Port Scanner

Computer Networks Assignment

GROUP NUMBER: 14

Submitted By:

Akash Gupta - 14116007 (ag.gupta730@gmail.com)
Arjun Gupta - 14116019 (arjun1177arjun @gmail.com)
Kunal Bansal - 14116035 (kunal001bansal@gmail.com)
Shobhit Mittal - 14116061 (shobhitmittal96@gmail.com)
Shubham Agarwal – 14116063 (shubhamagarwal269@gmail.com)
Swabhiman Patnaik – 14116069 (swabhimanpatnaik@gmail.com)

Introduction

The goal of this project is to implement a legitimate technique to get information about what goes in and out of various interconnected computers: *Port Scanning*. We hope that this work will help to generate better network intrusion detection systems and increase general network security. We have used JAVA language for implementation.

Objective

Port scan is an act of systematically scanning a computer's ports. As ports on a computer are the place where information is sent and received, port scanning is analogous to knocking on doors to see if someone is home. Port scanning has legitimate uses in managing networks, but port scanning also can be malicious in nature if someone is looking for a weakened access point to break into your computer. System and network administrators use port scans to identify open ports to a system so that they may be able to limit access to those ports, or shut them off entirely. Port scans represent a sizable portion of today's Internet traffic. The aim of this project is to analyze sample network traces to discover and classify properties of port scans. It helps to generate better network intrusion detection systems and increase general network security. It can also be used for revealing the presence of security devices such as firewalls that are present between the sender and the target. This technique is known as *fingerprinting*.

The *vertical scan* is a port scan that targets several destination ports on a single host. Single detection mechanisms are required in this scan. On the other hand a *horizontal scan* is a port scan that targets the same port on several hosts. Most often the attackers are aware of a particular vulnerability and wishes to find susceptible machines.

We are going to implement *block scans*, a combination of vertical and horizontal scanning styles over some well known ports which are:

Port 20: FTP | Data port

Port 21: FTP | Control (Command) port

Port 22: SSH| Secure logins and file transfer

Port 23: Telnet | Unencrypted text communications

Port 25: SMTP | Used for e-mail routing between mail servers

Port 80: HTTP | Hypertext Transfer Protocol

Procedure

The simplest port scan sends a carefully constructed packet with a chosen 32 bit destination IP address to each of the ports from 0 to 65535 to find out the ports that user is using. But as we are implementing block scan we will run this process for a number of users which are interconnected through a local area network (LAN). We have used SYN scan technique to detect whether an IP is using a port or not.

SYN scan: This technique is also called half-open scanning, because a TCP connection is not completed. A SYN packet is sent (as if we are going to open a connection), and the target host responds with a SYN+ACK, this indicates the port is listening, and an RST indicates a non-listener. The server process is never informed by the TCP layer because the connection did not complete.

This process is repeated for all the IP addresses in the network range.

While checking the ports, we have stored the time at which the user starts and stops using that port in a text file. This is done by using the "Date" class.

MAC Address Detection: We created a batch file. This batch file gets the list of MAC addresses of the active devices present in the network. This list is saved in a file.

Result

We have displayed the result in a GUI format which includes:

1.)IP address
2.)Whether the address is reachable or not
3.)Ports in use

We have made files for each IP address in which we have saved timings at which the user connects to a port and ends the connection.

Below photo shows which Port numbers (20 to 25) are being used: