Detecting the Misappropriation of Sensitive Information through Bottleneck Monitoring

Terrance Goan
Stottler Henke Associates, Inc.
1107 NE 45th St, Suite 310
Seattle, WA 98005
goan@stottlerhenke.com

Matthew Broadhead
Stottler Henke Associates, Inc.
1107 NE 45th St, Suite 310
Seattle, WA 98005
broadhead@stottlerhenke.com

Keywords: Intrusion Detection, Malicious Insiders, Digital Rights Management, Bottleneck Monitoring, Content Analysis

Abstract

The insider threat has proved a tough nut to crack. Previous work in this area has been dominated by efforts to model normal user behavior through statistical measures and then detect substantial anomalies. Unfortunately, while these methods have shown some ability in the detection of masqueraders, broader applications have proved ineffectual due to extremely high false alarm rates. In this paper we describe an alternative approach (SL-SAFE) that can achieve high levels of accuracy in detecting the unauthorized access and distribution of sensitive/proprietary information by insiders – the single most costly type of computer crime. SL-SAFE succeeds in this task by means of a stochastic sampling of bottlenecks through which information must flow in order to be useful to the malicious insider. Further, it achieves a low (and shrinking) false alarm rate by validating its suspicions through public information sources and eliciting feedback from the information owner.

1. INTRODUCTION

It is interesting that while public attention to information security focuses primarily on the often indiscriminant hostile acts of "outsiders," it is widely recognized in the computer security community that the trusted (but untrustworthy) "insider" represents a far greater threat. For instance, a joint Computer Security Institute and FBI study indicated that 80 percent of respondents reported insider abuse, and that attacks (theft of proprietary information) launched by insiders were far more costly—$2.7 million vs. $50,000 for the average attack [4]. In an independent study, the Gartner Group estimated that insiders cause 70 percent of the "cyber" attacks that cost the victim $20,000 or more [6]. And of course insiders also pose very real risks outside the private sector as witnessed by the fatal consequences of the acts of the convicted spies Robert Hanssen and Aldrich Ames.

But despite these clear risks, the insider threat remains largely unaddressed – with the primary network defenses being largely ineffectual against these attacks. Further, it is interesting to note the dearth of substantive work in the area of insider threat detection that takes an approach other than statistical anomaly detection. This is important because while anomaly detection has shown some promise in authentication (i.e., distinguishing legitimate user account activity from that of a masquerader), there is in fact little evidence that the assumption underpinning broader statistical anomaly detection (namely that detectable anomalies are strongly correlated with intrusive behavior) holds outside of software process monitoring.

In this paper we describe a unique approach to insider threat detection that neither relies on codified attack signatures nor on unreliable statistical anomaly detection, and targets the specific problem of detecting unauthorized access and distribution of sensitive information.

Of course, substantial progress has been made in the development of Digital Rights Management (DRM) solutions such as Authentica [1] which seeks to allow information owners active control over who can access, edit, copy, forward, and print documents even after they have been disseminated. With Authentica, documents are encrypted at rest on a server and can be downloaded by any user with access to that server. When opened, users are authenticated and decryption keys are sent to the client program to provide the user with access to the document's content. Through this scheme, the document's owner can at any time revoke the user's privileges to a document – preventing it from being opened again.

But while Authentica improves an organization's ability to control document access, it is by no means a complete secure knowledge management solution. As pointed out by Bruce Schneier, while Authentica's encryption scheme is quite secure, it relies on a trustworthy client, which can not be ensured in the case of an insider [19]. Further, it is clear that even if used extensively, Authentica could not prevent the unauthorized access or distribution of information through social engineering, masquerading, hard copy distribution, etc. Therefore, the deployment of such DRM solutions does not remove the need for methods of identifying the unauthorized access and distribution of sensitive information.

In our approach we seek to detect these untrustworthy activities by monitoring key "bottlenecks" through which information (i.e., documents) must pass in order to be exploited by the insider and by detecting the presence of potentially restricted content (i.e., strings of words that overlap with a document the user should not have access to). Our system, Stochastic Long String Analysis with Feedback (SL-SAFE), then screens out false alarms by searching for the identified strings in public sources of information. SL-SAFE further seeks to reduce false alarms by accepting feedback from document authors to incrementally learn which elements of a restricted document are not in fact sensitive.

