# DDDAS/ITR: A Data Mining and Exploration Middleware for Grid and Distributed Computing

Jon B. Weissman, Vipin Kumar, Varun Chandola, Eric Eilertson, Levent Ertoz, Gyorgy Simon, Seonho Kim, Jinoh Kim

Dept. of Computer Science and Engineering, University of Minnesota, Twin Cities
(jon@cs.umn.edu)

**Abstract.** We describe our project that marries data mining together with Grid computing. Specifically, we focus on one data mining application - the Minnesota Intrusion Detection System (MINDS), which uses a suite of data mining based algorithms to address different aspects of cyber security including malicious activities such as denial-of-service (DoS) traffic, worms, policy violations and inside abuse. MINDS has shown great operational success in detecting network intrusions in several real deployments. In sophisticated distributed cyber attacks using a multitude of wide-area nodes, combining the results of several MINDS instances can enable additional early-alert cyber security. We also describe a Grid service system that can deploy and manage multiple MINDS instances across a wide-area network.

**Keywords:** Data mining, Grid Computing

## 1 Introduction

MINDS contains various modules for collecting and analyzing massive amounts of network traffic (Figure 1). Typical analyses include behavioral anomaly detection, summarization, scan detection and profiling. Additionally, the system has modules for feature extraction and filtering out attacks for which good signatures have been learned [3]. Each of these modules will be individually described in the subsequent sections. Independently, each of these modules provides key insights into the network. When combined, which MINDS does automatically, these modules have a multiplicative effect on analysis. As shown in the figure, MINDS system involves a network analyst who provides feedback to each of the modules based on their performance to fine tune them for more accurate analysis.

While the anomaly detection and scan detection modules aim at detecting actual attacks and other abnormal activities in the network traffic, the profiling module detects the dominant modes of traffic to provide an effective profile of the network to the analyst. The summarization module aims at providing a concise representation of the network traffic and is typically applied to the output of the anomaly detection module to allow the analyst to investigate the anomalous traffic in very few screen-shots.

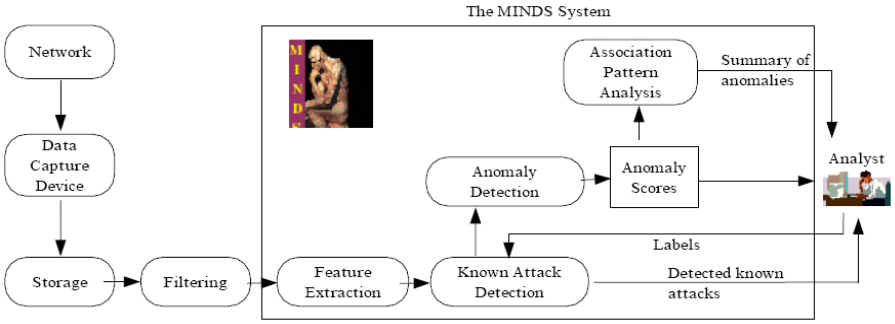


Fig. 1. Minds Architecture

MINDS is deployed at the University of Minnesota, where several hundred million network flows are recorded from a network of more than 40,000 computers every day. MINDS is also part of the Interrogator architecture at the US Army Research Lab - Center for Intrusion Monitoring and Protection (ARL-CIMP), where analysts collect and analyze network traffic from dozens of Department of Defense sites [4].

### 1.1 Anomaly Detection

The core of MINDS is a behavioral anomaly detection module that is based upon a novel data-driven technique for calculating the distance/similarity between points in a high dimensional space. A key advantage of this technique is that it makes it possible to meaningfully calculate similarity between records that have a mixture of categorical and numerical attributes (such as network traffic records) based upon the nature of the data. Unlike other anomaly detection methods extensively investigated by the intrusion detection community, this new framework does not suffer from a high number of false alarms. In fact, ARL-CIMP considers MINDS to have the first effective anomaly detection scheme for intrusion detection. A key strength of this technique is its ability to find behavioral anomalies. Some real examples from its use in the DoD network are identification of streaming video from a DoD office to a computer in a foreign country and identification of a back door on a hacked computer. To the best of our knowledge, no other existing anomaly detection technique is capable of finding such complex behavior anomalies while maintaining very low false alarm rate. A multi-threaded parallel formulation of the anomaly detection module allows analysis of network traffic from many sensors in near real time at the ARL-CIMP.

