International Journal of Research in Computer Science
eISSN 2249-8265 Volume 2 Issue 1 (2011) pp. 33-38

www.ijorcs.org

NETWORK SECURITY USING LINUX INTRUSION DETECTION SYSTEM

Arul Anitha,
Assistant Professor, Department of Computer Science,
Jayaraj Annapackiam College for Women (Autonomous), Periyakulam, Tamil Nadu, India

Abstract *- Attacks on the nation's computer infrastructures are becoming an increasingly serious problem. Firewalls provide a certain amount of security, but can be fooled at times by attacks like IP spoofing and the so called authorized users. So an intelligent system that can detect attacks and intrusions is required. The tool GRANT (Global Real-time Analysis of Network Traffic) being a Linux based Intrusion Detection System(LIDs), takes the advantage of the security of a Linux box and secures the other nodes in the perimeter of the network. It is capable of detecting intrusions and probes as and when they occur and capable of responding to "already" successful attacks, thus causing minimal or no damage to the entire network. For better performance, this Linux Intrusion Detection System should be part of a defense in depth strategy such as Firewall and Intrusion Prevention.*

Keywords: *Intrusion, attack, security, GRANT, LIDS*

I. INTRODUCTION

The literature and media abound with reports of successful violations of computer system security by both external attackers and internal users [1]. Intrusions are caused by attackers accessing the systems from the Internet, authorized users of the systems who attempt to gain additional privileges for which they are not authorized, and authorized users who misuse the privileges given them.

In earlier days, intruders were the system experts. They had a high level of expertise and personally constructed methods for breaking into systems. Today, absolutely anyone can attack a network due to the widespread and easy availability of intrusion tools and exploit scripts that duplicate known methods of attack. Intrusions and the damage they cause can occur in a matter of seconds. Nowadays, Intruders are able to totally hide their presence by, for example, disabling commonly used services and reinstalling their own versions, and by erasing their tracks in audit and log files [2].

Firewall provides a certain amount of security, but can be fooled at times by attacks like IP spoofing and the so called authorized users**.** It also does not always protect the server against the attacks because for a business to function properly behind a Firewall, some popular ports such as 80 and 21 on the Firewall will need to be left open, hence opening the doors for attacks [3]. As Figure 1 indicates, year by year, there is a dramatic increase in the number of security incidents reported. Here, one security incident involves hundreds or thousands of sites.

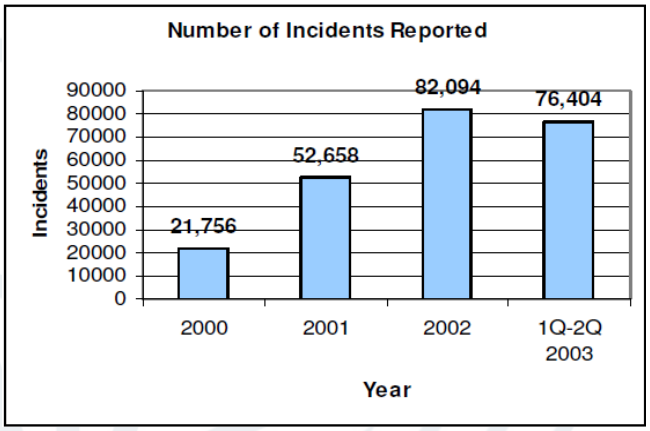

Figure 1. Number of Incidents Reported
Source: Computer Emergency Response Team (CERT)

Intrusion detection systems (IDSs) are software or hardware systems that automate the process of monitoring the events occurring in a computer system or network, analyzing them for signs of security problems. IDS are increasingly a key part of system defense, often operating under a high level of privilege to achieve their purposes [4]. As network attacks have increased in number and severity over the past decade, intrusion detection systems have become a necessary addition to the security infrastructure of most organizations [5]. Intrusion detection is one of the major and efficient defense methods against attacks and incidents in a computer network and system. ID's goal is to characterize attack manifestations to positively identify all true attacks without falsely identifying non-attacks. [6][7]. When there is an incident or an attack is encountered, an intrusion detection system (IDS) monitors and collects data from a target system that should be protected, processes and correlates the gathered information, and initiates responses [8].

Like any other operating system, application level security flaws leave Linux vulnerable to a variety of

malicious attacks. Over the years, many tools and techniques have been developed to protect Linux hosts and mitigate the risk posed by intruders [9]. The tool GRANT (Global Real-time Analysis of Network Traffic) being a Linux based Intrusion Detection System (LIDs), takes the advantage of the security of a Linux box and secures the other nodes in the perimeter of the network. It can help any Linux system to be protected from the hackers in the world. It provides kernel-level auditing information [10]. The Linux software was immensely extended and improved so that the Linux-based IDS of today is a complete, modern operating system, which can be used by programmers and non-programmers alike.

A. IDS Architecture

Intrusion detection systems (IDS) take either network or host based approach for recognizing and deflecting attacks. In either case, these products look for attack signatures (specific patterns) that usually indicate malicious or suspicious intent. When an IDS looks for these patterns in network traffic then it is Network based Intrusion Detection System. When an IDS looks for attack signatures in log files, then it is host based.

Figure 2 shows the typical sensor-based network intrusion detection architecture. A sensor is used to "sniff" packets off of the network where they are fed into a detection engine which will set off an alarm if any misuse is detected. These sensors are distributed to various mission-critical segments of the network. A central console is used to collect the alarms from multiple sensors.

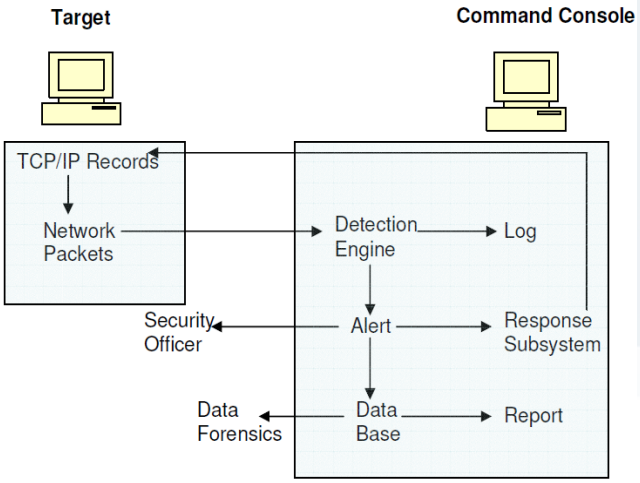

Figure 2: Typical Network IDS Architecture [4]

- When a computer in a network communicates another, the *network* packet is created.
- The *Intrusion* Detection System is used to detect the attacks and intrusion. If a pattern is detected, an alert is generated.
- The security officer is notified about the misuse. This *can* be done in a variety of methods such as audible, visual, pager, email or through any other method.
- A *response* to the intrusion is generated based on the *predefined* response or the responses from the security officer.
- The *alert* is stored for the later analysis.
- Reports are generated that summarize the alert activity.
- Data *forensics* is used to detect long-term trends.

II. LIDS WITH GRANT TOOL

Attackers are aiming at the Linux OS and other open-source applications because of the software's popularity. Even developers who believe they've adequately secured their development systems are looking at the trend with some trepidation [13]. Linux Intrusion Detection System is a suite of administrative tools that enhances security from within the Linux operating system's kernel. It implements several security features that are not in the Linux kernel natively [14]. A LID with GRANT tool enhances system security by reducing the root user's power. It also implements a low-level security model in the kernel. This Linux Intrusion Detection System with GRANT tool serves for the following purposes:

- *Security Protection:* This involves protecting the files and directories from unauthorized access on our hard disk. It also protects chosen files and directories from modifications by root user, so an unauthorized root access doesn't turn an intruder into a super villain. It protects important process from being terminated by anyone, including the root user and prevents raw I/O operations from access.
- *Incident Detection: LIDS* can detect when someone scans your system using port scanners and inform the system administrators via e-mail. LIDS can also notify the system administrator whenever it notices any violation of imposed rules and log detailed messages about the violations.
- *Incident-Response Capabilities:* LIDS can also take some necessary actions whenever any violations occur in the system. LIDS can not only log and send e-mail about the detected violations; it can even shut down an intruder's interactive session. It can disconnect a console whenever it finds a user trying to violate the rule.

III. WORKING WITH GRANT TOOL

The working principle of the GRANT tool is similar to the open source snort tool [15]. There are three main modes in which GRANT can be configured: sniffer, packet logger, and network intrusion detection system. Figure 3 shows the different types of modes and their options.

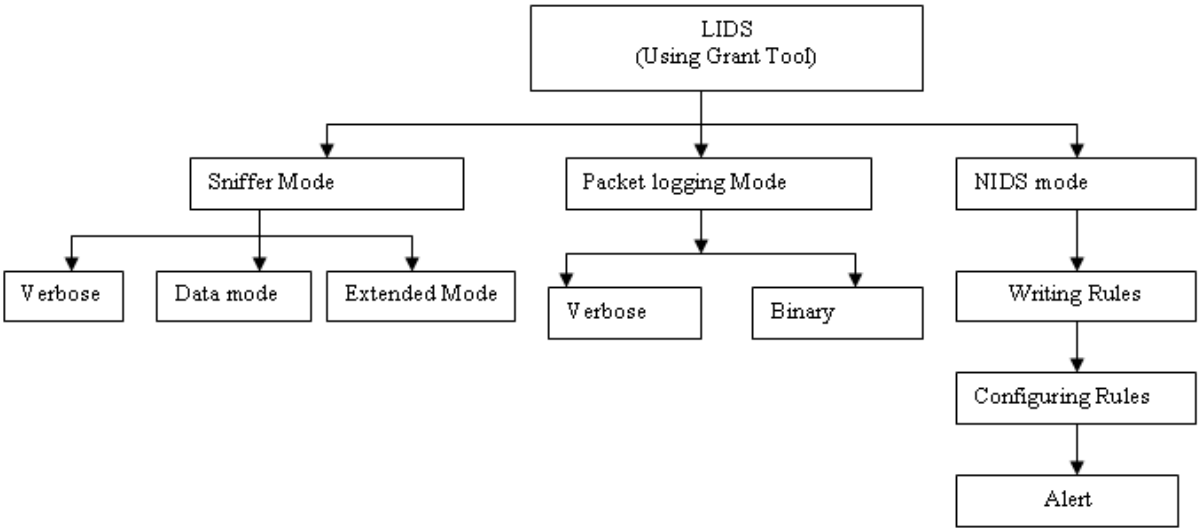

Figure 3: Different Modes of GRANT tool

- *Sniffer mode:* It *simply* reads the packets off of the network and displays them for you in a continuous stream on the console. The flow diagram of the sniffer mode is shown in figure 4. Sniffing can be done in three ways.
  - i) *Verbose mode:* It shows the IP address and TCP/UDP/ICMP headers.
  - ii) *Data mode:* It shows the packet data as well as headers.
  - iii) *Extended Details mode:* It shows the datalinklayer headers.

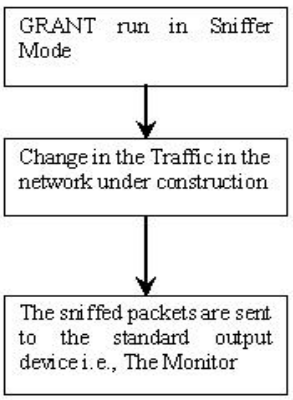

Figure 4. GRANT in Sniffer Mode

- *Packet Logger Mode:* Packet logger mode logs the packets to the disk as "log" file. An administrator can look at the log at leisure to have a clear idea of the traffic traversing his network. The packet logging can be done as shown in Figure 5.
- *Network IDS Mode:* Network intrusion detection mode is the most complex and configurable mode allowing Grant to analyze network traffic for matches against a user defined rule set and perform several actions based upon what it sees. How to run the GRANT tool in NIDS mode is shown in Figure 6.

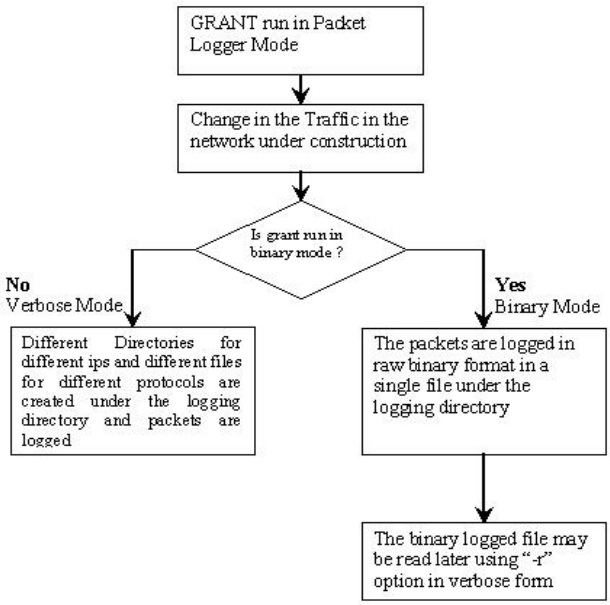

Figure 5. GRANT in Packet Logger Mode

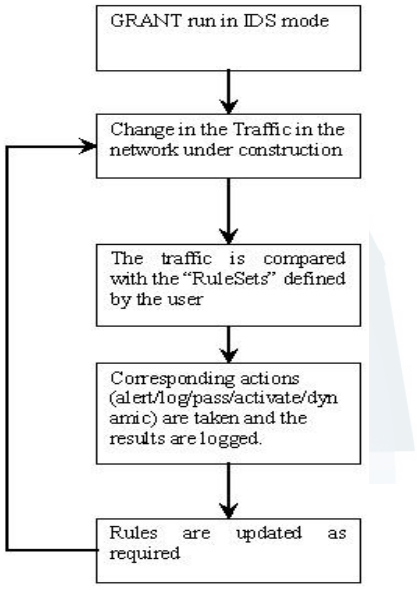

Figure 6. GRANT in NIDS Mode

Grant rules are divided into two logical sections, the rule header and the rule options. Table 1 gives the format of rule header with sample.

Table 1 - Rule Header Format with Sample Headers

Action	Protocol	Source IP	Source Port	Direction	Destination IP	Destination Port	Option
Alert	tcp	192.168.0.1	80	->	10.0.1.8	25	msg: "unauthorized"
Log	ftp	any	any	< >	any	any	ttl: 200

All Grant rule options are separated from each other using the semicolon ";" character. Rule option keywords are separated from their arguments with a colon ":" character Some supported rule option keywords are msg, content, dsize, logto, ttl, tos, etc., An example rule is given as follows:

alert icmp any any -> any any (msg: "Ping with TTL=200"; ttl: 200)

IV. RESULTS OBTAINED

A. GRANT with Verbose Option

The figure 7 shows result obtained by the GRANT running with verbose option. The verbose mode will display the IP and TCP/UDP/ICMP headers. The command that is used for the verbose mode is

./grant -v

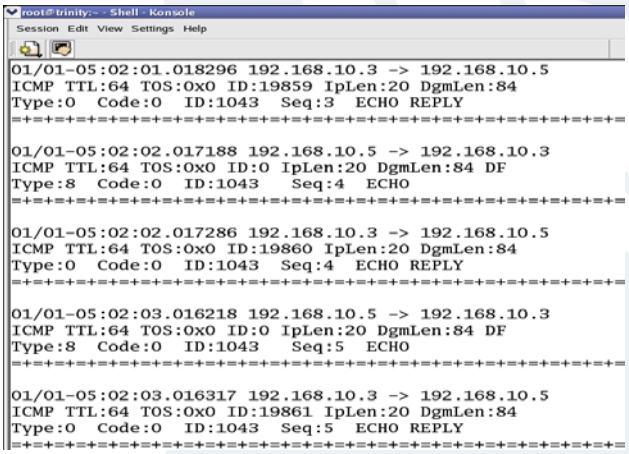

Figure 7. GRANT with Verbose option (v)

B. GRANT with Verbose and data option

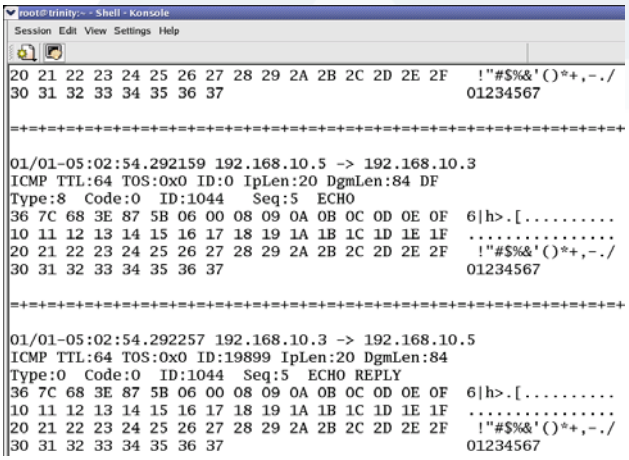

Figure 8. GRANT with Verbose and data option (vd)

As given in figure 8, the GRANT tool displays the packet data as well as the header details when it runs in verbose and data mode. This mode instructs Grant to display the packet data as well as the headers. The command that is used for this mode is

./grant -vd (or) ./grant -v -d

C. GRANT with Verbose, data and extended option

The figure 9 shows the result of GRANT when it runs with the verbose, data and extended option. This version displays the result with the headers, packet data and datalink layer details. The command that is used for this mode is

./grant -vde (or) ./grant -v -d –e

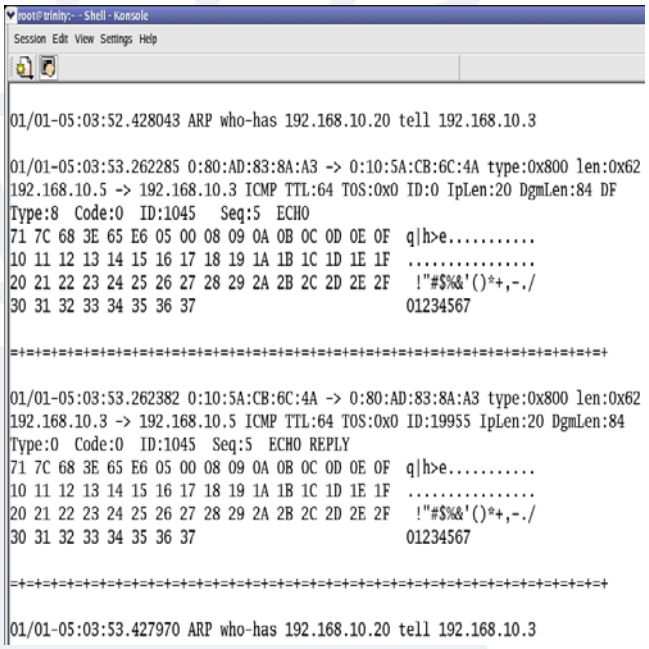

Figure 9. GRANT with Verbose, data and extended option (vde)

D. GRANT with Binary option in Packet Logger Mode

The figure 10 shows the result obtained by the GRANT tool running in Packet Logger Mode with binary option. Here, the result of the LIDS is displayed in binary format. The command that is used to log the packet in the binary format is

/grant -vde -l ./log –b

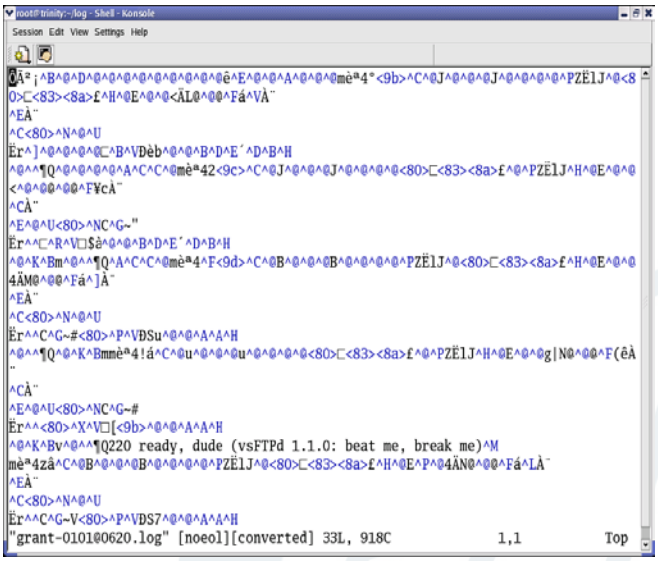

Figure 10. GRANT with Binary option

Here, the binary mode logs the packets in "tcpdump format" to a single binary file in the logging directory (log). Once the packets have been logged to the binary file, you can read the packets back out of the file with any sniffer that supports the tcpdump binary format like tcpdump.

E. *Result Statistics*

After the execution of the GRANT tool, the result statistics with the number of alerts that are created will be displayed as in Figure 11.

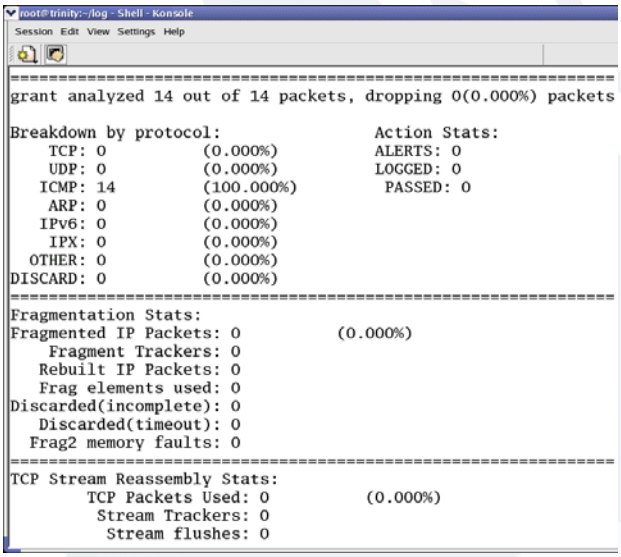

Figure 11. Result Statistics

This figure shows that 14 packets (ICMP) are analyzed by the GRANT tool. Since, there is no intrusion or misuse, no alert is generated. The packets are also not logged.

V. CONCLUSION AND FUTURE WORK

Like any other operating system, application level security flaws leave Linux vulnerable to a variety of malicious attacks. The Linux kernel provides for a multi-user operating system. This means that any user can run any program at any time without worrying about the other users. Due to the multi-user nature of the system, the kernel is also responsible for providing mechanisms for user authentication and access control to prevent users or applications from interfering with the activity of other users or programs. The sniffer, packet logger and NIDS mode of the GRANT tool detect the attacks and intrusions and take necessary actions thus protect the network with little or no damage. Thus it can be termed that Grant is a comprehensive and intelligent tool for protecting any network from external attacks and probes. There are also a number of unsolved issues concerning the analysis of the audit trail.

- Complex responses and recovery actions may be provided.
- New rule options may be set to counter the ever growing hacking and cracking.
- An automatic rule update based on the type of attack can be implemented.
- Artificial Intelligence can be implemented into the tool so that the LIDS can learn from a previous attack.
- The IDS can be made to integrate with a local firewall to be a more optimal tool.
- Data Mining techniques can be used to mine the interesting patterns from the log files.

VI. REFERENCES

[1] Rebecca Bace and Peter Mell, “Intrusion Detection System”, NIST special publication on Intrusion Detection System, 2001. doi:10.1201/1079/43273.29.5.20011101/31414.1

[2] Jeff Reinhard, “Network Intrusion Detection System”, PenTele Data, Palmerton

[3] Grant Users Manual, GrantRelease: 1.8.1, Hexa Bytes Pvt.Ltd.

[4] Harley Kozushko, “Intrusion Detection: Host-Based and Network Based Intrusion Detection Systems”, Independent Study- 2003.

[5] Giovanni Vigna and Christopher Kruegel, “Host Based Intrusion Detection System”, WL041/Bidgoli WL041-Bidgoli.cls June 15, 2005

[6] Vipin Das et al, “Network Intrusion Detection System Based On Machine Learning Algorithms”, International Journal of Computer Science & Information Technology (IJCSIT), Vol 2, No 6, December 2010. doi:10.5121/ijcsit.2010.2613

[7] Rafeeq Ur Rehman, “Intrusion Detection Systems with Snort, Library of Congress Cataloging-in-Publication Data, ISBN 0-13-140733-3

[8] Tejinder Aulakh, “Intrusion Detection and Prevention System: CGI Attacks”, The Faculty of the Department of Computer Science, San Jose State University, 2009.

[9] Sebastian Elbaum and John C. Munson, “Intrusion Detection through Dynamic Software Measurement”, Proceedings of the Workshop on Intrusion Detection

and Network Monitoring, Santa Clara, California, USA, April 9–12, 1999

[10] Taylor Merry, “Linux Kernel Hardening”, SANS Institute- 2003.

[11] J.R. Winkler, ‘‘A Unix Prototype for Intrusion and Anomaly Detection in Secure Networks,’’ *Proc. 13th National Computer Security Conference*, pp. 115-124, Washington, D.C., Oct. 1990.

[12] H-J Park and C. Sung-Bae, "Privilege Flows Modeling for Effective Intrusion Detection based on HMM," Proceedings CDWS2 in PRICAI, August, 2002.

[13] John McHugh et al, “The Role of Intrusion Detection Systems”, IEEE Software September/October 2000. doi:10.1109/52.877859

[14] L. Deri, R. Carbone, and S. Suin, Monitoring Networks Using Ntop, Proc. of IM 2001, Seattle, May 2001. doi:10.1109/INM.2001.918032

[15] Dong Yu, Deborah Frincke, “Towards Survivable Intrusion Detection System”, Proceedings of the 37th Hawaii International Conference on System Sciences – 2004. doi:10.1109/HICSS.2004.1265702

How to cite

Arul Anitha, "Network Security using Linux Intrusion Detection System". *International Journal of Research in Computer Science*, 2 (1): pp. 33-38, December 2011. doi:10.7815/ijorcs.21.2011.012