RAPPORT TP 3 ET TP4

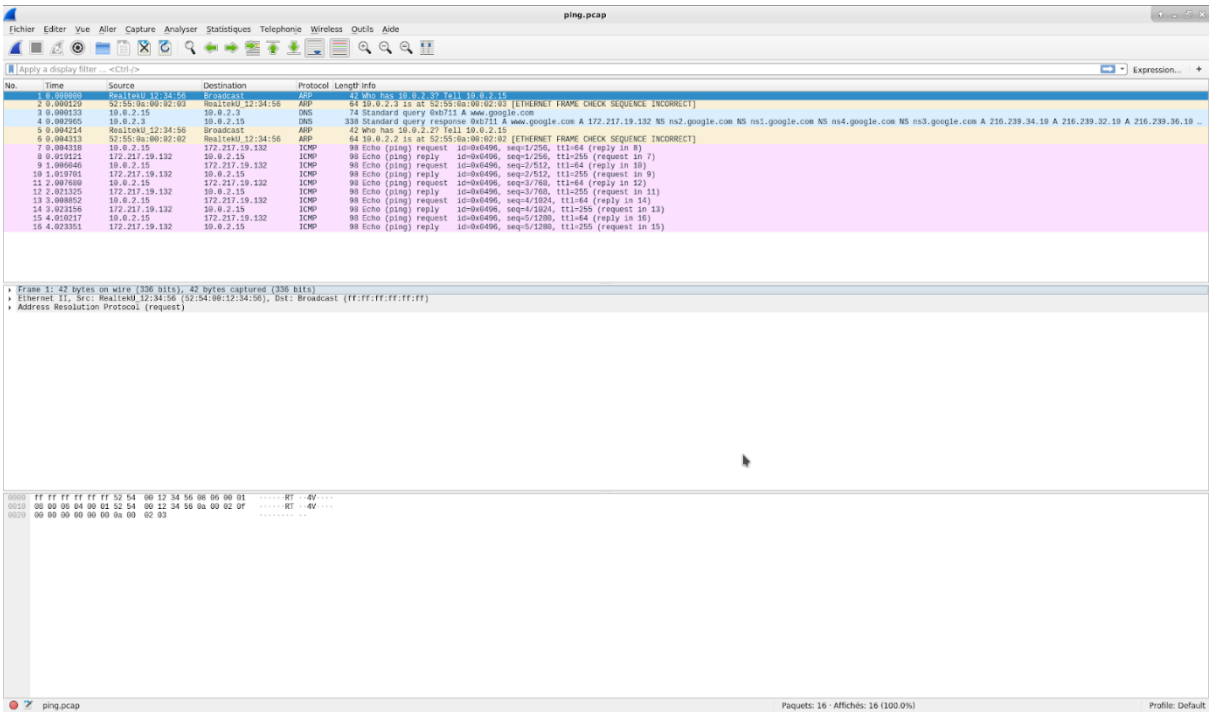
TP 3

1. ANALYSE DE TRAMES AVEC WIRESHARK

1.1. PREAMBULE

Adresse IP de la machine : 10.0.2.15
Masque du réseau : 255.255.255.0
Adresse IP de la passerelle : 10.0.2.2
Adresse du serveur : 1.0.2.3

1.2. PRISE EN MAIN DE WIRESHARK



1.3. PING

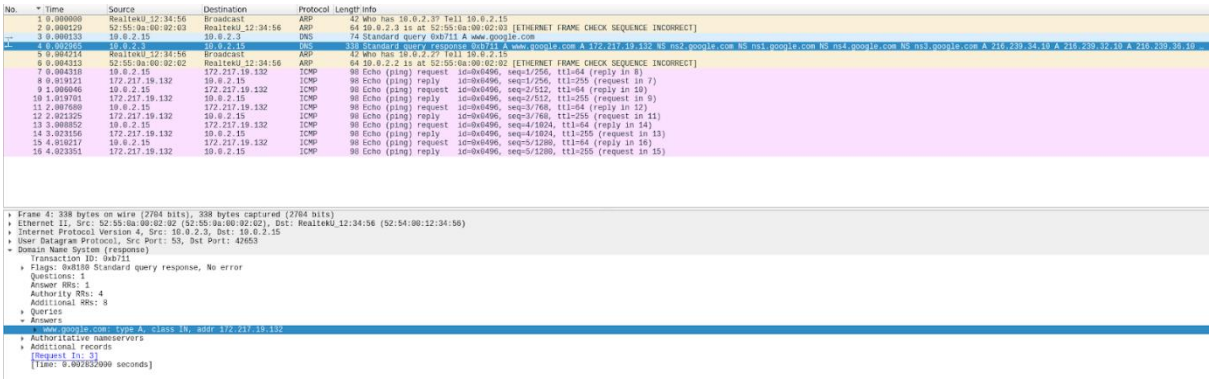
QUELLE ADRESSE ETHERNET EST DESTINEE LA REQUETE ARP (TRAME 1) EMISE PAR LA MACHINE CLIENTE ? IL S'AGIT EN FAIT DE L'ADRESSE DE DIFFUSION (BROADCAST). ELLE NE CORRESPOND A AUCUNE MACHINE PARTICULIERE ! A VOTRE AVIS POURQUOI DOIT ON PROCEDER AINSI ?

Broadcast (ff:ff:ff:ff:ff:ff) demande et récupère l'information de l'adresse source et de destination grâce au protocole ARP.

QUEL EST LE PROTOCOLE DE TRANSPORT UTILISE POUR LES ECHANGES DNS (TRAMES 3-4) ?

C'est le protocole UDP qui est utilisé.

OBSERVEZ EN DETAIL LA REPONSE DNS (SECTION ANSWERS) ET DECOUVREZ AINSI L'ADRESSE IP DE LA MACHINE WWW.GOOGLE.COM RETOURNE PAR LE SERVEUR DNS



L'adresse IP de la machine Google est 172.217.19.132.

LA REQUETE ARP WHO HAS (TRAME 5) CHERCHE A TROUVER L'ADRESSE ETHERNET DE LA MACHINE 10.0.2.2. POURQUOI CETTE MACHINE ET NON PAS LA MACHINE CIBLE WWW.GOOGLE.COM ?

La machine 10.0.2.2 sert de passerelle (je n'en suis pas certain car je n'ai pas trouvé d'informations précises).

VERIFIEZ L'ADRESSE ETHERNET DESTINATION UTILISEE POUR ENVOYER LA TRAME 7.

C'est l'adresse 172.217.19.132.

OBSERVEZ LA PREMIERE REQUETE / REPONSE ICMP (TRAMES 7-8) ET OBSERVEZ LA VALEUR DU CHAMPS TYPE DANS L'EN-TETE ICMP…

Pour la requête :

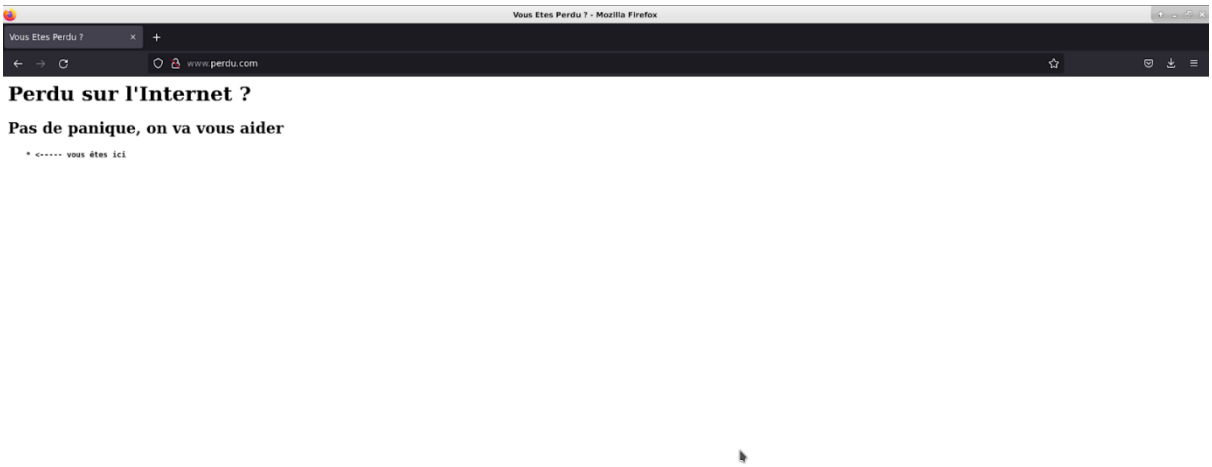
```
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
```

Pour la réponse :

```
Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
```

1.4. UNE PAGE WEB : JE SUIS PERDU !

QUE SE PASSE-T-IL QUAND JE CONSULTE UNE PAGE WEB (PAR EXEMPLE, HTTP://WWW.PERDU.COM) SUR INTERNET AVEC MON NAVIGATEUR PREFERE ?