### 1.2 Detecting Distributed Attacks

Another interesting aspect of the problem of Intrusion Detection is that often times the attacks are launched from multiple locations. In fact, individual attackers often control a large number of machines and may use different machines to launch a

different step of the whole attack. Moreover the targets of the attack could be distributed across multiple sites. Thus an intrusion detection system running at one site may not have enough information to detect the attack by itself. Rapid detection of such distributed cyber attacks requires an inter-connected system of IDSs that can ingest network traffic data in near real-time, detect anomalous connections, communicate their results to other IDSs, and incorporate the information from other IDSs to enhance the anomaly scores of such threats. Such a system consists of several autonomous IDSs that share their knowledge bases with each other for swift detection of malicious, large-scale cyber attacks. We illustrate the distributed aspect of this problem with the following example. Figure 2 shows the 2-dimensional global IP space such that every IP allocated in the world is represented in some block. The black region represents the unallocated IP space. Figure 3 shows a graphical illustration of suspicious connections originating from outside (box on the right) to machines inside the University of Minnesota's IP space (box on the left) in a typical time window of 10 minutes. Each red dot in the right box represents a connection made by that machine to an internal machine on port 80 that is suspicious. In this case, this means that the internal machine being contacted does not have a web-server running, thus making the external machines that are attempting to make connections to port 80, to be suspected attackers. The right box indicates that most of these potential attackers are clustered in specific address blocks of the Internet. A close examination shows that most of the dense areas belong to network blocks located in cable/AOL users in USA or blocks allocated to Asia and Latin America. There are totally 999 unique sources involved on the outside trying to contact 1126 destinations inside the U of M IP network space. The total number of involved flows is 1516 which means that most of the external sources made just one suspicious connection to inside. It is hard to tag a source as malicious based on just one connection. If multiple sites running the same analysis across the IP space report the same external source as suspicious, it would make the classification much more accurate.

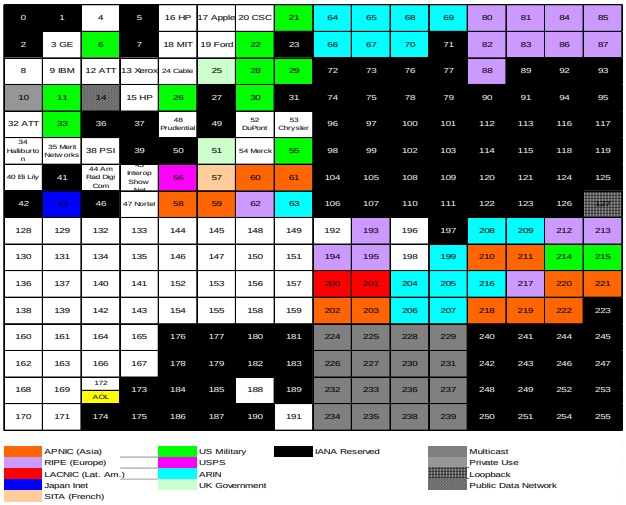


**Fig. 2.** Map of global IP space

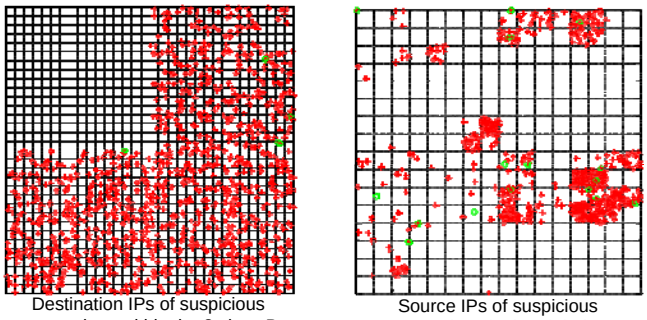


**Fig. 3.** Suspicious traffic on port 80

The ideal scenario would be that we bring in the data collected at these different sites at one place and then analyze it. But this is not feasible due to following reasons – firstly, the data is naturally distributed and is more suited for a distributed analysis; secondly, the cost of merging huge amounts of data and running analysis at one site is also very high and finally there are privacy, security and trust issues that arise in sharing the network data between different organizations. Thus what is really required is a distributed framework in which these different sites can independently analyze their data and then share the high-level patterns and results while honoring the privacy of data from individual sites. The implementation of such a system would require handling distributed data, privacy issues and the use of data mining tools, and would be much easier if a middleware provided these functions. Development and implementation of such a system (see Figure 4) is currently in progress as part of an NSF funded collaborative project between University of Minnesota, University of Florida and University of Illinois, Chicago.

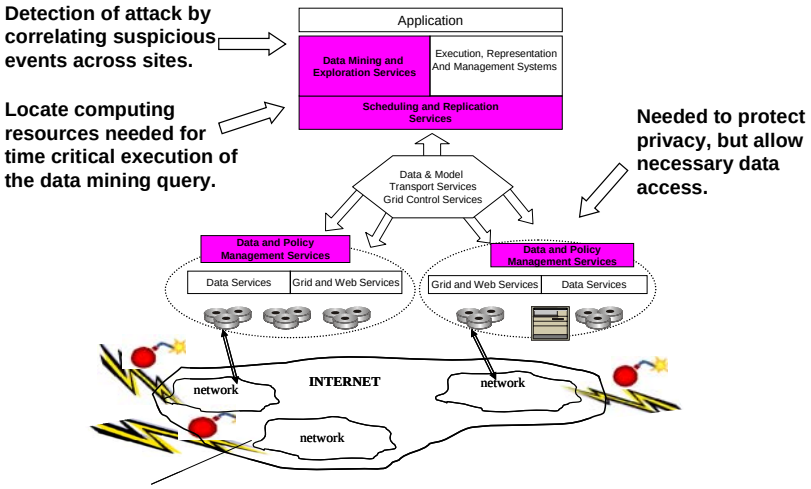


**Fig. 4.** Distributed network intrusion detection system

## 2 Grid Middleware

To support distributed data mining for network intrusion (as envisioned in Figure 4), we have developed a framework that leverages Grid technology. The MINDS system is transformed into a Grid service and deployed on distributed service containers. By separating the process of training set generation (data pre-processing) from the given input data, the data can be decomposed into multiple fragments and distributed among different MINDS Grid services for parallel analysis. The MINDS Grid service front-end is an entry point for the MINDS service requests. The scheduling module inside the front-end sets up a plan regarding where to pre-process data for training set generation if necessary, how to decompose the data, where to send the decomposed data (that is, where to run the MINDS Grid service back-end), whether to deploy new MINDS back-end Grid services if necessary, where to aggregate the result datasets, and finally where to store. The MINDS Grid service front-end may coordinate with different Grid services such as storage service, and with various middleware-level services such as replication service, data management service (e.g., GridFTP [6] for data transfer), and security infrastructure for authentication and authorization. For regulated data access within the data mining community, a community level security authority (CSA, Community Security Authority) is required. The authority expresses policies regarding four main principals in the community – users (and user groups), resource providers (storage provider and compute provider), data (and data groups), and applications - and relationships between them. The CSA uses a catalogue service to manage the catalogue of raw input data, processed data (alert and summary), and replicated datasets, and it maintains a database containing policies.

The system consists of three main component services: MINDS Service, Storage Service, and Community Security Authority (Figure 5). In order to address the first requirement - exploitation of geographically and organizationally distributed computing resources to solve data-intensive data mining problem, we designed the MINDS service as a composite service of a front-end and multiple back-ends. The MINDS Grid service front-end supports planning, scheduling, and resource allocation for MINDS anomaly analysis. MINDS Grid service front-end service should adapt dynamically changing environment to make efficient decisions. We have developed runtime middleware frameworks that can be plugged in the front-end: a dynamic service hosting framework [7], a resource management middleware for dynamic resource allocation and job scheduling [8]. In collaboration with these runtime middleware frameworks, the front-end can set up a plan for data pre-processing (training set generation), input data decomposition, and parallel anomaly analysis, and aggregation of analysis results from MINDS back-end services, and finally store the aggregated analysis result into one or more storage services. Each MINDS back-end service does the actual MINDS anomaly analysis by taking one or more decomposed input data fragments and a training data. The MINDS front-end service coordinates with different Grid services such as storage service, Grid Security Infrastructure, and other various middleware-level services such as catalogue service, replica management service, data management service (e.g., GridFTP for data transfer) on top of Globus Toolkit infrastructure. To ensure the privacy and sensitivity of data, every communication between clients and Grid services are encrypted based on TLS

(Transport Level Security)-based communication and PKI (Public Key Infrastructure)-based authentication and authorization.

### 2.1 Prototype Implementation

We have developed a Java-based prototype of the MINDS grid service consisting of a front-end service and multiple back-end service using GT4.0. Through a user interface, user-custom configuration files and the location of input network flow data are submitted to the MINDS front-end service. The front-end sets up a plan and runs analysis in parallel on multiple back-end services. On completion of analysis, each back-end service returns the result to the front-end and the front-end aggregates the results and calls selected storage service to store the analysis result. The MINDS grid service is packaged as a GAR (Grid Application aRchive) to be deployed into Globus container or is packaged as a Web Application aRchive file (WAR) to be deployed into Tomcat service container.

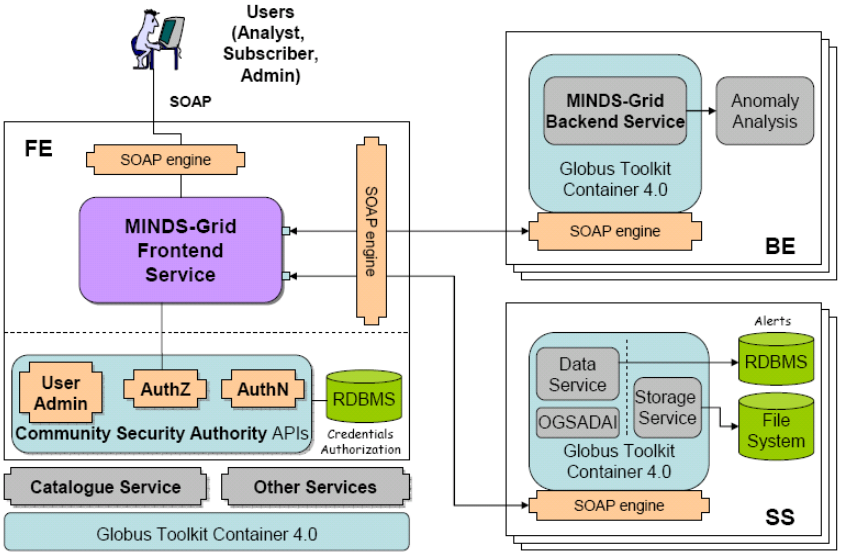


Fig. 5. MINDS Grid System Architecture

## 3 Performance Evaluation

We measure performance of the system on a testbed built on four Linux systems (2x 652 MHz Intel Pentium III processor with 2GB memory running ubuntu 6.06)

connected by 100MB Ethernet and a Windows system (1.8 GHz Intel Pentium Mobile processor with 1GB memory running Microsoft Windows XP) connected by Wireless LAN 802.11b. We deployed MINDS Grid service front-end on a Linux machine, three MINDS Grid service back-ends on other three Linux machines, and finally two different versions of storage services on a Windows machine. In all experiments, at least 10 trials are conducted and the measurements are averaged with error bars with a 95% confidence interval. The experiments consist of three parts. First, we measure the performance of each functional unit varying input data size. The functional units are 1) connection setup time, 2) application running time (MINDS analysis), 3) alert storing time, and 4) alert retrieval time. Second experiment is to evaluate the efficiency of distributed MINDS analysis. We use three different decomposition factors (1 to 3) in this experiment.

### 3.1 Performance evaluation functional units

We decompose the workflow into multiple functional units and measure the time delay of each functional unit to understand which functional units are dominant. With regard to input data, three different input data sets (1,000 records, 10,000 records, and 50,000 records) are used. As shown in figure 6 (top), the overall execution time linearly increases as the input data size increases. Figure 6 (bottom) shows detailed performance of each functional unit in a logarithmic scale. First of all, there is no difference in the connection setup time among different input size as expected. Secondly, MINDS analysis and Retrieval are dominant operations in the sense that the execution time of each operation increases sharply as the input size increases. On the other hand, the input size does not impact that much on the execution time of store operation. Hence, we can see Analysis and Retrieval operations can be bottlenecks in the overall performance.

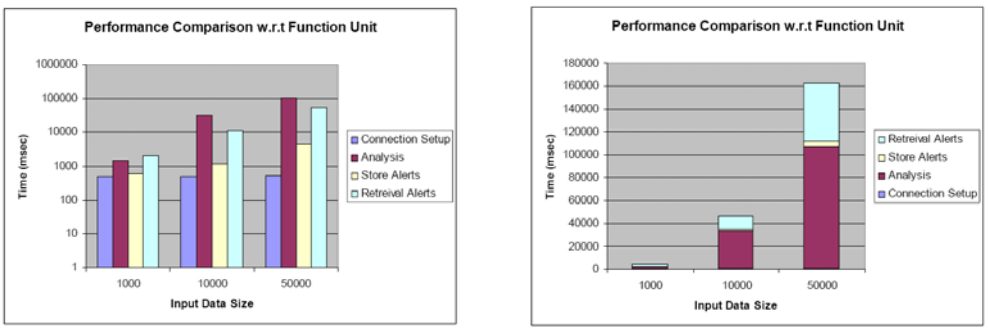


**Fig. 6.** Performance of Functional Units

## 4 Conclusion

We described the two components of this project. First, MINDS was presented which uses a suite of data mining based algorithms to address different aspects of cyber security including malicious activities such as denial-of-service (DoS) traffic, worms, policy violations and inside abuse. MINDS has shown great operational success in detecting network intrusions in several real deployments. We also presented a security enabled Grid system that supports distributed data mining, exploration and sharing using MINDS. The system addresses issues pertaining to the three main requirements of distributed data mining on Grid: 1) exploiting of geographically and organizationally distributed computing resources to solve data-intensive data mining problems, 2) ensuring the security and privacy of sensitive data, and 3) supporting seamless data/computing resource sharing. We designed a system architecture are built on a layered Grid system stack that address 1 and 3. For 2, we developed a community security authority (CSA) which supports secure communication between entities, authentication, and authorized access control. We leveraged existing technologies such as TLS, PKI, and grid security infrastructure to support secure communication, user authentication, and mutual authentication between software clients and servers. Two access schemes - RPBAC and SCBAC were developed to effectively regulate various security related activities and access in the MINDS Grid VO.

## References


[1] L. Ertoz, M. Steinbach, V. Kumar, *A New Shared Nearest Neighbor Clustering Algorithm and its Applications*, 2nd SIAM International Conference on Data Mining, 2002.

[2] V. Chandola and V. Kumar. *Summarization – Compressing Data into an Informative Representation*, Technical Report, TR 05-024, Dept. of Computer Science, Univ of Minnesota, Minneapolis, USA, 2005

[3] L. Ertoz, E. Eilertson, A. Lazarevic, P. Tan, J. Srivastava, V.Kumar, and P. Dokas, The MINDS - Minnesota Intrusion Detection System, *"Next Generation Data Mining*, MIT Press, 2004".

[4] E. Eilertson, L. Ertoz, V. Kumar, and K. Long, Minds -- a new approach to the information security process, *In the 24th Army Science Conference*. US Army, 2004.

[5] W. W. Cohen, Fast effective rule induction, In *International Conference on Machine Learning* (ICML), 1995.

[6] Globus GT4: www.globus.org, 2006

[7] J.B. Weissman, S. Kim, and D. England, A Dynamic Grid Service Architecture, *IEEE International Symposium on Cluster Computing and the Grid (CCGrid2005)*, May, 2005, Cardiff, UK.

[8] B. Lee and J.B. Weissman, ``Adaptive Resource Selection for Grid-Enabled Network Services", *2nd IEEE International Symposium on Network Computing and Applications*, April 2003.