

Tutoriel d'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.

Récupération / Installation de Wireshark

Wireshark est un logiciel d'analyse réseau (sniffer) qui permettant de visualiser l'ensemble des données transitant sur la machine qui l'exécute, et d'obtenir des informations sur les protocoles applicatifs utilisés. Les octets sont capturés en utilisant la librairie réseau PCAP, puis regroupés en blocs d'informations et analysés par le logiciel.

La dernière version de Wireshark est disponible en téléchargement sur www.wireshark.org. De nombreuses distributions linux incluent Wireshark dans leur gestionnaire de paquet. Ainsi sous ubuntu on tapera simplement **sudo apt-get install wireshark**.

Lancement de Wireshark

Wireshark permet d'analyser un trafic enregistré dans un fichier annexe, mais également et surtout le trafic en direct sur des interfaces réseau. Cette seconde fonction nécessite de posséder les droits administrateurs, ou d'appartenir à un groupe possédant ces droits.

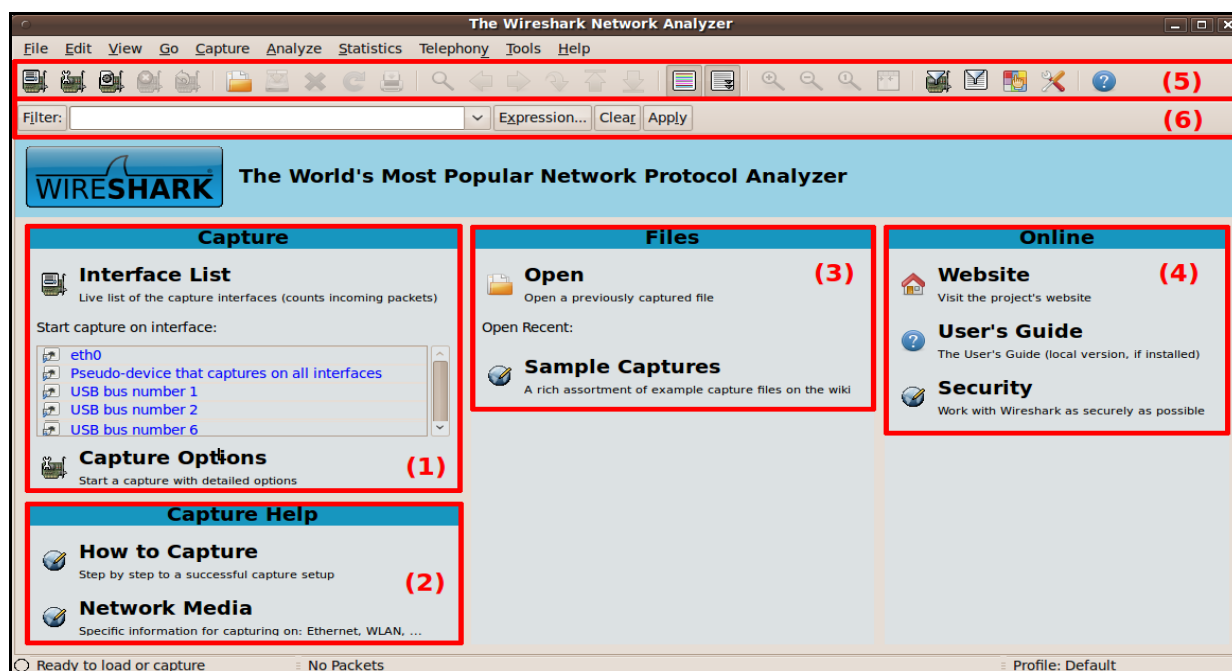


Figure 1: fenêtre d'ouverture de Wireshark

L'utilitaire s'ouvre sur l'interface présentée en Figure 1, découpé en quatre zones :

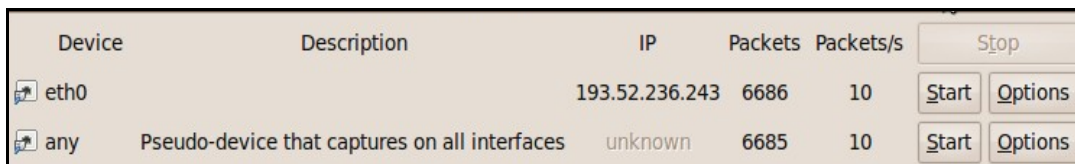
- (1) liste des interfaces et lancement rapide d'une capture
- (2) Aide sur la capture de paquets
- (3) Analyse d'une capture précédente enregistrée sur fichier
- (4) Aide online et manuel utilisateur

