Experiment No: 10	
Name	Vaibhav Sharma
PRN	22070126125
Date of Performance	16th October 2024
Title	Implementation of a Basic Port Scanner Using Python for Network Security Analysis
Theory (short)	**Domain Name System (DNS)** is a fundamental component of the internet that translates human-readable domain names (like www.google.com) into IP addresses (such as 142.250.190.14), which computers use to identify and communicate with each other. DNS functions like a phonebook for the internet, enabling users to access websites without needing to memorize complex numerical IP addresses. When a user types a URL into a browser, the DNS resolver sends a query to find the corresponding IP address by searching through a hierarchical network of servers, including root servers, top-level domain (TLD) servers, and authoritative name servers. DNS caching improves speed by storing recent lookups temporarily, but if a domain cannot be resolved, users encounter errors like DNS_PROBE_FINISHED_NXDOMAIN. Additionally, DNS plays a critical role in network security through protocols like DNSSEC (DNS Security Extensions), which protects against spoofing and cache poisoning attacks.
Program	``` import socket def port_scan(): link = input("Enter the link here: ") host = socket.gethostbyname(link) res = 'a' while(res != "bye"): min = input("Enter the lowest limit of range: ") max = input("Enter the highest limit of range: ") for port in range(int(min), int(max)): try: client_socket = socket.socket() print("Trying to connect ", host, " on port", port, "......") if client_socket.connect_ex((host, port)) == 0: print("Connection to ", host,"on port", port, "was SUCCESSFUL") else: ```

```
                    print("Connection to ", host,"on port",
port, "was FAILED")
                    port = port + 1
                    client_socket.close()
            except socket.error:
                print("Connection to ", host,"on port",
port, "was an ERROR")
            port = port + 1
            client_socket.close()
        res = input("The Port Scanning has concluded. To
exit, please type bye. However if you would like to
continue with different input, press y")
    print("Scanner has exited")

if __name__ == '__main__':
    port_scan()
```

Output Screenshots

```
Enter the link here: www.kali.org
Enter the lowest limit of range: 440
Enter the highest limit of range: 446
Trying to connect  104.18.5.159  on port 440 ......
Connection to  104.18.5.159 on port 440 was FAILED
Trying to connect  104.18.5.159  on port 441 ......
Connection to  104.18.5.159 on port 441 was FAILED
Trying to connect  104.18.5.159  on port 442 ......
Connection to  104.18.5.159 on port 442 was FAILED
Trying to connect  104.18.5.159  on port 443 ......
Connection to  104.18.5.159 on port 443 was SUCCESSFUL
Trying to connect  104.18.5.159  on port 444 ......
Connection to  104.18.5.159 on port 444 was FAILED
Trying to connect  104.18.5.159  on port 445 ......
Connection to  104.18.5.159 on port 445 was FAILED
The Port Scanning has concluded. To exit, please type bye. However if you would like to continue with different input, press ybye
Scanner has exited
```

Vaibhav Sharma

Observation

1. **Range of Ports Scanned**:
   - The program scanned **ports 440 to 445** on www.kali.org(104.18.5.159)
2. **Failed Connections**:
   - Most of the ports in the given range resulted in **"FAILURE"** messages, meaning the connection attempt to these ports was **blocked or closed**.

	3. **Open Port Detected**: o **Port 443** was marked as **"SUCCESSFUL"**, indicating it is **open**. Port 80 is typically used for **HTTPS (web traffic)**, which is expected for a web server like Google.
Self-Assesment QnA	Q: What is the role of DNS in network communication? Ans: DNS translates domain names into IP addresses, allowing users to access websites without needing to memorize numerical addresses, acting as the internet's phonebook. Q: How does DNS caching improve network performance? Ans: DNS caching stores recently looked-up domain names and their IP addresses temporarily, reducing the need for repeated queries and speeding up future access to those sites. Q: What security protocols help protect DNS, and how do they work? Ans: DNSSEC (DNS Security Extensions) protects against attacks like spoofing and cache poisoning by digitally signing DNS data to ensure its authenticity and integrity.
Conclusion	This program successfully demonstrates how port scanning helps identify open and closed ports on a server. The detection of port 80 as open aligns with its use for web traffic, while the other ports being closed reflects good security practices. This experiment showed how port scanning can be used to gather information about a server's active services and security posture.