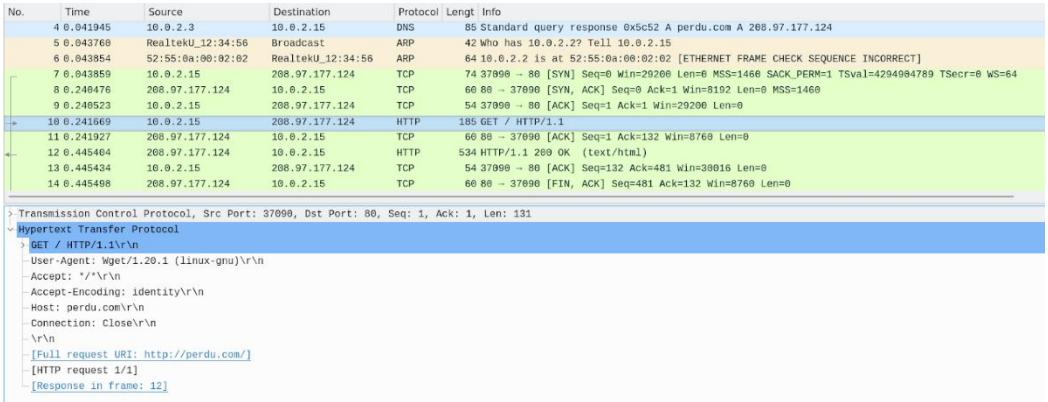
CONSIDERONS LA PREMIERE TRAME TCP QUI OUVRE LA CONNEXION (TRAME 7). TROUVEZ DANS L'EN-TETE TCP LE PORT SOURCE ET LE PORT DE DESTINATION. CE DERNIER EST STANDARD POUR TOUS LES SERVEUR WEB (80). A QUOI CORRESPOND LE FLAG SYN DANS CETTE EN-TETE ?

```
Transmission Control Protocol, Src Port: 37090, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 37090
    Destination Port: 80
```

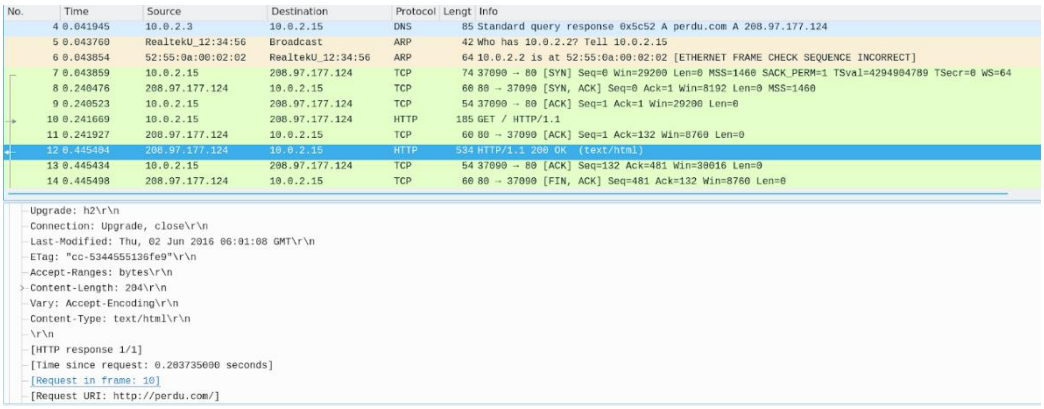
Le port source est 37090 et le port de destination est 80.
Le flag SYN (Synchronisation) permet la connexion entre le port source et le port destination.

