

UNIVERSITÉ NATIONALE DU VIETNAM À HANOÏ

Institut de la Francophonie pour l'Innovation



Conception et Architectures des Réseaux Travaux pratiques N°1

Systèmes Intelligents et Multimédia (SIM)

Promotion : 22

Deuxième partie TP1 : Outils réseaux sous Linux

Rédigé par :

ALPHONSE Rooldy

GOINT Mongetro

KINDA Zakaria

SEMEURAND Myderson

OUBDA Raphael Nicolas Wendyam

ZONGO Sylvain

Enseignant :

Dr Nguyen Hong Quang

Année académique : 2017 - 2018

Table des matières

1	Introduction	1
2	Outils de configuration	2
2.1	Liste des interfaces sur notre machine	2
2.2	Adresse IP de notre machine	2
2.3	Adresse MAC de notre machine	2
2.4	Adresse et masque du réseau	3
2.5	Table de routage de notre machine	3
2.6	Nom de domaine a partir de l'adresse IP	3
2.7	Explications sur la configuration des interfaces wifi sous Linux	4
2.8	Liste des routeurs traversée	4
2.9	Les serveurs de nom	5
2.9.1	Pour fpt.com.vn	5
2.9.2	Pour ifi.edu.vn	6
3	Analyse de protocole de communication à l'aide d'outils pour capture de trame	6
3.1	Analyse du protocole de résolution d'adresse ARP	6
3.2	Analyse des routes suivies par les paquets avec l'outil mtr	8
3.3	Analyse du protocole telnet et la capture des informations	10
3.3.1	Récupération du login.	12
3.3.2	Récupération du mot.	13
4	Analyse du protocole TCP	16
5	Conclusion	17

Table des figures

1	Liste des interfaces réseaux	2
2	Table de routage	3
3	Nom du domaine avec la commande nslookup	3
4	Ligne de code de configuration wifi sur linux	4
5	Liste des routeurs traversée	5
6	Nom de serveur de fpt.com.vn	5
7	Nom de serveur de ifi.edu.vn	6
8	Cache ARP	6
9	Résultat du Ping de 192.168.43.182	7
10	Request	7
11	Reply	8
12	Cache ARP	8
13	Suivie des paquets entre notre machine et le site burkina24.com	9
14	Capture des paquets générés par mtr	10
15	Connexion au serveur telnet	11
16	Datagrammes capturés	11
17	Datagramme contenant la première lettre du login	12
18	Datagramme contenant la deuxième lettre du login	12
19	Datagramme contenant la troisième lettre du login	13
20	Récupération du mot de passe	14
21	Datagramme contenant la première lettre du mot de passe	14
22	Datagramme contenant la deuxième lettre du mot de passe	15
23	Datagramme contenant la troisième lettre du mot de passe	15
24	Séquence de connexion de nos deux machines via TCP (tcpdump)	16
25	Séquence de connexion de nos deux machines via TCP (wireshark)	16

1 Introduction

De nos jours, la technologie tend à interconnecter le monde entier. Cette interconnexion est indispensable pour toute personne des IT. Cette interconnexion des équipements appelée “réseau” est une discipline qui existe depuis longtemps, ainsi pour comprendre les principes de cette discipline, le fonctionnement, l’analyse des échanges et les protocoles de communication à l’aide des programmes par les captures des trames , ce TP nous a été donné pour étude et manipulation des commandes. Les TPS ont été réalisés sur le système d’exploitation ubuntu. Ce présent rapport est le fruit de nos résultats obtenus et de l’analyse.

2 Outils de configuration

2.1 Liste des interfaces sur notre machine

Une interface définit la frontière de communication entre deux entités, comme des éléments de logiciel, des composants de matériel informatique, ou un utilisateur. Elle se réfère généralement à une image abstraite qu'une entité fournit d'elle-même à l'extérieur. A cet effet pour afficher les interfaces de notre machine, nous allons utiliser la commande **ifconfig**.

```
kinda-zak@kindazak:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 78:e3:b5:60:95:50
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          Packets reçus:0 erreurs:0 :0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:0 (0.0 B) Octets transmis:0 (0.0 B)
          Interruption:41 Adresse de base:0x2000

lo        Link encap:Boucle locale
          inet addr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          Packets reçus:40319 erreurs:0 :0 overruns:0 frame:0
          TX packets:40319 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          Octets reçus:4768947 (4.7 MB) Octets transmis:4768947 (4.7 MB)

wlan0     Link encap:Ethernet  HWaddr ac:81:12:cf:f5:92
          inet adr:192.168.0.131  Bcast:192.168.0.255  Masque:255.255.255.0
          adr inet6: fe80::ae81:12ff:fecf:f592/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:2198722 erreurs:0 :0 overruns:0 frame:0
          TX packets:2877734 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:913616628 (913.6 MB) Octets transmis:1416165623 (1.4 GB)
```

FIGURE 1 – Liste des interfaces réseaux

eth0 : Il s'agit de l'interface ethernet, sans aucune adresse ip car elle n'est connecté à aucune réseau et à pour adresse.

lo : est l'interface de l'hôte pour la boucle locale de notre machine.

wlan0 : Représente l'interface du réseau sans fil (Wifi) et est connecté à un réseau local via l'adresse : 192.168.0.131.

2.2 Adresse IP de notre machine

En utilisant la commande **ifconfig**, nous obtenons les adresses ip suivantes sur nos différentes interfaces :

lo : nous avons l'adresse ip **127.0.0.1** qui est l'adresse de boucle locale de notre machine.

wlan0 : L'adresse IP de l'interface sans fil est **192.168.0.131**.

2.3 Adresse MAC de notre machine

Toujours en utilisant la commande **ifconfig** nous pouvons présenter l'adresse Mac de nos différentes interfaces comme suite :

eth0 : L'adresse MAC de cette interface eth0 est **78 :e3 :b5 :60 :95 :50**

wlan0 : L'adresse MAC de l'interface wlan0 est **ac :81 :12 :cf :f5 :92**

2.4 Adresse et masque du réseau

Il s'agit pour nous de déterminer l'adresse du réseau en utilisant le masque du réseau et l'adresse ip de notre machine.

- Le masque du réseau est : **255.255.255.0**, de classe C.
- Adresse de notre machine est : **192.168.0.131**
- Adresse du réseau est : **192.168.0.0**

2.5 Table de routage de notre machine

Pour déterminer la table de routage de notre machine, nous allons utiliser la commande **route**.

```
kinda-zak@kindazak:~$ route
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref       Use Iface
default          192.168.0.19    0.0.0.0          UG     0      0        0 wlan0
link-local       *               255.255.0.0      U      1000    0        0 wlan0
192.168.0.0      *               255.255.255.0    U       2      0        0 wlan0
```

FIGURE 2 – Table de routage

Nous observons donc à travers cette commande que l'adresse du réseau est 192.168.0.0, le masque est **255.255.255.0** et la passerelle est **198.168.0.19** (figure 2). Ainsi pour sortir de notre réseau vers n'importe quelle destination, il faut passer cette adresse, sans oublier qu'elle est connectée à l'interface wlan0. Notons aussi que le masque de notre adresse réseau est : **255.255.255.0** correspondant à une adresse de classe C utilisé pour les réseaux privés.

2.6 Nom de domaine a partir de l'adresse IP

Pour obtenir le nom du seueur à partir de 'adresse IP 112.137.140.41 , nous utilisons la commande **nslookup**.

```
kinda-zak@kindazak:~$ nslookup 112.137.140.41
Server:          127.0.0.1
Address:         127.0.0.1#53

Non-authoritative answer:
*** Can't find 41.140.137.112.in-addr.arpa.: No answer

Authoritative answers can be found from:
```

FIGURE 3 – Nom du domaine avec la commande nslookup

L'utilisation de cette commande (`nslookup 112.137.140.41` ne donne aucune information sur le nom du domaine (figure 3). Cependant, notons que cette adresse **112.137.140.41** fait parti d'une plage d'adresse qui appartient à l'Université Nationale du Vietnam (**112.137.140.0** à **112.137.140.255**).

2.7 Explications sur la configuration des interfaces wifi sous Linux

Pour configurer une interface wifi sous Linux sans avoir recours à des outils graphiques, on a de deux moyens : soit on utilise des commandes, soit on modifie le fichier « `/etc/network/interfaces` ».

Ligne de commande : La commande « `ifconfig` » nous permet de modifier directement la configuration d'une interface wifi sous linux. Par exemple, si nous voulons attribuer l'adresse « `192.168.0.32` » à notre interface wifi, il nous faut taper la commande : « `ifconfig wlan0 "192.168.0.32" netmask 255.255.255.0` ».

Modification du fichier `/etc/network/interfaces` : dans la figure 4, nous montrons les lignes de code concernant la configuration de l'interface wifi sur linux.



```
GNU nano 2.5.3      File: /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto wlp5s0
iface wlp5s0 inet dhcp
wireless-essid "perso"
wireless-mode managed
wireless-key "password"

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

FIGURE 4 – Ligne de code de configuration wifi sur linux

2.8 Liste des routeurs traversée

Dans cette section, nous allons lister les différents routeurs traversés pour la connexion entre de machine. Pour ce faire nous installons tout d'abord `traceroute` à travers la commande : `aptitude install traceroute`. Après l'installation, nous tapons la commande `traceroute 112.137.140.41` qui permet de lister l'ensemble des routeurs par lesquels traversent les datagramme de notre machine à la machine dont l'adresse est `112.137.140.41`.

```

kinda-zak@kindazak:~$ traceroute 112.137.140.41
traceroute to 112.137.140.41 (112.137.140.41), 30 hops max, 60 byte packets
 1  logout.lan (10.223.20.1)  7.350 ms  8.985 ms  9.072 ms
 2  * * *
 3  172.31.99.22 (172.31.99.22)  20.889 ms  21.575 ms  21.161 ms
 4  static.vnpt-hanoi.com.vn (123.25.27.177)  13.987 ms  14.068 ms  13.692 ms
 5  static.vnpt.vn (113.171.16.245)  14.056 ms  13.941 ms  static.vnpt.vn (123.29
.5.41)  13.749 ms
 6  static.vnpt.vn (113.171.33.81)  13.864 ms  static.vnpt.vn (113.171.35.157)  7
.411 ms *
 7  static.vnpt.vn (113.171.5.10)  7.303 ms  static.vnpt.vn (113.171.34.114)  7.2
85 ms  11.482 ms
 8  125.235.241.5 (125.235.241.5)  7.333 ms  static.vnpt.vn (123.29.16.86)  12.90
1 ms  13.398 ms
 9  localhost (27.68.228.37)  14.215 ms  14.215 ms  16.203 ms
10  localhost (27.68.229.226)  82.625 ms  localhost (27.68.229.230)  6.876 ms  8.
338 ms
11  localhost (27.68.229.237)  11.744 ms  11.052 ms  12.130 ms
12  localhost (27.68.229.50)  8.340 ms  9.839 ms  9.406 ms
13  203.113.156.146 (203.113.156.146)  8.909 ms  8.914 ms  9.061 ms
14  112.137.140.41 (112.137.140.41)  7.763 ms  5.089 ms  5.280 ms

```

FIGURE 5 – Liste des routeurs traversée

Nous remarquons que pour atteindre la machine d'adresse 112. 137.140.41 les datagrammes ont traversés 14 routeurs comme listés sur la figure ci-dessus (figure 5).

2.9 Les serveurs de nom

Dans cette partie nous devons retrouver les noms des serveurs pour des domaines.

2.9.1 Pour fpt.com.vn

Nous utilisons la commande dig NS fpt.com.vn pour trouver le nom du serveur du domaine fpt.com.vn. La figure ci-dessous (figure 6) illustre le résultat obtenu :

```

kinda-zak@kindazak:~$ dig NS fpt.com.vn

;<<>> DiG 9.8.1-P1 <<>> NS fpt.com.vn
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 52377
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;fpt.com.vn.                IN      NS

;; ANSWER SECTION:
fpt.com.vn.                 3599    IN      NS      ns2.zonedns.vn.
fpt.com.vn.                 3599    IN      NS      ns1.zonedns.vn.
fpt.com.vn.                 3599    IN      NS      ns3.zonedns.vn.

;; Query time: 146 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed Jul 11 23:35:53 2018
;; MSG SIZE rcvd: 90

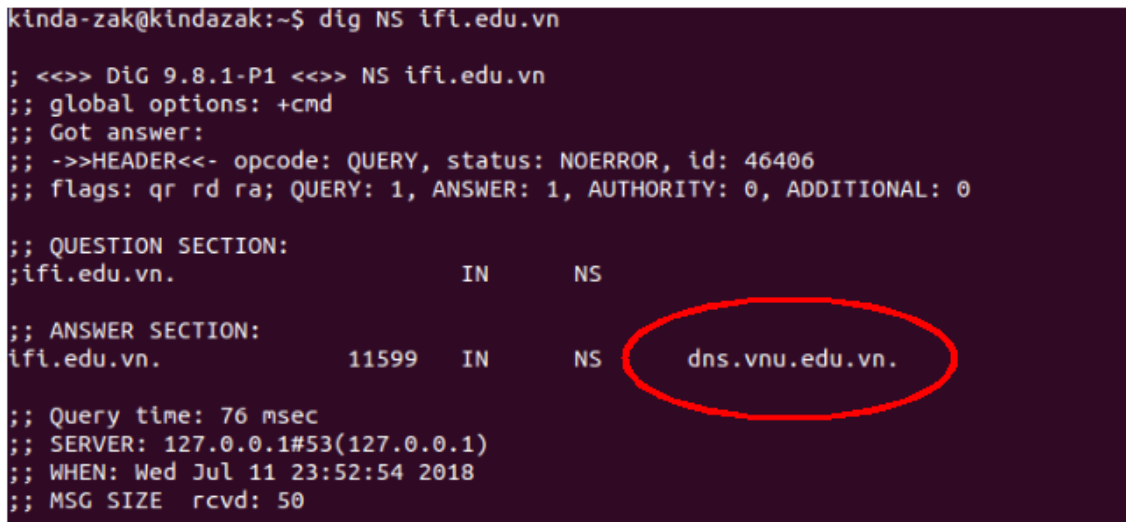
```

FIGURE 6 – Nom de serveur de fpt.com.vn

Les noms des serveurs sont encadrés en rouge sur la figure ci-dessus.

2.9.2 Pour ifi.edu.vn

Nous utilisons la commande dig NS ifi.edu.vn pour trouver le nom du serveur du domaine fpt.com.vn. La figure 7 illustre le résultat obtenu :



```
kinda-zak@kindazak:~$ dig NS ifi.edu.vn

; <<>> DiG 9.8.1-P1 <<>> NS ifi.edu.vn
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 46406
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;ifi.edu.vn.                IN      NS

;; ANSWER SECTION:
ifi.edu.vn.                11599   IN      NS      dns.vnu.edu.vn.

