ANALYSE DE TRAMES – OUTIL WIRESHARK

Projet en Binôme

Compte rendu : Groupe 1.1, 1.2, 2.1 => envoyer a malo.manini@b-com.com

Groupe 2.2, 3.1, 3.2 => envoyer a cedric.gueguen@irisa.fr

Indiquer le nom du groupe ainsi que vos nom dans le sujet suivant l'exemple suivant (groupe 1.1) :

Sujet du mail => [Compte rendu RES/ G1.1/Nom1 Prénom1 + Nom2 Prénom2]

A rendre à la fin des 2 séances

1. ANALYSE MANUELLE DE TRAMES

L'objectif de cette section est d'étudier « manuellement » une trame observée sur un réseau local Ethernet. L'obtention d'une telle trame peut se faire à l'aide d'un analyseur de réseau multiprotocolaire (un *sniffer*).

On considère la trame Ethernet ci-dessous, donnée sans préambule, ni CRC :

```
00 01 02 a5 fb 3a 00 01 02 a5 fc 8d 08 00 45 00
00 3c ec 26 40 00 40 06 cc cd 0a 21 b6 b6 84 e3
3c 0d 0e b5 00 50 a9 55 92 64 00 00 00 00 a0 02
3e bc a3 74 00 00 02 04 05 b4 04 02 08 0a 08 39
91 16 00 00 00 00 01 03 03 00
```

Analysez cette trame en vous aidant de l'annexe fournie à la fin du support.

1.1. Quelles informations de niveau liaison observez-vous ?

1.2. Quelles informations de niveau IP sont disponibles ?

1.3. Le paquet contient-il des options ? Justifiez.

1.4. Quelles sont la source et le destinataire du paquet ?

1.5. Quel est le protocole de transport utilisé ?

1.6. Identifiez (sans les interpréter) les différents champs contenus dans les données transportées dans le paquet IP.

2. UTILISATION DE L'ANALYSEUR DE TRAME WIRESHARK

Pour analyser les informations circulant sur les réseaux, les administrateurs disposent donc de *sniffers*. Ces outils se présentent sous la forme d'équipements pouvant se connecter directement sur le réseau ou de logiciels installés sur un ordinateur relié au réseau à analyser.

Lorsque le réseau à étudier est 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 toutes les trames qui circulent (y compris celles qui ne lui sont pas destinées), 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, ce qui permet d'ajouter la fonction « sniffer » à cette dernière avec un logiciel adéquat.

Le logiciel WireShark[1] est un analyseur de protocoles (sniffer). Il 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]. Dans un premier temps, nous vous fournirons des captures déjà réalisées. Vous utiliserez alors la fonction principale de l'outil : l'analyse multiprotocolaire.

2.1 Introduction à WireShark

Téléchargez les fichiers `capture_http.pcap` et `capture_ping.pcap` disponible à l'adresse suivante : http://yhaker.perso.sfr.fr/capture.zip

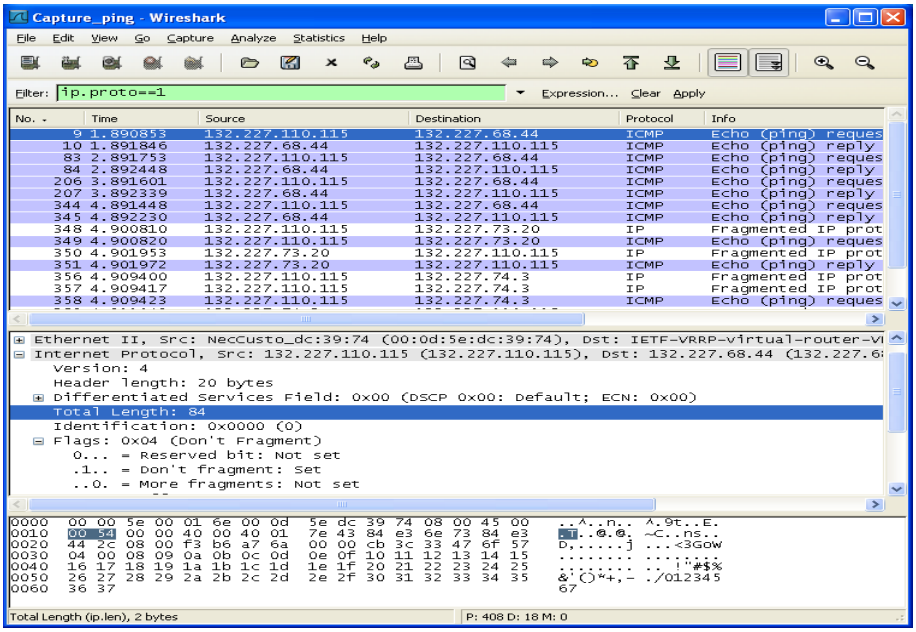
Dans un terminal, exécutez la commande `wireshark` (et avec la commande `man wireshark` obtenez les informations de base). **Vous devez lancer l'application en mode « sans privilège ».** Initialement l'écran est vide car aucune capture n'a été réalisée ou chargée. Cliquez sur le menu `File` et sélectionnez `Open`.

- Sélectionnez le fichier `capture_http.tcap` (il s'agit d'une capture précédemment réalisée que vous allez pouvoir analyser).
- Ne pas spécifier de filtres dans le champ `Filter` (nous y reviendrons plus loin).
- Dans le menu `Preferences`, désactivez les options `Enable MAC name resolution`, `Enable network name resolution` et `Enable transport name resolution`.
- Cliquez finalement sur Ouvrir.

L'interface de Wireshark est constituée de trois fenêtres principales (voir figure ci-dessous) :

[1] WireShark 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 télécharger sur www.wireshark.org.

[2] L'interface Ethernet doit être configurée par WireShark 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.



La fenêtre du haut liste les trames capturées et résume leurs caractéristiques. En cliquant sur l'une des trames dans cette fenêtre, le contenu de la trame apparaît en hexadécimal dans la fenêtre du bas, et le contenu de chaque champ est décrit dans la fenêtre du milieu. Ces trois fenêtres sont coordonnées (un clic dans l'une des fenêtres a un impact sur les deux autres).

La fenêtre du milieu décrit précisément chaque champ de la trame. Ces champs, ainsi que diverses informations fournies par le logiciel, y sont présentés dans une structure arborescente.

La fenêtre du bas est constituée essentiellement de la trame en hexadécimal. Les champs sélectionnés dans l'arbre de la fenêtre du milieu sont surlignés dans la trame. Les données sont présentées selon trois colonnes :

- La première colonne indique, sur 4 caractères hexadécimaux, le rang du premier octet de la ligne courante dans la trame ;
- La deuxième colonne affiche la valeur hexadécimale de 16 octets ;
- La troisième colonne représente les caractères ASCII correspondant aux 16 octets de la seconde colonne (la correspondance n'est significative que lorsque du texte lisible se trouve encodé dans ces octets).

2.1.1. Observez le contenu des trois fenêtres proposées par WireShark.

2.1.2. Dans quels formats sont représentées les données de la fenêtre du bas ?

2.1.3. Quels sont les différents protocoles que vous pouvez observer dans la capture affichée ?

2.1.4. Combien de protocoles est capable d'analyser la version de WireShark que vous utilisez ?

2.2 Filtres d'affichage WireShark

En plus des trois fenêtres décrites précédemment, il existe un champ fort intéressant, situé juste au dessus de la première fenêtre, et qui est le champ `Filter`. Il sert à sélectionner certaines trames en fonction de champs particuliers (par exemple les adresses, les protocoles, etc.). Après écriture du filtre, il est activé en cliquant sur le bouton `Apply`. Ainsi, seules les trames autorisées par le filtre sont visibles dans les fenêtres. Un clic sur le bouton `Clear` désactive le filtre d'affichage en cours d'utilisation.

La syntaxe d'un filtre simple est la suivante :

```
nom_du_champs relation valeur
```

Par exemple, si vous ne souhaitez voir apparaître que les trames correspondant au protocole http, vous pouvez appliquer le filtre suivant :

```
tcp.port == 80
```

Pour créer des filtres plus complexes, vous pouvez combiner plusieurs filtres simples grâce aux opérateurs logiques `and` et `or`. Pour vous faciliter l'écriture d'un filtre, cliquez sur l'onglet `Expression` à droite de l'espace réservé à l'écriture du filtre. Vous aurez ainsi accès aux différents noms de champs utilisables dans les filtres ainsi qu'à des valeurs prédéfinies. Si vous êtes plus à l'aise, vous pouvez écrire directement le filtre dans l'espace réservé.

2.2.1. Décrivez et appliquez un filtre qui ne sélectionne que les trames ARP de/vers l'interface ayant l'adresse MAC `00:1b:77:d2:d2:27`.

2.2.2. Supprimez le filtre précédent

3. ANALYSE D'UNE REQUETE HTTP

Nous allons maintenant étudier en détail l'échange contenu dans la trace `capture_http.pcap`.

3.1. Quels sont les protocoles présents ?

3.2. Quel est l'objectif de l'échange ARP initial ? Qui est le destinataire de la première trame ? Quelles sont les adresses des participants de cet échange ?

3.3. Quel est l'objectif des trames 3 à 6 ? Quel site est visé ? Quelle est la différence entre les trames 3 et 4, d'une part, et 5 et 6, d'autre part ?

3.4. Quel est le protocole de transport associé au protocole applicatif utilisé dans les trames 3 à 6 ? Quelles sont les fonctionnalités apportées par ce protocole de transport ?

3.5. Quel semble être le but de cet échange ?

3.6. Cette trace a été obtenue en tapant l'url http://ratp.fr dans un navigateur. D'après vous, pourquoi observe-t-on plusieurs requêtes (GET) ?

