A Layered Approach to Insider Threat Detection and Proactive Forensics

Phillip G. Bradford* Ning Hu*

An insider threat is a menace to computer security as a result of unauthorized system misuse by users of an organization. A recent study jointly published by the United States Secret Service and Carnegie Mellon University [7] confirms the prevalence of computer crimes perpetrated by insiders across America's organizations. Insider attacks can be more destructive and costly than attacks from the outside as a perpetrator often has deep understanding of and convenient accesses to a plethora of an organization's computer resources. This paper discusses augmenting intrusion detection systems with forensics tools to enhance the discovery and prosecution of internal attacks. Our research follows two approaches: One is using intrusion detection systems (IDSs) *[5] as black boxes and having them drive forensics tools. Likewise, we are looking at building our own statistical metrics for fleshing out long term changes in user behavior.*

Given their pervasive and destructive nature, early detection and documentation of *insider* breaches is unarguably vital to stakeholders' interests. Unfortunately, IDSs alone may not be suitable for this challenge for the following reasons:

- Detection accuracy: Either an IDS may overlook a real threat (false negative) or issue alerts about a normal event (false positive). *Signature-based ID (SID)* identifies known intrusive patterns, but fails to capture novel behavior. *Anomaly-based ID (AID)* is susceptible to false alarms due to the difficulty of distinguishing an attack from numerous anomalies. False negatives are obviously more detrimental than false positives because of their destructive nature. False positives are less urgent concerns but no less troublesome as they waste the resources needed for their verification. For example, a false positive may prompt an IDS to collect useless data on an apparently benign user over the long term.
- Costs: Lee, et al. [10] distinguish ID-related costs into: damage cost, response cost and operational cost. Gathering forensics data on suspected users will be costly and we do not want it to disturb or be intrusive for the user. It is an art to balance the trade-off between detection cost and accuracy. For example, system call tracing is a popular strategy, but can be very expensive, and thus needs to be used with discretion. Ideally, an IDS should consume the least possible amount of system resources on the host or network it monitors. Likewise, stealth data collection and application of forensics tools should be as undetectable as possible. Its operational cost ought to be minimized so as not to impact normal system usage yet achieve the desired results.
- Applicability: Generally speaking, the IDS is intended to provide a second line of defense behind a firewall against attacks stemming from outside an organization's computer network. Therefore, popular IDS techniques, such as vulnerability scans, network traffic sniffing, etc., have been more successfully applied to the detection of intrusion, rather than *extrusion* (which is unauthorized transfer of an organization's crucial computer assets as cause by a trusted insider, human error or criminal [8]). Insider threats are characterized by distinct qualities, patterns and assumptions (see [1] for a comprehensive taxonomy). For example, a malicious insider tends to possess a better knowledge of and more convenient access to the target than an external adversary. To account for such differences, efforts must be exerted to invent new or adapt old techniques for insider threat detection.

Furthermore, since proactive security is imperfect and fallible, a practical computer security system should have the capability to collect and retain important evidence that can be accessed later for examination and prosecution. It is a combination of these thoughts that gives rise to our ongoing efforts in integrating characteristics of digital forensics into intrusion detection. This combination is known as *Proactive Forensics*. This term has been used by some commercial computer forensics system vendors, e.g. Guidance Software, to mean the process of investigating surreptitious computer and network activities *proactively*. We extend this definition into the "design, construction and configuring of (computer) systems to make them most

[1]**The University of Alabama**, Department of Computer Science, Box 870290, Tuscaloosa, AL 35487-0290. **pgb@cs.ua.edu, nhu@cs.ua.edu**

amenable to digital forensics analyses in the future" [2]. Our primary focus is host-level computer forensics as we are interested in examining user activities on workstations.