Capture des trames sur le réseau

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 »

Pour lancer une capture live, plusieurs méthodes s'offrent à nous, parmi lesquelles :

- Cliquer directement sur l'interface désirée listée dans la zone (1) de Figure 1, . On cliquera par exemple sur le bouton « Start » associé au lien **eth0** pour lancer la capture
- Cliquer sur le premier icône de la barre des icônes intitulé liste des interfaces de captures disponibles. Une fenêtre s'ouvre (Figure 2) , il nous suffit alors de cliquer sur le bouton « Start » de l'interface de notre choix pour lancer la capture sur cette interface.



Device	Description	IP	Packets	Packets/s	Stop
eth0		193.52.236.243	6686	10	Start Options
any	Pseudo-device that captures on all interfaces	unknown	6685	10	Start Options

Figure 2: Liste des interfaces, menu « capture / interfaces », ou premier icône

- Cliquer sur le 3ème icône de la barre des icônes en partant de la gauche pour lancer directement une capture sur l'ensemble des interfaces.

Une fois lancée, la capture peut être interrompue en cliquant que le 4ème bouton de la barre d'icônes (en partant de la gauche). Ce bouton n'est actif uniquement lors d'une capture. Lorsqu'une capture est active, le logiciel présente l'interface de l'analyseur (Figure 3). Cette interface reste ensuite visible lorsque la capture est arrêtée.