The remainder of the paper is organized as follows. In Section 2 we present a discussion of related work. Then in Sections 3 and 4 we discuss the SL-SAFE approach and present some experimental results. We then conclude with a discussion of the limitations of our work and future directions.

2. RELATED WORK

In our research effort we took substantial inspiration from Lippmann's concept of Bottleneck Verification [15]. In particular, we sought a means of making high accuracy judgments regarding the appropriateness of insider actions and circumventing the long standing difficulties observed in applying more traditional intrusion detection methods. Where Bottleneck Verification seeks to detect a user's unexpected ability to execute commands requiring root privileges (without regard to how that capability was achieved), we seek a similarly exploit-independent means of detecting the unexpected ability to access restricted document content.

Our SL-SAFE system differs substantially from other efforts that purport to support the detection of insiders. NIDES [10] and EMERALD [18] are perhaps the best known examples of intrusion detection systems that utilize statistical profiling of computer users in the hopes of detecting previously unknown attacks and insider abuse that would not match attack signatures. In fact some have suggested that anomaly detection may be the only way to detect insiders [10]. Unfortunately little evidence exists that this approach can be made feasible given the mercurial nature of everyday, honest work.

Similarly, while Lane & Brodley [12] have demonstrated some success in detecting masqueraders by modeling users' typical UNIX command sequences and then detecting significant deviations, they were not able to provide clear evidence that this method supports insider threat detection. These results are substantially inline with the work by Goldring [7] and Yihua Liao [14] that applied similar approaches to modeling Microsoft Windows usage. Goldring himself has recently expressed serious doubt regarding the applicability of these anomaly detection techniques outside the task of authentication [8].

Other uses of anomaly detection have demonstrated substantially lower false positive rates by focusing their attention on modeling activity that is much more regular than general computer usage. Two examples include the DEMIDS system that seeks to identify anomalies in database access patterns for users with specific work roles [2]. Even more effective have been the application of anomaly detection methods to software process monitoring [5][13][17]. Although these methods do not specifically target insiders they are valuable tools for maintaining system security.

Lundin and Jonsson [16] provide a good discussion of why typical anomaly detection is not terribly useful for detecting system misuse. One of their more interesting observations is that a malicious insider could easily trick these systems into generating "masquerader" alerts in order to achieve a form of plausible deniability.

Finally, our approach is related to recent work done in the area of plagiarism detection. While our approach appears similar on the surface, the unique qualities of the insider detection problem mandate a unique approach. Hoad and Zobel [9], for instance, focus primarily on establishing document-to-document similarity between what are assumed to be virtually identical documents (changes due to small corrections, revisions, or reorganization). Their system is incapable of making a "yes or no" determination with regards to plagiarized content and instead returns a ranked list of most similar documents – an approach wholly unsatisfactory for generating intrusion alerts.

At the other end of the spectrum is the Turnitin product [20] which is closer to our own concept – extracting long strings of words from a submitted document and searching the Web for instances of those strings in order to identify the source of likely plagiarized material. While this approach can be quite effective when evaluating term papers, it does not account for the common (and legitimate) replication of content that may occur extensively within an organization's documents (e.g., corporate boilerplate).

3. SL-SAFE

3.1 Assumptions

The SL-SAFE approach to detecting the misappropriation of information by insiders makes two fundamental assumptions. First, it is assumed that some collection of documents has been identified as "access restricted" (subsequently referred to as *Restricted*) and that we know who has legitimate privileges to access that information. Second, we assume that the malicious insider will transport, view, or (at least temporarily) store the *Restricted* information within the monitored environment during the attempt to exploit it. This second assumption then allows us to focus on bottlenecks that we may monitor, including: transport over a network, writing to a hard drive, reading/writing through desktop applications (e.g., MS Word and Adobe Acrobat), and writing to removable media. Of course none of these monitoring points will be useful in all situations (e.g., due to encryption) and each has its own unique limitations and requirements, but the potential redundancy acts to prevent circumvention by the insider.

3.2 The Content Monitoring Strategy

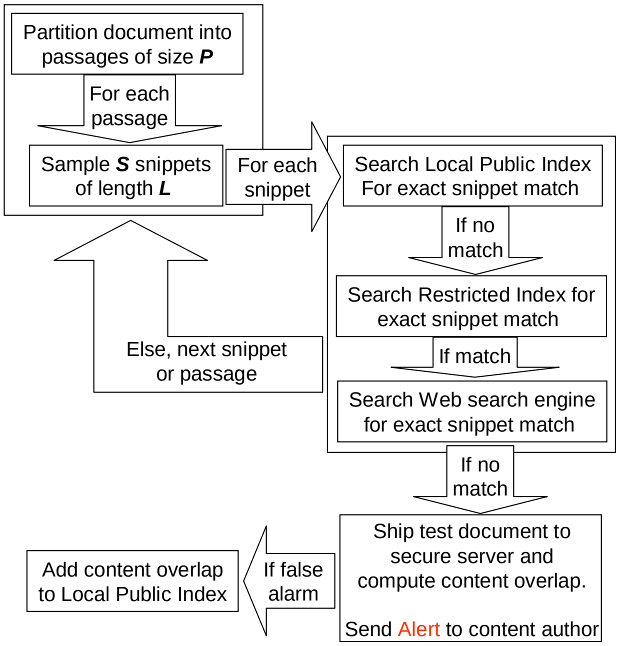
Partition document into passages of size ***P***
For each passage
Sample ***S*** snippets of length ***L***
For each snippet
Search Local Public Index For exact snippet match
If no match
Search Restricted Index for exact snippet match
If match
Search Web search engine for exact snippet match
Else, next snippet or passage
If no match
Ship test document to secure server and compute content overlap.
Send Alert to content author
If false alarm
Add content overlap to Local Public Index

Figure 1. The SL-SAFE Algorithm

Given a secure index of *Restricted* documents, an index of unrestricted documents (henceforth called *Public* documents), and a means for monitoring the creation, access, and distribution of documents, the SL-SAFE approach to content monitoring can be summarized as follows. When a new document is detected by one of our sensors, SL-SAFE processes it as follows:

1. The *Test* document is divided into **passages** of a fixed length *P* (e.g., 250 words). This initial partitioning of the text is done to distribute the following sampling in order to mitigate the threat that savvy insiders might somehow hide *Restricted* content within otherwise unremarkable text.

2. From each passage, *S* **snippets** (e.g., 3) of fixed length *L* (e.g., 5 words) are randomly selected.

3. SL-SAFE then searches for each of the *S* snippets in a Local index of *Public* information. This local index will contain the organization's documents and document fragments (e.g., corporate boilerplate) that are deemed non-sensitive.

4. If the snippet is not found within the local *Public* index, then our suspicions remain, and SL-SAFE searches the index of *Restricted* documents.

5. If the snippet is found within the *Restricted* documents, then SL-SAFE may attempt one more time to explain away the match by conducting a Web search.

6. If any snippet is found in *Restricted* documents (to which the suspect user does not have legitimate access), but not in any *Public* source, then SL-SAFE generates a complete list of overlapping text between the *Test* document and the identified *Restricted* document. This list is then delivered to the owner (or owning organization) of the *Restricted* document as an **Alert**. Note that the alert contains no document content that the Author did not originally include in their document.

7. If an alarm is found to be unjustified (meaning that the overlapping text did not indicate to the owner that an unauthorized access had occurred) then the overlapping text is added to the Local *Public* store to prevent future false alarms related to this content. Note, only the overlapping text is added to the *Public* store (not the whole document).

3.2.1 Algorithm Discussion

As justification for this algorithm let us discuss three issues/concerns/limitations in some detail.

1. Defeating SL-SAFE by adding restricted content to the *Public* store. One can imagine scenarios in which a malicious insider might be able to access *Restricted* content and publish it to the *Public* store prior to the registration of the original document as *Restricted* or before the unauthorized access is detected. The effect would be that SL-SAFE would disregard future detections of the *Restricted* content. However this threat can be mitigated by extracting even longer strings (say 7 words), and therefore even less likely to appear at random, from newly added *Restricted* documents (perhaps with the help of the author) which could be used to check (and then periodically monitor) the public store for wholesale additions of *Restricted* content to the *Public* store. This approach will not allow SL-SAFE to attribute the release to a particular insider, but it can allow the author of the *Restricted* content to recognize that a breach has occurred.

2. Security of indexes and queries. One might be concerned that indices of *Restricted* content might be attacked directly or that queries to the search engines might be vulnerable to snooping. To address these concerns we can protect indices of *Restricted* content

by employing privacy preserving indexing methods [2], and queries to search engines can themselves be encrypted. This of course means that a deployed system could not use existing Web search engines (unless a specialized interface is created), but that is not a significant concern.

3. String selection. It might appear initially that our sample of strings must be very carefully selected in order to achieve good results. This might involve any number of techniques to identify particularly important text elements in the document. We instead chose to use randomly selected long strings for two reasons. The first reason was the need to scale to large information environments which required that we not incur any unnecessary computational costs. The second reason, which is more important, is that, in early tests randomly selected long strings demonstrated high discriminative poser (i.e., the majority of five word strings sampled from non-public documents returned zero hits on the Google search engine).

4. Obfuscating the Text. The insider might use a number of methods to disguise the content he is attempting to misappropriate or disseminate, including:
   - Substituting key words
   - Hiding the content in unrelated material
   - Character-level mangling
   - Encryption
   - Conversion to Image

   These risks can be largely mitigated by the pervasive deployment of sensors that can detect new information content prior to substantial manipulation. Sensors embedded in document viewers or monitoring computer ports and hard drives could all be used for timely detection. Of course, the heavy or unusual use of encryption and screen capture might provide a reliable indication that the user is seeking to circumvent the system.

5. Use of Indirect Channels: The insider could use less direct methods of obtaining content, e.g., hardware-based attacks (keyboard capture, disk readers, etc.) and stealing document hardcopies. Obviously, depending on what the insider does with the content once he has obtained it, this may or may not fall outside the scope of our system. In the situation where some of this content makes its way through a monitored bottleneck, our approach will offer some capability to detect the activity (see Section 4.2).

4. EXPERIMENTS & DISCUSSION

Our intuition upon the formulation of this algorithm was that it would endow SL-SAFE with a number of interesting and attractive characteristics. This intuition led us to establish two primary hypotheses to investigate.

Hypothesis 1: Given a sufficiently large and representative index of *Public* documents, the false alarm rate should be expected to approach zero.

False alarms are generated when SL-SAFE encounters unrestricted text that it has not seen before. So, with a sufficiently robust starting set of *Public* documents, the probability of encountering text that is not in the *Public* index should be low, and should approach zero as it learns through feedback.

Hypothesis 2: SL-SAFE can reliably detect relatively small passages of *Restricted* text hidden within documents that are otherwise comprised of *Public* content.

Our hope here was that the simple steps taken to distribute SL-SAFE's text sampling across the document would make it difficult to simply hide *Restricted* content within a larger document.

4.1 Experiment 1

4.1.1 The Data Set

The data set we utilized was composed of research proposals written by employees of Sottler Henke Associates. In particular the data was partitioned based on the hypothetical scenario in which the Seattle office of Stottler Henke sought to restrict access to proposals related to its emerging *Aware* technology.

1. The *Restricted* document set was composed of 11 *Aware* related proposals written in 2002 and 2003.
2. The *Public* document set was composed of 15 proposals unrelated to the *Aware* technology written in 2002 and 2003 by researchers involved in *Aware*'s development.
3. The set of *Test* documents was composed to approximate a realistic distribution in a corporate environment:
   a. 1 proposal taken from the *Restricted* Set
   b. 3 Aware-related proposals from 2004
   c. 1 non-*Aware*-related proposal from 2004 written by the researchers involved in *Aware*'s development.
   d. 39 proposals from 2002-2004 written by Stottler Henke researchers *un*involved with *Aware*.

Also note that instead of explicitly manipulating documents to simulate the obfuscation tactics of an insider we relied on the fact that the 3 *Aware*-related proposals had only limited overlap with the *Restricted* document set (the largest contiguous overlap of *Restricted* content was under 50 words within proposals that average approximately 12,000 words).

