step1

April 19, 2023

[ ]:

```
from scapy.all import *

print(f"Scapy version {conf.version}")
print(f"Interface in use {conf.iface}")
print(f"\nRouting table: \n {conf.route}.")
print(f"\nGateway:", conf.route.route("0.0.0.0")[2])
```

```
Scapy version 2.5.0
Interface in use eth0

Routing table:
 Network          Netmask          Gateway     Iface    Output IP   Metric
0.0.0.0          0.0.0.0          172.16.5.1  eth0     172.16.5.4  100
127.0.0.0        255.0.0.0        0.0.0.0     lo       127.0.0.1   1
168.63.129.16    255.255.255.255  172.16.5.1  eth0     172.16.5.4  100
169.254.169.254  255.255.255.255  172.16.5.1  eth0     172.16.5.4  100
172.16.5.0       255.255.255.0    0.0.0.0     eth0     172.16.5.4  0
172.17.0.0       255.255.0.0      0.0.0.0     docker0  172.17.0.1  0     .

Gateway: 172.16.5.1
```

[ ]:

```
# see header format of a supported protocols

ls(ICMP)
```

```
type       : ByteEnumField                       = ('8')
code       : MultiEnumField (Depends on 8)       = ('0')
chksum     : XShortField                         = ('None')
id         : XShortField (Cond)                  = ('0')
seq        : XShortField (Cond)                  = ('0')
ts_ori     : ICMPTimeStampField (Cond)           = ('56399600')
ts_rx      : ICMPTimeStampField (Cond)           = ('56399600')
ts_tx      : ICMPTimeStampField (Cond)           = ('56399600')
gw         : IPField (Cond)                      = ("'0.0.0.0'")
ptr        : ByteField (Cond)                    = ('0')
reserved   : ByteField (Cond)                    = ('0')
length     : ByteField (Cond)                    = ('0')
addr_mask  : IPField (Cond)                      = ("'0.0.0.0'")
```

```
nexthopmtu : ShortField (Cond)                   = ('0')
unused     : MultipleTypeField (ShortField, IntField, StrFixedLenField) =
("b''")
```

[ ]:

```
# creating a packet

packet1 = IP()/UDP()

# packet1.summary()
# packet1.show()
# packet1.show2()

source_ip = "172.16.5.4"
destination_ip = "172.16.5.1"
packet2=IP(src=source_ip, dst=destination_ip)/UDP()
# packet2.show()

src_mac = "11:22:33:44:55:66"
dst_mac = "00:11:AA:BB:CC:DD"
src_ip = "127.0.0.1"
dst_ip = "www.google.fr"
frame = Ether(src=src_mac, dst=dst_mac)/IP(src=src_ip, dst=dst_ip)/TCP()/"allo"
# frame = Ether(src=src_mac, dst=dst_mac)/IP(src=src_ip, dst=dst_ip)/
 ↪TCP(flags="SA")/"allo"
frame.show2()
```

```
###[ Ethernet ]###
  dst       = 00:11:aa:bb:cc:dd
  src       = 11:22:33:44:55:66
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 44
     id        = 1
     flags     =
     frag      = 0
     ttl       = 64
     proto     = tcp
     chksum    = 0xb10c
     src       = 127.0.0.1
     dst       = 142.250.187.195
     \options   \
###[ TCP ]###
        sport     = ftp_data
        dport     = http
        seq       = 0
        ack       = 0
        dataofs   = 5
        reserved  = 0
        flags     = S
        window    = 8192
        chksum    = 0xf7df
        urgptr    = 0
        options   = []
###[ Raw ]###
           load      = 'allo'
```

[ ]:

```
# open pcap file in scapy

file = rdpcap("Ping_Google.pcapng")

# file.summary()
# file.show2()
# file[0].show()

# file[2]["IP"].show2()
# file[2]["IP"]
# file[2]["IP"].dst

file[2][Raw].load                    # to read the raw (icmp package)
file[2][Raw].load.split(sep=None)      # str sep w/ space into a list
file[2][Raw].load.split(sep=None)[2]   # the one we want
file[2][Raw].load.split(sep=None)[2].decode("UTF8")
```

[ ]:

```
'!"#$%&\'()*+,-./01234567'
```

[ ]:

```
# send frames or packets

packet = IP(dst="10.0.0.1", src="10.0.0.2")/ICMP()/"blabla"
send(packet)

# can be useful

# send(packet, loop=1)
# send(ping, loop=1, inter=1)
```

[ ]: