

Real-Time Intrusion Detection with Emphasis on Insider Attacks

Shambhu Upadhyaya

University at Buffalo, Buffalo, New York, 14260, USA
shambhu@cse.Buffalo.EDU

1 Introduction

Securing the cyberspace from attacks is critical to the economy and well being of any country. During the past few years, threats to cyberspace have risen dramatically. It is impossible to close all security loopholes in a computer system by building firewalls or using cryptographic techniques. As a result, intrusion detection has emerged as a key technique for cyber security. Currently there are more than 100 commercial tools and research prototypes for intrusion detection. These can be largely classified as either misuse or anomaly detection systems. While misuse detection looks for specific signs by comparing the current activity against a database of known activity, anomaly detection works by generating a reference line based on the system model and signaling significant deviations from it as intrusions. Both approaches rely on audit trails, which can be very huge. Moreover, conventionally they are off-line and offer little in terms of strong deterrence in the face of attacks.

In this talk, we will examine the intrusion detection tools and techniques from a taxonomical point of view and study the real-time properties and applicability to real systems and their shortcomings. Following the overview, we will present our own cost analysis-based framework, which quantifies and handles both misuse and anomalies in a unified way. Decisions regarding intrusions are seldom binary and we have developed a reasoning framework that makes decisions on a more informed basis. The overall reference graph is based on the user's profile and the intent obtained at the beginning of sessions. The uniqueness of each user's activity helps identify and arrest attempts by intruders to masquerade as genuine users, which is typically the case in insider attacks. We will examine this work and present some results.

2 Brief History

The goal of intrusion detection system (IDS) is to monitor network assets to detect misuse or anomalous behavior. Research on intrusion detection started in 1980 as a government project and under the leadership of Dorothy Denning at SRI International, the first model for intrusion detection was developed in 1983 [3]. Earlier versions of intrusion detection systems were largely host-based and the work of the "Haystack project" led to network level intrusion detection systems. Commercial intrusion detection systems were introduced in the 90's and today there are more than 100 tools and prototypes that can be purchased or experimented with. Most of these tools work on audit trail data or data packets obtained across the network.

First generation intrusion detection tools are essentially misuse signature-based and security is accomplished by iterative “penetrate and patch”. Today’s focus is on detecting novel intrusions. With the proliferation of the Internet and the increased power of the attacker, short term solutions and tools not very useful and new solutions must consider insider attacks, social engineering based break-ins and the myriad ways of perpetrating an attack. Some of the new ideas include combining intrusion detection with vulnerability analysis and considering recovery along with detection since detection is not fool proof.

3 Intrusion Detection Taxonomy

Debar, Dacier and Wespi gave the first taxonomy of intrusion detection systems [2]. They used Detection Method, Behavior on Detection, Audit Source Location, Detection Paradigm and User Frequency as the parameters of classification. Upadhyaya and Kwiat introduced a variation of the taxonomy and presented it in a tutorial in IEEE MILCOM 2002. According to this taxonomy, intrusion detection techniques and tools are classified along four major parameters as follows: (1) Detection Methodology — this classification uses information about attacks versus information about normal functioning of the system; (2) Scope — host based versus network based systems; (3) Monitor Philosophy — passive versus proactive; and (4) Monitor level — kernel level versus user operation level. Detection methodology also depends on the reasoning philosophy. For example, rule-based versus model-based. Host based IDS tools work on the host, can detect masquerade, account break-in etc. whereas Network based tools are applicable to large-scale networks and depend on information on network packets. Passive tools are noninvasive, non-intrusive but mostly are after-the-fact and use no communication with users whereas Proactive schemes are real-time, concurrent detection tools with low latency and employ user interrogation where needed. Finally, the Kernel level tools make use of low level information and process data to synthesize the attack scenarios whereas User operation level tools can capture user semantics and are capable of detecting subtle intrusions.

4 Insider Attacks

An insider threat is one in which someone with an authorized access to the organization could cause a loss to the organization if computer security went unchecked. The perpetrators are those who work for the target organization or those having relationships with the firm with some level of access. It could be employees, contractors, business partners, customers etc. The motives could range from financial, social, political to personal gains. There are two classes of insiders — logical insiders who are physically outside and physical insiders who are logically outside. The misuse could be intentional or accidental, obvious or hidden. Here are a few insider attacks that made headlines.

