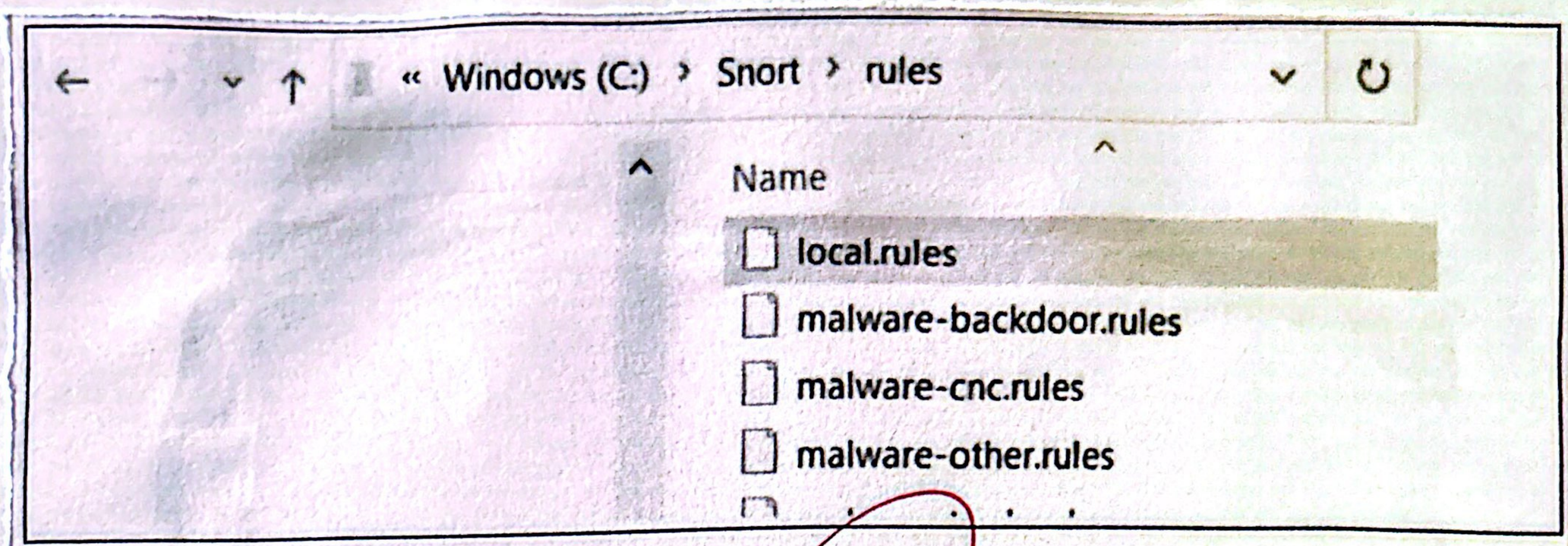


3.1 local.rules file location

C:\Snort\rules\local.rules - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

local.rules

```
1  # Copyright 2001-2023 Sourcefire, Inc. All Rights Reserved.
2  #
3  # This file contains (i) proprietary rules that were created, tested and certified by
4  # Sourcefire, Inc. (the "VRT Certified Rules") that are distributed under the VRT
5  # Certified Rules License Agreement (v 2.0), and (ii) rules that were created by
6  # Sourcefire and other third parties (the "GPL Rules") that are distributed under the
7  # GNU General Public License (GPL), v2.
8  #
9  # The VRT Certified Rules are owned by Sourcefire, Inc. The GPL Rules were created
10 # by Sourcefire and other third parties. The GPL Rules created by Sourcefire are
11 # owned by Sourcefire, Inc., and the GPL Rules not created by Sourcefire are owned by
12 # their respective creators. Please see http://www.snort.org/snort/snort-team/ for a
13 # list of third party owners and their respective copyrights.
14 #
15 # In order to determine what rules are VRT Certified Rules or GPL Rules, please refer
16 # to the VRT Certified Rules License Agreement (v2.0).
17 #
18 #------------
19 # LOCAL RULES
20 #------------
21
22 alert icmp any any -> any any (msg: "testing ICMP"; sid:1000001)
23 alert tcp any any -> any any (msg: "testing ICMP"; sid:1000002)
24 alert udp any any -> any any (msg: "testing ICMP"; sid:1000003)
25
```

3.2. Adding Rules for logging alerts

Date : 14-8-23

Adding Visuals and Getting Reports

Page No. : 6
Expt. No. : 3a

Aim

To add the visuals and get Reports

Procedure

To add the visuals, we need to access "local.rules" located in c:\snort\rules and open it.

