

9.6

Annexes

NSI 1ÈRE - JB DUTHOIT

9.6.1 Utilisation de Wireshark

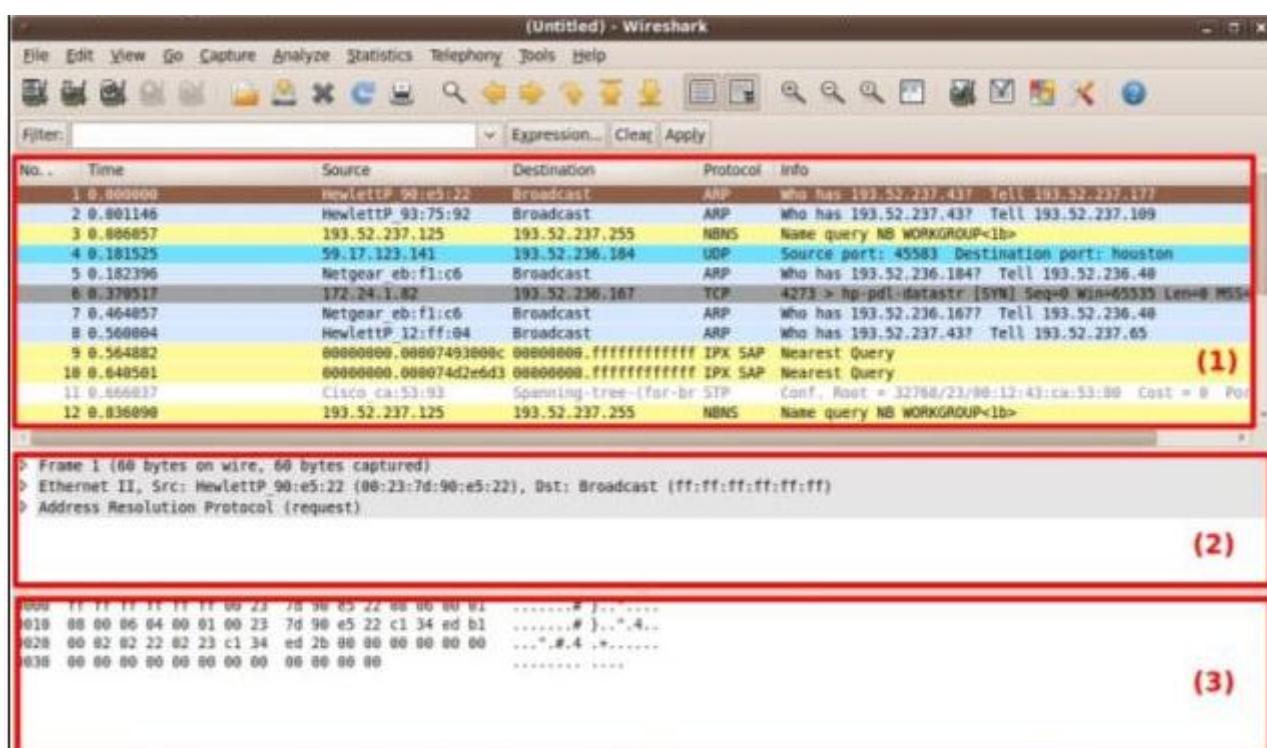
Ce tutoriel présente les principales fonctions de Wireshark nécessaires à une utilisation basique et se destine principalement à un public néophyte. Nous invitons le lecteur à se référer au manuel de l'utilisateur pour une utilisation avancée.

Capture des trames

La principale utilisation que nous ferons de Wireshark consistera en la capture de trames réseau en live. Les trois premiers boutons de la barre d'icônes permettent une telle capture, et sont des raccourcis aux éléments présentés dans le menu « capture »

L'interface de l'analyseur est découpée en trois zones :

- Zone supérieure, numérotée (1) sur la figure 3 : liste l'ensemble des paquets capturés
- Zone centrale, numérotée (2) : affiche le détail d'un paquet sélectionné dans la liste des paquets de la zone supérieure. Les informations présentées y sont de loin les plus pertinentes, puisqu'il est possible de visualiser aisément les différents en-têtes résultant de l'encapsulation d'un message.
- Zone inférieure, numérotée (3) : présente l'ensemble du paquet sous forme octale et ASCII. Ces octets contiennent les en-têtes des différentes couches de l'architecture TCP/IP ainsi que les données transmises par le processus à l'origine du message.



Analyse d'un paquet

Lorsqu'un paquet est sélectionné, la zone centrale permet de visualiser clairement les différentes couches d'encapsulation du paquet. Par exemple :

```

▶ Frame 5765 (65 bytes on wire, 65 bytes captured)
▶ Ethernet II, Src: Dell_b3:04:ee (00:1d:09:b3:04:ee), Dst: CompaqIn 41:3e:16 (00:1b:38:41:3e:16)
▶ Internet Protocol, Src: 193.52.236.243 (193.52.236.243), Dst: 193.52.236.247 (193.52.236.247)
▶ User Datagram Protocol, Src Port: 55056 (55056), Dst Port: terabase (4000)
▶ Data (23 bytes)
```

Les 5 entrées présentées correspondent à différentes encapsulations, ordonnées de haut en bas de la couche la plus basse à la couche la plus haute. Voici ce qu'on peut lire sur la figure précédente :

- Données sur le média de capture : Wire signifiant transmission filaire de la trame sous forme d'un signal électrique.
- Trame relative à la couche Interface : protocole Ethernet II avec les adresses MAC de la source et de l'expéditeur.
- Paquet relatif à la couche réseau : Internet Protocol, avec les adresses IP de la source et du destinataire.
- Paquet relatif à la couche transport : User Datagram Protocol avec les ports de la source et du destinataire.
- 5. Données de l'application qui regroupe généralement les couches session, présentation, application : 23 octets (= bytes) de données transmises

Détails d'un niveau d'encapsulation

Pour tout item correspondant à un niveau d'encapsulation, un clic sur le triangle en début de ligne permet de dérouler l'en-tête afin de voir l'ensemble des champs le composant. Certains champs peuvent également être déroulés. Sur l'exemple, nous avons étendu les entrées correspondant aux couches réseau, transport et application en cliquant sur les triangles correspondants :

```

> Frame 5765 (65 bytes on wire, 65 bytes captured) (1)
  > Ethernet II, Src: Dell_b3:04:ee (00:1d:09:b3:04:ee), Dst: CompaqIn_41:3e:16 (00:1b:38:41:3e:16)
    > Internet Protocol, Src: 193.52.236.243 (193.52.236.243), Dst: 193.52.236.247 (193.52.236.247)
      Version: 4 (2)
      Header length: 20 bytes (3)
      > Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        Total Length: 51 (4)
        Identification: 0x0000 (0)
        Flags: 0x04 (Don't Fragment)
          Fragment offset: 0
        Time to live: 64
        Protocol: UDP (0x11) (5)
        > Header checksum: 0xde65 [correct]
          Source: 193.52.236.243 (193.52.236.243) (6)
          Destination: 193.52.236.247 (193.52.236.247) (7)
        > User Datagram Protocol, Src Port: 55056 (55056), Dst Port: terabase (4000)
          Source port: 55056 (55056)
          Destination port: terabase (4000)
          Length: 31 (8)
          > Checksum: 0x5c85 [validation disabled]
        > Data (23 bytes) (9)
          Data: 74657374206427656E766F69206465206D657373616765
            [Length: 23]

0000  00 1b 38 41 3e 16 00 1d  09 b3 04 ee 08 00 45 00  ..8A>... ....E.
0010  00 33 00 40 00 40 11  de 65 c1 34 ec f3 c1 34  .3..@.. .e.4..4 (10)
0020  ec f7 d7 10 0f a0 00 1f  5c 85 74 65 73 74 20 64  ..... \.test d
0030  27 65 6e 76 6f 69 20 64  65 20 6d 65 73 73 61 67  'envoi d e messag (11)
0040  65


```

Nous pouvons y lire entre autres au niveau des numéros de référence que :

- (2) : Le paquet est de type IP v4.
- (5) : Le type de transport de ce paquet IP est un datagramme UDP.
- (6) : L'adresse IP de la machine source est 193.52.236.243
- (7) : L'adresse IP de la machine destination est 193.52.236.247

Nous pouvons également faire un point sur la taille des données et des entêtes à différents niveaux d'encapsulation :

- (9) : La taille des données envoyée par le processus est de 23 octets
- (8) : La taille totale du datagramme UDP est de 31 octets. Cette valeur est la somme entre la taille réelle des données (23 octets) et 8 octets d'entête du paquet (8 étant une valeur fixe pour un paquet UDP)
- (3) : La taille des entêtes du paquet IP est de 20 octets
- (4) : Le paquet IP contient une entête (20 octets) ainsi que le datagramme UDP (31 octets). Sa taille totale est de 51 octets.
- (1) : Si l'on ajoute les octets d'entête pour la couche Ethernet II, la taille totale de la trame est de 65 octets.

Notons ainsi que pour transférer 23 octets de données brutes, il nous a fallu transférer au total 65 octets (en fait il nous a même fallu transférer des octets supplémentaire avant la trame Ethernet).

Visualisation octale de la trame

La zone inférieure permet de visualiser la trame capturée sous forme octale. Un clic sur n'importe lequel des niveaux d'encapsulation permet de visualiser la portion d'octets correspondante dans la zone inférieure de l'analyseur.

Pour n'importe quel niveau, un clic sur une valeur de champs permet de visualiser la portion d'octets correspondant à cette valeur dans le paquet au niveau de la zone inférieure de l'analyseur. Réciproquement, un clic sur un octet quelconque affiche le champ correspondant dans la zone centrale.

Signalons enfin qu'il est possible de visualiser les données transmises dans ce datagramme UDP. En cliquant sur l'entrée « Data » de la zone centrale, les octets correspondants mais également leur codage ASCII sont mis en évidence dans la zone inférieure. En examinant ce codage ASCII, il est parfois très facile de décoder les messages. Si l'on examine les derniers caractères ASCII représentant le message - ref (11) sur Figure 5 - on devine aisément que le message envoyé était « test d'envoi de message ».

Mise en place de filtre

L'intégralité des paquets capturés est listée dans la zone supérieure de l'analyseur. Il est souvent utile de filtrer les paquets à capturer, afin de pouvoir visualiser correctement un certain type de paquets seulement. Wireshark permet de filtrer les paquets à capturer en fonction des informations des différentes couches d'encapsulation.

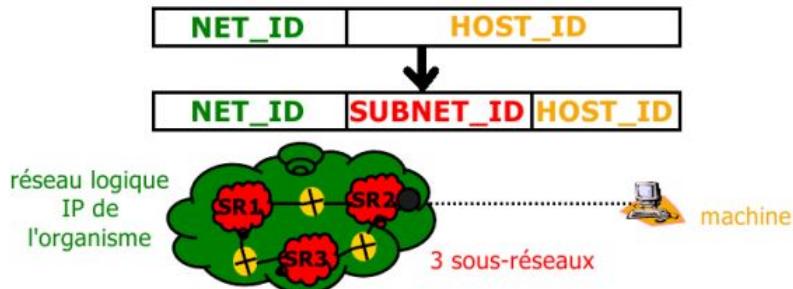
La mise en place d'un filtre s'effectue par le biais d'une règle de filtre à définir dans la zone « filtre » de l'analyseur. Une règle de filtre est constituée d'un ensemble de tests d'expressions impliquant des noms de champs et des valeurs. Un paquet n'est alors listé qu'à la condition qu'il satisfasse les conditions du filtre. L'ensemble des champs utilisables dans l'établissement des règles est listé dans la fenêtre pop-up accessible en cliquant sur le bouton « expression ». Les règles peuvent être élaborées en sélectionnant les champs à partir de cette fenêtre, ou en les écrivant directement dans la zone de filtre. Un ensemble de filtres prédefinis est accessible en cliquant sur le bouton « filter ». Cette liste de filtres peut être complétée d'entrées enregistrées. Une fois le filtre défini, il ne faut pas oublier de l'appliquer avec le bouton « Apply ».

