CS 491/591 Project 2

Network Packets and Intrusion Detection

(The first part of this project is taken from a homework problem set by Tom Ristenpart, University of Wisconsin. The second part is somewhat similar to a project set by Steve Zdancewic, University of Pennsylvania.)

Part 1. Due Wednesday, Nov. 14 (30%). This part must be completed by each student **individually** (no teams).

On the web site is a zip file containing packet traces that can be read by the tool WireShark (among others). For this problem there are 4 traces that you will need to investigate in order to find out the information asked for below. To get started, you will want to understand how to apply WireShark's filtering feature. You may also find the `whois` tool useful.

Hand in your answer as a neatly formatted file called `proj2.1.txt` or `proj2.1.pdf`, and send it as an attachment in an email to `apt@cs.pdx.edu`.

Trace 1: HTTP traffic
1. Give three websites visited from source IP address 128.12.173.14.
2. Give three search queries made from source IP address 128.12.173.14.
3. Determine the organization that 128.12.173.14 is associated to.

Trace 2: FTP traffic
FTP is the file transport protocol. There is a lot of information about it on the Internet.
1. What is the user name and password used to connect to the FTP server?
2. Explain the difference between a passive FTP connection and an active FTP connection.
3. Give the packet number ranges across which there were active connection(s).
4. Give the packet number ranges across which there were passive connection(s).
5. List any files that were downloaded.

Trace 3: Traceroute
Traceroute is a tool used to determine the route between two IP addresses.
1. Identify the source IP address that issued a traceroute command.
2. Identify the destination IP address of the traceroute command.
3. List the IP addresses on the route between the source and destination.
4. Determine the organizations associated with all the IP addresses along the route.

Trace 4: POP
The POP protocol is used for Email.
1. What is the POP username and password?
2. How many emails are in the users mailbox?
3. Give the contents of from, to, subject, and date for each email.

Part 2. Due Wednesday, Nov. 28 (70%). This part may be worked on in **teams** of no more than three students, with each team submitting a single solution.

In this part of the project, your will implment the rudiments of a web monitoring tool that could be used to detect intrusions or other traffic anomolies. Rather than using a tool like WireShark, you will use the low-level `pcap` library to inspect packets programmatically.

Your program should process a packet trace file (several large ones will be placed on the web site), and do the following:

1. Generate a list of the IP addresses/ports that appear to act as servers (e.g. web,mail,tcp).

2. For each server, list the total number of connections and the total number of bytes served.

3. Detect a probable port scan attack on a specified target address. This is very open-ended, but at a minimum you should be able to detect a simple SYN scan of a single host generated by `nmap` using its default target port selection (a random ordering of the 1000 most common ports). But feel free to get more elaborate.

Note that you can perform these analyses at the TCP/IP packet level; there is no need to reconstruct entire TCP messages. Moreover, you for this project, you can make the simplifying (though unrealistic) assumption that IP packets are not fragmented, so there is no need for reconstruction at the IP level either.

The `pcap` library is written in C, and is installed on most unix-ish systems; there is also a Windows variant. You may choose to use it directly by writing your program in C, but I recommend using a higher-level language binding for the library instead. For example, there are bindings for Java (details for using the `jpcap` library on the linuxlab machines are on the course web page), Haskell, and Python. If you use something other than C or Java, be sure your submission includes enough information for me to build your executable easily (e.g., a make file that you've tested on linuxlab).

Your submission (mailed to `apt@cs.pdx.edu`) should include a README describing your approach (especially to item 3), and complete code.