Type the following commands

alert icmp any any → any any (msg: "testing ICMP"; sid:1000001)
alert tcp any any → any any (msg "testing TCP"; sid:1000002)
alert udp any any → any any (msg" testing UDP"; sid: 1000003)

Save the local.rules

Open CMD in C:\snort\bin path

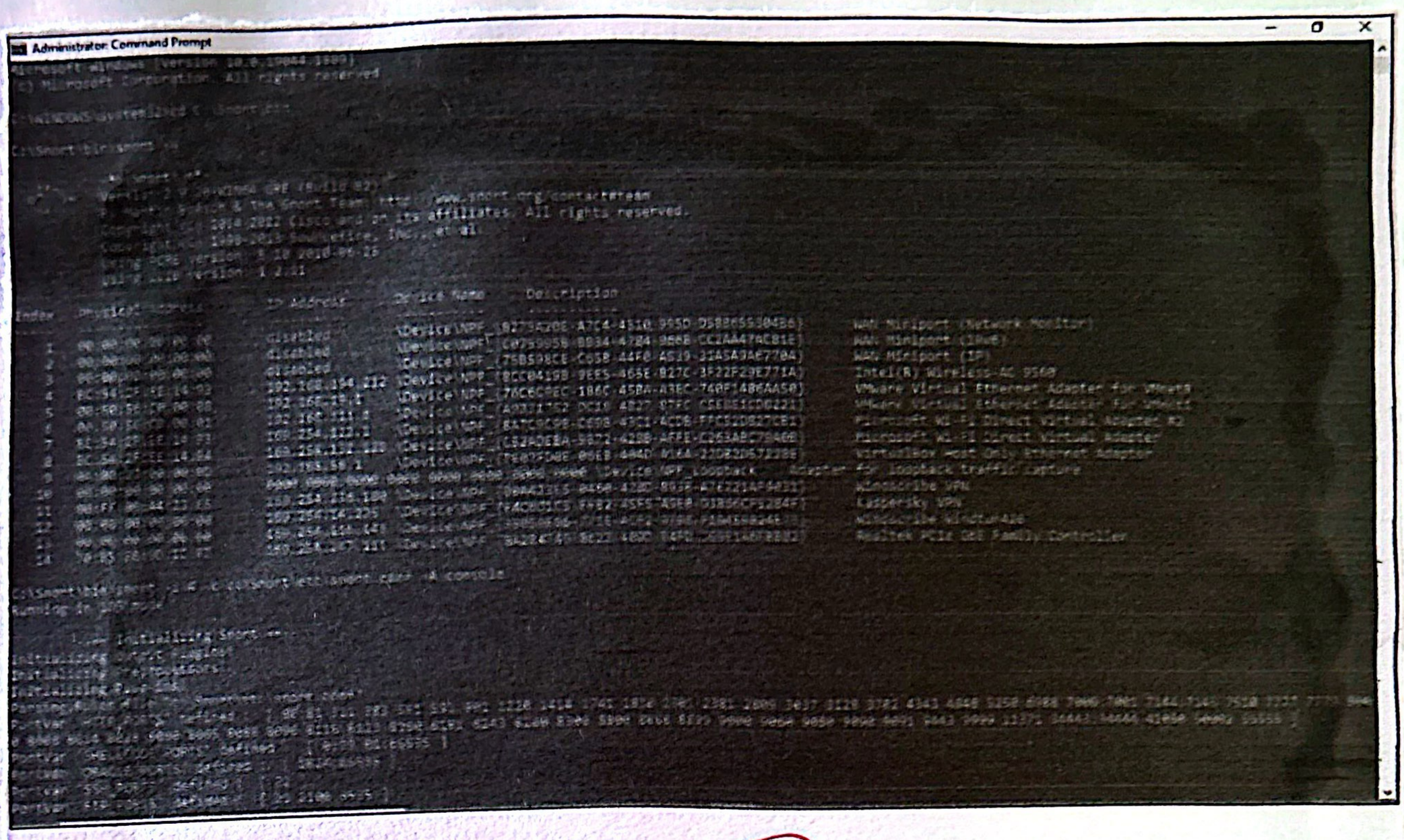
Type Command: Snort -i 3 -c c:\snort\etc\snort.conf

