International Journal of Advances in Engineering & Technology, July, 2014.
©IJAET ISSN: 22311963

INTRUSION DETECTION SYSTEM USING DYNAMIC AGENT SELECTION AND CONFIGURATION

Manish Kumar[1], M. Hanumanthappa[2]
[1]Assistant Professor, Dept. of Master of Computer Applications,
M. S. Ramaiah Institute of Technology, Bangalore
and Research Scholar, Department of Computers Science and Applications,
Bangalore University, Bangalore, India
[2]Dept. of Computer Science and Applications, Jnana Bharathi Campus, Bangalore University,
Bangalore -560 056, India

ABSTRACT
Intrusion detection is the process of monitoring the events occurring in a computer system or network and analysing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices. An intrusion detection system (IDS) monitors network traffic and monitors for suspicious activity and alerts the system or network administrator. It identifies unauthorized use, misuse, and abuse of computer systems by both system insiders and external penetrators. Intrusion detection systems (IDS) are essential components in a secure network environment, allowing for early detection of malicious activities and attacks. By employing information provided by IDS, it is possible to apply appropriate countermeasures and mitigate attacks that would otherwise seriously undermine network security. However, Increasing traffic and the necessity of stateful analysis impose strong computational requirements on network intrusion detection systems (NIDS), and motivate the need of architectures with multiple dynamic sensors. In a context of high traffic with heavy tailed characteristics, static rules for dispatching traffic slices among sensors cause severe imbalance. The current high volumes of network traffic overwhelm most IDS techniques requiring new approaches that are able to handle huge volume of log and packet analysis while still maintaining high throughput. This paper shows that the use of dynamic agents has practical advantages for intrusion detection. Our approach features unsupervised adjustment of its configuration and dynamic adaptation to the changing environment, which improvises the performance of IDS significantly.

KEYWORDS—*Intrusion Detection System, Agent Based IDS, Dynamic Sensor Selection.*

I. INTRODUCTION

Intrusion Detection is the process of monitoring and analysing the information sources, in order to detect malicious information. It has been an active field of research for over two decades. John Anderson's "Computer Security Threat Monitoring and Surveillance" was published in 1980 and has embarked upon this field. It was one of the earliest and most famous papers in the field. After that in 1987, Dorothy Denning published "An Intrusion Detection Model", provided a methodological framework that inspired many researchers around the world and has laid the groundwork for the early commercial products like Real Secure, Trip Wire, Snort, Shadow, and STAT etc.

Intrusion Detection technology has evolved and emerged as one of the most important security solutions. It has several advantages and it is unique compared to other security tools. As information systems have become more comprehensive and a higher value asset of organizations, intrusion detection systems have been incorporated as elements of operating systems and network.

Intrusion detection systems (IDS) have a few basic objectives. Among these objectives are Confidentiality, Integrity, Availability, and Accountability.

Intrusion Detection Systems (IDS) are important mechanisms which play a key role in network security and self-defending networks. Such systems perform automatic detection of intrusion attempts and malicious activities in a network through the analysis of traffic captures and collected data in general. Such data is aggregated, analysed and compared to a set of rules in order to identify attack signatures, which are traffic patterns present in captured traffic or security logs that are generated by specific types of attacks. In the process of identifying attacks and malicious activities an IDS parses large quantities of data searching for patterns which match the rules stored in its signature database. Such procedure demands high processing power and data storage access velocities in order to be executed efficiently in large networks. The next part of the paper discuss about classification of Intrusion Detection Systems. The section II of the paper discuss about dynamic sensor agents for improvising the performance of IDS. Section III discuss about the algorithm for using the dynamic agent for improvising the performance of IDS. Section IV analyse and show the improvement in performance of IDS implementation using agent followed by conclusion and future work in section V.

1.1 Classification of Intrusion Detection Systems

Intrusions can be divided into 6 main types:-

1. Attempted break-ins, which are detected by a typical behaviour profiles or violations of security constraints.
2. Masquerade attacks, which are detected by atypical behaviour profiles or violations of security constraints.
3. Penetration of the security control system, which are detected by monitoring for specific patterns of activity.
4. Leakage, which is detected by atypical use of system resources.
5. Denial of service, which is detected by atypical use of system resources.
6. Malicious use, which is detected by a typical behaviour profiles, violations of security constraints, or use of special privileges.

However, we can divide the techniques of intrusion detection into two main types. IDSs issue security alerts when an intrusion or suspect activity is detected through the analysis of different aspects of collected data (e.g. packet capture files and system logs). Classical intrusion detection systems are based on a set of attack signatures and filtering rules which model the network activity generated by known attacks and intrusion attempts [8]. Intrusion detection systems detect malicious activities through basically two approaches: anomaly detection and signature detection [9][21][20].

i. Anomaly Detection

This technique is based on the detection of traffic anomalies. The deviation of the monitored traffic from the normal profile is measured. Various different implementations of this technique have been proposed, based on the metrics used for measuring traffic profile deviation.

Anomaly detection techniques assume that all intrusive activities are necessarily anomalous. This means that if we could establish a "normal activity profile" for a system, we could, in theory, flag all system states varying from the established profile by statistically significant amounts as intrusion attempts. However, if we consider that the set of intrusive activities only intersects the set of anomalous activities instead of being exactly the same, we find a couple of interesting possibilities: (1) Anomalous activities that are not intrusive are flagged as intrusive. (2) Intrusive activities that are not anomalous result in false negatives (events are not flagged intrusive, though they actually are). This is a dangerous problem, and is far more serious than the problem of false positives.

The main issues in anomaly detection systems thus become the selection of threshold levels so that neither of the above 2 problems is unreasonably magnified, and the selection of features to monitor. Anomaly detection systems are also computationally expensive because of the overhead of keeping track of, and possibly updating several system profile metrics. Some systems based on this technique are discussed in Section 4 while a block diagram of a typical anomaly detection system is shown in Fig 1.

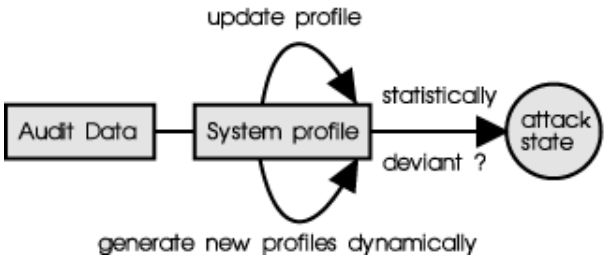

Fig 1:- IDS Anomaly Detection System

ii. Misuse Detection

This technique looks for patterns and signatures of already known attacks in the network traffic. A constantly updated database is usually used to store the signatures of known attacks. The way this technique deals with intrusion detection resembles the way that anti-virus software operates.

The concept behind misuse detection schemes is that there are ways to represent attacks in the form of a pattern or a signature so that even variations of the same attack can be detected. This means that these systems are not unlike virus detection systems -- they can detect many or all known attack patterns, but they are of little use for as yet unknown attack methods. An interesting point to note is that anomaly detection systems try to detect the complement of "bad" behaviour. Misuse detection systems try to recognize known "bad" behaviour. The main issues in misuse detection systems are how to write a signature that encompasses all possible variations of the pertinent attack, and how to write signatures that do not also match non-intrusive activity. Several methods of misuse detection, including a new pattern matching model are discussed later. A block diagram of a typical misuse detection system is shown in Fig 2 below.

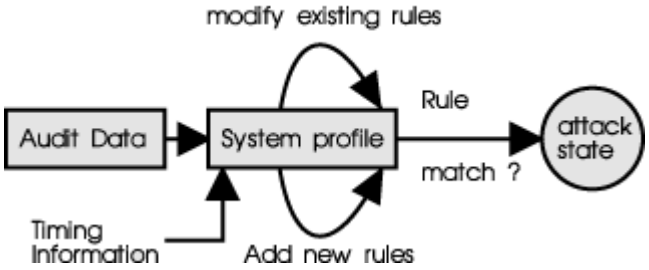

Fig 2:-IDS Misuse Detection System

Intrusion detection systems are further can also be classified in two groups, Network Intrusion Detection Systems (NIDS), which are based on data collected directly from the network, and Host Intrusion Detection Systems (HIDS), which are based on data collected from individual hosts. HIDSs are composed basically by software agents which analyse application and operating system logs, file system activities, local databases and other local data sources, reliably identifying local intrusion attempts. Such systems are not affected by switched network environments (which segment traffic flows) and is effective in environments where network packets are encrypted (thwarting usual traffic analysis techniques). However, they demand high processing power overloading the nodes' resources and may be affected by denial-of-service attacks. In face of the growing volume of network traffic and high transmission rates, software based NIDSs present performance issues, not being able to analyses all the captured packets rapidly enough. Some hardware based NIDSs offer the necessary analysis throughput but the cost of such systems is too high in relation to software based alternatives.

From the above, it is clear that as IDS grow in function and evolve in power, they also evolve in complexity. Agents of each new generation of IDS use agents of the previous generation as data sources, applying ever more sophisticated detection algorithms to determine ever more targeted responses. Often, one or more IDS and management system(s) may be deployed by an organization within its own network, with little regard to their neighbours or the global Internet. Just as all individual networks and intranets connect to form "The Internet", so can information from stand-alone

internal and perimeter host- and network-based intrusion detection systems be combined to create a distributed Intrusion Detection System (dIDS).

Current IDS technology is increasingly unable to protect the global information infrastructure due to several problems:

i. The existence of single intruder attacks that cannot be detected based on the observations of only a single site.
ii. Coordinated attacks involving multiple attackers that require global scope for assessment.
iii. Normal variations in system behaviour and changes in attack behaviour that cause false detection and identification.
iv. detection of attack intention and trending is needed for prevention
v. Advances in automated and autonomous attacks, i.e. rapidly spreading worms, require rapid assessment and mitigation, and
vi. The sheer volume of attack notifications received by ISPs and host owners can become overwhelming.
vii. If aggregated attack details are provided to the responsible party, the likelihood of a positive response increases.

II. DYNAMIC SENSOR SELECTION

In our proposed architecture, IDS LOG can be collected from multiple sensors or agent. In this section, we present a trust-based algorithm which dynamically determines the best aggregation agent and also the optimal number of malicious or legitimate behaviour necessary for the reliable identification of the best aggregation agent, while taking into account the: (i) past effectiveness of the individual aggregation agents and (ii) number of aggregation agents and the perceived differences in their effectiveness. We decided to use a trust-based approach for evaluating the aggregation agents, because it not only eliminates the noise in the background traffic and randomness of the challenge selection process, but accounts for the fact that attackers might try to manipulate the system by inserting misleading traffic flows. An attacker could insert fabricated flows [15] hoping they would cause the system to select an aggregation agent that is less sensitive to the threat the attacker actually intends to realize. When using trust, one tries to avoid this manipulation by dynamically adapting to more recent actions of an attacker [4][16].

The problem features a set of classifier agents $A = \{\alpha,...,\alpha_g\}$ that process a single, shared open-ended sequence $\Phi = \langle \varphi_1,...,\varphi_i... \rangle$ of incoming events and use their internal models to divide these events into two categories: normal and anomalous. The events are inherently of two fundamental types: legitimate and malicious, and the goal of the classifier agents is to ensure that the normal class as provided by the agent is the best possible match to the legitimate traffic class, while the anomalous class should match the malicious class. The classification thus has four possible outcomes [17] for each event ϕ, two of them being correct classifications and two of them the errors (see also the confusion matrix in Table 1).

Table 1: Confusion Matrix

		actual class	
		legitimate	malicious
classification	normal	true positive	false positive
	anomalous	false negative	true negative

The classifier agents actually provide more information, as they internally annotate the individual events φwith a continuous "normality" value in the [0, 1] interval, with the value1 corresponding to perfectly normal events and the value 0 to completely anomalous ones. This continuous anomaly value describes an agent's opinion regarding the anomaly of the event, and the agents apply adaptive or predefined thresholds to split the [0, 1] interval into the normal and anomalous classes.

Given that the characteristics of the individual classifier agents αk are unknown in the dynamically-changing environment, the system needs to be able to identify the optimal classifier autonomously. Furthermore, the system can have several users with different priorities regarding the detection of specific types of malicious events. In the network monitoring use-case, some of the users concentrate

on major, infrastructure-type events only (such as denial of service attacks),while the other users seek information about more subtle attack techniques targeting individual hosts. The users are represented by the user agents and these agents are assumed to know their users preferences. Their primary goal is to use their knowledge of user preferences to dynamically identify the optimal information source and to change the source when the characteristics of the environment or user preferences change. To reach this goal in an environment where they have no abstract model of classifier agents' performance, they rely on empirical analysis of classifier agents' response to a pre-classified set of challenges [11][19].
In the following, we will analyse the problem from the perspective of a single user agent, which tries to select the best available classification agent, while keeping the number of challenges as low as possible. The challenges are events with known classification, which can be inserted into the flow of background events as observed by the system, processed by the classifier agents together with the background events and finally removed before the system reports the results to the users. The processing of the challenges allows the user agents to identify the agent which achieves the best separation of the challenges that represent known instances of legitimate behaviour from the challenges that represent known malicious behaviour[13][7][18].

III. ALGORITHM

In this section we present a simple but adaptive algorithm for choosing the best classifier agent. For each time step $i \in \mathrm{N}$, the algorithm proceeds as follows:
For each time step i ∈ N, the algorithm proceeds as follows:

i. Let each aggregation agent classify a set of known instances of malicious or legitimate behaviour from different attack classes and selected legitimate known instances of malicious or legitimate behaviour.
ii. Update the trust value of each aggregation agent, based on its performance on the known instances of malicious or legitimate behaviour in time step i.
iii. Accept the output of the aggregation agent with the highest trust value as classification of the remaining events of time step i.

Known instances of malicious or legitimate behaviour detection and aggregation agents in each time step i with the sets of flows for which we already know the actual class, i.e. whether they are malicious or legitimate. So, we challenge an aggregation agent α with a set of malicious events, belonging to K attack classes and a set of legitimate events drawn from a single class. With respect to each class of attacks k, the performance of the agent is described by a mean and a standard deviation: $(\bar{x}^k, \sigma_x^k)$ for the set of malicious challenges and $(\bar{y}\ , \sigma_y)$ for the set of legitimate challenges. Both means lie in the interval [0, 1], and $\bar{x}^k$ close to 0 and $\bar{y}$ close to 1 signify accurate classifications of the agent respectively.

The system used to perform the experiments described in this paper incorporates five different anomaly detection [5] techniques presented in literature [10][1] [12][2]. Each of the methods works with a different traffic model based on a specific combination of aggregate traffic features, such as:

- Entropies of flow characteristics for individual source IP addresses.
- Deviation of flow entropies from the PCA-based prediction model of individual sources.
- Deviation of traffic volumes from the PCA-based prediction for individual major sources.
- Rapid surges in the number of flows with given characteristics from the individual sources and
- Ratios between the number of destination addresses and port numbers for individual sources.

These algorithms maintain a model of expected traffic on the network and compare it with real traffic to identify the discrepancies that are identified as possible attacks. They are effective against zero-day attacks and previously unknown threats, but suffer from a comparatively higher error rate [17][10][11], frequently classifying legitimate traffic as anomalous(false positives), or failing to spot malicious flows (false negatives). The classifier agents can be divided to two distinct classes:

- **Detection agents** analyse raw network flows by their anomaly detection algorithms, exchange the anomalies between them and use the aggregated anomalies to build and

update the long-term anomaly associated with the abstract traffic classes built by each agent. Each detection agent uses one of the five anomaly detection techniques mentioned above. All agents map the same events (flows), together with the same evaluation of these events, the aggregated immediate anomaly of these events determined by their anomaly detection algorithms, into the traffic clusters built using different features/metrics, thus building the aggregate anomaly hypothesis based on different premises. The aggregated anomalies associated with the individual traffic classes are built and maintained using the classic trust modelling techniques (not to be confused with the way trust is used in this work).

- **Aggregation agents** represent the various aggregation operators used to build the joint conclusion regarding the normality/anomaly of the flows from the individual opinions provided by the detection agents. Each agent uses a distinct averaging operator (based on order-weighted averaging or simple weighted averaging) to perform the Rgdet→ R transformation from the gdet-dimensional space to a single real value, thus defining one composite system output that integrates the results of several detection agents. The aggregation agents also dynamically determine the threshold values used to transform the continuous aggregated anomaly value in the [0, 1] interval into the crisp normal/ anomalous assessment for each flow.

The user agent functionality is implemented as a collection of the agents. The user agent creates individual challenge agents, each of them representing a specific incident in the past, and these temporary, single purpose agents interact with the data-provisioning layers of the system in order to insert the flows relative to the incident into the background traffic and to retrieve and analyse the detection results provided by the classifier agents.

IV. RESULTS AND PERFORMANCE ANALYSIS OF AGENT BASED IDS

We have simulated and tested the IDS using the KDD Cup 1999 dataset. The implementation gives us the expected results. Our Agent based IDS prototype we are testing detects the simulated attacks. The question is : why the realization of the system with agents is advantageous? We implement a centralized system with local sensor that forward filtered data to a central analysis node and compare it with Agent IDS.

Agent based IDS has proven itself to be capable of handling very high traffic. In such a design, the incoming network traffic is disseminated to a pool of agents, which process a fraction of the whole traffic, reducing the possibility of packet loss caused by overload. Agent IDS could support a load of up to 56 Mbps (450 packets/second) with zero traffic loss. Moreover, we focus on a second important criterion for IDS: detection delay which is defined as the duration from the time the attack starts to the time epoch that the attack is detected. We generate a set of packets varied from 1000 to 8000. For each set we simulate the attack and we calculate the detection delay. Figure3 plots the measurement results. The detection delay is significantly reduced; Agent IDS is much faster than the centralized IDS. For example, in the case of 8000 packets, we observe that detection delay is reduced by 56% (7.91second vs 4.4 second). This can be explained by the fact that agents operate directly on the host, where an action has to be taken, their response is faster than systems where the actions are taken by central coordinator.

In fact, one of the most pressing problems facing current IDSs is the processing of the enormous amounts of data generated by the network traffic monitoring tools and host-based audit logs. IDSs typically process most of this data locally. Agents offer an opportunity to reduce the network load by eliminating the need for this data transfer. Instead of transferring the data across the network, agents can be dispatched to the machine on which the data resides, essentially moving the computation to the data, instead of moving the data to the computation. It is obvious to see that the code-shipping versus data-shipping argument is only valid if, the agent's code and state that have to be transmitted are not larger than the amount of data that can be saved by the use of an agent. Agent IDS does not only perform better in terms of effectiveness but also in terms of detection delay.

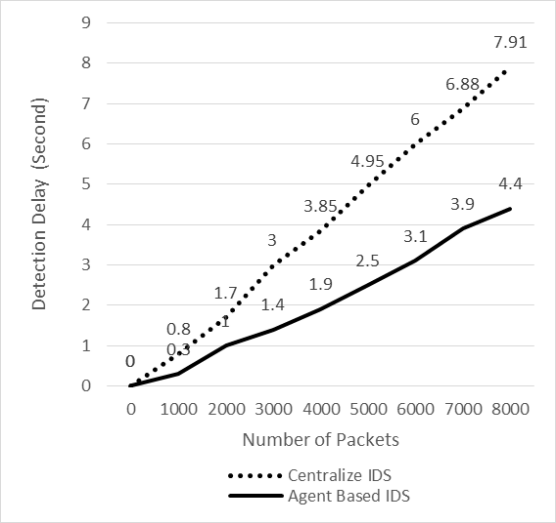

Fig 3:- Performance of Centralize IDS vs Agent Based IDS

V. CONCLUSION AND FUTURE WORK

We take advantage of the multi-agent paradigm especially concerning reducing Network Load. Indeed, agents offer the possibility to eliminate the need of transferring a huge amount of data to be analysed. In this paper we have explained the architectural design and performance analysis of a Centralize IDS vs. Agent based IDS. The experimental result was positive and we found that this work can be continued with several other improvement and performance analysis As network attacks are becoming more and more alarming, exploiting systems faults and performing malicious actions the need to provide effective intrusion detection methods increases. Network-based, distributed attacks are especially difficult to detect and require coordination among different intrusion detection components or systems. The experiments emphasize the aim of applying agent to detect some kind of intrusions and compete others IDS.

ACKNOWLEDGMENT

I would like to thank MSRIT Management, my colleagues and Dept. of Computer Science and Applications, Bangalore University, for their valuable suggestion, constant support and encouragement.

REFERENCES

[1] A. Lakhina, M. Crovella, and C. Diot. Mining Anomalies using Traffic Feature Distributions. In ACM SIGCOMM, Philadelphia, PA, August 2005,pages 217–228, New York, NY, USA, 2005. ACM Press.
[2] A. Sridharan, T. Ye, and S. Bhattacharyya. Connectionless port scan detection on the backbone. Phoenix, AZ, USA, 2006.
[3] Axelsson, Stefan, “Intrusion Detection Systems: A Taxonomy and Survey”, Technical Report No 99-15, Dept. of Computer Engineering, Chalmers University of Technology, Sweden, March 2000.
[4] Chang-Lung Tsai; Chang, A.Y.; Chun-Jung Chen; Wen-Jieh Yu; Ling-Hong Chen, "Dynamic intrusion detection system based on feature extraction and multidimensional hidden Markov model analysis," Security Technology, 2009. 43rd Annual 2009 International Carnahan Conference on , vol., no., pp.85,88, 5-8 Oct. 2009.
[5] D. E. Denning. An intrusion-detection model. IEEE Trans. Softw. Eng., 13(2):222–232, 1987.
[6] F. A. Barika & N. El Kadhi & K. Gh´edira, “Agent IDS based on Misuse Approach”, Journal of Software, Vol. 4, No. 6, 495-507, August 2009.
[7] Guangcheng Huo; Xiaodong Wang, "DIDS: A dynamic model of intrusion detection system in wireless sensor networks," Information and Automation, 2008. ICIA 2008. International Conference on , vol., no., pp.374,378, 20-23 June 2008.

[8] H.Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of Intrusion Detection Systems", The International Journal of Computer and Telecommunications Networking - Special issue on computer network security, Volume 31 Issue 9, Pages 805 – 822, April 23, 1999,
[9] Jun Wu; Chong-Jun Wang; Jun Wang; Shi-Fu Chen, "Dynamic Hierarchical Distributed Intrusion Detection System Based on Multi-Agent System," Web Intelligence and Intelligent Agent Technology Workshops, 2006. WI-IAT 2006 Workshops. 2006 IEEE/WIC/ACM International Conference on , vol., no., pp.89,93, Dec. 2006
[10] K. Xu, Z.-L. Zhang, and S. Bhattacharrya. Reducing Unwanted Traffic in a Backbone Network. In USENIX Workshop on Steps to Reduce Unwanted Traffic in the Internet (SRUTI), Boston, MA, July 2005.
[11] Kumar, G.V.P.; Reddy, D.K., "An Agent Based Intrusion Detection System for Wireless Network with Artificial Immune System (AIS) and Negative Clone Selection," Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conference on , vol., no., pp.429,433, 9-11 Jan. 2014.
[12] L. Ertoz, E. Eilertson, A. Lazarevic, P.-N. Tan,V. Kumar, J. Srivastava, and P. Dokas. Minds - minnesota intrusion detection system. In NextGeneration Data Mining. MIT Press, 2004.
[13] Lin Zhao-wen; Ren Xing-tian; Ma Yan, "Agent-based Distributed Cooperative Intrusion Detection System," Communications and Networking in China, 2007. CHINACOM '07. Second International Conference on , vol., no., pp.17,22, 22-24 Aug. 2007.
[14] Martin Rehak, Eugen Staab, Michal Pechoucek, Jan Stiborek, Martin Grill, and Karel Bartos. 2009. Dynamic information source selection for intrusion detection systems. In Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2(AAMAS '09), Vol. 2. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1009-1016.
[15] Martin Rehak, Eugen Staab, Volker Fusenig, Michal Pechoucek, Martin Grill, Jan Stiborek, Karel Bartos, and Thomas Engel, "Runtime Monitoring and Dynamic Reconfiguration for Intrusion Detection Systems", Proceedings of 12th International Symposium, RAID 2009, Saint-Malo, France, September 23-25, pp 61-80, 2009.
[16] Paez, R.; Torres, M., "Laocoonte: An agent based Intrusion Detection System," Collaborative Technologies and Systems, 2009. CTS '09. International Symposium on , vol., no., pp.217,224, 18-22 May 2009.
[17] S. Northcutt and J. Novak. Network Intrusion Detection: An Analyst's Handbook. New Riders Publishing, Thousand Oaks, CA, USA, 2002.
[18] Sun-il Kim; Nwanze, N.; Kintner, J., "Towards dynamic self-tuning for intrusion detection systems," Performance Computing and Communications Conference (IPCCC), 2010 IEEE 29th International, vol., no., pp.17,24, 9-11 Dec. 2010.
[19] Weijian Huang; Yan An; Wei Du, "A Multi-Agent-Based Distributed Intrusion Detection System," Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on , vol.3, no., pp.V3-141,V3-143, 20-22 Aug. 2010.
[20] Yinan Li; Zhihong Qian, "Mobile Agents-Based Intrusion Detection System for Mobile Ad Hoc Networks," Innovative Computing & Communication, 2010 Intl Conf on and Information Technology & Ocean Engineering, 2010 Asia-Pacific Conf on (CICC-ITOE) , vol., no., pp.145,148, 30-31 Jan. 2010.
[21] Yu Cai, Hetal Jasani, "Autonomous Agents based Dynamic Distributed (A2D2) Intrusion Detection System", Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications 2007, pp 527-533

AUTHORS

Manish Kumar is working as Asst. Professor in Department of Computer Applications, M. S. Ramaiah Institute of Technology, Bangalore, India. His areas of interest are Cryptography and Network Security, Computer Forensic, Mobile Computing and eGovernance. His specialization is in Network and Information Security. He has also worked on the R&D projects relates on theoretical and practical issues about a conceptual framework for E-Mail, Web site and Cell Phone tracking, which could assist in curbing misuse of Information Technology and Cyber Crime. He has published several papers in International and National Conferences and Journals. He has delivered expert lecture in various academic Institutions.

M Hanumanthappa is currently working as Associate Professor in the Department of Computer Science and Applications, Bangalore University, Bangalore, India. He has over 17 years of teaching (Post Graduate) as well as Industry experience. He is member of Board of Studies /Board of Examiners for various Universities in Karnataka, India. He is actively involved in the funded research projects and guiding research scholars in the field of Data Mining and Network Security.