**Mini Project Report on**



**ONLINE HARM DETECTION USING PYTHON**



**Submitted in partial fulfilment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**KRITI AGRAWAL**  **2018893**

***Under the Mentorship of***

**DR. ASHWINI KUMAR SINGH**

**ASSOCIATE PROFESSOR**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**July-2023**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Online Harm Detection using Python”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Dr. Ashwini Kumar Singh, Associate Professor** , Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Kriti Agrawal 2018893

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **1-2** |
| Chapter 2 | Literature Survey | **3-5** |
| Chapter 3 | Methodology | **6-8** |
| Chapter 4 | Result and Discussion | **9-10** |
| Chapter 5 | Conclusion and Future Work | **11** |
|  | References | **12** |

**Chapter 1**

**Introduction**

***In the following sections, a brief introduction and the problem statement for the work has been included.***

* 1. **Introduction**

The Internet has become an important part of our daily lives, and its multiplied use has expanded the hazard of online harm. Online harm can take many forms, including cyberbully, harassment, hate speech, and illegal behavior. In order to prevent and mitigate online harm, it’s important to have effective tools to monitor online activities. Network and port scanning is one such tool that can be used to identify potential sources of online damage on the Internet.

Network scanning is the approach of collecting facts about devices on a computer network through the use of network protocols. It helps to choose which hosts are active, which ports are open, and which operating systems are running on the network. Network administrators use scanning tools to look at network health, identifies potential weaknesses, and maintain network security. Penetration testers and ethical hackers use network scanning to assess network protection and discover potential entry points.

* 1. **Objectives**

The objectives are as follows:

1. Create a network scanning tool using Python and the Scapy module.
2. Use ARP scanning to pick out active hosts at the network.
3. Perform TCP scanning to became aware of open ports on active hosts.
4. Use the python-nmap module to find out the operating system of each active host.
5. Provide a user-friendly interface to input network parameters and show scan results.

**1.3 Scope**

It focuses on developing a network scanning tool using Python and Scapy modules. This includes the implementation of ARP scanning, TCP scanning, and operating system detection. The tool may be designed to look for a particular network range provided by the user. Scan outcomes will be displayed in a user-friendly layout.

**1.4 Significance**

The main importance of this project is its ability to help in preventing and reducing online damage. By identifying potential harm, the tool can help users take appropriate measures to protect themselves and others from damage. The tool will assist us in identifying active hosts, open ports, and operating systems running on the network, allowing administrators to monitor network health, identify potential network security threats, and ensure network security. Network administrators and law enforcement can use this tool to screen and detect illegal activity on the network. Overall, the project has the ability to contribute to a stable and secure online environment.

**Chapter 2**

**Literature Survey**

Network scanning for online harm detection is a well-researched area in the field of computer networks and cybersecurity. Several tools and techniques have been developed to perform network scanning and gather facts about devices on a network.

Network scanning techniques can be extensively categorised into two types -> active scanning and passive scanning. Active scanning entails sending packets to principal target hosts and analysing the responses to gather records. Passive scanning, alternatively, entails monitoring network traffic and analysing it to gather information about devices on the network. Some usually used network scanning techniques include:

* ARP scanning: it involves sending ARP requests to a range of IP addresses to decide if the hosts are active at the network.
* TCP scanning: it involves sending TCP packets to a range of ports on a target host to determine if the ports are open or closed.
* ICMP scanning: it involves sending ICMP packets to a range of IP addresses to determine if the hosts are active on the network or not.
* UDP scanning: it involves sending UDP packets to a range of ports on a target host to determine if the ports are open or closed.

There are several tools available for network scanning, both open-source and commercial. These tools provide an extensive range of features and capabilities for network scanning. Some popular tools include:

* Nmap: Nmap is a widely used open-source network scanning tool that provides a complete set of features for network exploration and security auditing. It supports a plenty of scanning techniques, which include TCP scanning, UDP scanning, and Operating System detection.
* Wireshark: Wireshark is popular open-source packet analyzer which allow users to capture and analyze network traffic. It can be used for passive scanning by monitoring network traffic and analyzing it to gather information about devices on the network.
* Zenmap: Zenmap is a graphical user interface for Nmap that provides an easy-to-use interface for performing network scans. It permits users to configure, scan parameters and view scan results in a user-friendly format.
* Angry IP Scanner: Angry IP Scanner is a lightweight open-source network scanning tool that allows users to scan IP addresses and ports. It provides a simple and intuitive interface for performing network scans.
* Nessus: Nessus is a commercial vulnerability scanner that provides comprehensive scanning capabilities for network security assessments. It can perform active scanning to identify vulnerabilities in network devices.

However, they may have limitations in terms of easy to use, customization, and integration with other tools. The proposed network scanning tool aims to address these limitations by providing a user-friendly interface, customizable scanning techniques, and integration with the Scapy module for packet manipulation.

Scapy is popular Python module which can be used for network scanning. It allows users to interact with packets on the network and create different network tools like ARP Spoofer, Network Scanner, packet dumpers, etc. Scapy can be used to create more advanced tools related to network security and ethical hacking.

We reviewed some literature related to network scanning for online harm detection:

[1] In a tutorial on building a network scanner using Scapy, the author explains how network scanning is an essential element for network administrator as well as a penetration tester. It allows user to map the network in order to find devices that are connected to same network. The tutorial explains how to build a simple network scanner using Scapy library in Python. The author assumes that the reader already has Scapy installed and provides links to tutorials on how to install Scapy

[2] Another tutorial on network scanning with Scapy explains how Scapy can be used to scan the network using ARP requests and to create list of IP address to MAC address mappings. The tutorial also explains how to send SYN packets to a range of port numbers, listen for SYN+ACK replies, and determine which ports are open. The tutorial provides a step-by-step guide on how to create a network scanner using Scapy in Python.

[3] In a blog post on host discovery, the author explains how host discovery is the first step to network reconnaissance. The goal of host discovery is to reduce a large set of IP ranges into a list of active or interesting hosts. The author explains how to use TCP SYN Ping and ICMP Ping to discover hosts on a network. The author also explains how to strike a balance between strict narrowing down and too lenient narrowing down when discovering hosts on a network.

[4] In a blog post on creating a simple network scanner using Python and Scapy, the author explains how to create a virtual environment, install Scapy, and create a scan function. The author also provides a step-by-step guide on how to scan a network using Scapy in Python. The author explains how to use the ARP Ping method to discover hosts on a local Ethernet network.

[5] "Network Scanning with scapy in python" by Zhang zeyu

This article on DEV Community discusses the process of gathering information about devices in a computer network through network scanning. It provides an overview of protocols used in network communications and the OSI model. The article also explains how to install and use Scapy for network scanning.

[6] "Simple Network scanner with python and scapy" by Maksym Postument  
This Medium post demonstrates how to create a simple network scanner using Python and Scapy. It provides step-by-step instructions on setting up a virtual environment, installing Scapy, and coding the network scanning functionality. The post focuses on using the ARP Ping method for discovering hosts on a local Ethernet network.

[7] "Network scanning using scapy Module - Python"  
This article on GeeksforGeeks explains the functionality of Scapy, a library for interacting with packets on the network. It discusses the creation of network tools using Scapy, such as ARP Spoofer, Network Scanner, and packet dumpers. The article also provides installation instructions for the Scapy module.

[8] "How to make a network scanner using scapy (Python)"  
This tutorial demonstrates how to build a simple network scanner using ARP requests and the Scapy library in Python. It explains the importance of network scanning for network administrators and penetration testers. The tutorial provides step-by-step instructions on using Scapy to scan the network and discover connected devices.

**Chapter 3**

**Methodology**

The main purpose of this project is to build a system automatic port scanning process to scan the network ports of the target information system about target hosts, listening ports, and services running through ports. There are too many network scans, network development tools, and Remediation Tools. these communication tools that are publicly disclosed could be used by purposeful criminals and attackers for a vicious reason. One of the tools is analysed in my system is the Nmap tool. As a result of scanning the host, point to the open ports again fo services and related IP.

