

INBOUNDS: The Integrated Network-Based Ohio University Network Detective Service

*Brett Tjaden, Lonnie Welch, Shawn Ostermann, David Chelberg, Ravindra Balupari, Marina Bykova,
Aaron Mitchell, Denis Lissitsyn, Lu Tong*
School Of Electrical Engineering and Computer Science, Ohio University
Athens, Ohio - 45701, USA

and

Michael Masters, Paul Werme, David Marlow, Brett Chapell, Philip Irely IV
The Naval Surface Warfare Center,
Dahlgren, Virginia - 22448, USA

Abstract

INBOUNDS is a real-time network based intrusion detection system being developed at Ohio University. INBOUNDS detects suspicious behavior by scrutinizing network information generated by TCPTrace [9] (a traffic analysis tool) and host data gathered by the monitors of DeSiDeRaTa [23-27] (dynamic, real-time resource management middleware). The use of these data sources is the major distinction between INBOUNDS and existing intrusion detection systems. By utilizing TCPTrace and DeSiDeRaTa INBOUNDS is able to function in a heterogeneous environment with fault tolerance, very low overhead, and a high degree of scalability. INBOUNDS is currently being used for around-the-clock intrusion detection at Ohio University.

Keywords: Intrusion Detection System, adaptive distributed real-time secure system, INBOUNDS, SECURE-RM.

1. Introduction

An Intrusion Detection System (IDS) monitors a computer system to identify attempts at intrusion and misuse. As highlighted in [2], the name Intrusion Detection System emphasizes the threat of intrusion or "attempts to use a computer system without authorization" by outsiders, but misuse or "abuse of existing privileges" by legitimate users is an equally important concern of any IDS. True IDSs are strictly detective and do not take any action upon identifying an intruder other than to notify a designated authority.

In [16], Intrusion Detection Systems are categorized as based on either the misuse detection model or the anomaly detection model defined as follows:

- Misuse detection model: Detection is performed by looking for the exploitation of known weak points in the system which can be described by a specific pattern or sequence of events or data (the "signature" of the intrusion).
- Anomaly detection model: Detection is performed by detecting changes in the patterns of utilization or behavior of the system. It is performed by building a statistical model that contains metrics derived from system

operation and flagging as intrusive any observed metrics that have a significant statistical deviation from the model.

IDSs based on the misuse detection model typically do a very good job of detecting the types of intrusions for which they have signatures while introducing little detection overhead into the system. Their major weakness, however, is that new or nonstandard attacks will go unnoticed until signatures for those specific methods of intrusion are added to the IDS. IDSs based on the anomaly detection model have a better chance of detecting new or nonstandard intrusion methods, but deciding what metrics to use and how to interpret them can be very difficult and might require significant processing by the IDS.

Intrusion Detection Systems can be further categorized as either host based (data from a single host is scrutinized), multihost based (data from multiple hosts is used), or network based (network traffic and possibly data from hosts attached to the network is analyzed). Furthermore, some IDSs are real-time because they monitor the system continuously and report intrusions as soon as they are detected, but some IDSs operate off-line by inspecting system logs at set intervals and then recounting any suspicious activity that was recorded. An off-line IDS typically reduces system overhead since it only runs periodically, but an off-line IDS also gives much less timely notification of incidents for that same reason.

Lastly, an IDS is centralized if data may be collected from various sources (hosts or networks) but is shipped to a centralized location where it will be analyzed. Distributed IDSs spread the detection mechanism across all of the hosts being monitored. A centralized IDS represents a single point of vulnerability and a bottleneck that could impede system scalability. Furthermore, as discussed in [19], an intruder could avoid detection by cleansing data being sent to the centralized IDS for analysis. Distributed IDSs must ensure cooperation and agreement among their various components since these represent possible weaknesses that an intruder could exploit to avoid detection.

