FAUGET

CONCEPTION AND HIGH-LEVEL DESIGN

INTRODUCTION

The Network Scanner project aims to develop a tool for scanning and exploring network devices within a given IP range. The scanner employs ping sweeping to identify active hosts and performs port scanning to determine open ports on the active hosts. This deliverable provides a high-level overview of the project's conception and design.

PROJECT OVERVIEW

The Network Scanner project focuses on automating the process of network exploration and port scanning. The tool is designed to be user-friendly and efficient, providing users with the ability to discover active hosts and identify open ports on their network.

KEY PROCESSES

- **IP Validation:** The project incorporates IP address validation to ensure that only valid IP addresses are considered for scanning.
- **Ping Sweeping**: Ping sweeping is employed to determine the active hosts within the specified IP range. The project utilizes platform-specific ping commands to send ICMP echo requests and detect responsive hosts.
- **Port Scanning:** The network scanner conducts port scanning on the active hosts using TCP connections. It attempts to establish a connection with each port to determine if it is open or closed.
- **Multithreading**: To optimize the scanning process, the project utilizes multithreading to concurrently scan multiple IP addresses and ports, improving the overall efficiency of the scanning process.

HIGH-LEVEL DESIGN

1. User Interface:
   - The program interacts with the user through a command line interface (CLI) to get input: the target IP address,number of IPs to scan and number of ports to scan.
2. IP Address Validation:
   - The program includes a function to validate the entered IP address using the validate_ipaddress() function from the ipaddress module.
   - This ensures that the entered IP address is in a valid format before proceeding with further operations.

3.ICMP Ping Scanning:

- The program performs ICMP ping scans to determine the availability of the entered IP address and subsequent IP addresses in the specified range.
- The ping() function sends ICMP Echo Request packets to each IP address using operating system-specific commands (ping -n for Windows, ping -c for Linux) and checks for a response indicating an active host.
- The results of successful pings are stored in the list2ping list.

4. TCP Port Scanning:
    - The program conducts TCP port scanning on the active IP addresses identified through ICMP ping scans.
    - The scanner() function is responsible for creating a TCP socket and attempting to establish a connection with each port specified for scanning.
    - The function uses the socket module to create a socket, and if a connection is successful, it indicates that the port is open.
    - The results of open ports are displayed, and the count of open ports is maintained.

5. Multithreading:
    - The program utilizes multithreading to enhance the scanning process and improve efficiency.
    - The threader() function is responsible for managing the threads that perform port scanning.
    - Multiple threads are created to process the scanning of ports on different IP addresses concurrently, utilizing the Queue data structure to distribute the workload.

1. Output:
   - the program displays the active IP addresses and open ports.
   - The program also calculates and displays the total number of active IP addresses and the total count of open ports.
   - Additionally, the program displays the execution time to measure the duration of the scanning process.