International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:6, 2008

Tool Tracker: A toolkit ensembling useful online networking tools for efficient management and operation of a network

Onkar Bhat Kodical, Sridhar Srinivasan and N.K. Srinath

***Abstract*—** Tool Tracker is a client-server based application. It is essentially a catalogue of various network monitoring and management tools that are available online. There is a database maintained on the server side that contains the information about various tools. Several clients can access this information simultaneously and utilize this information. The various categories of tools considered are packet sniffers, port mappers, port scanners, encryption tools, and vulnerability scanners etc for the development of this application. This application provides a front end through which the user can invoke any tool from a central repository for the purpose of packet sniffing, port scanning, network analysis etc. Apart from the tool, its description and the help files associated with it would also be stored in the central repository. This facility will enable the user to view the documentation pertaining to the tool without having to download and install the tool. The application would update the central repository with the latest versions of the tools. The application would inform the user about the availability of a newer version of the tool currently being used and give the choice of installing the newer version to the user. Thus ToolTracker provides any network administrator that much needed abstraction and ease-of-use with respect to the tools that he can use to efficiently monitor a network.

***Keywords*—** network monitoring, single platform, client/server application, version management

I. Introduction

Network monitoring for a corporate network is a critical IT function that can save money in network performance, employee productivity and infrastructure cost overruns. A network monitoring system monitors an internal network for problems. It can find and help resolve snail-paced webpage downloads, lost-in-space e-mail, questionable user activity and file delivery caused by overloaded, crashed servers, dicey network connections or other devices.

Network monitoring can be achieved using various softwares or a combination of plug-and-play hardware and software appliance solutions. Virtually any kind of network can be monitored. You can monitor devices on different operating systems with a multitude of functions, ranging from cell phones, to servers, routers and switches. These systems can help you identify specific activities and performance metrics, producing results that enable a business to address various and sundry needs, including meeting compliance requirements, stomping out internal security threats and providing more operational visibility.

II. Networking monitoring

A. Importance of Networking Monitoring

The reasons to insist on network monitoring can be summarized on a high level into maintaining the network's current health, ensuring availability and improving performance. A network monitoring software can also help you build a database of critical information that you can use to plan for future growth.

Reading network traffic is essential for system administrators, network engineers, and security analysts. At some point there will be a need to read the network traffic directly instead of monitoring application level details. Examples of situations that might require monitoring network traffic are, auditing network security, debugging network configurations, and analyzing usage patterns. For this task we use network monitoring software, or network sniffers, that sniff the traffic your computer is able to see on the network. What exactly your computer can see really depends on how the network is laid out, but the easiest way to figure out what it can see is just start sniffing.

The most common tools to do the job are readily available. Many are even free to download and use. Two of the most popular tools for monitoring network traffic are tcpdump and Wireshark. Both are freely available for Windows, Linux, and a variety of other platforms. There are also a number of specialized tools that perform different functions and suit different tasks. Some provide very basic and raw functionality of reading and parsing every layer out of the stream, others decode specific information they deem relevant out of the protocol, and others provide graphical stream tracing for viewing the flow of traffic.

B. Tools used for Network Monitoring

There are a plethora of free tools available online that can be

categorized into various categories some of which include packet sniffing tools, port mapping/scanning tools, encryption tools, security tools, vulnerability scanners.

A **Packet Sniffer** is a computer software or hardware that can intercept and log traffic passing over a digital network or part of a network[4]. As data streams flow across the network, the sniffer captures each packet and eventually decodes and analyzes its content according to the appropriate specifications. Packet sniffers can be used to detect network intrusion attempts, gain information for effecting a network intrusion, monitor network usage, gather and report network statistics, and filter suspect content from network traffic.

Encryption Tools are mostly used to secure or sign messages[5]. They protect the user's identity and provide the confidentiality of the data transmitted. Encryption tools are method of protecting secure information from hackers, by encrypting information. The only way of getting access to that information is by holding the decryption key. It is used in various fields ranging from protecting sensitive information or just protecting the privacy. Encryption tools use various ciphers to encrypt the information according to a key which then becomes the only way of decrypting the protected information. For example a user wants to upload something on a network drive but the information will be exposed to anyone that has access to that network. If the information is encrypted the user is safeguarded from information theft.

Vulnerability Scanners can be used to conduct network reconnaissance, which is typically carried out by a remote attacker attempting to gain information or access to a network he/she is not authorized or allowed[6]. Network reconnaissance is increasingly being used to exploit various network standards and automated communication methods in order to determine what types of computers are present, along with additional information about those computers, such as the type and version of its operating system. This information can be analyzed for known or recently discovered vulnerabilities that can be exploited to gain access to secure networks and computers.

A **Port Scanner** is a piece of software designed to search a network host for open ports[7]. This is often used by administrators to check the security of their networks and by hackers to compromise it. The information gathered by a port scan has many legitimate uses, including the ability to verify the security of a network. Port scanning can however also be used by those who intend to compromise security. Many exploits rely upon port scans to find open ports and send large quantities of data in an attempt to trigger a condition known as a buffer overflow. Such behavior can compromise the security of a network and the computers therein, resulting in the loss or exposure of sensitive information and the ability to do work.

C. Problems with Network Monitoring Tools

The lack of a single platform where the network monitoring tools can be found makes the process of searching for these tools a tedious task. After finding the required tool, the user may not be aware of the dependencies with respect to that tool. The absence of a mechanism, which informs of the latest version changes with respect to these tools, is also a hurdle towards using these tools effectively. There is no means by which a user can be informed of any new tool released online.

III. Comparisons With Existing Network Management Tools

There are several free network monitoring softwares available online[1]. These tools can be made available by means of a single platform in the form of the application developed by us, namely Tool Tracker. The tools can be launched and managed via this application which also provides the facility of managing version changes with respect to these softwares.

The tools pertaining to network operation and management require direct and efficient real-time access to data traveling on the network. Software tools are usually preferred due to their low cost and high versatility. But performance related problems on high-speed networks is a cause of concern with respect to software tools. The work presented in [3] demonstrates that, despite the common belief, the performance limits for software real-time network analysis tools has not yet been reached and it can be improved further with limited hardware support. Access to these quality tools can be provided by means of our application Tool Tracker which maintains a repository of the latest and widely used network monitoring tools.

NNStat[2] was designed as a statistical analysis system that has several attractive properties. It permits accurate summaries of an event, flexible specification of types of events to record, flexible storage of information about the events that are observed. NNStat has been desribed as a poor man’s intrusion detection system which sets alerts to fire when something happened which the network manager believed should not have happened.

NFR[2] is intellectually evolved from NNStat, but includes a more generalized and powerful filtering language, as well as the ability to trigger alerts and log complete packet information. NFR is to be used as a platform for exploring auditing and logging, while simultaneously providing a freely available, high quality data source for researchers working on intrusion detection.

While NNStat and NFR aim at providing a means of performing statistical analysis on packets passing through a network for intrusion detection, Tool Tracker aims at

providing a single patform not only for packet analysis but for other operations on packets which may include encryption, or completely different tasks such as scanning of ports, use of steganographic tools thereby allowing the user to use the best tool for a particular scenario. With the availability of several free tools online from different sources, the user faces problems such as searching for the appropriate tool, assessing if the tool meets their requirements, and keeping track of changes in versions for each tool. Apart from the previously mentioned problems, the users also suffers from the absence of a single platform for accessing the various tools present locally in their systems. The application developed by us aims at eliminating all these problems therefore enabling the user to focus only on the operation and management of the network.

