Exercise on Pointers

CSCE 496: Performance Analysis of O-O Systems

February 6, 2004

Abstract

In this exercise, the students are asked to write a program to analyze a packet trace file obtained through Solaris' "snoop" utility. You are responsible for capturing all the "tcp" and "udp" packets. This exercise is strictly optional and will not be graded.

1 Problem Statement

Solaris provides a very nice network analysis utility called "snoop". Typically, snoop is run as superuser to capture real-time network packets that pass through the Network Interface Card (NIC). Additionally, it also has the ability to store captured packets in a file. This file can also be interpreted by "snoop" for further analysis.

In this exercise, a trace file captured by "snoop" is given. You are to write a program that will analyze the packets on this trace file. For more information about the "snoop" file format, please refer to the following web-page.

http://www.faqs.org/rfcs/rfc1761.html

The given trace file contains the data during a ftp communication. Your task is to capture the basic packet information such as:

- source IP address
- destination IP address
- source port number
- destination port number
- timestamps (in second and millisecond)
- original and include packet lengths
- packet data, etc.

The information should be displayed on the screen separated by the packet number. For example, your information can be displayed as follows:

```
####### Packet Record - 1 #######
...

####### Packet Record - 8 #######
Orig. Length: 66
Inc. Length: 66
Cummulative Drops: 0
Time (sec): 920418599
Time (msec): 546256
payload size 68
padding 2
######## TCP/IP Info ########
Source IP:Port-> 130.184.206.101: 3879
Destination IP:Port-> 130.184.201.177: 21
####### Packet Record - 9 #######
Orig. Length: 54
Inc. Length: 54
Cummulative Drops: 0
Time (sec): 920418599
Time (msec): 547051
payload size 56
padding 2
######## TCP/IP Info ########
Source IP:Port-> 130.184.201.177: 21
Destination IP:Port-> 130.184.206.101: 3879
####### Packet Record - ...
```

Similarly, the payloads should be stored in a file. Again, each payload should be separated by the packet number.

```
============ packet - 1 =============
...

============ packet - 5 =============


^@^@^@^@^@^@
============ packet - 6 =============


220 csci.uark.edu FTP server (Version wu-2.4.2-academ[BETA-13]
```

```
(1) Wed Jun 11 22:58:42 CDT 1997) ready.^M^M^M/
============ packet - 7 =============


^@^@^@^@^@^@
============ packet - ...
```

Your program should take two arguments. The first argument is the input file (snooplog.bin) and the second argument is the payload file (e.g. payload.bin). The sample files (screen output and payload output) will be available on the Linux server.

2 Getting Started

You should first try to analyze the trace file given. To do so, you can log-in to your cse account and enter the following command:

prompt> /usr/sbin/snoop -i snooplog.bin > snoop.output

You can view the analyzed result from the snoop.output file.

If you choose to write your program on an X86 architecture, you have to be careful with byte-order. X86 uses little endian byte-order that is different from the network byte-order (big endian). Be careful with the packet information stored in the file, you must convert the information to the correct byte order before it can be displayed or stored. There are several utility functions for network data manipulation. Such functions include:

- ntohl
- ntohs
- inet_ntoa, etc.

Additionally, you may need to use file I/O function such as *fread*, *fwrite*, *feof*, etc. You also should look into several header files under /usr/include/inet. The list of those files are as follows:

- ip.h
- in.h
- tcp.h
- udp.h
- icmp.h

- if.h

Once you have successfully log-in, you can copy the sample file from:
witty/share/csce496/sample.tar.gz
Inside this file, you will find the packet trace file (snooplog.bin), a sample screen output file (output.txt), and a sample payload file (payload.bin). You should write your program to output the information in the similar fashion.
You can easily view the payload.bin file by using the following command:

prompt> strings payload.bin > payload.txt

The file, "payload.bin" is written in binary format. To view all the readable text, you can use "strings" command and redirect the output to "payload.txt".

3 What to create

There should be two files that you will need to create. First is your header file (mysnoop.h) which should at least have the structures for the snoop file header and the snoop packet record header. Second, you need to create the source file (mysnoop.c) which should be written to display and store information from both TCP and UDP packets (the trace file may only have the TCP packets).

That's all folks!