A high-speed port scanner, ZMap, was used to perform extensive Internet scanning on Port 80 in an attempt to determine the size of the Web. The size of the Web was estimated from the number of web servers detected during high-speed IPv4 scanning addresses. The focus of the study was to determine whether the census was accurate in a enough time using ZMap. Metrics to determine the success of this research have been a reduction in the time required to perform extensive internet scanning, in comparison historical experiments, and comparisons between scanning data collected in this study by data collected by Rapid7 Labs to check the accuracy of ZMap scanning.

This report introduces our effort and identifies possible uses of the embedded Linux platform for access (Port Scan). The method used was to develop software that enables port scanning using the slow opening method and udp. The software is then used on Linux based Single Board Computer (SBC) using the TS-Linux 2.4.23 kernel developed by Technology System (TS). It is interesting enough to find that despite the limitations of processing power, the performance of the system in embedded space is similar to other port scanners running on a PC that works much better. The findings show that the low-cost Linux embedded platform is suitable for secure network use and is sold at low cost with the added portable benefit.

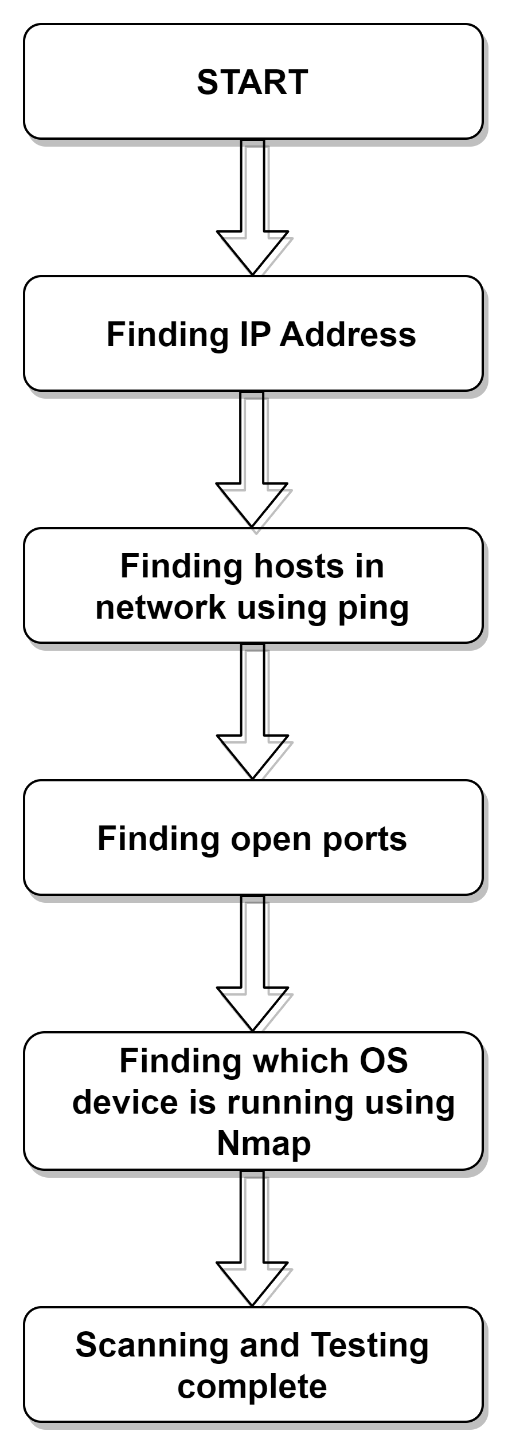
Today The Online Harm Detection System is a growing technology in network security, and especially researchers have focused on this field, some of them using a legal signature or process and some strategies that are uniquely based to improve network security. In this paper, we propose a Legal Entity Acquisition Principle for which we have developed the New port Scan Detection Act (EPSDR). These rules will be used to detect port scan naive attacks on a real-time network using the Snort and Basic Analysis Security Engine (BASE).

