ISSN No. 0976-5697

Volume 3, No. 3, May-June 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

# On the Design of A URL Capturing Tool for Monitoring Enterprise Networks

Longe, O.B (PhD)*
Fulbright Fellow International Centre for Information Technology & Development Southern University System
Baton Rouge, Louisiana, USA.
longeolumide@fulbrightmail.org

Adebimpe, L.A
Dept of Computer Science
Emmanuel Alayande College of Education,
Oyo, Nigeria
dradebimpela@yahoo.com

***Abstract*** - The power of computer networks enhances user experience to access distributed resources but places a toll on user security in a networked environment. Securing networks in corporate settings and on client systems are becoming more complex and definitely require the provision of more robust methodologies for personal and organizational use. Using the Java programming language and window Packet capturing API (Winpcap) we developed a URL capturing tool (acting as a packet sniffer application) that is able to support protocols with monitoring capability on Networks. Usage monitoring is achieved by capturing packets and extracting vital information from them. The intention is to provide administrators information on URL trails on a platform that can work in tandem with most network protocols in order to limit the complexity of packet monitoring and ensure low packet loss rate.

***Keywords***: Filter, Packets, Networks, Security, URL

## I. INTRODUCTION

Securing networks require the development of security tools that can analyze traffic, monitor activities, diagnose and test functionality while making network usage safe for users. In order for such tools to function, they will need to obtain data transiting over a network and capture data attributes while the network is operational. The capture process is composed of listening on networks and obtaining transiting frames independently from its source or destinations. Unfortunately, most network transmission techniques and communication protocols complicates these tasks since it is difficult to capture packet from fast networks at full speed without losing packets [1]. To overcome these challenges, network systems must be infused with basic low level features that are inculcated during the design phase so as to achieve the objective of scalable network monitoring and control.

### *A. Capturing Data from Networks:*

In order to capture data from a network, two techniques that can be employed. The first is the use of dedicated hardware. Dedicated hardware are mostly proprietary hardware components that work in line with system and network architecture in order to perform the capturing process. Its advantage lies in the fact that it is implemented on the machine and as such it is very fast in performing these tasks [2]. The overheard of the dedicated hardware solutions is that it usually experiences difficulties in deployment since these hardware components cannot be duplicated or easily moved. They are also less flexible than software solutions. The second method is to use network adapters in tandem with software tools to obtain frame packets from the network. The advantage is that software solution is cheaper, easier to modify and upgrade. However, software solutions results into high overhead resulting in low performance particularly on slow machines. Relying on the fact that the software solution yields to easy modification and upgrade, it is widely adopted on the most network architectures where the performance of dedicated hardware is not needed [3].

### *B. The Challenge Scenario:*

Today's organization thrives on information that is exchanged on networks. Since these organizations entrust vital and sensitive information over a network, it is therefore necessary to provide efficient security solution that can assist the monitoring of network activities and provide network administrators information on user activities consisting of time stamps and usage zoning in order to secure the network from both internal and external attacks. Internal attack usually results from within the organization when insiders attempt to access information to which they have no rights. It could come in the form of employees attempting to modify content or extracting some information from existing files. External attacks are usually threats from outside the organizational perimeters. They may come in the form of external programs attempting to access the network. Protection against such attacks can be achieved by use of firewalls and network monitoring devices that can signal intrusion activities [4].

Beyond usage in corporate environments, network access and usage has also increased on the home front. Society and especially parents are particularly concerned about how children interact online through the internet. Measures that provide information on usage patterns and allows some form of control and monitoring will go a long way in aiding ethical usage among children and youths and by so doing keep the internet safe from predators. Since most violations and attacks are packet based, the development of packet sniffing tools that can scale on most protocols to provide information on usage, usage permission and possible packet modification or manipulations is warranted.

This can be achieved by capturing all the Uniform Resource Locators (URLs) that are visited in tandem with their time stamps. For organizations, there are usually policies that limit the certain URL from being accessed during office hours. A system that monitors and captures URLs can also assist in identifying violations of corporate network security and usage policies [4][1][5]

### *C. Research Direction:*

Our research is targeted towards using the Java programming language and window Packet capturing API [6] to developed a URL capturing tool (acting as a packet sniffer application) that can run on most internet protocols with monitoring capability for Network.

The application supposed to be a high performance network analysis tools that monitor events transition over corporate and client PC networks while ensuring low packet loss rate.

### *D. Design Components:*

The determination of possible attack tactics is essential to developing responses and countermeasures. Usually, a system administrator or network manager presented with a successful intrusion has very little information with which to work; a possibly corrupted system log or a firewall log that do not posses adequate information to trace the culprit. In order to be able to monitor the network in an efficient manner, a network administrator needs an effective tool to work with that permits accurate and highly condensed summaries of an event over the network and storage of information about the events that are observed.

