International Journal of Electronics Communication and Computer Engineering
Volume 5, Issue 2, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Application Intrusion Detection Systems: The Next Step

Mr. Umapathy Balasubramanian
Research Scholar
E-mail: vasanth2k4@gmail.com

Dr. K. Krishnamoorthy
Professor, Sudharsan Engineering College,
Pudukkottai-622501, Tamilnadu
E-mail: kkr_510@yahoo.co.in

***Abstract* – Operating system intrusion detection systems (OS IDS) are frequently insufficient to catch internal intruders who neither significantly deviate from expected behavior nor perform a sequence of specific intrusive actions. We hypothesize that application intrusion detection systems (AppIDS) can use application semantics to detect more subtle attacks such as those carried out by internal intruders who possess legitimate access to the system and act within their bounds of normal behavior, but who are actually abusing the system. To test this hypothesis, we developed two extensive case studies from which we were able to discern some similarities and differences between the OS IDS and AppIDS. In particular, an AppIDS can observe the monitored system with a higher resolution of observable entities than an OS IDS allowing tighter thresholds to be set for the AppIDS' relations that differentiate normal and anomalous behavior thereby improving the overall effectiveness of the IDS.**

***Keywords* – Intrusion Detection Systems, AppIDS, OS IDS.**

I. INTRODUCTION

As information systems have become more comprehensive and a higher value asset of organizations, *intrusion detection systems* have been incorporated as elements of operating systems, although not typically applications. *Intrusion detection* involves determining that some entity, an *intruder*, has attempted to gain, or worse, has gained unauthorized access to the system. Intruders are classified into two groups. *External intruders* do not have any authorized access to the system they attack. *Internal intruders* have at least some authorized access to the system. Internal intruders are further subdivided into the following three categories. *Masqueraders* are external intruders who have succeeded in gaining access to the system and are acting as an authorized entity. *Legitimate* intruders have access to both the system and the data but misuse this access (misfeasors). *Clandestine* intruders have or have obtained supervisory (root) control of the system and as such can either operate below the level of auditing or can use the privileges to avoid being audited by stopping, modifying, or erasing the audit records [Anderson80].

Internal intruders are said to comprise at least fifty percent of intruders [ODS99], but OS intrusion detection systems are frequently insufficient to catch such intruders since they neither perform the specific intrusive actions because they are already legitimate users of the system, nor significantly deviate from expected behavior.

Intrusion Detection Approaches

Currently there are two basic approaches to intrusion detection. The first approach, *anomaly detection*, attempts to define and characterize correct static form of data and/or acceptable dynamic behavior. In effect, it searches for an anomaly in either stored data or in the system activity. IDS utilizing anomaly detection include Tripwire [Kim93], Self-Nonself [Forrest94], and NIDES [Anderson95].

Intrusion detection systems have been built to explore both approaches: anomaly detection and misuse detection. In some cases, they are combined in a complementary way in a single intrusion detector. There is a consensus in the community that both approaches continue to have value. Systems also apply these same approaches to detect intrusions across a network of computers. Representative systems include NADIR [Hochberg93], NSTAT [Kemmerer97], GrIDS [Staniford-Chen96], and EMERALD [Porras97].

Health Record Management

The second case study is of a hypothetical health record management (HRM) system that includes patient records, orders for drugs, tests, or procedures, and the schedules for filling these orders. This system is representative of a non-hierarchical information collecting and scheduling application that exists in many businesses.

Dependencies between OS IDS and AppIDS

OS IDS have been deployed for years without AppIDS, so OS IDS are certainly not dependent on AppIDS. Because all applications run on top of operating systems, the AppIDS will rely on the OS IDS to provide some basic security services such as access control protection of the application's code, data, and communications. Typically, only the OS IDS can detect and protect against the external intruders. An AppIDS may be able to detect an external intruder but only after the intruder gains access to the system at which point the intruder is reclassified as an internal intruder.

Cooperation between OS IDS and AppIDS

If both OS IDS and AppIDS exist for a system and an application, it is logical to explore how the two IDS could cooperate to improve the overall effectiveness of detecting intrusions. For example, an AppIDS may detect a manipulation intrusion but cannot determine the identification of the intruder. The OS IDS may be able to identify the intruder. Conversely, an OS IDS may detect that an application has had a sudden increase in the number of files created. However, the OS IDS cannot determine whether this should be happening, but the AppIDS may be able to decide because it has the additional information of the application semantics. Although the communication between the two with bi-directional information flows seems simple enough, there are some complications. The application and the OS operate at different semantic levels, so they must be able to communicate in terms that both can understand. Without being able to express requests and interpret

responses from the other domain, communications between OS IDS and AppIDS would be fruitless. The lowest common denominator is resource usage, so this appears to be the simplest starting point for developing the communication interface.

