Packet Capture

Colin Phillips

January 17,2012

A project for capturing packets

.

Acknowledgement of Sources

For all ideas taken from other sources (books, articles, internet), the source of the ideas is mentioned in the main text and fully referenced at the end of the report.

All material which is quoted essentially word-for-word from other sources is given in quotation marks and referenced.

Pictures and diagrams copied from the internet or other sources are labelled with a reference to the web page or book, article etc.

Signed ..... Date .....

1 Introduction

The purpose of this document is to provide a basic idea of what our packet capturing program does and the design decisions we made through out the course of the project.

1.1 Why raw sockets and not the interface card ?

I chose to use raw sockets to capture the packets since it makes more holistic sense in the way that all data is send and received in the Internet (through sockets) and wanted to keep that idea of communication between different entities in a network even between the capturing of these packets,but we do have an alternative code using pcap where packets are captured on the network interface card if you interested in it.

1.2 How do you make the socket connection ?

In the main program raw sockets were used to capture the packets comming in from the network, the libraries we included from c++ to make this possible was the TCP/IP socket libraries, and we specifically used the netinet classes for tcp, this basically opens a socket that makes it possible to capture tcp packets from our network.

1.3 Basic Methodology of code

What I did was to open a big while loop and first listened for a connection on the sockets or for an incomming packet if there is none we close the program but if there is we need to capture the packet and "unpack" it we do this first by looking at the IP header field to analyze the type of packet it is and we have different void method for each packet we capture if it is a tcp method we would go to let us say tcpprocesspacket method() where we unpack the other fields of the packet like the acknowledgement field doing this is very simple with the built in tcp functions.

Storing the captured data In storing the data a log file was used called works.txt to capture all the data we unpacked we used hton() to make sure that the data which is just numbers mostly integers can be read correctly on all platforms or anywhere so that the way these numbers are read from the receiver and sender is basically the same, we also stored the data pay load and converted it to "english" for the http messages to be read we also used a text file to store the output for the user which can be read easilly and

retransmitted packets are also showed here.A database is also included we decided to use sqlite 3 since it is small and easy to use which is just what we wanted for this task db includes tcp information.

2 Handling of HTTP Messages

Port 80 is a reserved port for tcp messages however it is not the only port in which http messages can be send so to find a way to deternmine an http message is not as simple as waiting on port 80,but the format of the http message is what we are looking for especialy request message line wh it has a format HTTP/x.x. (encoded in Hexidecimal) also the full response message ie status line also has this format which our log file captures so by going to the first line of each datadump file and decoding the first part we can find out if it is http or not.

2.1 Other Things

I also provide users to enter their passwords and user names before being to use our program we use a password.txt file to manually enter the passwords and users and then our login () method compares the user passwords and user names with each other.

3 Conclusion

In conclusion we learned how important packet capturing is to uncover the root cause of network problems, identify security threats, and ensure data communications and proper network usage throughout the Internet.

Bibliography

Douglas, S. (2002). *C++ Network Programming*, S.Vinoski; Addison-Wesley.

Schifmann, M. (2002), *Building Open Source Network Security Tools* , Berlin; Reading, MA.

Strydom, J. (2011). 'http://fyrewolfe.blogspot.com', *Internet*,