3.7. Quel est le protocole de transport utilisé à partir de la trame 7 ? Quelles sont les fonctionnalités apportées par ce protocole de transport ? Quelles différences pouvez-vous constater dans l'en-tête par rapport au protocole de transport utilisé dans les trames 3 à 6 ?

3.8. Sélectionnez toutes les trames relatives au protocole http (port 80). Quel filtre appliquez-vous ?

3.9. A quoi correspond la trame 45 ? Le fichier correspondant a-t-il été envoyé en une seule fois ?

3.10. Pourquoi y a-t-il une seconde requête DNS (trame 48) ? Est-elle liée au reste de l'échange ?

4. REALISATION DE CAPTURES A L'AIDE DE WIRESHARK

Comme nous avons pu le voir dans la première partie de ce TP, Wireshark permet d'analyser des captures préalablement enregistrées. Une autre de ses fonctionnalités est de pouvoir 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.

Dans cette partie du TP nous verrons comment créer de nouvelles captures. Elles auront pour but de mettre en évidence le principe de fonctionnement du cache du navigateur web ainsi que les protocoles de sécurité.

4.1. La première chose à faire est de sélectionner l'interface sur laquelle le logiciel doit capturer l'information. Dans l'interface d'accueil de Wireshark choisissez de capturer sur toutes les interfaces (any).

Maintenant que Wireshark est correctement paramétré, l'objectif est de générer du trafic à capturer.

4.3. Ouvrez maintenant un navigateur web (firefox par exemple).

   a. Accéder à l'adresse suivante : http://dechets.rennesmetropole.fr/

   b. Allez dans les options afin de vider le contenu du cache de votre navigateur. Cependant laissez la possibilité au cache de stocker des informations (activé par défaut).

4.4. Vous pouvez maintenant lancer la capture.

   a. Appliquez un filtre pour n'afficher que le protocole http

   b. Afin de générer du trafic, actualisez la page web.

4.5. A quoi correspondent les requêtes GET ?

4.6. Que retourne le serveur ?

Lancez maintenant une nouvelle simulation avec les mêmes paramètres. Actualisez la page web.

4.7. Observez-vous des différences par rapport à la capture précédente ? En au niveau des réponses du serveur.

4.8. Expliquez le phénomène en cause.

Nous allons maintenant effectuer une capture sur une page qui utilise le protocole https.

4.9. Lancez une nouvelle capture avec cette fois-ci un filtre sur l'adresse de destination,

ip.dst == 129.20.126.128

4.10. Accédez à l'adresse suivante : https://ent.univ-rennes1.fr

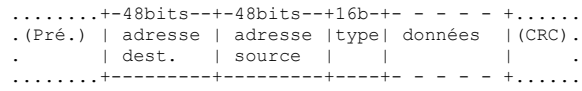
   a. Stoppez la capture au bout de quelques instants.

4.11. Quels sont les protocoles utilisés ? Détaillez leurs utilités.

4.12. Pourquoi ces protocoles diffèrent-ils de ceux utilisés dans les captures précédentes ?

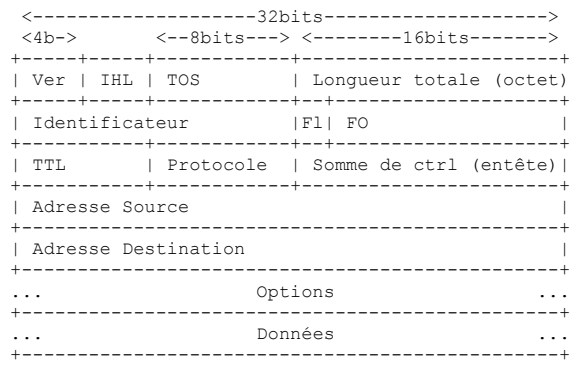
4.13. Quels sont les avantages que vous constatez à utiliser un protocole tel que https ?

Structure d'une trame Ethernet



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

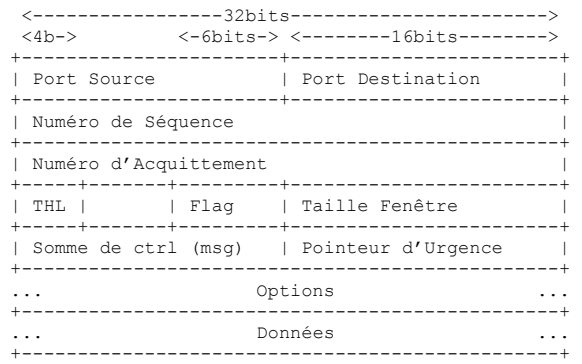
Structure d'un paquet IP



Ver = Version d'IP
IHL = Longueur de l'entête 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
 2 = IGMP
 4 = IP (encapsulation)
 5 = Stream
 6 = TCP
 8 = EGP
 11 = GLOUP
 17 = UDP
 36 = XTP
 46 = RSVP

Structure d'un segment TCP




```
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 = suite d'options 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)
```

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
rtraceroute 3765/tcp
```