IV. Introduction to Tool Tracker

A. Overview of the application

The purpose of our application is to eliminate all the above mentioned problems that users face when they use free network monitoring/management tools from the internet. Tool Tracker is a catalogue of various network monitoring/management tools available on the internet for free. It groups the tools into various categories, hence making it easier for the user to search for specific tools. It provides facilities to view the documentation of a tool online. It provides a facility to launch any installed tool from a single platform. It helps manage version changes for all installed tools.

B. Functional Specification

Tool Tracker divides its end users into two categories: Basic Users and Admin Users. The functional specifications for the end users are as follows:

1) The Basic user should have access to all contents of the database, but should not be able to modify the contents.
2) The Basic user should be able to launch all the installed tools from this application .The application will ensure that the user is made aware of the availability of newer versions of installed tools.
3) A Basic user should be able to register as an Admin.
4) An Admin user should be able to perform all the functions of a Basic user. Apart from these, an Admin user should be able to authenticate himself with the server.
5) An Admin user should be able to Add, Delete, Update or Search for any tool in the database.
6) An Admin user should be able to accept/reject user requests for Admin privileges.

V. High Level Design

A. Design Considerations

There are three entities as follows:

1) Client
2) Server
3) Database

Data flow is divided into the process:

1) Client sending a query to the server.
2) Server querying the database.
3) Database returning the results of the query to an XML generator.
4) XML generator sending the generated XML file back to the client.
5) Client requesting for the Admin Pages that reside on the server.
6) Server rendering the generated HTML pages on the Client's browser.

B. Architectural Strategies

The high level architecture of Tool Tracker is as shown in Fig 1. The client establishes a socket connection with the server to access information about the tools. The information resides in a database on the server. The server generates an XML file containing the results of the database query, and sends it back to the client via the same socket connection. The server forks a new child process for every new client connection that it receives. The client then parses the XML file and displays the results on its GUI to the end user.

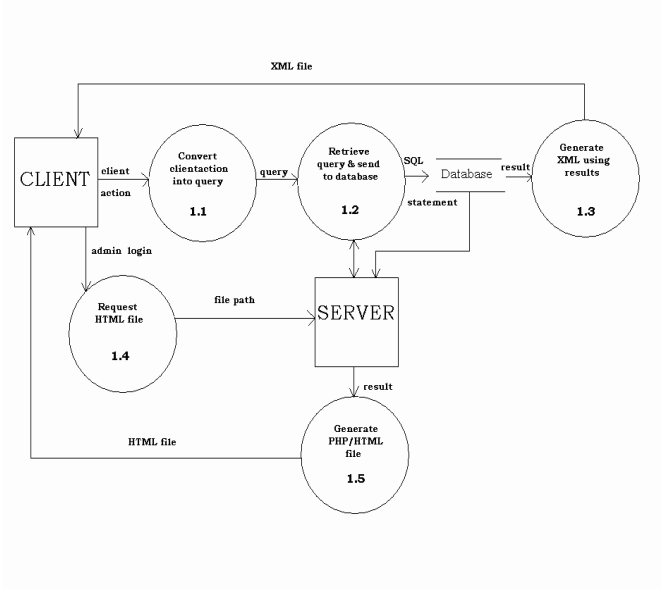

Fig. 1 Data Flow Diagram, Level 1

VI. Detailed Design

A. Activity Diagrams

Since ToolTracker categorizes its users into two categories, their activity diagrams are as shown in Fig 2 and 3.

