la continuité de la trame étudiée manuellement dans la section précédente :

1. Affichez en rouge les trames du trafic HTTP, puis sélectionnez seulement celle transportant des données HTTP.
2. Décrivez ce que vous observez, et si il y a plusieurs connexions, quelle est leur relation ?
3. Peut-on visualiser le contenu applicatif d'une connexion TCP ?

[1] `ethereal` est un logiciel libre. Il est disponible sur un grand nombre de plates-formes matérielles et systèmes d'exploitation (outre les machines à architecture **i368** avec système **GNU/Linux** que vous utilisez actuellement). Vous pouvez le télé-charger sur `www.ethereal.org`.

[2] L'interface Ethernet doit être configurée par `ethereal` dans le mode "*promiscuous*" afin d'accéder à tout le trafic diffusé sur le segment où votre machine est connectée. Pour changer de mode, l'application doit être exécutée avec les privilèges de l'administrateur.

[3] Vous pouvez aussi utiliser l'icône correspondant à `ethereal` dans votre environnement graphique.

Structure de la trame Ethernet

```
.......+-48bits--+-48bits--+16b-+- - - - -+......
.(Pré.)| adresse | adresse |type| données |(CRC).
.      | dest.   | source  |    |         |     .
.......+---------+---------+----+- - - - -+......

Quelques types : 0x0200 = XEROX PUP
                 0x0800 = DoD Internet
                 0x0806 = ARP
                 0x8035 = RARP
```

Structure ARP

```
+16b-+-16b-+8b+8b+16b+--lgHW--+lgP-+--lgHW--+lgP-+
|type|type |lg|lg|Op |Emetteur|Emt.|Récept. |Rcpt|
|HW  |Proto|HW|P |   |adr. HW |adrP|adr. HW |adrP|
+----+-----+--+--+---+--------+----+--------+----+

Quelques types : 0x0001 = Ethernet
                 0x0800 = DoD Internet
Opérations : 0x0001 = Requête
             0x0002 = Réponse
```

Structure du paquet IP

```
<-----------------32bits----------------------->
<-4b->      <--8bits--->  <--------16bits-------->
+-----+-----+-----------+-----------------------+
| Ver | IHL | TOS       | Longueur totale (octet)
+-----+-----+-----------+--+--------------------+
| Identificateur        |Fl| FO                 |
+-----------+-----------+--+--------------------+
| TTL       | Protocole | Somme de ctrl (entête)|
+-----------+-----------+-----------------------+
| Adresse Source                                |
+-----------------------------------------------+
| Adresse Destination                           |
+-----------------------------------------------+
...     Options
+-----------------------------------------------+
...     Données
+-----------------------------------------------+

Ver = Version d'IP
IHL = Longueur de l'en-tête IP (en mots de 32 bits)
TOS = Type de service (zero généralement)
Fl (3 premiers bits) = Bits pour la fragmentation
 * 1er = Reservé
 * 2me = Ne pas fragmenter
 * 3me = Fragment suivant existe
FO (13 bits suivants) = Décalage du fragment
TTL = Durée de vie restante

Quelques protocoles transportés :
        1 = ICMP                 8 = EGP
        2 = IGMP                11 = GLOUP
        4 = IP (encapsulation)  17 = UDP
        5 = Stream              36 = XTP
        6 = TCP                 46 = RSVP
```

Structure du paquet ICMP

```
<-----------------32bits----------------------->
+-----------+-----------+-----------------------+
|  Type     |  Code     | Somme de contrôle (msg)
+-----------+-----------+-----------------------+
|  Variable (généralement non utilisé)          |
+-----------------------------------------------+
...     Datagramme original + 8 octets
+-----------------------------------------------+

Quelques types ICMP :  0 = Demande d'écho
                       8 = Réponse d'echo
                      11 = Durée de vie écoulée
                      12 = Erreur de paramètre
```

Structure de segment TCP

```
<-----------------32bits----------------------->
<-4b->         <-6bits-><--------16bits-------->
+----------------------+-----------------------+
| Port Source          |  Port Destination     |
+----------------------+-----------------------+
| Numéro de Séquence                           |
+----------------------------------------------+
| Numéro d'Acquittement                        |
+-----+-------+---------+-----------------------+
| THL |       |  Flag   |  Taille Fenêtre       |
+-----+-------+---------+-----------------------+
| Somme de ctrl (message)  Pointeur d'Urgence  |
+----------------------+-----------------------+
...      Options
+----------------------------------------------+
...      Données
+----------------------------------------------+

THL = Longueur de l'entête TCP sur 4 bits (*32bits)
Flags = indicateur codé sur 6 bits gauche à droite
 * 1er = Données urgentes
 * 2me = Acquittement (ACK)
 * 3me = Données immédiates (Push)
 * 4me = Réinitialisation (Reset)
 * 5me = Synchronisation (SYN)
 * 6me = Fin
Options = suites d'option codées sur
 * 1 octet à 00 = Fin des options
 * 1 octet à 01 = NOP (pas d'opération)
 * plusieurs octets de type TLV
   T = un octet de type:
     2 Négociation de la taille max. du segment
     3 Adaptation de la taille de la fenêtre
     4 Autorisation des acquittements sélectifs
     8 Estampilles temporelles
   L = un octet pour la taille totale de l'option
   V = valeur de l'option (sur L-2 octets)
```

Structure de segment UDP

```
<-----------------32bits----------------------->
+----------------------+-----------------------+
| Port Source          | Port Destination      |
+----------------------+-----------------------+
| Longueur UDP         | Somme de ctrl (message)
+----------------------+-----------------------+
...      Données
+----------------------------------------------+
```

Services associés aux ports

ftp-data	20/tcp		
ftp	21/tcp		
ssh	22/tcp	ssh	22/udp
telnet	23/tcp		
smtp	25/tcp		
domain	53/tcp	domain	53/udp
		tftp	69/udp
finger	79/tcp		
www	80/tcp	www	80/udp
kerberos	88/tcp	kerberos	88/udp
pop-3	110/tcp	pop-3	110/udp
		snmp	161/udp
		snmp-trap	162/udp