***X–Laboratorio***

Si vuole scrivere un'applicazione client/server UDP per inviare un messaggio segreto che verrà salvato sul server.  
Il client, per far sì che il messaggio segreto venga salvato, deve prima inviare una precisa sequenza di messaggi (parole d’ordine) su differenti porte del server.

**Hint**: serverUtils.record(X) salva il contenuto della stringa X nella memoria del server.

**Client**

from socket import \*

serverName = 'www.knockingonheavensdoor.com'

clientSocket = socket(AF\_INET,SOCK\_DGRAM)

clientSocket.settimeout(2)

messageIsSent = False

while not messageIsSent:

serverPort = int(raw\_input('Numero Porta:'))

magicWordS = raw\_input('Parola d\'ordine:')

clientSocket.sendto(magicWordS, (serverName, serverPort))

try:

serverMessage, serverAddress = clientSocket.recvfrom(1024)

print serverMessage

if serverMessage == 'SUCCESSO\n':

secretMessage = raw\_input('Inserire il messaggio per il server: ')

clientSocket.sendto(secretMessage, serverAddress)

messageIsSent = True

except:

print 'Riprova.'

clientSocket.close()

**server**

from socket import \*

import serverUtils

serverPortS = [12000, 12004, 12008]

magicWordS = ['APRI', 'QUESTA', 'PORTA']

index = 0

while True:

serverSocket = socket(AF\_INET,SOCK\_DGRAM)

serverSocket.bind(( '', serverPortS[ index ] ) )

serverSocket.settimeout( 60 )

try:

clientMessage, clientAddress = serverSocket.recvfrom(1024)

if clientMessage == magicWordS[index]:

if index < len( serverPortS )-1:

serverSocket.sendto('OK\n',clientAddress)

index += 1

else:

serverSocket.sendto('SUCCESSO\n',clientAddress)

serverSocket.sendto('Server pronto a ricevere!\n',clientAddress)

serverSocket.settimeout( 120 )

clientMessage, clientAddress = serverSocket.recvfrom(1024)

serverUtils.record(clientMessage)

index = 0

else:

index = 0

except:

index = 0

serverSocket.close())

**Q1)** Dato il codice del server, completare la tabella specificando la sequenza di parole d'ordine e porte è necessario inviare per riuscire ad inviare il messaggio segreto che verrà salvato sul server.

|  |  |
| --- | --- |
| **Porta** | **Parola d’ordine** |
| **12000** | **APRI** |
| **12004** | **QUESTA** |
| **12008** | **PORTA** |
|  |  |
|  |  |

**Q2)** Completare il Codice del client in modo tale da avere un timeout di ricezione di 2 secondi, ed una dimensione del buffer di ricezione del client di 1024 Bytes

**Q3)** Completare il codice del server in modo tale da settare un timeout sulla ricezione del messaggio segreto di 120 secondi, e di 60 secondi sulla ricezione delle parole d’ordine.

**Q4)** Quando viene visualizzato il messaggio **'Server pronto a ricevere!\n'** dal client? Motivare la risposta.

Il messaggio non viene mai visualizzato perché, una volta ricevuto il messaggio 'SUCCESSO\n' dal server, non viene più fatta nessuna recvFrom().