Farmer and Venema [6] distinguish digital forensic analysis into three operational styles, i.e. static analysis, dynamic analysis and post-mortem analysis. Considering efficiency and accuracy, dynamic analysis is the most promising as it is generally performed in real time on an online computer system. Tools of this class are abundant. To mention just a few, *Lsof* [11] is a classic Unix-based diagnostic tool for listing information about any files that are currently open by running processes, and about any communications open by each process. *Systrace* [12] is another Unix program that supports the tracing of system calls as they cross the boundary between the kernel and user space. Along with online detection and analysis, we are also interested in identifying the information (evidence) that is potentially pertinent to future investigation. Information of this type can be collected proactively and later examined by some static, or post-mortem analysis utility, such as *The Coroner's Toolkit (TCT)* [13].

Our ultimate goal is to develop a system for online monitoring of user activities to detect potential system misuses and abuses, and for proactive collection and analysis of important evidence about security incidents stemming from insider threats. Monitoring and understanding user behavior is a complicated operation as so many interacting factors, e.g. systems, files, processes, memory, user actions, habits, roles, etc., need to be considered. Currently, we restrict our attention to user processes, since much can be learned about a user's activities by examining her processes. We are interested in investigating the possibility of cost reduction in carrying out dynamic analysis at the process level. We will follow a *black-box approach* [6] to understanding a user process by analyzing its inputs and outputs and their relationships without knowledge of the process' internal structure.

An important feature of our model is that a user under analysis is not treated as an isolated case, but as a member of one or more user groups. This approach hinges on the assumption that users in similar job functions tend to exhibit similar system usage patterns [2, 3]. These anticipated patterns can be leveraged to detect any deviations in individiual users' behavior. For example, financial analysts are expected to use certain business applications for most of their working day, but rarely use other software utilities, such as a P2P file sharing program. Subsequently, the recurrence of deviations observed in one of the analysts' processes is a reasonable concern for the computer security expert. We name this observation the *small-user-world principle* [2].

To improve detection accuracy and potentially decrease operational cost, we propose a layered architecture in implementing a Proactive Process Monitor (PPM). In the current design, the prototype comprises three detection levels:

- The top layer compiles a growing name list of confirmed malicious or unauthorized processes for all users. The primary interest of this layer is to quickly identify many common misuse processes running on user hosts. Once a process is matched by name, its detailed information can be logged at a separate secure site for off-line analysis by a forensic tool. The process will be ruled out of further examination in the subsequent layers. Admittedly, a cunning attacker can easily circumvent this layer by disguising the name of an illegitimate process. Since this layer operates by a straightforward name-matching technique, its resource commitment is kept at a minimum.

- The middle layer utilizes a pre-defined rule base to capture the unauthorized processes associated with particular user roles, which have eluded detection from the top layer. The rule base consists of rules that can be leveraged to identify potential threats from unexpected correlations of user roles and process patterns. The original rules are derived from mapping user roles to expected/permitted process usage in accordance with the target organization's specific security policies. The context of a user executing a process is important to a correct mapping of roles to actions. For example, if a user belongs to several user groups, she may exhibit a more diversified and complex process pattern than single-role users. A Genetic Algorithm (GA) is applied to constantly optimize this rule set as new misuse patterns are reported and added to the rule set. A detected process is recorded in the same way as in the top layer to be made accessible to forensic analysis. The middle layer consumes a reasonable amount of computational resource without recourse to an indiscriminate system call tracing approach.

- At the bottom layer, the remaining processes are further examined by a statistical analyzer for any "low-and-slow" shifts toward anomalous behavior. The analyzer employs a set of process metrics, e.g. system calls, CPU usage, memory usage, process elapse time, etc. We designed the statistical analysis in keeping with the afore-mentioned *small-user-world principle*: a user's process metrics are continuously monitored and compared with those of his peers. The analyzer logs those processes that it considers necessary for offline forensic analysis. These can either be suspicious

processes showing novel anomalous patterns, or processes for which the cost of online analysis exceeds the estimated damage cost [10].

