

**École Polytechnique de Montréal**  
**Département de génie informatique**

**INF8402 - Sécurité des réseaux fixes et mobiles**  
**Automne 2015**

**Travail Pratique N°1 : WIRESHARK COLLECTE ET ANALYSE D'INFORMATION**  
**Equipe No 16**

**Informations générales**

<b>Public cible</b>	Étudiants de 2 <sup>e</sup> et 3 <sup>e</sup> cycle de génie informatique
<b>Titre du cours</b>	<b>INF8402 - Sécurité des réseaux fixes et mobiles</b>
<b>Session</b>	Automne 2015
<b>Date et lieu de réalisation</b>	Laboratoire de réseautique (L-4708)
<b>Taille de chaque équipe</b>	4 étudiants
<b>Chargé du laboratoire</b>	(Mauricio.Mendoza---Medellin@polymtl.ca)
<b>Étudiants</b>	Ivan Christopher Koupa Lendouba : 1394313 Sara Farshadfar : 1760128 Mohamad Nour Tamer : 1755044

## 1.4 Rappel et description avec ipconfig /all

```
Microsoft Windows [version 6.3.9600]
(c) 2013 Microsoft Corporation. Tous droits réservés.

C:\Windows\System32>ipconfig/all

Configuration IP de Windows

    Nom de l'hôte . . . . . : L4708-12
    Suffixe DNS principal . . . . . : gigl.polymtl.ca
    Type de noeud . . . . . : Hybride
    Routage IP activé . . . . . : Non
    Proxy WINS activé . . . . . : Non
    Liste de recherche du suffixe DNS.: gigl.polymtl.ca
                                           lerb.polymtl.ca

Carte Ethernet Ethernet 5 :

    Suffixe DNS propre à la connexion. . . :
    Description. . . . . : Intel(R) PRO/1000 GT Desktop Adapter

    Adresse physique . . . . . : 90-E2-BA-53-DC-1D
    DHCP activé. . . . . : Oui
    Configuration automatique activée. . . : Oui
    Adresse IPv6 de liaison locale. . . . : fe80::952a:a079:c461:f8e9%21<préféré>
)
    Adresse IPv4. . . . . : 192.168.44.69<préféré>
    Masque de sous-réseau. . . . . : 255.255.255.0
    Bail obtenu. . . . . : 11 septembre 2015 12:27:47
    Bail expirant. . . . . : 11 septembre 2015 16:28:16
    Passerelle par défaut. . . . . :
    Serveur DHCP . . . . . : 192.168.44.198
    IAID DHCPv6 . . . . . : 361816762
    DUID de client DHCPv6. . . . . : 00-01-00-01-1B-59-9E-11-00-22-4D-9E-51
-80
    Serveurs DNS. . . . . : fec0:0:0:ffff::1%1
                                           fec0:0:0:ffff::2%1
                                           fec0:0:0:ffff::3%1
    NetBIOS sur Tcpip. . . . . : Activé

Carte Ethernet Ethernet 3 :

    Suffixe DNS propre à la connexion. . . : lerb.polymtl.ca
    Description. . . . . : Intel(R) Ethernet Connection I217-U
    Adresse physique . . . . . : 08-62-66-4C-81-C6
    DHCP activé. . . . . : Oui
    Configuration automatique activée. . . : Oui
    Adresse IPv6 de liaison locale. . . . : fe80::f126:9e3c:ea25:e33%19<préféré>

    Adresse IPv4. . . . . : 132.207.29.112<préféré>
    Masque de sous-réseau. . . . . : 255.255.255.0
    Bail obtenu. . . . . : 11 septembre 2015 12:27:29
    Bail expirant. . . . . : 12 septembre 2015 12:28:16
    Passerelle par défaut. . . . . : 132.207.29.1
    Serveur DHCP . . . . . : 132.207.29.7
    IAID DHCPv6 . . . . . : 333463369
    DUID de client DHCPv6. . . . . : 00-01-00-01-1B-59-9E-11-00-22-4D-9E-51
-80
    Serveurs DNS. . . . . : 132.207.185.70
                                           132.207.29.2
                                           132.207.144.2
    NetBIOS sur Tcpip. . . . . : Activé
```

- Hostname : L4708-12
- Nom de suffixe DNS : gigl.polymtl.ca
- Nombre d'interface réseau : 15

## Carte Intel I217-V :

```
Carte Ethernet Ethernet 3 :
Suffixe DNS propre à la connexion. . . : lerb.polymtl.ca
Description. . . . . : Intel(R) Ethernet Connection I217-V
Adresse physique . . . . . : 08-62-66-4C-81-C6
DHCP activé. . . . . : Oui
Configuration automatique activée. . . : Oui
Adresse IPv6 de liaison locale. . . . : fe80::f126:9e3c:ea25:e33%19<préféré>

Adresse IPv4. . . . . : 132.207.29.112<préféré>
Masque de sous-réseau. . . . . : 255.255.255.0
Bail obtenu. . . . . : 11 septembre 2015 12:27:29
Bail expirant. . . . . : 12 septembre 2015 12:28:16
Passerelle par défaut. . . . . : 132.207.29.1
Serveur DHCP . . . . . : 132.207.29.7
IAID DHCPv6 . . . . . : 333463369
DUID de client DHCPv6. . . . . : 00-01-00-01-1B-59-9E-11-00-22-4D-9E-51
-80
Serveurs DNS. . . . . : 132.207.185.70
                        132.207.29.2
                        132.207.144.2
NetBIOS sur Tcpip. . . . . : Activé
```

