# Designing and Implementing a Family of Intrusion Detection Systems

**Richard A. Kemmerer**
Reliable Software Group
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106
USA

kemm@cs.ucsb.edu

Intrusion detection systems (IDSs) analyze information about the activities performed in a computer system or network, looking for evidence of malicious behavior. Attacks against a system manifest themselves in terms of events. These events can be of a different nature and level of granularity. For example, they may be represented by network packets, operating system calls, audit records produced by the operating system auditing facilities, or log messages produced by applications. The goal of intrusion detection systems is to analyze one or more event streams and identify manifestations of attacks.

The intrusion detection community has developed a number of different tools that perform intrusion detection in particular domains (e.g., hosts or networks), in specific environments (e.g., Windows NT or Solaris), and at different levels of abstraction (e.g., kernel-level tools and alert correlation systems). These tools suffer from two main limitations: they are developed *ad hoc* for certain types of domains and/or environments, and they are difficult to configure, extend, and control remotely.

In the specific case of signature-based intrusion detection systems the sensors are equipped with a number of attack models that are matched against a stream of incoming events. The attack models are described using an *ad hoc*, domain-specific language (e.g., N-code, which is the language used by the Network Flight Recorder intrusion detection system). Therefore, performing intrusion detection in a new environment requires the development of both a new system and a new attack modeling language. As intrusion detection is applied to new and previously unforeseen domains, this approach results in increased development effort.

Today’s network are not only heterogeneous, but also dynamic. Therefore, intrusion detection systems need to support mechanisms to dynamically change their configuration as the security state of the protected system evolves. Most existing intrusion detection systems are initialized with a set of signatures at startup time. Updating the signature set requires stopping the IDS, adding new signatures, and then restarting execution. Some of these systems provide a way to enable/disable some of the available signatures, but few systems allow for the dynamic inclusion of new signatures at execution time. In addition, the *ad hoc* nature of existing IDSs does not allow one to dynamically configure a running sensor so that a new event stream can be used as input for the security analysis.

Another limitation of existing IDSs is the relatively static configuration of responses. Normally it is possible to choose only from a specific subset of possible responses. In addition, to our knowledge, none of the systems allows one to associate a response with *intermediate* steps of an attack. This is a severe limitation, especially in the case of distributed attacks carried out over a long time span.

Finally, the configuration of existing IDSs is mostly performed manually and at a very low level. This task is particularly error-prone, especially if the intrusion detection systems are deployed across a very heterogeneous environment and with very different configurations.

This talk describes a framework for the development of intrusion detection systems, called STAT, that overcomes these limitations. The STAT framework includes a domain-independent attack modeling language and a domain-independent event processing analysis engine. The framework can be extended in a well-defined way to match new domains, new event sources, and new responses. The resulting set of applications is a software family whose members share a number of features, including dynamic reconfigurability and a fine-grained control over a wide range of characteristics. The main advantage of this approach is the limited development effort and the increased reuse that result from using an object-oriented framework and a component-based approach.

STAT is both unique and novel. First, STAT is the only known framework-based approach to the development of intrusion detection systems. Second, even though the use of frameworks to develop families of systems is a well-known approach, the STAT framework is novel in the fact that the framework extension process includes, as a by-product, the generation of an attack modeling language closely tailored to the target environment. This talk focuses primarily on the STAT framework.

*Paper presented at the RTO IST Symposium on “Adaptive Defence in Unclassified Networks”,*
*held in Toulouse, France, 19 - 20 April 2004, and published in RTO-MP-IST-041.*