Faculte Des Sciences

Universite d'Etat d'Haiti

Securite Informatique

Devoir 1 – ICMP

NOM	:	THEODORE
PRENOM	:	Barbara G.
CLASSE	:	3eme Annee
SECTION	:	Electronique
LIEN GITHUB	:	https://github.com/BGTheodore/ICMP-Traceroute_program.git

PROFESSEUR : M. MATHIEU V.

DATE : Samedi, 7 Novembre 2020

L’objectif du devoir est de developper un programme cote serveur et cote client comme le fait ICMP. Il faudra implementer ping et traceroute ; et utiliser un port 6332 cote serveur.

Ce devoir a ete realise a partir de visual studio code sur un ordinateur hebergeant le systeme d’exploitation Linux Ubuntu.

Implementation du serveur

```
import socket

host = '127.0.0.1'  # Adresse du serveur local alias localhost
port = 6332        # Port exige
```

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as sock: #AF_INET pour preciser la famille d’adresse, dans ce cas IPV4 et SOCK_STREAM pour forcer TCP

```
    sock.bind((host, port))
    sock.listen()
    connection, adresse = sock.accept()
    with connection:
        print('Connecte par', adresse)
        while True:
            donnee = connection.recv(1024)
            if not donnee:
                break
            connection.sendall(donnee)
```

Implementation du client

```
import socket

with soc
host = '127.0.0.1'
port = 6332

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as sock:
    sock.connect((host, port))
    sock.sendall(b'Hello, world')
    donnee = sock.recv(1024)

print('Recu', repr(donnee))
```

Implementation du ping

```
import socket
from datetime import datetime
import http.server
import socketserver

adresse_ip = input("Enter the IP address: ")
debut = datetime.now()
```

```
tempsExec= 0.025
port = 6332
socket.setdefaulttimeout(tempsExec)

def scan(adresse):
  sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
  socket.setdefaulttimeout(1)
  resultat = sock.connect_ex((adresse,port))
  if resultat == 0:
    return 1
  else :
    return 0

def ping(adresse):
    if (scan(adresse)):
      fin = datetime.now()
      total = fin - debut
      print('Succes !!')
      print ("from {}: ttl={} time={} ".format(adresse, tempsExec, total))
    else:
      print('Port unreachable')

# Lancement
ping(adresse_ip )
```

Implementation du traceroute

```
#Programme a compiler en mode privilegie (sudo)
import socket
import time
port = 6332
maxhop = 50

def Traceroute(nomDestination):
   adresseDestination= socket.gethostbyname(nomDestination)
   icmp = socket.getprotobyname( "icmp")
   udp = socket.getprotobyname('udp')
   tempsExec = 1
   while True:
      envoiPaquet = socket.socket(socket.AF_INET, socket.SOCK_DGRAM, udp)
       receptionPaquet = socket.socket(socket.AF_INET, socket.SOCK_RAW, icmp)
      envoiPaquet.setsockopt(socket.SOL_IP, socket.IP_TTL, tempsExec)
      receptionPaquet.bind(("", port))
      envoiPaquet.sendto("TEST", (nomDestination, port))

      hopPaquet = None
      hopNom = None

      try:
         donnee, hopPaquet = receptionPaquet.recvfrom(512)
```

```
                hopPaquet= hopPaquet[0]
                try:
                    hopNom = socket.gethostbyaddr(hopPaquet)[0]

                except socket.error:
                    hopNom = hopPaquet
            except socket.error:
                print('timeout error')
            finally:
                envoiPaquet.close()
                receptionPaquet.close()

            if hopPaquet is not None:
                actuel= "%s (%s)" % (hopNom, hopPaquet)
            else:
                actuel = "*"
            print "%d\t%s" % (tempsExec, actuel)

            tempsExec += 1

            if actuel == adresseDestination or tempsExec > maxhop:
                break

    if __name__ == "__main__":
        nomDestination = raw_input('Enter the destination : ')
        print('Traceroute to {} ({}) on port : {}, {} hops max'.format(nomDestination,
socket.gethostbyname(nomDestination), port ,maxhop))
        startTime = time.time()
        Traceroute(nomDestination)
```