

Relazione Progetto

Traccia 3 - Python Telnet Server

Andrea Acampora
0000873699

May 2020

0.1 Introduzione

Il progetto consiste nel realizzare un programma con il linguaggio *Python* che permetta una connessione client-server sfruttando il protocollo *TCP-IP*.

0.2 Descrizione

Sebbene esista il comando *Telnet* per connettersi ad un server tramite il protocollo Telnet ho invece deciso di realizzare anche la parte Client in modo da rendere il programma più complesso e personalizzabile. Ho realizzato quindi due diversi programmi:

- **Server.py**
Eseguibile con :
"python3 Server.py"
- **Client.py**
Eseguibile con :
"python3 Client.py"

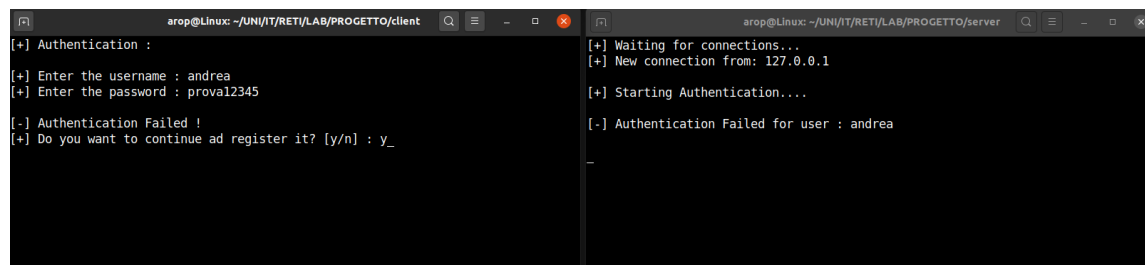
E' necessario avere installato *Python3* sul dispositivo.
E' consigliato eseguire i programmi dalle rispettive cartelle **server/** e **client/** dentro alla cartella del progetto in modo da agevolare il cambio di directory e l'invio/caricamento di file.
In entrambe le cartelle, infatti, saranno presenti due file di testo di prova : *server-file.txt* e *client-file.txt*. I file servono per testare il download di file dal server al client e l'upload dal client al server.

Una volta eseguito il server rimarrà in attesa di eventuali connessioni sull'indirizzo locale 127.0.0.1 .

```
[+] Waiting for connections...
```

Nel momento in cui il client viene eseguito esso si connette al server e quest'ultimo chiederà al client di autenticarsi inserendo *username* e *password*. L'autenticazione viene gestita attraverso lettura e scrittura su un file nascosto chiamato *.credentials* nella home dell'utente contenente tutti gli account registrati.

Il Server controllerà quindi se l'username e la password inseriti dal client sono presenti nel file ed in caso negativo permetterà di registrarsi salvando le nuove credenziali per le prossime connessioni.



```
arop@Linux: ~/UNI/IT/RETI/LAB/PROGETTO/client
[+] Authentication :
[+] Enter the username : andrea
[+] Enter the password : proval2345
[-] Authentication Failed !
[+] Do you want to continue ad register it? [y/n] : y_

arop@Linux: ~/UNI/IT/RETI/LAB/PROGETTO/server
[+] Waiting for connections...
[+] New connection from: 127.0.0.1
[+] Starting Authentication...
[-] Authentication Failed for user : andrea
```

Una volta autenticato il Client potrà svolgere diverse operazioni sul Server tra le quali:

- **1** Restituisce la lista dei file nella cartella corrente del server
- **2** Restituisce il percorso (PATH) della cartella corrente del server
- **3** Permette di eseguire un comando sulla shell del server
- **4** Permette di cambiare cartella sul server
- **5** Permette di leggere il contenuto di un file del server sul client
- **6** Permette di scaricare un file dal server al client.
- **7** Permette di caricare un file dal client al server.

Per ogni azione eseguita il Server terrà traccia dell'operazione stampando a video il completamento dell'operazione.

```

aerop@Linux: ~/UNI/IT/RETI/LAB/PROGETTO/server
[+] Waiting for connections...
[+] New connection from: 127.0.0.1
[+] Starting Authentication....
[-] Authentication Failed for user : andrea
[+] Authentication Completed -> new user andrea registered !
[+] Sending the list of files in current directory
[+] Sending the path of current directory
[+] Sending server_file.txt to Client
[+] Changed directory to ../client
```

Il Client invece avrà un prompt per inserire i comandi da lanciare sul server e permetterà di visualizzare le opzioni disponibili digitando *help*.

```
Welcome to Telnet Server !

Options:

-> 1   Return the list of files in the directory
-> 2   Return the path of the directory
-> 3   Execute a command on the Server shell and get the output
-> 4   Change directory
-> 5   Read the content of a Server file
-> 6   Download a file from the Server
-> 7   Upload a file to the Server

-> 0   Close the connection

Enter [0-7] or [help] -> _
```

0.3 Dettagli Implementativi

Il Client è stato implementato in modo da connettersi a qualsiasi indirizzo IP inserendolo come argomento.

Esempio : `python3 Client.py -ip 192.168.10.35`

Nel caso in cui l'indirizzo non venga specificato il programma si connette in localhost.

I messaggi vengono codificati e successivamente decodificati in formato *json* tramite la rispettiva libreria in modo da garantire una maggior sicurezza ed efficienza nell'invio di file o dati di grosse dimensioni.

Tutto il ciclo di invio e ricezione di messaggi viene effettuato dentro ad un *try-catch* in modo da non chiudere la connessione in caso di eccezioni durante l'esecuzione di comandi.

Nelle fasi di download e upload di files tra client e server viene eseguito un controllo sulla presenza del file e in caso negativo viene comunicato ad entrambi il fallimento dell'operazione.

0.3.1 Librerie utilizzate

- os
- sys
- subprocess
- argparse
- socket
- json

In caso di problemi con la cartella compressa , è possibile visionare il progetto al seguente link:

<https://github.com/andrea-acampora/Telnet-Networking-Project>