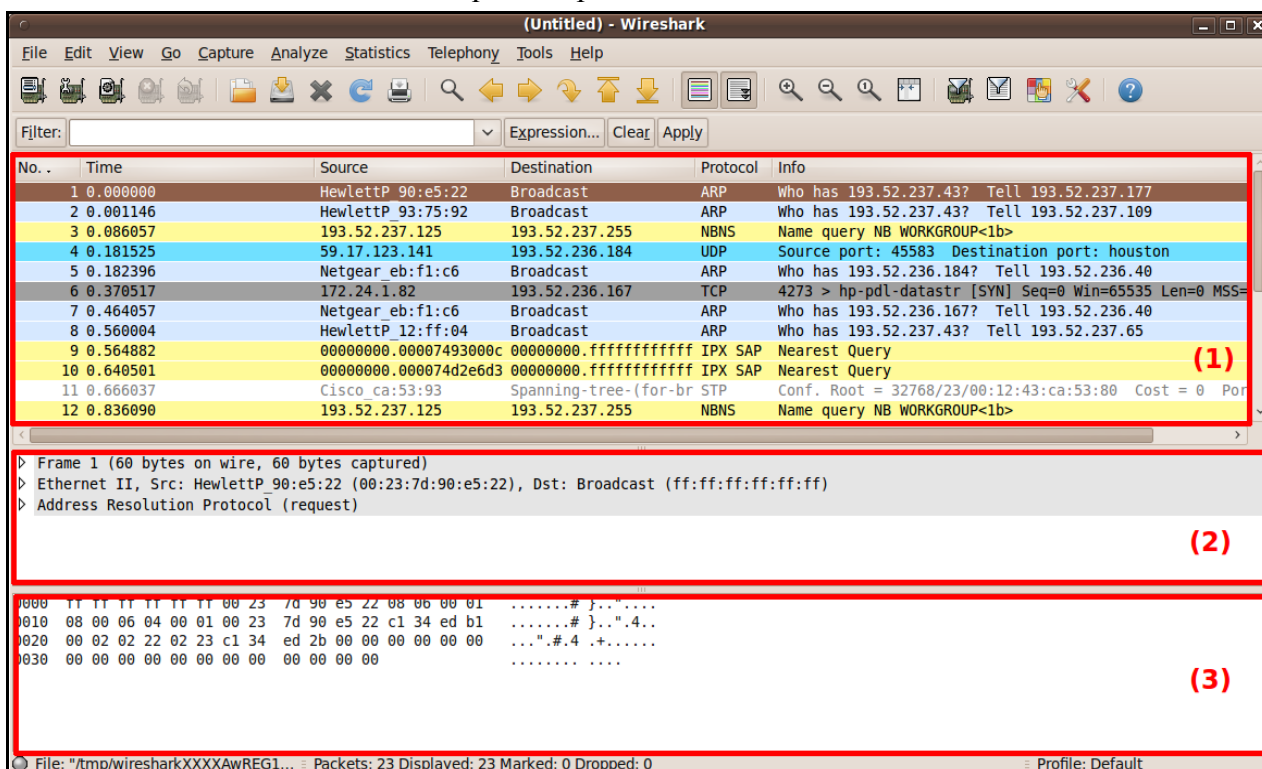


Figure 3: interface de l'analyseur

L'interface de l'analyseur est découpé en trois zone :

- Zone supérieure, numérotée (1) sur Figure 3 : liste l'ensemble des paquets capturés
- Zone centrale, numérotée (2) sur Figure 3 : 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) sur Figure 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

Encapsulation d'un paquet

Lorsqu'un paquet est sélectionné, la zone centrale permet de visualiser clairement les différentes couches d'encapsulation du paquet. Par exemple si l'on sélectionne un paquet de type UDP, la zone centrale pourrait afficher quelque chose de similaire à ce qui est présenté Figure 4. Les 5 entrées présentées correspondent à différentes encapsulations, ordonnées de la couche la plus basse à la couche la plus haute :

1. Données sur le média de capture : Wire = filaire sur Figure 4
2. Trame relative à la couche liaison de donnée : Ethernet II sur Figure 4
3. Paquet relatif à la couche réseau : Internet Protocol sur Figure 4
4. Datagramme relatif à la couche transport : User Datagram Protocol sur Figure 4
5. Données de l'application : regroupe généralement les couches session, présentation, application.

```
▶ Frame 5765 (65 bytes on wire, 65 bytes captured)
▶ Ethernet II, Src: Dell_b3:04:ee (00:1d:09:b3:04:ee), Dst: CompalIn 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)
```

Figure 4: Encapsulation d'un paquet UDP, zone centrale de l'analyseur.

Détail de chaque 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 présenté en Figure 5, nous avons étendu les entrées correspondant aux couches réseau, transport et application en cliquant sur les triangles correspondants. Nous pouvons voir entre autre que :

- Le paquet est de type IP v4 : ref (2) sur Figure 5
- Le type de données de ce paquet IP est un datagramme UDP : ref (5) sur Figure 5
- L'ip de la machine source est 193.52.236.243 : ref (6) sur Figure 5

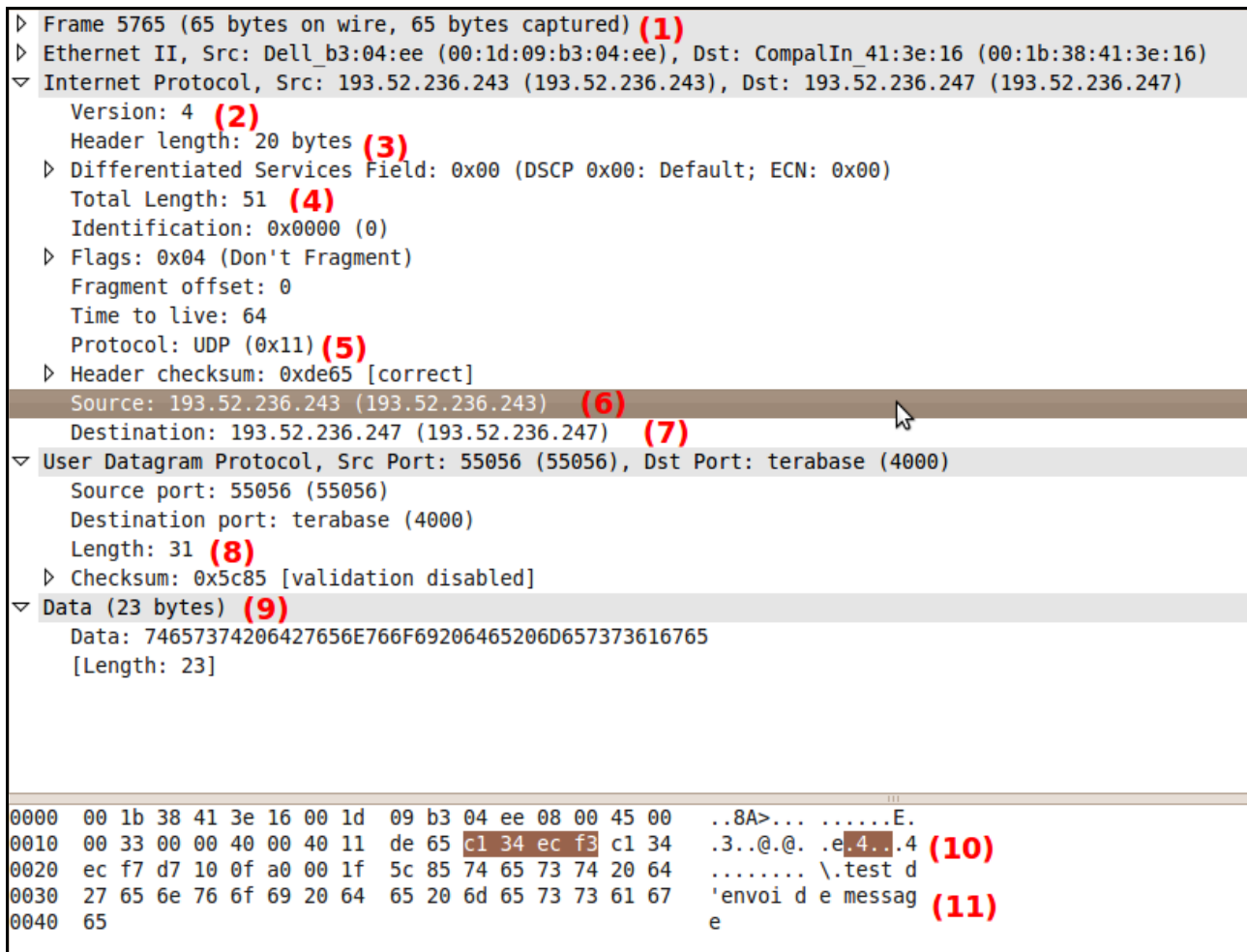


Figure 5: Visualisation détaillée des en-tetes d'un paquet

- L'ip de la machine destination est 193.52.236.247 : ref (7) sur Figure 5

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

- La taille des données envoyée par le processus est de 23 octets : ref (9) sur Figure 5
- La taille totale du datagramme UDP est de 31 octets - ref (8) sur Figure 5 - . Cette valeur est la somme entre la taille réelle des données (23 octets) et 8 octets d'en-tête du paquet (8 étant une valeur fixe pour un paquet UDP)
- La taille des en-têtes du paquet IP est de 20 octets : ref (3) sur Figure 5
- Le paquet IP contient un en-tete (20 octets) ainsi que le datagramme UDP (31 octets). Sa taille totale est de 51 octets, taille rappelée en ref (4) sur Figure 5
- La taille totale du paquet IP est de (flèche). Cette valeur somme la taille du paquet UDP à la taille de l'en-tête IP.
- Si l'on ajoute 12 octets d'en-tete pour la couche Ethernet II (taille fixe), la taille totale de la trame est de 65 octets, comme présentée en ref (1) sur Figure 5.

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 supplémentaires avant la trame Ethernet. Ces octets seront ici passés sous silence).

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. Ainsi dans l'exemple présenté en figure Figure 5, un clic sur le champ « Source » de la couche réseau - voir ref (6) sur Figure 5 - mettra en évidence dans la zone inférieure les octets codant cette source - voir ref (10) sur Figure 5 - et réciproquement.

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 seraient 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 ».

Filtrage de paquets

Principe et mise en place d'un 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 pré-définis 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 ».

Quelques filtres

Nous terminerons ce tutoriel en énonçant quelques filtres possibles :

Désignation	Filtre associé :
Segments TCP uniquement.....	tcp
Paquets relatifs à TCP uniquement.....	ip.proto == 0x06
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
Trafic MSN.....	tcp && tcp.port ==1863