The major challenge facing this design is whether we can reduce false negatives to an acceptable threshold with reasonable cost. False positives are a secondary concern here as already mentioned. To improve detection results, we draw on the theoretical model proposed by Li, et al. [9]. In essence, the ID model evolves three feature ranges labeled "normal", "anomalous" and "suspicious", respectively. The suspicious range encompasses features that appear in both normal and anomalous ranges, and produces most of the false alarms. To apply the model to our work, however, much remains to be done as to the selection of appropriate features for the classification of user processes.

Although one of our primary approaches is to harness IDSs as black-boxes, we are also interested in the analysis based on slowly gathering evidence of potential perpetrators. On this front, so far we have implemented a statistical process analysis prototype to monitor and analyze the deviations from both a user's individual and group patterns. Statistical approaches are classical in intrusion detection research, which was outlined by Denning [4] nearly two decades ago. Similar to Denning, we apply *sequential hypothesis testing* [14] to insider threat detection. This testing method can be applied to a sequence of ordered samples to calculate probability ratios during the test of a hypothesis, e.g. a process is anomalous. We hope that this test is useful to detect some of a user's deviations from her group patterns. To measure a user's deviations from her individual pattern we keep track of the *moving averages* of her process metrics.

Statistical analysis can be leveraged in generating initial auditing reports on suspicious user behavior. Admittedly, our current selection of process variables is coarse-grained and prone to false alarms. We are interested in investigating more stringent metrics to refine detection results. Eventually, our work will lead to the establishment of a set of appropriate metrics that can be monitored to invoke and assist with forensic tools.

References

[1] R. Brackney and R. Anderson, "Understanding the Insider Threat," Proc. of an ARDA Workshop, March 2004, Rockville, MD.

[2] P. G. Bradford, M. Brown, J. Perdue, and B. Self, "Towards Proactive Computer-System Forensics," Proc. of International Conference on Information Technology: Coding and Computing, Vol. 2, 2004 (ITCC 2004), 648-652.

[3] R. DelZoppo, E. Browns, M. Downey, E. D. Liddy, S. Symonenko, J. S. Park, S. M. Ho, M. D. Eredita, and A. Natarajan, "A Multi-Disciplinary Approach for Countering Insider Threats," Proc. of the Workshop on Secure Knowledge Management (SKM), Amherst, NY, September 23-24, 2004.

[4] Dorothy E. Denning: "An Intrusion-Detection Model," IEEE Transactions on Software Engineering, Vol. SE-13, No. 2, Feb. 1987, 222-232.

[5] H. Debar, M. Dacier, and A. Wespi, "A Revised Taxonomy of Intrusion-Detection Systems," Annales des Telecommunications, 55 (7-8), 2000, 83-100.

[6] D. Farmer and W. Venema, Forensic Discovery, Pearson Education, Inc., 2005.

[7] M. Keeney, E. Kowalski, D. Cappelli, A. Moore, T. Shimeall, and S. Rogers, "Insider Threat Study: Computer System Sabotage in Critical Infrastructure Sectors," May 2005. http://www.secretservice.gov/ntac/its_report_050516.pdf

[8] R. I. Koenig, "Be Aware of Insider Threats to Your Security," CyberDefense Magazine, August 2004, 52-53.

[9] Z. Li, A. Das, and J. Zhou, "Theoretical Basis for Intrusion Detection," Proc. of 6th IEEE Information Assurance Workshop (IAW), June 15-17, 2005, West Point, NY.

[10] W. Lee, W. Fan, M. Miller, S. Stolfo, and E. Zadok, "Toward Cost-Sensitive Modeling for Intrusion Detection and Response," Journal of Computer Security, Vol. 10, Numbers 1,2, 2002.

[11] V. Abell, Lsof, http://freshmeat.net/projects/lsof/

[12] N. Provos, "Improving Host Security with System Call Policies," Proc. of the 12th USENIX Security Symposium, 2003.

[13] D. Farmer, and W. Venema, The Coroner's Toolkit (TCT), 2004, http://www.porcupine.org/tct.html

[14] A. Wald, Sequential Analysis, New York, Wiley and Sons, 1947.