2. Related Work

There are a number of network based IDSs to which INBOUNDS is related. The U.S. Air Force's Automated Security Incident Measurement (ASIM) system [5] is used for

off-line misuse detection at a number of AirForce installations around the world. Los Alamos National Laboratories' Network Anomaly Detection and Intrusion Reporter (NADIR)[7] performs off-line anomaly detection on a large network of supercomputers and has been recently updated to operate as a real-time IDS (UNICORN) [4]. Commercially-available centralized network based IDSs include: CyberCop [11] by Network General Corporation, RealSecure [8] by Internet Security Systems Incorporated, INTOUCH INSA [14] by Touch Technologies Incorporated, and Network Associates' Sniffer Total Network Visibility (TNV) [10]. These are centralized systems that perform misuse detection.

Distributed network based IDSs are still in the early research stage of development. The University of California, Davis' Graph-based Intrusion Detection System (GrIDS) [3] and SRI's Event Monitoring Enabling Response to Anomalous Live Disturbances (EMERALD) [17] are two such research projects for which prototypes are currently being built. GrIDS creates activity graphs which "approximately represent the causal structure of large scale distributed activities." [3]. Activity graphs are built dynamically with nodes representing hosts and edges representing communication activity between two hosts. GrIDS examines activity graphs for patterns that indicate hostile or intrusive activities. For example, the spread of a worm over a computer network would produce a tell-tale propagation pattern in the corresponding activity graph. SRI's EMERALD utilizes "highly distributed, independently tunable, surveillance and response monitors that are deployed at various abstract layers in the network." [17]. Analysis is performed in a hierarchical manner with localized checking for misuse of "key

domain services and assets" and globalized analysis aimed at detecting "network-wide coordinated attacks." EMERALD's main concern is external intruders who attempt to subvert or bypass network interfaces and controls to gain unauthorized access to system resources.

3. The INBOUNDS System

INBOUNDS is a network-based, real-time, hierarchical IDS that performs both misuse and anomaly detection. INBOUNDS is designed to function in a large, distributed real-time systems that have execution times and resource utilizations which cannot be characterized *a priori*. (The motivation for our work is provided in part by the characteristics of *combat systems*, as described in [6].) There are several implications of these characteristics: 1) demand space workload characterizations may need to be determined *a posteriori*, and 2) an adaptive approach to resource allocation may be necessary to accommodate dynamic workload changes. Thus, distributed computing resources must be managed continuously and dynamically to insure that the specified security and quality of service (QoS) requirements are satisfied. This is accomplished by continuously computing and assessing security, QoS, and resource utilization metrics that are determined *a posteriori*. This paper presents an adaptive distributed system reference architecture that is suitable for such an approach. This reference architecture provides the capabilities and infrastructure needed to construct multi-component, replicated, distributed object real-time secure systems that negotiate for a given level of service from the underlying distributed computing resources.

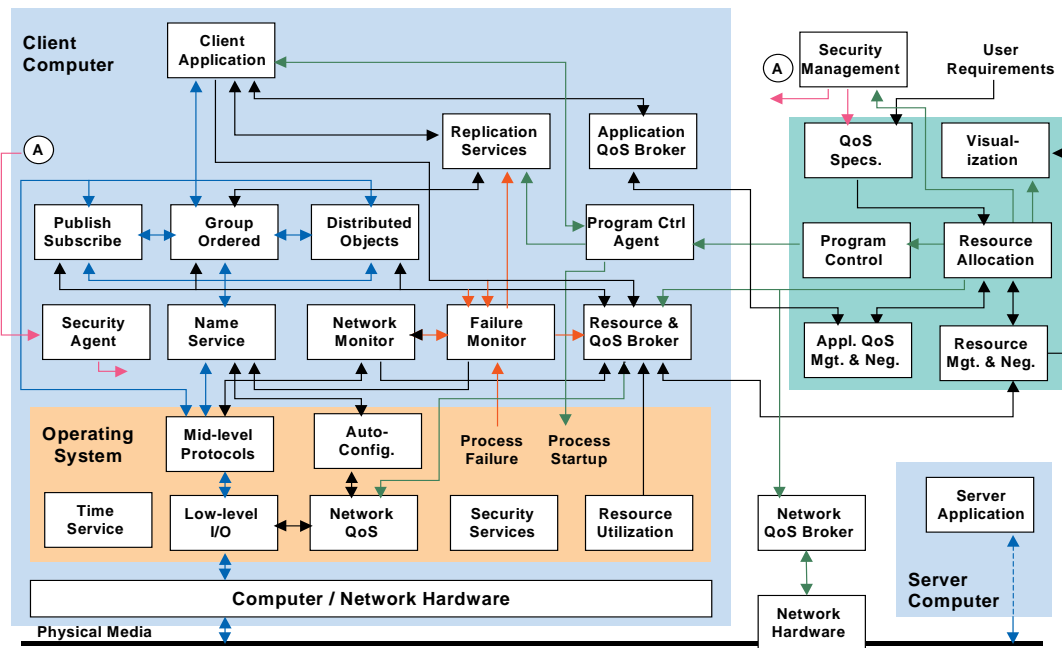


Figure 1: A distributed system reference architecture

Figure 1 depicts the distributed system reference architecture. The diagram shows the functional architecture structure needed

to support distributed application programs for a computer containing a client application. This structure is repeated

throughout the computers of the distributed system; in particular, the computer hosting the server application contains a comparable structure. Also executing somewhere in the distributed system's collection of computers is a set of QoS management components that interact with the computers, network components and applications of the distributed system to provide QoS management.

Besides the applications themselves, the components of the reference architecture consist of four primary types: operating systems, network services, high-level communication and state management middleware services, and resource management services. The operating system services are those needed to support real-time applications. The network services provide low-level network communications and time management capabilities. The middleware components provide the ability to communicate in accord with three high level communication models: publish-subscribe; group programming, with associated ordered multicast and state data synchronization services; and distributed object programming. The resource management services consist of computing resource and application status monitoring and reporting services, QoS negotiation services, and program control services.

Taken together, these services allow distributed applications to perform allocated processing functions, to communicate with

each other, to perform fault detection and recovery activities, to perform load sharing activities, and to negotiate resource utilization and QoS requirements with the underlying distributed system infrastructure. This architecture has been partially implemented and successfully employed for dynamic management of QoS and distributed computing resources within a Navy distributed computing testbed. Features of the testbed include real-time mission critical computing [6], fault tolerance and scalability (for a description of the testbed, see [27]).

The remainder of this paper will focus on the portion of the reference architecture which pertains to security. The components which constitute security services are: (1) data collection, (2) current data repository, (3) historical data repository, (4) data visualization (display), and (5) intrusion detection services. These five components are currently realized by the INBOUNDS intrusion detection system.

3.1 Architecture

As illustrated in Figure 2 the INBOUNDS system is composed of five high-level components: data collection, a current data repository, a historical data repository, data visualization (display), and intrusion detection services. We will discuss each of these components in turn.

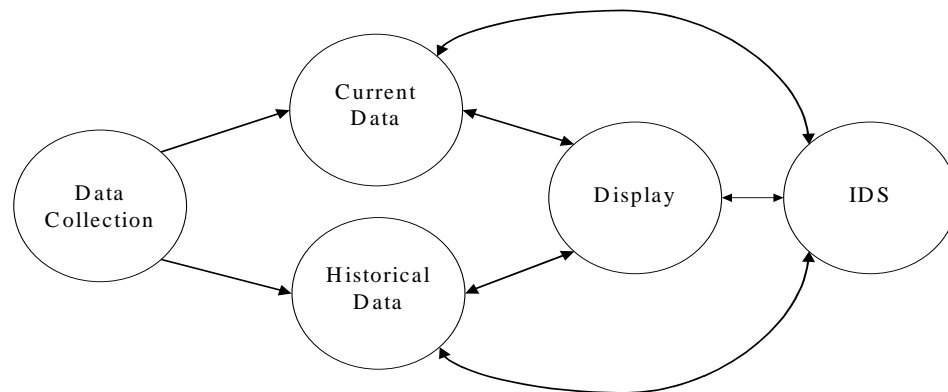


Figure 2: Architecture of the INBOUNDS System

Data collection is performed by a group of modules that capture, filter, process, and summarize information about the networks, hosts, users, processes, and resources of a dynamic, distributed system. TCPTrace [9] is an example of one data collection module that provides information about a network and the TCP connections being carried on it. A DeSiDeRaTa [23-27] host monitor is another data collection module which tracks host and resource utilization. Each data collection module produces a real-time stream of information which flows optionally through a filter and then to the current and historical data repositories.

The current data repository accepts streams of data from the various data collection modules and passes on only the streams (or parts of streams) requested by the analysis and visualization

modules. Examples include connection notifications from TCPTrace [9] and host and resource usage updates from the host monitors. The data that passes through the current data repository is the most recent and detailed data in the system.

The historical data repository contains older and more coarse-grained information. This module accepts the same real-time streams from the data collection modules, but it abstracts the information storing only a brief summary of the stream. The data analysis and visualization modules interact differently with the historical than with the current data repository. Pieces of information must be explicitly requested from the historical data repository using a client-server model of interaction. For example, by interacting with the historical data repository an analysis module could learn the remote sites from which a user

has connected in the past, what kinds of services and resources a user or site typically uses, and whether or not a site or user has engaged in suspicious behavior in the past.

The data visualization module is intended to allow a human to view and make sense of the huge amount of current and historical data in the system. This module's job is to allow the user to navigate between various levels of abstraction and different views of the data to be able to discern the information in which he or she is interested. Currently this module is a graphical user interface designed using the Java programming language.

The data analysis modules perform the actual intrusion detection functions. There are two separate modules which can communicate with each other if necessary. These modules include a misuse detection module and an anomaly detection module. Within each of these are submodules for action-based and resource-based detection. The misuse detection module looks for sequences of actions that match the "footprints" for a set of known attacks. Action-based anomaly detection keeps detailed statistics about normal network, host, and user behavior and raises alarms whenever current behavior differs significantly from the norm. Resource-based anomaly detection performs a related type of statistical analysis of resource behavior and usage and issues warnings when it observes discrepancies. Whenever any of the intrusion detection modules recognize suspicious behavior they notify the historical data repository and visualization subsystem so that the alert can be recorded and displayed to the user.

3.2 INBOUNDS in Action

In addition to serving as an around-the-clock intrusion detection system at Ohio University, INBOUNDS has been integrated with the DeSiDeRaTa [23-27] dynamic, real-time resource management middleware. Upon notification of intrusion events by INBOUNDS the DeSiDeRaTa [23-27] system takes various actions based on the severity of the intrusion. For innocuous attacks the alert may be ignored entirely or mission-critical applications may be migrated off of the host being attacked. For more serious intrusion events the resource manager moves all programs off the targeted host immediately to prevent an attacker from causing a quality-of-service violation.

3.3 Future Work

Currently we are working to extend the number and types of attacks INBOUNDS can detect. We also hope to elaborate the link between INBOUNDS and DeSiDeRaTa [23-27] so that we can analyze an attacker's (or group of attackers') actions to determine a probable strategy (in terms of resources being targeted). We would then like to assist a decision maker in developing a course of counteraction (See Figure 3). This would include interacting with the decision maker to apprise him/her of the current state of the system, present a probability-based diagnosis of the intruder's likely strategies and goals, and recommend possible response actions including their costs and probable outcomes.

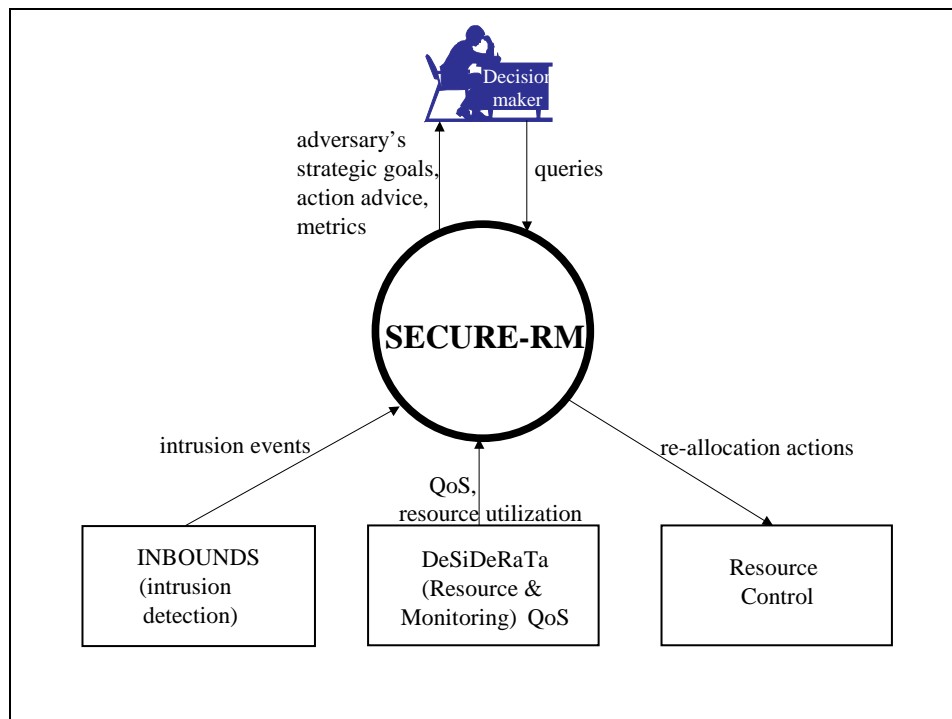


Figure 3: Architecture for SECURE-RM

Once the decision maker has selected a response action it could be optimized and carried out by INBOUNDS and the resource manager. We call this system that combines security with resource management SECURE-RM

A look inside the SECURE-RM module illustrates its planned functionality (Figure 4). The primary architectural components of the SECURE-RM architecture are (1) attack strategy analyzer, (2) action advisor, and (3) allocation optimizer. The attack strategy analyzer (ASA) determines the strategic goals

of the attacker by relating individual intrusion events with the demand and supply space models used within the DeSiDeRaTa resource management system. These models represent software systems' attributes (such as system composition), hardware systems' attributes (e.g., topology), and allocation state (the current mapping of software components to hardware resources, observed QoS, and resource utilization). Additionally, the ASA will present attack strategy information to the decision maker in a graphical form that is at a relevant

level of abstraction. The action advisor (AA) will determine possible reallocation actions to respond to attacks by considering software and hardware systems' attributes, allocation state, QoS of application systems, resource utilization and strategic goals of attackers. To assess possible reallocation actions, the AA will consult the allocation optimizer, which will calculate the *benefit* of candidate reallocation actions.

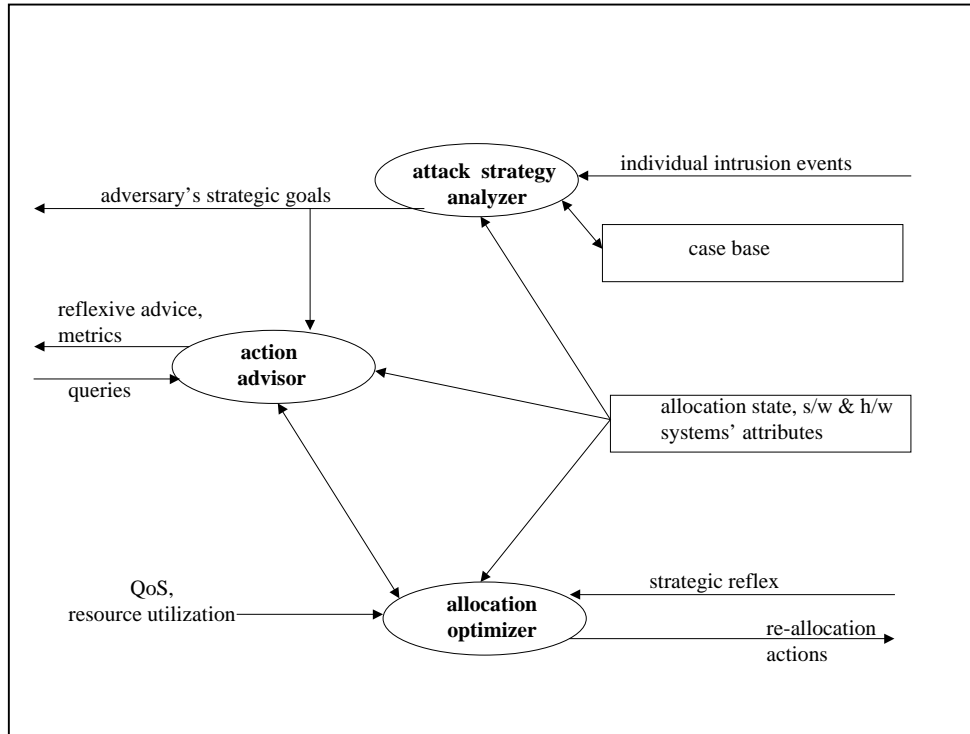


Figure 4: Internal view of the SECURE-RM component.

4. Conclusion

INBOUNDS is a network-based, real-time, hierarchical IDS that performs both misuse and anomaly detection. INBOUNDS detects suspicious behavior by scrutinizing network information generated by TCPTrace [9] (a traffic analysis tool) and host data gathered by the monitors of DeSiDeRaTa [23-27] (dynamic, real-time resource management middleware). This novel approach has significant advantages in that it does not require the addition of any type of software agent on the computers being monitored and thus is non-intrusive and low overhead. This allows INBOUNDS to detect a wide-range of important attacks including:

- Abnormal network protocol behavior including SYN and RESET attacks
- Suspicious keywords in interactive sessions/email
- Suspicious patterns of data, such as the fan-out patterns commonly seen with email viruses

In addition, by keeping historical information about previous network traffic, INBOUNDS detects the following types of attacks:

- Communication over unusual network ports, which are common when attackers target seldom used and insecure servers
- Connections from unknown/unusual hosts
- Abnormal data patterns for a particular time of day
- Unusual data patterns on known ports, such as would be seen when an attacker installs programs using the finger port as in the Morris Worm

INBOUNDS is currently being used for around-the-clock intrusion detection at Ohio University and is being integrated with the DeSiDeRaTa resource manager to create SECURE-RM, a software system to address the general problems of *situation awareness* and *course of action development and execution*, in the context of security and resource management for dependable, real-time, dynamic, distributed systems.

5. References

- [1]. D. Anderson, T.F. Lunt, H. Javitz, A. Tamaru, and A. Valdes, "Detecting Unusual Program Behavior Using the Statistical Component of the Next-Generation Intrusion Detection Expert System (NIDES)", SRI-CSL-95-06, SRI International, Menlo Park, CA, May, 1995.
- [2]. Jai Balasubramaniyan, Jose Omar Garcia-Fernandez, David Isacoff, E. H. Spafford, and Diego Zamboni, "An Architecture for Intrusion Detection using Autonomous Agents", Department of Computer Sciences, Purdue University; Coast TR 98-05; 1998.
- [3]. S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, and D. Zerkle, "GrIDS - A Graph Based Intrusion Detection System for Large Networks", Proceedings of the 19th National Information Systems Security Conference, October, 1996, pp 361-370.
- [4]. G.G. Cristoph, K.A. Jackson, M.C. Neumann, C. Siciliano, L.B. Ch, D.D. Simmonds, and C.A. Stallings, "UNICORN: Misuse detection for UNICOS", Proceedings of Supercomputing '95.
- [5]. GAO Executive Report - B-266140, "Information Security - Computer Attacks at Department of Defense Pose Increasing Risks", May, 1996.
- [6]. Robert D. Harrison Jr., "Combat system prerequisites on supercomputer performance analysis," in Proceedings of the NATO Advanced Study Institute on Real Time Computing, NATO ASI Series F(127), 512-513, Springer-Verlag 1994.
- [7]. J. Hochberg, K. Jackson, C. Stallings, J. McIlary, J. DuBois, and D. Ford, "NADIR: An automated system for detecting network intrusions and misuse", Computers and Security 12(1993)3, May, 1993, pp 253-248.
- [8]. <http://www.iss.net/prod/rs.html>
- [9]. <http://jarok.cs.ohiou.edu/software/tcptrace/tcptrace.html>
- [10]. http://www.nai.com/asp_set/products/tnv/intro.asp
- [11]. http://www.ngc.com/product_info/cybercop/ccdata/ccdata1.html
- [12]. <http://www.nswc.navy.mil/ISSEC/CID>
- [13]. <http://www-rnks.informatik.tu-cottbus.de/~sobirey/aid.e.html>
- [14]. http://www.ttisms.com/tti/nsa_www.html
- [15]. Gene Kim and Gene Spafford, "Monitoring File System Integrity with Tripwire", InfoSecurity News, July 1993.
- [16]. Biswanath Mukherjee, Todd L. Heberlein, and Karl N. Levitt, "Network Intrusion Detection", IEEE Network, 8(3): 26-41, May/June 1994.
- [17]. A. Porras and P.G. Neumann, "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances", Proceedings of the National Information Systems Security Conference, October 1997.
- [18]. P. Proctor, "Audit Reduction and misuse detection in heterogeneous environments: Framework and applications", Proceedings of the 10th Annual Computer Security Applications Conference, December 1994, pp 117-125.
- [19]. Thomas H. Ptacek and Timothy N. Newsham, "Insertion, evasion, and denial of service: Eluding network intrusion detection", Technical report, Secure Networks, Inc., January 1998.
- [20]. S.E. Smaha, "Haystack: An intrusion detection system", Proceedings of the IEEE 4th Aerospace Computer Security Applications Conference, December 1988, pp 37-44.
- [21]. S.E. Smaha and J. Winslow, "Misuse detection tools", Computer Security Journal 10(1994), pp 39-49.
- [22]. S.R. Snapp, S.E. Smaha, T. Grance, and D.M. Teal, "The DIDS (Distributed Intrusion Detection System) Prototype", Proceedings of the USENIX Summer 1992 Technical Conference, June, 1992, pp 227-233.
- [23]. L. R. Welch, P. Shirolkar, B. Shirazi, et al., "Adaptive Resource Management For Scalable Dependable Real-Time Systems: Middleware Services and Applications to shipboard computing systems", Technical Report, TR-CSE-97-009, The University of Texas at Arlington, December 1997.
- [24]. L. R. Welch, B. A. Shirazi, B. Ravindran and C. Bruggeman, "DeSiDeRaTa: QoS Management Technology for Dynamic, Scalable, Dependable, Real-Time Systems", Proceedings of The 15th IFAC Workshop on Distributed Computer Control Systems, September 1998.
- [25]. L. R. Welch, Binoy Ravindran, Robert D. Harrison, Leslie Madden, Michael W. Masters and Wayne Mills, "Challenges in Engineering Distributed Shipboard Control Systems", in Proceedings of Work-in-Progress Session of The IEEE Real-Time Systems Symposium, December 1996, 19-22.
- [26]. L. R. Welch, B. Ravindran, B. Shirazi and C. Bruggeman, "Specification and analysis of dynamic, distributed real-time systems", in Proceedings of the 19th IEEE Real-Time Systems Symposium, 72-81, IEEE Computer Society Press, 1998.
- [27]. L. R. Welch, Paul V. Werme, Larry A. Fontenot, Michael W. Masters, Behrooz A. Shirazi, Binoy Ravindran and D. Wayne Mills, "Adaptive QoS and Resource Management Using A Posteriori Workload Characterizations, "The IEEE Real-time Technology and Applications Symposium, June 1999.
- [28]. G.B. White and U.W. Pooch, "Cooperating Security Managers: distributed intrusion detection systems", Computers and Security 15(1996)5, pp 441-450.
- [29]. J.R. Winkler, and L.C. Landry, "Intrusion and Anomaly detection, ISOA update", Proceedings of the 15th National Computer Security Conference, October, 1992, pp 272-281.