## II. RELATED WORKS

Degioanni et al [7] proposed a system that captures a packet on a network using a capturing application that interacts directly with the network hardware. In this scenario, the operating system offers a set of capture primitives to communicate and receive data directly from the network adapter by capturing packets from the network (hiding the interaction with the network adapter), and transferring them to the calling programs or to a storage location for future reference. Berkeley Packet filters (BPF) is a kernel level component for packet capture that act as a device driver by interacting with the network interface through the interface’s driver. BPF is essentially a device driver that can be used by UNIX applications to read the packets from the network through the network adapter in a highly optimized way [8]. Firewalls have also been used to enforce policy defined to govern the exchange of information within a network or between groups of networks. Curtin [9] lends credence to this fact by defining a firewall as “a system or a group of systems that enforces an access control policy between two networks”. Some systems are also designed to serve as control points to and from a network. They achieve this by evaluating connection request as they are receiving and checking whether or not the network traffic should be allowed, base on a predefined set of rules [10].

Fuller & Pagan [11] identified systems that are typically routers with packet filtering capabilities. Some application layer systems are proxy or application gateway that inspect traffic at the application level in addition to lower levels [12][13][14].

Intrusion Detection Systems (IDS) have also been used to secure networks. IDS systema cen be misuse detection models and anomaly-based detection models. MIDS are raw packet-parsing engines which capture network traffic and compare them with a set of known attack patterns or signatures. Anomaly-based IDS approaches are to understand the patterns of users and traffic on the network, and find deviation in those patterns. In theory, an anomaly based IDS could detect that something was wrong without knowing specifically the source of the problem [15][16][17].

## III. SYSTEM DESIGN

In existing systems, each time a network card receives an Ethernet packets of data, the first line of action is to check its destination MAC address if it matches its own or not. If the destination MAC address matches the host address, it generates an interrupt request. The routine in charge of handling the interrupt is the Systems network card driver. The functions of the driver are to timestamp the received data and copy it from the card buffer to a block of memory in the Kernel An existing scenario without a packet filter is depicted in Fig. 1 below.

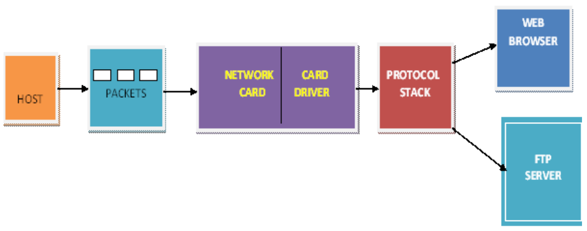


Figure 1: Packet Usage Without Packet Filter

The picture above depicts the stages that a packet pass through before getting to the user level where it is being consumed. In order capture data packet for the network monitoring tools (Sniffers, Network Monitors), there is need to have a packet filter that can interface directly with the card driver [18][6].

It can also and receive the same data packets that is been transferred to protocol stack. The introduction of this intermediate process is depicted below.

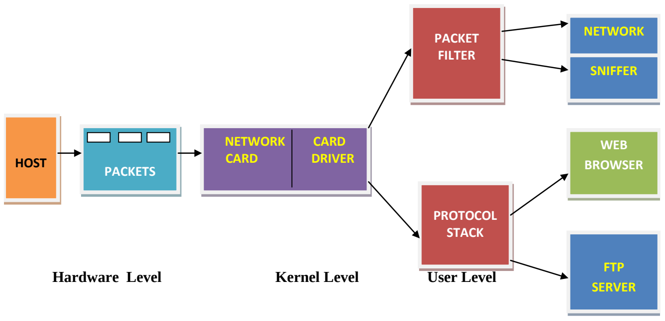


Figure.2: Packet Usage With Packet Filter

The packet capture driver interacts with the network adapters device drivers through NDIS (Network Driver Interface Specification) which is a part of the network code of Win32. NDIS is responsible of the management of the various network adapters and of the communication between the adapters and the software portions that implement the protocols. A basic network capture driver can be quite simple. It needs only to read the packets from the network driver and copy them to the application. However, in order to obtain acceptable performances, substantial improvements need to be done to this basic structure.

### *A. The Proposed System:*

Our efforts are targeted at designing a system that can be used to capture data packets as it is transiting over the network for it to be properly analysed. In as much as we need these packets to perform various forms of analysis. We must be cautious not allow the functionality of packet filter to affect the functionality or the performance of the network as a whole. At a particular point in time, a network administrator may be interested in getting to know about network activities such as knowing who is using a particular application, files or monitor the behaviour of the network over a period of time. Also he may wants to know instantly each time a new IP address tries to connect to his network. On client applications, parents and guardians are mostly concerned about the way their wards interact with the world around and are so concern about the type of web content they have access to behind their back. This now necessitates the development of mechanisms that can be used to track or monitor the kind activities that kids engage in on networks.

### *a. Systems Analysis:*

Here we visualize the kind of operation or the kind of information that a Network Administrator or client system users may wants to extracts from the data packet being captured on the network.

For the Network Administrator, we need to capture the following:

a) Capture Time Logged on to the Network.
b) Capture all the URL Visited.
c) Capture and show Contents of Download made (File size).
d) Capture IP addresses on the NW and Show Bandwidth Usage.
e) Capture all the Server IP Addresses.
f) Capture File Request by each user from the Server (Show Root Folders).
g) Capture all files that is been sent out of the Network.
h) The application should be able to throw error Message each time a new IP addresses connect to the Network and the IP is not registered in the Packet Capture Application.
i) For the Parent Application, we need to capture the following:
j) Capture Time Logged on to the Network.
k) Capture all the URL Visited.
l) Capture Username and Password of each user.
m) Capture and show Contents of Download made (File size).

Using this application, a parent may likely be interested in knowing the followings:

i. Time Logged on to the Network.
ii. Uniform Resource Locator Visited (URL's).
iii. Contents of all Downloads made (Size).
iv. Username and Password of each user.

To analyse the system efficiently, there are tools that can help us to model the system in a way that help us to get full grasp of the requirement that we need for the applications. Among the viable tools is *Use Case Modeling, Class Diagrams in Unified Model Language.*

### *b. Use-Case Modeling:*

The use case diagram below represents the functionality of the system from the users point of view. It is used to displays the relationship among actors and the cases. The two main components of a use case diagram are *cases* and *actors*. These actors actually represent roles that people or devices play as the system operates.

Defined somewhat more formally, an actor is anything that communicates with the system or product and that is external to the system itself. The assumption that we make during the analysis of the system for the client application is that each user on the network must have an aaccount on client applications and PCs that can only be accessed by distinct usermane and ppassword.

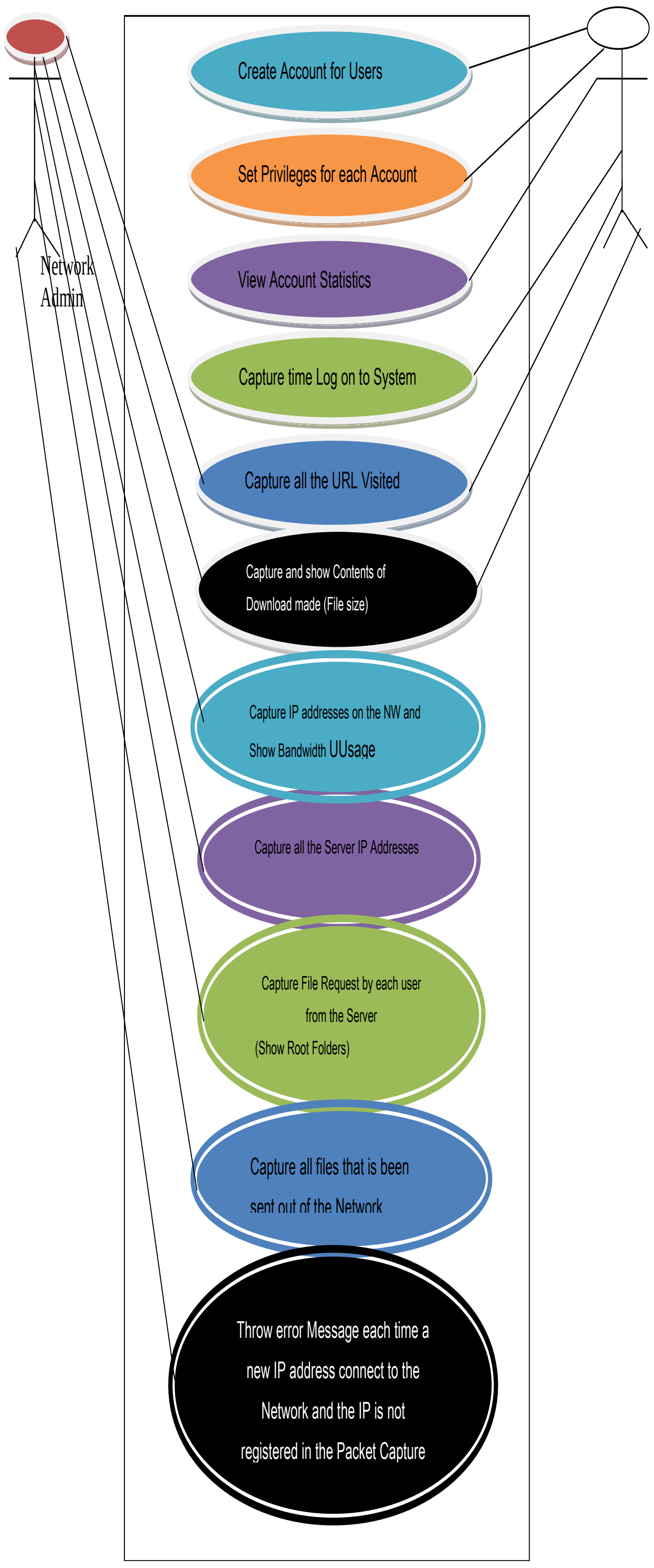

Network Admin
Create Account for Users
Set Privileges for each Account
View Account Statistics
Capture time Log on to System
Capture all the URL Visited
Capture and show Contents of Download made (File size)
Capture IP addresses on the NW and Show Bandwidth UUsage
Capture all the Server IP Addresses
Capture File Request by each user from the Server (Show Root Folders)
Capture all files that is been sent out of the Network
Throw error Message each time a new IP address connect to the Network and the IP is not registered in the Packet Capture

Figure. 3: Use-Case Modelling

The class diagram below gives a vivid description of how the objects in our system are related.