;; Query time: 76 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed Jul 11 23:52:54 2018
;; MSG SIZE rcvd: 50
```

FIGURE 7 – Nom de serveur de ifi.edu.vn

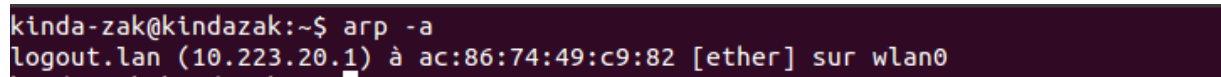
A travers la figure ci-dessus nous remarquons le nom du serveur (indiquer en rouge) du domaine ifi.edu.vn.

3 Analyse de protocole de communication à l'aide d'outils pour capture de trame

Dans cette section, nous devons analyser les protocoles de communication et capturer les trames du réseau en utilisant des outils de capture comme wireshark.

3.1 Analyse du protocole de résolution d'adresse ARP

Pour analyser le protocole de résolution d'adresse ARP, nous consultons d'abord le cache ARP à l'aide de la commande arp -a. La figure 8 illustre le résultat obtenu.



```
kinda-zak@kindazak:~$ arp -a
logout.lan (10.223.20.1) à ac:86:74:49:c9:82 [ether] sur wlan0
```

FIGURE 8 – Cache ARP

Effectuons le ping de l'adresse 192.168.43.182. La figure 9 ci-dessous illustre le résultat obtenu :

```

kinda-zak@kindazak:~$ ping 192.168.43.182
PING 192.168.43.182 (192.168.43.182) 56(84) bytes of data.
64 bytes from 192.168.43.182: icmp_req=1 ttl=64 time=1047 ms
64 bytes from 192.168.43.182: icmp_req=2 ttl=64 time=40.7 ms
64 bytes from 192.168.43.182: icmp_req=3 ttl=64 time=6.80 ms
64 bytes from 192.168.43.182: icmp_req=4 ttl=64 time=6.28 ms
64 bytes from 192.168.43.182: icmp_req=5 ttl=64 time=75.8 ms
64 bytes from 192.168.43.182: icmp_req=6 ttl=64 time=153 ms
64 bytes from 192.168.43.182: icmp_req=7 ttl=64 time=3.03 ms
64 bytes from 192.168.43.182: icmp_req=8 ttl=64 time=6.29 ms
64 bytes from 192.168.43.182: icmp_req=9 ttl=64 time=2.35 ms
64 bytes from 192.168.43.182: icmp_req=10 ttl=64 time=421 ms
64 bytes from 192.168.43.182: icmp_req=11 ttl=64 time=446 ms
64 bytes from 192.168.43.182: icmp_req=12 ttl=64 time=469 ms
64 bytes from 192.168.43.182: icmp_req=13 ttl=64 time=898 ms
64 bytes from 192.168.43.182: icmp_req=14 ttl=64 time=410 ms
64 bytes from 192.168.43.182: icmp_req=15 ttl=64 time=567 ms
64 bytes from 192.168.43.182: icmp_req=16 ttl=64 time=459 ms
64 bytes from 192.168.43.182: icmp_req=17 ttl=64 time=483 ms
64 bytes from 192.168.43.182: icmp_req=18 ttl=64 time=5.27 ms
64 bytes from 192.168.43.182: icmp_req=19 ttl=64 time=2.74 ms
64 bytes from 192.168.43.182: icmp_req=20 ttl=64 time=453 ms
64 bytes from 192.168.43.182: icmp_req=21 ttl=64 time=477 ms

```

FIGURE 9 – Résultat du Ping de 192.168.43.182

Nous constatons que la machine d'adresse IP 192.168.43.182 de notre réseau répond. A l'aide de l'outil de capture **wireshark**, capturons les trames pendant le ping de 192.168.43.182. La figure ci-dessous (figure 10) illustre le résultat obtenu.

```

▶ Frame 1967: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)
▼ Ethernet II, Src: GemtekTe_cf:f5:92 (ac:81:12:cf:f5:92), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    Address: Broadcast (ff:ff:ff:ff:ff:ff)
      .... 1 .... = IG bit: Group address (multicast/broadcast)
      .... 1 .... = LG bit: Locally administered address (this is NOT the factory default)
  Source: GemtekTe_cf:f5:92 (ac:81:12:cf:f5:92)
    Address: GemtekTe_cf:f5:92 (ac:81:12:cf:f5:92)
      .... 0 .... = IG bit: Individual address (unicast)
      .... 0 .... = LG bit: Globally unique address (factory default)
  Type: ARP (0x0806)
▼ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  [Is gratuitous: False]
  Sender MAC address: GemtekTe_cf:f5:92 (ac:81:12:cf:f5:92)
  Sender IP address: 192.168.43.32 (192.168.43.32)
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.43.182 (192.168.43.182)

0000  ff ff ff ff ff ff ac 81 12 cf f5 92 08 06 00 01 .....
0010  08 00 06 04 00 01 ac 81 12 cf f5 92 c0 a8 2b 20 .....+
0020  00 00 00 00 00 00 c0 a8 2b b6 .....+.

```

FIGURE 10 – Request

A travers la figure ci-dessus nous constatons que notre machine d'adresse MAC(14:1f:78:ea:14:85) émet une requête ARP(**Request**) en broadcast demandant l'adresse MAC de la machine qui possède l'adresse 192.168.43.182. De ce fait tous les équipements connectés au même segment reçoivent la requête mais seule l'équipement à l'adresse 192.168.43.182 peut y répondre. La figure 11 présente le résultat de la réponse à la requête obtenu par notre machine.

No.	Time	Source	Destination	Protocol	Length	Info
1998	616.949799	40:e2:30:d6:e8:a4	GemtekTe_cf:f5:92	ARP	42	192.168.43.182 is at 40:e2:30:d6:e8:a4
Frame 1998: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)						
Ethernet II, Src: 40:e2:30:d6:e8:a4 (40:e2:30:d6:e8:a4), Dst: GemtekTe_cf:f5:92 (ac:81:12:cf:f5:92)						
Destination: GemtekTe_cf:f5:92 (ac:81:12:cf:f5:92)						
Address: GemtekTe_cf:f5:92 (ac:81:12:cf:f5:92)						
....0.... = IG bit: Individual address (unicast)						
...0.... = LG bit: Globally unique address (factory default)						
Source: 40:e2:30:d6:e8:a4 (40:e2:30:d6:e8:a4)						
Address: 40:e2:30:d6:e8:a4 (40:e2:30:d6:e8:a4)						
....0.... = IG bit: Individual address (unicast)						
...0.... = LG bit: Globally unique address (factory default)						
Type: ARP (0x0806)						
Address Resolution Protocol (reply)						
Hardware type: Ethernet (1)						
Protocol type: IP (0x0800)						
Hardware size: 6						
Protocol size: 4						
Opcode: reply (2)						
[Is gratuitous: False]						
Sender MAC address: 40:e2:30:d6:e8:a4 (40:e2:30:d6:e8:a4)						
Sender IP address: 192.168.43.182 (192.168.43.182)						
Target MAC address: GemtekTe_cf:f5:92 (ac:81:12:cf:f5:92)						
Target IP address: 192.168.43.32 (192.168.43.32)						
0000 ac 81 12 cf f5 92 40 e2 30 d6 e8 a4 08 06 00 01@. 0.....						
0010 08 00 06 04 00 02 40 e2 30 d6 e8 a4 c0 a8 2b b6@. 0.....+						

FIGURE 11 – Reply

A travers cette figure, nous constatons que seule la machine d'adresse IP 192.168.43.182 répond(Reply) au request. La machine répond en transmettant son adresse MAC(40 :e2 :30 :d6 :e3 :a4) à notre machine.

En effet, la communication est établi entre les deux machines. De ce fait elles peuvent commencer à communiquer. Retapons la commande arp -a afin de révisualiser la cache ARP. La figure 12 ci-dessous représente le résultat obtenu :

```
kinda-zak@kindazak:~$ arp -a
one-CX61-2QF (192.168.43.182) à 40:e2:30:d6:e8:a4 [ether] sur wlan0
? (192.168.43.1) à a0:39:f7:75:1a:43 [ether] sur wlan0
kinda-zak@kindazak:~$
```

FIGURE 12 – Cache ARP

En effet, notre machine a ajouté dans le cache ARP l'adresse IP que nous avons pinguer ainsi que l'adresse MAC correspondant. De ce fait, la communication a été établi entre les deux machines les permettant ainsi, d'échanger des messages.

3.2 Analyse des routes suivies par les paquets avec l'outil mtr

MTR est une sorte de traceroute combinée avec ping. MTR indique chaque bond effectué par les paquets pour arriver à destination et il donne pour chaque bond le nombre de paquets perdus, la latence et des données statistiques. Nous installons le paquet MTR avec la commande : aptitude get install mtr-tiny

Dans cette section, nous allons suivre les différents paquets qui circulent entre notre machine et le

site **www.burkina24.com** sur internet. Pour cela nous utilisons la commande mtr **www.burkina24.com**. La figure 13 ci-dessous illustre le résultat obtenu.