The methodology consists of the following steps:

* Install Scapy and other vital modules: The first step is to install the Scapy module and different required modules, such as nmap. Scapy is a effective packet switching device that permits us to create, send, receive network packets.
* Find the IP address of the host: The next step is to locate the IP address of the host. This can be done with the socket module in Python. the IP address will be used to determine the network prefix.
* Host discovery: The next step is to use Scapy to discover all the hosts on the network. This involves sending ARP requests to all IP addresses in the user-provided network range. The Ether and ARP layers in Scapy will be used to send ARP requests.
* TCP scanning: Once all the active hosts have been identified, the next step is to scan for open ports using TCP scanning. This requires sending TCP packets to all the ports on the active hosts. a response indicates that the port is open.
* Operating system Detection: The final step is to use nmap to detect the operating system of the active host. This involves sending probes to active hosts and analysing the responses to determine the activity pattern.
* Identification of potential sources of online harm: Based on the network scan result, potential sources of online harm will be identified. This involves analysing open ports and operating systems in active hosts to determine if they pose a threat.
* Reporting: The final step is to generate the network scan results. The output will include information about active hosts, open ports, and operating systems. It will identify potential sources of online harm and offer recommendations to mitigate risks.

So summarising our methodology, it involves installing the required modules, discovering all the hosts in the network, scanning for open ports, detecting the operating system of the active hosts, identifying potential sources of online harm, and generating a report of the network scan results. The project will utilize various scanning techniques to provide information about the network and identify potential sources of online harm.