I is interface, C - location of file where you want it to run

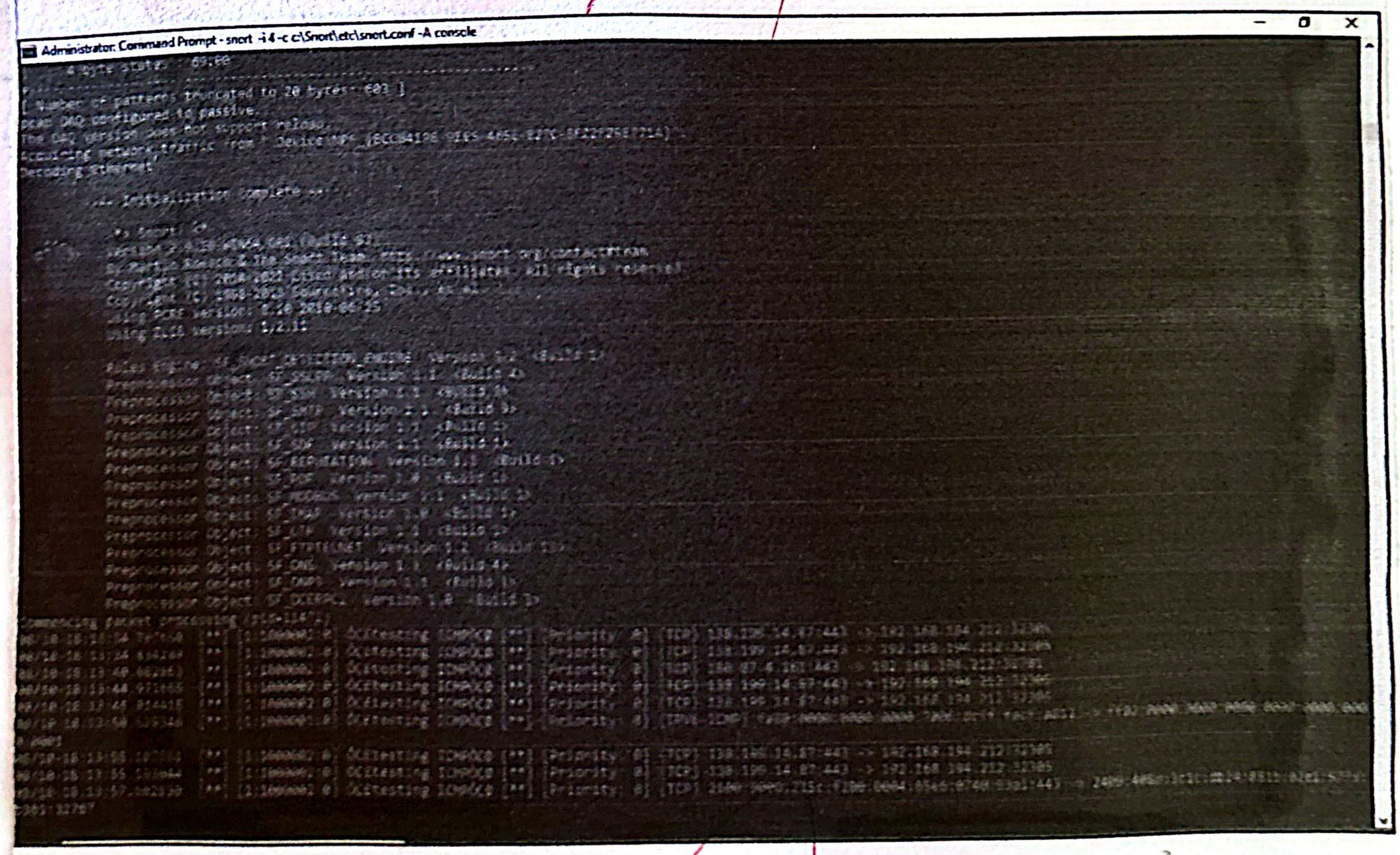
A -> print o/p in terminal

Finally Snort will sniff the network interface specified and all the traffic that's passing through our network

Teacher's Signature:



3.3. Executing Snort Command for log collection



3.4. Collecting logs in network traffic

Date : 14/8/23

Page No. : 7

Expt. No. : 3a

Result

Thus the visuals are added and or modified and the reports are Collected

Teacher's Signature:

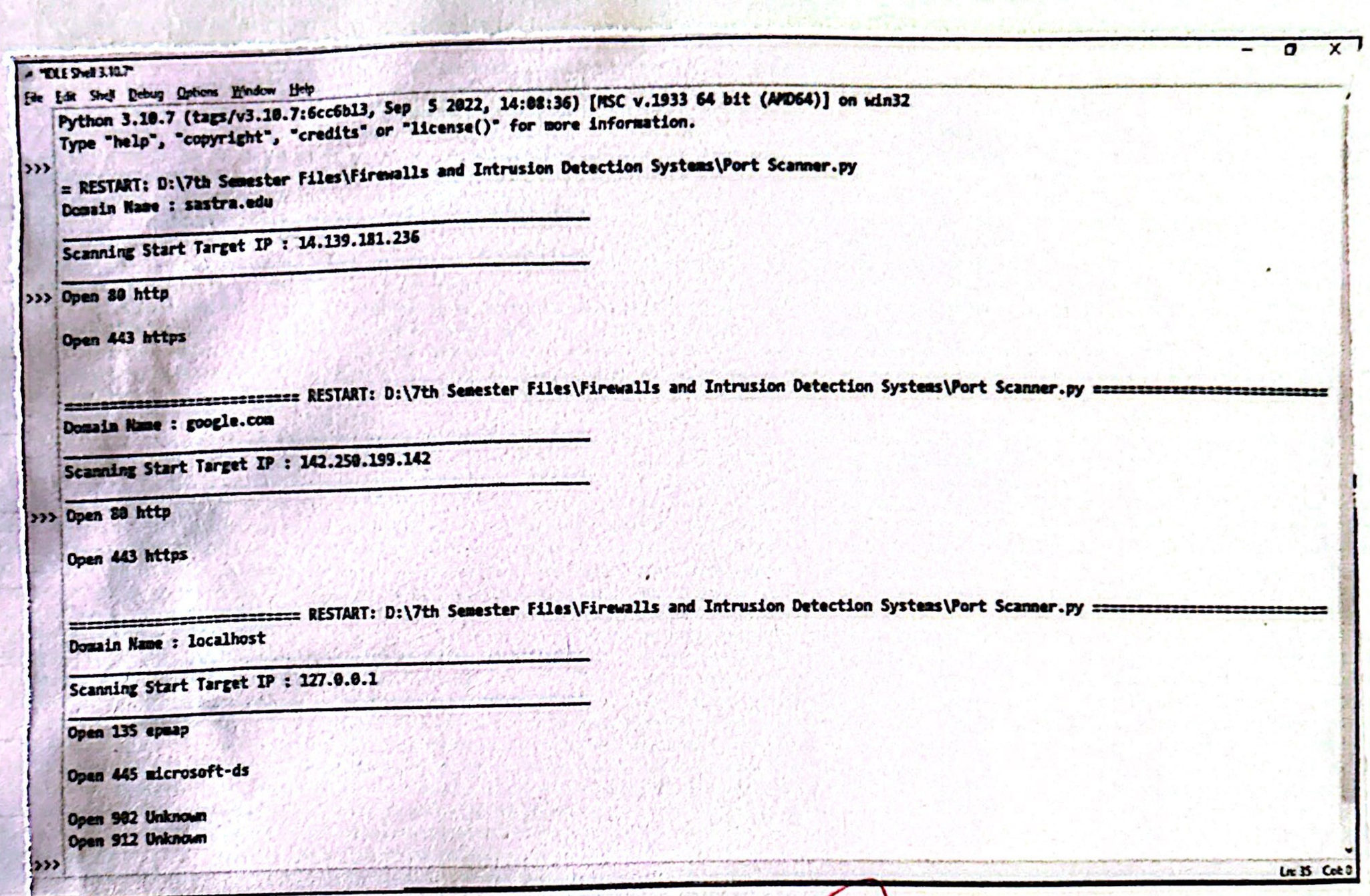


Fig-3a . Security holes in the network

Date : 14/8/23

SubTask - Implementing Port Scan using Python

Page No. : 8
Expt. No. : 3b

Aim To identify security holes in networking ports using python.

Code →

```
import socket, threading, time

Print_lock = threading.Lock()
host = input ("Domain Name")
 ip = Socket.gethostbyname(host)
print ("_"*60)
print (f"Scanning start Target IP : {ip}")
print ("_"*60)


def port_scanner(port):
    Soc = Socket.socket (Socket.AF_INET, socket.SOCK_STREAM)
    Soc.settimeout(1)
    result = SOC.Connect_ex (ip,port)

    if not result:
        try:
            ServiceName = Socket.getservbyport (port, "tcp")
            print (f"Open {port} {serviceName} \n")

        except:
            print (f"Open {port} Unknown \n")
            Soc.close()
for port in range (1,1025):
    th = threading.Thread (target = port_scanner, args = (Port,))
    th.start()
```

Teacher's Signature:

Date : 14/8/23

Page No. : 9

Expt. No. : 3b

Result

Thus, the security holes in the networking ports are successfully identified using python.

Teacher's Signature: