PORT Scanner using Python

Project Report

Industrial Training Seminar (IDS553)

B.Tech CS&E DATA SCIENCE
(In Collabration with I-nurture)

SUBMITTED BY :
Sahil Jain (TCA2166016)

DEC. , 2023
Session 2023-24

FACULTY OF ENGINEERING & COMPUTING SCIENCES

TEERTHANKER MAHAVEER UNIVERSITY, MORADABAD

DECLARATION

We hereby declare that this Project Report titled **PORT SCANNER WITH PYTHON** submitted by me and approved by our project guide, Faculty of Engineering & Computing Sciences. Teerthanker Mahaveer University, Moradabad, is a bonafide work undertaken by us and it is not submitted to any other University or Institution for the award of any degree diploma / certificate or published any time before.

Project Name	Student Name	Signature
Port Scanner With Python	Sahil Jain	

Table of Contents

Appendix

A : Data Flow Diagram (DFD)

B : Entity Relationship Diagram (ERD)

C : Use Case Diagram (UCD)

D : Data Dictionary (DD)

E : Screenshots

1 Port Scanning

Port scanning is a method of **determining which ports on a network are open and could be receiving or sending data**. It is also a process for sending packets to specific ports on a host and analyzing responses to identify vulnerabilities.

2 Problem Statement

A port vulnerability scanner is an application designed to probe a server or host for open ports. Most scanners run for a period of time, assessing open ports on a host and then producing a report to identify potential security compromises on the enterprise systems scanned to the end user.

Scanners were previously known to cause issues for all ITM applications; specifically, the component would stall.

2.1 Symptom

Port scanners typically establish multiple connections in order to run various tests designed to detect security vulnerabilities.
During a scan, messages such as the following are typically written to the ITM application's RAS log:

(612E1A14.0004-30:kdebbrx.c,44,"KDEB_BaseReceive") Status
1DE0000B=KDE1_STC_DISCONNECTED=104: Connection reset by peer
(612E1A14.0005-30:kdebbtx.c,47,"KDEB_BaseTransmit") Status
1DE0000B=KDE1_STC_DISCONNECTED=32: Broken pipe
(612E1A14.0006-30:kdebeal.c,81,"ssl_provider_open") GSKit error 410: GSK_ERROR_BAD_MESSAGE - errno 32
(612E1A14.0046-60:kdebp0r.c,240,"receive_pipe") Status
1DE00074=KDE1_STC_DATASTREAMINTEGRITYLOST
(612E1A14.0047-60:kdeprxi.c,82,"KDEP_ReceiveXID") Status 1DE0003C=KDE1_STC_RECEIVEXIDFAILURE

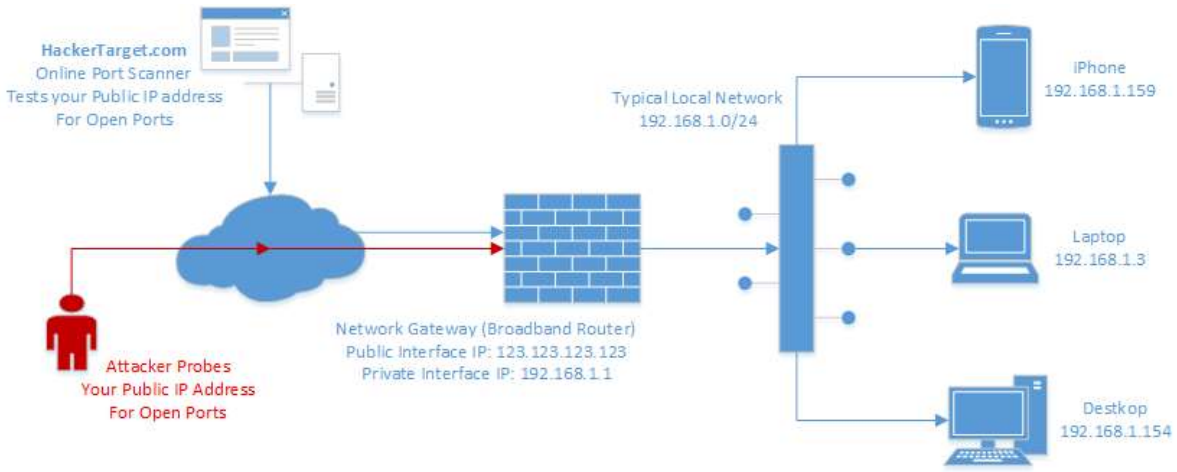
3 Project Description

The defect was resolved by APAR IJ21264 which was made generally available in 6.3.0.7 Service Pack 3 (or higher)
Ensure 6.3.0.7 Service Pack 3 (or higher) is installed. The service pack can be applied to the TEMS, TEPS and all agents in the environment. If you have non-OS agents, ensure that the agent framework is at SP3 (or higher) level.

3.1 Scope of the Work

The defect was resolved by APAR IJ21264 which was made generally available in 6.3.0.7 Service Pack 3 (or higher)
Ensure 6.3.0.7 Service Pack 3 (or higher) is installed. The service pack can be applied to the TEMS, TEPS and all agents in the environment. If you have non-OS agents, ensure that the agent framework is at SP3 (or higher) level.Project Modules.

3.2 Context Diagram (High Level)



4 Implementation Methodology

A port is a point on a computer where information exchange between multiple programs and the internet to devices or other computers takes place. To ensure consistency and simplify programming processes, ports are assigned port numbers. This, in conjunction with an IP address, forms vital information that each internet service provider (ISP) uses to fulfill requests.

Port numbers range from 0 through to 65,536 and are ranked in terms of popularity. Ports numbered 0 to 1,023 are called “well-known" ports, which are typically reserved for internet usage but can also have specialized purposes. These ports, which are assigned by the Internet Assigned Numbers Authority (IANA), are held by leading businesses and Structured Query Language (SQL) services.

Ports are generally managed by the Transmission Control Protocol (TCP), which defines how to establish and maintain a network conversation between applications, and User Datagram Protocol (UDP), which is primarily used for establishing low-latency and loss-tolerating connections between applications. Some of the most popular and most frequently used ports include:

1. Port 20 (UDP): File Transfer Protocol (FTP) used for transferring data
2. Port 22 (TCP): Secure Shell (SSH) protocol used for FTP, port forwarding, and secure logins
3. Port 23 (TCP): The Telnet protocol used for unencrypted communication
4. Port 53 (UDP): The Domain Name System (DNS), which translates internet domain names into machine-readable IP addresses
5. Port 80 (TCP): The World Wide Web Hypertext Transfer Protocol (HTTP)

Technologies to be used

4.1 Software Platform

a) Front-end

b) Back-end

4.2 Hardware Platform

RAM, Hard Disk, OS, Editor, Browser etc.

4.3 Tools, if any

1. Ping scans: A ping scan is considered the simplest port scanning technique. They are also known as internet control message protocol (ICMP) requests. Ping scans send a group of several ICMP requests to various servers in an attempt to get a response. A ping scan can be used by an administrator to troubleshoot issues, and pings can be blocked and disabled by a firewall.

2. Vanilla scan: Another basic port scanning technique, a vanilla scan attempts to connect to all of the 65,536 ports at the same time. It sends a synchronize (SYN) flag, or a connect request. When it receives a SYN-ACK response, or an acknowledgment of connection, it responds with an ACK flag. This scan is accurate but easily detectable because a full connection is always logged by firewalls.

3. SYN scan: Also called a half-open scan, this sends a SYN flag to the target and waits for a SYN-ACK response. In the event of a response, the scanner does not respond back, which means the TCP connection was not completed. Therefore, the interaction is not logged, but the sender learns if the port is open. This is a quick technique that hackers use to find weaknesses.

- XMAS and FIN scans: Christmas tree scans (XMAS scans) and FIN scans are more discrete attack methods. XMAS scans take their name from the set of flags that are turned on within a packet which, when viewed in a protocol analyzer like Wireshark, appear to be blinking like a Christmas tree. This type of scan sends a set of flags, which, when responded to, can disclose insights about the firewall and the state of the ports. A FIN scan sees an attacker send a FIN flag, often used to end an established session, to a specific port. The system's response to it can help the Project Repository Location
- When an attacker scans only a few ports, say less than 20 ports, in a given time, it's referred to as the strobe mode of port scanning. On the other hand, when an attacker listens to a port for a longer duration, say for one month, and gradually executes port scanning, it's known as the stealth mode. In both the modes, the attacker goes unnoticed.

5 Definitions, Acronyms, and Abbreviations

An attacker tries to connect to the target host by communicating with all the 65536 available system ports. Firewalls respond to this attack in one of three ways, depending on the status of the port:

- If the port is open, it redirects the traffic to the specific host.
- If the port is closed, the traffic isn't redirected. However, the firewall responds with a "Denied" notification.
- If the port is blocked by your firewall, it doesn't respond to the request.

6 Conclusion

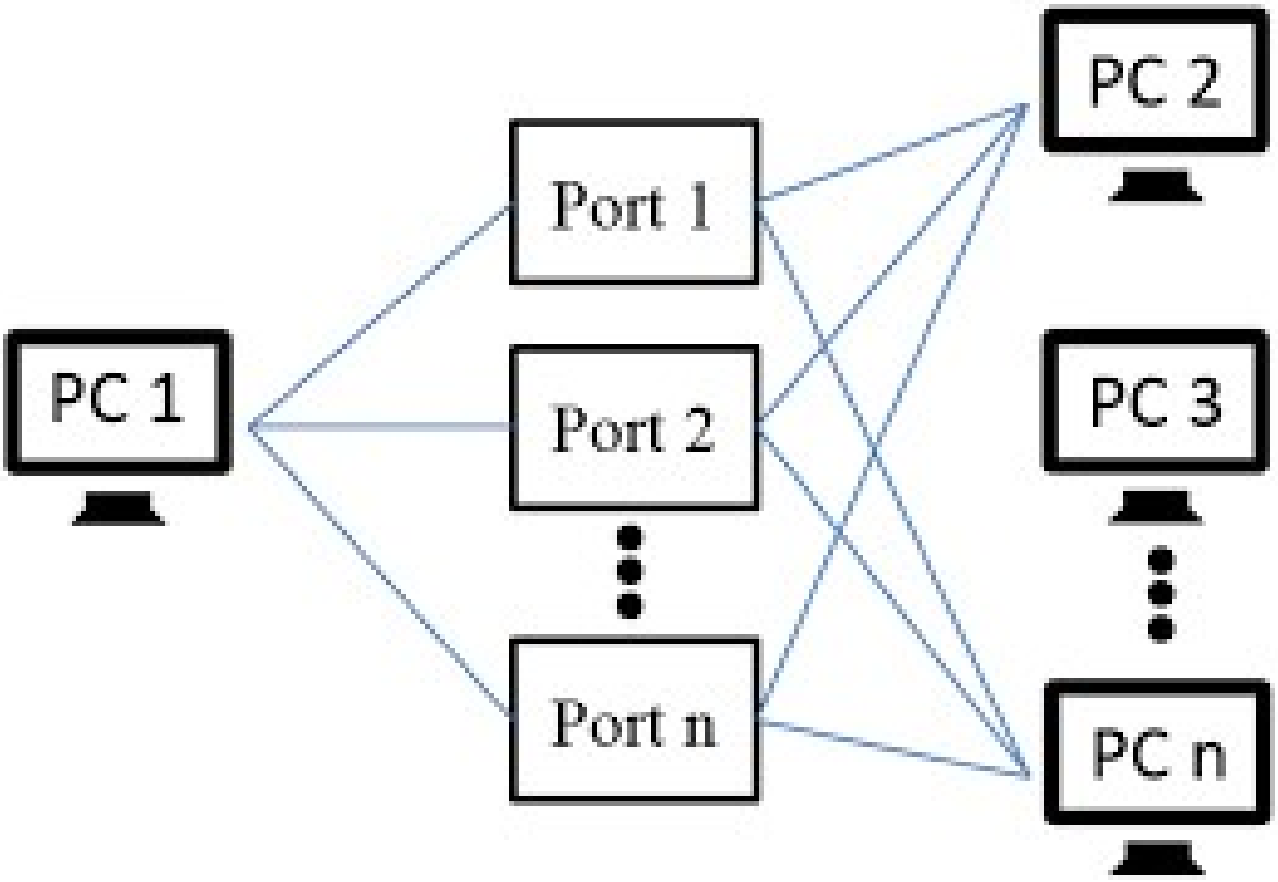
Computer ports are crucial components in application programming and networking since they provide a central docking point for exchanging information between two entities. A port number provides consistency and is combined with the target host IP address to form the vital information that the internet service provider uses to fulfill requestsg is a common technique used by hackers to identify open ports that can be used as attack vectors on the remote host. The intrusion technique often follows the host discovery phase and is used to reveal the presence of security devices between the sending and listening ports.
In this article, we discuss what a port scan

7 References

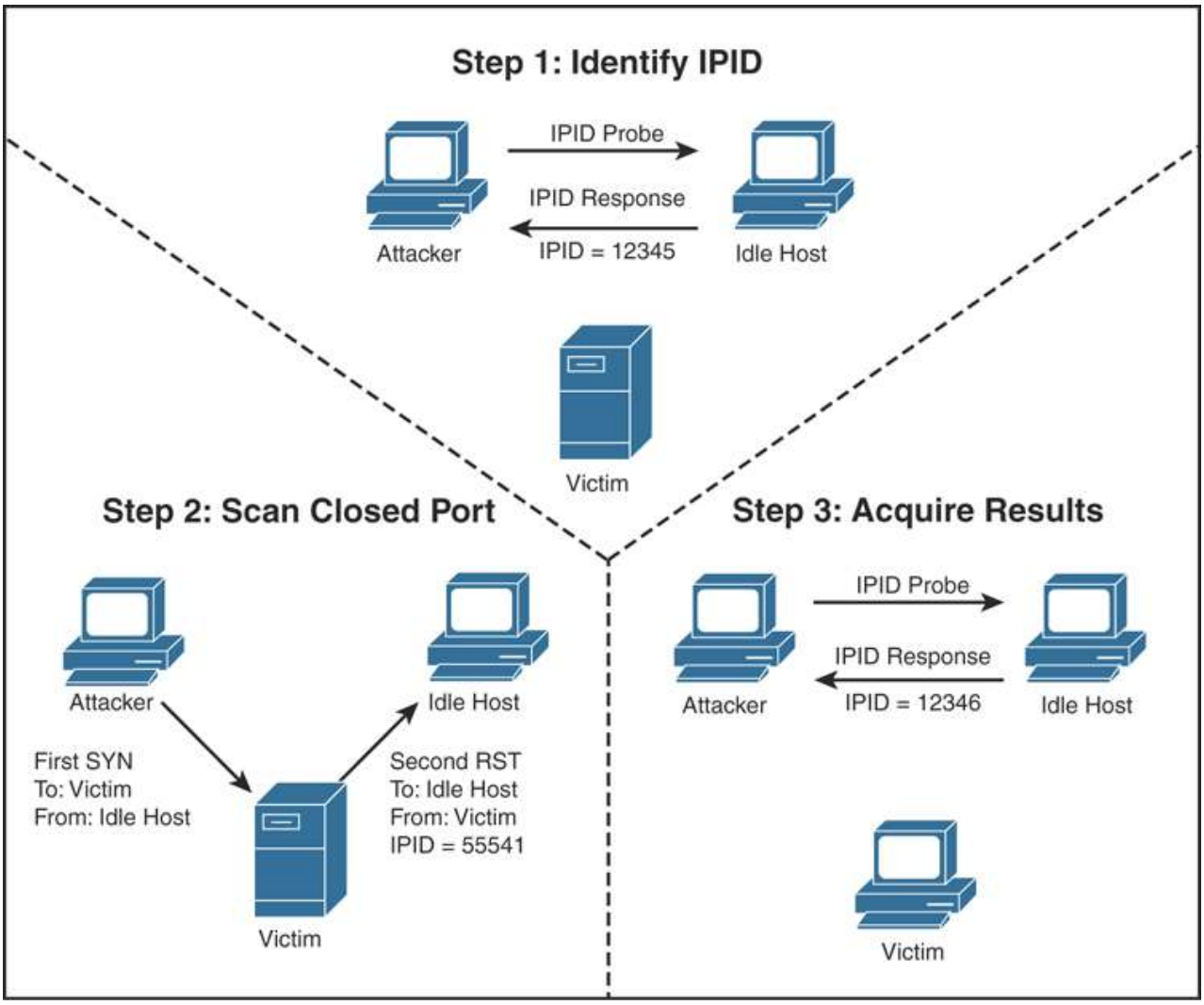
A port is a communication endpoint that facilitates data transfer between two devices, or an application and a device. If a port is open, it is being used for a particular service or application and is actively listening to requests sent to that application. If the applications using open ports aren't patched well, these ports can be exploited and used for launching attacks. A port scan is a method that is used to spot open ports on a network. Running a port scan reveals the open ports in the network and network security devices such as firewalls deployed between the sender and the receiver.

SR No.	Reference Details	Owner	Version	Date
1.	Port Scanner with Python		1.0	
2.	VS Code , pycharm , spider		1.0	

Annexure A
Data Flow Diagram (DFD)
(Mandatory)

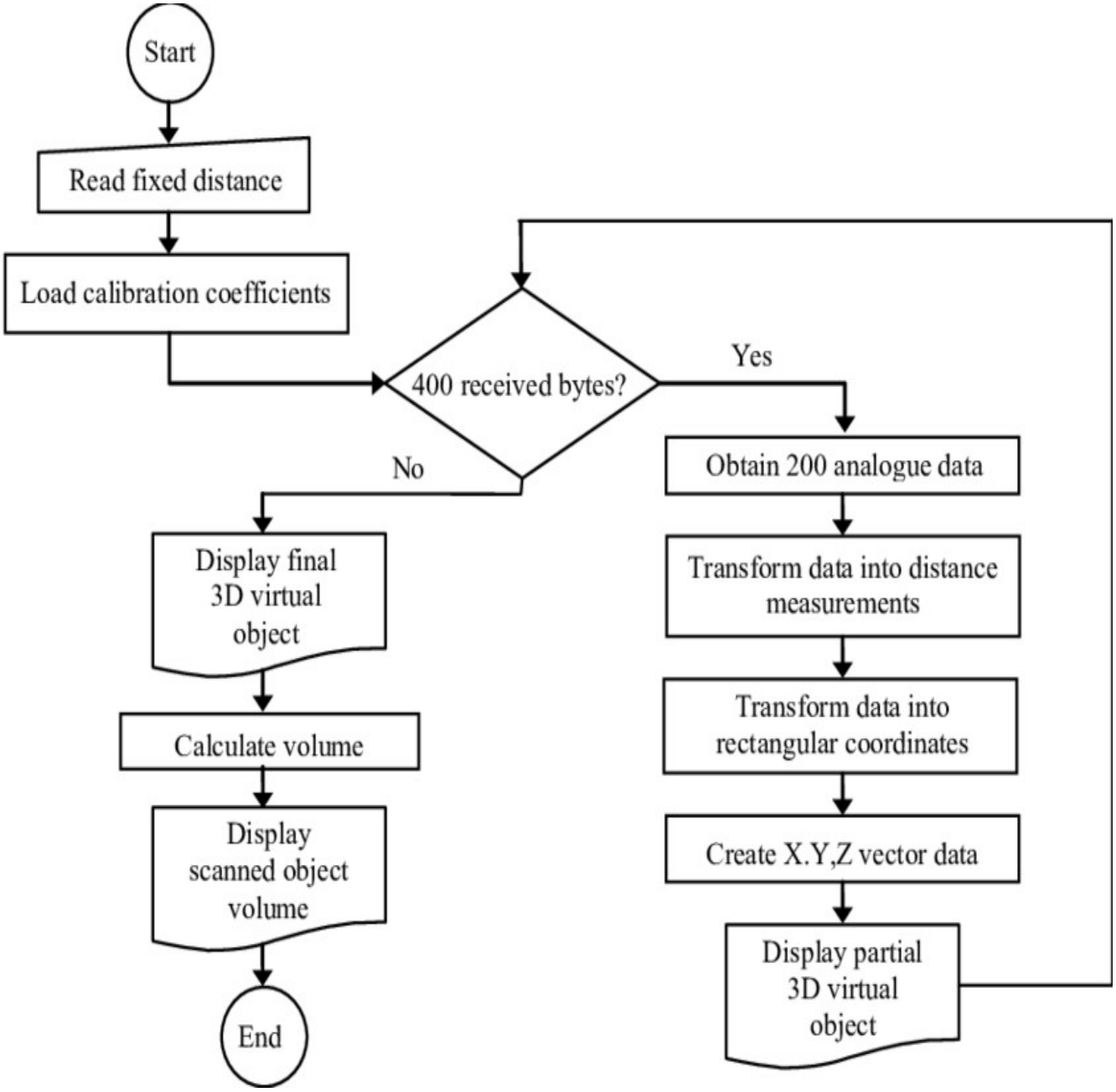


Annexure B
Entity-Relationship Diagram (ERD)
(Mandatory)



Annexure C
Use-Case Diagram (UCD)

(Optional)



Annexure D
Data Dictionary (DD)
(Mandatory)

Example:

User Table (USR)

Fields	Data type	Description
Sqare	Enter the ports	IP
End point	Last point	Declare the ports

Supplier Table (SUPP)

Fields	Data type	Description
SUPP-ID	Number	Supplier ID
-i	Enter ports	Name of the service
-e	Enter service	Supplier Address
-h	Number	Supplier no
-l -Limit	Number	Limit

Annexure E
Screen Shots

```
import socket
import time
import threading

from queue import Queue
socket.setdefaulttimeout(0.25)
print_lock = threading.Lock()

target = input('Enter the host to be scanned: ')
t_IP = socket.gethostbyname(target)
print ('Starting scan on host: ', t_IP)

def portscan(port):
  s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
  try:
    con = s.connect((t_IP, port))
    with print_lock:
      print(port, 'is open')
    con.close()
  except:
    pass

def threader():
  while True:
    worker = q.get()
    portscan(worker)
    q.task_done()

q = Queue()
  startTime = time.time()

for x in range(100):
  t = threading.Thread(target = threader)
  t.daemon = True
  t.start()

for worker in range(1, 500):
  q.put(worker)

q.join()
print('Time taken:', time.time() - startTime)
```