- **Adresse physique MAC:** 08-62-66-4C-81-C6  
Il s'agit de la compagnie ASUSTek COMPUTER INC.
- **Adresse IPv4 :**132.207.29.112
- **Masque réseau :** 255.255.255.0  
Cette adresse a été obtenue par le serveur DHCP
- **Adresse IPv6 :** fe80::f126:9e3c:ea25:e33
- **Serveur DHCP :** 132.207.29.7
- **Serveur DNS :** 132.207.185.70  
132.207.29.2  
132.207.144.2
- **Server Wins :** n'est pas spécifié

Interfaces virtuelles pour VMware " Le tunnel 6to4 adapter "

```

Carte Tunnel 6T04 Adapter :
  Suffixe DNS propre à la connexion. . . : lerb.polymtl.ca
  Description. . . . . : Microsoft 6to4 Adapter
  Adresse physique . . . . . : 00-00-00-00-00-00-E0
  DHCP activé. . . . . : Non
  Configuration automatique activée. . . : Oui
  Adresse IPv6. . . . . : 2002:84cf:1d70::84cf:1d70<préféré>
  Passerelle par défaut. . . . . :
  IAID DHCPv6 . . . . . : 150994944
  DUID de client DHCPv6. . . . . : 00-01-00-01-1B-59-9E-11-00-22-4D-9E-51
-80
  Serveurs DNS. . . . . : 132.207.185.70
                           132.207.29.2
                           132.207.144.2
  NetBIOS sur TCP/IP. . . . . : Désactivé

```

- Adresse du tunnel 6to4 adapter : 2002 :84cf :1d70 ::84cf :1d70

## 1.5 Partie A Général

- Adresse de Windows 7 : 192.168.68.129
- Adresse de Bitnami : 192.168.68.130
- Adresse de Kali : 192.168.68.131

Pinging the Windows machine has failed because of the firewall, so we can make sure that Kali machine is on the same network by running the arp command

```

root@kali:~# arp -a
? (192.168.68.254) at 00:50:56:f9:40:95 [ether] on eth0
? (192.168.68.130) at 00:0c:29:a3:45:e5 [ether] on eth0
? (192.168.68.129) at 00:0c:29:bd:95:34 [ether] on eth0
? (192.168.68.2) at 00:50:56:eb:2c:9e [ether] on eth0
root@kali:~#

```

Runing the command « sudo ufw disable » on Bitnami

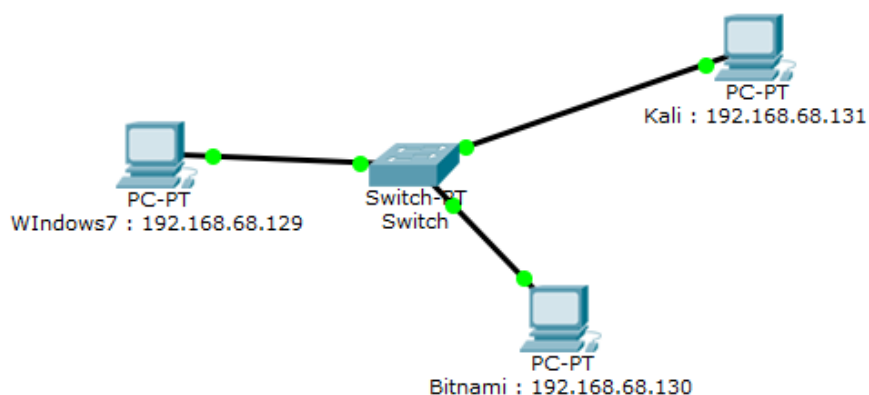
```

linux login: bitnami
Password:
Last login: Thu Sep 10 20:04:45 UTC 2015 on tty1
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.13.0-55-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
bitnami@linux:~$ sudo
sudo          sudoedit      sudoreplay
bitnami@linux:~$ sudo
sudo          sudoedit      sudoreplay
bitnami@linux:~$ sudo ufw disable
[sudo] password for bitnami:
Firewall stopped and disabled on system startup
bitnami@linux:~$ _

```

## Une graphique de la configuration des équipes



## 1.6 Partie B TCP/UDP

1- Ping avec l'option -t (ping en continu) au Bitnami

```
C:\Users\Administrator>ping 192.168.68.130 -t

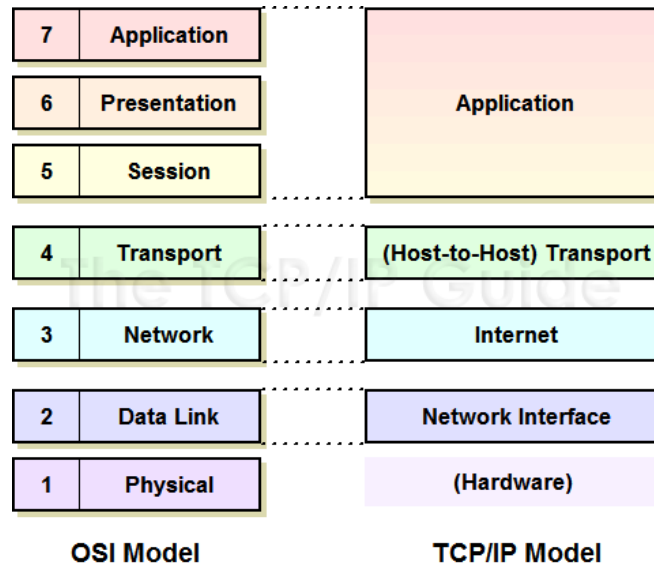
Pinging 192.168.68.130 with 32 bytes of data:
Reply from 192.168.68.130: bytes=32 time<1ms TTL=64
Reply from 192.168.68.130: bytes=32 time<1ms TTL=64
Reply from 192.168.68.130: bytes=32 time<1ms TTL=64
Reply from 192.168.68.130: bytes=32 time<1ms TTL=64
Reply from 192.168.68.130: bytes=32 time<1ms TTL=64
Reply from 192.168.68.130: bytes=32 time<1ms TTL=64
Reply from 192.168.68.130: bytes=32 time<1ms TTL=64
Reply from 192.168.68.130: bytes=32 time<1ms TTL=64
Reply from 192.168.68.130: bytes=32 time<1ms TTL=64
Reply from 192.168.68.130: bytes=32 time<1ms TTL=64
```