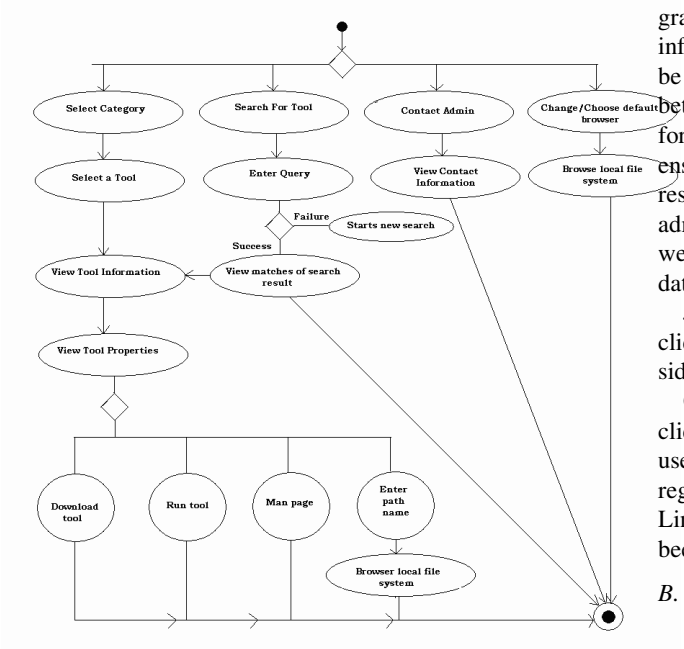

Fig. 2 Activity Diagram for a Basic User.

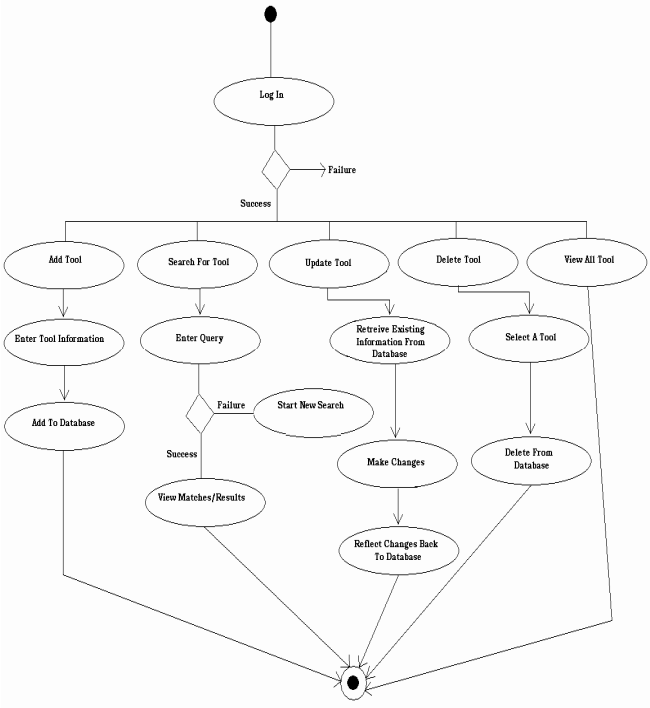

Fig. 3 Activity Diagram for an Admin user.

VII. Implementation of ToolTracker

A. Overview of the implementation process

ToolTracker was implemented as a client/server application which maintains a database of free network monitoring tools available online. It provides a simple graphical user interface to allow the user to view the information and make the right decision as to which tool can be used in a given scenario. It provides a level of abstraction between the user and the tools by means of a single platform for accessing the tools present in the database. The application ensures that the user is aware of the version changes with respect to the tools present in the database. It also allows the administrator of the application to log on to an administrator's website, to add, delete, and update any tool present in the database.

Java has been used for implementing the front end, the client server communication and the xml parser on the client side.

On the server side, Java has been used for implementing the client server communication. Pre Hypertext Processor has been used for implementing the xml generator. The information regarding each tool is stored on a database on the server. Linux, Apache, MySQL and Pre Hypertext Processor have been used to implement the administrator's web pages.

B. Challenges faced during implementation

There were various challenges that we faced during the development of this application. Some of them are listed below:

1) Listening to multiple client requests and servicing them simultaneously with the help of a robust multi-threaded model.
2) Generating an XML file from the results of the database query on the server.
3) Waiting for the complete XML file to arrive on the client machine from the server before commencing the parsing process.
4) Parsing the XML file in order to display the Tool information on the client's GUI

C. Strategies used to tackle difficulties

These are the strategies and techniques we used to tackle the difficulties mentioned above.

