

Common Issues for Remote Analysis and Adaptive Security

Dennis Heimbigner
University of Colorado
dennis.heimbigner@colorado.edu

Abstract

Adaptive security and remote analysis have many features in common, both operationally and architecturally. Joint efforts would be welcome to develop common infrastructure for inserting sensors, for distributing events, and for managing and viewing large numbers of events.

1. Introduction

Adaptive security has recently surfaced as a general approach to solving a number of security problems. The basic adaptive security model is based on a general sense-analyze-respond model common to other kinds of adaptive systems. In the case of security, however, the three elements apply in security specific ways:

- Sense – sense the current state of a collection of software (and hardware) systems; typically these systems are distributed across a network.
- Analyze – analyze the sequence of sensor readings to detect anomalies in the behavior of the sensed systems; these anomalies may be indicators of malicious behavior.
- Respond – modify the properties and behavior of the sensed systems to achieve, for example, more detailed sensor readings or to improve the systems' intrusion tolerance (the ability to resist or work around attacks).

The first phase of the adaptive security cycle (sense plus analyze) is adopted from the larger intrusion detection community. The goal of intrusion detection is the identification of malicious behavior at the network level (e.g. distributed denial of service) or at the host level (e.g. buffer overflow attacks). Adaptive security extends this paradigm to include (semi-)automated responses to detected attacks using embedded actuators.

The remote analysis paradigm and the adaptive security paradigms are almost identical, and so it should be expected that techniques and infrastructure from each would be of considerable interest to the other community. Both require the ability to insert sensors and actuators into

software systems either off-line (during development and before deployment) or on-line (during deployment or at runtime). They may differ, of course, in the placement of the sensors and actuators and in their operation, but nevertheless, many of the goals of remote analysis – lightweight monitoring, failure diagnosis, dynamic modification of deployed systems – are goals for adaptive security as well.

2. Examples of Adaptive Security Systems

The SERL group at the University of Colorado has developed and is developing several adaptive security systems. They are described in the following sections.

2.1 Willow

The Willow project [4,5] is a joint effort with researchers at the University of Virginia and the University of California, Davis. The goal of the Willow project is to allow software systems to protect themselves when attacked or threatened. The approach is to define protective configurations of the software system and to support the dynamic reconfiguration of the system into a protective posture when attacked.

Willow provides an automated framework for reconfiguration of large-scale, heterogeneous, distributed systems. It is an early example of an adaptive security system. It utilizes software sensors to detect potential future problems (port scans, for example) and to detect current anomalous behavior. Once sensed and analyzed, Willow can initiate both proactive and reactive responses that modify the behavior of the protected systems by reconfiguring them.

Proactive reconfiguration modifies configurations to cause a system to assume postures that achieve enterprise-wide intrusion tolerance goals, such as increased resilience to specific kinds of attacks or increased preparedness for recovery from specific kinds of failures. Proactive reconfiguration can also cause a relaxation of tolerance procedures once a threat has passed, in order to reduce costs, increase system performance, or even restore previously excised data and functionality. In a complementary fashion, reactive reconfiguration modifies

configurations to restore the integrity of a system in bounded time once an intrusion has been detected and the system is known or suspected to have been compromised. Recovery strategies made possible by reactive reconfiguration include restoring the system to some previously consistent state, adapting the system to some alternative non-compromised configuration, or gracefully shedding non-trustworthy data and functionality.

2.2 EventTrails

The EventTrails system [1] targets the “insider-threat” problem. That is, it supports the analysis of sensed events relevant to potential misuse by insiders, and it generated responses that attempt to mitigate the consequences of that misuse. These responses include reconfiguration of an application to modify its behavior, the insertion of additional monitoring capabilities, the insertion of on-the-fly access control policies, and novel deception and functional degradation capabilities that support presenting an increasingly hostile environment to a malicious insider.

EventTrails focuses on document-related activities as opposed to general sorts of actions by insiders. It supports the placing of sensors into a heterogeneous collection of applications including editors (e.g., MS Office), browsers (e.g., Internet Explorer), domain specific applications, and operating systems. Its architecture is similar to the Willow architecture but the kinds of sensors, analyzers, and actuators differ.

On the analysis side, the goal of EventTrails is to support a dynamically changing set of analysis tools capable of providing such diverse capabilities as visualization, persistence, forensics, and automated detection of insider misuse signals. When misuse is detected, the system provides a graduated sequence of responses.

EventTrails is starting to make novel use of hypermedia technology to capture generated events for use by misuse detection tools and by security administrators. It provides persistent storage for captured events and organizes events into hierarchical trails that support analysis at different levels of granularity. It also supports queries against particular event patterns, which is useful in defining signatures for suspicious activities. Analysts can also make use of a browser to visually analyze the contents of an event trail, to compare trails “side-by-side”, and to examine the activities of a specific user or document.

Automated detection is supported by the use of probabilistic classification models. Classifiers act on features extracted from events in real-time. Multiple classifiers will be used to track activity based on user roles and with respect to document types. If a particular user falls under suspicion, classifiers can be dynamically invoked to monitor his/her activities. Results of classifiers

will be delivered to security personnel for feedback; this feedback will be used to further train the classifiers to reduce false positives.

2.3 Terminator

The Terminator project [3] is a new adaptive security system that targets the problem of revoking selected security privileges of users. It builds off both Willow and EventTrails, and it has a similar architecture.

When any employee leaves an organization, it creates the potential for a breach of security. The employee will have knowledge of passwords and access rights to multiple resources. It is difficult and tedious for security administrators to make sure that all permissions for a terminated employee are removed at the time of termination. The fundamental problem is that security-related information such as policies, authority grants, passwords, and access lists are distributed across multiple domains and across multiple machines in a network. Further, there may be many informal paths by which access is granted but not recorded. It is not uncommon for people to share passwords without bothering to go through any formal mechanism.

The terminator infrastructure is novel in that it is based on the use of sensors, but the sensors are intended to be inserted into the security infrastructure itself to record security related events such as accesses to resources or uses of passwords. The actuation mechanisms are also directed at security systems and perform actions such as remote changing of passwords or remove removal of access rights.

Although in its earliest stages, Terminator will utilize a variant of the EventTrails event capturing database. The database must record the actual resources accessed by the user. This includes such things as specific passwords used, specific policy changes (i.e., setting access rights), and specific files read or written. Second, the database must represent the complete set of resources known to be accessible to each user.

3. Common Issues

In the course of developing a number of adaptive security systems, it has become clear that there are a number of common issues for which further research is required. It is probable that these same issues are important in varying degrees for remote analysis. The next sections discuss some of these common issues.

3.1 Software Engineering for Sensors

Sensors are the key to all sense-analyze-respond systems. But existing software systems have only limited ability to generate events about their internal state.

Microsoft operating systems provide some ability to insert sensors into the file system, and many applications provide some sensors via the use of COM interfaces. It is also possible to gain additional information by wrapping shared libraries (DLLs) and by trapping the user interface event system.

It is possible to insert arbitrary sensors into open source operating systems such as Linux or FreeBSD because the source is freely available. Linux has gone a step further by defining the Linux Security Module interface [6] to support controlled insertion of sensors for a variety of operating system level activities. Unfortunately, very few Unix-based applications support any sort of ability to attach sensors except again when their source is available.

An important research topic, then, would involve the development of general software engineering principles, processes, and architectures that support the insertion of sensors into applications.

3.2 Dissemination

Willow, EventTrails, and Terminator currently all exploit a common infrastructure for collecting events from sensors and distributing them where required. That infrastructure is the University of Colorado Siena [2] scalable content-based routing technology. Siena should be useful in the remote analysis community as well. Siena is not, however, designed specifically for this purpose, and it seems possible to build a simpler and more secure infrastructure for collecting events and for distributing responses. Further, this infrastructure must itself be highly secure because the events it distributes have potentially sensitive information. Again, this new dissemination infrastructure should be useful for both communities.

3.3 Database Support

Both adaptive security systems and remote analysis systems have a need to manage and view large numbers of events. The database, or whatever is used, that holds these events needs to be augmented with viewers that can slice

data and aggregate it to present interesting views of the data. It seems probable that viewers developed for one community may be usefully applied in the other.

4. Conclusion

Adaptive security and remote analysis have many features in common, both operationally and architecturally. Joint efforts would be welcome to develop common infrastructure for inserting sensors, for distributing events, and for managing and viewing large numbers of events.

5. References

- [1] Anderson, K., A. Carzaniga, D. Heimbigner, and A.L. Wolf, "Event-based Document Sensing for Insider Threats," University of Colorado, Computer Science Technical Report CUCS-968-04, February 2004.
- [2] Antonio Carzaniga, David S. Rosenblum, and Alexander L. Wolf, *Design and evaluation of a wide-area event notification service*, ACM Transactions on Computer Systems, 19(3):332–383, August 2001.
- [3] Heimbigner, D., "Managing Access Rights for Terminated Employees," Proc. 2004 Security and Management Conference, Las Vegas, NV, 21-24 June 2004.
- [4] Knight, J., D. Heimbigner, A. Wolf, A. Carzaniga, J. Hill, and P. Devanbu, "The Willow Survivability Architecture," Proc. of the Fourth Information Survivability Workshop, Vancouver, B.C, March 2001.
- [5] Wolf, A.L., D. Heimbigner, J. Knight, P. Devanbu, M. Gertz, and A. Carzaniga, "Bend, Don't Break: Using Reconfiguration to Achieve Survivability," Proc. of the Third Information Survivability Workshop, Boston, Mass., October 2000.
- [6] Wright C., C. Cowan, S. Smalley, J. Morris, G. Kroah-Hartman, "Linux Security Modules: General Security Support for the Linux Kernel," 11th Usenix Security Symposium, 5-9 August 2002, San Francisco, CA.