- An individual faces federal criminal charges in US District Court in Miami for allegedly downloading a virus into his employer’s computer system, crashing the network for nearly two full days (NIPC Daily Report, Aug. 29, 2001).

- Former programmer Timothy Lloyd, who was fired in 1996, retaliated by setting off a logic bomb that destroyed employer Omega Engineering's primary database, causing \$12 million damage and forcing 80 people to be laid off (Security Wire Digest, Vol. 3, No. 12, Feb. 12, 2001).
- Feds charge 3 in massive credit fraud scheme. Initial losses estimated at \$2.7 million (CNN.com, 2002).

CSI/FBI 1999 Computer Crime Survey indicated that 55% of the reported attacks were from insiders. CSI/FBI 2000 Computer Crime Survey stated that 71% of the respondents had been the victim of internal attacks. Dealing with this problem involves three steps: modeling the insider, prevention of internal misuse and detection, and analysis and identification of misuse. Approaches to prevention are to install and execute appropriate anti-virus tools, install software updates and patches, encrypt databases, key system files and even executables, electronically "watermark" documents so that their passage through any electronic gate can be automatically detected and prevented and isolate privileged execution domain from less privileged execution domains and implement multilevel security policies.

5 Surveillance Issues

The questions that need to be addressed to mitigate insider attacks are: what kind of model one should develop?, should we consider prevention or detection or both?, should the method be passive or proactive? For optimizing detection, the technology must be tamper-resistant, must not burden the monitoring system and must be cost-effective. In this talk, we consider just the detection problem. The insider attack detection approaches are generally anomaly-based and could range from rule-based detection, to statistical anomaly detection and proactive schemes such as query-based encapsulation of owner intent [4].

6 A New Proactive Scheme for Insider Threat Detection

Our approach relies on user level anomaly detection that avoids after-the-fact solutions such as audit data analysis. We leverage ideas from fault tolerance where concurrent monitoring is used to detect control-flow errors. We capture owner's intent and use it as a reference signature for monitoring and a reasoning framework is developed for making rational decisions about intrusions. Certain engineering methodologies such as Divide and Conquer are used to address scalability of the approach.

We obtain the reference graph by *Encapsulation of owner's intent*, which requires an implicit or explicit query of users for a session-scope. We then translate it into a set of verifiable assertions. Actual operations are monitored at user command level and the user behavior is assessed. The advantages of this approach are: no need to process huge audit data and both external/internal abuse can be handled uniformly [4]. Reasoning about intrusions is done by a stochastic modeling of job activity. A double-threshold scheme is used to resolve situations arising when job activity cost maps into an ambiguous region. Cost gradients are used to shrink the window of uncertainty so that a speedy decision on intrusion can be arrived [5]. Anomaly detectors are always

faced with the scalability problem due to the large number of partial orderings of operations. While no improvements over worst-case complexity exist, we employ engineering approaches such as divide and conquer and use the concept of job priorities, user workspaces and meta-operations for monitoring [1]. We are currently building a prototype of the system to demonstrate the proof-of-concept.

References

1. Chinchani, R., Upadhyaya, S., and Kwiat, K.: Towards the scalable implementation of a user level anomaly detection system. *IEEE MILCOM 2002*, Anaheim, CA (October 2002)
2. Debar, H., Dacier, M., and Wespi, A.: Towards a Taxonomy of Intrusion Detection Systems. *Computer Networks*, 31 (1999) 805–822
3. Denning, D.: An Intrusion-Detection Model. *IEEE Transactions on Software Engineering*, Vol. SE-13, No. 2 (February 1987) 222–232
4. Upadhyaya, S. and Kwiat, K.: A distributed concurrent intrusion detection scheme based on assertions. *SCS Int. Symposium on Performance Evaluation of Computer and Telecommunication Systems*, Chicago, IL (July 1999) 369–376
5. Upadhyaya S., Chinchani, R., and Kwiat, K.: An analytical framework for reasoning about intrusions. *IEEE Symposium on Reliable Distributed Systems*, New Orleans, LA (October 2001) 99–108