Désignation	Filtre associé :
Segments TCP uniquement	tcp
Paquets relatifs à TCP uniquement	ip.proto == 0x06
Adresse ip 192.168.0.1	ip.addr == 192.168.0.1
Adresse ip 192.168.0.1 ou 192.168.1.5	ip.addr == 192.168.0.1 ip.addr == 192.168.1.5
Trafic HTTP uniquement	http
Segment TCP sauf sur port 80	tcp && !(tcp.port == 80)
Adresse Ethernet 00:FF:12:34:AE:FF	eth.addr == 00:FF:12:34:AE:FF
Trafic 192.168.0.1 vers 197.168.10.5	ip.src == 192.168.0.1 && ip.dst == 197.168.10.5
Trafic UDP entre ports 40 et 67	udp && udp.port >= 40 && udp.port <=67

9.6.2 Tableau récapitulatif OSI

7) Applications	Interface utilisateur (SMTP,HTTP,SSH..)
6) Présentation	Assure que les données sont présentées sous un format acceptable (ASCII, unicode..). Notamment que les informations de la couche Application soient lisibles par la couche application d'un autre système.
5) Session	Décide de commencer ou d'arrêter la connexion entre deux machines(socket)
4) Transport	Découpe le message en paquets numérotés et le reconstitue ensuite . Il vérifie la stabilité de la transmission(Protocole TCP,UDP)
3) Réseau	S'occupe du routage des paquets, du trajet entre source et destinataire. Protocole IP, ARP
2) Liaison	S'occupe d'identifier deux stations sur le même support physique. Adressage physique (MAC). S'occupe aussi de la livraison des trames.
1) Physique	Transmets de manière effective les bits (Ethernet, Wifi...)

9.6.3 Complément sur les masques des sous réseaux en IPv4



Supposons que l'on dispose d'une adresse qui permet d'adresser 254 machines avec le masque 255.255.255.0.

Il est possible de découper ce réseau en deux sous réseaux de 126 machines avec le masque 255.255.255.128 ($128 = 10000000$).

Considérons le réseau (classe C) ayant pour NetID 192.168.1.0 avec donc le masque 255.255.255.0. On veut découper ce réseau en deux sous-réseaux.

$255.255.255.0 <-> 11111111\ 11111111\ 11111111\ 00000000$

En ajoutant 2 bits, on obtient :

11111111 11111111 11111111 11000000, ce qui correspond à 255.255.255.192.

- Le NetID de chaque sous-réseau aura donc une adresse sur 26 bits :

- Les 24 premiers bits seront l'écriture binaire de 192.168.1.0.
- Les 2 bits suivants seront constitués du numéro des sous-réseaux 00,01,10,11.

- Parmi les 4 numéros de sous-reseaux, deux sont interdits (00 et 11) : bit de haut et bit de bas.
- il reste donc deux numéros utilisables :
 - * 192.168.1.00xxxxxx - Non utilisable
 - * 192.168.1.01xxxxxx - Utilisable
 - * 192.168.1.10xxxxxx - Utilisable
 - * 192.168.1.11xxxxxx - Non utilisable
- Les deux identifiants sous-réseaux sont donc 192.168.1.64 et 192.168.1.128
- Et la HostID aura donc :
 - Adresses IP de Premier sous-réseau : 192.168.1.64 (62 machines)
 - * 192.168.1.01000000 : Non utilisable
 - * 192.168.1.01000001 = 192.168.1.65
 - * 192.168.1.01000010 = 192.168.1.66
 - *etc...
 - * 192.168.1.01111110 = 192.168.1.126
 - * 192.168.1.01111111 : Non utilisable
 - * L'adressage du premier sous-réseau est de 192.168.1.56 à 192.168.1.126
 - Adresses IP de Deuxième sous-réseau : 192.168.1.128 (également 62 machines)
 - * 192.168.1.10000000 : Non utilisable
 - * 192.168.1.10000001 = 192.168.1.129
 - * 192.168.1.10000010 = 192.168.1.130
 - * ...etc...
 - * 192.168.1.10111110 = 192.168.1.190
 - * 192.168.1.10111111 : Non utilisable
 - * L'adressage du deuxième sous-réseau est de 192.168.1.129 à 192.168.1.190
- Pour obtenir l'adresse de diffusion dans chaque sous-réseau ; on met à 1 tous les bits de HostID.
 - * L'adresse de diffusion de premier sous-réseau est 192.168.1.01111111, soit 192.168.1.127.
 - * L'adresse de diffusion de deuxième sous-réseau est 192.168.1.10111111, soit 192.168.1.191