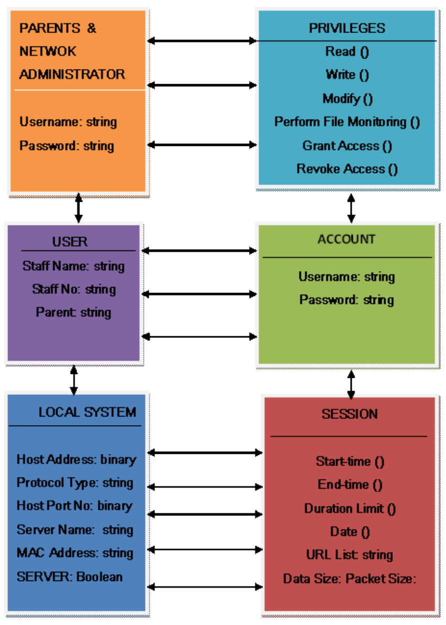


Figure 4: The Class Diagram

## IV. SYSTEM IMPLEMENTATION

The choice of implementation tools used are those that goes in line with the low level functionality of the system requirement and give us something that is easily accessible by the developer, the Network administrator and the clients. The implementation tools used are NetBeans 6.5.1 IDE, Microsoft Structured Query Language; MySQL 2005 and Microsoft Access 2007. The NetBeans IDE is an open-source integrated development environment that supports development of all Java application types. It also refers to both a platform framework for Java desktop applications, and an integrated development environment (IDE) for developing with Java, JavaScript, PHP, Python e.t.c The NetBeans Platform allows applications to be developed from a set of modular software components called *modules*.

Microsoft Office Access is a pseudo-relational database management system that combines the relational Microsoft Jet Database Engine with a graphical user interface and software-development tools. Microsoft Access is used to create simple database solutions. Access tables support a variety of standard field types, indices, and referential integrity. Access also includes a query interface, forms to display and enter data, and reports for printing. Database solutions created entirely in Microsoft Access are well suited for individual and workgroup use across a network. The number of simultaneous users that can be supported depends on the amount of data, the tasks being performed, level of use, and application design. Microsoft Access is particularly appropriate for meeting end-user database needs and for rapid application development (RAD) and is easy enough for end users to create their own queries, forms and reports, laying out fields and groupings, setting formats etc. Third party tools such as Macromedia Fireworks is also used for interface designs.

### A. *The Network Administrator's Application:*

The network administrator application like any other application will enable the user of the application to create an account before accessing the system. Since the application will be running on server, the administrator has the exclusive right to grant access to any user he or she feels is safe to access the system. So before any user can access (i.e. Login) system there is need to create account and this account serves as routes to the entire application utility sections where users can perform one process or the other.

### B. *The Login Page:*

After creating the account, the application immediately brings out the login page for the new user to be able to logon to the application.

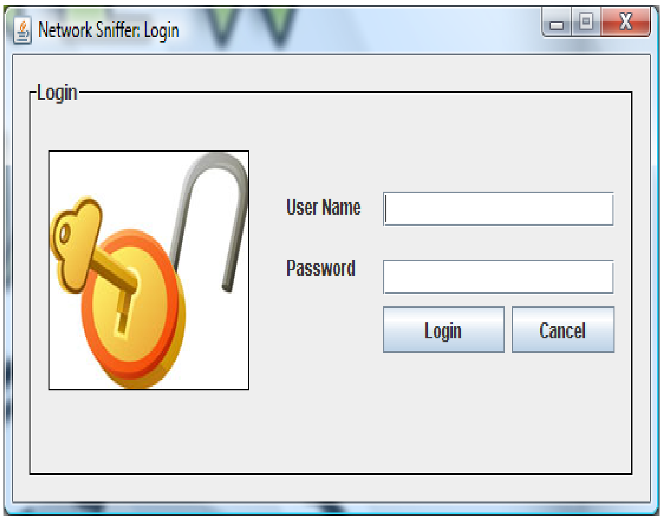
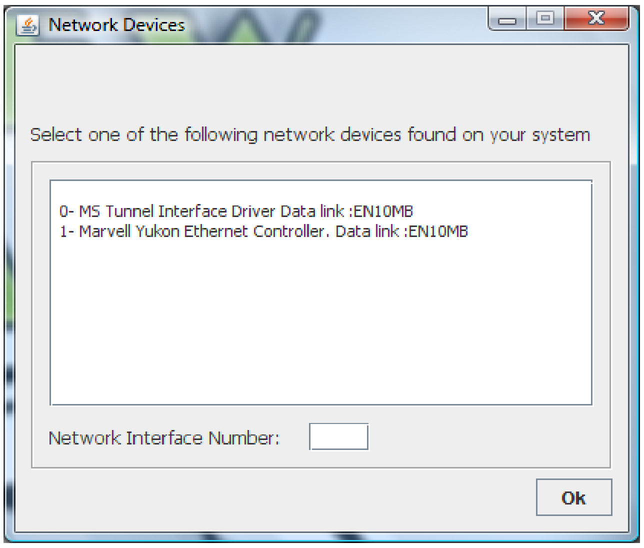


Figure.4: Login Page.

### C. *The Network Devices Interface:*

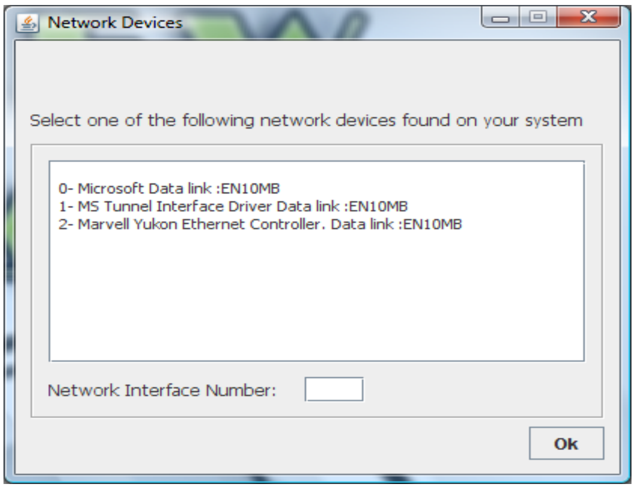
After Login to the system, the application automatically execute a loop through the system to get the various type network card that is on the machine that receive or get the outgoing packet from. The administrator is then presented with options to choose from various network card that is present on the system with indexing starting from zero.



Select one of the following network devices found on your system

0- MS Tunnel Interface Driver Data link :EN10MB
1- Marvell Yukon Ethernet Controller. Data link :EN10MB

Figure. 5: Network Devices Interface (Without Wireless Signal On).



Network Devices

Select one of the following network devices found on your system

```
0- Microsoft Data link :EN10MB
1- MS Tunnel Interface Driver Data link :EN10MB
2- Marvell Yukon Ethernet Controller. Data link :EN10MB
```

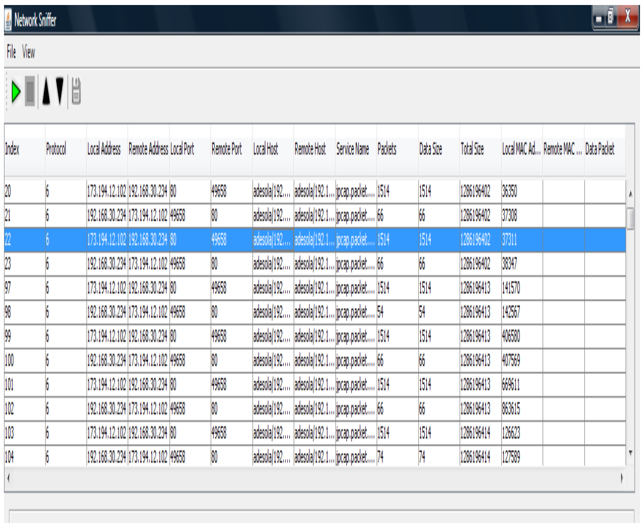
Network Interface Number:

Figure.6: Network Devices Interface (With Wireless Signal On).

## D. *System Testing and Integration:*

The program code was run through to spot algorithmic, data, and syntax faults. The code was compared with the specifications and with the design to make sure that all relevant cases have been considered.

After each unit were developed, they were tested based on Unit testing called *Code Walkthroughs*. After being guaranteed that the system components (different classes representing sections and modules) are working correctly and meet the specified objectives, the components were then combined into a working system. This integration was planned and coordinated so that when a failure occurs, we have some idea of what caused it. The entire system was tested as a whole to ensure that the whole system functions accordingly by checking that the code and the interface work with each other properly. After carrying out all of these testing phases on the application, we have verified and validated software that can be used to monitor the network that is presented below.



| Index | Protocol | Local Address | Remote Address | Local Port | Remote Port | Local Host | Remote Host | Service Name | Packets | Data Size | Total Size | Local MAC Ad... | Remote MAC ... | Data Packet |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 20 | 6 | 173.194.12.102 | 192.168.30.234 | 80 | 49658 | adesola/192.... | adesola/192.1... | jpcap.packet.... | 1514 | 1514 | 1286196402 | 36350 | | |
| 21 | 6 | 192.168.30.234 | 173.194.12.102 | 49658 | 80 | adesola/192.... | adesola/192.1... | jpcap.packet.... | 66 | 66 | 1286196402 | 37308 | | |
| 22 | 6 | 173.194.12.102 | 192.168.30.234 | 80 | 49658 | adesola/192.... | adesola/192.1... | jpcap.packet.... | 1514 | 1514 | 1286196402 | 37311 | | |
| 23 | 6 | 192.168.30.234 | 173.194.12.102 | 49658 | 80 | adesola/192.... | adesola/192.1... | jpcap.packet.... | 66 | 66 | 1286196402 | 38347 | | |
| 97 | 6 | 173.194.12.102 | 192.168.30.234 | 80 | 49658 | adesola/192.... | adesola/192.1... | jpcap.packet.... | 1514 | 1514 | 1286196413 | 141570 | | |
| 98 | 6 | 192.168.30.234 | 173.194.12.102 | 49658 | 80 | adesola/192.... | adesola/192.1... | jpcap.packet.... | 54 | 54 | 1286196413 | 142567 | | |
| 99 | 6 | 173.194.12.102 | 192.168.30.234 | 80 | 49658 | adesola/192.... | adesola/192.1... | jpcap.packet.... | 1514 | 1514 | 1286196413 | 406580 | | |
| 100 | 6 | 192.168.30.234 | 173.194.12.102 | 49658 | 80 | adesola/192.... | adesola/192.1... | jpcap.packet.... | 66 | 66 | 1286196413 | 407569 | | |
| 101 | 6 | 173.194.12.102 | 192.168.30.234 | 80 | 49658 | adesola/192.... | adesola/192.1... | jpcap.packet.... | 1514 | 1514 | 1286196413 | 669611 | | |
| 102 | 6 | 192.168.30.234 | 173.194.12.102 | 49658 | 80 | adesola/192.... | adesola/192.1... | jpcap.packet.... | 66 | 66 | 1286196413 | 863615 | | |
| 103 | 6 | 173.194.12.102 | 192.168.30.234 | 80 | 49658 | adesola/192.... | adesola/192.1... | jpcap.packet.... | 1514 | 1514 | 1286196414 | 126623 | | |
| 104 | 6 | 192.168.30.234 | 173.194.12.102 | 49658 | 80 | adesola/192.... | adesola/192.1... | jpcap.packet.... | 74 | 74 | 1286196414 | 127589 | | |

Figure.7: The Network Administrator Application Interface

## E. *The Single Client Application:*

Also, the Parent Application like any other application will enable the user of the application to create an account before accessing the system. So before any user can access the system, there is the need to create account and this account serves as a route to the entire application utility sections where user will perform one process or the other.

The assumption that we made during the analysis of the system for client aapplication is that each user on the computer system must have an aaccount on the client PC that can only be access by distinct Username and Password and each must endeavour to log on to the system with details.

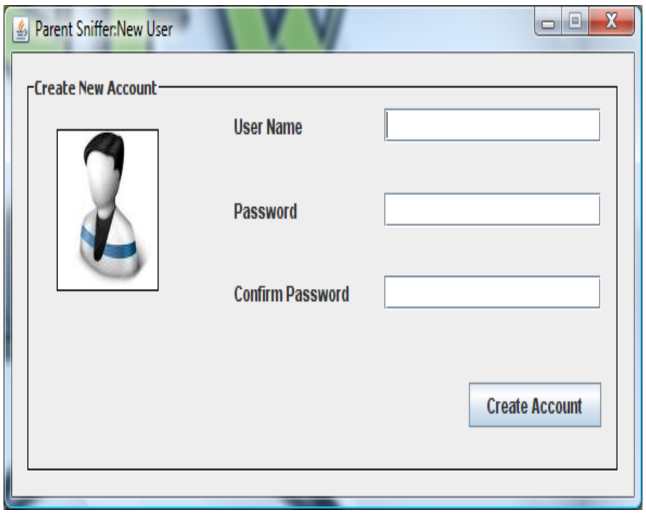


Figure. 4.7: New User Page

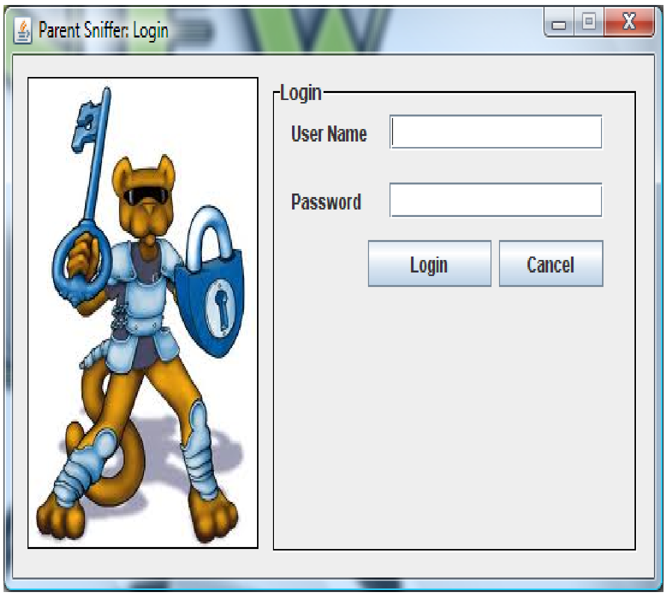
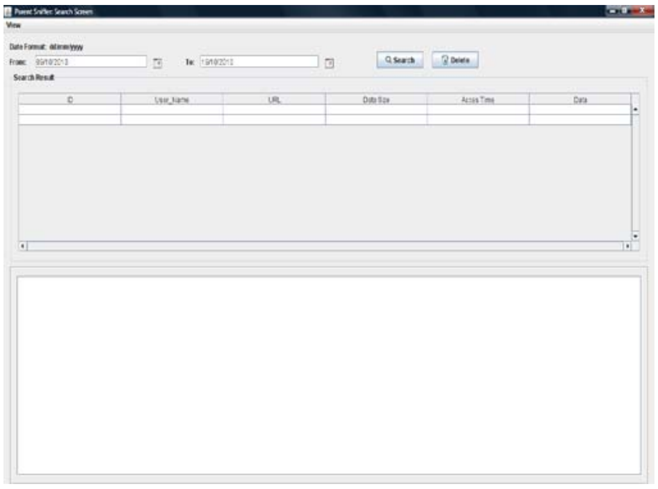


Figure. 4.8: Login Page.

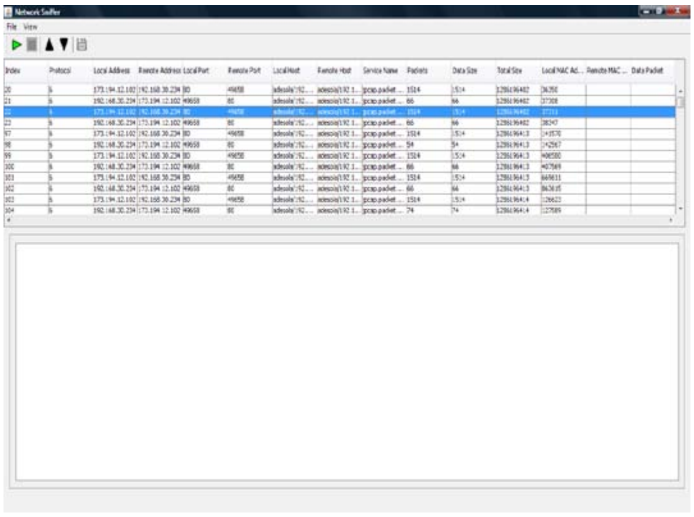


Date Format: dd/mm/yyyy

From: 09/10/2013 To: 10/10/2013

| ID | User_Name | URL | Data Size | Access Time | Data |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |

Figure. 4.8: Client Interface where they search for day to day activities.



| Index | Protocol | Local Address | Remote Address | Local Port | Remote Port | Local Host | Remote Host | Service Name | Packets | Data Size | Total Size | Local MAC Ad... | Remote MAC ... | Data Packet |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 20 | 6 | 173.194.12.102 | 192.168.30.234 | 80 | 49658 | adesola/192.... | adesola/192 1... | pcap.packet ... | 1514 | 1514 | 1298196402 | 36350 | | |
| 21 | 6 | 192.168.30.234 | 173.194.12.102 | 49658 | 80 | adesola/192.... | adesola/192 1... | pcap.packet ... | 66 | 66 | 1298196402 | 37308 | | |
| 22 | 6 | 173.194.12.102 | 192.168.30.234 | 80 | 49658 | adesola/192.... | adesola/192 1... | pcap.packet ... | 1514 | 1514 | 1298196402 | 37311 | | |
| 23 | 6 | 192.168.30.234 | 173.194.12.102 | 49658 | 80 | adesola/192.... | adesola/192 1... | pcap.packet ... | 66 | 66 | 1298196402 | 38347 | | |
| 97 | 6 | 173.194.12.102 | 192.168.30.234 | 80 | 49658 | adesola/192.... | adesola/192 1... | pcap.packet ... | 1514 | 1514 | 1298196413 | 143570 | | |
| 98 | 6 | 192.168.30.234 | 173.194.12.102 | 49658 | 80 | adesola/192.... | adesola/192 1... | pcap.packet ... | 54 | 54 | 1298196413 | 142567 | | |
| 99 | 6 | 173.194.12.102 | 192.168.30.234 | 80 | 49658 | adesola/192.... | adesola/192 1... | pcap.packet ... | 1514 | 1514 | 1298196413 | 406580 | | |
| 100 | 6 | 192.168.30.234 | 173.194.12.102 | 49658 | 80 | adesola/192.... | adesola/192 1... | pcap.packet ... | 66 | 66 | 1298196413 | 407569 | | |
| 101 | 6 | 173.194.12.102 | 192.168.30.234 | 80 | 49658 | adesola/192.... | adesola/192 1... | pcap.packet ... | 1514 | 1514 | 1298196413 | 669611 | | |
| 102 | 6 | 192.168.30.234 | 173.194.12.102 | 49658 | 80 | adesola/192.... | adesola/192 1... | pcap.packet ... | 66 | 66 | 1298196413 | 863615 | | |
| 103 | 6 | 173.194.12.102 | 192.168.30.234 | 80 | 49658 | adesola/192.... | adesola/192 1... | pcap.packet ... | 1514 | 1514 | 1298196414 | 126623 | | |
| 104 | 6 | 192.168.30.234 | 173.194.12.102 | 49658 | 80 | adesola/192.... | adesola/192 1... | pcap.packet ... | 74 | 74 | 1298196414 | 127589 | | |

Figure. 4.10: A Network administrator interface

## V. CONCLUDING REMARKS

We have developed a tool that provides Network Administrators and parents the opportunity to monitor network usage and content being consumed by users. The tool is able to monitor events transition over a network whether on a client PC or Office Pc through use of packets capture and extracting vital information from it; which can help parents to have an idea of what their children do have access to on the internet. It can also work in tandem with most network protocols in order to limit the complexity of the packet monitoring and ensure low packet loss rate.

## VI. FUTURE WORKS

Future work will enhance this tool to be more scalable and interactive. The existing system serves as a basis for future developments in this dynamic domain of research by acting as a foundation providing background knowledge in packets capturing that other researchers can build on.

## VII. ACKNOWLEDGEMENT

The contributions of Mr. Adesi Adesola is appreciated for programming the client interface.

## VIII. REFERENCE

[1] Smith, M. & Van, J. (1993). The BSD Packet Filter: A New Architecture for User-level Packet Capture. Proceedings of the 1993 Winter USENIX Technical Conference San Diego, CA, Jan. 1993.

[2] Marcus J., Ranum, K., Landfield, M., Stolarchuk, M., Sienkiewicz, Andrew, L. & Eric, W. (1997). Implementing a Generalized Tool for Network Monitoring. Proceedings of the Eleventh Systems Administration Conference (LISA97) San Diego, California, October 1997. Network Flight Recorder Incorporation.

[3] Loris, D. (2000). Development of an Architecture for Packet Capture and Network Traffic Analysis, Graduation Thesis, Politecnico Di Torino. Turin, Italy, Mar. 2000. pp 12-15

[4] Rebecca, B. (2002) Detection of break-ins or break in. Curtin University Press.

[5] Simpson. W. (1993) The Point-to-Point Protocol (PPP), RFC 1548, Daydreamer Incorporation December 1993. pp7-9

[6] Winpcap (2012). The Wincap web site. Available at http://netgroup-serv.polito.it/winpcap

[7] Degioanni, L.; Baldi, M.; Risso, F.; Varenni, G.; Dipt. di Autom (2003). Profiling and Optimization of Software-Based Network-Analysis Applications. Proceedings. 15th Symposium on Computer Architecture and High Performance Computing, 2003. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1250342

[8] Steven McCanne and Van Jacobson (1995). A Flexible Framework for Packet Video. ACM Multimedia 1995: p 511-522

[9] Curtin M. and Ranum M.J. (1994). Internet Firewalls; Addison Wesley Longman. Pp 7

[10] Avolio, F. and Ranum, M. (1994). A network perimeter with secure external access. In Proceedings of the Internet Society Symposium on Network and Distributed System Security, San Diego, CA, February 3, 1994.

[11] Scott Fuller, Kevin Pagan (1997). Internet Firewalls Ventana Communications Group Inc. January 1997. ISBN: 1566045061

[12] Treese, W. and Wolman, A. (1993) Through the firewall, and other application relays. In USENIX Conference Proceedings, Cincinnati, OH, June 1993.pages 87-99.

[13] Bellovin S., Cheswick W. & Rubin A. (2003). Firewalls and Internet Security: Repelling the Wily Hacker, Second Edition; Addison Wesley Longman.

[14] Luis, M.G. (2008), Programming with Libpcap Sniffing the Network from Our Own Application. http://www.citeulike.org/user/shtrom/article/9201670

[15] Tom, K. (2006) Packet Sniffing In a Switched Environment. SANS Institute of Info-Security. http://www.sans.org/reading_room/whitepapers/networkdevs/packet-sniffing-switched-environment_244

[16] Ryan, S. (2003) Packet Sniffer Detection with AntiSniff. University of Wisconsin – Whitewater, Department of Computer and Network Administration.

[17] Mike, S., George, B. & Craig, F. (2008) Homebrew Network Monitoring: A Prelude to Network Management. Curtin University Press

[18] Microsoft Corporation, 3Com Corporation, NDIS, Network Driver Interface Specification, May 1988. www.microsoft.com