

# Relazione S2L3

## Codice 1

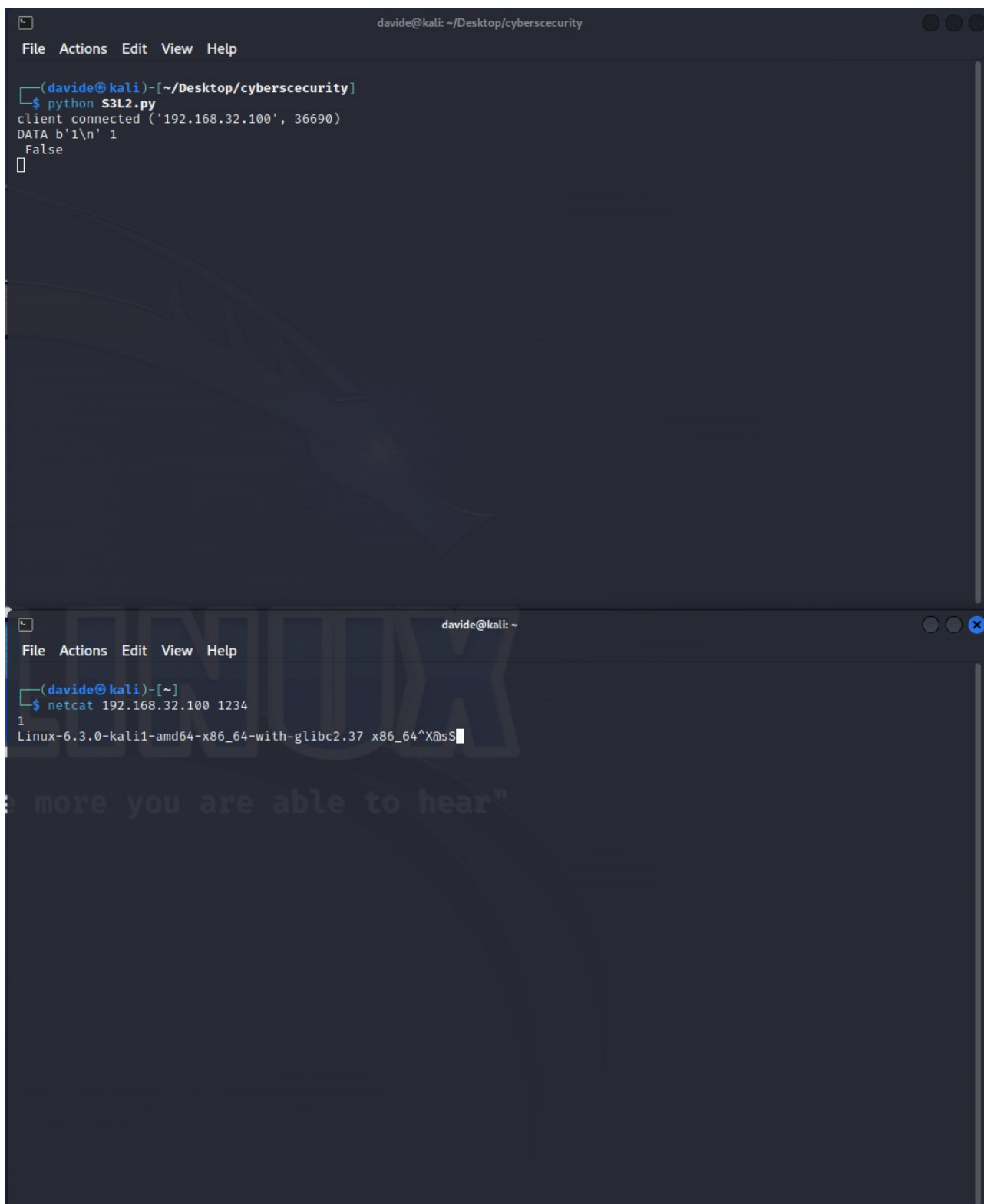
```
1 import socket, platform, os
2
3 SRV_ADDR = "192.168.32.100"
4 SRV_PORT = 1234
5
6 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7 s.bind((SRV_ADDR, SRV_PORT))
8 s.listen()
9 connection, address = s.accept()
10
11 print ("client connected", address)
12
13 while 1:
14     try:
15         data = connection.recv(1024)
16         except: continue
17
18         if (data.decode("utf-8").strip() == "1"):
19             tosend = platform.platform() + " " + platform.machine()
20             connection.sendall(tosend.encode())
21         elif (data.decode("utf-8").strip() == "2"):
22             data = connection.recv(1024)
23             data = data.decode("utf-8").strip()
24
25             try:
26                 filelist = os.listdir(data)
27                 tosend = ""
28                 for x in filelist:
29                     tosend += "," + x
30             except OSError as e:
31                 tosend = f'Errore: {str(e)}'
32             connection.sendall(tosend.encode())
33         elif (data.decode("utf-8").strip() == "0"):
34             connection.close()
35             connection, address = s.accept()
36
```

**N.B. Il codice è stato modificato per funzionare correttamente**

Questo codice crea un socket sulla macchina kali linux e resta in ascolto per una connessione da parte del client.

Quando dal client si scrive un testo il server controlla che il testo inserito sia 1, 2 o 0.

Nel caso in cui il testo corrisponda a 1, il server invia al client informazioni sulla macchina server:



The image shows two terminal windows from a Kali Linux machine. The top window is titled 'davide@kali: ~/Desktop/cyberscsecurity' and shows a Python script 'S3L2.py' being executed. The script receives a client connection from '192.168.32.100' on port '36690' and receives the data 'b'1\n' 1'. The script then prints 'False'. The bottom window is titled 'davide@kali: ~' and shows a netcat listener on '192.168.32.100' port '1234'. It receives the input '1' and responds with the system information: 'Linux-6.3.0-kali1-amd64-x86\_64-with-glibc2.37 x86\_64^X@sS'.

```
(davide@kali)-[~/Desktop/cyberscsecurity]
$ python S3L2.py
client connected ('192.168.32.100', 36690)
DATA b'1\n' 1
False
[]
```

```
(davide@kali)-[~]
$ netcat 192.168.32.100 1234
1
Linux-6.3.0-kali1-amd64-x86_64-with-glibc2.37 x86_64^X@sS
```

Come si può vedere dopo aver digitato 1 abbiamo avuto la risposta sul client da parte del server.

Se digitiamo 2 invece la macchina ci restituirà l'elenco delle cartelle in base al path che indichiamo:



The image consists of two terminal window screenshots. The top window shows a netcat listener on Kali Linux at IP 192.168.32.100, port 1234. It receives a connection from 192.168.32.100. The bottom window shows the same netcat listener after the user enters '2', displaying a directory listing of the remote system.

```
davide@kali: ~/Desktop/cybersecurity
File Actions Edit View Help

(davide@kali)-[~/Desktop/cybersecurity]
$ python S3L2.py
client connected ('192.168.32.100', 59292)
[]

davide@kali: ~
File Actions Edit View Help

(davide@kali)-[~]
$ netcat 192.168.32.100 1234
2
/
,root,mnt,media,var,tmp,home,boot,opt,sbin,usr,libx32,lost+found,lib32,vmlinuz.old,proc,lib64,vmlinuz,dev,etc,lib,initrd.img,sys,initrd.img.old,.cache[]
```

Come si può vedere dopo aver digitato 2 possiamo inserire un altro carattere e sarà valido ci verranno mostrate le cartelle sottostanti.

Col tasto 0 invece la connessione viene chiusa

## Codice 2

```
1 import socket
2
3 SRV_ADDR = input("Type the server IP address: ")
4 SRV_PORT = int(input("Type the server port: "))
5
6 def print_menu():
7     print("""\n\n
8         0) Close the connection
9         1) Get system info
10        2) List directory contents""")
11
12 my_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
13 my_sock.connect((SRV_ADDR, SRV_PORT))
14
15 print("Connection established")
16 print_menu()
17
18 while 1:
19     message = input("\n-Select an option: ")
20
21     if(message.strip() == "0"):
22         my_sock.sendall(message.encode())
23         my_sock.close()
24         break
25
26     elif(message.strip() == "1"):
27         my_sock.sendall(message.encode())
28         data = my_sock.recv(1024)
29         if not data: break
30         print(data.decode("utf-8"))
31
32     elif(message.strip() == "2"):
33         path = input("Insert the path: ")
34         my_sock.sendall(message.encode())
35         my_sock.sendall(path.encode())
36         data = my_sock.recv(1024)
37         data = data.decode("utf-8").split(",")
38         print(" "*40)
39         for x in data:
40             print(x)
41         print(" "*40)
42
```

Questo codice a differenza del precedente, è utilizzato per creare un socket lato client con cui connettersi a quello server di cui abbiamo parlato sopra.

Quindi per connetterci al server non abbiamo bisogno di seguire il comando **netcat ip** ma ci basterà eseguire questo file.

Quello che facciamo è impostare una connessione col server dando in input l'ip e la porta su cui sta running il server.

Una volta effettuato viene lanciata la funzione `print_menu()` che ci fa scegliere tra tre opzioni: 0, 1, 2.

Queste opzioni corrispondono a quelle lato client, infatti gli sono stati assegnati gli stessi numeri e ritornano informazioni sul sistema su cui gira il server e sui file che contiene inserendo il path appropriato.

```
davide@kali: ~/Desktop/cyberscecurity
File Actions Edit View Help

(davide@kali)-[~/Desktop/cyberscecurity]
$ python S3L2.py
client connected ('192.168.32.100', 45106)
█

davide@kali: ~/Desktop/cyberscecurity
File Actions Edit View Help

(davide@kali)-[~/Desktop/cyberscecurity]
$ python S3L2-2.py
Type the server IP address: 192.168.32.100
Type the server port: 1234
Connection established

0) Close the connection
1) Get system info
2) List directory contents

-Select an option: 1
Linux-6.3.0-kali1-amd64-x86_64-with-glibc2.37 x86_64

-Select an option: █
```

```
davide@kali: ~/Desktop/cyberssecurity
File Actions Edit View Help

(davide@kali)-[~/Desktop/cyberssecurity]
$ python S3L2.py
client connected ('192.168.32.100', 45106)
[]

davide@kali: ~/Desktop/cyberssecurity
File Actions Edit View Help
-Select an option: 2
Insert the path: /
*****
root
mnt
media
var
tmp
home
boot
opt
sbin
srv
bin
run
usr
libx32
lost+found
lib32
vmlinuz.old
proc
lib64
vmlinuz
dev
etc
lib
initrd.img
sys
initrd.img.old
.cache
*****
-Select an option: █
```

Davide Lecci