2- Les trames ICMP (Ping).

1	0.000000000	192.168.68.129	192.168.68.130	ICMP	74	Echo (ping)
2	0.000099000	192.168.68.130	192.168.68.129	ICMP	74	Echo (ping)
5	1.000736000	192.168.68.129	192.168.68.130	ICMP	74	Echo (ping)
6	1.000817000	192.168.68.130	192.168.68.129	ICMP	74	Echo (ping)
7	1.999816000	192.168.68.129	192.168.68.130	ICMP	74	Echo (ping)
8	1.999900000	192.168.68.130	192.168.68.129	ICMP	74	Echo (ping)
9	3.000208000	192.168.68.129	192.168.68.130	ICMP	74	Echo (ping)
10	3.000215000	192.168.68.130	192.168.68.129	ICMP	74	Echo (ping)
11	4.000786000	192.168.68.129	192.168.68.130	ICMP	74	Echo (ping)

3- If you're trying to capture network traffic that's *not* being sent to or from the machine running Wireshark, i.e. traffic between “ Windows & Bitnami ) you will have to capture in "promiscuous mode", and, on a switched Ethernet network, Wireshark provide us with this feature.

#### 4- Couches du modèle OSI



As you can see, there are three OSI layers showing (Layer 2 & 3 & 4)

```

▶ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▶ Ethernet II, Src: Vmware_bd:95:34 (00:0c:29:bd:95:34), Dst: Vmware_a3:45:e5 (00:0c:29:a3:45:e5)
▶ Internet Protocol Version 4, Src: 192.168.68.129 (192.168.68.129), Dst: 192.168.68.130 (192.168.68.130)
▶ Internet Control Message Protocol

```

#### 5- Les champs de l'entête Ethernet

```

▼ Ethernet II, Src: Vmware_a3:45:e5 (00:0c:29:a3:45:e5), Dst: Vmware_bd:95:34 (00:0c:29:bd:95:34)
  ▼ Destination: Vmware_bd:95:34 (00:0c:29:bd:95:34)
    Address: Vmware_bd:95:34 (00:0c:29:bd:95:34)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0 .... = IG bit: Individual address (unicast)
  ▼ Source: Vmware_a3:45:e5 (00:0c:29:a3:45:e5)
    Address: Vmware_a3:45:e5 (00:0c:29:a3:45:e5)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0 .... = IG bit: Individual address (unicast)
  Type: IP (0x0800)

```

It's the same manufacturer (Vmware).

6- recherche pour **00-0c-0c**, à qui appartient ce OUI

00-0C-0C	(hex)	APPRO TECHNOLOGY INC.
000C0C	(base 16)	APPRO TECHNOLOGY INC.
		13F, No. 66 Chung-Cheng Rd,
		Hsin-Chuang Taipei 242
		TW

By looking for the **00-0c-0c** it seems that is not Cisco related OUI & what has been found for cisco is the below screen shot

F4-CF-E2	(hex)	Cisco Systems, Inc
F4CFE2	(base 16)	Cisco Systems, Inc
		170 West Tasman Drive
		San Jose CA 95134
		US
50-1C-BF	(hex)	Cisco Systems, Inc
501CBF	(base 16)	Cisco Systems, Inc
		170 West Tasman Drive
		San Jose CA 95134
		US

7- Filtrer les paquets pour icmp seulement et vérifier les valeurs des champs type pour les trames request et reply.

As you can see from the below screen shots the ICMP value of 1 for both the request & reply packet

No.	Time	Source	Destination	Protocol
6	1.000817000	192.168.68.130	192.168.68.129	ICMP
8	1.999900000	192.168.68.130	192.168.68.129	ICMP
10	3.000215000	192.168.68.130	192.168.68.129	ICMP
12	4.000860000	192.168.68.130	192.168.68.129	ICMP
14	5.000610000	192.168.68.130	192.168.68.129	ICMP

► Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: No

Total Length: 60

Identification: 0x8b8d (35725)

► Flags: 0x00

Fragment offset: 0

Time to live: 64

Protocol: **ICMP (1)**

► Header checksum: 0xe4df [validation disabled]

No.	Time	Source	Destination	Protocol
1	0.000000000	192.168.68.129	192.168.68.130	ICMP
5	1.000736000	192.168.68.129	192.168.68.130	ICMP
7	1.999816000	192.168.68.129	192.168.68.130	ICMP
9	3.000208000	192.168.68.129	192.168.68.130	ICMP
11	4.000786000	192.168.68.129	192.168.68.130	ICMP

Header Length: 20 bytes

- Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not ECN Capable)
- Total Length: 60
- Identification: 0x0ddc (3548)
- Flags: 0x00
- Fragment offset: 0
- Time to live: 128
- Protocol: **ICMP (1)**
- Header checksum: 0x2291 [validation disabled]

## 8- Packets type

	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.68.129	192.168.68.130	ICMP	74	Echo (ping) request
5	1.000736000	192.168.68.129	192.168.68.130	ICMP	74	Echo (ping) request
7	1.999816000	192.168.68.129	192.168.68.130	ICMP	74	Echo (ping) request
9	3.000208000	192.168.68.129	192.168.68.130	ICMP	74	Echo (ping) request
11	4.000786000	192.168.68.129	192.168.68.130	ICMP	74	Echo (ping) request

[Destination GeoIP: Unknown]

▼ Internet Control Message Protocol

Type: **8 (Echo (ping) request)**

	Time	Source	Destination	Protocol	Length	Info
12	4.000000000	192.168.68.130	192.168.68.129	ICMP	74	Echo (ping) reply
14	5.000610000	192.168.68.130	192.168.68.129	ICMP	74	Echo (ping) reply
17	6.000549000	192.168.68.130	192.168.68.129	ICMP	74	Echo (ping) reply
19	7.000071000	192.168.68.130	192.168.68.129	ICMP	74	Echo (ping) reply
21	8.000708000	192.168.68.130	192.168.68.129	ICMP	74	Echo (ping) reply
23	9.000494000	192.168.68.130	192.168.68.129	ICMP	74	Echo (ping) reply

▼ Ethernet II, Src: Vmware\_a3:45:e5 (00:0c:29:a3:45:e5), Dst: Vmware\_bd:95:34 (00:0c:29:bd:95:34)

▼ Internet Protocol Version 4, Src: 192.168.68.130 (192.168.68.130), Dst: 192.168.68.129 (192.168.68.129)

▼ Internet Control Message Protocol

Type: **0 (Echo (ping) reply)**



## 9- Filtrez sur adresse ip

Filter: <input type="text" value="ip.addr==192.168.68.131"/> Expression... Clear Apply Save						
	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.68.131	205.251.242.54	TCP	54	42062→80 [ACK] Seq=1
2	0.000192000	205.251.242.54	192.168.68.131	TCP	60	[TCP ACKed unseen seq
3	0.032018000	192.168.68.131	173.194.123.84	TCP	54	55873→80 [ACK] Seq=1
4	0.032072000	173.194.123.84	192.168.68.131	TCP	60	[TCP ACKed unseen seq
5	0.224148000	192.168.68.131	23.9.111.240	TCP	54	58123→80 [ACK] Seq=1

## 10-Filtrez l'adresse de toyota.jp

Filter: <input type="text" value="ip.addr==52.69.82.147"/> Expression... Clear Apply Save						
	Time	Source	Destination	Protocol	Length	Info
23	9.347492000	192.168.68.131	52.69.82.147	TCP	74	46538→80 [SYN] Seq=0
24	9.514605000	52.69.82.147	192.168.68.131	TCP	60	80→46538 [SYN, ACK] S
25	9.514648000	192.168.68.131	52.69.82.147	TCP	54	46538→80 [ACK] Seq=1
26	9.514837000	192.168.68.131	52.69.82.147	HTTP	354	GET / HTTP/1.1
27	9.514960000	52.69.82.147	192.168.68.131	TCP	60	80→46538 [ACK] Seq=1
28	9.682008000	52.69.82.147	192.168.68.131	HTTP	485	HTTP/1.1 301 Moved Per
29	9.682023000	192.168.68.131	52.69.82.147	TCP	54	46538→80 [ACK] Seq=30
69	19.680000000	192.168.68.131	52.69.82.147	TCP	54	[TCP Keep-Alive] 46538
70	19.680147000	52.69.82.147	192.168.68.131	TCP	60	[TCP Keep-Alive ACK] 8

## 11-Identifiez le premier 3 paquets connexion.( As you can see packets 23,24,25)

By looking at those three packets, you will see the TCP 3 way handshakes (SYN, SYN ACK, ACK).

23	9.347492000	192.168.68.131	52.69.82.147	TCP	74	46538→80 [SYN] Seq=0
24	9.514605000	52.69.82.147	192.168.68.131	TCP	60	80→46538 [SYN, ACK] S
25	9.514648000	192.168.68.131	52.69.82.147	TCP	54	46538→80 [ACK] Seq=1
26	9.514837000	192.168.68.131	52.69.82.147	HTTP	354	GET / HTTP/1.1
27	9.514960000	52.69.82.147	192.168.68.131	TCP	60	80→46538 [ACK] Seq=1
28	9.682008000	52.69.82.147	192.168.68.131	HTTP	485	HTTP/1.1 301 Moved Pe
29	9.682023000	192.168.68.131	52.69.82.147	TCP	54	46538→80 [ACK] Seq=30

Total Length: 60

Identification: 0x7403 (29699)

Flags: 0x02 (Don't Fragment)

- 0... .... = Reserved bit: Not set
- .1.. .... = Don't fragment: Set
- ..0. .... = More fragments: Not set

Fragment offset: 0

23	9.347492000	192.168.68.131	52.69.82.147	TCP	74	46538→80 [SYN] Seq=0
24	9.514605000	52.69.82.147	192.168.68.131	TCP	60	80→46538 [SYN, ACK] S
25	9.514648000	192.168.68.131	52.69.82.147	TCP	54	46538→80 [ACK] Seq=1
26	9.514837000	192.168.68.131	52.69.82.147	HTTP	354	GET / HTTP/1.1
27	9.514960000	52.69.82.147	192.168.68.131	TCP	60	80→46538 [ACK] Seq=1
28	9.682008000	52.69.82.147	192.168.68.131	HTTP	485	HTTP/1.1 301 Moved Pe
29	9.682023000	192.168.68.131	52.69.82.147	TCP	54	46538→80 [ACK] Seq=30

Total Length: 44  
 Identification: 0x02c7 (711)  
 ▾ Flags: 0x00  
   0... .... = Reserved bit: Not set  
   .0... .... = Don't fragment: Not set  
   ..0. .... = More fragments: Not set  
 Fragment offset: 0

23	9.347492000	192.168.68.131	52.69.82.147	TCP	74	46538→80 [SYN] Seq=0
24	9.514605000	52.69.82.147	192.168.68.131	TCP	60	80→46538 [SYN, ACK] S
25	9.514648000	192.168.68.131	52.69.82.147	TCP	54	46538→80 [ACK] Seq=1
26	9.514837000	192.168.68.131	52.69.82.147	HTTP	354	GET / HTTP/1.1
27	9.514960000	52.69.82.147	192.168.68.131	TCP	60	80→46538 [ACK] Seq=1
28	9.682008000	52.69.82.147	192.168.68.131	HTTP	485	HTTP/1.1 301 Moved Pe
29	9.682023000	192.168.68.131	52.69.82.147	TCP	54	46538→80 [ACK] Seq=30

Total Length: 40  
 Identification: 0x7404 (29700)  
 ▾ Flags: 0x02 (Don't Fragment)  
   0... .... = Reserved bit: Not set  
   .1... .... = Don't fragment: Set  
   ..0. .... = More fragments: Not set  
 Fragment offset: 0

## 12- TCP vs UDP

TCP is suited for applications that require high reliability, and transmission time is relatively less critical.

UDP is suitable for applications that need fast, efficient transmission, such as games. UDP's stateless nature is also useful for servers that answer small queries from huge numbers of clients.

ethO - Graph Analysis

Time	192.168.68.131	192.168.68.2	Comment
0.000000000	(37856)	Standard que... (53)	DNS: Standard query 0x76ea A redhat.com
0.000736000	(37856)	Standard que... (53)	DNS: Standard query response 0x76ea A 209.132.183.105

Follow UDP Stream

Stream Content

v.....redhat.com.....v.....redhat.com.....i

## 1.7 Partie B TELNET

- 1- Connecting to Bitnami from Windows machine using Telnet with a username/password : bitnami/Bitnami

As you can see that the password has been sent as a clear text which make the Telnet protocol unsecure to be used

```
Follow TCP Stream (tcp.stream eq 0)
Stream Content
.....#...'.....#...'.....P.....'.....ANSI.....!.....!.....!
Ubuntu 14.04.2 LTS
linux login: ...bbiittnaammii
Password: bitnami
Last login: Fri Sep 11 18:29:15 UTC 2015 on tty1
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.13.0-55-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
bitnami@linux:~$ |
```

- 2- As you can see from the below screen shot how Wireshark is able to sniff & interpret the Telnet traffic

28	1.527959000	192.168.68.130	192.168.68.129	TELNET	60	Telnet Data ...
29	1.732865000	192.168.68.130	192.168.68.129	TCP	60	[TCP Keep-Alive] 23->4
30	1.733124000	192.168.68.129	192.168.68.130	TCP	66	49476->23 [ACK] Seq=60
31	1.846888000	192.168.68.129	192.168.68.130	TELNET	60	Telnet Data ...
32	1.847394000	192.168.68.130	192.168.68.129	TELNET	60	Telnet Data ...

▶ Frame 28: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0  
 ▶ Ethernet II, Src: Vmware\_a3:45:e5 (00:0c:29:a3:45:e5), Dst: Vmware\_bd:95:34 (00:0c:29:bd:95:34)  
 ▶ Internet Protocol Version 4, Src: 192.168.68.130 (192.168.68.130), Dst: 192.168.68.129 (192.168.68.129)  
 ▶ Transmission Control Protocol, Src Port: 23 (23), Dst Port: 49476 (49476), Seq: 79, Ack: 60, Len: 60  
 ▼ Telnet  
 Data: b

	Time	Source	Destination	Protocol	Length	Info
27	1.527435000	192.168.68.130	192.168.68.129	TCP	60	23→49476 [ACK] Seq=79
28	1.527959000	192.168.68.130	192.168.68.129	TELNET	60	Telnet Data ...
29	1.732865000	192.168.68.130	192.168.68.129	TCP	60	[TCP Keep-Alive] 23→4
30	1.733124000	192.168.68.129	192.168.68.130	TCP	66	49476→23 [ACK] Seq=60
31	1.846888000	192.168.68.129	192.168.68.130	TELNET	60	Telnet Data ...
32	1.847394000	192.168.68.130	192.168.68.129	TELNET	60	Telnet Data ...

▶ Frame 31: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0  
 ▶ Ethernet II, Src: Vmware\_bd:95:34 (00:0c:29:bd:95:34), Dst: Vmware\_a3:45:e5 (00:0c:29:a3:45:e5)  
 ▶ Internet Protocol Version 4, Src: 192.168.68.129 (192.168.68.129), Dst: 192.168.68.130 (192.168.68.130)  
 ▶ Transmission Control Protocol, Src Port: 49476 (49476), Dst Port: 23 (23), Seq: 60, Ack: 80, Len: 60  
 ▶ Telnet  
 Data: i

	Time	Source	Destination	Protocol	Length	Info
29	1.732865000	192.168.68.130	192.168.68.129	TCP	60	[TCP Keep-Alive] 23→4
30	1.733124000	192.168.68.129	192.168.68.130	TCP	66	49476→23 [ACK] Seq=60
31	1.846888000	192.168.68.129	192.168.68.130	TELNET	60	Telnet Data ...
32	1.847394000	192.168.68.130	192.168.68.129	TELNET	60	Telnet Data ...
33	2.053325000	192.168.68.129	192.168.68.130	TCP	60	49476→23 [ACK] Seq=61
34	2.103258000	192.168.68.129	192.168.68.130	TELNET	60	Telnet Data ...
35	2.103763000	192.168.68.130	192.168.68.129	TELNET	60	Telnet Data ...

▶ Frame 34: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0  
 ▶ Ethernet II, Src: Vmware\_bd:95:34 (00:0c:29:bd:95:34), Dst: Vmware\_a3:45:e5 (00:0c:29:a3:45:e5)  
 ▶ Internet Protocol Version 4, Src: 192.168.68.129 (192.168.68.129), Dst: 192.168.68.130 (192.168.68.130)  
 ▶ Transmission Control Protocol, Src Port: 49476 (49476), Dst Port: 23 (23), Seq: 61, Ack: 81, Len: 60  
 ▶ Telnet  
 Data: t

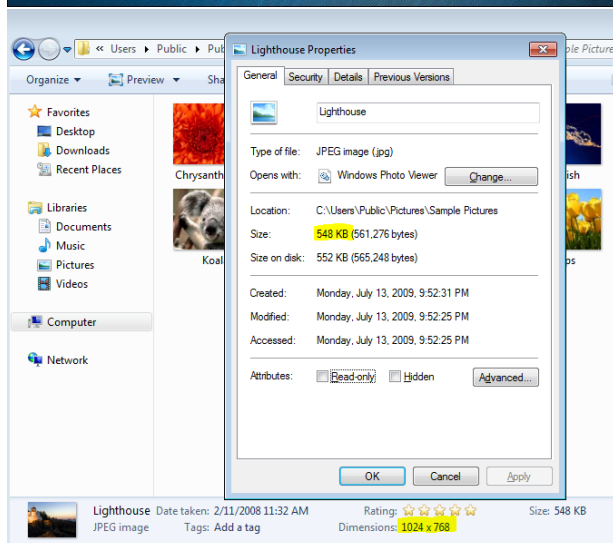
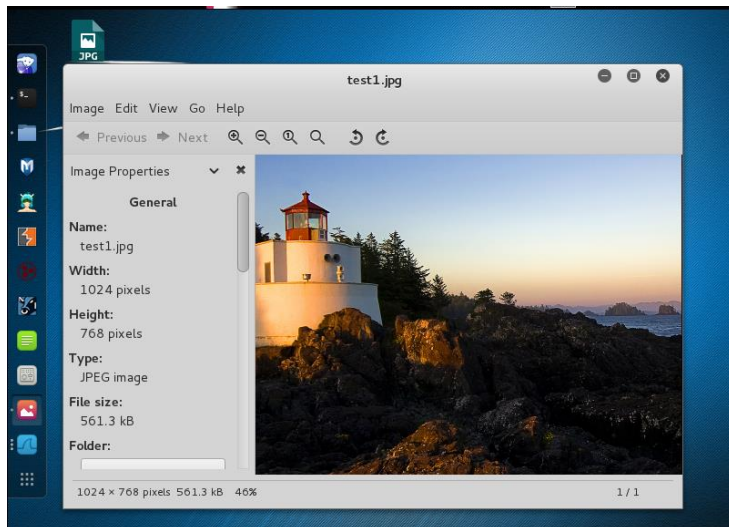
## 1.8 Partie C FTP

Follow TCP Stream (tcp.stream eq 0)	
Stream Content	
220	linux FTP server (Version 6.4/OpenBSD/Linux-ftpd-0.17) ready.
USER	bitnami
331	Password required for bitnami.
PASS	bitnami
230	User bitnami logged in.
PWD	
257	"/home/bitnami" is current directory.

En ce qui a trait au transfert du nom de l'utilisateur et du mot de passe, le protocole FTP et Telnet sont équivalents car nous sommes en mesure de consulter ces données en clair sur le réseau.



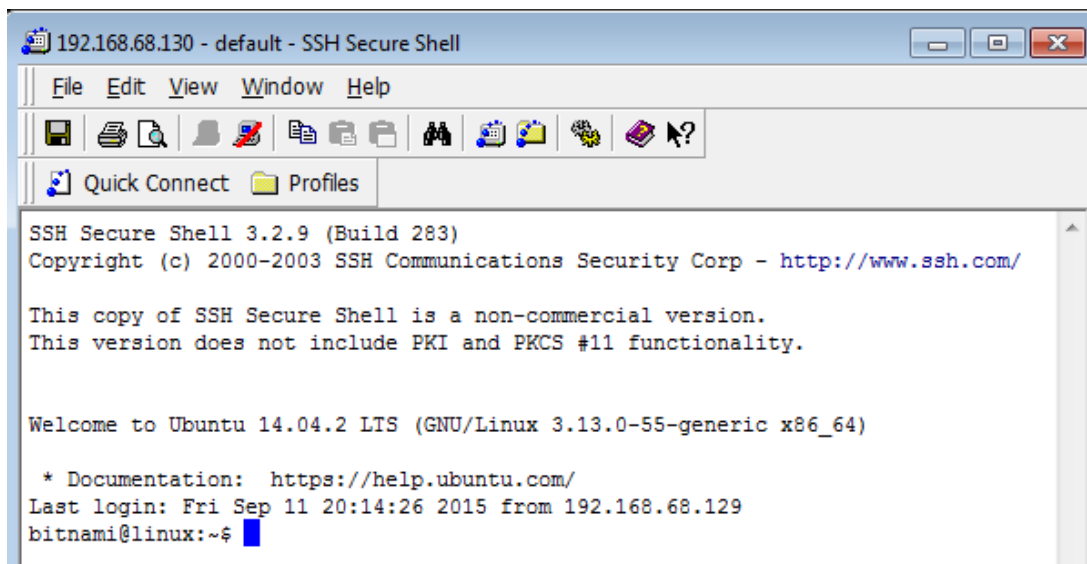
Nous sommes en mesure de consulter l'image transférée par FTP à travers la machine Kali tel que présenté dans les captures d'écran suivantes.



En consultant les propriétés de l'image sur la machine Windows et sur la machine Kali, nous constatons que les attributs sont assez équivalents, donc l'image se voit très peu altérée.

## 1.9 Partie D SSH

- 1- Connecting to Bitnami from Windows machine using SSH with a username/password : bitnami/Bitnami



```
192.168.68.130 - default - SSH Secure Shell
File Edit View Window Help
SSH Secure Shell 3.2.9 (Build 283)
Copyright (c) 2000-2003 SSH Communications Security Corp - http://www.ssh.com/

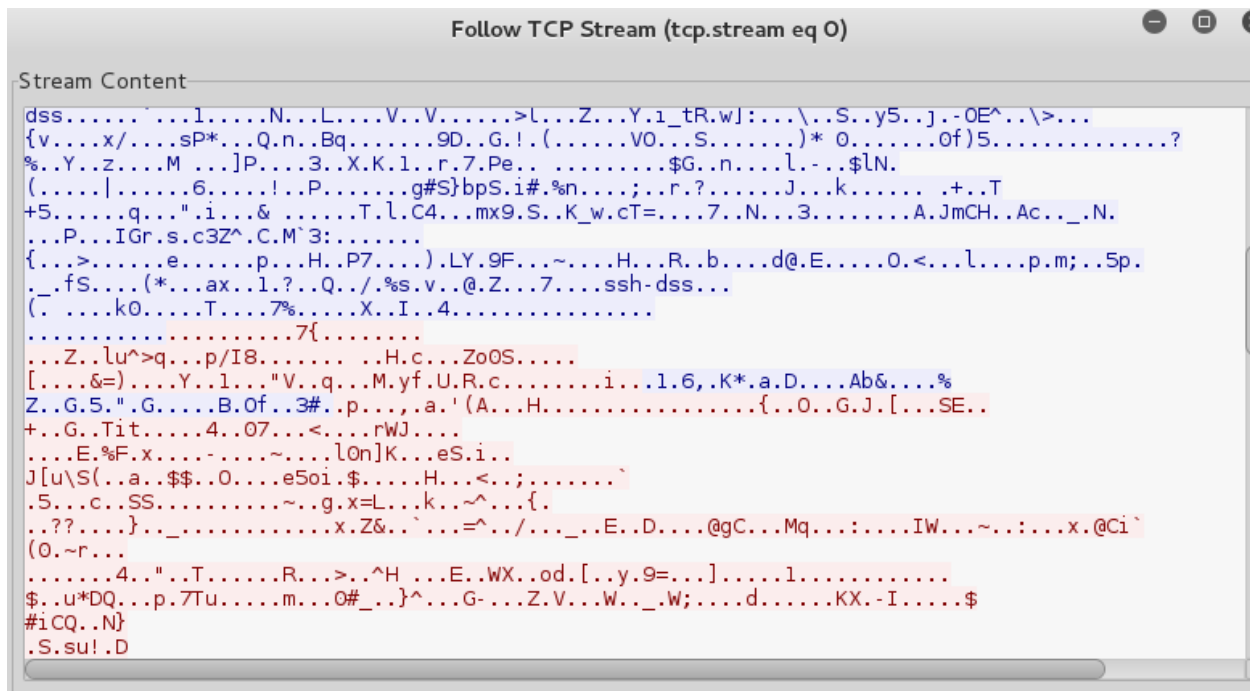
This copy of SSH Secure Shell is a non-commercial version.
This version does not include PKI and PKCS #11 functionality.

Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.13.0-55-generic x86_64)

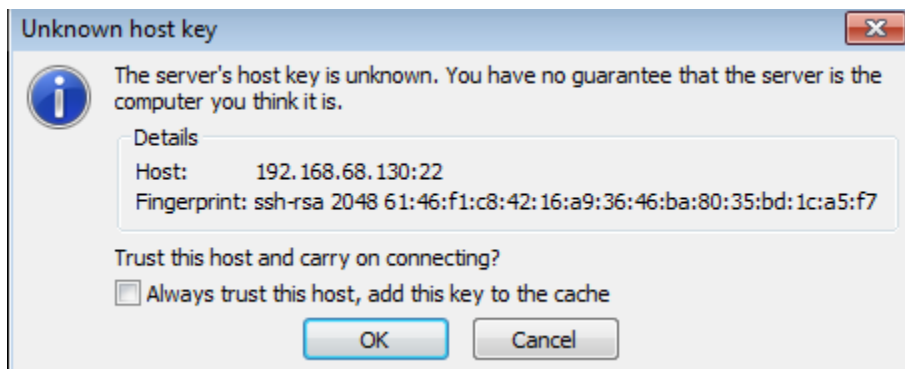
* Documentation: https://help.ubuntu.com/
Last login: Fri Sep 11 20:14:26 2015 from 192.168.68.129
bitnami@linux:~$
```

- 2- SSH is an encrypted way of communication based on encrypting the Sent data using the Public key which has been already send by the destination target which will decrypted the data using his private key associated with the sent public key

Source	Destination	Protocol	Length	Info
192.168.68.130	192.168.68.129	SSHv2	95	Server: Protocol (SSH-2.0-OpenSSH_6.6)
192.168.68.129	192.168.68.130	SSHv2	98	Client: Protocol (SSH-1.99-3.2.9 SSH
192.168.68.130	192.168.68.129	TCP	60	22->49478 [ACK] Seq=42 Ack=45 Win=2924
192.168.68.129	192.168.68.130	SSHv2	390	Client: Ignore, Key Exchange Init
192.168.68.130	192.168.68.129	TCP	60	22->49478 [ACK] Seq=42 Ack=381 Win=302
192.168.68.130	192.168.68.129	TCP	1514	[TCP segment of a reassembled PDU]
192.168.68.130	192.168.68.129	SSHv2	242	Server: Key Exchange Init
192.168.68.129	192.168.68.130	TCP	60	49478->22 [ACK] Seq=381 Ack=1690 Win=6
192.168.68.129	192.168.68.130	SSHv2	214	Client: Ignore, Diffie-Hellman Key Ex
192.168.68.130	192.168.68.129	SSHv2	710	Server: Diffie-Hellman Key Exchange R
192.168.68.129	192.168.68.130	TCP	60	49478->22 [ACK] Seq=541 Ack=2346 Win=6
192.168.68.129	192.168.68.130	SSHv2	86	Client: Ignore, New Keys
192.168.68.130	192.168.68.129	TCP	60	22->49478 [ACK] Seq=2346 Ack=573 Win=3
192.168.68.129	192.168.68.130	SSHv2	134	Client: Encrypted packet (len=80)



## 1.10 Partie D SFTP



Il s'agit de la clé publique du serveur avec laquelle le client va chiffrer les données à envoyer vers celui-ci.



```
SSH-2.0-OpenSSH_6.6.1p1_Ubuntu-2ubuntu2
SSH-2.0-PuTTY_Local:_Nov_21_2010_15:53:55
...|.....F);(.....`.....diffie-hellman-group-exchange-sha256,diffie-hellman-group-exchange-
sha1,diffie-hellman-group14-sha1,diffie-hellman-group1-sha1,rsa2048-sha256,rsa1024-
sha1....ssh-rsa,ssh-dss....aes256-ctr,aes256-cbc,rijndael-cbc@lysator.liu.se,aes192-
ctr,aes192-cbc,aes128-ctr,aes128-cbc,blowfish-ctr,blowfish-cbc,3des-ctr,3des-
cbc,arcfour256,arcfour128....aes256-ctr,aes256-cbc,rijndael-cbc@lysator.liu.se,aes192-
ctr,aes192-cbc,aes128-ctr,aes128-cbc,blowfish-ctr,blowfish-cbc,3des-ctr,3des-
cbc,arcfour256,arcfour128....hmac-sha1,hmac-sha1-96,hmac-md5....hmac-sha1,hmac-sha1-96,hmac-
md5....none,zlib....none,zlib.....L9..p...l
..h..>.....v...X....curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-
sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-group-exchange-sha1,diffie-
hellman-group14-sha1,diffie-hellman-group1-sha1.../ssh-rsa,ssh-dss,ecdsa-sha2-nistp256,ssh-
ed25519....aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-
gcm@openssh.com,aes256-gcm@openssh.com,chacha20-poly1305@openssh.com,aes128-cbc,3des-
cbc,blowfish-cbc,cast128-cbc,aes192-cbc,aes256-cbc,arcfour,rijndael-
cbc@lysator.liu.se....aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-
gcm@openssh.com,aes256-gcm@openssh.com,chacha20-poly1305@openssh.com,aes128-cbc,3des-
cbc,blowfish-cbc,cast128-cbc,aes192-cbc,aes256-cbc,arcfour,rijndael-
cbc@lysator.liu.se....hmac-md5-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-64-
etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-
etm@openssh.com,hmac-ripemd160-etm@openssh.com,hmac-sha1-96-etm@openssh.com,hmac-md5-96-
etm@openssh.com,hmac-md5,hmac-sha1,umac-64@openssh.com,umac-128@openssh.com,hmac-
sha2-256,hmac-sha2-512,hmac-ripemd160,hmac-ripemd160@openssh.com,hmac-sha1-96,hmac-
```

Toute l'information qui circule est encryptée. Mais nous pouvons voir le processus d'échange de clé initialisé afin de rendre la connexion sécurisée.

.Pour analyser la sécurité d'une entreprise en utilisant Wireshark, nous nous assurerons qu'aucune information sensible ne transite par les protocoles FTP et Telnet, mais bien par un dispositif sécurisé tel que SFTP ou SSH.