# COOPERATIVE INTRUSION DETECTION SYSTEM (CIDS) IN GRID ENVIRONMENT ON UNLABELLED DATA

Abdul Samad Haji Ismail[1], Dahliyusmanto[2], Witcha Chimphlee[3], Abdul Hanan Abdullah[4], Kamalrulnizam Abu Bakar[5], Md Asri Ngadi[6]

[1,4,5,6]Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia, Tel: (607)-5532003, Fax: (607)-5565044

[2]Department of Electrical Engineering, Faculty of Engineering, University of Riau. Kampus Bina Widya Km 12.5 Simpang Baru 28293, Riau, Indonesia, , Tel: (62)-76163266, Fax: (62)-76163279

[3]Faculty of Science and Technology, Suan Dusit Rajabhat University, 295 Rajsima Road, Dusit, Bangkok, Thailand, Tel: (60)-2445224, Fax: (60)-6687136

[1]abdsamad@utm.my, [2] yoes_mantho@sigma-snt.com, [3]witcha_chi@dusit.ac.th, [4]hanan@utm.my,, [5]knizam@utm.my, [6]dr.asri@utm.my

**KEYWORDS**

Anomaly detection, unsupervised clustering, unlabeled data, Fuzzy Clustering, Grid Environment.

**ABSTRACT**

Intrusions pose a serious security risk in a network environment. The intrusion detection in computer networks is a complex research problem. Applying intrusion detection to the fast growing computational Grid environments improves the security and is considered to be the heart of this new field. Flexible cooperative distributed intrusion detection architecture is introduced that suits to and benefits from the underlying Grid environment. Intrusion detection techniques fall into two general categories: anomaly detection and signature recognition, with each one complements one other. Anomaly intrusion detection normally has high false alarm rates, and a high volume of false alarms will prevent system administrators from identifying the real attacks. This paper presents a clustering-based anomaly intrusion detection algorithm which trains on unlabeled data in order to detect new intrusions. This work does not make a strict hypothetical requirement with the percentage of attacks has to be less than a certain threshold (e.g.,~1.5%). It also does not label clusters by considering the sparse density is attacks. We propose a new labelling cluster algorithms, called NMF (Normal Membership Factor) that is capable of increasing normal detection which would be indicative of decrease false positive rate. Our method is able to detect many different types of intrusions, while maintaining a low false positive rate as verified over the Knowledge Discovery and Data Mining-KDD CUP 1999 dataset.

## 1. INTRODUCTION

Research and development efforts within the Grid community have produced protocols, services, and tools that address the challenges arising when seeking to build scalable virtual organization (VOs). A virtual organization is defined as a set of individuals and/or institutions sharing resources and service under a set of rules and policies governing the extent and conditions for that sharing. As stated in [1], "the sharing that Grid environments are concerned with is not primarily file exchange but rather direct access to computers, software, data and other resources, as is required by a range of collaborative problem-solving and resource-brokering strategies emerging in industry, science, and engineering

This sharing is, necessarily, high controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. The technologies that have evolved from the Grid community include security solutions that support management of credentials and policies when computations span multiple institutions; resource management protocols and services that support secure remote access to computing and data resources and the co-allocation of multiple resources. The recent history of attacks against the information system and widespread vulnerabilities indicate that security threats have dramatically escalated in speed, impact and frequency. Grid Infrastructures like Globus [2], Legion [3] and Condor [4] are particularly vulnerable to intrusions and require and adequate level of security for users, data and resources. Grid are open environments, compromise of single resource may provide unauthorized access to data and services from other

Proceedings of the 2008 High Performance Computing & Simulation Conference ©ECMS
Waleed W. Smari (Ed.)
ISBN: 978-0-9553018-7-2 / ISBN: 978-0-9553018-6-5 (CD)

system. Grids are vulnerable to large-scale attacks that may cause disruption of the Grid services. Thus, it is essential for Grids to support prevention, detection and automatic response to intrusion attempts and work cooperatively.

## 2. RELATED WORK AND BACKGROUND

### 2.1 Grid Computing

Grid was proposed in the mid 1990 and has been widely used in many areas, such as bioinformatics, medicine, astronomy, chemistry, agriculture, business and engineering design, to solve large-scale and complex problems [5]. Today, many organizations such as Compaq, Sun Microsystems, Fujitsu, Hitachi and NEC fall into Grid research. They have adopted Globus Toolkit, developed by USC'S Information Science Institute (ISI) and Argonne National Laboratory, as their basic platform. Globus toolkit is an open-architecture, consisting of security, information infrastructure, resource management, data management and communication components. It facilitates creation of usable Grids, enabling high-speed coupling of people, computers, databases, and instruments [6].

In Grid environment, the security challenges face under three categories [7]: integration with existing security architectures and model across platforms and hosting environment, interoperability with different hosting environments (e.g., J2EE servers, .NET servers, Linux systems) at multiple level such protocol level, policy level, and identity level, and trust relationships among interacting hosting environments.

### 2.2 Intrusion Detection System (IDS)

Most traditional intrusion detection systems (IDSs) [8] take either a network-based or a host-based approach for recognizing and responding to attacks. These systems look either for attack signatures, specific patterns that indicate malicious or suspicious intent or deviation from a normal profile (anomaly) that indicate an attack. A network-based IDS looks for these patterns and anomalies in network traffic.

A host-based IDS looks for attack signatures and anomalies in system audit trails or application logs. In most cases intruder exploit vulnerabilities and misconfiguration in application server to break into a system. The application-level attacks can enter a system through the same open "door" in the perimeter defences used by legitimate users. Therefore, these attacks are difficult to detect. Current Network-based and Host-based IDSs work in isolation from access control for the application the systems aim to protect. The lack coordination and inter-operation between these components prevents detecting sophisticated attacks and responding to ongoing attacks in real time, before they cause damage. Another disadvantage of currently available IDSs is a large number of false positive (IDS reports an attack when none has occurred). Reports of attacks can trigger response actions (e.g., termination of the offending connections). Thus an inaccurate IDS decision (a false alarm) may result in disruption of service to legitimate users. Therefore, successful intrusion detection requires accurate and efficient models for analyzing a large amount of application, system and network audit data and real time response to the attacks.

## 3. THE PROPOSED GRID INTRUSION DETECTION ARCHITECTURE

In this section, the research proposes an architecture of Grid intrusion detection. This architecture was designed with the Grid characteristics in mind. The Grid intrusion detection architecture has two main parts as shown in Figure 1. The first is intrusion detection agent (represented by small black circle) that is responsible for gathering information. And the second part is the intrusion detection server (represented by big circle) that is responsible for analyzing the gathered information and cooperating with other IDSs to detect intrusions.

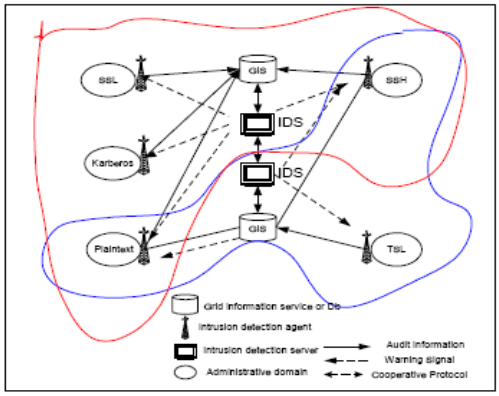
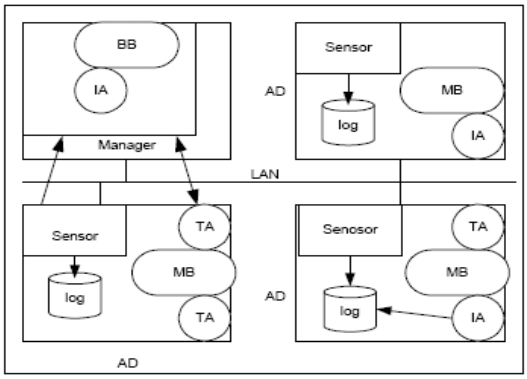



Figure 1: Proposed Grid Intrusion Detection Agent

### 3.1 Structure of Intrusion Detection Agent

In many conventional network intrusion detection systems, each target system transfers its system log to an intrusion detection server, and the server analyzes the entire log in search of intrusions. Methods of this kind fall under the client/server paradigm. In a large-scale network deploying an intrusion detection system, network traffic will be extremely high, since the volume of the system logs that are routinely transferred is very large, though most of it has no information related to intrusions. Therefore, this type of intrusion detection system on a large-scale network does not fulfil its function efficiently. To solve this problem, we adopted a mobile-agent paradigm in detecting intrusions. Mobile-agents autonomously migrate in host in this case is the administrative domain to collect only information

related to intrusions, eliminating the need to transfer system logs to the server. We can deploy Intrusion Detection Agent on a local area network, the protocol of which is TCP/IP. Intrusion Detection Agent consists of sensors, message boards, tracing agents, and information-gathering agents. The system details are show in Figure 2.




AD : Administrative Domain
MB: Message Board
IA : Information_Gathering Agent
TA : Tracing Agent
BB : Bulletin Board

Figure 2: Structure of Intrusion Detection Agent

- **Manager**

The manager analyzes information gathered by information gathering agents and detects intrusions. It manages the mobile agents and bulletin boards and provides an interface between administrators and the system. The manager accumulates and weighs the information entered by the mobile agents on the bulletin board, and if the weights exceed a set threshold, the manager concludes that an intrusion has occurred. One manager resides on each network segment.

- **Sensor**

The sensors, present on each host, monitor system logs in search of suspicious activity. If a sensor finds that activity, it reports this finding to the manager.

- **Tracing Agent**

The intrusion-route tracing agent, called simply the tracing agent, traces the path of an intrusion and identities its point of origin; the place from which the user leaving a n activity remotely logged onto the target host. En route to finding the origin, a tracing agent can find any intermediate nodes that may be compromised. The manager, sensor, and tracing agent work together in the following way. First, the sensor detects a suspicious activity and reports it to the manager, and then the manager launches a tracing agent to the host system. The tracing agent migrates autonomously from machine to machine and traces the intrusion independently; without the manager. When many suspicious activities are found by a single host system in different sessions over a short period of time, many tracing agents corresponding to the suspicious activity are launched into the host system. A tracing agent makes no judgments about intrusions, and is not capable of deciding whether or not an intrusion has occurred. A tracing agent can migrate to any system in which Intrusion Detection Agent (IDA) is installed.

- **Information Gathering Agent**

An information-gathering agent, which is mobile, gleans information related to a suspicious activity from a target system. Each time a tracing agent in pursuit of an intruder is dispatched into a target system, it activates an information gathering agent in that system. Then the information gathering agent gleans information depending on the type of activities, returns to the manager and reports. If the tracing agent migrates to another target system, it will activate another information-gathering agent, which will gather information on the next host system. Many information gathering agents may be activated by many different tracing agents on the same target system. An information-gathering agent is not capable of deciding whether an intrusion has
occurred.

- **Bulletin Board and Message Board**

The bulletin board and the message board are common use area that can be accessed by tracing agents and information gathering agents, and means of information exchange. There is a message board on each target system, used by tracing agents for exchanging information; any tracing agent can know whether a track under its scrutiny has already been traced by other agents, and can use this information in deciding where to go. The bulletin board is on the manager machine and is used for recording information gathered from host systems by information-gathering agents, as well as for integrating the information gathered about every tracing route.

### 3.2 Action of Intrusion Detection Agent

The following is of intrusion detection agent works after a sensor detects a suspicious activity on an administrative domain. Intrusion detection agent accumulates the data required by intrusion-route tracing (i.e., about network connection, the various processes running on the system, etc.) on each target system in advance.

i. Each sensor on the administrative domain seeks a suspicious activity from the system log.
ii. If the sensor detects a suspicious activity, it reports it to the manager.
iii. The manager dispatches a tracing agent to the target system where the suspicious activity was detected.

iv. The tracing agent arrives at the administrative domain and activates an information-gathering agent.
v. The information-gathering agent collects information related to the suspicious activity on the administrative domain.
vi. After activating the information-gathering agent, the tracing agent investigates the point of origin of the suspicious activity in an effort to identify the user's remote site. The tracing agent can derive this from the accumulated data about network connection and processes running on the system.
vii. After collecting information, the information gathering agent, independent of the tracing agent, returns to the manager, and enters the information on the bulletin board.
viii. The tracing agent moves to the next administrative domain on the tracing route, and it activates a new information-gathering agent.
ix. If the tracing agent arrives at the origin of the route, or cannot move anywhere, or if other tracing agents have chased the route it could follow, it returns to the manager.

In cases where a sensor detects many suspicious activities on an administrative domain occurring over a short period of time, or if sensors detect suspicious activities on many administrative domain, intrusion detection agent works as described above for all suspicious activities detected. The SSL, Kerberos Plaintext, TLS and SSH are the administrative domains (resources) in a Grid environment. Each administrative domain will have an intrusion detection agent to collect data and the intrusion detection agent will register with one or more IDSs which will analyze the gathered data. Intrusion detection agent will be designed for each class of resources to handle heterogeneity.

### 3.3 Lab Environment and Software

This section gives an overview of the configuration of the software and hardware used in our lab. We built a very small scenario. It is the simplest Grid environment, intended to help illustrate the concepts and components behind the Grid and GT4.0.5. An Ethernet LAN, three Intel® Pentium IV machines, and one Laptop Intel® Centrino Duo machine were used. In Figure 3, we illustrate this environment with the host names and the functionality of each machine. The host names are T1, T2, T3 and T4. Also, an infrastructure server, called m0, was set. The machines should have a clock speed of 1 GHz, 512 MB of minimum memory, and hard drives totalling 40-120 GB.

If we have more than five machines available, we can build a bigger scenario for Proof-of-Concepts (PoC) proposals and/or demos. For that, you simply include more servers, such as T5, T6, and so on.

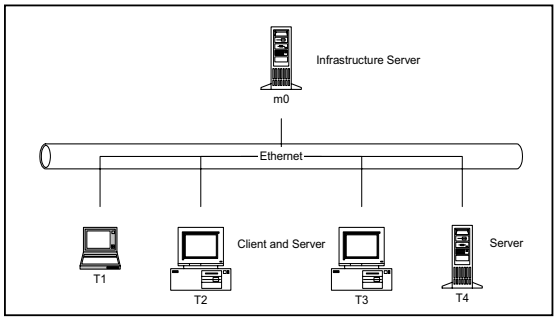



Figure 3: Hardware Environment and Software Functions of Each Machine

We give summarizes the names of the machines to be used in the Grid, their IP addresses, and the software to be installed on them as shown in Table 1.

Table 1: Host names and IP addressing

| **Hostname** | **Internet Protocol (IP)** | **Description** |
|---|---|---|
| T1.grid.research.com | 192.168.0.241 | Globus server and Client machine |
| T2.grid.research.com | 192.168.0.242 | Globus server and Client machine |
| T3.grid.research.com | 192.168.0.243 | Globus server and Client machine |
| T4.grid.research.com | 192.168.0.244 | Globus server |
| M0.grid.research.com | 192.168.0.10 | Infrastructure server |

Next, we define the users and groups that you want to use before implementation. Table 2 contains the list of user and group IDs used in our lab.

Table 2: IDs and Passwords

| User ID | Group ID | password | Activities |
|---|---|---|---|
| root | root | <passwd> | Super user needs |
| globus | globus | <passwd> | Globus Toolkit environment. For installation and execution of the toolkit |
| logicacmg | logicacmg | <passwd> | End user environment. For job execution on the Grid. |

GT4.0.5 needs several files, or tools, in order to complete the installation such as Java SDK, Apache Ant, Junit, Postgresql and Globus Toolkit. In this installation, we used Globus Toolkit bundle Version 4.0.5.

### 3.4 Implementation of Grid Intrusion Detection Architecture

This research uses two stages to test the proposed Grid intrusion detection architecture. The first stage simulates the intrusion detection agent and the Grid environment. Most of the available Grid simulation toolkits are designed for resource management and scheduling problems. For this reason this research uses the grid simulation toolkit based on GridSim [9] to satisfy the needs. The simulation environment simulates users with different behaviours, resources with associated intrusion detection agents, and intrusion detection agent registration with IDSs.

This allows us to perform the required experiments. Each experiment generate a dataset consisting of one or more log file. Figure 3 shows the simulation environment with dummy IDSs that only generate log files reflecting the data to be analyzed. The next stage implements the IDS modules and test them with the data generated from the simulation stage (Figure 4). In this initial implementation we use homogeneous IDSs for simplification. We believe that currently the best intrusion detection technique to use in this case is host-based anomaly intrusion detection [10].

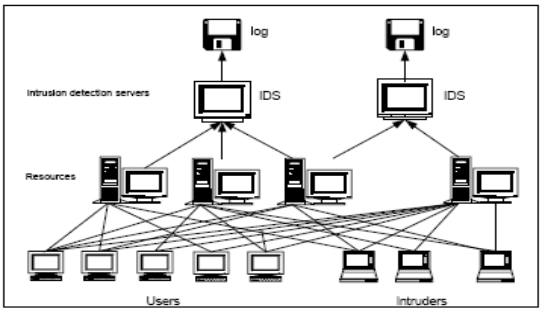



Figure 4: A Simulated Grid and Data Modules

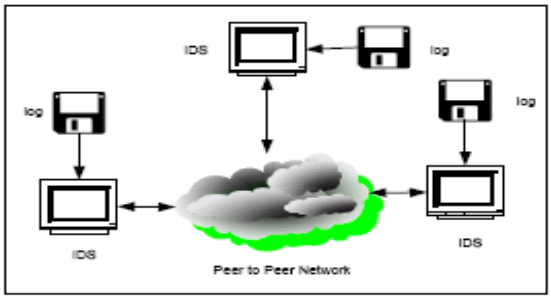



Figure 5. The implementation of IDS

## 4. NORMAL MEMBERSHIP FACTOR

The host in this case is the administrative domain with all its resources. The assigned intrusion detection agents will gather information about the user's interactions with this domain. The anomaly detection is implemented using *Normal Membership Factor* (NMF). The NMF is labelling clusters technique that identify number of instances in term of normal and attack. It is important to note that when labelling the clusters its relation to each of the clusters is taken into consideration. The results of labelling based on these factors will therefore include a degree of probability of the clusters belonging to normal group. As in Portnoy et al [11], they were determined the algorithm for cluster labelling follow their first assumption. Their first assumption about data is that normal instances constitute an overwhelmingly large portion (> 98%) of the training dataset. Under this assumption it is highly probable that cluster containing normal data will have a much larger number of instances associated with them then would clusters containing anomalies. Therefore, they labelled some percentage N of the clusters containing the largest number of instances associated with them as normal and the rest of the clusters are labelled as anomalous and are considered to contain attack.

In this paper, the portion of normal instances is not constituted to large or more than 98%. Only if the normal instance portion should more than 80 percent like Pornoy work. Then, it is not probable to identify only the group of the largest instances is normal type. Therefore, this work uses a new labelling algorithm to identify the other clusters that may be having normal pattern and then gather into normal group to reduce the false positive rate as well. For labelling the normal cluster, the two factors as described above are taken into account.

In this approach, to calculate the weight of clusters and its NMF of cluster are calculated as follows:
Weight of clusters (ci): the weight or size of the cluster is considered greatly reduces the affect of anomalies such as outliers. By multiplying the inverse distance to the cluster centres by the weight of the cluster, and dividing by the summarized weight of all the cluster centres as Equation (1).

$$WC(c_i) = \frac{1}{d(Normal, c_i)} \times \frac{number\ of\ instance\ in\ c_i}{number\ of\ all\ instance} \quad (1)$$

where $d(Normal, c_i)$ is the distance between the normal cluster and the other clusters, and i is the number of clusters. Normal Membership Factor (ci): In order to get the probability of clusters belonging to the normal cluster, the Normal Membership Factor is calculated as Equation (2).

$$NMF(c_i) = \frac{WC(c_i)}{\sum_{i=1}^{c} c_i} \quad (2)$$

where $\sum_{i=1}^{c} c_i$ is the summarized of all the weight of cluster. If NMF values have greater than 40 percents then gather that clusters into normal group.

## 5. EXPERIMENTAL SETUP AND RESULT

There are numerous methods that discuss the evaluations of intrusion detection systems. Some methods emphasise the important of detection rate (DR) and false positive rates (FPR); while other look into the novel pattern detection rate. The performance of classifiers is evaluated with respect to their classification of unseen normal and intrusive patterns. The metrics embrace here are the generalisation abilities of the classifiers because they are the most important aspect of an anomaly detection scheme. Evaluation of the generalisation capability of any intrusion detection should consider the ability of the system to recognise new normal as well as intrusive behaviours.

The best performing instances of classifiers for each data set are chosen. Six major metrics are developed to quantify the performance of the classifiers in this thesis. These metrics are calculated based on the testing patterns according to following relations.

- **Normal Generalisation (NG):** the ratio of correctly classified intrusive vectors to the total number of intrusive vectors.
- **Intrusive Generalisation or Detection Rate (DR):** the ratio of correctly classified intrusive vectors to the total number of intrusive vectors.
- **Overall Generalisation (OG):** the ratio of correctly classified vectors to the total number of the vectors. It is important to mention that this metric is sensitive to the imbalanced numbers of the normal and abnormal testing patterns.
- **Discrimination Ability (DA):** the average of the normal generalisation and the intrusive generalisation. This metric is developed due to the problem of imbalanced number of the testing patterns of normal and abnormal behaviours which affect the overall generalisation metric. This metric is dependent on the *percentage* of the generalisation of both behaviours. It is not like the overall generalization metric which is dependent on the *number* of testing.
- **False Positive (FPR):** the ratio of incorrectly classified normal vectors to the total number of normal vectors.
- **False Negative (FNR):** the ratio of incorrectly classified intrusive vectors to the total number of intrusive vectors.

In choosing the best performance of algorithm is the greatest in first four metric (NG, DR, OG, DA) and lowest in two last metric (FPR, FNR). Thus, if values of first four metrics are high and two last metrics are low it means algorithm is good, on the other hand it means that algorithm is not good for detection. The results show in Table 2 with NG, DR, OG, DA, FPR and FNR in 92.29, 95.09, 92.05, 93.69, 7.71, and 4.91 respectively.

Tabel 3: The Six Metric Results

| Metric | % |
|---|---|
| NG | 92.29 |
| DR | 95.09 |
| OG | 92.05 |
| DA | 93.69 |
| FPR | 7.71 |
| FNR | 4.91 |

## 6. CONCLUSION

The effect of trust relationships between different resource owners and the use of heterogeneous IDSs will be further investigated. With Heterogeneous IDSs and trust relationships more complex algorithms will be needed for the cooperation module that will need further investigations. The application of the Grid in real problems will help in building a knowledge base of attack signatures that will enable the use of misuse intrusion detection with the Grid.

The experimental performance shows the outstanding result in all of the evaluation criteria. The results show that the high accuracy and low false positive rate. Intrusion detection model is a composition model that needs various theories and techniques. One or two models can hardly offer satisfying results. We plan to apply other theories and techniques to operate in a high accurate and low false alarm rate in intrusion detection in our future work.

## ACKNOWLEDGEMENTS

This research was supported in part by grants from MOSTI, Malaysia. Project Number (UTM0000505).

## REFERENCES


Foster. I., Kesselman. C., and Tuecke. S. 2001. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations." International Journal of Supercomputer Applications. 15(3).

Foster. I., and Kesselmen. C. 1997. "Globus: A Meta computing Infrastructure Toolkit." International Journal of Supercomputer Applications.

Lewis. M., and Grimshaw. A. 1996. "The Core Legion Object Model". In proceedings of the 5th IEEE International Symposium on High Performance Distributed Computing.

Lizkow. M., Livny. M., and Mutka. M. 1998. "Condor – a hunter of idle workstations." In proceedings of the 8th International Conference on Distributed Computing Systems.

Foster. I., and Kesselmen. C. 1999 (eds.). "The Grid: Blueprint for a New Computing Infrastructure." Morgan Kaufmann.

Nagaratnam. N., Janson. P., Dayka. J., Nadalin. A., Siebenlist. F., Welch. V., Foster. I., and Tuecke. S. 2002. "The Security Architecture for Open Grid Services." Open Grid Service Architecture Security Working Group (OGSA-SEC-WG). Global Grid Forum.

Bace. R., and Mell. P. 2001. "Intrusion Detection Systems." National Institute of Standard and Technology (NIST) Special Publication on Intrusion Detection Systems.

M. Murshed, R. Buyya, and D. Abramson. 2002. "GridSim:A Grid Simulation Toolkit for Resource Management and Scheduling in Large-Scale Grid Computing Environments". 17th IEEE International Symposium on Parallel and Distributed Processing (IPDPS 2002), April 15-19, Fort Lauderdale, FL, USA.

M. Tolba, M. Abdel-Wahab, I. Taha, and A. Al-Shishtawy. 2000. "Distributed Intrusion Detection System for Computational Grids". Second International Conference on Intelligent Computing and Information Systems, March.

Portnoy, L., Eskin, E. and Stofo, S.J. 2001. "Intrusion detection with unlabeled data using clustering." in Proceedings of ACM CSS workshop on data mining applied to security (DMSA-2001).



**BDUL SAMAD ISMAIL,** He is a Deputy Dean (Development) and An a Associate Professor at Faculty of Computer Science & Information System, University Teknologi Malaysia. He received his B.Sc. in Mathematical./Computer Science from University of Wisconsin Superior, his M.Sc. in Computer Science received from Central Michigan Univeristy, he also graduated his Ph.D in similar field (Computer Science) from University of Swansea. His email is address is : abdsamad@utm.my.



**DAHLIYUSMANTO** was born in Pekanbaru, Riau - Indonesia and went to the University of Putra Indonesia Padang – West Sumatera, where he studied computer engineering and obtained his degree in 1996. He worked for Danamon Bank of Indonesia before moving in 2001 to the Universiti Teknologi Malaysia to continue his Master degree. He studied computer sains and completed his study in 2004. Now, he is a Ph.D student focus on Grid security field. At the same time, he was approved as Lecturer in Faculty of Engineering, University of Riau, Indonesia. His e-mail address is : `yoes_mantho@sigma-snt.com` and his Web-page can be found at `http://www.sigma-snt.com`.



**ABDUL HANAN ABDULLAH,** He is a Dean and Professor at Faculty of Computer Science & Information System, Universiti Teknologi Malaysia. He received his B.Sc and M.Sc in computer science from University of San Fransisco, California USA. He also graduated his Ph.D in the similar field (Computer Science) from Aston University, Birmingham. His email is : hanan@utm.my and and his Web-page can be found at http://www.csc.fsksm.utm.my/~hanan