DECEMBER 7, 2013

PORTSCANNER

A NETWORK ADMINISTRATOR'S TOOL

SRI HARI VARDHAN VELLANKI
SECURITY INFORMATICS
svellank@indiana.edu

Project Write-up

Project Description

PortScanner Tool enables a Network Administrator to find the open port on a remote server. This project is developed in C language and makes use of raw-socket programming. This tool supports IPv4 and can perform six types of scans namely TCP SYN, NULL, FIN, Xmas, ACK scans and UDP scan. This basic tool stores the results of each scan and makes a conclusion on whether a port is open, closed, filtered, unfiltered or open-filtered. Along with these scans it also finds the version of service running on few pre-defined standard ports like SSH, Whois, HTTP etc.,

Contents of PortScanner Application

The below are the contents of this project that are delivered in a tar ball…

- C code :
    - ✓ main.c
    - ✓ ps_setup.c [.h]
    - ✓ ps_scan.c [.h]
    - ✓ ps_prepare.c [.h]
    - ✓ ps_version.c [.h]
- Makefile
- httprequest

main.c

The main function of this application resides on this file. The main function is responsible for calling other functions in this project to complete the job. The main function will decide on where to run as a single threaded application or to run as a multithreaded application for that instance depending on the user’s input. In case of multithreaded application, it is here in this function a pool of threads is created and required mutex is initialized. This main thread would then wait for the thread pool to finish all the jobs that are assigned and then releases the thread pool resources back to the system and calls appropriate functions to print the results.

ps_setup.c [.h]

The code in this file handles the user input and prepares the current configuration of that portScanner instance. The functions in this file enable this application to calculate the network addresses in a subnet specified by an ip-prefix and read the ip-addresses from a file. It also has routines to create a job queue, populate the result list and print the results appropriately.

ps_scan.c [.h]

The logic of different scans that this application supports goes into this file. The important functions of this file are ps_default (single threaded), ps_threaded (multithreaded) and got_packet. The ps_default function iterates through the job list and does each job until all the jobs are finished. ps_threaded function is the entry point of each pthread spawned by the main thread. Each thread would retrieve a job from the job queue and would perform the task until all the jobs are done. The difference between these two functions is that, ps_default used pcap library to read the reply packets while ps_threaded relies on the sockets to read the replies. The got_packet function is called by both ps_default and ps_threaded to analyze the reply and populate the result accordingly.

ps_prepare.c [.h]

This file essentially has the routines to prepare a datagram as needed by the calling functions, either it be a TCP or a DNS packet accordingly.

ps_version.c [.h]

ps_version.c hold the routines to detect the version of software running on the pre-defined standard ports if those ports are open. Also this file takes care of populating the version detection results and displaying those results.

Makefile

This file compiles the portScanner application and prepares the executable.

Httprequest

This file hold the get message that has to be sent to the httpserver while trying to detect the version of the server on the httpport.

Credits

- Soumya Echuru Somasekhar (project partner)

During the course of our project, we have participated in design discussions with the following groups:

- Giridhar Gomatom & Mohammed Korayem
- Awani Marathe & Chintan Gosalia Krupa
- Krupa Tadepalli & Venkata Sreeja Ketineni

Resources

1) http://www.tcpdump.org/pcap.htm

2) http://www.binarytides.com

3) http://sokratisg.net/2012/04/01/udp-tcp-checksum-errors-from-tcpdump-nic-hardware-offloading/

4) http://sock-raw.org/papers/syn_scanner

5) http://beej.us/guide/bgnet/output/html/singlepage/bgnet.html

6) http://www.thinkage.ca/english/gcos/expl/b/lib/printf.html

7) http://www.linuxquestions.org/questions/programming-9/how-to-calculate-time-difference-in-milliseconds-in-c-c-711096/

8) http://www.bloof.de/tcp_checksumming

Team Spirit

In our team, we both have contributed equally starting from the design of this project till date. I do not have any complaints on my partner in this regard.