ALGORITHM FOR INTELLIGENT NETWORK VISUALIZATION AND THREAT ANALYSIS

Amit Kumar, Prashant Sharma, Dr. Shishir Kumar

Department of CSE, Jaypee University of Engineering & Technology, Guna (MP) INDIA

amitrathi10@yahoo.co.in
sharma_prashant10@yahoo.co.uk
dr.shishir@yahoo.com

Abstract - Network security is an intriguing field which perplexes many and escapes more. Unfortunately, the issues surrounding network security are not such that we can afford for people either to ignore or to disregard policy and precautions put in place to protect users of a computer network. Network security is a difficult problem. As time progresses, user-bases increase dramatically and with them they bring application diversity and a larger variation in knowledge. Professionals in the I.T. industry are as guilty of ignorance as young children or the elderly when it comes to threats to security posed by networks. While it is commonplace for new users of Microsoft Windows to learn where to adjust screen settings or how to install a printer on their computers from the 'Getting Started Tour', there is no manual, memo or video distributed to new users of computer systems to teach them of the dangers of networks and network-powered applications. System administrators constantly run up against conflicts of interest between good business and good security and more often than not, avoiding disruption to or complication of the business triumphs over implementing secure policies. It is this behavior that makes it so important that users are well informed and take responsibility for their actions. This paper is concerned with development of application that helps user to visualize their network and analyze the serious threats. Our application serves as an attempt to provide an intuitive application to visualize threat that exit when ever user logon to computer attached to any network.

Key Words
Threat, security, monitoring, packet, port.

I. INTRODUCTION

Network security[2][5] is an intriguing field which perplexes many and escapes more. Unfortunately, the issues surrounding network security are not such that we can afford for people either to ignore or to disregard policy and

precautions put in place to protect users of a computer network. The title of this paper refers to an application developed to help users to visualize their network and analyze the serious threats.

The digital age has excelled at bringing the power and benefit of globally shared knowledge to anyone willing to look for it on the Internet. The dependency on the high-capacity links we take for granted for access to our email, work, banking or other favorite repositories is immense and while we progress in the development of technology to widen the reach of the Internet, the speed of wide area connections available to users is rapidly approaching local network rates[6][7]. With the number of users of the Internet ever-increasing, it would be foolish to assume that the number of crimes being committed in this hostile environment was not spiraling exponentially. The harsh reality is that a massive number of people who utilize the network, be it at work, school or home, do so without the basic skills and knowledge required to protect themselves or those they are exposed to from the potentially malicious actions or intentions of others.

Here we attempt to provide users of all ages and backgrounds with an intuitive application to visualize that exist whenever users logon to the computer attached to any network. The aim of paper is to aware the user about intrusion and assist him in detecting the threats[1][2]. Humans have been shown to abide by and enforce policies, rules and laws far more effectively when they understand the reasons for their introduction and agree with the goal. The aim of educating users is to increase awareness and as a result to decrease the likelihood of an intrusion as a result of their actions or negligence.

As explained above the urgent need of this system we have analyzed what all already exist and what is further required from this very system that is not being delivered by the pervious similar systems.

II. BLOCK DIAGRAM

Block diagram for the visualization and threat analysis system is shown in fig 1.

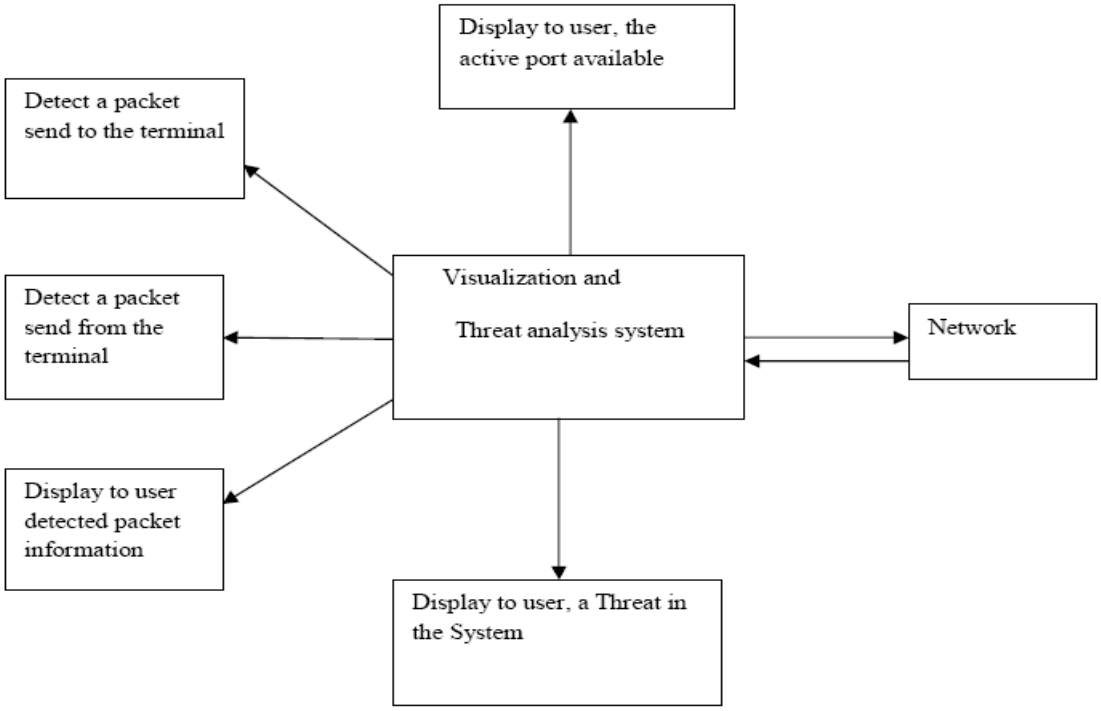

Fig. 1Block Diagram fro Visualization and Threat Analysis

III. DATA FLOW DIAGRAM

A. Context Diagram

This diagram explains the sole principle of the paper that with which object the Network Visualization and Threat analysis would be interacting and the output of the paper. As we can clearly see in the figure below the software is interacting with the user system. LAN network, analyses of packets and output for the same would be the log file.

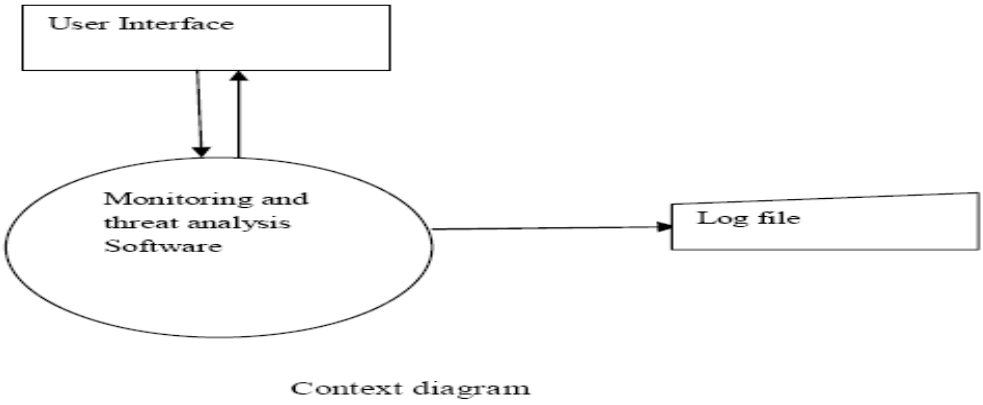

Fig. 2 Context DFD

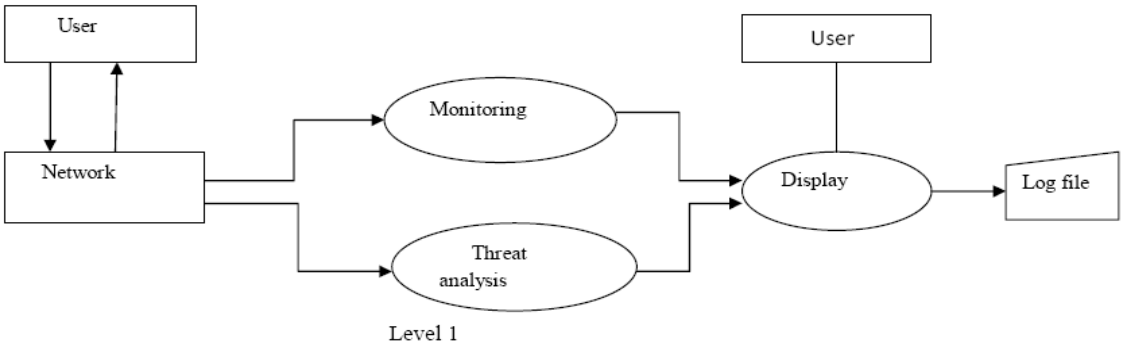

Fig. 3 DFD for Level 1

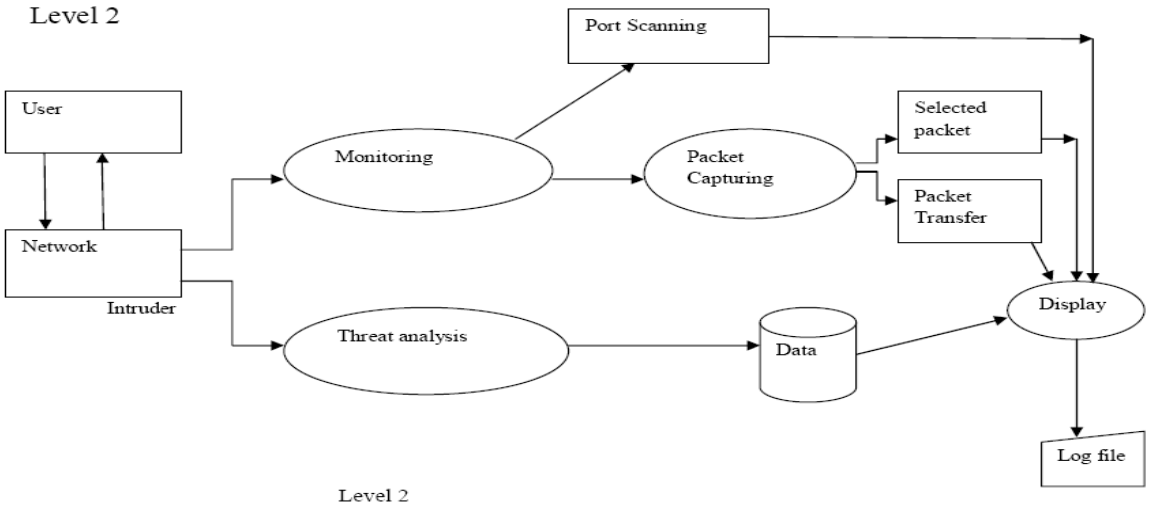

Fig. 4 DFD for Level 2

IV. FLOW CHART FOR MONITORING AND VISUALIZATION

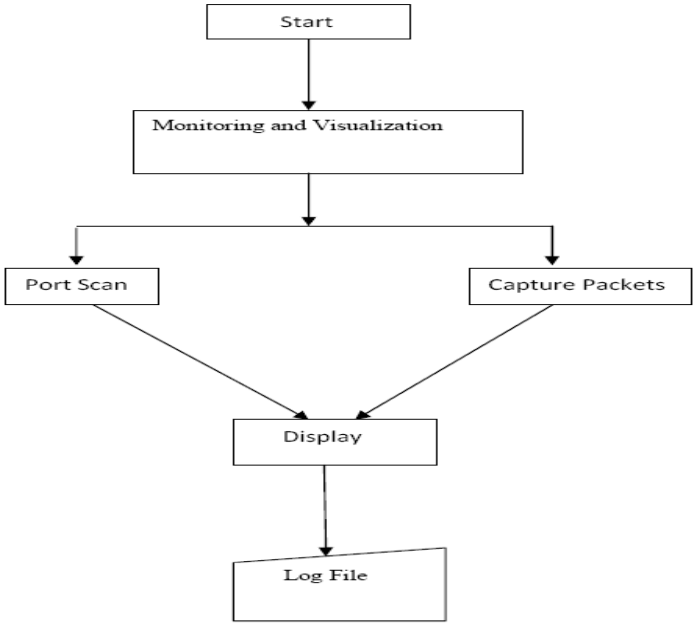

Fig. 5 Flow Chart for Monitoring and Visualization

V. CLASS DIAGRAM FOR PORT SCANNING CLASS

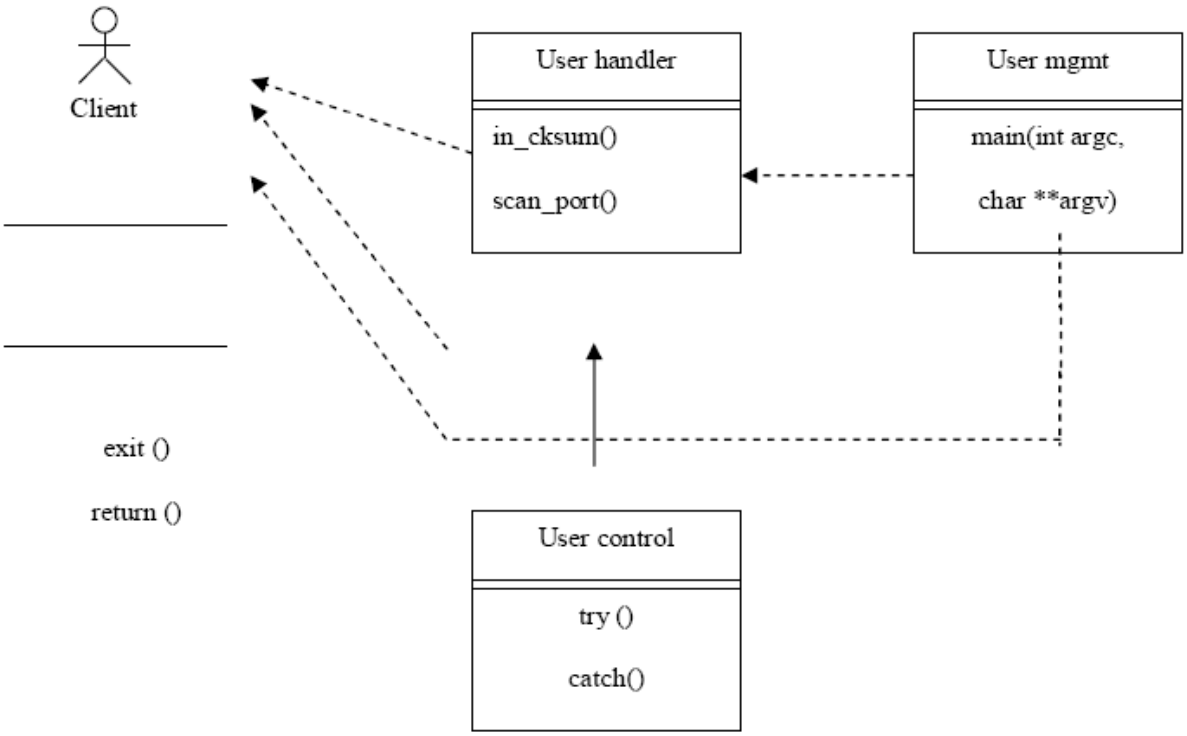

Fig. 6 Class diagram for port scanning

VI. PROPOSED ALGORITHM

A. Algorithm for Port Scanning

Step 1: Variable Declaration

1.1 Declare InetAddress = null;

1.2 Declare String host = null;

Step 2: Input

2.1 Enter value of Host name.

2.2 Put value of Host name to InetAddress by InetAddress.getByname().

Step3: Scanning

3.1 Declare variable port = 0.

3.2 Declare initial port = value.

3.3 Declare final port = value.

3.4 Check if the port is available between initial port and final port.

3.5 port ++

3.6 Repeat step 3.4 upto final port.

Step4: Display

4.1 Display all the active ports in GUI format on screen in active port list.

Step 5: End

B. Algorithm for packet capturing

Step 1: Obtaining the list of network interfaces

1.1 Create an array named ‘devices’

1.2 Detect network interfaces present in user system.

1.3 Store the above list in the ‘devices’ variable.

Step 2: Displaying the list of network interfaces

2.1 integer variable i=0

2.2 Till the value of i is less than the length of the array ‘devices’, goto Step 2.3 else go

2.3 print out the name and description of the captured NI.

2.4 i=i+1

2.5 Goto Step 2.2

Step 3: Open the network interface.

3.1 Declare integer variable J=0

3.2 Till J < devices array's length, Goto Step 3.3 else Goto Step 3.7

3.3 Check if the network interface at J^{th} index number in devices array is selected.

If yes goto Step 3.6 else goto Step 3.

3.4 J=J+1

3.5 Goto Step 3.2

3.6 Open the selected network interface i.e. NI at J^{th} index, then Goto Step 4

3.7 Display that the network interface has not yet been selected by the user.

3.8 Goto Step8

Step 4: Capture packets from the network interface

4.1 Is the menu button of stop capture packet selected?

If yes goto Step 3.8 else goto Step 4.2

4.2 Capture the upcoming single packet from the network.

4.3 Display the captured packet by going to Step 5

Step 5: Display the captured packet to the user in proper GUI format.

5.1 Detect user's menu choice of the format in which captured packet's to be displayed

5.2 Analyze the packet. If choice is Hexadecimal format

Then, Display in Hexadecimal format

Else, Display in char format

5.3 Goto Step 6 to save the packets to a temporary file

5.4 Go back to Step 4.1

Step 6: Save captured packets into a file

6.1 Create a file named 'temp'.

6.2 Create class objects to write captured packets data into the 'temp' file.

6.3 Save captured packets into the opened file

6.4 Go back to Step 5.4

Step 7: Close all the open network interface

7.1 If user selects the menu choice of making file permanent,

Then rename the file to the user's given name.

Else, delete the temporary file 'temp'.

7.2 Close the network interface.

Step 8: End

VII. PSEUDO CODES

A. Pseudo code for opening of network interfaces using JPCAP functions

```
//Obtain the list of network interfaces
NetworkInterface[] devices = JpcapCaptor.getDeviceList();
for ( i = 1 to devices.length)  //for each network interface
System.out.println(i+": "+devices[i].name + "("+ devices[i].description+")");
//print out its datalink name and description
System.out.println(" datalink: "+devices[i].datalink_name + "(" + devices[i].datalink_description+")");
System.out.print (" MAC address:");  //print out its MAC address
for (byte b : devices[i].mac_address)
System.out.print (Integer.toHexString(b&0xff) + ":");
System.out.println ();
//print out its IP address, subnet mask and broadcast address
for (NetworkInterfaceAddress a : devices[i].addresses)
System.out.println (" address:"+a.address + " " + a.subnet + " "+ a.broadcast);
```

The following steps obtain the list of network interfaces and prints out their information:

1. First of all prepare an array list of all the network interfaces on the node.
2. Now for getting the device description we have to print the datalink description, MAC address, IP address, subnet mask in a file. And then print all the data of file
3. Finally we will take any one interface open it and will check for packets entering through it.

B. Pseudo code for capturing of packet in the network

```
//this method is called every time Jpcap captures a packet
public void receivePacket(Packet packet) {
//just print out a captured packet
System.out.println(packet);
//call processPacket() to let Jpcap call PacketPrinter.receivePacket() for every packet capture.
captor.processPacket(10,new PacketPrinter());
captor.close();
```

VIII. RESULTS AND DISCUSSION

Comparison between exiting and proposed system

The gathering of information being done via different ways till date as checking the remote information of the system at the system itself or asking about the same to the administrator of the web server and it was painstaking to perform the task. We have designed an algorithm called **"Algorithm for Intelligent Network Visualization and Threat Analysis"** with the goal to ease the work of administrators and other users. The system is easy to use and also acts as a guide to both user alone and administrators.

The concept of this paper will allow the administrator and user to access the remote information via a same tool the user will not have to go here and there to access and collect the information but the same is available at his own system itself and this all is available in a easy and secure way. Means there is reduction in time, stress, and increased flexibility, and information.

IX. CONCLUSION AND FUTURE WORK

We were successful in capturing packet and scanning all active ports with help of network visualization. The other part of the paper was able to successfully deliver good results about web server information related to administrator. In future we look for some other facilities in this paper that are:

- More User Familiar graphical interface that will help novice user
- Addition functionality by exploring system registry.
- Implementation of Security.
- Simulate Same concept of this paper on other OS also.

Future Enhancements

There are certain features that would have enjoyed to incorporate in our paper .Following features can be enhanced:

- Auto-run facility
- Looks
- Threat analysis tool

We would very much have liked to make the software auto run. When we say we want our algorithm to be auto run we mean to say that it should run by itself as soon as the system is booted. If the algorithm is implemented then we can provide a good look to the software based on our algorithm. Threat analysis tool can be improved as according to the application domain.

REFERENCES

[1] [BAC99] Bace, Rebecca, "An Introduction to Intrusion Detection and Assessment: for System and Network Security Management," ICSA White Paper, 1998.

[2]http://www.windowsecurity.com/whitepaper/FAQ_Network_Intrusion_Detection_Systems_. Html

[3] http://www.intrusion.com/Default.aspx?DN=bee1192e-5a5b-4a44-b653-efce9f846523

[4] www.linuxsecurity.com/resource_files/intrusion_detection/network-intrusion-detection.html

[5]www.windowsecurity.com/articles/What_You_Need_to_Know_About_Intrusion_Detection_Systems.html

[6] Computer Networks, Fourth Edition by Andrew S. Tanenbaum (Prentice Hall PTR), Vrije Universiteit, Amsterdam, The Netherlands

[7] Data Communications and Networking, Fourth Edition by Behrouz A Forouzan

[8] Software Engineering – A Practitioner's Guide, Sixth Edition by Roger S. Pressman (McGraw –Hill International Edition)

[9] D. Comer (Ed.), Internetworking with TCP/IP: Principles, Protocols and Architecture, Prentice-Hall, Upper Saddle River, New Jersey, 1991.

[10] D. Denning, Information Warfare and Security, Addison Wesley, Reading, Massachusetts, 1999.

[11] M. Allman, "A Web Server's View of the Transport Layer," Computer Communication Review, 30(5), Oct. 2000.

[12] H. Balakrishnan, S. Seshan, M. Stemm, and R. Katz, "Analyzing Stability in Wide-Area Network Performance," In Proc. ACM SIGMETRICS' 97,June 1997.

[13] M. Crovella and A. Bestavros, "Self-similarity in World Wide Web Traffic: Evidence and Possible Causes," IEEE/ACM Transactions on Networking, 5(6):835-846, December, 1997.

[14] C. Cranor, T. Johnson, V. Shkapenyuk, and O. Spatschek. Gigascope: High performance network monitoring with a SQL interface. Sigmod 2002 demonstration, 2002.

[15] R. Buyya. PARMON: a portable and scalable monitoring system for clusters. Software - Practice and Experience, 30(7):723–739, 2000.

[16] D. Carney, U. Cetintemel, A. Rasin, S. B. Zdonik, M. Cherniack, and M. Stonebraker. Operator scheduling in a data stream manager. In VLDB, 2003.