4.1.2 Methodology

We established a testing paradigm that simulated the process of multiple users accessing documents within the test set in a uniform random pattern over time. Specifically, in each of three test runs, we randomly selected *Test* documents (with replacement) a fixed number of times (1000). During each test we allowed SL-SAFE to accumulate feedback by simulating the responses made by the owner of the *Restricted* documents that were found to overlap suspiciously with the *Test* documents.

Space limitations prevent a full discussion of the range of experiments we conducted, but a representative example utilized passages of 250 words, and 3 snippets of 5 words per passage. Note that these settings directly control the "sensitivity" of the system in detecting *Restricted* content hidden in larger texts, thus controlling the accuracy of alerts as well as how quickly the system learns to reduce false alarms through feedback.

4.1.3 Results

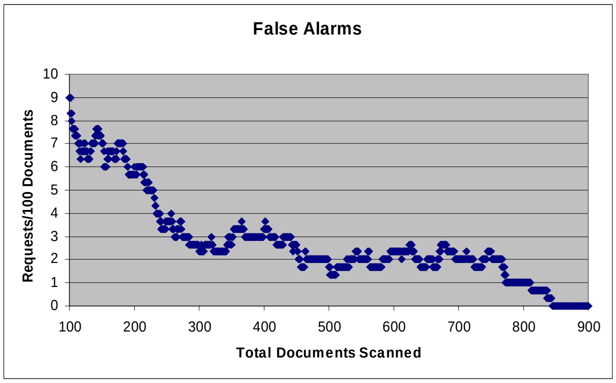

Figure 2. Number of false alarms (feedback requests) observed in a sliding window of 100 test documents.

In this experiment we found that 85.7% of the documents that should have been marked *Restricted* were correctly classified; and, as shown in Figure 2, the false alarm rate falls rapidly as early feedback is accumulated, and proceeds to zero. In fact we found that over the span of the experiment, only an average of 2.6 (and a median of 1.67) false alarms (feedback requests) were generated per *Test* document. Given the small size of the starting *Public* set, these figures imply that the time and effort required on the part of document owners should be quite low in practice. Further, these figures could easily be improved through document preprocessing steps that could identify much of the unrestricted text in documents at the time they are added to the *Restricted* index.

4.2 Experiment 2

While the previous experiment approximated the conditions under which we originally intended to utilize SL-SAFE, we saw an opportunity to tackle an even harder problem. In particular, we wanted to determine if SL-SAFE could be used to reliably detect paraphrased content drawn from *Restricted* documents.

In this second experiment we asked four test subjects to pretend to engage in corporate espionage. Each subject was given 30 minutes to review an *Aware* related proposal that was left open on a colleague's computer and take written notes regarding critical details of interest to a competitor. Based on these written notes, the test subjects authored an email intended to be sent to an outside contact. We added these emails to the set of test documents utilized in the previous experiment and ran an additional three runs.

The results of this experiment were very encouraging. We found a hit rate across our test subjects were 18%, 22%, 55%, and 64%. These figures are surprisingly high given that each test subjects used a variety of shorthand and ungrammatical sentences in their emails. The explanation appears to be that there were certain portions that were far easier to repeat verbatim than to paraphrase – especially under time pressure.

5. CONCLUSIONS AND FUTURE WORK

SL-SAFE represents a scalable and effective means for detecting the misappropriation or unauthorized distribution of *Restricted* content by insiders, without requiring the ability to detect the precise means of attack. SL-SAFE succeeds in detecting these activities through the stochastic sampling of information passing through bottlenecks while maintaining low false alarm rates by validating suspicions through public information sources and by soliciting limited feedback from the information owner.

Clearly the proposed approach does not represent a complete solution to the problem of unauthorized information access and distribution by insiders. And we recognize the risk that savvy insiders will know that their activities are being monitored, and may even have some idea of the nature of the monitoring (i.e., content-based). Given this, SL-SAFE represents a new and substantial impediment to insiders.