IDENTIFIEZ DANS LA CONVERSATION TCP LES TRAMES CORRESPONDANT A LA REQUETE HTTP ET A LA REPONSE HTTP…

La trame 10 correspond à la requête HTTP :



Et comme indiqué sur cette trame, la trame 12 est la réponse à celle-ci :



No.	Time	Source	Destination	Protocol	Lengt	Info
4	0.041945	10.0.2.3	10.0.2.15	DNS	85	Standard query response 0x5c52 A perdu.com A 208.97.177.124
5	0.043760	RealtekU_12:34:56	Broadcast	ARP	42	Who has 10.0.2.2? Tell 10.0.2.15
6	0.043854	52:55:0a:00:02:02	RealtekU_12:34:56	ARP	64	10.0.2.2 is at 52:55:0a:00:02:02 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
7	0.043859	10.0.2.15	208.97.177.124	TCP	74	37090 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=4294904789 TSecr=0 WS=64
8	0.240476	208.97.177.124	10.0.2.15	TCP	60	80 → 37090 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
9	0.240523	10.0.2.15	208.97.177.124	TCP	54	37090 → 80 [ACK] Seq=1 Ack=1 Win=29200 Len=0
10	0.241669	10.0.2.15	208.97.177.124	HTTP	185	GET / HTTP/1.1
11	0.241927	208.97.177.124	10.0.2.15	TCP	60	80 → 37090 [ACK] Seq=1 Ack=132 Win=8760 Len=0
12	0.445404	208.97.177.124	10.0.2.15	HTTP	534	HTTP/1.1 200 OK (text/html)
13	0.445434	10.0.2.15	208.97.177.124	TCP	54	37090 → 80 [ACK] Seq=132 Ack=481 Win=30016 Len=0
14	0.445498	208.97.177.124	10.0.2.15	TCP	60	80 → 37090 [FIN, ACK] Seq=481 Ack=132 Win=8760 Len=0

```
Upgrade: h2\r\n
Connection: Upgrade, close\r\n
Last-Modified: Thu, 02 Jun 2016 06:01:08 GMT\r\n
ETag: "cc-5344555136fe9"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 204\r\n
Vary: Accept-Encoding\r\n
Content-Type: text/html\r\n
\r\n
[HTTP response 1/1]
[Time since request: 0.203735000 seconds]
[Request in frame: 10]
[Request URI: http://perdu.com/]
```

DANS L'EN-TETE DE LA REQUETE HTTP, ON OBSERVE SUR LA PREMIERE LIGNE QU'IL S'AGIT DE LA REQUETE GET / HTTP/1.1. IDENTIFIEZ LE ROLE DES CHAMPS SUIVANTS : USER-AGENT, HOST, CONNECTION.

User-Agent permet aux serveurs d'identifier le système, l'application, la version etc de la machine ayant envoyé la requête.
Host correspond à l'URL et le port du serveur demandé (ici le port est sous-entendu).
Connection : Définit si la connexion avec le serveur reste ouverte ou fermée après l'interaction avec ce même serveur.

```
Hypertext Transfer Protocol
> GET / HTTP/1.1\r\n
  User-Agent: Wget/1.20.1 (linux-gnu)\r\n
  Accept: */*\r\n
  Accept-Encoding: identity\r\n
  Host: perdu.com\r\n
  Connection: Close\r\n
  \r\n
  [Full request URI: http://perdu.com/]
  [HTTP request 1/1]
  [Response in frame: 12]
```

OBSERVEZ MAINTENANT LES DIFFERENTS CHAMPS DANS LA REPONSE HTTP ET EN DEDUIRE LE LOGICIEL SERVEUR, LA LONGUEUR ET LE TYPE DE CONTENU DANS CETTE REPONSE.

