

Talk title:

Design and implementation issues for embedded sensors in intrusion detection

Suggested topic category:

Innovative approaches / New IDS methodologies and technologies

Authors:

Eugene H. Spafford
Director,
CERIAS, Purdue University
spaf@cerias.purdue.edu

Diego Zamboni
Graduate student
CERIAS, Purdue University
zamboni@cerias.purdue.edu

Mailing address for both authors:

1315 Recitation Building
Purdue University
West Lafayette, IN 47907-1315
U. S. A.

Extended abstract:

In this talk we will discuss the implementation of an architecture for performing intrusion detection using sensors embedded in computer systems. We will also discuss our initial results. More details are given below.

We propose the use of embedded sensors to monitor a system and to perform intrusion detection. We define a sensor as a piece of code that is added to the code of a program and monitors a specific variable, activity or condition. For example, the following section of code may be subject to a buffer-overflow attack through the HOME environment variable:

```
char buf[256];  
...  
strcpy(buf, getenv("HOME"));  
...
```

After inserting a sensor to detect this problem, the code may look like this (new lines prepended by "|"):

```
char buf[256];  
...
```

```
| { if (strlen(getenv("HOME"))>255) {  
|     log_alert("buffer overflow");  
| }  
| }  
    strcpy(buf, getenv("HOME"));  
    ...
```

Sensors can be embedded in any program that could be the target of an attack or present a vulnerability, or from which the attack could be detected. This includes the Unix kernel and any of its subsystems and utilities. The sensors monitor the host directly and not through an audit trail or other indirect means, therefore we say that they perform "target monitoring".

In our approach, sensors are initially implemented as detectors for specific intrusions or attacks. However, we use a process in which sensors can later be modified, fused or split in an informed fashion to give them better placement, structure or function. In this way, sensors are implemented as a unified group and not as disconnected units. The final objective is to have a large number of sensors that provide data that can be used to detect both known and new intrusions.

Sensors are intended to be as small and efficient as possible and to detect and log specific conditions, rather than general events. We have also developed special logging mechanisms for use by the sensors when they need to report something, instead of using a built-in mechanism such as syslog. Using a different logging mechanism is useful to keep the information from the sensors separate from general system audit data, to be able to optimize the logging mechanism as much as possible, and to have the possibility of incorporating other auxiliary features (for example, message rate limitation to protect against attacks on the intrusion detection system itself) without affecting general system logging.

Because sensors can be designed to detect attack attempts and not only their success, the attacks they detect do not need to exist in the host where they are implemented, and in fact could be attacks for a completely different platform. In this sense, a system instrumented with embedded sensors could work like a "universal honeypot" because it will be able to accept and monitor attacks for different platforms.

We have started implementing sensors on the OpenBSD operating system. This system was selected because its source code is available and centrally managed, it has a stable code base, and has gone through an extensive security-oriented code audit.

As a source of initial ideas for sensors to implement, we are using the Common Vulnerability and Enumeration (CVE) list from MITRE. The CVE is not a taxonomy or a classification scheme, and is used only as a fairly complete list of known vulnerabilities and attacks. By going through the list and implementing sensors to detect each entry, we can cover a wide variety of types of security problems.

The embedded sensors architecture was born from the experience obtained with the AAFID (Autonomous Agents for Intrusion Detection) architecture, to solve some of its limitations. However, embedded sensors are not meant as a complete replacement for

AAFID, but as a powerful complement in the collection and analysis of information.

In this talk we will describe our sensor infrastructure, the selection of the implementation platform, the facilities implemented for building the sensors, some of the sensors that have been implemented and how some sensors have "evolved" during the implementation process to provide better data for intrusion detection. We will also talk about design and implementation issues encountered, and discuss our initial results in terms of detection capabilities (of both old and new problems) and performance impact on the host. Finally, we will present our view for the future of embedded sensors, and how their use could benefit existing intrusion detection systems.