While our experiments have proved successful, there remain a large number of potential improvements that can be made to the core SL-SAFE approach in order to decrease false alarms, including more front loading of the solicitation of feedback from document owners, and being more intelligent about the selection of text samples. In particular we are now experimenting with a new lightweight sampling technique that rejects a sampled snippet if it is composed entirely of very common (stop) words. We are also seeking to optimize the computational bottleneck in our prototype – document overlap calculation.

Finally, we will begin to address the larger architectural issues required to field SL-SAFE and conduct live exercises. This will involve working out schemes to maintain and protect sensors, maintain the index of protected documents and associated access lists, coordinate alerts to document authors, etc.

6. ACKNOWLEDGMENTS

This work was supported by the Advanced Research and Development Activity (ARDA)'s Advanced Countermeasures for Insider Threat (ACIT) program under contract NBCHC030077.

7. REFERENCES

[1] Authentica. http://www.authentica.com

[2] Bawa, Bayardo, and Agrawal. "Privacy-Preserving Indexing of Documents on the Network." VLDB 2003: 922-933

[3] Chung, C, Chung, Gertz, M., and Levitt, K. "DEMIDS: A Misuse Detection System for Database Systems." In Proceedings of the 3rd International Working Conference on Integrity and Internal Control in Information Systems (IICIS'99).

[4] 2003 CSI/FBI Computer Crime and Security Survey. A copy can be obtained from CSI at their Web site http://www.gocsi.com/

[5] Forrest, S., Hofmeyr, S. A., Somayaji, A., and Longstaff, T. A., "A Sense of Self for Unix Processes," presented at Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, CA, 1996.

[6] Gartner Inc., Estimating Loss From Infrastructure Compromise: A Model

[7] Goldring, T. Recent experiences with user profiling for windows nt. In Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection, Johns Hopkins University, 11-13 June 2002. http://www.mts.jhu.edu/cidwkshop/.

[8] Goldring, T. Presentation at the ARDA-sponsored Advanced Countermeasures for Insider Threat (ACIT) Program Kick-off Meeting.

[9] Hoad, T.C. and Zobel, J. "Methods for identifying versioned and plagiarized documents" Journal of the American Society for Information Science and Technology. 2003.

[10] Jagannathan and Lunt. "System Design Document: Next-generation Intrusion Detection Expert System (NIDES). Technical report, Computer Science Laboratory, SRI International, Menlo Park, California, 1993

[11] Javitz, H. and Valdes, A.: The NIDES Statistical Component: Description and Justification, SRI International, 1993.

[12] Lane and Brodley. "Temporal Sequence Learning and Data Reduction for Anomaly Detection," ACM Transactions on Information and System Security, 2(3), 1999

[13] W. Lee, S. J. Stolfo, and P. K. Chan. Learning patterns from UNIX process execution traces for intrusion detection. In AAAI Workshop: AI Approaches to Fraud Detection and Risk Management, pages 50-56. AAAI Press, July 1997.

[14] Yihua Liao, "Windows NT User Profiling with Support Vector Machines", in Proceedings of the 2002 UC Davis Student Workshop on Computing, Technical Report CSE-2002-28, Dept. Computer Science, UC Davis 2002.

[15] Lippmann, R. P., Dan Wyschogrod, Seth E. Webster, Dan J. Weber, and Sam Gorton, Using Bottleneck Verification to Find Novel New Attacks with a Low False-Alarm Rate, First International Workshop on Recent Advances in Intrusion Detection, Louvain-la-Neuve, Belgium, 1998.

[16] Lundin, E. and Jonsson, E. Some practical and fundamental problems with anomaly detection. In Proceedings of the Fourth Nordic Workshop on Secure IT systems, Kista, Sweden, November 1999.

[17] Nguyen, N., Kuenning, G., and Reiher, P., "Detecting Insider Threats by Monitoring System Call Activity," IEEE Information Assurance Workshop, 2003.

[18] Porras, P. and Neumann, P. EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances. National Information Systems Security Conference. October, 1997.

[19] Schneier, B., "The doghouse: Authentica," Crypto-Gram, August 15, 2000, http://www.schneier.com/crypto-gram-0008.html, Last accessed August 19th, 2004.

[20] Turnitin. http://www.turnitin.com