```

My traceroute [v0.80]
kindazak (0.0.0.0) Fri Jul 13 01:01:46 2018
Keys: Help Display mode Restart statistics Order of fields quit
          Packets
Host      Loss%  Snt   Last   Avg    Best  Wrst  StDev
1.  logout.lan      2.9%  105    1.5    7.2    1.2  275.2  32.7
2.  ???
3.  118.70.0.13     5.8%  105    9.7    8.3    2.3  174.4  20.3
4.  113.22.4.117    4.8%  105   14.6    9.0    2.5  226.1  25.3
5.  42.112.2.194    4.8%  105    3.8    8.6    1.9  177.3  22.0
6.  42.112.2.195    2.9%  105    4.6    6.3    2.4  127.0  12.7
7.  118.69.166.149  3.8%  105   22.1   24.4   21.3   94.4   8.1
8.  te0-1-0-22.br03.hkg15.pccwbtn.ne 3.8%  105   53.0   56.1   50.0  264.0  28.3
9.  HundredGE0-5-0-0.br02.hkg08.pccw 2.9%  105   52.0   54.4   49.6  244.1  24.9
10. HundredGE0-5-0-0.br02.hkg08.pccw 1.9%  105   49.7   53.5   49.1  192.6  17.7
11. telnet.ge9-32.br01.hkg08.pccwbtn 2.9%  105   49.9   52.8   49.2  143.1  11.1
12. ae-6.r25.tkokhk01.hk.bb.gin.ntt. 1.9%  105   55.3   53.7   49.3  232.0  18.4
13. ae-1.r24.osakjp02.jp.bb.gin.ntt. 4.8%  105  100.8  103.4   99.4  281.1  18.3
14. ae-2.r22.snjsca04.us.bb.gin.ntt. 26.0%  104  201.1  204.3  200.0  318.8  14.0
15. ae-19.r01.snjsca04.us.bb.gin.ntt 2.9%  104  201.3  206.4  200.7  492.2  29.7
16. ae-1.a01.snjsca04.us.bb.gin.ntt. 4.8%  104  204.7  212.4  203.5  523.6  39.8
17. ae-0.endurance.snjsca04.us.bb.gi 4.8%  104  240.5  246.3  239.3  583.7  37.6
18. 162-144-240-159.unifiedlayer.com  6.8%  104  229.3  228.2  222.2  531.8  33.5
19. 162-144-240-55.unifiedlayer.com  4.9%  104  220.3  225.0  218.2  480.9  31.3
20. server.burkina24.com 4.8%  104  228.3  231.4  223.5  616.1  44.2

```

FIGURE 13 – Suivre des paquets entre notre machine et le site **burkina24.com**

A travers ce résultat, nous constatons que nos paquets passent par 20 routeurs pour arriver à destination c'est-à-dire au serveur qui héberge le site de **burkina.com**. L'utilisation de cet outil permet d'observer plusieurs informations. Il s'agit donc de :

- Loss : c'est le pourcentage des paquets perdus sur chaque noeud du chemin jusqu'à la destination ;
- Snt : il s'agit de nombre de paquets envoyés ;
- Last : il s'agit de la latence du dernier paquet envoyé, ainsi que la valeur moyenne ;
- Avg : est le temps maximal de réponse ;
- Best : est le temps minimal de réponse ;
- StDev : il s'agit de la déviation standard ou l'écart type des temps de réponse.

Pour mieux visualiser les paquets échangés entre notre machine et le serveur hébergeant le site web **burkina.com**, nous utilisons l'outil **wireshark**. La figure 14 présente le résultat obtenu :

No.	Time	Source	Destination	Protocol	Length	Info
7230	175.610630625	HundredGE0-5-0-0.br...	10.223.20.38	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
7231	175.611196180	10.223.20.38	server.burkina24.com	ICMP	78	Echo (ping) request id=0xea3b, seq=8437/62752, ttl=10 (no...
7232	175.661036517	HundredGE0-5-0-0.br...	10.223.20.38	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
7233	175.661432741	10.223.20.38	server.burkina24.com	ICMP	78	Echo (ping) request id=0xea3b, seq=8693/62753, ttl=11 (no...
7234	175.712362489	10.223.20.38	server.burkina24.com	ICMP	78	Echo (ping) request id=0xea3b, seq=8949/62754, ttl=12 (no...
7235	175.712505049	telnet.ges3-32.br01...	10.223.20.38	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
7236	175.762615521	ae-6.r25.tkokhk01.h...	10.223.20.38	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
7237	175.762769059	10.223.20.38	server.burkina24.com	ICMP	78	Echo (ping) request id=0xea3b, seq=9205/62755, ttl=13 (no...
7238	175.813259948	10.223.20.38	server.burkina24.com	ICMP	78	Echo (ping) request id=0xea3b, seq=9461/62756, ttl=14 (no...
7239	175.863591889	ae-1.r24.osakjp02.j...	10.223.20.38	ICMP	186	Time-to-live exceeded (Time to live exceeded in transit)
7240	175.864346101	10.223.20.38	server.burkina24.com	ICMP	78	Echo (ping) request id=0xea3b, seq=9717/62757, ttl=15 (no...
7241	175.915063627	10.223.20.38	server.burkina24.com	ICMP	78	Echo (ping) request id=0xea3b, seq=9973/62758, ttl=16 (no...
7242	175.965989517	10.223.20.38	server.burkina24.com	ICMP	78	Echo (ping) request id=0xea3b, seq=10229/62759, ttl=17 (n...
7243	176.016526237	10.223.20.38	server.burkina24.com	ICMP	78	Echo (ping) request id=0xea3b, seq=10485/62760, ttl=18 (n...
7244	176.065974068	ae-19.r01.snjsca04...	10.223.20.38	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
7245	176.067209792	10.223.20.38	server.burkina24.com	ICMP	78	Echo (ping) request id=0xea3b, seq=10741/62761, ttl=19 (n...
7246	176.118121006	10.223.20.38	server.burkina24.com	ICMP	78	Echo (ping) request id=0xea3b, seq=10997/62762, ttl=20 (r...
7247	176.119285314	ae-1.a01.snjsca04.u...	10.223.20.38	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
7248	176.168491487	10.223.20.38	server.burkina24.com	ICMP	78	Echo (ping) request id=0xea3b, seq=11253/62763, ttl=1 (no...
7249	176.171368987	logout.lan	10.223.20.38	ICMP	106	Time-to-live exceeded (Time to live exceeded in transit)
7250	176.208475530	ae-0.endurance.snjs...	10.223.20.38	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
7251	176.219046412	10.223.20.38	server.burkina24.com	ICMP	78	Echo (ping) request id=0xea3b, seq=11509/62764, ttl=2 (no...
7252	176.239613726	102-144-240-159.un1...	10.223.20.38	ICMP	106	Time-to-live exceeded (Time to live exceeded in transit)
7253	176.240433108	10.223.20.38	server.burkina24.com	ICMP	78	Echo (ping) request id=0xea3b, seq=11765/62765, ttl=3 (no...
7254	176.260800026	102-144-240-55.un1f...	10.223.20.38	ICMP	106	Time-to-live exceeded (Time to live exceeded in transit)
7255	176.291123797	10.223.20.38	server.burkina24.com	ICMP	78	Echo (ping) request id=0xea3b, seq=12021/62766, ttl=4 (no...
7256	176.292286086	118.70.0.13	10.223.20.38	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
7257	176.294846826	113.22.4.117	10.223.20.38	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
7258	176.341806127	10.223.20.38	server.burkina24.com	ICMP	78	Echo (ping) request id=0xea3b, seq=12277/62767, ttl=5 (no...
7259	176.343367065	server.burkina24.com	10.223.20.38	ICMP	78	Echo (ping) reply id=0xea3b, seq=10997/62762, ttl=46 (r...
7260	176.344066033	42.112.2.194	10.223.20.38	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
7261	176.392257470	10.223.20.38	server.burkina24.com	ICMP	78	Echo (ping) request id=0xea3b, seq=12533/62768, ttl=6 (no...

FIGURE 14 – Capture des paquets générés par mtr

Analyse du résultat.

Tout d'abord, nous constatons que l'adresse IP de notre machine ainsi que celle du serveur du site web sont fixes.

Lorsque nous lançons la commande mtr sur `www.burkina24.com`, notre machine envoie un paquet icmp qui contient l'adresse de `www.burkina24.com` et initialise avec time to live(TTL) 1. Lorsque notre machine ne reçoit pas de réponse provenant du serveur hébergeant le site web, il incrémente le TTL de 1, et renvoie un autre paquet vers le serveur jusqu'à obtenir une réponse de ce dernier. Elle réinitialise le TTL à 1.

3.3 Analyse du protocole telnet et la capture des informations

Dans cette section nous allons analyser le protocole telnet qui permet la connexion à distance et nous capturerons les informations avec l'outil Wireshark. Pour ce faire, nous installons d'abord telnet avec la commande suivante : `sudo apt-get install telnetd`.

Après l'installation nous nous connectons au serveur telnet à travers la commande `telnet add_serveur`. Nous renseignons le login et le password du serveur. La connexion est ainsi établie et nous précisons la version d'ubuntu installée sur le serveur.

Notons que le port source est 50887 et le port de destination 23. La figure 15 illustre cette connexion.

login : zak ;

password : zak.

```

nicolas@nicolas-Aspire-5749Z:~$ telnet 192.168.43.32
Trying 192.168.43.32...
Connected to 192.168.43.32.
Escape character is '^]'.
Ubuntu 12.04.5 LTS
kindazak login: zak
Password:
Last login: Fri Jul 13 04:04:07 ICT 2018 from nicolas-Aspire-5749Z on pts/7
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.2.0-126-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

New release '14.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

This Ubuntu 12.04 LTS system is past its End of Life, and is no longer
receiving security updates. To protect the integrity of this system, it's
critical that you enable Extended Security Maintenance updates:
 * https://www.ubuntu.com/esm

zak@kindazak:~$ █

```

FIGURE 15 – Connexion au serveur telnet

Avec l'outil Wireshark nous capturons les datagrammes du réseau. Pour ce faire nous filtrons uniquement les datagrammes telnet. La figure suivante (figure 16) illustre les datagrammes capturés.

```

nicolas@nicolas-Aspire-5749Z:~$ telnet 192.168.43.32
Trying 192.168.43.32...
Connected to 192.168.43.32.
Escape character is '^]'.
Ubuntu 12.04.5 LTS
kindazak login: zak
Password:
Last login: Fri Jul 13 04:04:07 ICT 2018 from nicolas-Aspire-5749Z on pts/7
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.2.0-126-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

New release '14.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

This Ubuntu 12.04 LTS system is past its End of Life, and is no longer
receiving security updates. To protect the integrity of this system, it's
critical that you enable Extended Security Maintenance updates:
 * https://www.ubuntu.com/esm

zak@kindazak:~$ █

```

FIGURE 16 – Datagrammes capturés

La capture des datagrammes enregistre les mots de passe et les logins du serveur. Donc à partir des datagrammes obtenus nous pouvons récupérer le login et le mot de passe du serveur. De ce fait, cette récupération se fait lettre par lettre contenu dans les datagrammes. À travers les figures 17 ci-dessous nous allons récupérer le login et le mot de passe du serveur.

3.3.1 Récupération du login.

Lorsque l'utilisateur entre le login pour la connexion au serveur, l'analyse des captures des datagrammes du réseau, nous permet de reconstituer le login qui est **zak**.

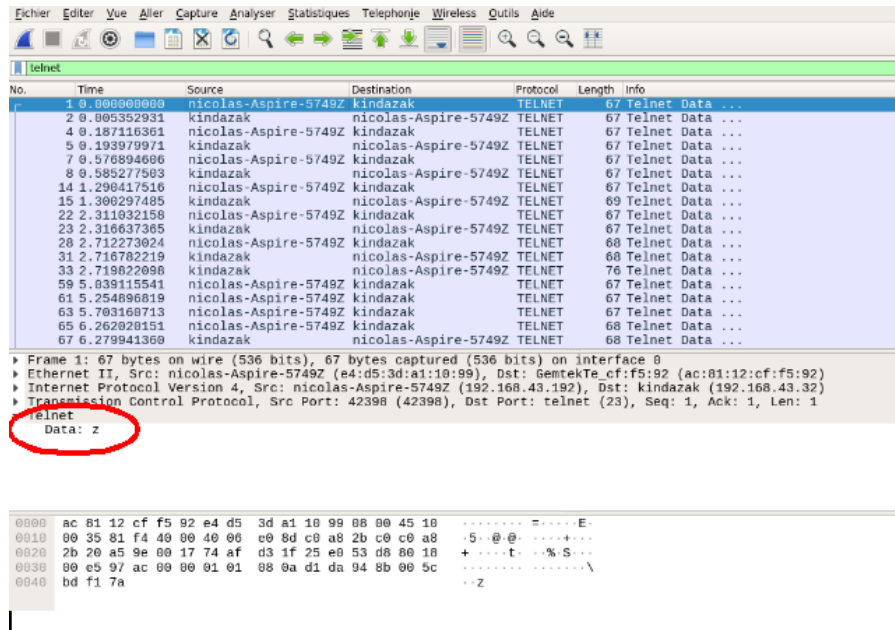


FIGURE 17 – Datagramme contenant la première lettre du login

Dans la figure 18 ci-dessous nous encadrons en rouge la première lettre récupérée du login qui est le **Z**. La figure ci-dessous présente la deuxième lettre récupérée qui est le **a**.

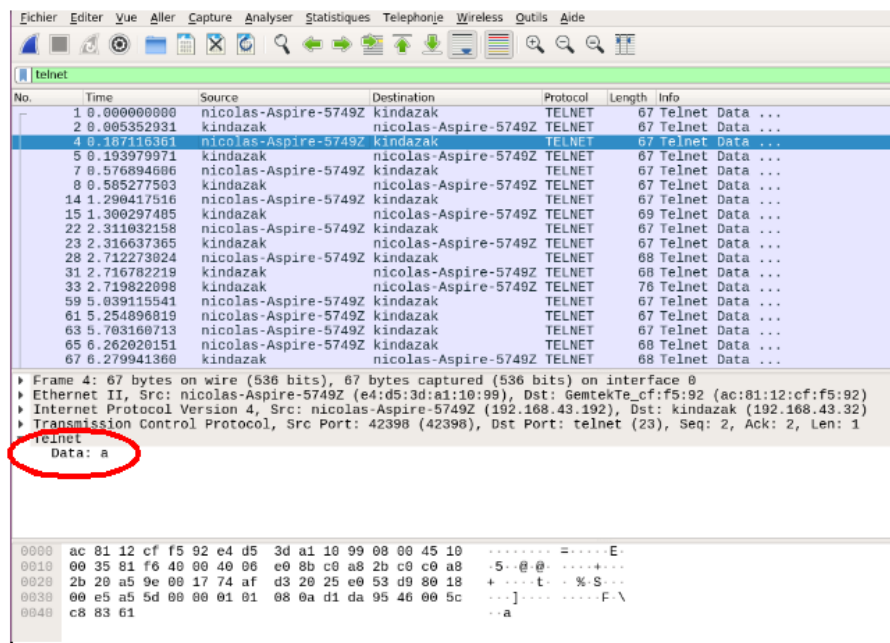


FIGURE 18 – Datagramme contenant la deuxième lettre du login

Enfin la figure suivante (figure 19) présente le datagramme de la dernière lettre du login récupéré qui correspond à la lettre **k**.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	nicolas-Aspire-5749Z	kindazak	TELNET	67	Telnet Data ...
2	0.005352931	kindazak	nicolas-Aspire-5749Z	TELNET	67	Telnet Data ...
4	0.187116361	nicolas-Aspire-5749Z	kindazak	TELNET	67	Telnet Data ...
5	0.193979971	kindazak	nicolas-Aspire-5749Z	TELNET	67	Telnet Data ...
7	0.576894606	nicolas-Aspire-5749Z	kindazak	TELNET	67	Telnet Data ...
8	0.585277503	kindazak	nicolas-Aspire-5749Z	TELNET	67	Telnet Data ...
14	1.298417516	nicolas-Aspire-5749Z	kindazak	TELNET	67	Telnet Data ...
15	1.300297485	kindazak	nicolas-Aspire-5749Z	TELNET	69	Telnet Data ...
22	2.311032158	nicolas-Aspire-5749Z	kindazak	TELNET	67	Telnet Data ...
23	2.316637365	kindazak	nicolas-Aspire-5749Z	TELNET	67	Telnet Data ...
28	2.712273024	nicolas-Aspire-5749Z	kindazak	TELNET	68	Telnet Data ...
31	2.716782219	kindazak	nicolas-Aspire-5749Z	TELNET	68	Telnet Data ...
33	2.719822098	kindazak	nicolas-Aspire-5749Z	TELNET	76	Telnet Data ...
59	5.039115541	nicolas-Aspire-5749Z	kindazak	TELNET	67	Telnet Data ...
61	5.254896819	nicolas-Aspire-5749Z	kindazak	TELNET	67	Telnet Data ...
63	5.703160713	nicolas-Aspire-5749Z	kindazak	TELNET	67	Telnet Data ...
65	6.262020151	nicolas-Aspire-5749Z	kindazak	TELNET	68	Telnet Data ...