1) We created a JAVA program on the server side that constantly listens on a pre-defined port for client requests.
2) Once a request has arrived, we fork a new process to handle the request.
3) The forked process converts the client request into a database query and invokes an appropriate PHP Script that queries the database for results.
4) Then, another dedicated PHP Script is invoked that converts the database results into an XML file.
5) Once the XML file is created, the background JAVA Program pushes the file into a buffer and in-turn back to the client that requested for the data.
6) In the client side, once a request has been sent, we make fork another process that waits for the XML file to arrive on the client's buffer. Until a notification occurs from the forked process, we make the main program sleep.

7) After receiving the full XML file we use an in-built XML Parser provided in the JAVA development Toolkit.
8) The XML Parser gives out a DOM (Document Object Model) object as a result of the parsing process.
9) We then use this DOM object and display the requested contents on the client's GUI.

VIII. Conclusion

A. Conclusion

Effective management of various network monitoring and management tools on every users machine is becoming the need of the hour. Providing this much needed facility to the users helps them to easily monitor or manage networks, carry out simulations, test networking equipment and do many other functions without having to focus much on the tools that are needed to perform these functions. An automatic version management mechanism is very much needed as the users will be always in sync with the latest versions of tools that are available on the internet.

Tool Tracker provides these functionalities to a user in a very simple and innovative way using Open Source technologies. It caters to a wide range of users as it comes with different versions for Windows and Linux Operating Systems. It also provides the much needed level of abstraction that a user would expect from this kind of an application. By segmenting the users into Basic and Admin categories, the application ensures that no unauthorized access is allowed, thereby always making sure that only legitimate content is stored in the database and no harm is done to the clients' machines.

B. Limitations

Although Tool Tracker has the necessary functionality and features in it to cater to a large segment of users, there are some limitations.

1) As of now, we haven't extended this application to cater to users of other Operating Systems like Macintosh, etc.
2) After installation of a particular tool, the user has to manually enter the path-name of its executable; this is done in order to launch the tool from the application in the future. This can be seen as a limitation, but it provides the user full freedom to install the tool in any location on his/her machine.
3) This tool can be implemented as a Web service by using SOAP (Simple Object Access Protocol) this will make the client log into the application over the internet, thereby not having to install the application locally.

Appendix

Some of the snapshots of the ToolTracker client and server applications running are shown in the figures below.

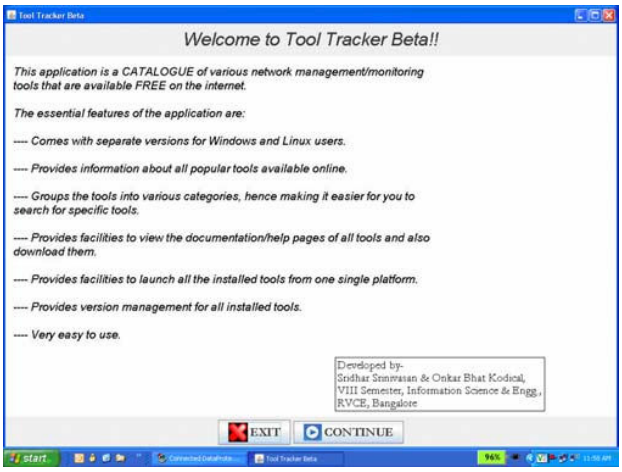

Fig. 4 ToolTracker Client Welcome Page.

