Name : Akash Rathod

Project Name : Network Port Scanner

Introduction:

The network port scanner is a tool that is widely used in the field of cybersecurity for network reconnaissance. This tool is used to identify open ports on a target machine, which helps security analysts to identify vulnerable services and potential attack vectors. In this project, a port scanner has been developed in Python language which scans for open ports on a given target IP address. The purpose of this report is to provide an overview of the project and explain the methodology used for developing this tool.

Methodology:

The port scanner has been developed using the Python programming language. The project makes use of the socket library which is a standard library in Python for low-level network programming. The scanner takes a target IP address as input and scans for open ports between the range of 0 to 100. The range of ports can be easily modified by changing the values in the for loop.

The project starts by checking the number of arguments passed by the user. If the number of arguments is not equal to 2, then the program displays an error message and exits. If the number of arguments is equal to 2, then the program proceeds to resolve the target IP address using the socket.gethostbyname() function. This function takes a hostname as input and returns the corresponding IPv4 address.

Once the IP address has been resolved, the program displays a banner indicating the target IP address and the time at which the scan was initiated. The scanner then proceeds to scan for open ports by iterating through the range of ports using a for loop. For each port, a new socket is created using the socket.socket() function. The setdefaulttimeout() function is used to set a timeout value of 1 second for each socket connection. The connect_ex() function is used to attempt a connection to the target IP address on the specified port. If the connection is successful, i.e., the result returned by connect_ex() is equal to 0, then the program displays a message indicating that the port is open. Finally, the socket is closed using the close() function.

In case of any errors such as a keyboard interrupt or a socket error, the program displays an error message and exits.

Conclusion:

In conclusion, a network port scanner has been developed in Python language that can be used to scan for open ports on a target machine. The scanner makes use of the socket library and scans for open ports in the range of 0 to 100. The tool can be used by security analysts for network reconnaissance and identifying potential attack vectors. The code for the project is provided above, and it can be easily modified to suit specific requirements.