67	6.279941360	kindazak	nicolas-Aspire-5749Z	TELNET	68	Telnet Data ...

▶ Frame 22: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface 0 ▶ Ethernet II, Src: nicolas-Aspire-5749Z (e4:d5:3d:a1:10:99), Dst: GemtekTe cf:f5:92 (ac:81:12:cf:f5:92) ▶ Internet Protocol Version 4, Src: nicolas-Aspire-5749Z (192.168.43.192), Dst: kindazak (192.168.43.32) ▶ Transmission Control Protocol, Src Port: 42398 (42398), Dst Port: telnet (23), Seq: 5, Ack: 7, Len: 1 ▶ Telnet Data: k						
--	--	--	--	--	--	--

0000	ac 81 12 cf f5 92 e4 d5	3d a1 10 99 08 00 45 10E.
0010	00 35 81 fc 40 00 40 06	e0 85 c0 a8 2b c0 c0 a8	-5-@-+...
0020	2b 20 a5 9e 00 17 74 af	d3 23 25 e0 53 de 00 18	+...t-##S...
0030	00 e5 01 c5 00 00 01 01	08 0a d1 da 9d 92 00 5c\
0040	c9 c7 6b		..k

FIGURE 19 – Datagramme contenant la troisième lettre du login

A travers la capture des datagrammes, nous avons pu reconstituer le login du serveur telnet. Donc les datagrammes contiennent des data(données). Ainsi nous avons reconstituer le login zak comme le montre les figures ci-dessous.

3.3.2 Récupération du mot.

Pour la récupération du mot de passe nous passons aussi par une analyse des datagrammes.

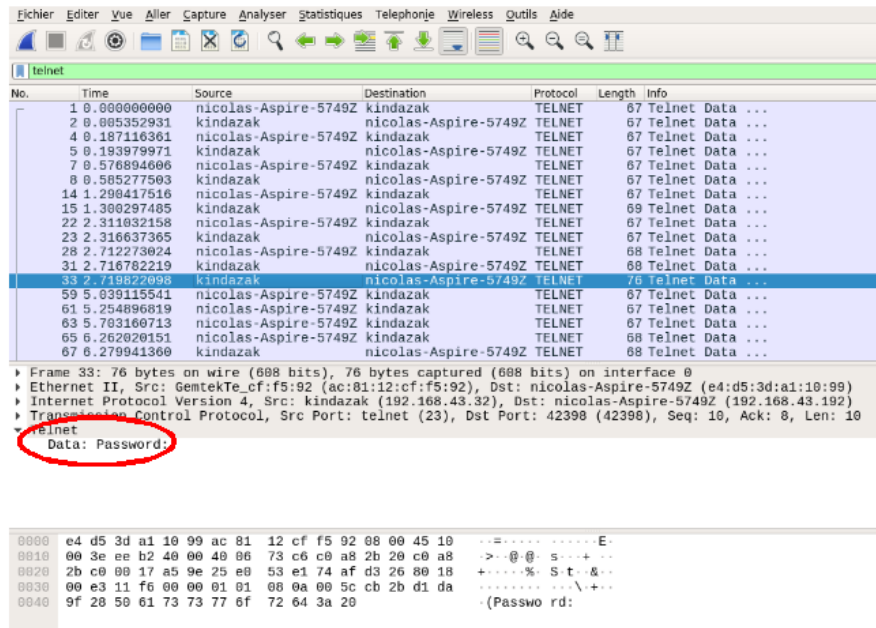


FIGURE 20 – Récupération du mot de passe

Cela permet de préparer la récupération du mot de passe qui est envoyé lettre par lettre dans les datagrammes. Ainsi la figure 21 montre la récupération de la première lettre du mot de passe.

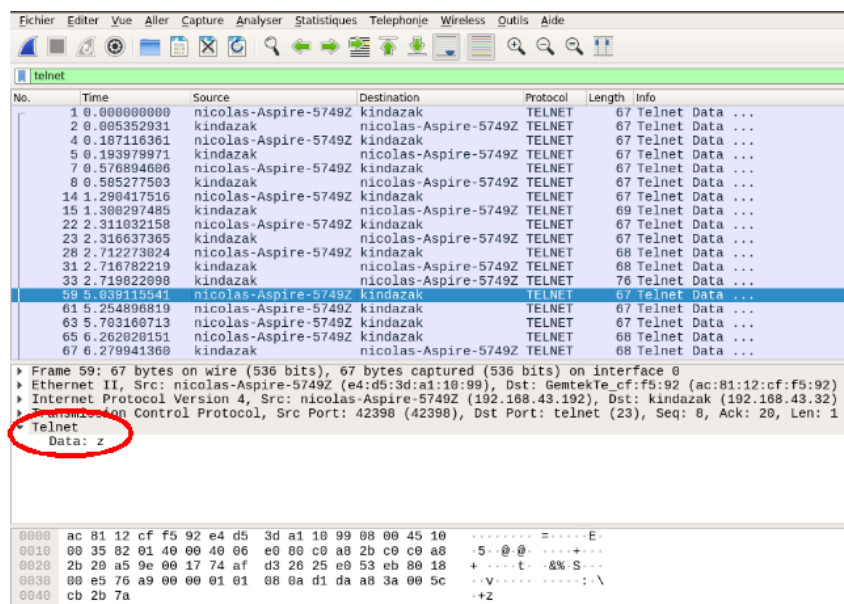


FIGURE 21 – Datagramme contenant la première lettre du mot de passe

Cet datagramme contient la lettre **Z** qui est la première lettre. La figure 22 présente la récupération de la deuxième lettre.

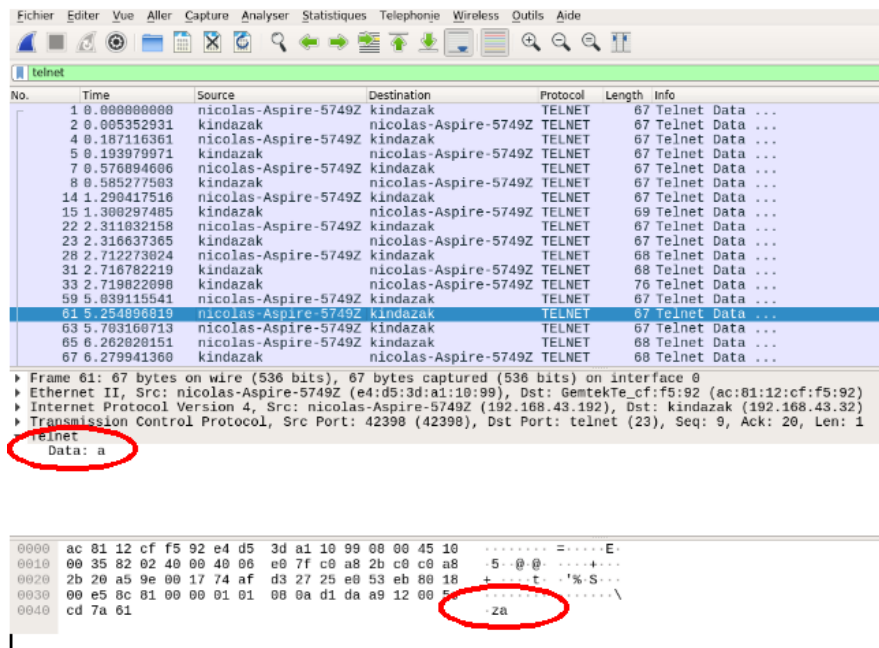


FIGURE 22 – Datagramme contenant la deuxième lettre du mot de passe

Nous récupérons la dernière lettre du mot de passe qui est présenté par la figure 23 ci-dessous.

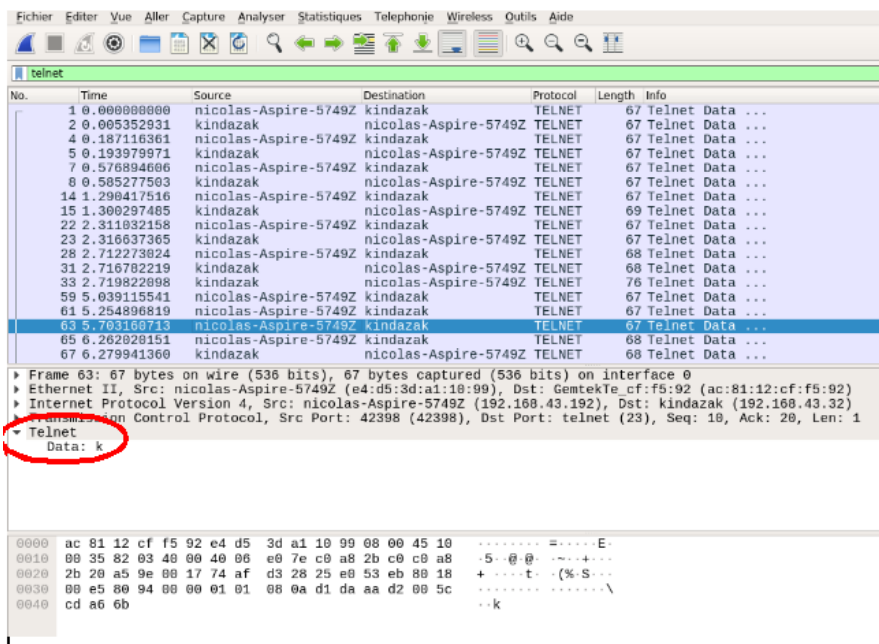


FIGURE 23 – Datagramme contenant la troisième lettre du mot de passe

En somme nous pouvons dire que les datagrammes contiennent des informations. A partir des datagrammes récupérés lors de la connexion nous avons récupéré le login **zak** et le mot de passe **zak** à travers l'analyse des datagrammes.

4 Analyse du protocole TCP

Connexion entre les machines : Pour analyser les paquets échangés durant l'utilisation du protocole TCP, nous avons entré dans un terminal la commande « `tcpdump -i wlan1 port http` » afin de n'écouter que l'interface wifi. Dans un second terminal nous avons tapé la commande "`wget http ://fad.ifi.edu.vn/iffad/file.php/28/documents/WS_user-guide-a4.pdf`" afin de télécharger le fichier "TP1_outilsReseaux.pdf".

```
03:03:48.342341 IP SmeudOpenSource-PC.39736 > 112.137.140.42.http: Flags [F.], seq 434, ack 11736, win 421, options [nop,nop,TS val 1348916427 ecr 52543488], length 0
03:03:48.346007 IP 112.137.140.42.http > SmeudOpenSource-PC.39736: Flags [F.], seq 11736, ack 435, win 62, options [nop,nop,TS val 52543490 ecr 1348916427], length 0
03:03:48.346059 IP SmeudOpenSource-PC.39736 > 112.137.140.42.http: Flags [.], ack 11737, win 421, options [nop,nop,TS val 1348916431 ecr 52543490], length 0
```

FIGURE 24 – Séquence de connexion de nos deux machines via TCP (*tcpdump*)

Selon la figure 24 ci-dessus nous pouvons observer les différents échanges de nos deux machines. Dans un premier temps notre machine envoie un paquet syn au serveur, qui lui le répond syn+ack pour enfin notre machine répond au serveur par un paquet ack.

No.	Time	Source	Destination	Protocol	Length	Info
62	23.141892975	one-CX61-2QF	fad.ifi.edu.vn	TCP	74	37810 → http(80) [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK PER...
63	23.267351419	fad.ifi.edu.vn	one-CX61-2QF	TCP	74	http(80) → 37810 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=140...
64	23.267387614	one-CX61-2QF	fad.ifi.edu.vn	TCP	66	37810 → http(80) [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=39875...
65	23.267472276	one-CX61-2QF	fad.ifi.edu.vn	HTTP	256	GET /iffad/file.php/28/documents/WS_user-guide-a4.pdf HTTP/1.1...
66	23.471769961	fad.ifi.edu.vn	one-CX61-2QF	TCP	66	http(80) → 37810 [ACK] Seq=1 Ack=191 Win=6912 Len=0 TSval=5275...
67	23.774195826	fad.ifi.edu.vn	one-CX61-2QF	HTTP	822	HTTP/1.1 303 See Other (text/html)
68	23.774225098	one-CX61-2QF	fad.ifi.edu.vn	TCP	66	37810 → http(80) [ACK] Seq=191 Ack=757 Win=30720 Len=0 TSval=3...
69	23.774451939	one-CX61-2QF	fad.ifi.edu.vn	HTTP	309	GET /iffad/login/index.php HTTP/1.1
70	23.813375888	fad.ifi.edu.vn	one-CX61-2QF	TCP	66	http(80) → 37810 [ACK] Seq=757 Ack=434 Win=7936 Len=0 TSval=52...
71	24.009995483	fad.ifi.edu.vn	one-CX61-2QF	TCP	1454	[TCP segment of a reassembled PDU]
72	24.013384072	fad.ifi.edu.vn	one-CX61-2QF	TCP	1454	[TCP segment of a reassembled PDU]
73	24.013406146	one-CX61-2QF	fad.ifi.edu.vn	TCP	66	37810 → http(80) [ACK] Seq=434 Ack=3533 Win=36608 Len=0 TSval=...
74	24.031123371	fad.ifi.edu.vn	one-CX61-2QF	TCP	1454	[TCP segment of a reassembled PDU]
75	24.069441182	one-CX61-2QF	fad.ifi.edu.vn	TCP	66	37810 → http(80) [ACK] Seq=434 Ack=4921 Win=39424 Len=0 TSval=...
76	24.114662914	fad.ifi.edu.vn	one-CX61-2QF	TCP	1454	[TCP segment of a reassembled PDU]
77	24.114682461	one-CX61-2QF	fad.ifi.edu.vn	TCP	66	37810 → http(80) [ACK] Seq=434 Ack=6309 Win=42368 Len=0 TSval=...
78	24.115283056	fad.ifi.edu.vn	one-CX61-2QF	TCP	1454	[TCP segment of a reassembled PDU]
79	24.115290497	one-CX61-2QF	fad.ifi.edu.vn	TCP	66	37810 → http(80) [ACK] Seq=434 Ack=7697 Win=45312 Len=0 TSval=...
80	24.132063531	fad.ifi.edu.vn	one-CX61-2QF	TCP	1454	[TCP segment of a reassembled PDU]
81	24.132102628	one-CX61-2QF	fad.ifi.edu.vn	TCP	66	37810 → http(80) [ACK] Seq=434 Ack=9085 Win=48128 Len=0 TSval=...

FIGURE 25 – Séquence de connexion de nos deux machines via TCP (*wireshark*)

5 Conclusion

A travers cet TP, nous avons appris à manipuler de différentes commandes linux pour la mise en place d'un réseau informatique. En plus de ces commandes nous avons appris à utiliser les outils d'analyse du réseau que sont « mtr », « wireshark » et « tcpdump » qui nous ont permis de comprendre le fonctionnement et le rôle de certains protocoles tels que : ARP, TCP et Telnet. Enfin nous remarquons que l'utilisation de Telnet expose les données transférées. Enfin ce TP nous a permis de mieux comprendre la configuration des cartes réseaux et de faire des analyse sur les résultats de ces protocoles de certains protocoles.