

((گزارش پروژه اول مبانی امنیت اطلاعات))

گردآورنده : عرفان ماجدی 9831099

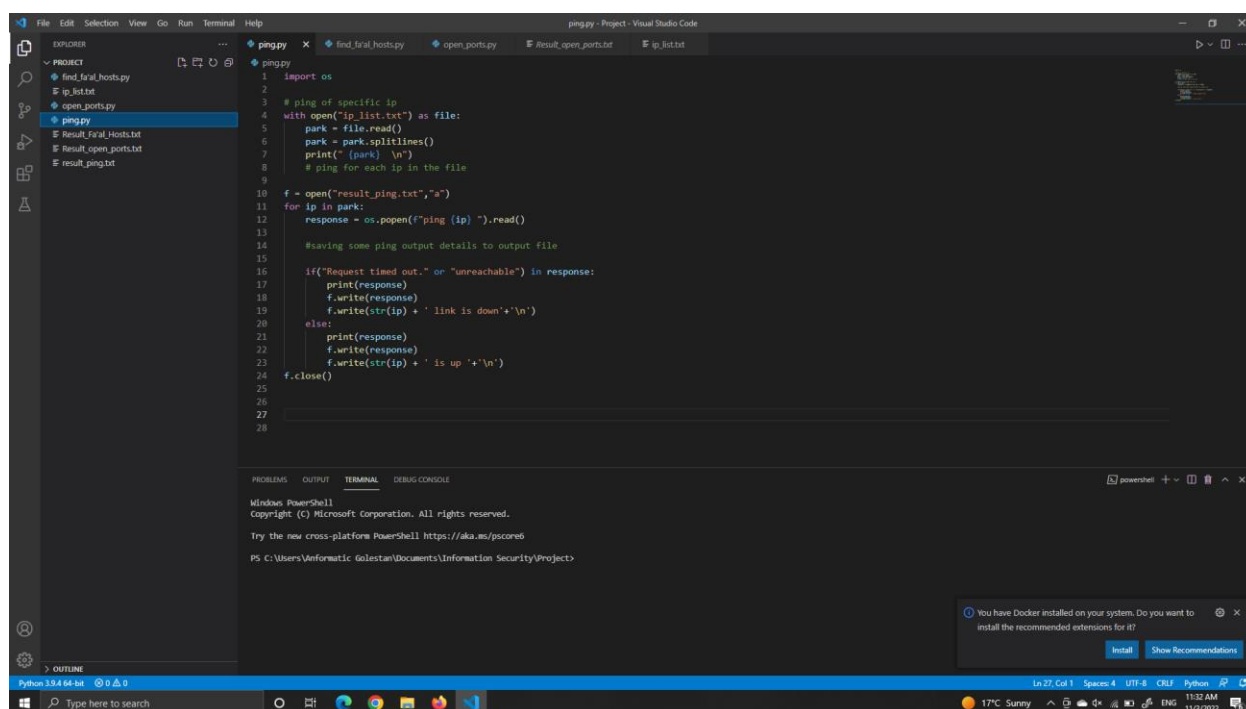
مبانی امنیت اطلاعات

دکتر شهریاری

گزارش بخش اول (

1) گرفتن ping از یک آی پی خاص :

کد این قسمت به شکل زیر است :



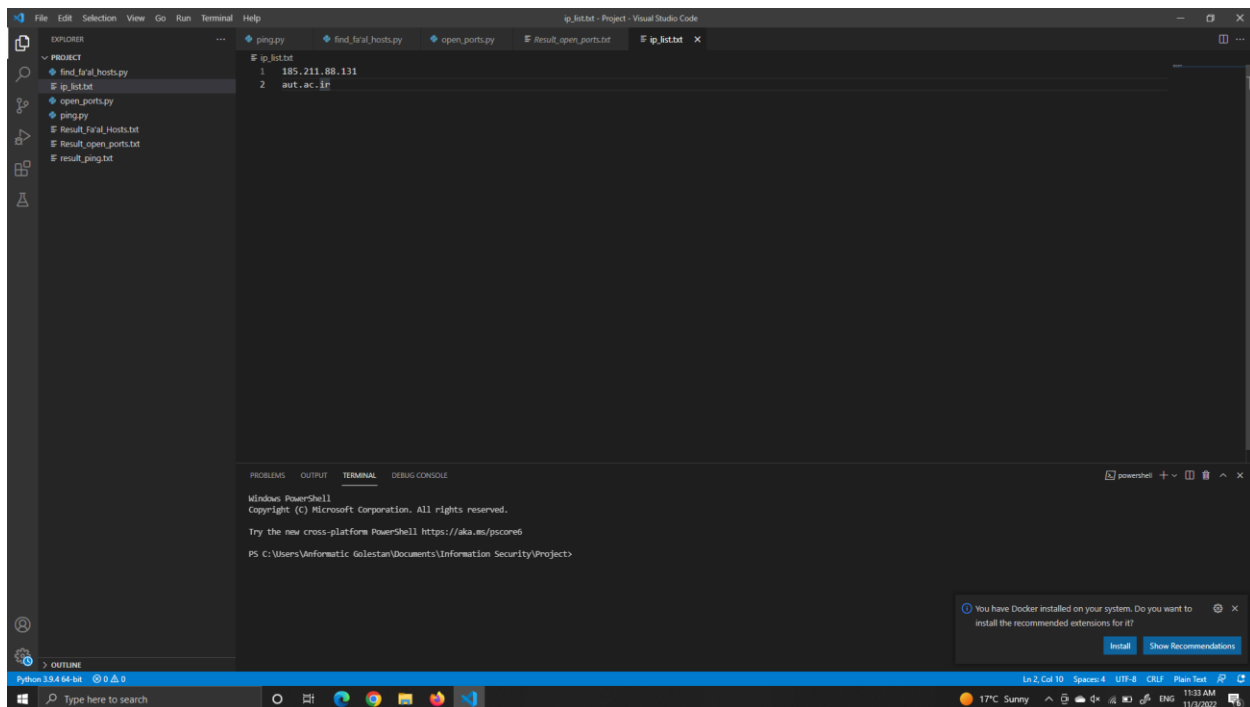
```
1 import os
2
3 # ping of specific ip
4 with open("ip_list.txt") as file:
5     park = file.read()
6     park = park.splitlines()
7     print(" park \n")
8     # ping for each ip in the file
9
10 f = open("result_ping.txt","a")
11 for ip in park:
12     response = os.popen(f'ping {ip} ').read()
13
14     #saving some ping output details to output file
15
16     if("Request timed out." or "unreachable") in response:
17         print(response)
18         f.write(response)
19         f.write(str(ip) + ' link is down'+'\n')
20     else:
21         print(response)
22         f.write(response)
23         f.write(str(ip) + ' is up '+'\n')
24 f.close()
25
26
27
28
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell <https://aka.ms/powershell>
PS C:\Users\Verformatic\Golestan\Documents\Information Security\Project>

You have Docker installed on your system. Do you want to install the recommended extensions for it?
[Install] [Show Recommendations]

کد بخش 1



محتوای فایل ip_list

در این بخش ابتدا با استفاده از `cmd`، `ip address` سایت دانشگاه به نشانی `aut.ac.ir` را پیدا کرده و این ادرس و دامنه ی سایت را در یک فایل `txt` ذخیره کردیم. در پیاده سازی کد از کتابخانه ی `os` استفاده کردیم. در ابتدا فایل `txt` را باز کردیم و محتوای آن را خواندیم و در متغیر `park` ذخیره کردیم و سپس آن را `print` کردیم. سپس فایل `result_ping` را در قالب یک فایل `txt` درست کردیم و در متغیر `f` قرار دادیم. حال متغیر `park` که شامل `ip` ها می باشد را در یک حلقه ی `for` گذاشتیم تا محتوای آن را گردش کنیم و به ازای هر `ip` از `method` ای استفاده کردیم به نام `os.popen` که در واقع کاری همان گرفتن `ping` را انجام می دهد و سپس در آخر خط یک `method read()` قرار دادیم تا محتوای `object` ساخته شده را بخواند. در آخرین مرحله دو حالت وجود دارد :

(1) درخواست ما `time out` شود یا پیام `unreachable` دریافت کنیم -> ادرس `ip` ما `down` است.

(2) پیام های بالا را دریافت نکنیم -> ادرس `ip` ما `up` است.

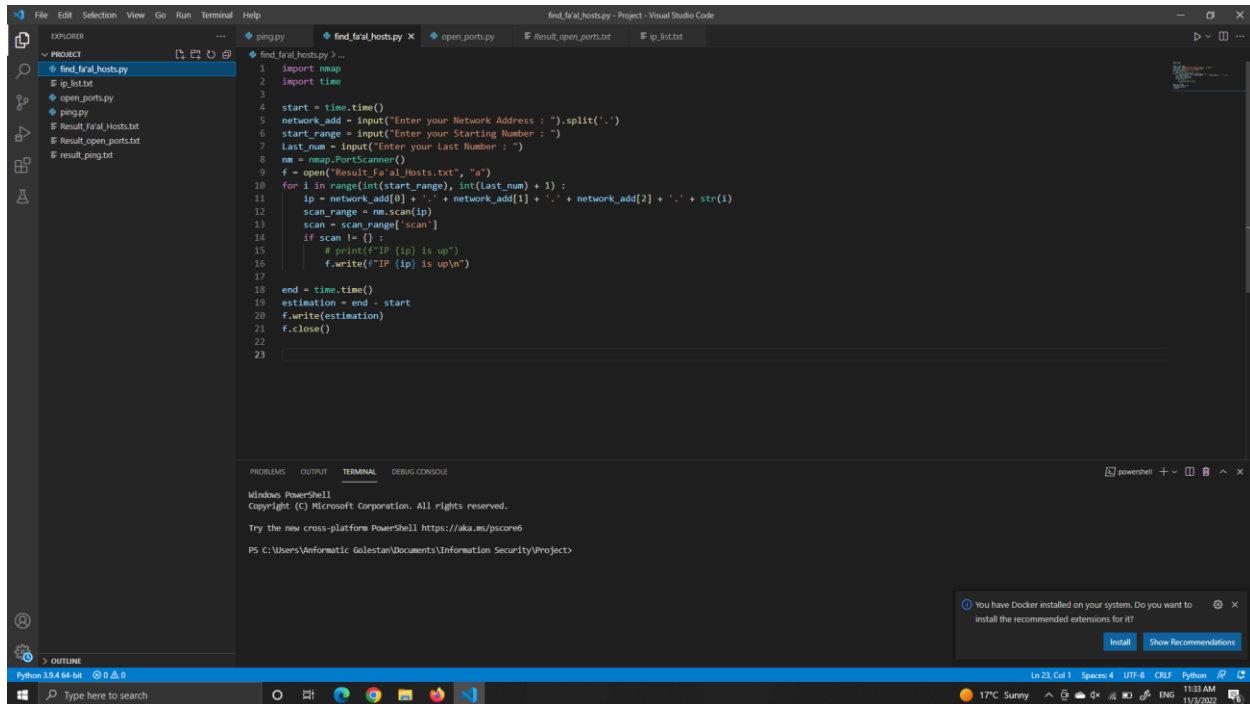
دو حالت توضیح داده شده را در قالب یک `if - else` می نویسیم و سپس نتیجه را در فایلی که در `f` ریختیم ذخیره می کنیم. نتیجه به صورت زیر خواهد بود :

```
1
2 Ping 185.211.88.131 with 32 bytes of data:
3 Reply from 185.211.88.131: bytes=32 time=4ms TTL=61
4 Reply from 185.211.88.131: bytes=32 time=4ms TTL=61
5 Reply from 185.211.88.131: bytes=32 time=7ms TTL=61
6 Reply from 185.211.88.131: bytes=32 time=4ms TTL=61
7
8 Ping statistics for 185.211.88.131:
9     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
10    Approximate round trip times in milli-seconds:
11        Minimum = 4ms, Maximum = 7ms, Average = 4ms
12 185.211.88.131 is up
13
14 Ping aut.ac.ir [185.211.88.131] with 32 bytes of data:
15 Reply from 185.211.88.131: bytes=32 time=4ms TTL=61
16 Reply from 185.211.88.131: bytes=32 time=4ms TTL=61
17 Reply from 185.211.88.131: bytes=32 time=4ms TTL=61
18 Reply from 185.211.88.131: bytes=32 time=3ms TTL=61
19
20 Ping statistics for 185.211.88.131:
21     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
22    Approximate round trip times in milli-seconds:
23        Minimum = 3ms, Maximum = 4ms, Average = 3ms
24 aut.ac.ir is up
25
```

محتوای فایل result_ping

2) اسکن یک محدوده آی پی خاص و یافتن هاست های فعال :

کد این بخش نیز به صورت زیر است :



```
1 import nmap
2 import time
3
4 start = time.time()
5 network_add = input("Enter your Network Address : ").split('.')
6 start_range = input("Enter your Starting Number : ")
7 last_num = input("Enter your Last Number : ")
8 nm = nmap.PortScanner()
9 f = open("Result_Fa'al_Hosts.txt", "a")
10 for i in range(int(start_range), int(last_num) + 1):
11     ip = network_add[0] + '.' + network_add[1] + '.' + network_add[2] + '.' + str(i)
12     scan_range = nm.scan(ip)
13     scan = scan_range['scan']
14     if scan != {}:
15         print(f"IP {ip} is up")
16         f.write(f"IP {ip} is up\n")
17
18 end = time.time()
19 estimation = end - start
20 f.write(estimation)
21 f.close()
22
23
```

Terminal output:

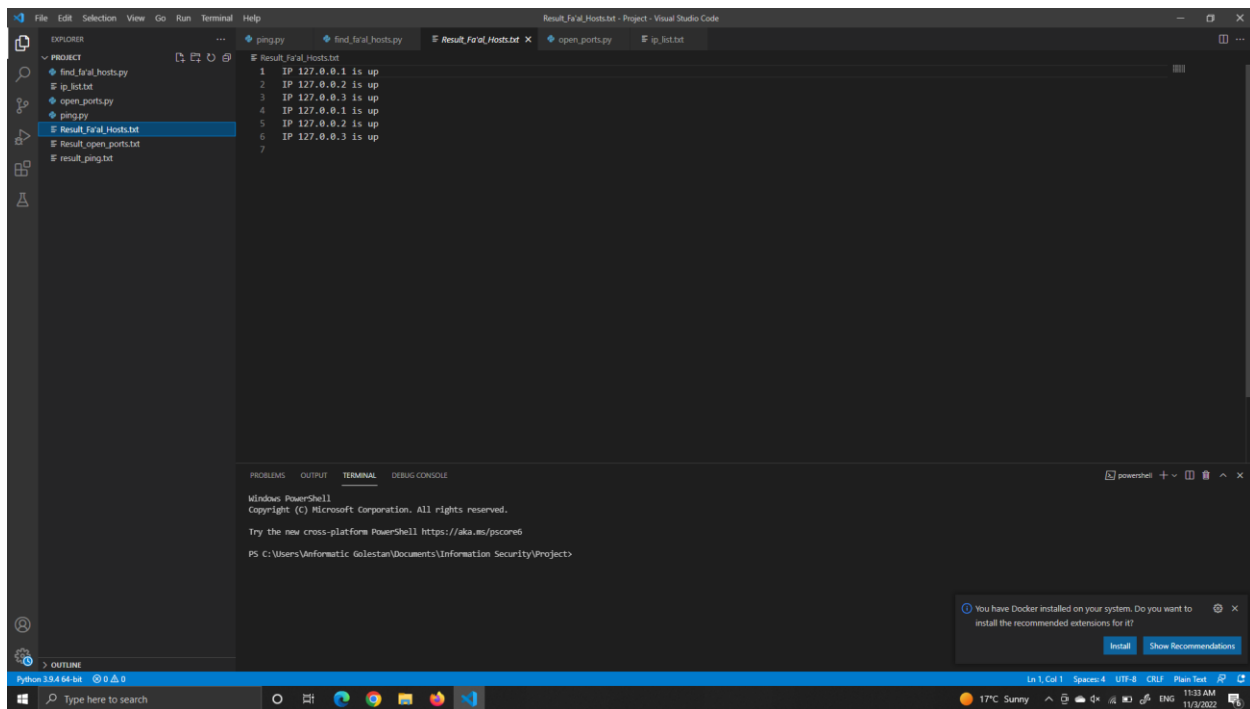
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\Veformatic\Golestan\Documents\Information Security\Project>
```