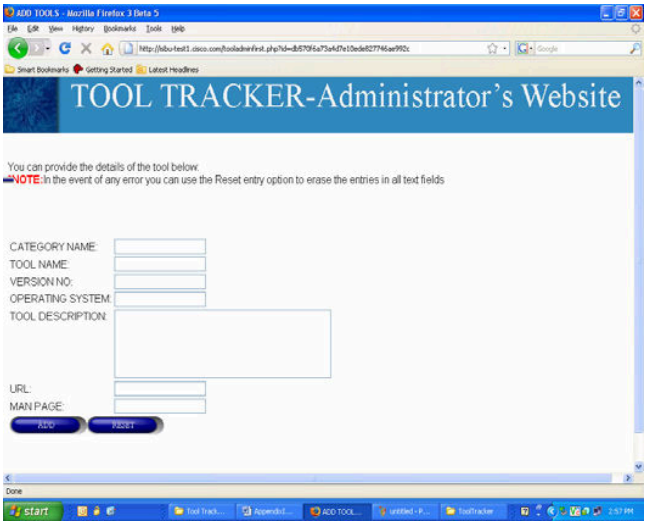

Fig 5 ToolTracker Administrator's Page

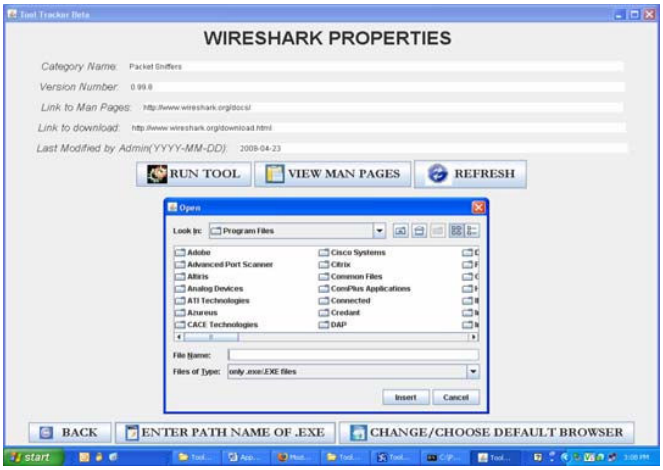

Fig. 6 Tool Properties on ToolTracker Windows client

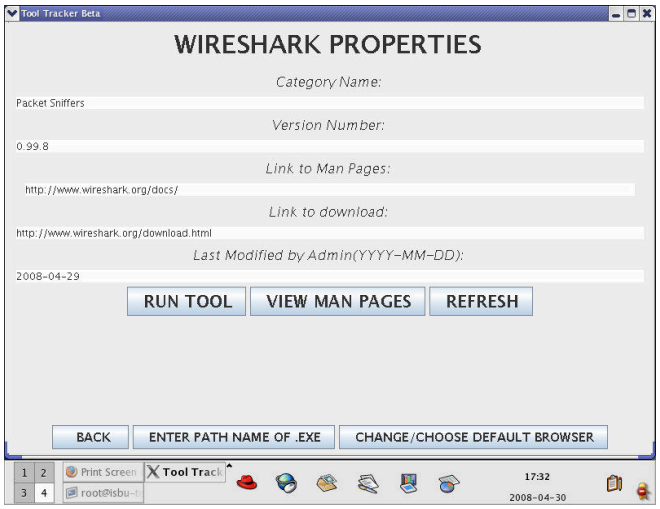

Fig. 7 Tool {Properties on ToolTracker Linux client

ACKNOWLEDGMENT

Our project was the result of the encouragement of many people who helped in shaping it by providing their feedback, direction and valuable support. It is with hearty gratitude that we acknowledge their contributions to our project.

We would like to thank our guide ***Mrs. Minal Moharir***, Lecturer, Department of Information Science & Engineering, RVCE, for the constant help and support extended towards us during the course of the project. We are also grateful to our external guides, ***Ms. Lavanya Ramani***, Software Engineer, CISCO Systems (India), ***Mr. Sriram Murthy***, Software Engineer, CISCO Systems (India) and ***Mr. Shyam Murthy***, Technical Leader, CISCO Systems (India) for guiding us throughout this project and encouraging us.

REFERENCES

[1]AdventNET Inc. , Free Network Monitoring Software for Small Networks (Whitepaper), 2005.

[2] Marcus J. Ranum, Kent Landfield, Mike Stolarchuk, Mark Sienkiewicz, Andrew Lambeth, and Eric Wall (Network Flight Recorder, Inc.) , "Implementing a Generalized Tool for Network Monitoring", LISA '97, San Diego, California, October 1997.

[3] Loris Degioanni, Mario Baldi, Fulvio Risso, and Gianluca Varenni, "Profiling and Optimization of Software-Based Network Analysis Applications", Proceedings of the 15th IEEE Symposium on Computer Architecture and High Performance Computing (SBAC-PAD03), pg 226-234, Sao Paulo, Brazil, November 2003.

[4] A. R. Bharambe, M. Agrawal, and S. Seshan. Mercury: Supporting scalable multi-attribute range queries. In *Proceedings of SIGCOMM 2004*.

[5] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, R. Rockell, D. Moll, T. Seely, and C. Diot. Packet-level traf_c measurements from the Sprint IP backbone. *IEEE Network*, 2003.

[6] G. Iannaccone, C. Diot, D. McAuley, A. Moore, I. Pratt, and L. Rizzo. The CoMo White Paper. Technical Report IRC-TR-04-017, Intel Research, Sept. 2004.

[7] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley. Inferring TCP connection characteristics through passive measurements. In *Proceedings of IEEE Infocom*, Mar. 2004.

[8] T. Karagiannis, A. Broido, M. Faloutsos, and K. claffy. Transport layer identi_cation of P2P traf_c. In *Proceedings of ACM Sigcomm Internet Measurement Conference*, Oct. 2004.

[9] K. G. Anagnostakis, S. Ioannidis, S. Miltchev, J. Ioannidis, M. B. Greenwald, and J. M. Smith. Efficient packet monitoring for network management. In *Proceedings of the 8th IEEE/IFIP Network Operations and Management Symposium (NOMS)*, April 2002.

[11] K. G. Anagnostakis, S. Ioannidis, S. Miltchev, and J. M. Smith. Practical network applications on a Leight weight active management environment. In *Proceedings of the 3rd IFIP International Working Conference on Active Networks (IWAN)*, October 2001.

[12] L. Deri and S. Suin. Effective Traffic Measurement using ntop. *IEEE Communications Magazine*, 38(5):138–145, May 2000.

[13] Gordon Lyon, website at : http://www.sectools.org/sniffers.html

[14] Gordon Lyon, website at : http://sectools.org/crypto.html

[15] Gordon Lyon, website at : http://sectools.org/vuln-scanners.html

[16] Gordon Lyon, website at : http://sectools.org/port-scanners.html

Onkar Bhat Kodical has completed his Bachelor of Engineering in Information Science from R V College of Engineering, Bangalore, that is affiliated to Visvesvaraya Technological University, Belgaum, Karnataka, in the year 2008. He worked as an intern at CISCO Systems India Pvt Ltd in the year 2008 for a period of four months. He has also interned at the Centre for Mathematical Modeling and Computer Simulation, National Aerospace Laboratories in the year 2006 for a period of two months.

Sridhar Srinivasan has completed his Bachelor of Engineering in Information Science from R V College of Engineering, Bangalore, that is affiliated to Visvesvaraya Technological University, Belgaum, Karnataka in the year 2008. He worked as an intern at CISCO Systems India Pvt Ltd in the year 2008 for a period of four months. He has also interned at the Centre for Mathematical Modeling and Computer Simulation, National Aerospace Laboratories in the year 2006 for a period of two months.

Prof. N.K. Srinath B.E, MTech. (Ph.D), Professor and HOD, Department of Information Science and Engineering, Mysore Road, Bangalore. E-mail: Srinath_nk@yahoo.com.