**SYSTEM ARCHITECTURE**

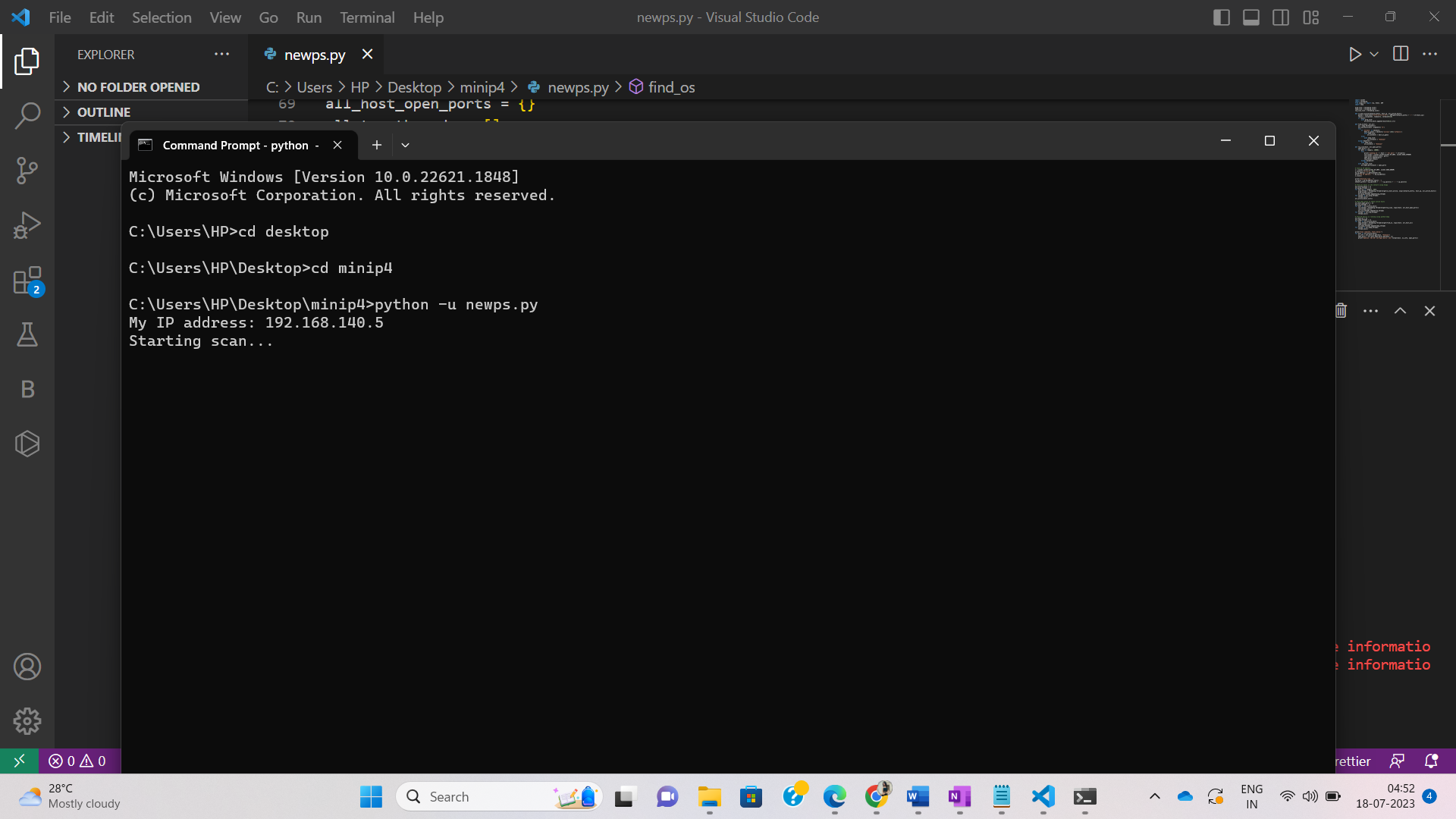
****

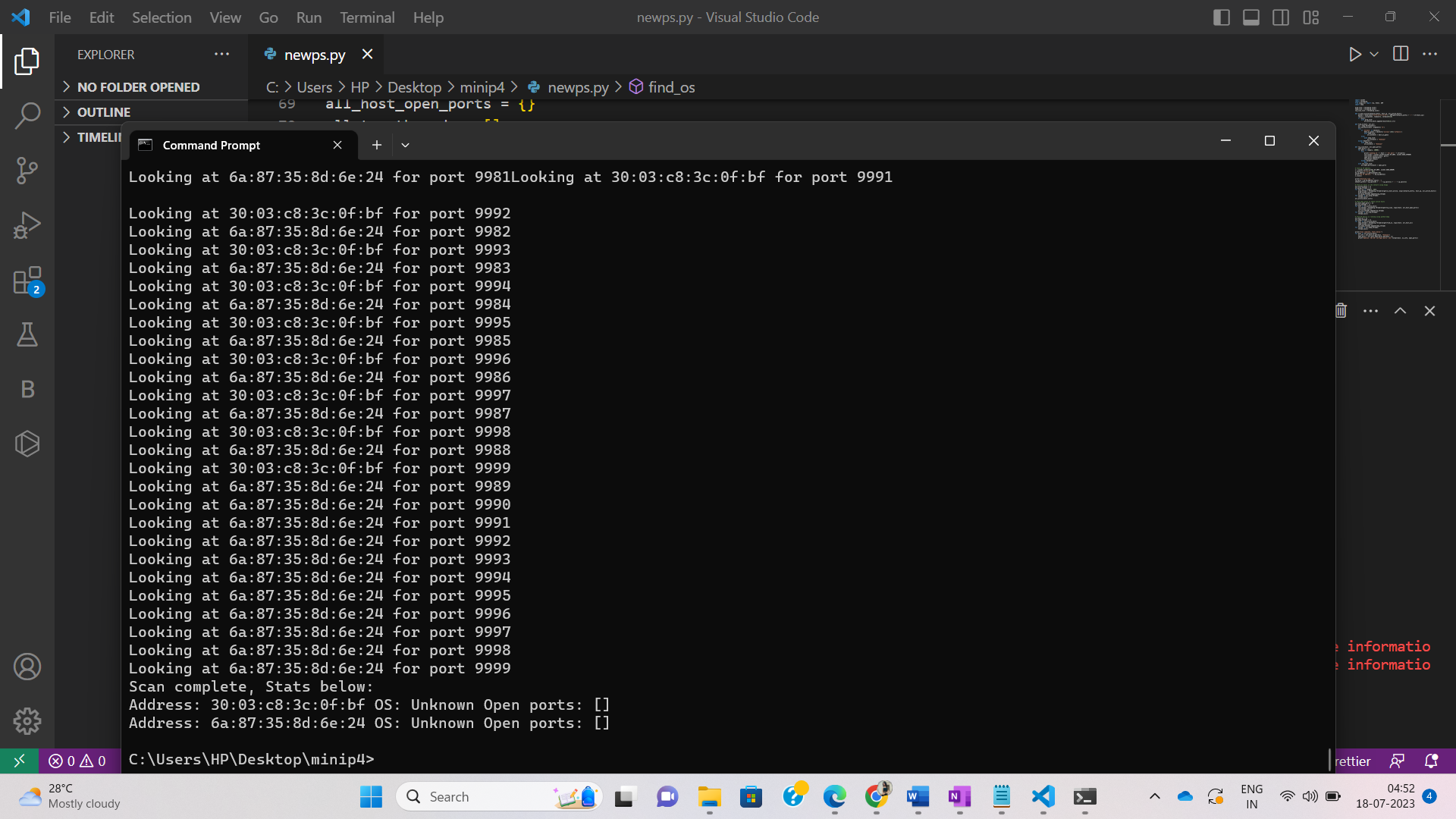
**Chapter 4**

**Result and Discussion**

Port scanner for Online harm detection is an application designed to detect a server or host for open ports. Such an application can be used by administrators to verify the security policies of their networks and by attackers to identify network services that work on hackers and exploit the risk.

Our proposed tool successfully identifies potential sources of online harm by detecting active hosts, open ports, and the operating system running on each device. The scan results provide valuable information for further analysis and mitigation of online harm.





It shows the addresses of two devices on the network, but it was not able to identify the operating system or any open ports on either device. The absence of open ports means that there are no services running on these devices that are accessible from the network.

**Calculations**:

Total addresses = A

Time to discover H hosts = A/T

Discovered hosts = H

Number of ports checked = P

Time to discover H hosts

Time to connect to P port = H \* P

If we execute the program serially, the time complexity would be O(H \*P).

But because of parallel execution on T threads, time complexity is O((H \* P)/ T

H = O(A)

Total time = A/T + (H\*P)/T

= A/T + (A\*P)/T

= (A\*P)/T

**Chapter 5**

**Conclusion and Future Work**

In our project, we first identified the IP addresses of various devices, then found all the hosts in network using ping, and we used threading so as to detect fast Operating System. Without a port scanner Some ports might remain continually open, presenting a potential network vulnerability. An intruder can access an open port to create difficulties in the normal flow of network operations. network ports should be carefully monitored by means of an effective advance network port scanning device to keep away from any information leakage. This additionally helps secure communications among the computing entities inside the network.

Our project was telling all the open ports and dangerous ports, so we are planning to improve the system so that it automatically highlights all the dangerous ports and closes them. Running port scans without authorization can be considered an aggressive action, and if we are on a shared network, we might scan a system that isn’t under our control, which isn’t good. Port scans are an important part of constructing a good defence from cyberattacks. Attackers use port scans, as well. We have to beat them to the punch and close down viable attack vectors, and make their lives as difficult as possible.

More Future work could include:

* Enhancing the tool's functionality to include additional scanning techniques, such as UDP scanning and service detection.
* Implementing ML algorithms to analyse network traffic and identify patterns of online harm.
* Developing user-friendly interface for tool to make it more accessible to network administrators and security professionals.

**References**

[1] Zhang Zeyu. "Network Scanning with Scapy in Python." Plain English Python, 2 Aug. 2020.

<https://python.plainenglish.io/network-scanning-with-scapy-in-python-708ed176e63>

[2] Postument, Maksym. "Simple Network Scanner with Python and Scapy." Medium, 3 Mar. 2019, <https://mpostument.medium.com/simple-network-scanner-with-python-and-scapy-802645073190>

[3] "Network Scanning using scapy module - Python." GeeksforGeeks, 1 Mar. 2020. <https://www.geeksforgeeks.org/network-scanning-using-scapy-module-python/>

[5] "Network Scanning with Scapy into Python." DEV Community, 1 Aug. 2020. <https://dev.to/zeyu2001/network-scanning-with-scapy-in-python-3off>

[6] <https://www.researchgate.net/publication/220195237_A_review_of_port_scanning_techniques>

[7] <https://www.researchgate.net/publication/345959645_Automatic_Port_Scanner>

[8] <https://en.cnki.com.cn/Article_en/CJFDTotal-JZCK201007063.htm>

[9] <https://www.osti.gov/biblio/4338629>

[10] <https://ieeexplore.ieee.org/abstract/document/4786717>