کد بخش دوم

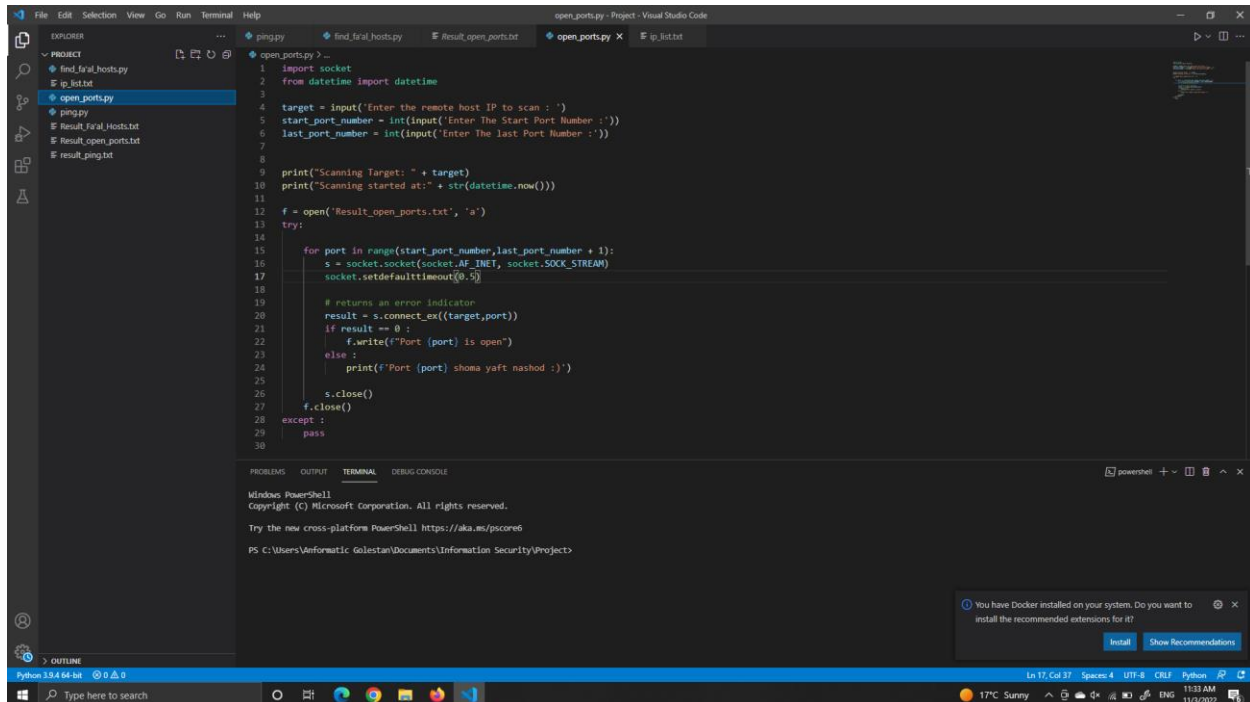
در این بخش از کتابخانه های `nmap` و `time` استفاده کردیم. در اولین خط کد شروع زمان گیری را انجام میدهم و آن را `start` نامیدیم. سپس سه تا ورودی از کاربر میگیریم که شامل ادرس `ip` است که با نقطه `split` کردیم، عدد شروع و پایان `ip` را هم گرفتیم. سپس یک `object` از تابع `nmap` ساختیم با استفاده از `nmap.PortScanner()` و در `nm` ریختیم. برای ذخیره ی نتیجه ی این کد هم یک فایل `txt` به اسم `Result_Fa'al_Hosts` درست کردیم. حال یک حلقه `for` در `range` عدد شروع و (پایان + 1) زدیم و توجه کنید چون ورودی `str` است آن ها را به `int` تبدیل کردیم. حال `ip` ها را در این محدوده درست کردیم (خط 11) و کار `scan` را با استفاده از `method` ای به نام `nm.scan()` انجام دادیم و چون به صورت یک `dictionary` است و ما با `key = scan` کار داریم، `value` های آن را در متغیر `scan` ذخیره می کنیم. حال اگر این `scan` خالی نباشد می گوئیم `ip` فعال است و سپس پایان زمان گیری را انجام داده و `estimate` می کنیم و نتایج را در فایل ایجاد شده می نویسیم که به صورت زیر است (توجه کنید به عنوان ورودی ادرس `ip 127.0.0.0` را دادیم) :



محتوای فایل Report_Fa'al_Hosts

3) اسکن پورت های باز یک هاست فعال :

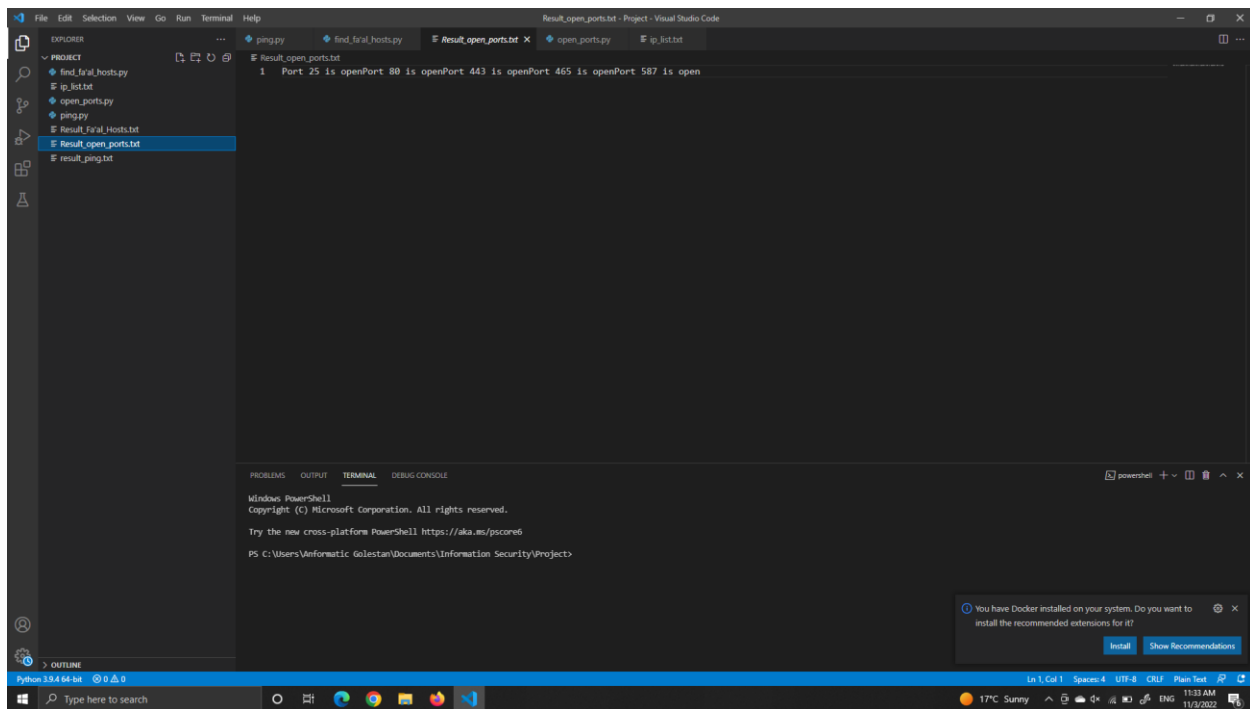
کد این بخش به صورت زیر است :



```
1 import socket
2 from datetime import datetime
3
4 target = input('Enter the remote host IP to scan : ')
5 start_port_number = int(input('Enter The Start Port Number :'))
6 last_port_number = int(input('Enter The last Port Number :'))
7
8
9 print("Scanning Target: " + target)
10 print("Scanning started at:" + str(datetime.now()))
11
12 f = open('Result_open_ports.txt', 'a')
13
14 try:
15     for port in range(start_port_number, last_port_number + 1):
16         s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
17         socket.setdefaulttimeout(0.5)
18
19         # returns an error indicator
20         result = s.connect_ex((target,port))
21         if result == 0 :
22             f.write("Port (port) is open")
23         else :
24             print(f'Port (port) shoma yaft nashod :')
25
26     s.close()
27     f.close()
28 except :
29     pass
30
```

کد بخش سوم

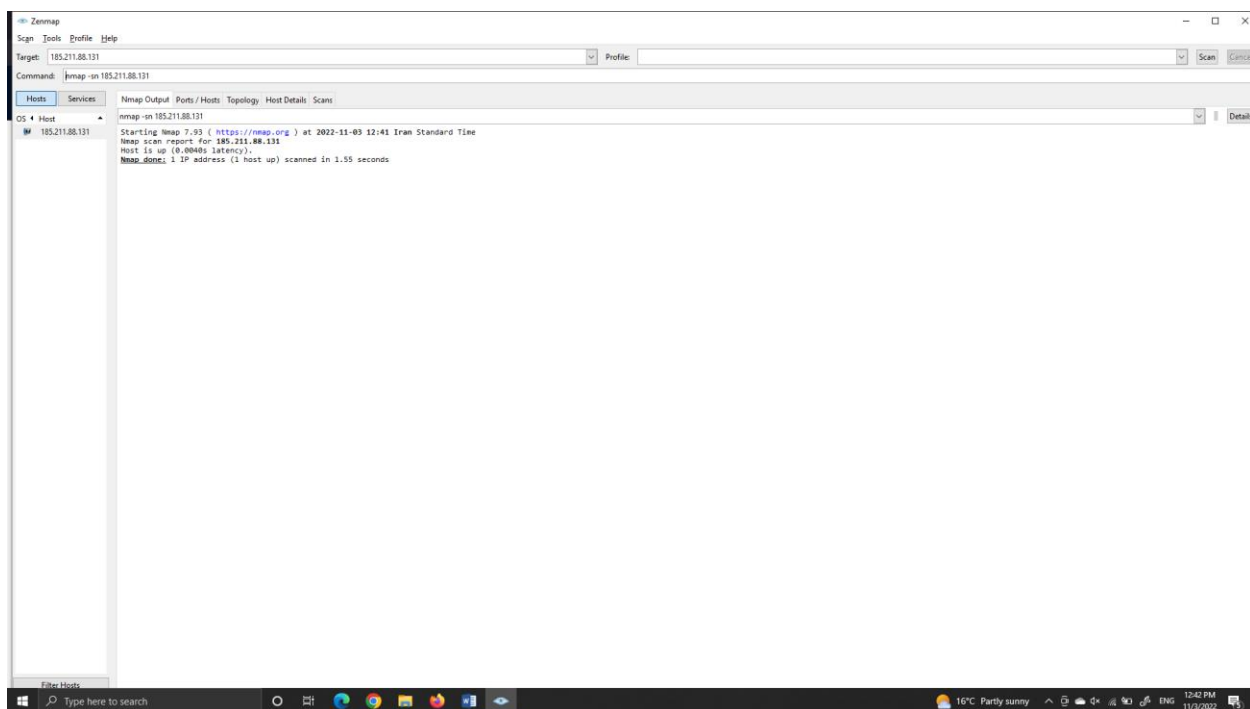
در این بخش از کتابخانه ی socket و datetime استفاده کردیم . سه تا ورودی شامل host ip ، عدد شروع port و عدد پایان آن را گرفتیم. (توجه کنید که اعداد شروع و پایان را به int تبدیل کردیم) سپس host ip و زمان شروع اجرا را print کردیم و یک فایل txt برای ذخیره نتایج درست کردیم. در قسمت اصلی کد یک try – except ایجاد کردیم و در قسمت try آن یک حلقه روی port ها از اعداد شروع تا (پایان + 1) می زنیم. سپس یک socket روی پروتکل tcp درست می کنیم و آن را s می نامیم و default time out آن را روی 0.5 قرار میدهیم. از متد connect_ex که یک tuple از (target, port) می گیرد استفاده میکنیم تا port های فعال را پیدا کنیم و در صورت فعال بودن این method مقدار صفر و در غیر این صورت مقدار یک را برمیگرداند و نتیجه ی آن را در result می ریزیم . result اگر برابر با صفر بود در فایل می نویسیم که port مورد نظر فعال است و در غیر این صورت یافت نشده است سپس سوکت را می بندیم و فایل را نیز می بندیم و در قسمت except چون کاری انجام نمیدهیم pass می گذاریم. نتیجه ی این کد را روی ip سایت اصلی دانشگاه aut.ac.ir در زیر مشاهده می کنید :



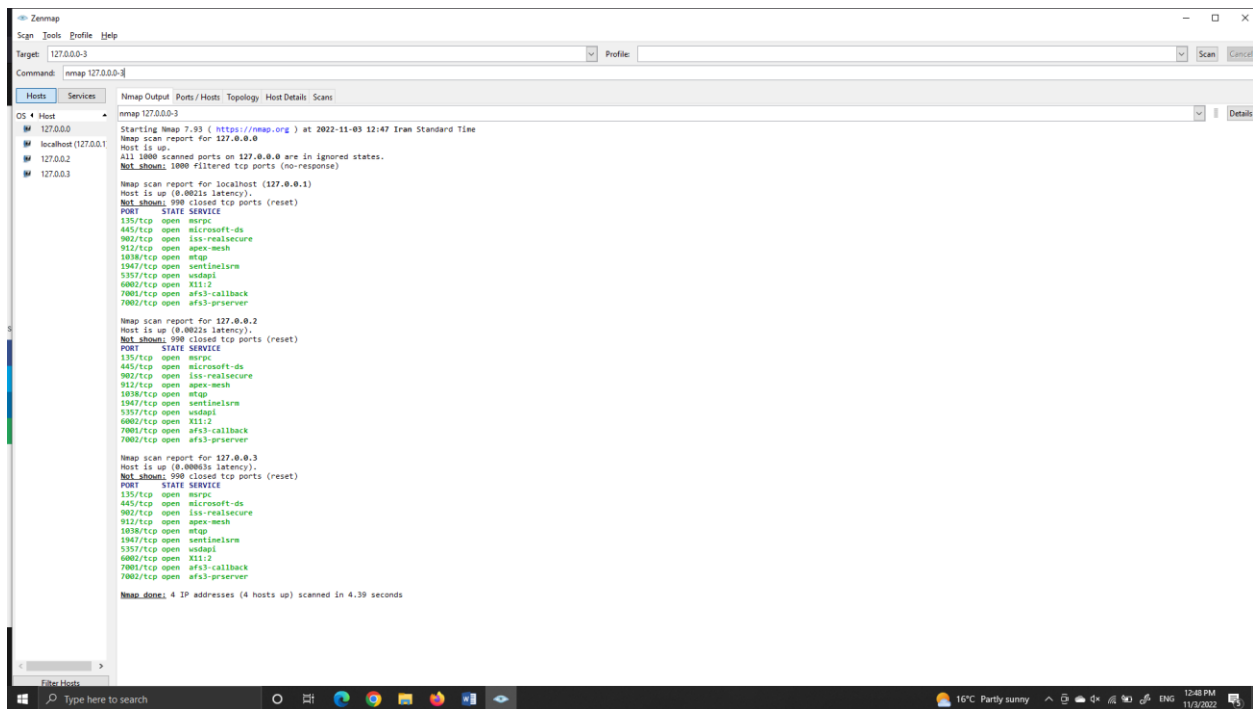
محتوای فایل Result_open_ports

گزارش بخش دوم (

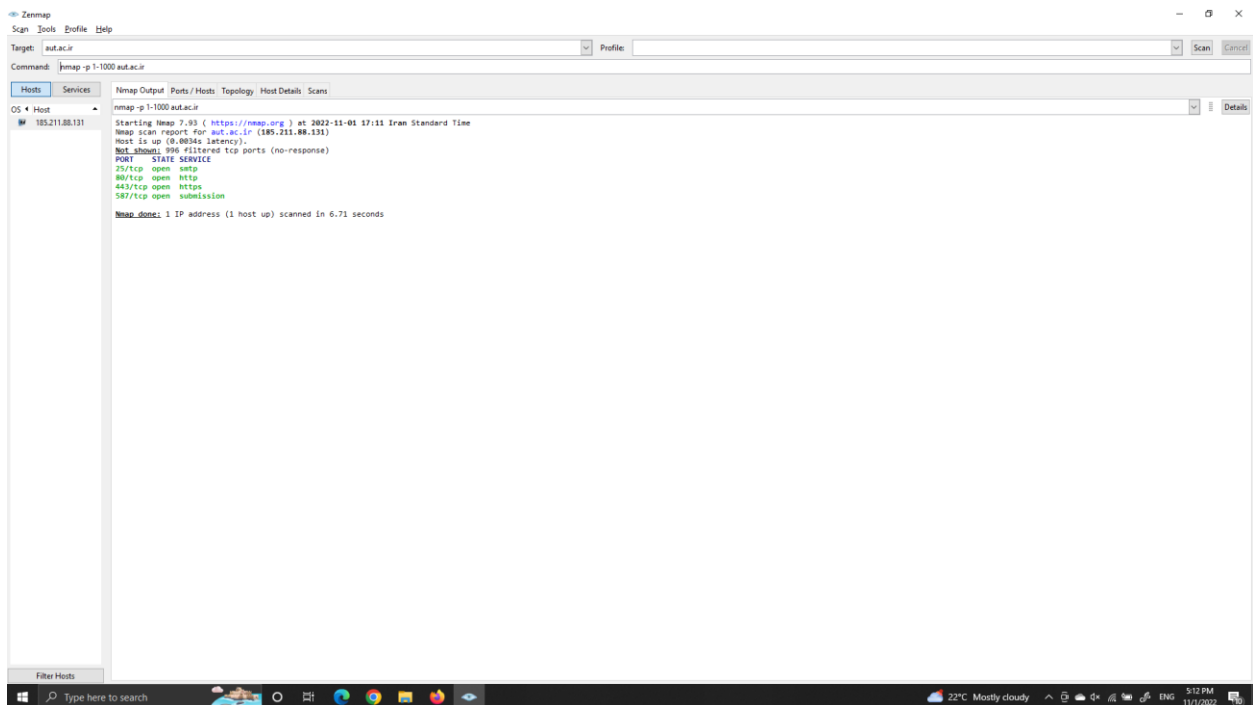
در این قسمت با استفاده از نرم افزار nmap صحت کارهایی که در بخش اول انجام داده ایم را با استفاده از اسکرین شات های زیر بررسی می کنیم :



صحت گرفتن ping از یک آی پی خاص

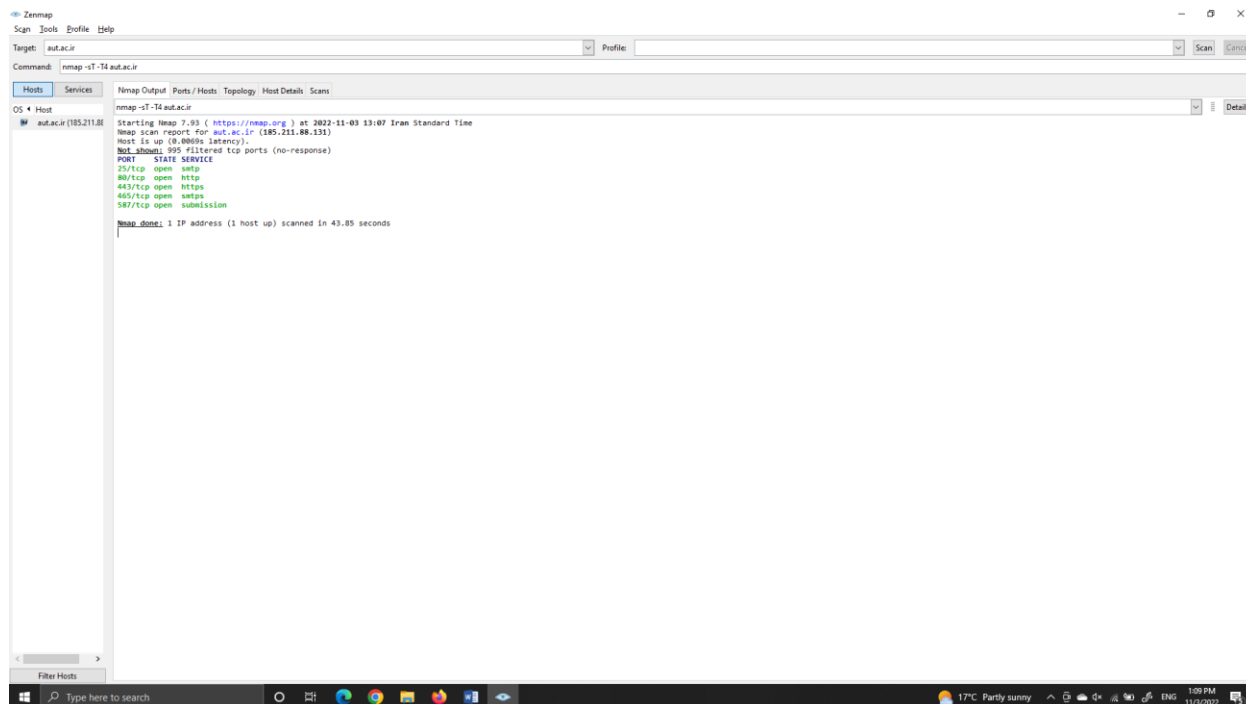


صحت یافتن هاست های فعال یک ip range

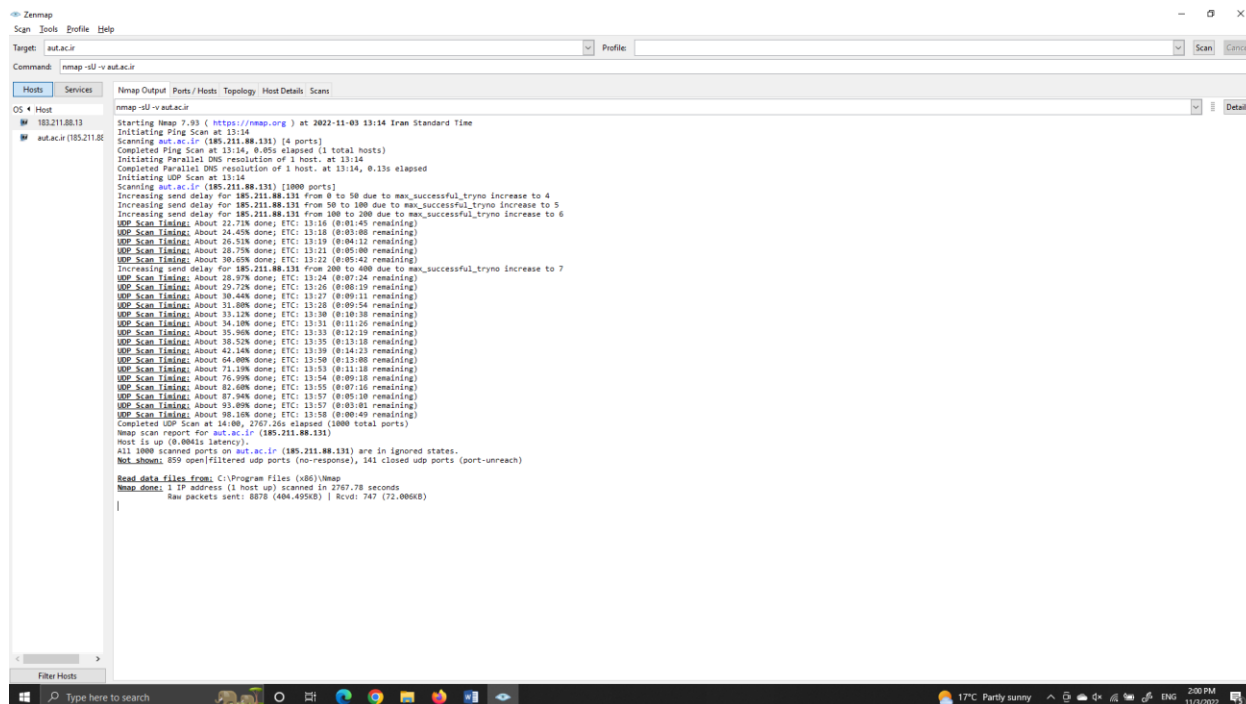


یافتن پورت های باز یک هاست

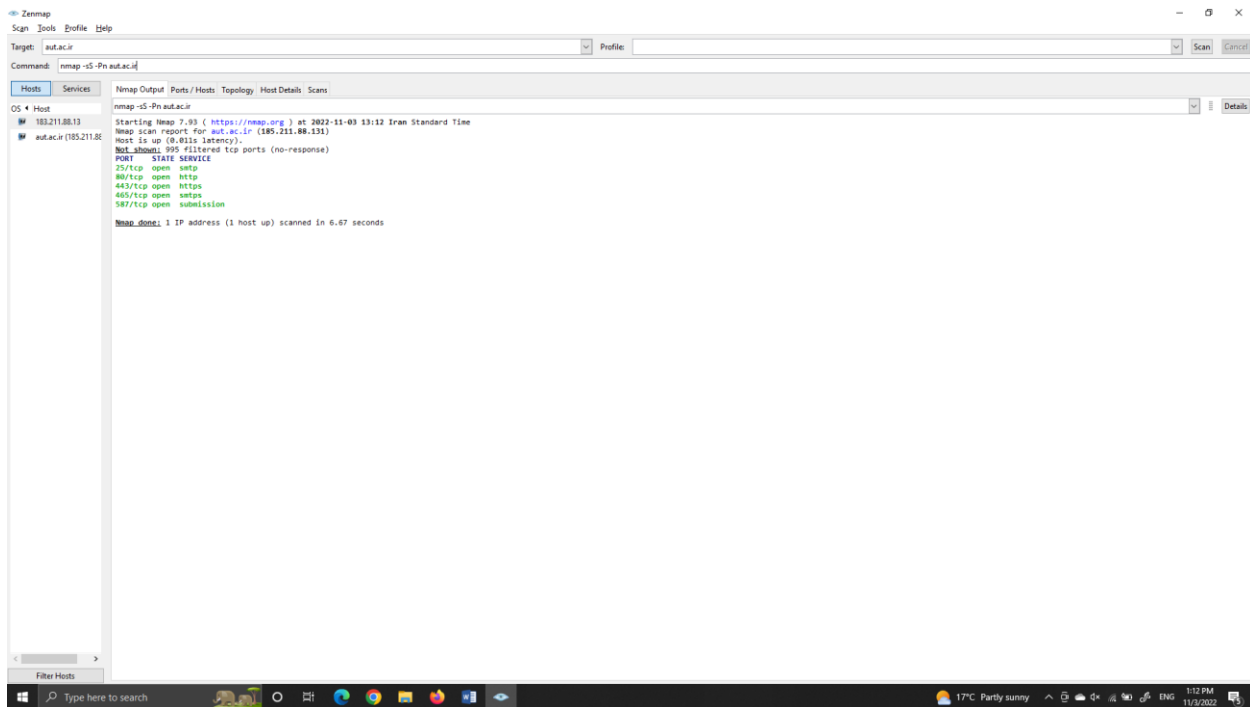
دستورات خواسته شده برای اجرا در nmap :



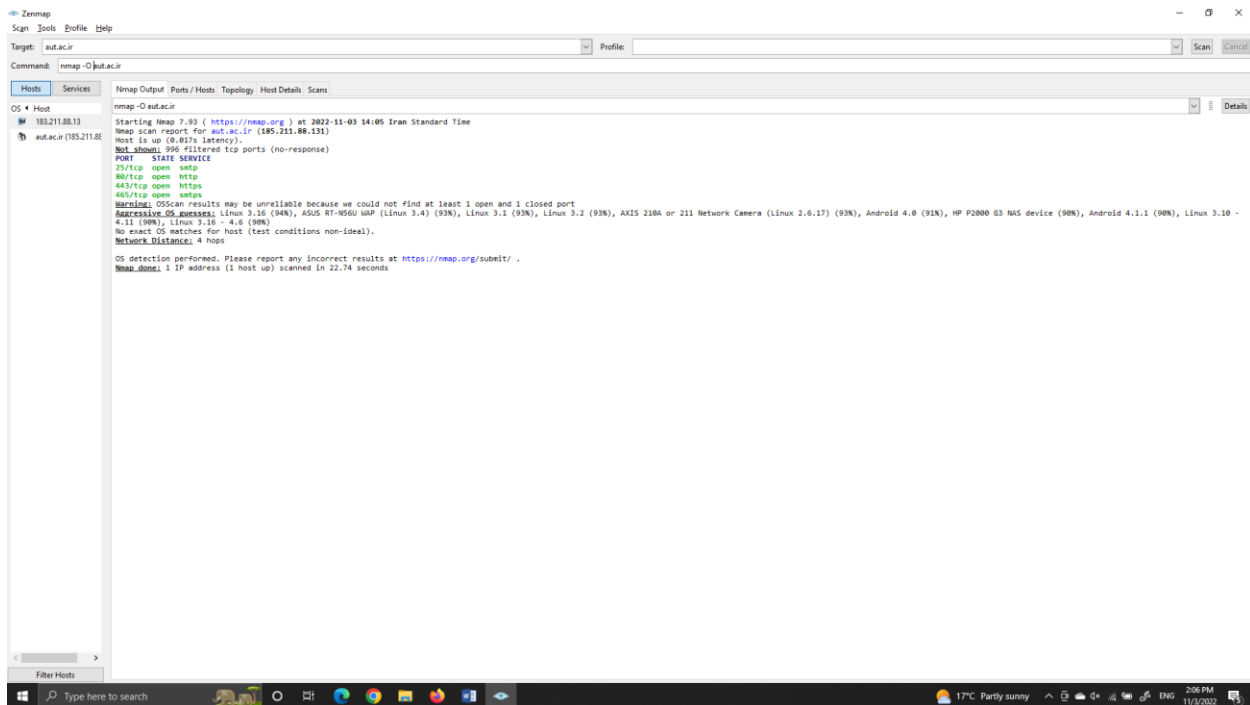
TCP Full Scan



UDP Full Scan



Stealth Scan



Fingerprint (OS) scan