

SECURE-RM: Security and Resource Management for Dynamic Real-Time Systems

Brett Tjaden, Lonnie Welch, Shawn Ostermann, David Chelberg, Ravindra Balupari, Marina Bykova, Aaron Mitchell, Lu Tong

School Of Electrical Engineering and Computer Science, Ohio University
Athens, Ohio - 45701, USA

Abstract

The global Internet has made real-time computer systems world-wide vulnerable to an ever-changing array of attacks for which current defense mechanisms are insufficient. In order to combat intruders in this new environment new techniques must be developed that enable decision makers to *detect* unusual behavior in their systems, *correlate* anomalies into higher-level attacker goals, *plan* appropriate response actions and *execute* their plans. We are developing SECURE-RM, a security management system that combines an intrusion detection system (INBOUNDS) with adaptive resource management middleware (DeSiDeRaTa) for this purpose. INBOUNDS is a network-based, real-time, hierarchical software system for misuse and anomaly detection. Intrusion events, such as pre-attack probes and denial of service attacks, are detected and are reported to SECURE-RM, which employs artificial intelligence techniques for deriving impacts of attacks on operational functions and mission goals. A strong belief in an attack strategy triggers a resource reallocation by DeSiDeRaTa for response execution.

1. Overview of SECURE-RM

Figure 1 depicts an overview of the SECURE-RM architecture, which will be used to describe our approach for providing security and resource management for dynamic real-time systems. INBOUNDS notifies SECURE-RM of individual intrusion events. This information is combined with knowledge of the software system attributes and the hardware system attributes [1], and information about the current allocation of (hardware) resources to the software systems [1] to ascertain the adversary's strategic goals. The results of the analysis are presented to the decision maker in terms pertaining to system structure, QoS and mission goals.

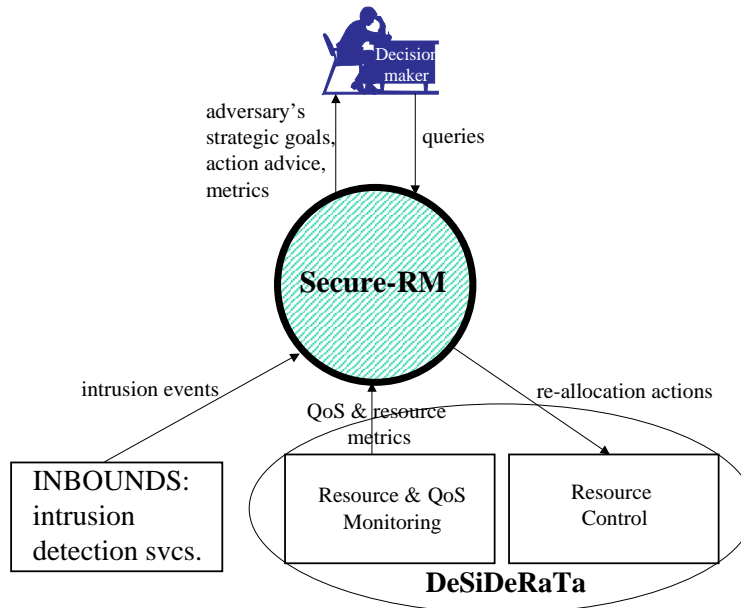


Figure 1 – The Architecture of SECURE-RM

Upon perception of an attack, a set of reflexive actions is developed by SECURE-RM. The action development strategy first considers defensive mechanisms to achieve catharsis; if these are deemed inadequate, appropriate system functional realignments are discovered by assessing QoS and resource utilization (current and projected), as well as knowledge of the software system attributes and the hardware system attributes. The actions are recommended to the human decision maker, who may approve a recommended strategic reflex action, or may make queries to determine if alternate reflexes would be more appropriate. Upon selection of a particular strategic action, SECURE-RM considers QoS and resource utilization to determine a detailed set of reallocation operations to enact the stratagem, and dispatches the set of operations to the resource control component. For a more detailed understanding of our approach, consider the internal view of the SECURE-RM component shown in Figure 2. The primary architectural components of the SECURE-RM architecture are (1) attack strategy analyzer, (2) action advisor, (3) allocation optimizer, and (4) the hierarchical belief network.

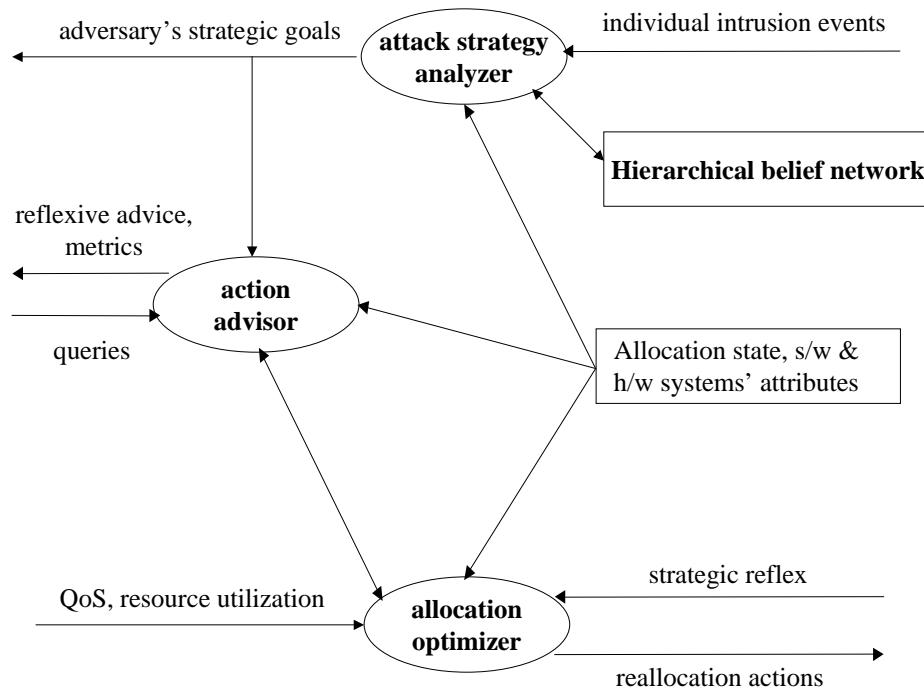


Figure 2 – Internal View of SECURE-RM

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 (described in [2] and [3]) 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 presents attack strategy information to the decision maker in a graphical form that is at a relevant level of abstraction. The action advisor (AA) determines 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, AA will consult the allocation optimizer (AO), which calculates the *benefit* of candidate reallocation actions.

2. The INBOUNDS Intrusion Detection System

INBOUNDS [4] is a network-based, real-time, hierarchical IDS that performs both misuse and anomaly detection. As illustrated in Figure 3, 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.

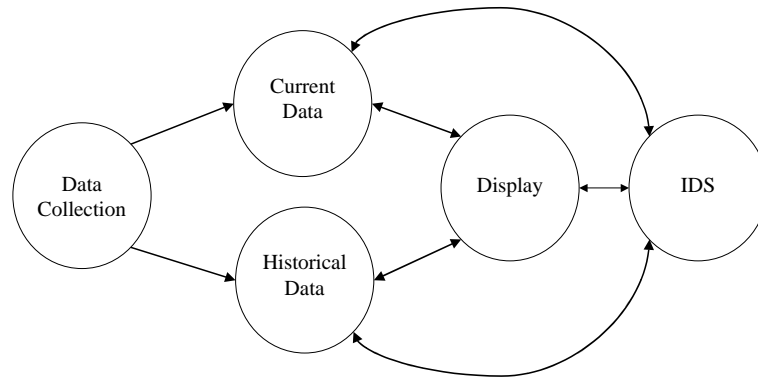


Figure 3 – The Architecture of INBOUNDS

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 [5] [6] is an example of one data collection module that provides information about a network and the TCP connections being carried on it. A DeSiDeRaTa host monitor [2] [3] 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 [5] [6] 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. SECURE-RM

Individual intrusion alerts are passed from INBOUNDS to the Attack Strategy Analyzer, which uses a hierarchical belief network to reason about an attacker's probable goals and strategies (figure 4).

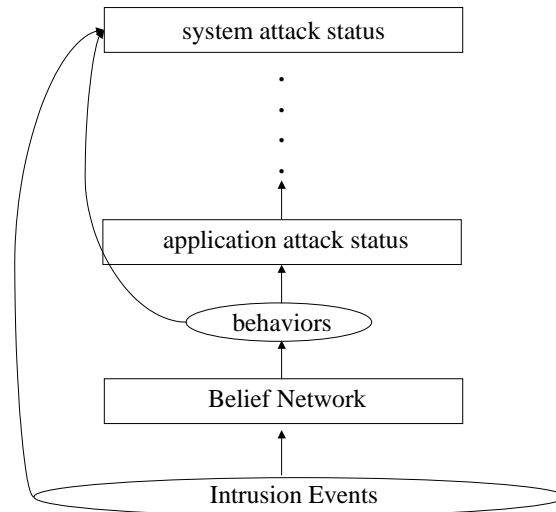


Figure 4 – The ASA's Hierarchical Belief Network

SECURE-RM begins response planning immediately upon notification of intrusion events. However, response execution is not triggered until there is a strong belief in the attacker's strategy or a high degree of danger in his actions. Response planning and execution is performed in consultation with and under the direction of the resource manager.

4. The DeSiDeRaTa Resource Manager

The DeSiDeRaTa project [1] [2] [3] provides an adaptive resource management approach that is appropriate for systems which experience large variations in workload. A distributed collection of computing resources is managed by continuously computing and assessing QoS metrics and resource utilization metrics that are determined a posteriori. The DeSiDeRaTa project provides a specification language for describing environment-dependent features of dynamic real-time systems. Also provided is an abstract model that is constructed (statically) from the specifications, and is augmented (dynamically) with the state of environment-dependent features. The model is being used to develop algorithms for QoS monitoring, QoS diagnosis, and resource allocation analysis. Experimental results show the effectiveness of the approach for specification of real-time QoS, detection and diagnosis of QoS failures, and restoration of acceptable QoS via reallocation of distributed computer and network resources. The logical architecture of the DeSiDeRaTa QoS management software is shown in Figure 5.

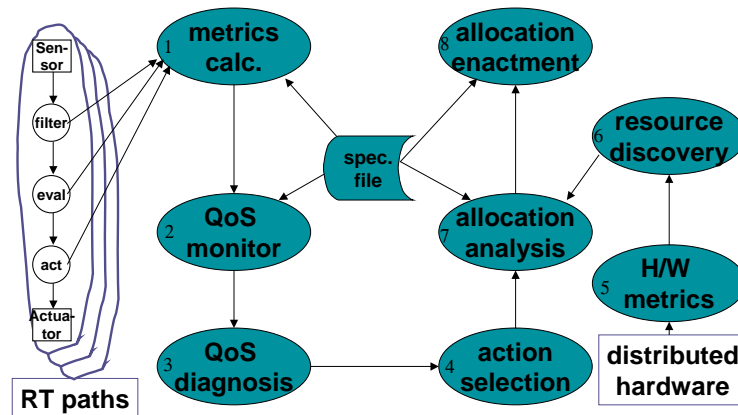


Figure 5 – DeSiDeRaTa Architecture

The application programs of real-time control paths send time-stamped events to the *QoS metrics* component, which calculates path-level QoS metrics and sends them to the *QoS monitor*. The monitor checks for conformance of observed QoS to required QoS, and notifies the *QoS diagnosis* component when a QoS violation occurs (figure 6). The diagnoser notifies the *action selection* component of the cause(s) of poor QoS and recommends actions (e.g., move a program to a different host or LAN, shed a program, or replicate a program) to improve QoS. Action selection ranks the recommended actions, identifies redundant actions, and forwards the results to the *allocation analysis* component; this component consults resource discovery for host and LAN load index metrics and determines a good way to allocate the hardware resources in order to perform the actions, and requests the actions be performed by the *allocation enactment* component.

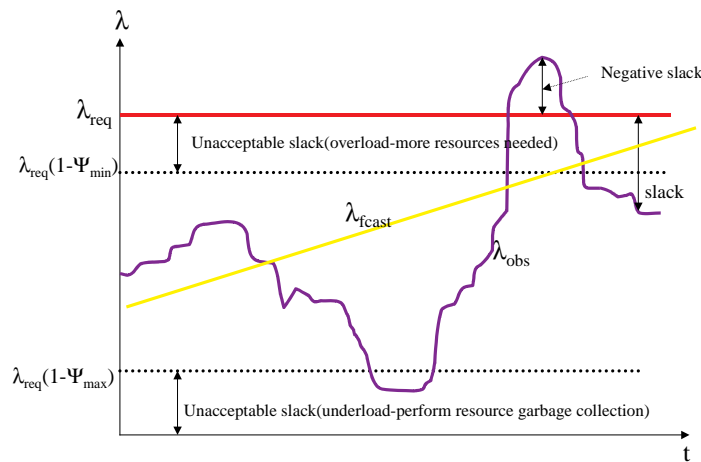


Figure 6 – DeSiDeRaTa QoS Management

By interacting with the Resource Manager, SECURE-RM can determine the effects of its planned response actions on real-time QoS. The Resource Manager is also employed to carry out SECURE-RM's response actions (by actually moving applications or reconfiguring networks).

5. Summary

The global Internet has made real-time computer systems world-wide vulnerable to an ever-changing array of attacks for which current defense mechanisms are insufficient. In order to combat intruders in this new environment new techniques must be developed that enable decision makers to *detect* unusual behavior in their systems, *correlate* anomalies into higher-level attacker goals, *plan* appropriate response actions and *execute* their plans. We are developing SECURE-RM, a security management system that combines an intrusion detection system (INBOUNDS) with adaptive resource management middleware (DeSiDeRaTa) for this purpose. INBOUNDS is a network-based, real-time, hierarchical software system for misuse and anomaly detection. Intrusion events, such as pre-attack probes and denial of service attacks, are detected and are reported to SECURE-RM, which employs artificial intelligence techniques for deriving impacts of attacks on operational functions and mission goals. A strong belief in an attack strategy triggers a resource reallocation by DeSiDeRaTa for response execution.

6. References

- [1] 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.
- [2] 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.
- [3] 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.
- [4] Brett C. Tjaden, Lonnie R. Welch, Shawn D. Ostermann, David Chelberg, et. al, "INBOUNDS: The Integrated Network-Based Ohio University Network Detective Service", 4th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2000) and 6th International Conference on Information Systems Analysis and Synthesis (ISAS 2000), Orlando, Florida, July 23-26, 2000.
- [5] Hans Kruse, Mark Allman, Jim Griner, Shawn Ostermann, Eric Helvey, "Satellite Network Performance Measurements Using Simulated Multi-User Internet Traffic," *Proceedings of the Seventh International Conference on Telecommunication Systems*, March 1999.
- [6] Mark Allman, Chris Hayes, Hans Kruse, and Shawn Ostermann. TCP Performance Over Satellite Links. In Proceedings of the 5th International Conference on Telecommunication Systems, pages 456--469, March 1997.