CSCE 665 Homework 1

**Ke Wang 822002009**

How to build and how to use

This program is written in C++ using visual studio IDE. So the compilation and build is quite simple. Because we use Wincap API, so first we need to tell visual studio where is the “Include” and “lib” folder of Wincap development package. After the “sniffer.exe” we can run it through command line “sniffer [80|21|23] filename”, where the number “80”,” 21”,” 23” represent the port number of http, ftp and telnet respectively. The “filename” represent the corresponding trace file (x.pcap) we want to load. The program will print all the sessions of a certain protocol to the console.

Design

The program loads a session file and returns a “pcap” handle to the user to parse the packets. This is done by function pcap\_open\_offline(). Then it enters a loop to process each packet in using function pcap\_loop().

The process procedure is written in a callback function dispatcher\_handler(). In this function, we can get a packet header structure which shows the packet size and timestamp. We can also got the packet content and parse out the Ethernet, IP, TCP header and the payload according to their size and offset. We need the TCP port number to decide the protocol of the packet. We also need source and destination IP address to find different sessions. We use a session map to keep information of different sessions. The map key is the logic AND value of source and destination IP address. In this case, packets from both directions will be added to the same session. The map value is the content of the session read from the packet payload.

After the pcap\_loop() return. If all of the packets are processed correctly, we will traverse the session map and print out the content of each session. There will be some none ASCII character in the session content. So we use isascii() function to detect them and print out their hex value instead of the ASCII characters.

Result analysis

Fig 1 show two http sessions. we can clearly see the http request and response.

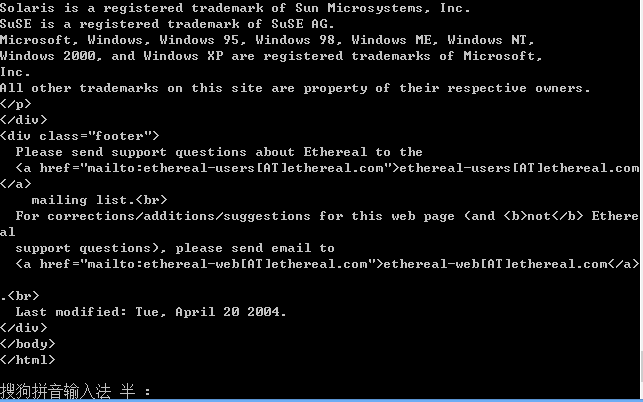


Fig 2 shows the telnet session. Here we can see the username and password. So if we set our Ethernet card in promiscuous mode, we are able to get packet not sending to us. Because content in telnet is not encrypted, we can easily get other user’s real username and password, which is very dangerous to the user. So telnet is not a secure protocol.

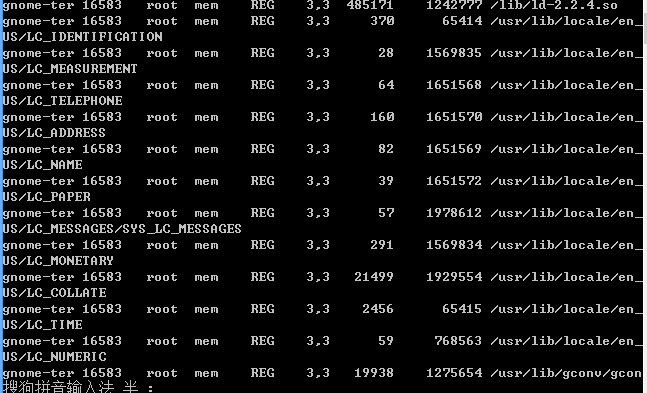
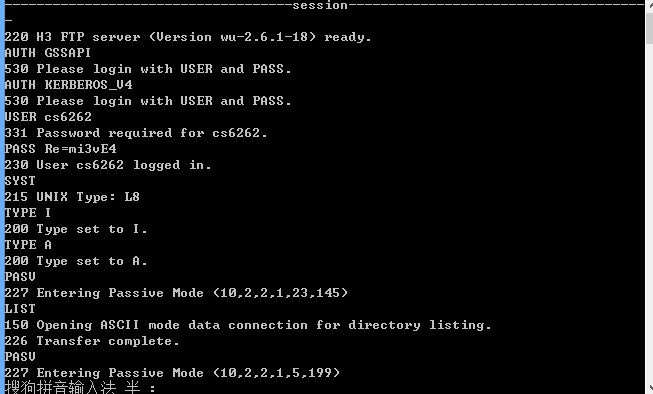


Fig 3 shows the session of FTP and we can see it has similar security problem.

****