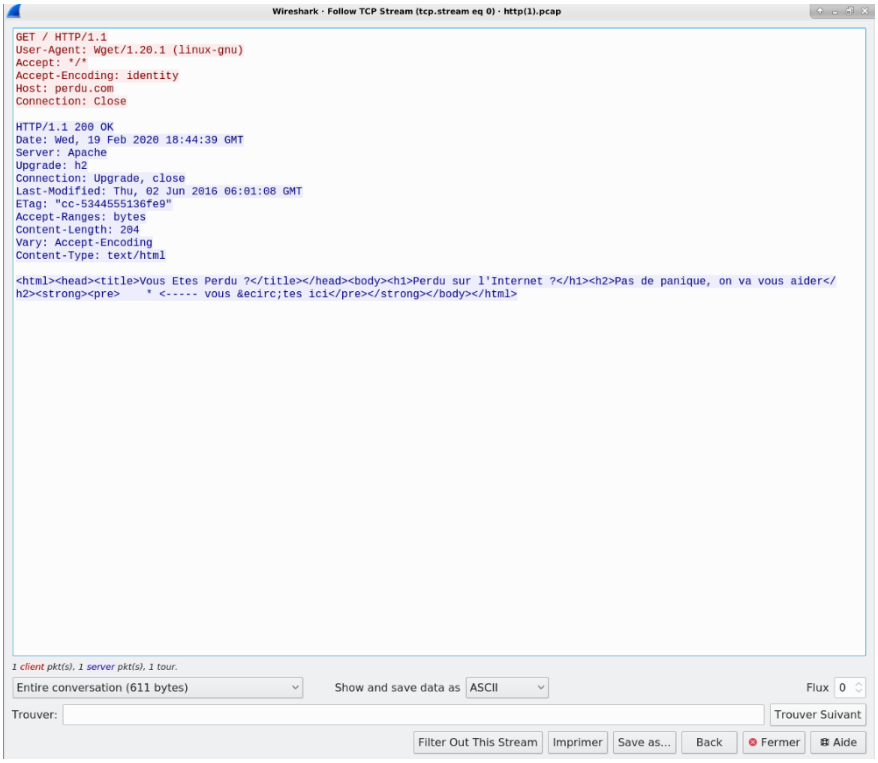
Le logiciel serveur est Apache, la longueur est de 204 bits et le type de contenu est du HTML.

```
Hypertext Transfer Protocol
> HTTP/1.1 200 OK\r\n
  Date: Wed, 19 Feb 2020 18:44:39 GMT\r\n
  Server: Apache\r\n
  Upgrade: h2\r\n
  Connection: Upgrade, close\r\n
  Last-Modified: Thu, 02 Jun 2016 06:01:08 GMT\r\n
  ETag: "cc-5344555136fe9"\r\n
  Accept-Ranges: bytes\r\n
> Content-Length: 204\r\n
  Vary: Accept-Encoding\r\n
  Content-Type: text/html\r\n
  \r\n
```

IMMEDIATEMENT APRES L'EN-TETE HTTP, VOUS POUVEZ IDENTIFIER LE CODE HTML DE LA PAGE WEB : <HTML>...</HTML>

```
Line-based text data: text/html (1 lines)
  <html><head><title>Vous Etes Perdu ?</title></head><body><h1>Perdu sur l'Internet ?</h1><h2>Pas de panique, on va vous aider</h2><strong><pre>    * <----- vous &ecirc;tes ici</pre></strong></…
```

TRAMES 7-16 : POUR LIRE PLUS FACILEMENT LA CONVERSATION TCP, VOUS POUVEZ FAIRE UN "CLIC DROIT" SUR UN DES PAQUETS TCP ET SELECTIONNER SUIVRE (FOLLOW) → FLUX TCP (TCP STREAM) DANS LE MENU DEROULANT. NOTEZ QU'IL EST POSSIBLE DE RECONSTRUIRE PRECISEMENT LE FIL DE LA CONVERSATION GRACE AUX NUMEROS DE SEQUENCE (EN OCTETS) QUI SE TROUVE DANS L'EN-TETE TCP.



TP4

1. MANIPULATION DE PAQUETS AVEC SCAPY

1.1. DEMARRAGE D'UN RESEAU VIRTUEL

LANCEZ TCPDUMP -I ETH0 SUR LA PASSERELLE IMMORTAL AFIN D'ESPIONNER LE TRAFIC ECHANGE ENTRE LES AUTRES MACHINES. LANCEZ NETSTAT -TUPL POUR VOIR QUELS SERVICES (ET DONC QUELS PORTS) SONT OUVERTS SUR OPETH (OU SYL).

```
root@immortal:~# tcpdump -i eth0
[   41.569846] device eth0 entered promiscuous mode
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

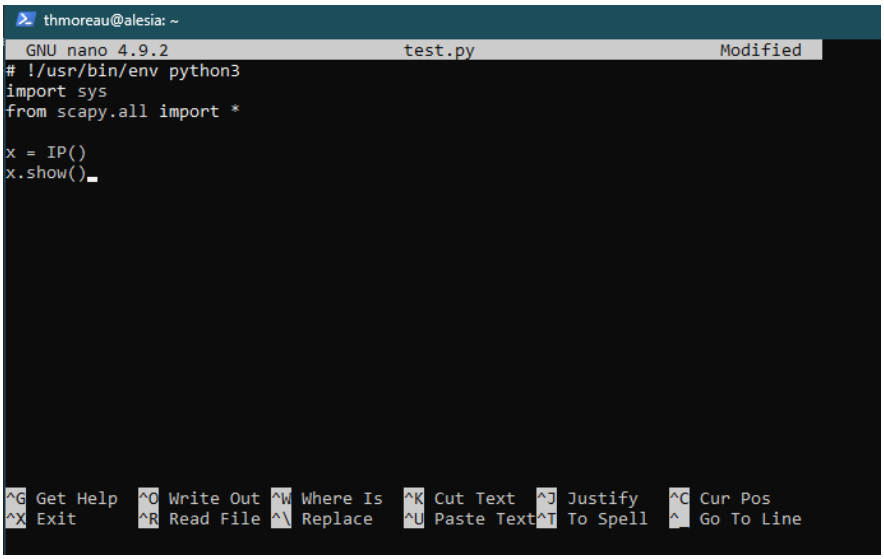
```
root@opeth:~# netstat -tupl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN      363/sshd
tcp        0      0 0.0.0.0:telnet          0.0.0.0:*               LISTEN      277/inetd
tcp        0      0 0.0.0.0:echo            0.0.0.0:*               LISTEN      277/inetd
tcp        0      0 0.0.0.0:daytime         0.0.0.0:*               LISTEN      277/inetd
tcp6       0      0 [::]:http               [::]:*                  LISTEN      377/apache2
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN      363/sshd
udp        0      0 0.0.0.0:echo            0.0.0.0:*                           277/inetd
udp        0      0 0.0.0.0:daytime         0.0.0.0:*                           277/inetd
```

1.2. PRISE EN MAIN DE SCAPY

LANCEZ SCAPY3 SUR LA MACHINE GRAVE EN TANT QUE ROOT (OU SUDOER)

```
>>> x = IP()
>>> x.show()
###[ IP ]###
  version= 4
  ihl= None
  tos= 0x0
  len= None
  id= 1
  flags=
  frag= 0
  ttl= 64
  proto= hopopt
  chksum= None
  src= 127.0.0.1
  dst= 127.0.0.1
  \options\
```

IL EST EGALEMENT POSSIBLE D'ECRIRE DES PROGRAMMES SCAPY SOUS FORME D'UN SCRIPT PYTHON, QU'IL FAUT ENREGISTRER AVEC UN EDITEUR DE TEXTE, COMME NANO OU EMACS




```
root@grave:~# python3 test.py

Bad key "text.kerning_factor" on line 4 in
/usr/share/matplotlib/mpl-data/stylelib/_classic_test_patch.mplstyle.
You probably need to get an updated matplotlibrc file from
http://github.com/matplotlib/matplotlib/blob/master/matplotlibrc.template
or from the matplotlib source distribution
###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = hopopt
  chksum    = None
  src       = 127.0.0.1
  dst       = 127.0.0.1
  \options   \
```

1.3. PING

REGARDEZ DANS LE FICHIER PING.PY UN EXEMPLE D'UTILISATION DE SCAPY QUI ENVOIE UN PING (ICMP) PUIS RECUPERE LA REPONSE. ESSAYEZ-LE EN RECOPIANT LE PROGRAMME LIGNE PAR LIGNE, OU EN FAISANT UN COPIER/COLLER.

```
root@grave:~# python3 ping.py

Bad key "text.kerning_factor" on line 4 in
/usr/share/matplotlib/mpl-data/stylelib/_classic_test_patch.mplstyle.
You probably need to get an updated matplotlibrc file from
http://github.com/matplotlib/matplotlib/blob/master/matplotlibrc.template
or from the matplotlib source distribution
###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = icmp
  chksum    = None
  src       = 147.210.0.2
  dst       = 192.168.0.2
  \options   \
###[ ICMP ]###
     type      = echo-request
     code      = 0
     chksum    = None
     id        = 0x0
     seq       = 0x0

[ 1248.694708] device eth0 entered promiscuous mode
[ 1248.696092] device lo entered promiscuous mode
Begin emission:
.Finished sending 1 packets.
*
Received 2 packets, got 1 answers, remaining 0 packets
[ 1249.700777] device eth0 left promiscuous mode
[ 1249.702272] device lo left promiscuous mode
###[ IP ]###
  version   = 4
  ihl       = 5
  tos       = 0x0
  len       = 28
  id        = 15059
  flags     =
  frag      = 0
  ttl       = 63
  proto     = icmp
  chksum    = 0xec8f
  src       = 192.168.0.2
  dst       = 147.210.0.2
  \options   \
###[ ICMP ]###
     type      = echo-reply
     code      = 0
     chksum    = 0xffff
     id        = 0x0
     seq       = 0x0
###[ Padding ]###
        load      = '\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
```

1.4. ARP

RAPPELEZ LE FONCTIONNEMENT DU PROTOCOLE ARP. NOTEZ QUE LE PROTOCOLE ARP NE DISPOSE QUE DE DEUX OPERATIONS : LA REQUETE (WHO HAS) ET LA REPONSE. ON PEUT ALORS UTILISER CE PROTOCOLE POUR EFFECTUER UN PING DANS LE RESEAU LOCAL ETHERNET. IL S'AGIT D'ENVOYER UNE REQUETE ARP. SI LA MACHINE REPOND, C'EST BIEN QU'ELLE EST EN VIE !

Le protocole ARP fait le lien entre une adresse IP et une adresse physique. Il va d'abord interroger tous les périphériques et si la machine recherchée est bien présente elle enverra une réponse directe à l'émetteur.

COMMENCEZ PAR CONSTRUIRE UNE TRAME ETHERNET AVEC ETHER() VERS L'ADRESSE DE BROADCAST FF:FF:FF:FF:FF:FF ET ENCAPSULEZ LE DATAGRAMME ARP() A DESTINATION DE L'ADRESSE IP VISEE. POUR ENVOYER ET RECEVOIR UNE TRAME ETHERNET, IL FAUT UTILISER LA FONCTION SRP1() (A LA PLACE DE LA FONCTION SR1() RESERVE AUX PAQUETS IP). ON PEUT AUSSI UTILISER DANS CETTE FONCTION L'OPTION TIMEOUT=1 POUR LIMITER LE TEMPS D'ATTENTE A 1 SECONDE, DANS LE CAS OU IL N'Y A PAS DE REPONSE.

```python
#!/usr/bin/env python3

import sys
from scapy.all import *

ans, unans = srp1(Ether(dst = 'FF:FF:FF:FF:FF:FF') / ARP(pdst = sys.argv[1]), timeout = 1)
```

1.5. SERVICES UDP : DAYTIME ET ECHO

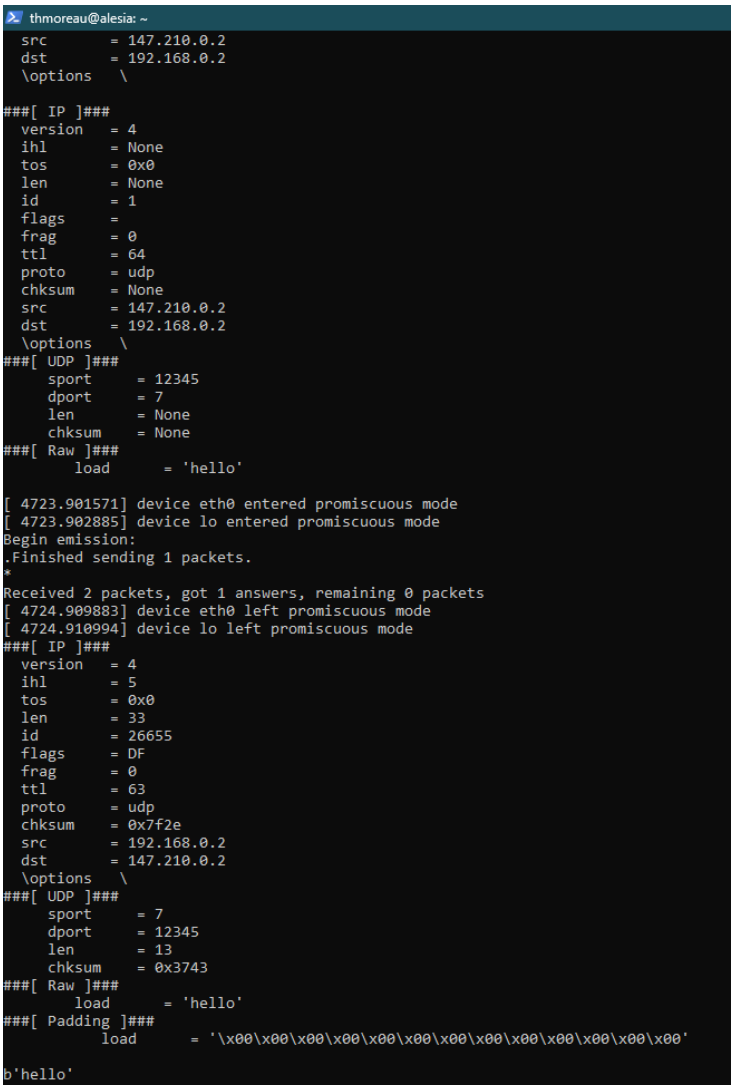
SUIVEZ L'EXEMPLE DU FICHIER DAYTIME.PY QUI ENVOIE UN PAQUET UDP SUR LE PORT DAYTIME (13) PUIS RECUPERE ET AFFICHE LA DATE ENVOYEE EN REPONSE. ESSAYEZ PAS A PAS.

```
root@grave:~# python3 daytime.py

Bad key "text.kerning_factor" on line 4 in
/usr/share/matplotlib/mpl-data/stylelib/_classic_test_patch.mplstyle.
You probably need to get an updated matplotlibrc file from
http://github.com/matplotlib/matplotlib/blob/master/matplotlibrc.template
or from the matplotlib source distribution
###[ UDP ]###
  sport     = domain
  dport     = domain
  len       = None
  chksum    = None
###[ Raw ]###
     load      = 'hello\n'

###[ UDP ]###
  sport     = 12345
  dport     = daytime
  len       = None
  chksum    = None
###[ Raw ]###
     load      = 'hello\n'

###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = hopopt
  chksum    = None
  src       = 147.210.0.2
  dst       = 192.168.0.2
  \options   \

###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = udp
  chksum    = None
  src       = 147.210.0.2
  dst       = 192.168.0.2
  \options   \
###[ UDP ]###
     sport     = 12345
     dport     = daytime
     len       = None
     chksum    = None
###[ Raw ]###
        load      = 'hello\n'
```

```
[ 3922.632532] device eth0 entered promiscuous mode
[ 3922.633516] device lo entered promiscuous mode
Begin emission:
.Finished sending 1 packets.
*
Received 2 packets, got 1 answers, remaining 0 packets
[ 3923.641361] device eth0 left promiscuous mode
[ 3923.642795] device lo left promiscuous mode
###[ IP ]###
  version   = 4
  ihl       = 5
  tos       = 0x0
  len       = 54
  id        = 30700
  flags     = DF
  frag      = 0
  ttl       = 63
  proto     = udp
  chksum    = 0x6f4c
  src       = 192.168.0.2
  dst       = 147.210.0.2
  \options   \
###[ UDP ]###
     sport     = daytime
     dport     = 12345
     len       = 34
     chksum    = 0x6338
###[ Raw ]###
        load      = 'Sun Oct 16 23:01:32 2022\r\n'

b'Sun Oct 16 23:01:32 2022\r\n'
```

VOUS AVEZ PU REMARQUER QUE LE SERVICE UDP ECHO EST OUVERT (PORT 7). TESTEZ CE SERVICE EN ENVOYANT LE MESSAGE 'HELLO'. QUELLE EST LA REPONSE ?

Avec le port udp echo :



La réponse est visible sur la dernière ligne : b’hello’.