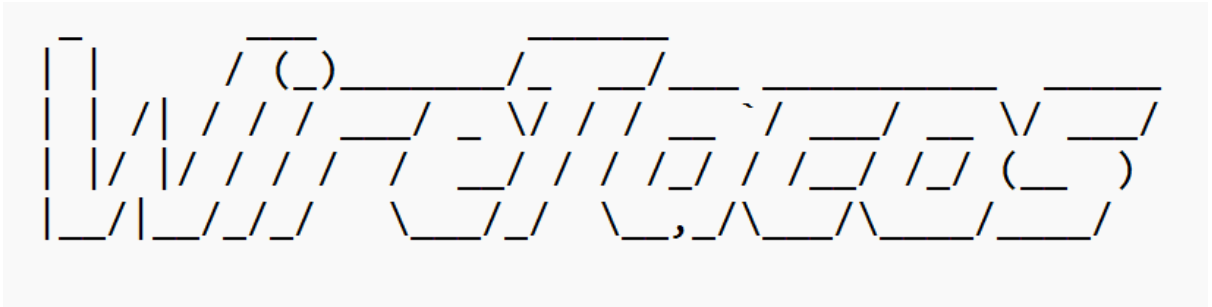


HOW TO



Étudiants: Audigier Roman
Iordache Paul-Tiberiu

Groupe: 8

Année universitaire: 2022-2023

1. Téléchargez le document.
2. Lancez le terminal.
3. Mettez vous dans le répertoire à l'aide de la commande `cd nom_fichier`.

```
● roman@roman-X405UA:~/Documents/L3/Reseaux$ cd WireTacos
○ roman@roman-X405UA:~/Documents/L3/Reseaux/WireTacos$
```

4. Compiler le programme en tapant la commande `make` dans le terminal.

```
● roman@roman-X405UA:~/Documents/L3/Reseaux/WireTacos$ make
gcc -Wall -g -c lecture.c
gcc -Wall -g -c format.c
gcc -Wall -g -c liste.c
gcc -Wall -g -c ethernet.c
gcc -Wall -g -c IP.c
gcc -Wall -g -c ARP.c
gcc -Wall -g -c ICMP.c
gcc -Wall -g -c TCP.c
gcc -Wall -g -c HTTP.c
gcc -Wall -g -c menu.c
gcc -g -o main.o -c main.c
gcc -o main lecture.o format.o liste.o ethernet.o IP.o ARP.o ICMP.o TCP.o HTTP.o menu.o main.o
○ roman@roman-X405UA:~/Documents/L3/Reseaux/WireTacos$
```

5. Pour exécuter et lancer le programme, tapez `./WireTacos.x`

```
○ roman@roman-X405UA:~/Documents/L3/Reseaux/WireTacos$ ./WireTacos.x
WireTacos v1
© IORDACHE - AUDIGIER
Bonjour! Bienvenue dans WireTacos!
Veuillez rentrer le fichier que vous voulez analyser dans le format (ex: nom_fichier.txt)

```

6. Maintenant vous êtes dans le logiciel WireTacos. Suivez les instructions. Lorsque la phrase: Veuillez entrer le fichier que vous voulez analyser dans le format (ex: nom_fichier.txt), apparaît, introduisez le nom du fichier en format .txt (nous préciserons que le fichier test.txt doit être dans le même dossier avec le logiciel, dans ce cas, dans le dossier WireTacos). Pour donner un exemple, on tape dans le terminal `test.txt`.

[illegible]

7. Lorsque la phrase: Veuillez entrer le nom du fichier d'output apparaît, taper le nom du fichier d'output où vous voulez enregistrer l'analyse des flux. Exemple: (nous l'avons appelé output)

```

Veillez entrer le nom du fichier d'output (il sera au format.txt)
output
Voulez-vous ajouter des filtres? (y/n)

```

8. Lorsque la phrase: Voulez-vous ajouter des filtres apparaît, tapez **y** pour oui et **n** pour non. Si vous ne souhaitez pas de filtres, alors un fichier .txt va apparaître dans le dossier

WireTacos et vous devrez voir le flux du fichier que vous avez introduit. Pour une meilleure lisibilité, nous vous recommandons d'ouvrir le fichier avec vim. En plus, notre logiciel va aussi afficher dans le terminal le flux de votre fichier. Voici un exemple d'affichage dans le terminal.

```
Voulez-vous ajouter des filtres? (y/n)
n

-----
n°1
IP:      src : 0.0.0.0 -----> 255.255.255.255 : dest

protocole encapsulé par IP non pris en charge ... pour l'instant

-----
n°2
IP:      src : 192.168.99.1 -----> 192.168.99.199 : dest

protocole encapsulé par IP non pris en charge ... pour l'instant

-----
n°3
IP:      src : 0.0.0.0 -----> 255.255.255.255 : dest

protocole encapsulé par IP non pris en charge ... pour l'instant

-----
n°4
IP:      src : 192.168.99.1 -----> 192.168.99.199 : dest

protocole encapsulé par IP non pris en charge ... pour l'instant

-----
n°5
      src : f0:18:98:59:ae:32 -----> ff:ff:ff:ff:ff:ff : dest

ARP:    Who has 192.168.99.1 ? tell 192.168.99.199

-----
n°6
      src : 0c:8d:db:1a:1e:88 -----> f0:18:98:59:ae:32 : dest

ARP:    192.168.99.1 is at 0c:8d:db:1a:1e:88
```

Comme on a déjà dit, pour une meilleure lisibilité vous pouvez ouvrir le fichier output.txt qui apparait dans le dossier WireTacos, soit d'ouvrir le fichier avec l'éditeur vim (nous vous recommandons de l'utiliser). Après avoir installé vim, voici un exemple d'affichage.

```
-----
n°37
IP:      src : 10.8.0.254 -----> 255.255.255.255 : dest

protocole encapsulé par IP non pris en charge ... pour l'instant
[tibi@fedora WireTacos]$ vim output.txt
```

```

tibi@fedora:~/WireTacos — vim output.txt
-----|
n°1                                           0.0.0.0---->255.255.255.255
(UDP)
-----|
n°2                                           192.168.99.199<-----192.168.99.1
(UDP)
-----|
n°3                                           0.0.0.0---->255.255.255.255
(UDP)
-----|
n°4                                           192.168.99.199<-----192.168.99.1
(UDP)
-----|
n°5                                           f0:18:98:59:ae:32---->ff:ff:ff:ff:ff:ff
ARP:   Who has 192.168.99.1 ? tell 192.168.99.199
-----|
n°6                                           f0:18:98:59:ae:32<-----0c:8d:db:1a:1e:88
ARP:   192.168.99.199 is at 0c:8d:db:1a:1e:88
-----|
n°7                                           192.168.99.199----->35.176.240.100
(UDP)
-----|
n°8                                           192.168.99.199<-----35.176.240.100
(UDP)
-----|
n°9                                           192.168.99.199----->128.119.245.12
TCP :   source port :60779  ->  80 : dest port

```

9. Dans le cas ou vous souhaitez de filtres, vous tapez y et vous devez voir ça:

```

output
Voulez-vous ajouter des filtres? (y/n)
y
-----
Quel filtre voulez-vous ajouter?
1-Ethernet
2-IPv4
3-ARP
4-ICMP
5-TCP
6-Numero de la trame (dans le fichier)

```

10. Disons, par exemple, qu'on veut ajouter un filtre pour les adresses IP (vous allez voir chaque protocole que notre analyseur sera capable de traiter). Alors, vous tapez 2 dans le terminal. Puis, vous avez 2 choix: soit de faire un filtrage selon l'adresse IP, soit selon le protocole encapsulé par IP. Si vous choisissez le filtre selon le protocole, tapez 2. Puis, c'est demandé d'entrer la valeur du protocole en hexadécimal. Dans ce cas, nous avons choisi le protocole TCP (valeur 6 en décimal). Après avoir tapé 6 dans le terminal, vous devrez voir le flux affiché dans le terminal, ainsi que dans le fichier output. Voici l'exemple d'affichage dans le terminal.

```
-----
Quel filtre voulez-vous ajouter?
1-Ethernet
2-IPv4
3-ARP
4-ICMP
5-TCP
6-Numero de la trame (dans le fichier)
2
-----
1 - ip addr == ?
2 - protocol == ? (RAPPEL : ce programme supporte ICMP et TCP)
2
-----
entrez la valeur du protocole en hexadecimal
0x6

-----
n°1
IP:      src : 192.168.99.199 ----->128.119.245.12 : dest
TCP :    source port :60779 -> 80 : dest port
      SYN Seq:3230305653 Ack:0 Window:65535

-----
n°2
IP:      src : 128.119.245.12 ----->192.168.99.199 : dest
TCP :    source port :80 -> 60779 : dest port
      SYN ACK Seq:2750420604 Ack:3230305654 Window:28960

-----
n°3
IP:      src : 192.168.99.199 ----->128.119.245.12 : dest
TCP :    source port :60779 -> 80 : dest port
      ACK Seq:3230305654 Ack:2750420605 Window:2058

-----
n°4
IP:      src : 192.168.99.199 ----->128.119.245.12 : dest
TCP :    source port :60779 -> 80 : dest port
      PSH ACK Seq:3230305654 Ack:2750420605 Window:2058
HTTP: GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
```

Ici, vous avez l'exemple affiché dans le fichier output.txt (de nouveau avec l'aide de l'éditeur vim en suivant les mêmes commandes que pour le point 8).

```
n°1
TCP : source port :60779 -> 80 : dest port 192.168.99.199----->128.119.245.12
      SYN Seq:3230305653 Ack:0 Window:65535

n°2
TCP : source port :80 -> 60779 : dest port 192.168.99.199<-----128.119.245.12
      SYN ACK Seq:2750420604 Ack:3230305654 Window:28960

n°3
TCP : source port :60779 -> 80 : dest port 192.168.99.199----->128.119.245.12
      ACK Seq:3230305654 Ack:2750420605 Window:2058

n°4
TCP : source port :60779 -> 80 : dest port 192.168.99.199----->128.119.245.12
      PSH ACK Seq:3230305654 Ack:2750420605 Window:2058
HTTP: GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1

n°5
TCP : source port :80 -> 60779 : dest port 192.168.99.199<-----128.119.245.12
      ACK Seq:2750420605 Ack:3230306200 Window:235

n°6
TCP : source port :80 -> 60779 : dest port 192.168.99.199<-----128.119.245.12
      PSH ACK Seq:2750420605 Ack:3230306200 Window:235
HTTP: HTTP/1.1 200 OK

n°7
TCP : source port :60779 -> 80 : dest port 192.168.99.199----->128.119.245.12
      ACK Seq:3230306200 Ack:2750421043 Window:2052
```

11. Maintenant, c'est à vous de jouer! Essayez les différents filtres pour trouver les informations que vous cherchez.