Building an Intrusion-Detection System to Detect Suspicious Process Behavior

Type of submission

Paper

Topic category

Innovative Approaches / Assessing IDS

Authors

Andreas Wespi and Hervé Debar

Corresponding author / speaker

Andreas Wespi
Mail: anw@zurich.ibm.com
Phone: +41 1 724 82 64
Fax: +41 1 724 89 53

Institution

IBM Zurich Research Laboratory
Global Security Analysis Lab
Säumerstrasse 4
8803 Rüschlikon
Switzerland

Biography

Andreas Wespi

Andreas Wespi is a research scientist at the IBM Zurich Research Laboratory in the Information Technology Solutions department. He holds an M. Sc. in Computer Science from the University of Berne, Switzerland. His research interests include intrusion detection, network security in general, and distributed and parallel computing.

Hervé Debar

Hervé Debar is a research scientist in the global security analysis laboratory at the IBM Zurich Research Laboratory, where he works on system and network security (in particular intrusion detection) as well as system management. His interests include secure systems and artificial intelligence. Dr. Debar holds a Ph.D. from the University of Paris, France, and a Telecommunications Engineering degree from the Institut National des Télécommunications in Evry (France).

Building an Intrusion-Detection System to Detect Suspicious Process Behavior

Abstract

As has been shown in S. Forrest's seminal work [1], there are Unix processes whose normal behavior can be modeled by a set of characteristic patterns, a pattern being a subsequence of system calls that a process can generate. Well-suited processes are network services such as ftpd or sendmail. Intrusion-detection systems that make use of this observation first need to build the table of characteristic patterns. The patterns are determined by letting the process invoke as many subcommands as possible, then extracting the patterns from the corresponding sequences of system calls. During real-time operation, a pattern-matching algorithm is applied to match on the fly the system calls generated by the process examined with entries of the pattern table. Based on how well the matching can be done, it is decided whether the sequence of system calls represents normal or anomalous behavior.

As an analysis of intrusion-detection systems that are based on the above concepts [1,2,3] reveals, these systems share the same main components, but they differ in the way the individual components are implemented. Furthermore, each system uses some implicit parameters that are based on experimental observations and may only be applicable to the test cases examined. In our presentation, we will give an overview of the various techniques to implement the individual components, and assess their impact on the detection capability of the intrusion-detection system.

Specifically, we will discuss the following components:

- **Recording system calls:** System calls can be recorded either by using a tool such as ``trace'' or by activating the auditing system. The technique used may influence the overall system performance and the number of events recorded.
- **Generating training data:** The intrusion-detection system needs to be trained to learn what ``normal'' behavior is. We differentiate between ``synthetic'' and ``real'' normal behavior. Synthetic normal behavior is generated by exercising a program in as many modes as possible and tracing its behavior. Real normal behavior corresponds to tracing the behavior of a program in a live user environment.
- **Building the process model:** The process model is a pattern table that consists of either fixed- or variable-length patterns. While the creation of the fixed-length pattern table is quite straightforward, various techniques can be used to build a table of variable-length patterns.
- **Comparing real process data with the process model:** The goal of the detection component is to differentiate between normal and anomalous behavior. For doing so, the sequences of system calls generated by a process in real operation is

matched with entries of the pattern table. Basically one can differentiate between two pattern-matching techniques: overlapping and juxtaposed pattern matching.

?? **Detecting attacks:** The number and the alignment of the unmatched events are the two main criteria used to differentiate between normal and anomalous behavior.

The above components are not separate entities. They are interrelated, and the decision to implement a component in a certain way may influence the design choices available for other components. We will show these dependencies, and assess the advantages and disadvantages of the various implementations of the above components based on a series of experiments. The experiments were performed in a testbed environment [4] built specifically to compare the kind of intrusion-detection systems we are investigating.

References

[1] S. Forrest, S.A. Hofmeyr, A. Somayaji, and T.A. Longstaff, ``A Sense of Self for Unix Processes," Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy, pp. 120-128, IEEE Computer Society Press, Los Alamitos, CA, May 6 - 8, 1996.

[2] H. Debar, M. Dacier, M. Nassehi, and A. Wespi, ``Fixed vs. Variable-Length Patterns for Detecting Suspicious Process Behavior," Esorics '98, 5th European Symposium on Research in Computer Security, pp. 1-15, Louvain-la-Neuve, Belgium, Sep. 16 - 18, 1998.

[3] A. Wespi, H. Debar, and M. Dacier, ``An Intrusion-Detection System Based on the Teiresias Pattern-Discovery Algorithm," Eicar '99, Aalborg, Denmark, Feb. 27 - Mar. 2, 1999.

[4] H. Debar, M. Dacier, A. Wespi, and S. Lampart, ``A Workbench for Intrusion Detection Systems," IBM Research Report RZ 2998, IBM Research Division, March 1998.