III. CONCLUSIONS

By exploring the possibilities of intrusion detection at the application level, we hoped to explore the opportunities and limits of utilizing application semantics to perform intrusion detection. From reviewing the state of practice for OS IDS and two extensive case studies, we determined that application semantics can be used to detect internal intruders of an application using similar techniques to those employed by OS IDS. We found that the AppIDS will be mostly concerned with anomaly detection but use both forms of relations, statistical and rule-based, to detect intruders. Although the threats may be similar for both IDS, the higher resolution of observable entities with which the AppIDS operates allows it to detect intrusions that the OS IDS cannot detect. Because the AppIDS relies on the OS IDS for some basic protection and only the OS IDS can detect the external intruders, it is not reasonable to replace and OS IDS with an AppIDS, only to augment the OS IDS. If both IDS are present, there is potential for the two to cooperate in detecting some intrusions using a bi-directional communication interface.

Little research has been done into performing intrusion detection at the application level, so there is a general lack of literature on the subject. The construction of an actual AppIDS would also be beneficial in furthering these conclusions. A possible structure for constructing multiple AppIDS has been discussed elsewhere [Sielken99]. We have shown that the approach of detecting intrusions at the application level appears to be fruitful. The next step is an actual implementation of an AppIDS for representative applications.

REFERENCES

[1] [Anderson80] Anderson, J.P. "Computer Security Threat Monitoring and Surveillance." Technical Report, James P. Anderson Co., Fort Washington, Pennsylvania, April 1980.

[2] [Anderson95] Anderson, D., T. Frivold and A. Valdes. "Next-generation Intrusion Detection Expert System (NIDES): A Summary." SRI International Computer Science Laboratory Technical Report SRI-CSL-95-07, May 1995.

[3] [Forrest94] Forrest, S., L. Allen, A.S. Perelson, and R. Cherukuri. "Self-Nonself Discrimination in a Computer." *Proceedings of the 1994 IEEE Symposium on Researh in Security and Privacy*, 1994.

[4] [Hochberg93] Hochberg, J., K. Jackson, C.Stallings, J.F. McClary, D. DuBois, and J. Ford. "NADIR: An Automated System for Detecting Network Intrusion and Misuse." *Computers and Security*, 12.3 (1993) 235-248, http://nadir.lanl.gov/libLA-UR-93-137.html.

[5] [Kemmerer97] Kemmerer, R.A. "NSTAT: A Model-based Real-time Network Intrusion Detection System." *University of California-Santa Barbara Technical Report TRCS97-18*, November 1997.

[6] [Kim93] Kim, G.H. and E.H. Spafford. "A Design and Implementation of Tripwire: A File System Integrity Checker." *Purdue Technical Report CSD-TR-93-071*, November 1993.

[7] [ODS99] ODS Networks, Inc. "Extreme Access . . . Infinite Possibilities." *ODS Networks White Paper*, March 1999, http://www.ods.com/white/whi_0004.shtml.

[8] [Porras92] Porras, P.A. and R.A. Kemmerer. "Penetration State Transition Analysis: A Rule-Based Intrusion Detection Approach." *Proceedings of the Eighth Annual Computer Security Applications Conference*, December 1992.

[9] [Porras97] Porras, P.A. and P.G. Neumann. "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances." *19th National Information System Security Conference (NISSC)*, 1997, http://www.csl.sri.com/emerald/emerald-niss97.html.

[10] [Sebring88] Sebring, M.M., E. Shellhouse, M. Hanna and R. Whitehurst. "Expert Systems in Intrusion Detection: A Case Study." *Proceedings of the 11th National Computer Security Conference*, October 1988.

[11] [Sielken99] Sielken, Robert S. "Application Intrusion Detection." *University of Virginia Computer Science epartment Technical Report CS-99-17*, June 1999.

[12] [Staniford-Chen96] Staniford-Chen, S., S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, and D. Zerkle. "GrIDS – A Graph Based Intrusion Detection System for Large Networks." *20th National Information System Security Conference (NISSC)*, October 1996, http://olympus.cs.ucdavis.edu/arpa/grids/nissc96.ps.